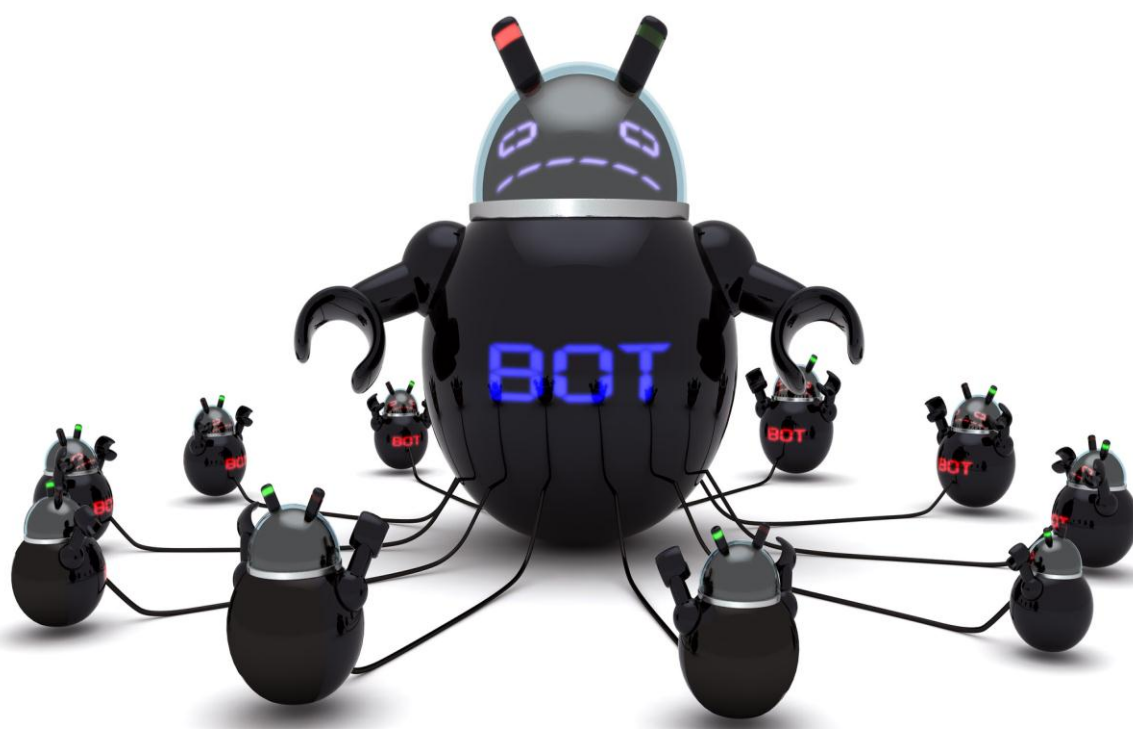




Botnets: Detection, Measurement, Disinfection & Defence



ABOUT ENISA

The European Network and Information Security Agency (ENISA) is an EU agency created to advance the functioning of the internal market. ENISA is a centre of expertise for the European Member States and European institutions in network and information security, giving advice and recommendations and acting as a central source of information on good practice. Moreover, the agency facilitates contacts between European institutions, the Member States, and private business and industry players.

This work takes place in the context of ENISA's Emerging and Future Risk programme.

CONTACT DETAILS

Editor: Dr. Giles Hogben giles.hogben [at] enisa.europa.eu
Internet: <http://www.enisa.europa.eu>

Authors: Daniel Plohmann daniel.plohmann [at] fkie.fraunhofer.de
Elmar Gerhards-Padilla elmar.gerhards-padilla [at] fkie.fraunhofer.de
Felix Leder felix.leder [at] fkie.fraunhofer.de

Legal notice

Notice must be taken that this publication represents the views and interpretations of the editors, unless stated otherwise. This publication should not be construed to be an action of ENISA or the ENISA bodies unless adopted pursuant to ENISA Regulation (EC) No 460/2004. This publication does not necessarily represent the state-of-the-art in botnet measurement, defence and disinfection and it may be updated from time to time.

Third-party sources are quoted as appropriate. ENISA is not responsible for the content of the external sources, including external websites referenced in this publication.

This publication is intended for educational and information purposes only. Neither ENISA nor any person acting on its behalf is responsible for the use that might be made of the information contained in this publication.

Reproduction is authorised provided the source is acknowledged.

© European Network and Information Security Agency (ENISA), 2011

LIST OF CONTRIBUTORS

This report is the result of a project group consisting of representatives from ENISA, Fraunhofer FKIE, and University of Bonn, Germany, using input and comments from a group selected for their expertise in the subject area, including industry, academic and government experts.

Andre Ludwig	
Angelo Dell'Aera	Security Reply
Ben Stock	University of Mannheim
Christian Czosseck	Cooperative Cyber Defence Centre of Excellence
Christopher Elisan	Damballa, Inc.
Dave Dittrich	Applied Physics Laboratory, University of Washington
Dave Woutersen	GOVCERT.NL
David Barroso	S21sec
Damian Menscher	Google
Ivo Ivanov	eco-Association of the German Internet Industry
Jan Göbel	University of Mannheim
Jorge Mieres	
Jose Nazario	Arbor Networks
Kandy Zabka	Komfort Linux, Inc.
Kushan Sharma	TechCERT / A Division of LK Domain Registry, Sri Lanka
Mahmud Ab Rahman	MyCERT, CyberSecurity Malaysia
Marco Riccardi	The Italian Honeynet Chapter and Barcelona Digital
Mikko Hypponen	F-Secure
Nicolas Fischbach	Colt
Pierre-Marc Burreau	ESET
Raoul Chiesa	@ Mediaservice.net
Risto Vaarandi	Cooperative Cyber Defence Centre of Excellence
Shane Shook	
Sven Karge	eco-Association of the German Internet Industry
Tobias Knecht	abusix.org
Vincent Hinderer	CERT-LEXSI
Yuan Xu	CNCERT/CC

The above list contains only experts who have permitted the publication of their contribution. The views expressed in this publication are those of the editor, unless stated otherwise, and do not necessarily reflect the opinions of the participating experts.

EXECUTIVE SUMMARY

Botnets are networks of compromised, remotely controlled computer systems. So far, their main purposes include the distribution of spam e-mails, coordination of distributed denial-of-service attacks, and automated identity theft, e.g. credit card information and general banking data for financial fraud. Their presence is supported by the increasing global availability of broadband access to the Internet for network-enabled devices, which at the same time increases the value of the assets they threaten.

A shift in the motivation for the creation of malicious software has led to a financially-oriented underground economy of criminals acting in cyberspace. The total annual global economic loss attributed to malicious software activities is estimated at more than US\$ 10 billion [1]. Other events demonstrate the significance of botnets as a threat to national security as, for example, the botnet-driven attacks against Estonia in 2007, against Georgia in 2008, and Iran in 2009.

The following key conclusions regarding the current situation of botnets have been drawn in this report:

- Existing approaches to measuring the size of botnets commonly lack accuracy, in that the numbers produced are only reliable to a very limited degree. Additionally, statements about botnet sizes seldom include a clear mention of applied measurement methodology or provide a scientific basis. Furthermore, it is important to note that the size of botnets alone is only one factor in assessing the threat caused by them.
- Instead of a generic threat measurement for assessing botnets, a view divided into stakeholder perspectives is proposed. Different types of botnets represent different functionality. Furthermore, the possibility of remotely updating malware may extend a botnet's capabilities rapidly, multiplying the threat.
- The current legal frameworks of various EU Member States and their national diversity in the context of cybercrime are a key factor in the efficiency of the fight against botnets. The applicability of promising detection and mitigation approaches is also limited through certain conflicts between data protection laws and laws that ensure a secure operation of IT services. Finally, working processes increase the reaction time to the extent that they can be evaded with little effort by criminal individuals, capitalising on the ease with which botnets can be configured. For more information on the legal issues identified in the context of botnets, see [2].
- The global botnet threat is best countered by close international cooperation between governments and technically-oriented and legislative institutions. For an efficient supranational mitigation strategy to work, cooperation between stakeholders must be intensified and strengthened by political will and support.

In this context, the standardisation of processes for information exchange plays an important role. This includes reports about incidents, identified threats, and evidence against criminal individuals, ideally leading to their arrest, as well as mechanisms for maintaining the confidentiality of shared information and establishing the trustworthiness of its source.

The spread and success of botnets are affected by a number of factors:

- The ease and cost of infecting a user's PC with malware.
- The profit which can be gained by running a botnet (which is related to the effectiveness of defensive measures against up-and-running botnets).
- The probability and severity of criminal sanctions against the perpetrator.

This report surveys and analyses different approaches to the detection, measurement, disinfection and defence against botnets that address the above factors. These approaches have been identified through a survey among international experts from different stakeholder groups, including academia, security solution vendors, computer emergency response teams, Internet Service Providers, regulatory bodies and law enforcement. In addition, further relevant topics in the context of botnets were intensively debated in a group discussion phase [3]. Detailed desktop research supplements the findings.

The results are divided into techniques for *detection and measurement* on the one hand and *countermeasures against botnets* on the other. The former have been grouped into passive and active methods; the latter have been split into technical, social and regulatory approaches.

The presentation of recommendations and good practice, derived from the analysis of approaches, is divided into 3 generic objectives and specific recommendations for 4 stakeholder groups, namely regulators and law enforcement, Internet Service Providers, researchers, end-users and companies.

The three high-level objectives for engaging the botnet threat are:

- **Mitigation of existing botnets:** Primarily, this requires a reduction of existing infections; to achieve this it is crucial to:
 - Support owners of compromised computers in this non-trivial task. This means Internet Service Providers should be more strongly incentivised to make use of their position as an access channel to the Internet for their customers and to use their special position for detection efforts.
 - Improve botnet identification and monitoring, and malware analysis.
 - Continue botnet takedown efforts.
 - Share information between responsible stakeholder groups.

- Harmonise laws against cybercrime at an international level to simplify cross-border activities.
- Perform takedowns only in cases where the whole botnet command-and-control infrastructure is covered, as even single, unaffected servers may provide the botmaster with a chance to regain full control and increase the resilience of the botnet against future takedown attempts.
- **Prevention of new infections:**
 - Public awareness has to be raised in particular towards social and civic responsibility for good security practice.
 - Vulnerability management and system protection should be improved, both by the manufacturers of devices and software and the users of these products, in order to slow down the spread of new malware and consequently botnets.
- **Minimising the profitability of botnets and cybercrime:**
 - With malware usage patterns shifting to financial benefits, countermeasures have to focus on the profit margins of cybercrime. The incentives for cybercrime may be reduced by tackling the overall value-creation schemes and improving anti-fraud mechanisms.
 - This should be consolidated on a social level through increased deterrence, obtained through tougher prosecution of cybercrime.

Top recommendations aimed at specific stakeholder groups:

- Regulators should:
 - Intensify their efforts to modernise existing legal frameworks and their interpretation on a national level in order to create a practical basis for dealing with different aspects of cybercrime. This include laws covering the creation and operation of botnets as well as the use of botnet services (such as the capability to send spam email and perform Distributed Denial of Service) and goods (such as stolen credentials and intelligence) provided by them.
 - Harmonise laws in a European context in order to facilitate mitigation processes and facilitate cooperation at an international level.
 - Establish a closely-linked, responsive network of responsible parties across Member States. Furthermore, the roles and responsibilities of affected stakeholders should be clearly defined.
- End-users have a social obligation to protect their systems against malware. Owing to the complexity of this task, they should be supported by various means:

- The raising of security awareness in general and awareness of civic responsibility.
 - Easily available guidance on the possible precautions against infections, as well as their detection and handling.
 - Notifying end-users about remotely identified infections through their Internet Service Providers is a useful and effective approach (although ISPs should be given appropriate incentives for bearing the costs of this activity).
- Research institutions should be more strongly integrated and, where appropriate, empowered in the fight against botnets. Research should focus on techniques which can be implemented in large-scale operational environments subject to typical cost constraints. They should be supported in studying methods for the detection of botnets and the analysis of malware, in order to provide efficient tools to reduce the reaction time when dealing with complex and sophisticated malware threats. As the results of research may be of interest for ongoing investigations, the process of publishing these results should reflect the responsibility associated with them.
- Information sharing between all responsible parties in the fight against botnets, including law enforcement, Internet Service Providers and research institutions, has to be further improved. The main objective is to support investigations, thereby solving key challenges such as:
 - Creating and maintaining trust between contributors
 - Efficiently processing and organising data gathered from different sources
 - Clearly defining data exchange formats tailored to the needs of participants.
 - The balance between, on the one hand, laws for the protection of privacy and data protection of users and, on the other, the protection of Internet security and the stability of critical infrastructures.

Table of Contents

About ENISA	2
Contact details	2
List of contributors	3
Executive Summary.....	4
1 Introduction and Fundamentals.....	10
1.1 Characterisation of Malware.....	11
1.2 Characterisation of Botnets.....	14
1.2.1 Components of Botnet Infrastructures	14
1.2.2 Motivation and Usage of Botnets	20
1.2.3 Attack Potential and Threat Characterisation	25
2 Methodical Approach of this Study	32
2.1 Information Gathering	32
2.1.1 Online Survey among Experts	32
2.1.2 Desktop Research	33
2.1.3 Group Discussions between Experts	33
2.1.4 Peer Review Group	34
2.2 Applied Features for Analysis.....	34
2.2.1 General Characteristics of the Approach	35
2.2.2 Quality of Results	36
2.2.3 Required Effort and Resources	38
2.2.4 Limitations	39
3 Measurement and Detection Techniques.....	40
3.1 Passive Techniques.....	40
3.1.1 Packet Inspection	40
3.1.2 Analysis of Flow Records	42
3.1.3 DNS-based approaches	44
3.1.4 Analysis of Spam Records	46
3.1.5 Analysis of (Application) Log Files	48
3.1.6 Honeypots	49
3.1.7 Evaluation of Anti-Virus Software Feedback	53
3.2 Active Techniques.....	54
3.2.1 Sinkholing	54
3.2.2 Infiltration	57
3.2.3 DNS Cache Snooping	57
3.2.4 Tracking of Fast-Flux networks	59
3.2.5 IRC-based measurement and detection	61
3.2.6 Enumeration of Peer-to-Peer Networks	62

3.3 Other	64
3.3.1 Malware Reverse Engineering	64
3.3.2 C&C Forensics and Abuse Desks	65
3.4 Analysis.....	65
3.4.1 General Characteristics	66
3.4.2 Quality of Results	68
3.4.3 Required Effort and Resources	72
3.4.4 Limitations	74
3.5 Conclusion	76
4 Countermeasures and Initiatives.....	79
4.1 Technical Countermeasures	79
4.1.1 Blacklisting	79
4.1.2 Distribution of Fake/Traceable Credentials	80
4.1.3 BGP Blackholing	81
4.1.4 DNS-based Countermeasures	82
4.1.5 Direct Takedown of Command-and-Control Server	84
4.1.6 Packet Filtering on Network and Application Level	85
4.1.7 Port 25 Blocking	86
4.1.8 Walled Gardens	87
4.1.9 Peer-to-Peer Countermeasures	88
4.1.10 Infiltration and Remote Disinfection	89
4.2 Regulatory and Social Countermeasures.....	90
4.2.1 Dedicated Laws on Cybercrime	90
4.2.2 User Awareness raising and Special Training	92
4.2.3 Central Incident Help Desk	93
4.2.4 Enhance Cooperation between Stakeholders	94
4.3 Initiatives and Institutions.....	95
4.3.1 National Initiatives	96
4.3.2 International Initiatives	99
4.3.3 Targeted Botnet Mitigation Workgroups	101
4.3.4 Other Public Private Partnerships (PPPs)	102
4.4 Analysis.....	103
4.4.1 General Characteristics	104
4.4.2 Quality of Results	107
4.4.3 Required Effort and Resources	112
4.4.4 Limitations	115
4.5 Conclusion	116
5 Recommendations for Good Practice.....	119
5.1 Assumed Threat Picture	119
5.2 Integrated Approach against the Botnet Threat.....	119
5.2.1 Mitigate Existing Botnets	121
5.2.2 Prevent New Infections	123
5.2.3 Minimise profitability of Botnets and Cybercrime	124

5.3	Recommendations for Stakeholder Groups	125
5.3.1	Recommendations for Regulators and Law Enforcement	126
5.3.2	Recommendations for Internet Service Providers	127
5.3.3	Recommendations for Researchers	128
5.3.4	Recommendations for End-Users and Companies	129
6	Future Trends	132
7	Appendix	136
7.1	Abbreviations	136
8	References	138

1 INTRODUCTION AND FUNDAMENTALS

Malicious software, ‘malware’ for short, has acquired an important position in modern high-tech life. Starting from the earliest use of programmable systems, approaches to infecting them with software containing malicious functionality have existed but, in the past, malware often had just limited or local impact. The success of the Internet also became a starting point for reports about widespread malware infections affecting several million systems around the globe. Almost concurrently, remotely controlled networks of hijacked computers, so-called botnets, became popular. One critical observation is that the motivation for creating malware has changed dramatically over the last decade. No longer is it the primary aim to establish a reputation within an almost mystical community of technically highly-skilled individuals. With the Internet easily accessible to everyone and the use of financially-oriented services such as electronic shopping and banking now widespread, casual users with minimal technical knowledge have become promising targets for criminals. Financial gain is now the leading motivation for criminal online operations and malware creation.

As a result, today’s global economies face a wide range of malware that, taken together, does a great deal of damage. But estimates of how much differ widely. An ITU report, giving an overview of financial studies about malware, presented numbers ranging from \$US13.2 billion for the global economy in 2006 to \$US67.2 billion for US businesses alone in 2005. Consumer reports estimated direct costs to US citizens of malware and spam at \$US7.1 billion in 2007. The cost of click fraud in 2007 in the US was estimated to be \$US1 billion [1]. While the number of malware samples has been increasing at an exponential rate over the last few years [4], Computer Economics has measured a declining worldwide impact of malware attacks on businesses, with financial costs of \$US17.5 billion in 2004, \$US14.2 billion in 2005 and \$US13.3 billion in 2006 [5]. Giving their reasons for this decline, Computer Economics argued that, in these years, anti-virus products have been deployed more widely in companies and that the goal of malware creators has shifted from creating havoc to direct financial gain. As a consequence, indirect and secondary costs incurred by malware are

increasing. At the same time, the shift of focus towards stealing information and credentials and the financial losses caused by, for example, credit card abuse have significantly increased the potential threat to private Internet users.

Different terms like virus, worm, Trojan horse, rootkit, and others have been established to classify and distinguish various types of malware, according to their functionality and purpose. The following section will give a brief overview of these different types and outline their relevance in the context of botnets. In addition, a more detailed description of how botnets work will be provided. Different kinds of infrastructure and significant attributes of recent botnet approaches will be introduced. Since many parties are involved in the production and exploitation of modern botnets, the economic motivations behind managing and renting a botnet will be outlined and the relevant economic drivers for the operation of botnets listed.

1.1 CHARACTERISATION OF MALWARE

In recent years, the diversity of malware has grown almost exponentially. New variants appear every day, with constant improvement to the techniques employed and a growing degree of sophistication. An often cited indicator of malware evolution is the number of samples of malware specimens or detection signatures generated by anti-malware software providers that have been discovered. For example, Symantec states that, in 2009, a total of 2,895,802 new signatures for the detection of malware were created, 51% of all the signatures ever created by them [6]. Kaspersky identified about 15 million unique samples of malware specimens in 2009, which means that one unknown sample was discovered roughly every 2 seconds [7].

The growth in the number of samples is symptomatic of the widespread application of the concept of polymorphism to binary malware files. Polymorphic malware contains a fixed code sequence that modifies the malware binary code during propagation but remains unchanged itself. Metamorphic malware is changed fully within every propagation attempt.

Polymorphism and metamorphism are just two examples of techniques employed by malware developers and users to reach their goals. The main intention, when developing malware nowadays, is to gain financial benefit, mainly through connections with various fraud schemes. In general, the motives behind the creation of malware are, for example, personal financial or material benefit, political interests or just interest in its technical possibilities.

COMPUTER VIRUS

Even though the term 'malware' groups together a whole family of software, related typically via their hostile and intrusive properties, different terms have been coined within this area. The predominant term to have emerged in conjunction with malware is the so-called computer virus. The term is often used to describe different types of malware, even if these do not match the attributes related to the actual definition of a

virus. Apart from its use as a catch-all phrase, a virus is one specific type of malicious software characterised by self-replication [8]. Every virus needs a host, e.g. an executable file, to exist and into which it will integrate itself. It spreads itself by copying itself to other host systems. Actual propagation among systems happens when infected files are transferred to other systems; a virus can therefore be characterised as passive. Depending on its development, a computer virus may use all kinds of different media for these reproductive abilities, such as network-based file systems or removable media.

COMPUTER WORM

In contrast to a computer virus a worm not only has the ability to copy itself to different mediums but is also able to spread itself actively. A computer worm is able to search autonomously for other computers within the network and infect them, if they have been identified as vulnerable. This is typically achieved by means of exploiting known or unknown vulnerabilities in the operating system or additional software installed in it. A worm does not necessarily contain destructive or intrusive routines that harm a victim's systems directly. Some worms are designed exclusively to spread and to eventually establish a communication channel with some controlling entity. In this context they serve as active carriers. However, it has to be stated that worms generally harm the system or the underlying network indirectly. By consuming resources such as computing power and network bandwidth, they often introduce instability into host systems. Continuous scanning activities, especially, consume a lot of resources and can affect the network significantly.

TROJAN HORSE

While worms operate silently and autonomously, a Trojan horse is a piece of malware that follows a different approach. In general, a Trojan horse hides malicious routines by pretending to be legitimate software performing bona fide tasks. By pretending to have a legitimate purpose, it tricks the user into installing or executing software containing the Trojan horse, which then loads the embedded malicious routines.

SPYWARE, KEYLOGGER, SNIFFER

A feature regularly present in malware is the ability to extract live data from a remote system. This is typically achieved by subverting functions at the operating system level. Subgroups of malware are named after their activity. Spyware is a generic term for software written with the intention of data extraction. This can range from monitoring a user's behaviour for the optimisation of advertisements to aggressive forms, like stealing serial numbers for software or other sensitive data, like credit card information. Malware that records keystrokes in order to capture credentials is usually called a keylogger. Tools originating from network analyses, which have been found useful in a malicious context, and eavesdrop on network traffic in order to filter it for credentials, are called sniffers.

ROOTKIT

In general, malware inhabits the victim's computer system silently, unbeknown to the system's owner (the root account). Even if the full effects of the activities of a malware instance cannot usually be hidden, the malware will try to stay as inconspicuous as possible. Different approaches exist to hide programs in the operating system's hierarchy. A frequently used term in this context is the so-called rootkit. Generally, a rootkit is a collection of tools that help developers prevent certain routines and processes from being detected or disabled. The idea behind a rootkit is to ensure the continuing presence of its own processes or to maintain access to a remote system. Usually, special privileges or specific functionality are enabled on the compromised system. Because of their invasiveness in the target system, rootkits are often difficult to remove.

HISTORIC EVENTS LEADING TO BOTNETS

The term bot is short for robot and refers to the clients in a botnet. It is derived from the Czech word "robota", which literally means work or labour. Alternative names for bots are zombies or drones. In the following, the historical origin of the most relevant concepts responsible for the appearance of botnets are explained.

Historically, the concept of bots did not include harmful behaviour by default. The term was originally used for control instances located in the chat rooms of Internet Relay Chat (IRC), which appeared from 1989 onwards. They were able to interpret simple commands, provide administration support or offer services like simple games to chat users. The first known IRC bot is Eggdrop [9], first published in 1993 and further developed since. Next, following the release of Eggdrop, malicious IRC bots appeared, adopting the basic idea, but created primarily in order to attack other IRC users or even entire servers. Shortly after, Denial of Service (DoS) and then Distributed Denial of Service (DDoS) were implemented in these bots.

With computer systems becoming available to the wider public, Distributed Denial of Service became even more popular in the late 1990s. Tools like Trin00 [10], Stacheldraht [11], shaft [12], and Tribal Flood Network 2000 [13] were programmed and optimised in concentrated attacks from multiple sources.

The concept of computer worms goes back to 1971, when the first specimen, known as Creeper, was written [14]. This program copied itself between machines of the Advanced Research Projects Agency Network (ARPANET). It displayed a message on the screen but always tried to remove its presence from the source computer. Later worms, in the early 1980s, were designed with good intentions, for example with the purpose of notifying other users on the network (Town Crier Worm) or managing computing capacity at night, when nobody was using the machines (Vampire Worm). In 1988, the Morris Worm [14] appeared in Cornell University and had a massive impact, probably infecting a tenth of all computers on the Internet at that time with around 6.000 machines. Because the Morris Worm received a lot of media attention,

the idea of network-based, autonomously-spreading programs was carried into the wild. Malevolent implementations showing worm behaviour were identified shortly after these events, some of the first ones [14] being, for example, the Father Christmas Worm or Worms Against Nuclear Killers. Computer worms were one of the primary propagation vectors of early botnets.

With the rise of remote access tools like Back Orifice 2k [15] or SubSeven [16] in the late 1990s, a prototype for the concept of botnets with control over only one machine was created. These tools already contained functionality like the logging of keystrokes and forwarding of connections.

The combination of the aforementioned functionality finally resulted in the concept of botnets, with early specimens like Pretty Park, GTbot, SDBot, Agobot, Spybot, Rbot and several more [17].

BOTNETS

A modern definition of “bot” is a concept of advanced malicious software that incorporates usually one or more aspects of the aforementioned techniques introduced by viruses, worms, Trojan horses and rootkits for propagation and hostile integration into a foreign system, providing the functionality of the compromised system to the attacker.

A defining characteristic of bots is that they connect back to a central server or other infected machines after successfully compromising the host system, thus forming a network. This network is the so-called botnet. The bots provide a range of implemented features to a corresponding controlling entity. This entity is commonly a command-and-control server under the control of one or more persons, called the botmasters or botherders, who relay commands through this server. Depending on the network infrastructure, the bots may be connected with each other to enable this desired control structure. Alternatively, they can exist completely independently, not knowing of the existence of other bots. The characteristics of botnet communication models are discussed in more detail in section 1.2.1. A typical function that bots provide to their masters includes the automated extraction of a victim’s credentials, the organised distribution of spam, the ability to participate in denial of service attacks, or the extension of the botnet by recruiting new bots. The motivation for these activities is driven mostly by the financial interests of the botmasters [1]. This has led to a botnet-centric economy with distinct role models and stakeholders (cp. section 1.2.2).

1.2 CHARACTERISATION OF BOTNETS

1.2.1 COMPONENTS OF BOTNET INFRASTRUCTURES

The most important part of a botnet is the so-called command-and-control infrastructure (C&C). This infrastructure consists of the bots and a control entity that can be either centralised or distributed. One or more communication protocols are

used by the botmasters to command the victim computers and to coordinate their actions. The sets of instructions and functionality of botnets vary widely with the motivation behind their use. Concrete scenarios for the use of botnets will be given in section 2.3. This section focuses on existing types of botnet infrastructures and provides information that is necessary for understanding the approaches for the detection, measurement and mitigation of botnets, as presented in later chapters.

The C&C infrastructure typically serves as the only way to control bots within the botnet. The bots are required to maintain a stable connection within this infrastructure in order to operate efficiently. Therefore, the architecture of the C&C infrastructure determines robustness, stability and reaction time. In general, centralised and decentralised approaches can be distinguished. The centralised approach is comparable to the classic client-server network model. In these botnets, the bots act as clients and connect back to one or more central servers, from which they receive their commands. Decentralised C&C models often require the bots to act at least partially autonomously. The bots maintain connectivity to other bots and issue requests for new commands to the botnet. Because there is no single set of command servers that can serve as a single point of failure, and the botmaster can hide inside the network of bots when giving commands, this approach is harder to mitigate.

Modern botnets require great flexibility and robustness to be able to handle large numbers of bots and to maximise the profit that will be generated. Early botnets used standard, mainly well-known, protocols almost without modification. Current C&C technology has developed rapidly, introducing fully customised instruction sets and the use of cryptography [18], [19], [20]. The application of mitigation approaches leads to a constant evolution of command-and-control protocols.

CENTRALIZED C&C ARCHITECTURE

In a centralised C&C infrastructure, all bots establish their communication channel with one, or a few, single connection points, as illustrated in figure 1. These are usually command-and-control servers, under the control of the botmaster. Because all bots connect to these servers, botmasters are able to communicate with the bots simultaneously and can issue commands to all the bots that are both online and connected to the botnet. This offers them low reaction times and a good means of coordination. Direct feedback enables easy monitoring of the botnet status for the botmaster and gives information about fundamental properties, such as the number of active bots or their global distribution.

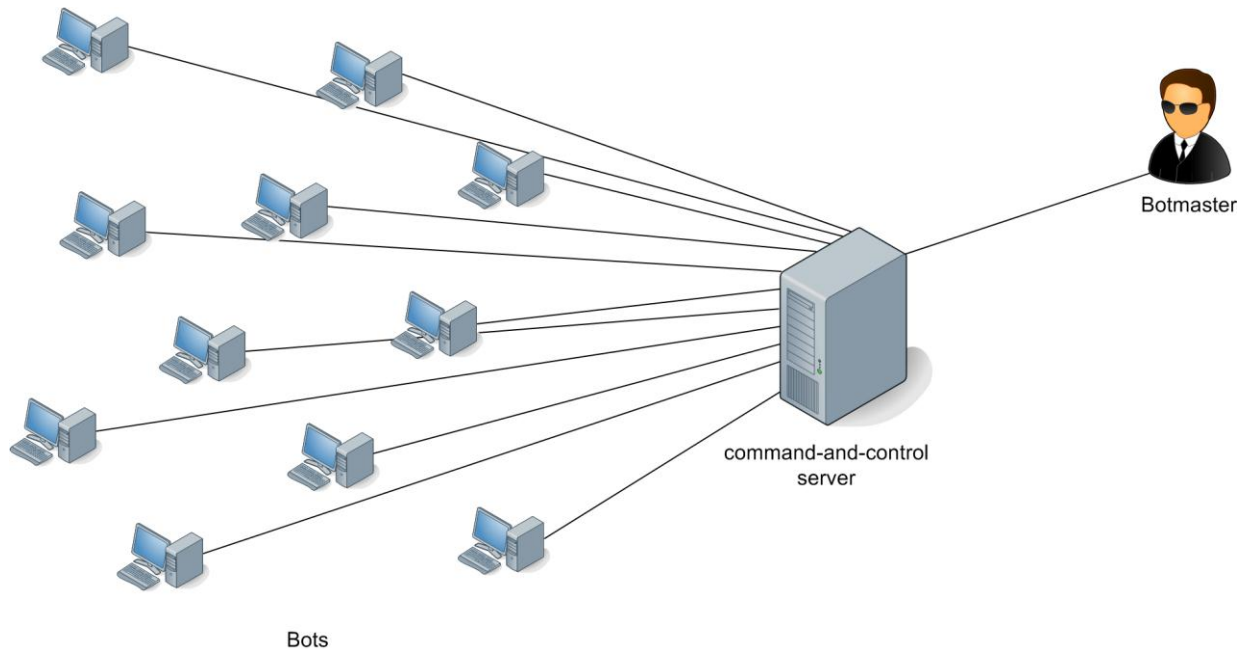


Figure 1: Centralised botnet.

The idea of botnets originated from Internet Relay Chat (IRC), a text-based chat-system that organises communication in channels [17]. The IRC protocol still serves as an important technology for botnet control and enables a centralised communication model. According to the 2010 Symantec Internet Security Threat Report [6], 31% of the centralised C&C servers that were observed used IRC as a communication protocol in 2009. One important property of this protocol is that the number of potential participants within one channel is technically not limited. This allows the collection of many bots in one such channel and the ability to command them in parallel. Additionally, private conversations are possible on a one-to-one basis. This enables the direct manipulation of single bots. Because the IRC protocol is text-based, it is easy to implement and customise. In the context of botnets, these properties offer a robust, well-established and easy-to-implement approach to commanding a botnet. IRC channels for botnet control are either hosted on public IRC servers or on servers owned by the botmaster. The bots usually implement only a subset of the IRC instruction set. This is enough to allow the operator to control the bots and reduce functionality to the required minimum. If their own servers are used, arbitrary modifications to the protocol can be made using their own instruction sets and encryption. This strengthens the botnet against countermeasures.

A well-known standard used throughout the Internet is the Hypertext Transfer Protocol (HTTP). HTTP is the protocol most commonly used for the delivery of data over the Internet. This includes human-readable content like websites and images, and also binary data transported in uploads and downloads. Because of these important

features, HTTP is available in nearly every network connected to the Internet and is rarely filtered. This is especially interesting for botnet operators, because it makes the protocol viable as a command-and-control protocol. HTTP bots have to periodically issue requests to the target C&C server. These requests commonly consist of a status report, on the basis of which the server decides which commands are transferred to this particular bot. Centralised command-and-control servers based on HTTP make up for the other 69% [6] of all C&C servers and are therefore the most common way to control a botnet. A typical example of botnets using HTTP for communication are those generated with the commercial Zeus crimeware toolkit, featuring a tool for the construction of binaries and a graphical user interface situated on the C&C server. This allows the botnet to be managed with a low amount of technical skill.

In some cases, a botnet will be organised in multiple tiers. For example, instead of a single central C&C server, there can be a server infrastructure. This infrastructure usually has a hierarchical design and can include, for example, dedicated servers for orchestrating bots into subgroups, similar to a load balancer, and more servers for the delivery of content, such as spam templates, to bots [21]. This structure may also include the differentiation of bots, e.g. those directly reachable from the Internet, acting as proxy nodes for the botnet, and those hidden in internal networks as workers [22].

DECENTRALIZED C&C ARCHITECTURE

In decentralised command-and-control architectures, loosely coupled links between the bots enable communication within the botnet and provide the basis for its organization. A common term for this class of botnets is peer-to-peer botnets, as this is the name of the corresponding network model. The knowledge about participating peers is distributed throughout the botnet itself. Consequently, information about the whole botnet cannot be obtained directly, and commands have to be injected into one peer of the botnet. Usually, this is either realised over the communication protocol directly or via the update functionality. In the latter case, bots will exchange their revision number upon communication and, if these vary, the older bot is updated to the version of the new bot. In doing so, a revision is propagated through the botnet over time. The insertion of such updates and commands into the botnet usually happen from an arbitrary point, making localisation of the botmaster almost impossible. This provides a high degree of anonymity. Monitoring its activities or following such a new release through the network is very difficult. In figure 2, the simplified design of a peer-to-peer botnet is shown as an example of a decentralised C&C approach. With regard to robustness, peer-to-peer botnets have the major advantage that no central server can be attacked to mitigate them directly. On the other hand, relying on the self-propagation of commands through the botnet means a low reaction time. There is at least one known case, the SpamThru botnet [23], where peer-to-peer functionality was used as a backup channel. In this case, optional centralised servers are used additionally for commanding, thus making the botnet hybrid.

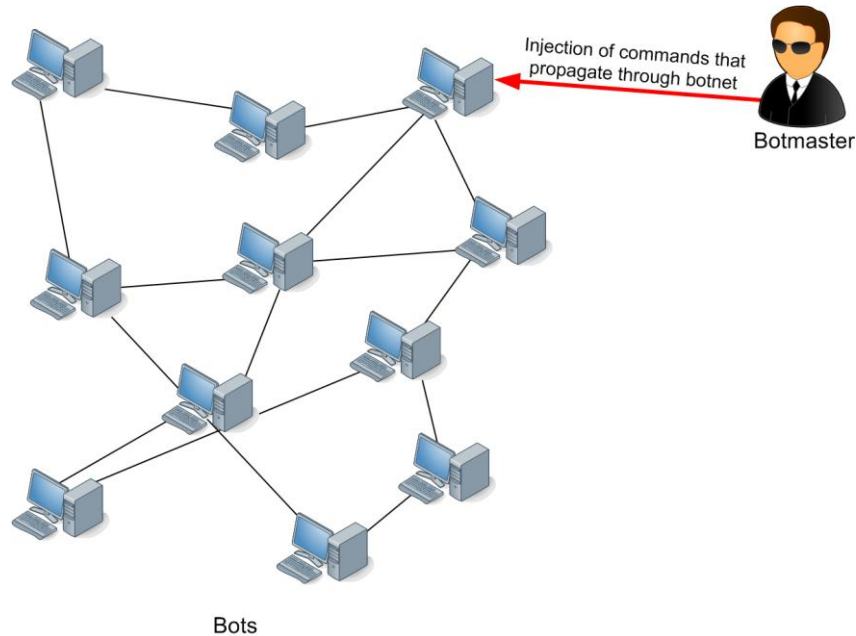


Figure 2: Peer-to-peer botnet.

THE ROLE OF THE DOMAIN NAME SYSTEM FOR BOTNETS

For centralised approaches, the Domain Name System (DNS) has an important role, as it allows changes to the C&C infrastructure to be performed dynamically. When DNS is used, the command-and-control server is identified by a (command-and-control) domain name that is resolved by DNS to an IP address. Multiple, successful botnet mitigation techniques aimed at DNS have been applied in practice. An example of this is the deregistration of malicious domain names. However, the worldwide distributed structure of DNS complicates the mitigation process.

The technique described below illustrates how botnet creation and management use approaches of highly technical complexity. The principle of so-called Fast-Flux Service Networks (FFSN) is an invention of botnet developers, designed to increase resilience and anonymity, and it has achieved significance for several botnets. The idea of fast-flux is comparable to Content Delivery Networks (CDN), as depicted in figure 3. When a malicious domain name has to be resolved, a query is usually first sent to the nearest DNS server and then handed through the DNS system to a DNS server controlled by a botmaster. Fast-flux networking exploits the properties of DNS as follows: The response to the query will usually include a large number of IP addresses associated with bots. These bots function as proxy servers, forwarding all the client's communication to a server, which hides behind the proxy layer. Behind this proxy layer, malicious services like web page phishing, suspicious advertisements, or malware archives for the download of additional modules, can be hidden. Generally,

the DNS response records have a very short period of validity, which results in rapidly changing responses for the same domain name.

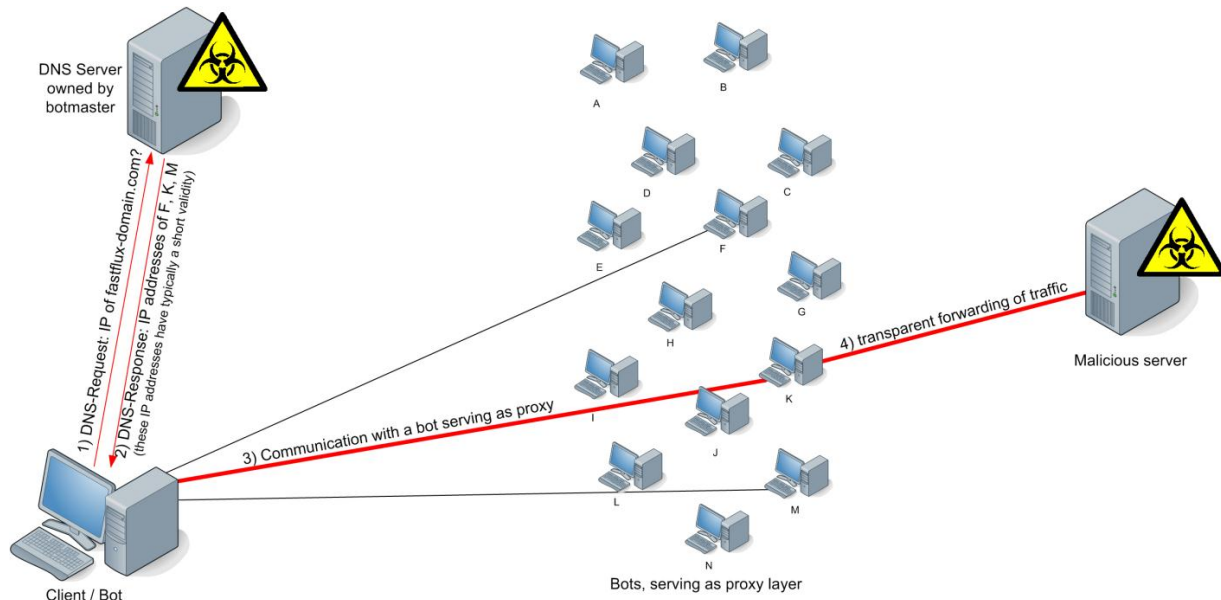


Figure 3: Fast-flux service networks.

As mentioned earlier, a common approach to mitigating botnets is to block malicious domain names. Depending on the DNS service provider in charge, this procedure can require considerable effort. In any case, single domains can be blocked or taken down comparatively easily. This has led to another concept that has been invented for use in botnets, Domain Generation Algorithms (DGA). The idea behind DGAs is to generate C&C domain names (domain names linked by DNS records to C&C servers) depending on one or more external information sources serving predictable seed values called markers that can be accessed by bots and botmasters alike. Markers that have been used in this context can be timestamps as well as data from popular websites such as Twitter Trends [24]. While timestamps provide the ability to generate domain names far in advance, the use of dynamic web content eliminates this element of predictability. Hundreds, or even thousands, of domain names can be generated at short intervals. A typical time period for the validity of such domains is one day. Any attempt to mitigate this kind of botnet attack now faces numerous suspicious domains. The effort required to defend against this kind of attack increases massively, as the botmaster can choose to simply pick one of the possible domains, register it and make sure it provides new instructions, or an update, during the validity of this domain name.

A special case of centralised botnets is so-called locomotive botnets. The idea behind locomotion is to rely on a centralised C&C model and to synchronously switch the centralised component regularly. Figure 4 illustrates the concept of locomotion. By changing domain names or associated servers frequently, the botmaster exploits the

administrative efforts needed to obtain the deregistration of a domain name or the takedown of a server. The migration of the central server can be realised via updates or Domain Name Generation.

INDIRECTION OF COMMAND-AND-CONTROL

In addition to the approaches mentioned, other technologies have been exploited for botnet command-and-control architectures in order to achieve a certain level of indirection. The basic idea is to establish a covert channel within other common available Internet technologies and services, such as Instant Messaging (IM), Really Simple Syndication (RSS) or social networks. The motivation behind the use of these existing infrastructures is an implicit guarantee of network stability, because the providers maintain their legitimate services. Furthermore, these services require low verification of identity when considering initial registration and can be exploited for indirection of control flow. It can be assumed that almost every Internet-related technology is under investigation by criminals in search of tools for botnet operations.

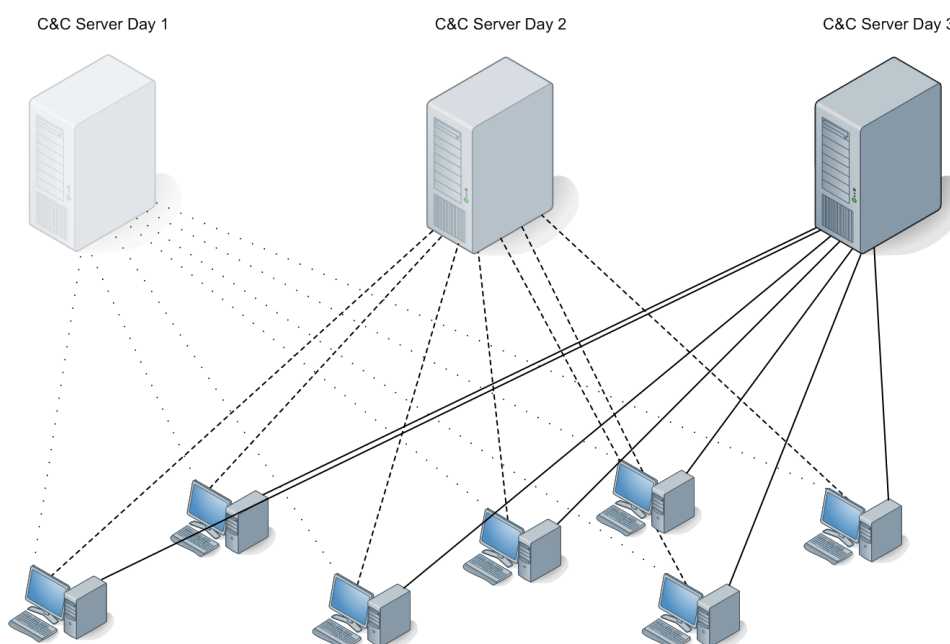


Figure 4: Locomotive botnet.

1.2.2 MOTIVATION AND USAGE OF BOTNETS

The fast-paced evolution of malware and botnets from a technical point of view has already been illustrated in the previous sections. Together with the technology, an underground cybercrime economy has developed around the field of botnets (cp. figure 5). This section focuses on the motivation behind the operation of botnets. This includes services offered by botnets that can be used to generate benefits, and role models within the malware industry.

First, malware and botnet software has to be created. This is usually done within the malware industry by malware developers and virus creators. Developers are not necessarily the same people who use their creations. Often, pre-compiled malware, software packages for customised creation of executables or extension modules are sold to botmasters. Other “products” suitable for spreading malware are exploits or compilations of exploits that are integrated into malware and support the propagation process. Selling malware can also be divided into multiple stages of a supply chain, including malware distribution services or botnet or crimeware construction kits. In this case, another layer is introduced into their business, providing additional anonymity for the developers. For some botnet creation kits, some kind of “service level agreements” exist, mainly targeting updates and patches. Even botnet services featuring telephone support has been detected [25], [26], [27].

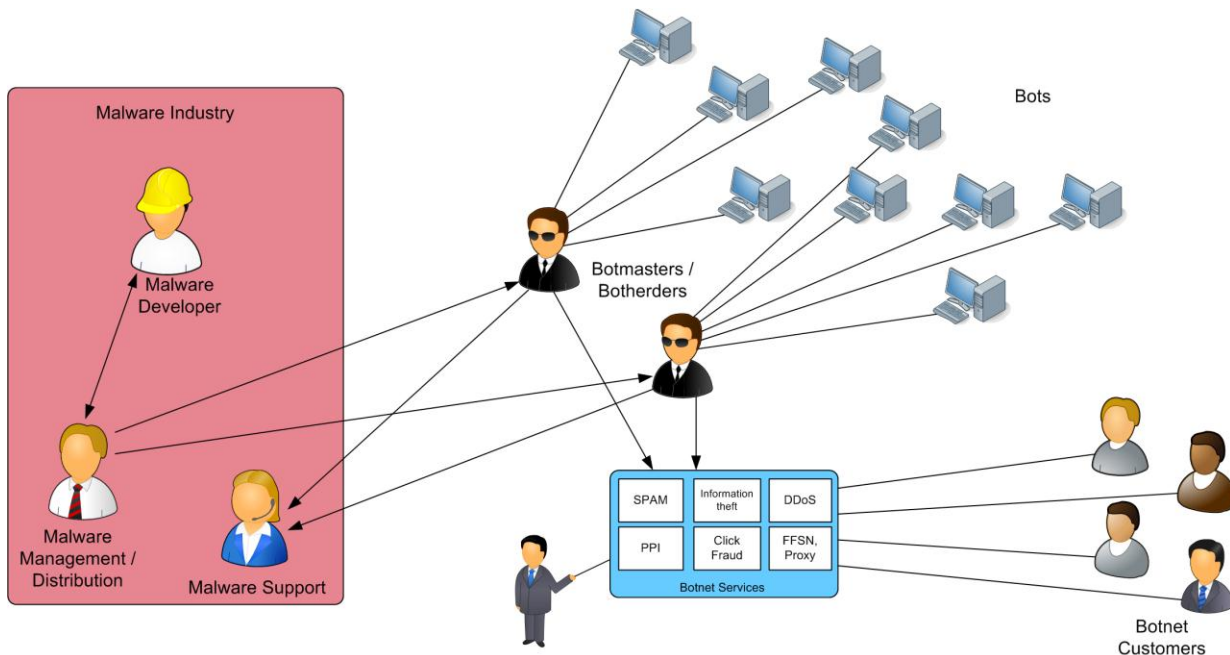


Figure 5: Simplified role model of the malware economy.

The direct customers of the malware supply chain are botmasters or botherders. Use of botnets requires a far lower level of technical skill than the development of malware. Many construction kits are well documented and have a graphical user interface. Botmasters control the bots, work on the expansion of the botnet, and use the botnet to generate profit. Services, such as the maintenance of botnets, command-and-control servers or bots, are all available for purchase through underground communities.

Further generation of profits comes from offering botnet services to third parties. Illegitimate control of as many as hundreds of thousands or even millions of remote computer systems, usually with the highest privilege levels, provides botmasters with

enormous computing power and bandwidth capacity for use as a business asset [28]. The infrastructure bundled into a single botnet is worth several magnitudes of the initial effort needed to acquire them through malware. Indeed, botnets have various characteristics in common with regular cloud computing, such as heterogeneous resources, decentralised scheduling of commands or network overlays, and resilience and mechanisms for failover [29]. From this point of view, botnets can be considered as a “black-market cloud” used for malicious purposes.

The main motivation for the operation of botnets is to generate financial profit from the activities they allow. Therefore the capabilities of the compromised hosts and the data stored on them usually have to be monetised. Other possible motivations include political or even military interests. In the last few years, a few applications related to botnets have taken a leading role and can be regarded as characteristic.

IDENTITY THEFT

A major use of botnets, with the intention of gaining financial benefits, is for the automated extraction of user data and credentials from infected hosts. Key targets include passwords for various services like e-mail accounts, web shops, banking platforms or social networking platforms [30]. This technique is often called ‘identity theft’, because it enables botmasters to impersonate the victim, making further actions, like fraud, possible. For example, an e-mail login can reveal the owner’s contacts through the inbox and address books:

- These contacts can be abused by sending emails with a malicious attachment, increasing the chance of recruiting additional machines for the botnet.
- The extracted e-mail addresses can be used for spam, as they are confirmed as valid addresses.
- Many websites allow the re-sending of a password to the registered e-mail address. The login to an e-mail address may serve as a key to more credentials.

Many of these cases can be applied to social networks too. In these, the personal relationships of the victim can facilitate attacks. Credentials for web shops and banking platforms promise immediate profit to an attacker. Banking platforms can be abused for fraudulent money transfers and for buying goods, which are then sold on. The same is applicable to credit card information extracted from the victim’s machine. The crimes described here do not have to be performed by the botmaster directly. A viable market for trading in stolen identities and credentials is another option for monetising data.

SPAM E-MAIL

One of the most popular uses of botnets is for unsolicited mass mailing, also known as spamming. While classic spamming has been achieved with single computers or

comparatively small networks owned directly by the spammers, botnets have enabled them to perform a cost shift, in terms of computation power, bandwidth and reputation, towards the owners of compromised computers. Also, the ability of botnets to use bots' IP addresses to hide the true originator of the spam email complicates countermeasures such as the blacklisting of suspicious IP addresses.

The primary intention behind mass mailing is advertising. Commonly, third parties buy a batch of spam mails for delivery by botmasters in order to advertise their products. In addition, spam mails can be used either as a malware vector with attachments or as links to malicious websites. Password fishing, known as 'phishing', is another use of spam closely related to the manipulative techniques of social engineering. It is used to steal identity information. Over the past few years, estimates of spam mails' share of all e-mails range from 80%-90% [31]. According to Ferris Research, the global financial cost of spam mails in 2009 was around 130 billion US\$ [32].

CLICK FRAUD AND PAY-PER-INSTALL

Another way of monetising botnets is through what is called click fraud. First, the attacker sets up an account with an online advertiser, who pays for page visits or for additional advertising links by, for example, clicking on a banner. Second, the attacker uses the controlled bots to visit those pages and to generate clicks on the target banners. This is possible because he has full control of the victim's machine and may use it to simulate surfing behaviour. In this case, the attacker gains money directly from the advertising company, which in turn does not benefit from the traffic generated. In the context of botnets, clicks can be sold to third-party advertisers in order to optimise their click popularity and ranking in search engines. Alternatively, generating such traffic can be used to influence online polls.

- Click Forensics [33], a company specialised in enhancing traffic quality for online advertising, reports an average attempted click-fraud rate of 18.6% (of all clicks recorded on monitored adverts) for Q2 of 2010, which has increased continually from 14.1% in Q3 of 2009.
- Anchor Intelligence [34], which also specialises in the analysis of traffic quality, estimates a much higher rate of attempted click fraud at 28.9% for Q2 of 2010.

Related to the click fraud as described above is a business model called pay-per-install (PPI). The botmaster offers to install software on target machines for his customers. Popular targets are: advertising software (adware) that, for example, displays additional advertising in browser pop-ups; spyware that monitors user behaviour; or even other types of malware that integrate the victim into another botnet. This may serve as a backup technique, if a botmaster has lost control of his botnet and wants to acquire a large number of machines simply by paying a third party for them.

DISTRIBUTED DENIAL OF SERVICE

Botnets usually consist of such large numbers of remote machines that their cumulative bandwidth can reach multiple gigabytes of upstream traffic per second. This enables botmasters to start targeted sabotage attacks against websites. By commanding bots to contact a website frequently, the servers are rendered unreachable because they cannot handle the incoming traffic. This attack is called a Distributed Denial of Service (DDoS) attack; distributed, because a large number of geographically-distributed bots are involved in the attack. These attacks happen regularly and the profit scheme connected with this use of botnets is extortion. Many companies depend on web-based services, e.g. web shops, and downtime causes a loss of business volume. A common example of this is online gaming and gambling websites, especially around the time of the American Super Bowl, when gambling websites were a prominent target [35].

In the course of the publication of more than 250,000 US embassy cables by WikiLeaks, several large companies' services and websites have been successfully attacked via DDoS attacks. In this context, an online community, consisting of a varying number of anonymous users acting in a coordinated manner (known as Anonymous), used the open-source tool, Low Orbit Ion Cannon (LOIC) [36], to perform DDoS attacks. This tool allows volunteer users to connect to an IRC server that will in turn provide their instance of the tool with targets that can be flooded with requests or packets. These actions are synchronised among all the volunteers through the IRC channel. It should be noted that this is not actually a botnet: In the case of LOIC, the software used for DDoS is installed by the user himself and, furthermore, the user can decide at any time when and against which targets he wants to provide his bandwidth to support the attack. Those targeted by Anonymous were financial institutions that ceased transferring donations to WikiLeaks and institutions involved in investigations against WikiLeaks' chief editor, Julian Assange.

The Shadowserver Foundation [37] recently identified a new botnet type called "Darkness" [38]. Darkness is generally comparable to the BlackEnergy bot [39], which also specialises in DDoS attacks, but is claimed to be more effective. It has been already actively used in some DDoS attacks and is advertised as being able to take down even large websites with just a few thousand bots [38].

POLITICAL INTERESTS

In addition to their economic interests, botnets may also be used in political or military contexts. A significant example of a successful DDoS attack happened in April 2007 in Estonia. For around two weeks, several federal, banking, and news websites were targets of concentrated DDoS attacks connected with botnets. It was supposed that the cause of these attacks was the moving of a Russian war monument in Tallinn, Estonia. The attacks were supported through Russian language forums and blogs, which provided a platform for the distribution of attack tools and scripts [40]. These

communication channels were also used for coordinating attacks, in respect of both potential targets and attack time. These were considered to be the first politically motivated cyber attacks of this size. The DDoS attacks had a significant impact on the Estonian population [41].

In 2009 and 2010, two espionage botnets were explored in depth, GhostNet [42] and the Shadow Network [43]. The investigations of GhostNet have led to the discovery of 1295 infected machines in 103 countries, with around 30% of the infected machines considered as “high-value”, because they were situated in government institutions. These included computers in various embassies, ministries, and commissions. Several network traces captured from these machines showed communication between infected hosts and IP addresses of command-and-control servers situated in China. Those traces, proved the extraction of sensitive documents [43].

Another recent example is the Stuxnet Worm. Shortly after its appearance, this malware specimen was considered one of the most complex pieces of malware [44] ever identified, attracting major attention in 2010. After intense analysis, the main parts of its functionality were revealed [45]. Although it is debatable whether Stuxnet should be classified as a botnet, it contains many typical botnet features. After successful infection, the compromised host verifies Internet connectivity and then tries to connect to two possible C&C servers in order to send information about the system and ask for an update [45]. Stuxnet’s features include routines that identify and attack only industrial systems containing a specifically-defined configuration. It is therefore the first worm to target critical infrastructure. In addition, four unknown (zero day) exploits and two digital certificates were used for Stuxnet’s spreading mechanisms. Because the identification and development of such exploits, as well as the use of detailed knowledge of industrial systems, was applied to the creation of this worm, it was probably designed and programmed by a team of developers.

1.2.3 ATTACK POTENTIAL AND THREAT CHARACTERISATION

In this section, a range of metrics for assessing the attack potential and threat of botnets is proposed. The general goal of botnet measurement and threat characterisation is to provide evidence that is helpful for:

- Deciding on investments in security technology and architecture. This is important for both governments and businesses.
- Defining the political agenda. The operation of botnets is major organised crime, and a threat to society, and therefore has to be engaged with at government level.
- Reporting and journalism. By providing information to the public, awareness of security issues and corresponding threats is increased.

First of all, it is important to note that botnets should always be evaluated with metrics suited to the scope of the affected stakeholder groups. The following examples illustrate the dependency on context when assessing the direct effects of botnets:

- Service providers who offer email services are interested in the amount of spam produced by botnets.
- Companies focusing on e-commerce may be primarily concerned about the power of DDoS attacks that can harm their ability to operate.
- In order to protect their customers, financial organisations want to assess the potential of botnets for incurring financial loss.
- Governments need to shield themselves against the targeted theft of classified information.

Additionally, it should be pointed out that attacks caused by botnets create multiple layers of financial loss:

- The intended victim of an attack suffers directly from the effects, as in the examples described above.
- Internet Service Providers suffer from malicious traffic that is generated by botnets and carried through their networks.
- Third parties possessing compromised hosts and participating involuntarily in botnets incur costs caused by malware, for example: the instability of computers and services; the cleaning of machines; reestablishing the integrity of their networks.

Even within certain stakeholder groups direct measurement and comparison of the threat potential of botnets remains challenging. The reason for this lies primarily in the shortage of sufficiently accurate and comparable measurement methodologies. In chapter 3, existing methods for the detection and measurement of botnets are described and analysed in terms of their ability to assess the size of botnets.

In the following, a selection of metrics for characterising the threat of botnets is presented.

BOTNET SIZE AND BOT ORIGIN

The most common and most intuitive measurement feature used for the description of the threat potential of botnets is their size in terms of the number of compromised hosts. While this number alone has limited significance, it serves as a scaling factor for various botnet effects, such as the potential for sending spam mails, the capacity available for DDoS attacks, or a medium from which valuable data can be extracted. In general, two aspects of botnet size can be distinguished [46]:

- A botnet's footprint is the total number of compromised machines infected with a certain specimen of malware and commanded through the same command-and-control infrastructure at any point in the botnet's lifetime. In particular, this feature ignores the current system's status in terms of being powered on and being connected to a network or even the botnet.
- A botnet's live population is the current number of compromised hosts that are able to interact with the command-and-control infrastructure at a specified point in time.

Whilst a footprint gives an impression of how widespread a botnet-related malware is, the live population is a more meaningful measure for estimating how many systems are accessible to the botmaster and available for the execution of commands at a certain time. It is also easier to measure without access to the infected systems. A footprint can be generated by tracking a botnet, i.e. by further processing of snapshots taken continuously of the live population. Techniques suitable for this approach are presented in chapter 3.

In the following, some examples of numbers in botnets are provided. Where possible, these are provided subject to the advice on interpretation given by the operators of the tracking service. As a result, some numbers may appear to vary considerably. This indicates how difficult it is to measure botnets accurately, as explained later in the report:

- The Shadowserver Foundation [37], a non-profit organisation whose objective is to capture and analyse malware with the intention of monitoring and reporting associated botnets, provides up-to-date tracking information on botnets and bots. In the course of 2010, Shadowserver continuously tracked between 5000-6000 botnet command-and-control servers. According to the interpretation advice given on their webpage regarding the measurement of bots, they also tracked a total of between 100,000 and 250,000 compromised hosts.
- Abuse.ch offers a tracking service for botnets created with the ZeuS [47] and SpyEye [48] crimeware toolkits. At the time of writing, a total of 447 ZeuS servers were tracked with 144 online servers, and a total of 251 SpyEye servers with 70 online servers were listed.
- Since February 2009, the Conficker Working Group [49] has been providing data about their sinkholing efforts from a centralised source. At the time of writing, according to interpretation advice on the published numbers given by CWG, between 1,000,000 and 3,000,000 hosts are still likely to be infected with Conficker malware.
- Stone-Gross et al. [50] had the opportunity to take over the command-and-control of the Torpig botnet and to perform measurements from within the

network. This, together with a little good fortune, enabled them to measure 182,000 distinct hosts during the 10-day measurement period.

These numbers illustrate that botnets are a widespread problem. While providing accurate numbers is difficult, the given estimates still imply that a large number of computers worldwide suffer from malware infections and are participating in botnets. This assumption is supported by data from Eurostat [51] published for the year 2010, in which 22% of Internet users in the European Union encountered infections with malware within the last 12 months.

A metric of particular interest to governments, and which is also related to the size of botnets, is achieved by resolving the bot's source IP addresses to their origin, such as an Internet Service Provider. As a further step, this specifically enables the derivation of how many infections occur per country by mapping those identified sources where possible to countries. Malware removal data is, for example, generated by Microsoft in their Security Intelligence Report, which compiles the results of more than 600 million installations of Microsoft security tools [52].

SPAM

As mentioned in the introduction to this section, spam, as an effect of botnets, can be of interest to stakeholder groups. Around spam, various metrics can be defined in order to assess a botnet:

- Spam throughput. This can be measured in, for example, received emails per time unit. Issues with this metric are identifying with certainty which botnet sent which spam emails to single botnets and applying it across mail providers to achieve an accurate view.
- "Freshness" of machines used for spamming. A remarkable amount of spam is identified through the source IP address used for delivery. If an IP address is not listed prior to receiving a spam email, the probability that it will pass spam filters increases. If a large number of "fresh" IP addresses are used for spamming, this is an indication that the botnet may draw bots from a large pool of compromised hosts.
- Quality and correlation of target email addresses. When orchestrating a spam campaign, the selection of target email addresses may influence the impact of the campaign.

BANDWIDTH USABLE FOR DISTRIBUTED DENIAL OF SERVICE ATTACKS

When assessing the threat posed by botnets, Distributed Denial of Service attacks are worth considering as a central functionality. One metric that is useful for estimating DDoS is a botnet's capacity measured in gigabits per second. This allows the attack potential to be weighed against defensive measures.

HARVESTED DATA

Many botnets have functionality for automatically harvesting sensitive data from victim machines and submitting it to drop zones. Data harvesting can be measured using two metrics, one describing quantity, the other quality.

The amount of data gathered is related to botnets' size, as discussed earlier. Alternatively, the intrusiveness and aggression with which credentials are sought on the infected system, for example serial numbers of office products and computer games, expensive design and construction software, or certificates, can also serve as a metric. Furthermore, targeted attacks may aim specifically at protected knowledge, such as industrial blueprints, businesses' research results [53] or classified information. The following list gives some examples of targets of interest to hostile data mining performed by botnets:

- Email addresses extracted from address books.
- Product serial numbers stored in configuration files or the system registry.
- Credentials for various services, allowing the attacker to impersonate the victim, such as accounts for communication services like email and social networks, or login data for servers.
- Credentials and information enabling financial fraud, such as credit card information or login data for online banking services or financial institutions.
- Internal business data containing specifically-secured information like operational plans or intellectual property, e.g. blueprints or source code.
- Classified documents or general information about governments and the military.

DIRECT FINANCIAL DAMAGE

Botnets often serve as a tool for systematic financial fraud. The aforementioned extraction of credentials can be used directly for illegitimate bank transfers or abuse of credit card information in payment processing. In this context, the volume of financial claims by customers about transactions they have never made can serve as a metric. While this metric is of interest to financial institutions, attributing fraudulent activities to certain botnets is complicated, as the stolen data may have been sold in between across multiple parties.

Denial of Service attacks caused by botnets are also a source of financial damage. The downtime of an e-business platform, and consequently the inability to operate, may express itself in financial losses. This is especially critical for businesses which depend on the constant availability or certain circumstances, such as online betting platforms during major sports events. While downtime caused by DDoS is relatively hard to measure, another relevant metric in the context of DDoS is extortion coupled

with the threat of an attack. Because this actually involves communication with the criminal, concrete values are known to the victim.

Another metric in the context of direct financial losses related to botnets is click fraud. Here, the volume of generated clicks, or certain techniques for bypassing detection, are of interest. Faking clicks on an advertisement is motivated either by the desire to create an advantage over competing parties for whoever is perpetrating the click fraud, e.g. by directly creating costs or by consuming the competitor's advertising quota. Another alternative is to generate money through compensation schemes like Pay Per Click. Click fraud is hard to detect, as it targets legitimate website visitors who are by definition an anonymous group. This makes it hard to account for losses explicitly caused by click fraud.

SECOND LEVEL FINANCIAL DAMAGE

This type of metric aims at assessing costs that are occurred by third parties, by for example measuring the efforts required for cleaning infected computer systems or restoring a network's integrity.

MALWARE DEFENCE MECHANISMS / MALWARE RESILIENCE

One rather abstract indicator targets the defensive mechanisms that a botnet may use in order to increase its resilience. This includes aspects like the infrastructure used for command-and-control, e.g. use of bullet-proof hosting services or the presence of a Fast-Flux Service Network. This also addresses the use of the server-side polymorphism of malware binaries, which is a technique for massively disrupting automated malware detection. A further indication is the presence of encryption for the protection of command-and-control traffic or the possibility of digitally sign commands with to prevent takeovers through impersonation.

Indicators detectable on the local machine of the the threat level of posed by malware include the use of obfuscation techniques to increase analysis time or to stay undetected, and the sophistication used to maintain a presence on the infected system.

Finally, if the malware provides an update, or even a dedicated Pay-Per-Install service, this also provides the possibility of re-infecting the computer system or introducing some other malware in order to prolong its presence.

Setting up a classification, or rating scheme, based on the presence of these features may be possible, but the uniqueness of many botnet characteristics means that they are hard to compare. Common features, such as polymorphism and the use of fast flux, are however meaningful indicators, which may be compared between malware samples.

MALWARE OFFENSE MECHANISMS / MALWARE PROPAGATION

The metrics for offensive mechanisms aim to rate the aggressiveness that malware uses to distribute itself. A good example of a highly aggressive malware is Conficker, which used a vulnerable network service in order to spread autonomously over the Internet. Conficker also spreads via removable media like thumb drives and network shares. While Conficker used a known vulnerability when it first appeared, Stuxnet used multiple zero-day exploits. This is another metric for rating offensive malware mechanisms, analogous to defence mechanisms. It involves the classification of malware according to the number and type (e.g. zero day or known) of offensive mechanisms.

2 METHODOLOGICAL APPROACH OF THIS STUDY

One major aim of this study is to provide a comprehensive view of the different approaches that have emerged in the fight against botnets. This fight is an integrated process that combines detection, measurement and the resulting countermeasures. For this study, a distinction has been made between detection and measurement on the one hand and countermeasures on the other. However, accurate detection and measurement can be interpreted as a prerequisite to efficient application of countermeasures. Where possible, interdependencies of the techniques examined have been outlined. In the following, the process of information gathering is presented, together with a feature set for the evaluation of approaches.

2.1 INFORMATION GATHERING

This section covers the information sources from which the descriptions and analyses in the following chapters were created. It includes a survey among international experts on botnets, supplementary desktop research, and moderated discussions by the expert group.

2.1.1 ONLINE SURVEY AMONG EXPERTS

As a preparation for the identification of approaches for the detection, measurement and countering of botnets, an extensive feature set was created. This feature set allows the evaluation of approaches and is divided into four key factors:

- General quality of the considered approach, covering aspects like generality and flexibility.
- Effectiveness and efficiency of the approach.
- Requirements for the application and efforts needed for a successful operation.
- Limiting factors that can influence the applicability of a method.

Based on this feature set, four questionnaires were developed:

- Technical aspects of detection/measurement.
- Technical aspects of countermeasures.
- Social and regulatory approaches towards the mitigation of botnets.
- Legal questions concerning botnet countermeasures.

Based on these questionnaires, a survey was performed via an online system, e-mail, and telephone interviews with experts in 12 different stakeholder groups, details of which are:

- 1 National and pan-European Internet Service Providers
- 2 Anti-virus software developers and security solutions providers
- 3 Operating system providers
- 4 Application and network providers and developers
- 5 Web 2.0 and social network site providers
- 6 Academia
- 7 CERTs
- 8 Online user communities and consumer protection associations
- 9 Regulators and policy makers
- 10 Law enforcement agencies
- 11 Pan-European Associations of Providers
- 12 Financial institutions

The full questionnaire templates are available on request. Overall, 69 questionnaires were answered in full. A further 55 questionnaires contained partial, but still very valuable, answers. In total, 124 questionnaires were answered by the experts.

2.1.2 DESKTOP RESEARCH

In addition to the survey extensive desktop research was carried out. The reason for this was to supplement the pool of information gathered through the survey. Primary sources for the desktop research were academic publications, technical reports and whitepapers, scientific blog posts and reputable news providers.

2.1.3 GROUP DISCUSSIONS BETWEEN EXPERTS

In addition to the survey and desktop research, moderated group discussions were held on a mailing list. The topics that were used to structure the discussions were:

- Incentives and challenges for information sharing
- Extrapolation
- Jurisdiction
- Detection/Measurement/Countermeasures method rating
- Role of EU/governmental organisations
- Missing pieces
- Unseen ratio

- Responsibility
- If you had one billion dollars...
- Future botnet trends.

The key results of these group discussions are included in this report. Because the discussions were of such excellent quality, and revealed many interesting findings, a separate document summarising the findings was created. It is available under [3].

2.1.4 PEER REVIEW GROUP

The process of the entire project, “botnets - detection, measurement, disinfection & defence”, is accompanied by the input and recommendations of a distinguished group of international experts consisting of:

T.J. Campana, Microsoft
Christian Czossec, Nato CCD COE
Dave Dittrich, Washington State University
Paul Ferguson, Trend Micro
Willi Herzig, BSI (German Federal Office for Information Security)
Thorsten Holz, Ruhr-Universität Bochum, Germany
Mikko Hypponen, F-Secure
Sven Karge, eco - Verband der deutschen Internetwirtschaft e.V.
Vitaly Kamluk, Kaspersky Lab
Jose Nazario, Arbor Networks
Richard Perlotto, Shadowserver Foundation
Rogier Spoor, SURFnet
David Watson, The Honeynet Project
Dave Woutersen, Govcert.NL

2.2 APPLIED FEATURES FOR ANALYSIS

In this section, a general set of features for assessing techniques is proposed. The main use of these features is to investigate which factors can have an impact on the quality of certain methods and therefore should be taken into consideration when considering applying them in the wild.

Therefore, the term ‘quality of results’ is characterised from different points of view. Additionally, possible dependencies are given on technical and financial resources, or input from third parties that may arise when working against botnets. One outcome of such dependencies might be limitations that can downgrade the achievable effectiveness. In addition, laws and regulations can constrain methods to the point where their application is either not legal at all, e.g. when violating the privacy of user data, or too expensive to implement.

2.2.1 GENERAL CHARACTERISTICS OF THE APPROACH

The first group of features targets the general quality of the methods in terms of their conception and usability.

GENERALITY

The generality of a method describes those cases where it is applicable. In the first instance, this means the botnets or classes of botnets that can be targeted by the method. Possible factors that influence the method might be, for example, the botnet's infrastructure, effectively rendering the method useful only for certain groups of botnets. Alternatively, the method could be targeting a specific family of malware by exploiting the peculiarity of its functionality.

Generality also addresses the extensibility of the method and how much effort is needed if the method is to be available for a greater diversity of botnets.

FLEXIBILITY

While 'generality' expresses which botnets can be addressed by a method, 'flexibility' describes how easily the method can be adapted if changes are made to a given botnet. A possible scenario would be the monitoring of traffic flows for certain communication patterns and the adaptation of the approach when the botnet protocol is extended with encryption.

The measure in this case is whether the method is adaptable at all and also how much effort has to be applied to successfully perform the transition.

REACTION TIME

On the one hand, the reaction time of a method includes the preparation, development, and maintenance time that needs to precede the actual application. For example, the technique presented in section 3.2.6 (Enumeration of Peer-to-Peer Networks) usually requires enormous effort to reverse-engineer the communication protocol and to implement a dedicated tool to carry out the actual enumeration of a peer-to-peer network.

On the other hand, reaction time may also include the time required for gathering data before a method can be applied. An example of this is a machine-learning algorithm that has to be trained with reference data.

STEALTHINESS (DETECTION AND MEASUREMENT ONLY)

'Stealthiness' describes the degree to which the application of the method may be observed by the botnet owner. In terms of measurement methods, the less noise generated by the method, the better the results, because the botnet owner will remain unsuspecting and is unlikely to implement any (counter) countermeasures or changes that may prevent further application of the method.

With respect to countermeasures (as opposed to detection and measurement techniques), it is not possible to avoid the botnet owner noticing the application of these approaches. In the best case, if the method has been performed stealthily, it will already be too late for the botnet owner to react and he will not be able to regain control of the botnet if, for example, the command-and-control server has been taken down and the bot software does not have a backup communication channel.

2.2.2 QUALITY OF RESULTS

This group of features discusses the actual quality and immediate benefits that result from the output data generated by the approaches presented. In the case of countermeasures, quality refers to the observable effectiveness in terms of disruption of a botnet's functionality or coverage when cleaning infected hosts.

COMPLETENESS (DETECTION AND MEASUREMENT ONLY)

The term 'completeness' describes the fraction of the whole botnet population that can be measured or affected. Considerations in this context result from the definitions in the introductory chapter. While the live population contains only those bots that are connected to the botnet for the duration of the measurement, the footprint consists of all infections that can be attributed to the botnet involved. The completeness can even be independent from the moment the targeted method is applied. For example, if databases of stolen credentials can be extracted from a command-and-control server, the botnet's history can be reconstructed and, in the best case, be used to even identify victims whose machines have already been cleaned by other measures.

ACCURACY AND UNCERTAINTY MEASURES (DETECTION AND MEASUREMENT ONLY)

Measured data must be viewed in relation to the full context in which it was gathered. Therefore, it is very important to analyse the accuracy of results and their uncertainty and to clarify their significance. In the case of measurements, this includes information about false-positives and false-negatives as a measure for the technique's reliability.

TIMELINESS OF MEASURED DATA (DETECTION AND MEASUREMENT ONLY)

'Timeliness' is a feature that focuses on possible delays that might occur in the results of a method, due for example to processing time or the availability of data input. In relation to this, if the method does not generate output in real time, but is based on daily reports, this also has to be taken into account when considering timeliness. This is of especial interest when considering measured data as input for countermeasures to ensure effective operation.

POSSIBLE INTERFERENCES (DETECTION AND MEASUREMENT ONLY)

As well as robustness from a technical point of view, possible side effects also have to be taken into account. This feature therefore tries to capture the possible effects that can occur due to the measurement software's interference with the botnet. Even more interesting for countermeasures is the circumstance when botnet owners' counter- and defensive strategies are taken into account.

COVERAGE AND SUCCESS RATE (COUNTERMEASURES ONLY)

Finally, the efficiency of the method is not only characterised in terms of the percentage of infections covered but also the degree of recovery. For example, all affected hosts may be treated, but disinfection is limited to the removal of malicious software only. In this case, the systems might be left vulnerable or unstable, due to changes in the configuration inflicted by the malware.

DURABILITY (COUNTERMEASURES ONLY)

Another feature that is important in the context of countermeasures is the distinction between whether the method has to be applied once, and provides ultimate disruption or even disinfection, or if frequent or constant reapplication is needed. Therefore the matter of durability can influence the financial or administrative efforts connected with the application of a method.

MEASURABLE EFFECT (COUNTERMEASURES ONLY)

Where the approach is a dedicated countermeasure, it is desirable to be able to evaluate its effects when it is applied. As direct observations are often not possible, indirect indicators have to be used. Those indicators can be, for example, a decreasing number of bots and infections, a measurable reduction of spam, or other attacks associated with the target of the mitigation attempt. Possible side effects, or even collateral damage, also have to be taken into account.

SCOPE OF CONSIDERED DATA AND INFORMATION

Another important feature of measurement and detection techniques is the scope within which these operate. This includes the amount of data that was analysed as input, e.g. the period over which measurements are performed, the number and type of sensors or systems used, and if filters are applied to reduce the raw data in terms of computation capacity in real-time applications. As regards the output data, restrictions on content and availability, as well as data format, are of interest if the results are to be useful.

(TECHNICAL) ROBUSTNESS

'Technical robustness' summarises the factors that can influence an approach from the perspective of the natural environment in which it operates. More specifically, it targets the robustness on the assumption that the targeted botnet continues its behaviour without reacting to the method applied. For example, different operating system

versions, configured localisation settings and patch levels can affect the functional efficiency of a method.

2.2.3 REQUIRED EFFORT AND RESOURCES

Every technical means requires some kind of resources. This is independent of the class of the methodology and applies to detection, measurement, and mitigation alike. The actual types of resource required vary greatly between the different methods. A lack of resources can drastically influence the effectiveness and efficiency of methods. This group of features plays a major role in the evaluation of approaches.

TECHNICAL EFFORT

As most of the approaches presented in chapter 3 and 4 are of a technical nature, it goes without saying that a lot of resources are allocated to this category. Typical requirements are infrastructure in terms of hardware, e.g. to provide computation capacity and sufficient amounts of data storage. Since botnets usually operate over the Internet, the available bandwidth and topological position in the Internet will also affect the applied techniques.

DEVELOPMENT EFFORT

When evaluating the 'effort' of an approach, the time needed to develop it from an initial idea to an operational system has to be taken into account. This can be important, especially where the method lacks generality and if changes to a given botnet's structure lead to more development effort, or even render the technique useless.

ADMINISTRATIVE EFFORT

Not only is the initial effort involved in the development of an approach important but also the work associated with the eventual ongoing maintenance. Apart from this, methods may depend on active or manual monitoring, which requires frequent surveillance. Connected with this is the need for calibration.

REQUIRED EXPERT KNOWLEDGE

A valid assumption for all the approaches presented is that their application requires a certain degree of expert knowledge from different fields. Where sophisticated techniques or models are concerned, this can have an impact on several of the other features described in this section, as scalability might be limited simply by virtue of the availability of experts with the necessary competences to perform such tasks.

COST FACTORS

All the features mentioned earlier are linked directly to cost, which can limit the general feasibility of a method. The question of who should be involved in covering the expenses for the enforcement of complex approaches is closely related to the question of general responsibility for the mitigation of botnets.

2.2.4 LIMITATIONS

Apart from the characteristics of a method, and in addition to the required resources, potential 'limitations' are an important factor in estimating the value of approaches that target botnets. Again, such limitations may have various origins.

RESTRICTING LAWS

First of all, many approaches may be subject to laws that influence their applicability. Data protection laws can prohibit the collection of the data needed. The remote identification of infected machines, or the recovery of credentials from a malicious server, may violate privacy laws and can complicate data collection. The legitimacy of investigative tools can easily become another concern.

DEPENDENCY ON THIRD PARTIES

Many approaches of botnet measurement, detection and mitigation techniques involve cooperation between different stakeholders. This situation implies the existence of various dependencies. An increased need for resources, further delays and financial costs can constrain a method's effectiveness. Usually, the affected stakeholders are law enforcement agencies, coordinating institutions like CERTs, or Internet Service Providers.

LACK OF ACCESS TO DATA, SYSTEMS OR INFRASTRUCTURE

In some cases, the application of a method is limited through restricted access to data, the affected computer systems or network infrastructure in general. This may result in some approaches being applied only in certain environments that fulfil these relevant conditions.

TOPOGRAPHIC RESTRICTIONS

For some approaches, a distributed setup may be mandatory in order to ensure their effectiveness. In other cases, regional or national laws can influence the applicability of methods. Therefore, topographic restrictions have to be taken into account.

3 MEASUREMENT AND DETECTION TECHNIQUES

In this chapter, various botnet measurement and detection approaches are presented. The techniques were identified from a survey of experts among different stakeholders working in the field of botnet mitigation. Additional desktop research was carried out to supplement the findings,

3.1 PASSIVE TECHNIQUES

This group of ‘passive measurement techniques’ consists of those where data is gathered solely through observation. Through monitoring, activities can be tracked without interfering with the environment or tampering with the evidence. This makes these approaches transparent and, in many cases, their application can be hidden from botmasters. Note, however, that passive methods may also limit the amount of data that can be gathered for analysis.

In the following sections, different approaches to the passive measurement and detection of botnets used in day to-day activities are presented. First, methods for the analysis of network traffic are described, including techniques focusing on certain data abstractions and specialised protocols used regularly in the field of botnets. This is followed by approaches that focus on the recognition of attacks and interpretation of malware binaries.

3.1.1 PACKET INSPECTION

A popular concept for increasing a network's security is to inspect the network data packets. The basic idea is to match various protocol fields, or the payload of a packet, against pre-defined patterns of abnormal or suspicious content. This may, for instance, be a packet containing shell-code sequences used to spread malware, communication with an Internet address that is a proven host of malicious content, or a file server that suddenly begins to communicate via the chat protocol, IRC. These patterns are also called ‘detection signatures’.

An important application of this concept is ‘intrusion detection systems’ (IDS). The purpose of an IDS is to guard the environment in which it is deployed and to issue a warning if an attack is recognised. Subgroups of IDS are categorised into two types, depending on where the component is positioned: network-based (NIDS) or host-based intrusion detection systems (HIDS).

If an IDS does not only issue a warning when identifying an attack but takes further action, it is also called an ‘intrusion prevention system’ (IPS). Typical actions include rejecting the involved packet or closing the related connection. Another option is to forward the packet contents to an analysis system. Information extracted about the attack can also be used to contribute to a blacklist or to update firewall rules.

However, intrusion detection systems have some drawbacks:

- Full inspection of all packet contents is difficult to scale on networks with high traffic loads, at least when applied from a central viewpoint. Employing techniques like packet sampling or filtering prior to analysis increases the risk of malicious packets being missed. The scalability of deep-packet inspection for traffic monitoring was, for example, examined in the Large-scale Monitoring of Broadband Internet Infrastructures (LOBSTER) Project [54]. In this project, a decentralised monitoring infrastructure with passive sensors was deployed. In the course of the project, multiple tools were developed with a focus on traffic inspection and anonymization with respect to the required data sharing.
- Only communication streams that include known patterns are detected. Hence different ways of evading detection exist. One technique is to split the malicious content among several packets, concealing the payload. This has the effect that only parts of a signature will be included in single packets. Whilst this technique bypasses Intrusion Detection Systems that work exclusively with packet-wise matching, others reassemble the payload of multiple packets. However, other techniques that use special encoding or encryption in order to obfuscate the payload are still able to circumvent detection.
- IDSs are likely to have a noticeable rate of false positives. On the one hand sensors must be calibrated in order to not miss any attacks; on the other this may lead to problems classifying benign packets as malicious.

In the context of botnet research, packet inspection and IDS have been applied in several projects for automated detection and measurement.

Example: An early approach was shown by Blinkley and Singh [55]. They presented an anomaly-based detection approach, which uses packet inspection to gather data. They developed an algorithm for detecting IRC-based botnets, based on their scanning behaviour. The approach consists of two components, one for TCP and one for IRC. The first component measures the TCP work weight for an IP address. This is defined as the number of TCP connection control packets (namely SYN, RST and FIN) in relation to all TCP packets, sampling every thirty seconds. A value close to 1 is considered abnormal and is a strong indicator of scanning activity. The second component consists of two IRC tracking modules that collect statistics about IRC channels on the one hand and the activity of distinct source IP addresses on the other. They correlated the data from both components to identify those IRC channels which were likely to be host-infected machines that appeared suspicious due to a high work weight. This system was tested on the local campus network to identify bots. As a drawback, and possible countermeasure, they noted that encryption of IRC messages or a non-standard encoding of the communication protocol would prevent the ability to correlate IRC to TCP work weight and therefore proper detection.

Example: A method that combines several existing monitoring techniques was

presented by Gu et al. [56]. Their tool “BotHunter” observes inbound and outbound packet flows and performs a dialogue-based correlation to detect infections. The centre of the correlation engine consists of the network intrusion detection system Snort [57], enhanced with customised rules and two malware-focused plug-ins. These components contain detection mechanisms for the different stages in the infection process covering, for example, the identification and consecutive attack of a target. Also, subsequent actions are considered as attempts to contact a command-and-control server in instances of a successful infection and the download activities of additional malicious modules. The system is capable of generating reports summing up the infection steps. An example of the output can be found at the SRI International Honeynet project website [58].

3.1.2 ANALYSIS OF FLOW RECORDS

‘Analysis of flow records’ can be considered as a technique for tracing network traffic at an abstract level. Instead of inspecting individual packets, as described in the previous section, communication streams are considered in an aggregated form. In this context, a flow record consists of several properties that describe a network data stream. Typical attributes are: source and destination address; the related port numbers and also the protocol used inside the packets; the duration of the session; and the cumulative size and number of transmitted packets.

As the actual payload of packets is ignored by this approach, higher amounts of traffic can be handled than with packet inspection. However, to infer the flows from packets, the session-related headers have to be tracked. Usually, for each session currently being monitored, the activity is tracked with an ‘ageing’ counter. If activity occurs, the counter is set to a defined length of time and decreases constantly from then on. If the counter has run out, the current session is assumed to have ended. This tracking usually produces a noticeable load on the systems logging the flows. A technique to handle this is to sample the network traffic, i.e. incorporate only one in n packets. Commonly used sample rates are 0.1-1%. Where sampling is carried out, session data is strongly influenced and statistics about the number of packets and total number of bytes transferred become unreliable.

The network protocol “NetFlow” from Cisco Systems can be considered as a de facto standard for flow analysis. Routers can be configured to aggregate traffic travelling through them into flow records which are then sent to an external unit that aggregates the data.

The aim of flow record analysis is to identify traffic patterns that can be used to separate benign from malicious traffic and to create a scheme for detecting potential malicious communication.

The analysis of flow records for botnet detection and measurement benefits greatly from the input of further detection and measurement methods, as this supports the

specification of rules and heuristics. It enables, for example, the identification of hosts that interact with known command-and-control servers. The identification of particular infections inside a network is therefore possible. Application of this technique can be regarded as a means of supporting active incident handling and disinfection processes.

A comprehensive list of tools for capturing and analysing flows is maintained by SWITCH.ch [59].

Example: Strayer et al. [60] have analysed different machine-learning algorithms for detecting IRC chat traffic and IRC-based C&C botnet traffic. They extracted flows from experimental traffic logs and split the examination into two stages. The first stage consists of identifying general IRC traffic through flows. The second stage performs a separation of benign and malicious IRC traffic. They stated that the selection of input parameters in the form of flow attributes for the machine-learning algorithms had a significant impact on the results. The strongest discriminating features of flow attributes they identified were bytes per packet and the variance of this value over time.

Example: Zeidanloo et al. [61] proposed a framework for the detection of P2P botnets. It is based on the assumption that bots belonging to the same botnet are likely to behave in a similar way regarding the sending and receiving of control messages, performing malicious activities like scanning and spreading, and also spamming. Their system is divided into different units. The first unit aggregates traffic from network sources like routers and switches. The second unit filters out known benign IP addresses and domain names from the collected packets, in order to reduce system load or incomplete connection attempts (which also indicate scanning). In the next step the traffic is aggregated into flow records using the network-auditing tool ARGUS. These flow records are clustered into groups of similar structure, using different traffic features. Packet inspection is performed in parallel. The output of both techniques is then correlated to improve the results.

Example: Yen and Reiter [62] presented an approach called Traffic Aggregation for Malware Detection, or TAMD for short, with an identically-named tool. This tool performs pattern matching on network flows. Three features are selected: first, flows that communicate with a common destination that is busier than the average of all destinations; second, those that have a similar payload with edit distance [63] as a measure for similarity; and third, those flows that belong to hosts with a common operating system (OS), as most malware is specific to a certain OS. They applied TAMD on flows extracted from a university network with more than 33,000 distinct and active IP addresses. The flows were overlaid with captured malware traffic, consisting of both single bots and data from a captured botnet that contained more than 300 bots. Their system was able to detect all minor malware activities and 87.5% of the bots from the large botnet traffic capture.

Example: Jelasity and Bilicki [64] analysed peer-to-peer botnet detection on the

(Autonomous System) AS level via simulations. In response to the fact that many current peer-to-peer botnet detection approaches need high amounts of manual effort in terms of reverse-engineering and protocol reconstruction, they examined an automated algorithm for the detection of such P2P networks using Traffic Dispersion Graphs [65]. Even by using a rather simple network model, they found that applying local approaches for P2P network detection are not very promising, as the visibility of these networks can be very limited inside one AS, especially when the bots aim to keep the number of connections to other bots low.

3.1.3 DNS-BASED APPROACHES

When a host has been compromised by a botnet, communication has to be established to either a commanding server or other infected hosts, depending on the botnet infrastructure. This requires the integration of a communication protocol into the malware. Two ways of specifying a firm contact point are available for this purpose:

- Fixed IP addresses can be integrated into the bot, executable upon distribution.
- A domain name (or set of domain names) can be defined that will be contacted when the host system is compromised.

Use of a domain name offers flexibility in various ways. First of all, one domain name may be associated with multiple IP addresses, helping to create a redundant architecture that is more robust against machine takedowns. These IP addresses do not have to be static but can be changed dynamically on demand. Fast-flux service networks based on the Domain Name System and described in more detail in section 1.2 (Components of Botnet Infrastructures), are a consequent evolution of this approach. The contact information needed to register a domain is usually forged by botmasters and therefore not usable for investigations.

Once a malicious domain name has been identified, it can be used for further actions. For example, passive DNS replication [66] can be used to collect DNS information from servers and archive it for later processing. This enables, for example, the identification of hosts that have queried a malicious domain name before this domain was flagged as malicious, or the automated detection of “typo squatting” [67]. Typo squatting is the approach used to achieve a high number of visitors ‘by accident’ to a webpage by using domain names derived by altering the spelling of legitimate high-profile domains. Famous examples are

- “goggle.com” instead of “google.com”, which has infected visitors with spyware in the past.
- “facebooi.com” instead of “facebook.com”, which even adopts the design of the original social networking platform.

The latter case demonstrates the potential for this technique for use in phishing.

From a technical point of view, malicious domain names can be blocked easily by registrars at a global level or by the operators of DNS servers within a local scope. Alternatively, the domain name may be taken over in cooperation with the operators of DNS servers in order to monitor the incoming requests for the botnet command-and-control server. More on this approach, known as “sinkholing”, can be found in section 3.2.1 (Sinkholing).

In contrast to this, if one or more fixed IP addresses are implemented in the malware, then taking down the associated servers behind those IP addresses will render the specific malware samples that rely on them useless to the botnet. With the use of direct IP addresses, no DNS queries are needed, and the botnet cannot be taken down by deregistering the domain name.

This leads directly to potential measurement techniques. If a domain name has already been identified as malicious, it is much more likely that all incoming queries for this entry are issued by infected hosts. It therefore enables the infected machine to be tracked whenever it uses a DNS server. Furthermore, sophisticated methods can be developed using the characteristics of queries for malicious domain names, typically including features like spatial or temporal relationships. This enables approaches to be applied based on anomaly detection on the domain name system, as described in the following.

Example: Choi et al. [68] presented an anomaly-based detection approach for C&C servers and bots that targets special properties in the query behaviour of bots. They observed that botnets tend to exhibit coordinated behaviour they called “group activity”. For example, each time a C&C server has a link failure or is migrated to a new domain name, it results in an event affecting all participating bots. The associated members of the botnet will almost simultaneously start to query for their missing C&C server. Furthermore, coordinated malicious activities like DDoS and the dispatch of spam will also be likely to result in DNS group activity that gives information about individual bots participating in these activities. Additionally, the use of dynamic DNS (DDNS) services is often an indication of C&C servers, which is also incorporated into their analysis model.

Example: Villamarín-Salomon and Brustoloni [69] also presented an anomaly-based detection approach for C&C servers. In their work they compared different algorithms and evaluated their efficiency. They concluded that abnormal amounts of recurring NXDOMAIN-responses are a suitable indicator of the presence of a botnet. A NXDOMAIN-response is generated by a DNS server if a domain name cannot be resolved. In the context of botnets, this is often the result of a takedown or migration of a C&C server. Furthermore, they stated that the approach based on NXDOMAIN-responses is less prone to false-positives than the other algorithms evaluated, as it is likely not to report high profile sites that use low TTLs for load balancing.

Example: Another approach that compares with the above-mentioned host-based

method was presented by Morales et al. [70]. They explored how processes react to a response received from a DNS server after a query. Apart from direct connection attempts made in order to join the botnet after resolving a domain name, some malware specimens they analysed also made reverse DNS queries and sometimes even on the IP addresses received initially. The intention of such behaviour is to obtain additional domain names connected to the botnet in order to create redundancy. Based on these findings, Morales et al. created a heuristic that can help to improve the detection rate of host-based malware detection tools.

Example: Musahi et al. [71] have studied changes in DNS traffic patterns on computers infected with mass mailing worms. They observed that receiving email via the Post Office Protocol (POP) and Simple Mail Transfer Protocol (SMTP) generates considerably less DNS traffic than sending email via SMTP. To explain this behaviour they identified that emails usually have multiple destination domain names that have to be resolved. Where there was an infection from a mass-mailing worm, this SMTP-related traffic increased notably, enabling the researchers to detect infections solely through DNS traffic.

3.1.4 ANALYSIS OF SPAM RECORDS

One common purpose of botnets is the distribution of unsolicited email, known as spam. A fairly indirect approach to measuring botnets and their corresponding activities is the analysis of spam records. In this context, indirect means that, instead of observing communication such as command and control messages, information is derived from the investigation of the spam messages sent by a botnet. Obviously this method will only observe botnets that actually perform spamming. To work efficiently, mapping between spam messages and botnets must be created. This is often possible because spam is likely to be organised in so-called spam campaigns. In this context the term 'campaign' describes the actual messages' property of being identical or at least having a common observable pattern. As spam mails have to be generated by the bot, they will follow a similar pattern, which forms the basis of the actual generation process. Even though the content of the message itself provides a good starting point for such a matching, characterisation also takes into account more properties, such as the characteristics of the underlying SMTP conversation and corresponding protocol fields in the email header.

All these properties enable spam messages to be compared, aggregating them into spam campaigns and finally associating the clusters retrieved with a corresponding botnet. To do this efficiently, spam templates can be extracted from bots that act as drones when they are executed in a controlled environment. Based on spam message headers, it is possible to draw conclusions about the location of bots and thus about the global distribution of the botnet. Clearly, only bots participating in spam campaigns are recognised at all. For various reasons, bots may participate in the botnet but are not able, or commanded, to send out spam mails. One possible reason for this is that a

module for sending spam messages is missing from the bot's implementation or has not been installed on the victim's machine. Alternatively, the bot can decide not to send spam, if it is suspected to be running inside an analysis environment.

In terms of detection, the deployment of spam traps can be a beneficial addition to this approach. Spam traps are email addresses with no productive functionality other than to receive unsolicited email, and so they can be considered as a special type of honeypots, usually referred to as honeytokens [72]. The difference between these two instruments is that spam traps have to be advertised, e.g. by subscribing to a lot of newsletters, discussion forums etc. Where spam is used as a spreading vector, the evaluation of spam messages with respect to attachments and included links can lead to the detection of as yet unknown malware families.

Example: Kreibich et al. [22], [73] have performed a deep analysis of the spamming behaviour of the Storm botnet. They were able to extract different data sets from the botnet. Three approaches were taken:

- Spam email templates were harvested over the C&C infrastructure. A crawler bot was used, frequently requesting new templates.
- They installed their own layer of proxy nodes inside the botnet. This enabled them to modify spam email templates and include their own information, which was then used to deduce information about spam email conversion and success rates.
- Crafted "marker" email addresses were injected into the harvesting process of the botnet. This enabled them to investigate the use of email addresses extracted from victim computers for further spamming.

The main results of their studies are:

- Email addresses harvested from victim computers are seldom found on the Internet. As they are therefore likely to be unreachable through crawling of web pages, this indicates the value to the botmaster of harvesting email addresses.
- The evaluation of more than 460,000,000 spam emails resulted in the following statistics. Only 1 in 12,500,000 pharmacy spam mails led to a purchase. Alternatively, seasonal emails seem to be more effective: 1 in 265,000 greeting card spam mails and 1 in 178,000 April Fool's Day spam mails would have led to an infection. Furthermore, 1 in 10 people visiting a prepared website downloaded and ran an executable that was offered.
- On average, domain names used in spam mails are usable for a few days only before they are included on blacklists

Example: Taking into account that the control channels of modern botnets are becoming stealthier, Hao and Feamster [74] developed a new method for estimating botnet population and membership dynamics by analysing attack traffic, e.g. spam

mail records. They showed that even an examination of small subsamples of just a local spread can be sufficient for a reliable estimate of a global perspective. The cross-validation of their results with data from another research group, which measured the Storm botnet with peer-to-peer enumeration (for more information on this technique, see section 3.2.6 Enumeration of Peer-to-Peer Networks), showed that their spam-based method produced only a 4%-10% deviation in this case.

Example: Dunagan et al. [75] showed that the analysis of large amounts of spam can be successfully used to map identified spam bots to distinct botnets. For this, they made substantial use of the concept of spam campaigns. The main identifiers for these campaigns were a limited group of related topics or advertised products, or similar phrases used across e-mails.

3.1.5 ANALYSIS OF (APPLICATION) LOG FILES

Modern applications and computer systems are designed to not only perform a specific task but also to record their activities in the form of accounting and logging. This typically implies recording (inter-)actions and writing these records into files in a processable or even human-readable format. These records are called log files. They can be understood as data records that offer an (ideally exhaustive) protocol of the applications' activities, including requests received over a network or the Internet and the corresponding replies.

As with spam records, the analysis of log files is an indirect measurement and detection method which deals with data records accrued as a consequence of bots' activities. Hence both approaches have basic similarities. Log files which become a target of this examination may have various sources, such as firewalls, network services or infrastructure devices. Analysis can be performed in parallel on multiple input sources.

While techniques for the automatic discrimination of spam from regular emails have reached a sophisticated level and work with satisfying efficiency (in the end, manual classification is feasible anyway), it can be quite difficult to differentiate between regular, benign processing and abusive behaviour caused by bots by just considering log files. An example of an obvious case would be a distributed denial of service attack targeting a web server, where the analysis of log files could yield a list of participating IP addresses. In particular cases, such a distinction may not be possible at all, either locally or from a distributed viewpoint. So the method can only be applied when characteristics of malicious behaviour can be clearly identified in order for mechanisms for automated classification to be developed.

Attempts have been made to discover abnormal events and anomalies, such as the atypical utilisation of a service or conspicuous and nonsensical sequences of requests and queries. In this context, utilisation means both quantitative [76] and qualitative. For example, a significant increase in requests for a network service can expose botnet

activities. As mentioned, even the way a certain service or protocol is used in terms of syntactic or semantic peculiarities can help to reveal a bot, especially when typical interaction patterns are taken into account.

It is generally observed that bots access URLs and hence give rise to recognisable log entries for various reasons. For instance, the Conficker botnet uses simple HTTP requests aimed at high-profile websites (such as Yahoo.com, CNN.com, etc.) to synchronise the bots' internal clocks and verify Internet connectivity [77]. At first sight, log entries created through this behaviour will look like ordinary HTTP requests, which could originate from a benign user interacting with the website. This illustrates the difficulty with differentiating behaviour based on malicious code from regular activities. Characteristics can be ambiguous. In this particular case, Conficker will only query for the index page itself, but not for the content linked on the index page (like pictures and scripts), which is usually loaded by a web browser. Also, if an identical mechanism is used by two or more distinct botnets, only the presence of the bot can be detected, not the botnet it is associated with.

Example: By concentrating on anomalies in application log files, Linari et al. [76] showed that botnets can be detected and measured without necessarily making any assumptions about the underlying architecture. They demonstrated their approach in a case study in which anomalies in the querying patterns of the distributed WHOIS service were used to detect and track botnets' activities.

3.1.6 HONEYPOTS

A 'honeypot' is an intentionally vulnerable resource deployed in a network with the aim of soliciting attacks or even compromise by a malicious entity. Although this definition seems rather wide, it is necessary to capture the versatility of honeypots and how they fit with the various ways this concept has been realised. Two features are primarily used to distinguish between the two categories of honeypots: client and server honeypots and low interaction honeypots.

The main reason for researching and developing honeypots is to discover new information about the practices and strategies used by creators of malware and hackers. In general, two kinds of information can be gathered by honeypots:

- Types of attack vectors in operating systems and software used for attacks, as well as the actual exploit code which corresponds to them.
- Actions performed on an exploited machine. These can be recorded, while malware loaded on to the system can be preserved for further investigation.

Of particular interest to botnets is that honeypots can be used to measure automated attacks originating from botnets, and so lead to the identification of infected systems. Furthermore, the automated extraction of malware binaries can lead to the detection of new botnets.

Client honeypots aim to emulate regular behaviour performed by a user or software. They are mainly characterised by actively trying to get attacked. By contrast, server honeypots offer an emulated, legitimate service and stay passive while waiting to be attacked.

In addition to these two categories, honeypots may be characterised by the degree of interaction when responding to actions [78].

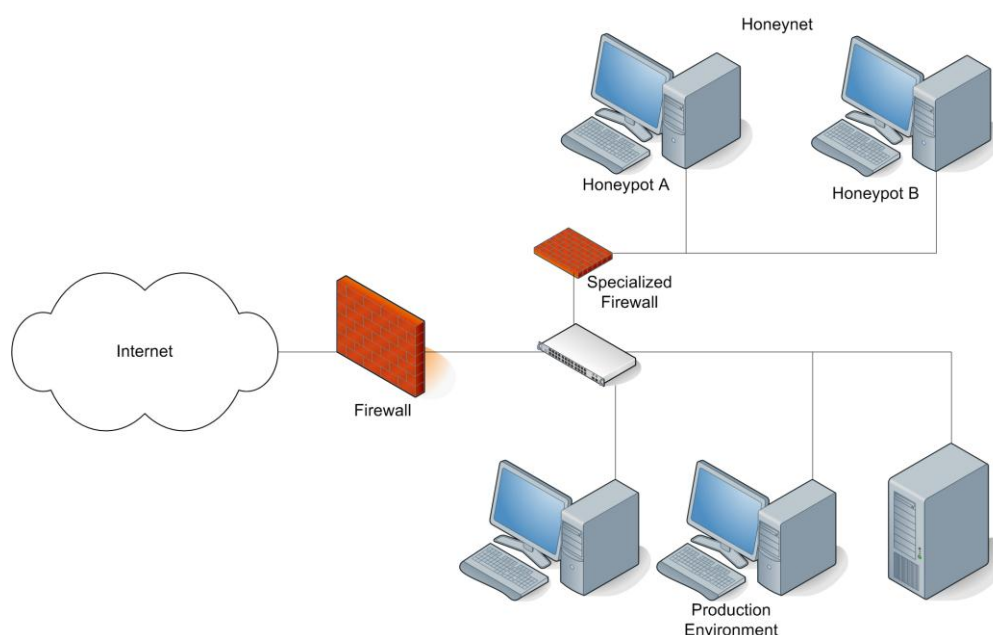


Figure 6: High-interaction honeypots.

'High-interaction honeypots' (cp. figure 6) are fully native systems deployed with the purpose of being infiltrated. Since such a system can be completely compromised and used for further malicious activities, it is very important to limit the environment to specific actions. High-interaction honeypots usually have hidden components installed that enable all of the attacker's activities on the system to be monitored. Often, more than one high-interaction honeypot is deployed at the same time with the aim of covering different operating systems or a variety of services. Parallel processing is allowed either by forwarding traffic simultaneously to all of them or by even simulating a sub-network, which is then referred to as a honeynet.

A general note on high-interaction honeypots relates to the responsibility connected with their deployment. The most important objective still has to be to prevent any damage being carried through from a compromised system to other systems in the same company or beyond. Therefore it is common for traffic coming from honeypots to be specifically filtered.

Low-interaction honeypots only simulate certain network services or behavioural aspects of an operating system. This gives them the ability to separate the underlying operating system from the target system presented to the attacker. For example, a selection of several software flaws written for Windows can be exposed to the network via a honeypot running on a Linux machine. This reveals a major limitation of the low-interaction concept. Since just part of a whole system is simulated, this variant offers only a limited view of the attacker's techniques. The use of simulation also offers several other benefits.

- The Packet filtering may also be applied at ISP of parallel observable attacks through the use of simulation is limited only by the host system's computing capacity.
- Exposing only a set of services, instead of a full operating system, offers the additional benefit that no safe state has to be restored.
- Ideally, exploitation of the apparently vulnerable service does not lead to a compromise of the underlying system. This helps to keep maintenance of the honeypot low and also offers good scalability, as multiple vulnerabilities can be emulated in parallel.
- Because the system on which it is running remains trusted, it can be used directly to analyse the attack.

Advanced implementations of honeypots even support modularity in the context of exploitable services, automatic post-processing of attack logs and report generation. Popular implementations of low-interaction honeypots are, amongst others, Dionaea [79], its predecessor Nepenthes [80], mwcollected [81], Honeytrap [82], and honeyd [83].

In general, a honeypot's benefit depends on its ability to classify network traffic and internal activities, e.g. changes to the files system or memory. This creates a need for close and complete monitoring of processes. It is critically important to have sophisticated approaches for the analysis and interpretation of malicious behaviour in order to recognise as yet unknown (zero day) attacks.

Honeypots play an important role in the context of analysing botnets. They are used as a tool to detect scanning events and malware outbreaks. They enable the identification and enumeration of hosts involved and can automatically gather malware specimens for later analysis. Of course, a honeypot can only see incoming traffic from the IP addresses assigned to it. This means that general conclusions about the size and activity of botnets can only be drawn by means of extrapolation.

In this context, a network telescope, or darknet, can be used to provide input for honeypots that are aggregated in a honeyfarm [84]. A darknet can be created upon the IP addresses that are assigned to an institution by the Internet Assigned Numbers Authority (IANA) but are actually not used. Examples of active darknets working on a

global scale are the ATLAS system of Arbor Networks, those supervised by Team Cymru [85] or the darknet operated by the Cooperative Association for Internet Data Analysis (CAIDA) [86].

Several institutions running honeypots publish the gathered data in the form of statistics or detailed, automatically-generated reports. In addition to ongoing monitoring projects, Honeypots have also been extensively used and analysed in several research projects.

Example: Li et al. [87] employed a combined darknet and honeynet, consisting of 2,540 addresses from 10 continuous class C networks, and analysed the incoming traffic for a year (2006). Half of the sensors were non-responsive and the other half used honeyd, which is short for honey daemon, a honeypot implementation [83]. After filtering the incoming traffic, they observed 43 global scan botnet events carried out by 63,851 unique sender addresses. They discovered that 75% of all successful scanning events led to an attack with a malicious payload.

Example: Goebel et al. [88] deployed the honeypot software nepenthes in a university environment to observe 16,000 addresses for eight weeks from Dec '06 to Jan '07. During this period, they gathered 13,400,000 malware binaries, from which 2,558 had a unique MD5 hash. Three further steps of analysis were executed after the initial collection of malware samples:

1. GFI Sandbox (formerly CWSandbox) [89] was used to characterise the malware's behaviour.
2. Four different antivirus scanners were used to diagnose the malware family.
3. Analysis was accomplished with the software botspy [90], a tool which automatically extracts information about the remote control functionality of a given binary.

Through these samples, they identified a total of 40 C&C servers from their unique IP addresses and/or domain names.

Example: Pham and Dacier [91] carried out a long-term study on low-interaction honeypots. They evaluated results from 40 deployments of multiple honeyd [83] instances, each running for more than 800 days. By splitting the data set by the originating countries of the attacking machines, identified through source IP addresses and by attacks separated through honeypot deployments, different attack event correlations were observed. Amongst other results this underlines, in particular, the importance of distributed deployments in obtaining a broad view of attacks. By applying a similarity measure, single botnets, or groups of botnets with common characteristics, could be identified, showing lifetimes of more than 750 days.

Example: Small et al. [92] presented a low-interaction, web-based honeypot with the ability to create dynamic responses to malicious queries. Their main goal was the

detection of bots that employed search engines to find vulnerable hosts. While running their experimental set-up for 72 days, they recognised more than 386,000 attacks against their honeypot, targeting about 45,000 unique vulnerable scripts emulated by the honeypot. Because many web-based exploits use script injection, this malicious content has to be hosted somewhere in the Internet. An analysis of URLs included in the attack scripts collected showed 5,648 distinct script repositories. This gives an impression of the number of comprised hosts used for such attacks. With their approach, they were also able to discover more than 10 zero-day attacks that were deducted and classified by the system. Both results can be used in further botnet mitigation.

3.1.7 EVALUATION OF ANTI-VIRUS SOFTWARE FEEDBACK

In recent years, providers of anti-virus suites implemented solutions based on distributed information processing, which incorporate feedback from software installed on their users' machines. In these models, their customers perform the role of sensors, forming a large sensor network. These approaches are also often called 'cloud anti-virus solutions'.

Depending on the degree of detail this feedback function provides, the approach enables various anti-virus companies to react faster to emerging threats and to gain information about the quantity and geographic distribution of potential infections.

The main difference from classic anti-virus products, where only updates are downloaded by customers, is the two-way communication between the AV provider's infrastructure and the customer's machine. If the anti-virus software installed locally on the user's machine encounters a suspicious object, e.g. an executable file or, in the case of network activities, an IP address or domain name, it can send a request to the cloud in order to obtain more information on how to treat the object. If no information is available, the object is analysed and an entry is generated [93].

Example: A prominent example is Microsoft's Malicious Software Removal Tool [94]. Machines with this software installed provide feedback to Microsoft, which is then incorporated into the biannual Security Intelligence Report. The latest report (January 1 – June 30 2010) included data from more than 600 million systems worldwide [52].

In addition, there are services which offer the opportunity to upload files and to run virus checks on them, like virustotal.com [95] and jotti.org [96]. Some websites even provide freely-usable sandboxes to perform a behavioural analysis on submitted files. Examples are GFI Sandbox [89], Anubis [97] and Norman Sandbox [98].

The major advantages of these submission services are:

- Sandboxes complement regular AV tools, as they can analyse single files in more depth by providing a closed environment.
- Submission of files can be interpreted as a sensor for the detection of new

malware, therefore as a precursor of further investigations and actions.

3.2 ACTIVE TECHNIQUES

The group of ‘active measurement techniques’ contains approaches that involve interaction with the information sources being monitored. Although this enables the performance of deeper measurements, their application may leave traces that influence results, or include activities that can be observed by the botherder. This can cause reactions, such as a DDoS attack against the analyst or the introduction of changes to the botnet structure that will complicate measurements, even including migration of the service to evade monitoring.

First, two general techniques for active measurement are presented. Both can be applied in a very intrusive form that is already connected to possible botnet countermeasures, but they may also be applied silently. The following approaches focus on certain protocols used in botnets or special types of botnet architecture.

3.2.1 SINKHOLING

In general, the term ‘sinkholing’ describes a technical countermeasure for cutting off a malicious control source from the rest of the botnet. It can be performed against a variety of targets, most likely against botnets’ command-and-control servers or trojan dropzones. For more details on this topic, see section 4.1.3 (DNS-based countermeasures).

One of the most common variants of this technique is changing the targeted malicious domain name so that it points to a machine controlled by a trusted party, usually investigators or researchers, as depicted in figure 7. A similar effect can be achieved by changing the routing of a static IP address in the same way. Both approaches, with little additional effort, enable the setting-up of analysis frameworks to measure the size of a target botnet. Apart from sinkholing connections, transparent forwarding to a secondary destination is also possible, but it is usually less desirable because it keeps the botnet alive. This may prove useful where the botherder fails to detect ongoing investigations and as part of the preparation of a larger, coordinated takedown of several of the botnet’s C&C servers.

The basic principle of this measurement relies on the fact that the domain is very likely to be contacted by infected machines trying to reach their command-and-control server. As the domain names and IP addresses used for such purposes often only serve malicious purposes, this approach tends to have a low false-positive rate.

One drawback of this technique is that measurement accuracy depends largely on the information available to the target host. If the bots contacting the sinkhole provide information that can lead to unique identification, the achievable accuracy should be fairly high. On the other hand, if the amount of extractable information provided by the contacting hosts is very low, the measurement results will be influenced by a significant variance. For example, where all the incoming packet payloads are fully encrypted, only the IP address (and other details of packet headers) may be used as a feature for identification.

Measuring through unique IP addresses implies various aspects that can markedly affect the results, as illustrated by the following examples. On the one hand, if multiple infected machines are connected to a single router that appears with a static IP address outside the LAN, multiple infections will be counted just once. On the other, if many of the IP addresses are dynamically allocated, this can introduce multiple counting to the statistics. This is particularly so with Internet access from mobile devices, when several mobile addresses are translated via central gateways [99] or if a customer frequently connects and disconnects, obtaining a fresh IP address each time.

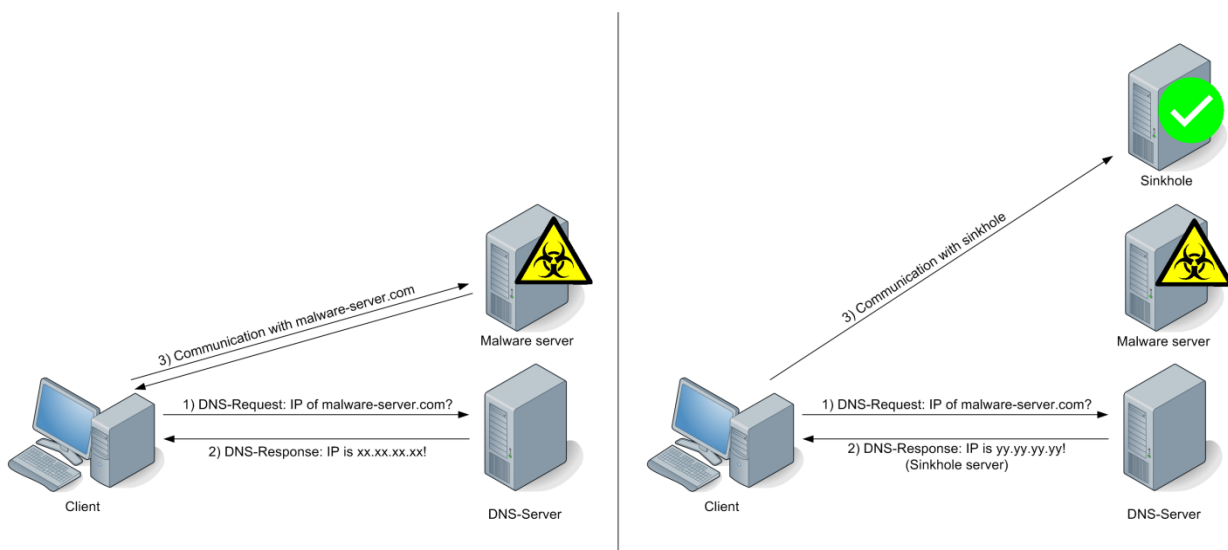


Figure 7: Sinkholing.

The operators of a sinkhole have to take into account that incoming data will often contain serious amounts of sensitive content. For example, if the sinkholed server receives data from malware with features for identity theft, the incoming packets will include the credentials for various accounts or financial information, such as credit card or bank account data.

Example: This technique may also be applied against decentralised botnets that use dynamic contact points for occasional updates or injecting commands. A way to design contacts dynamically is to generate domain names that have limited validity, e.g. for

one day only. Probably the best-known case where this technique has been intensively applied since February 2009 is against the botnet known as Downadup or Conficker. The Conficker Working Group coordinates the blocking of 500 domain names per day and tracks any connection attempts to these via sinkholing. The results of this effort are shown on the project's website [49]. At the time of writing, Conficker-related domains are still contacted by an average 5-6 million unique IP addresses per day.

Example: This approach to measurement experiments was examined in detail by, for example, Dagon et al. [100]. After capturing malware samples through various sources like honeypots and spam traps, they analysed them for potential DNS query commands. The next step consisted of identifying domain names included in the malware and resolving their authoritative name servers. When the domains were identified as active, they contacted the responsible registrar and DNS authority. In cooperation with these institutions, they arranged subsequent requests for the malicious domains to be forwarded to their logging machine (sinkhole). With this technique, they uncovered a botnet with more than 350,000 hosts, which was the largest known count of bots in a single botnet at the time of their experiments. This number was derived by counting the number of unique IP addresses that contacted the sinkholed domain name during the period of observation. Therefore the number may be influenced by the extent to which DHCP is used, as they themselves stated. A major finding of their studies was the markedly diurnal character (i.e. having a daily cycle) of activity in the botnet caused by the geographic locations of infected hosts distributed over several continents. Based on the data obtained, they created a diurnal model of botnet propagation and examined the impact of the release time of new malware on the propagation.

Example: The most recent use of sinkholing for measurement purpose at the time of writing was by Symantec in the case of W32.Stuxnet [45]. By monitoring the traffic to the Stuxnet command-and-control servers, statistics could be derived about infection rates or global distribution. From July 20 to September 29 2010, approximately 100,000 infected hosts were identified through traffic monitoring. Distinct hosts were identified by their combined characteristics, such as computer name, OS version, and internal and external IP address, which could be extracted from the incoming status data packets sent by the targeted hosts. In total, over 40,000 unique external IP addresses were observed. This illustrates that counting bots only by unique IP addresses can lead to a significant degree of uncertainty, both in over- and underestimating the number of bots.

3.2.2 INFILTRATION

The ‘infiltration’ of botnets can be divided into software- and hardware-based techniques. The first covers research on the bot executable and monitored traffic to achieve control and conduct measurements. The latter can be applied if access to the command-and-control server is possible and may be used to wiretap the communication. This includes physical machines as well as virtual machines that might be running in a data centre.

Software-based infiltration extends the ideas of the enumeration approaches presented in sections 3.2.5 (IRC-based Measurement and Detection) and section 3.2.6 (Enumeration of Peer-to-Peer Networks). Instead of emulating or modifying the bot software on a controlled host with the intention of joining the botnet and measuring it internally, infiltration goes a step further and aims to take control of the botnet.

This usually requires as its starting point the reverse-engineering of the communication mechanisms used by the botnet. Such a precise analysis may lead to the identification of potential weaknesses. This procedure may be compared to a security audit or penetration-test of the botnet and its infrastructure. Knowledge obtained in the process can be exploited in further steps to achieve a commanding position inside the botnet. This may lead to the possibility of performing measurements or revealing information about infected hosts, or even the botherder.

The other approach, hardware-based infiltration, may be applied if an IP address belonging to a command-and-control server has been identified and a relationship to a data processing centre or hosting company can be established. By obtaining a connection to a mirror port on the suspected servers, the communication can be wiretapped and analysed. This enables all traffic to and from the server to be monitored, which also allows information about number, location and other attributes of infected hosts to be gathered. The limitations of this approach are comparable to sinkholing. For example, traffic encryption can reduce the number of usable attributes and therefore influence the accuracy of the measurements.

3.2.3 DNS CACHE SNOOPING

The measurement technique called DNS Cache Snooping [101] is based on the caching property implemented and used by many DNS servers. If a DNS server is queried for a domain for which it has no entry defined, it will issue a query towards the responsible authoritative name server on behalf of the querying client and store the resulting data record afterwards in a local cache. Caching is mainly used to increase the performance of a name server and reduce its traffic load.

The functionality may also be used in an unintended way, for measurement purposes. The central idea is to check indirectly if a target domain has been queried through a specific domain server by testing if a cached answer is stored, as shown in figure 8. Two variants can be used for this; which one to use depends on the configuration of

the DNS server.

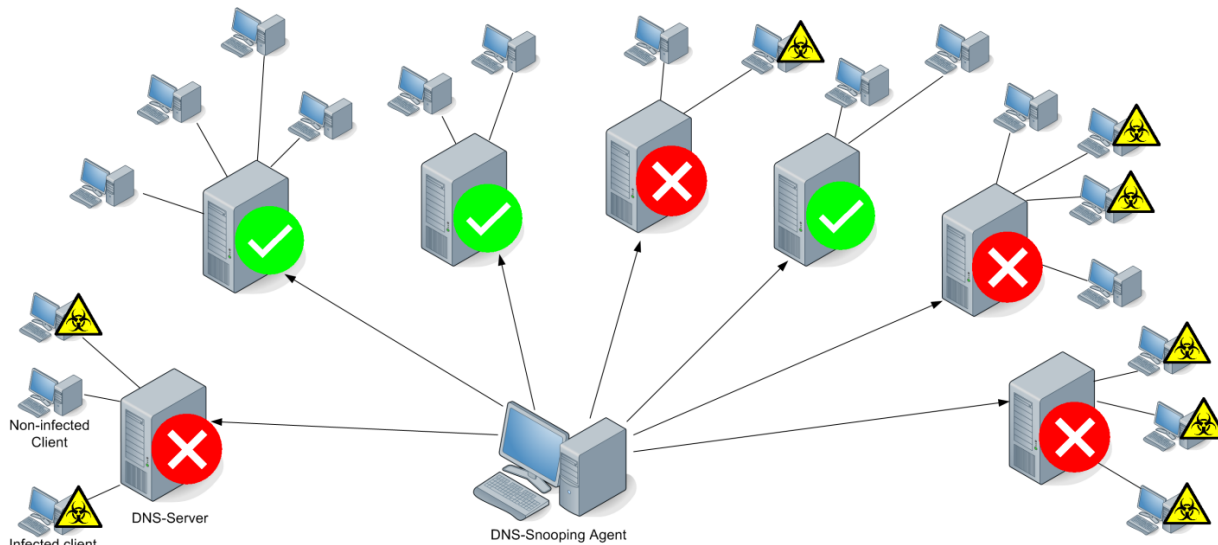


Figure 8: DNS cache snooping.

In the first variant, a query is sent to the DNS server with a special variable, the no-recursion-flag set, that will prohibit the server from forwarding the query to the responsible authoritative name server. The server's behaviour will differ, depending on whether the originally-queried server has stored a cached answer for the targeted domain name. It will either directly send an answer to the query, including resolved IP addresses, or, if the DNS server cannot answer, it will send an answer containing authoritative servers that will be contacted further. This method will not work with every DNS server, as many servers will simply refuse special queries like this in order to protect themselves against Denial-of-Service attacks.

The second variant will work with any DNS server, but will pollute the targeted server's cache. In this case the variable mentioned is not set and the server is allowed to forward the query to other name servers. This makes it possible to determine whether or not an answer is already cached by analysing when the specific domain name was last queried. This can be achieved by evaluating the Time-To-Live (TTL) value included in the response. If this value is set to the server's default value (which can be determined through querying for a non-existent domain) no information will have been cached before the query. If it is lower than the default value, the domain would already have been queried before, because this indicates that the server's TTL counter for this domain has already been started by a query made before the snooping query. This technique is possible, because the server does not refresh the TTL value for a cached domain name when it is queried. One drawback of this variant is that the analyser's query will leave a cache entry, which will block the possibility of using this technique on the server again for the server's caching duration.

The technique has good scaling, as the targeted DNS servers are usually not related to each other, which allows full parallelisation of the process. Constraints arise regarding the number of domains that are subject to tracking. Many DNS servers limit the number of queries they will process for one client IP address. This means that the tracking of domain names might have to be staggered. Depending on the total number of tracked domains, this will result in delayed results.

The general information value of this method depends on how many DNS servers are incorporated into the tracking. It is desirable to widen and maintain a list of as many servers as possible.

Concerning the accuracy of this approach, it is fairly obvious that it is not suitable, or intended, for revealing the actual size of a botnet. Instead, it is useful for exploiting the natural partitioning of the Internet, provided through the relationship of hosts identified by IP addresses and the DNS servers that are usually responsible for them. This enables information to be deduced on how widespread the connection of certain malware to a domain name is among those defined partitions. Furthermore, of the knowledge that a DNS server has been queried for malicious domains can lead to further investigations if, for example, the DNS server is hosted by a company and responsible only for its own network.

Example: Rajab et al. [102] have performed an extensive analysis on cache snooping. They first created a model for normal behaviour of domain name servers in response to queries by arbitrary hosts in order to extrapolate cache hits. With this, they could derive an estimate of the presence of cached name entries by repeatedly querying the DNS servers and measuring the arrival times of DNS requests. They validated this model in an office environment, where data was available from both the DNS servers themselves and cache snooping. For the following experiments, they used a list of 768,000 co-operative DNS resolvers to monitor cached domain names related to botnets and performed cross-validation with IRC logs to reveal the botnet's actual size.

3.2.4 TRACKING OF FAST-FLUX NETWORKS

Some botnets use fast-flux networks, as described in section 1.2 (Components of Botnet Infrastructures), to introduce covertness of their activities and increase the reliability of their network and command structure. Fast-Flux networks use rapidly-changing DNS records, pointing at a large number of hosts, acting as an additional proxy layer to hide the actual content-delivery systems. The proxy nodes are usually compromised hosts of the botnet itself.

By associating a large volume of IP addresses with only a few, or even single, domain names, the network becomes much more robust against countermeasures. However, the very specific properties of DNS records serving this type of network enable them to be distinguished from benign networks. Records of domains connected to fast-flux networks typically have a short validity period of just a few minutes. This is indicated by the Time-To-Live (TTL) value included in the response [101] generated by the

original DNS server at the end of the DNS query chain, which is under the control of the botmaster. A malicious DNS server is usually operated directly by the botmaster, or installed as a service on a compromised server. After this period has passed, a new query will usually result in a different set of associated IP addresses with little to no similarity or topological relationship to each other. An additional, and even more characteristic, feature that can be observed is the variety of IP addresses returned for a fast-flux domain. Such IP addresses will usually originate from several networks and ISPs. Benign services with a low TTL, e.g. high-profile websites like google.com or facebook.com, will usually return IP addresses that have strong similarity, indicating that they originate from the same network and are related to each other.

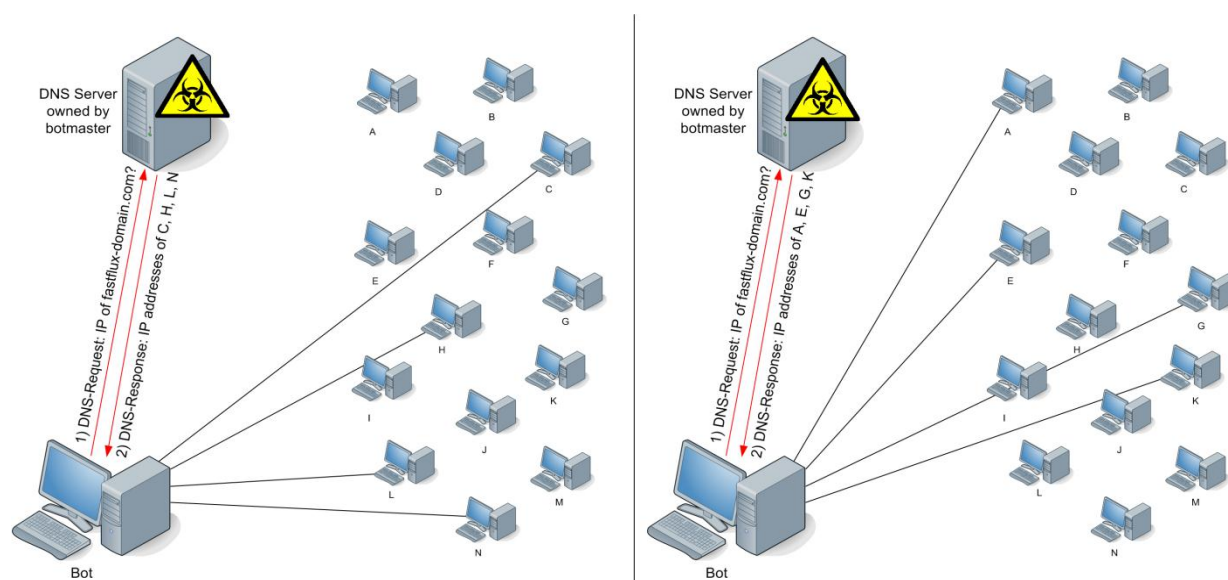


Figure 9: Tracking of fast-flux service networks.

Monitoring domains whose DNS responses feature a low TTL not only enables the identification of fast-flux domains, but by issuing repeated DNS queries hosts participating in the network may be extracted and collected from these records. This is shown in figure 9.

Example: Nazario and Holz [103] presented the results of their monitoring of fast-flux botnets. They used Arbor Networks' Active Threat Level Analysis System (ATLAS) [104], which is a globally-distributed data network consisting of honeypot-like sensors, Denial of Service monitors and other data aggregators. The functionality for tracking fast-flux networks was incorporated in 2008. The system uses various sources for potential fast-flux domains like URLs extracted from spam mails and domain blacklists, as well as domain names from automatically-analysed malware samples and those found by manual analysis. These candidate domains are then rated with a heuristic based on multiple DNS queries that includes, for example, the TTL, number and distance of the answering IP addresses and their AS numbers, together with

comparable characteristics for the name servers listed. The data in this publication was gathered and analysed from January to May 2008. During this period, 928 different fast-flux domains with a total of over 15 million associated IP addresses (some mappings were counted multiple times) were identified with the above heuristics. With regard to the lifetime of fast-flux domains, Nazario and Holz found that more than a third of domains were active for less than a week, with a peak of just one day or less. Nevertheless, the average usage time for the fast-flux domains observed was 18.5 days. Another related finding was that the identified domains used for fast-flux were, in 80% of all the cases, activated more than one month after they had been registered. As for botnet measurements, ATLAS was used to track, amongst others, the Storm botnet, which used fast flux. Further analysis and comparison with results from a peer crawler for the Storm botnet revealed that, on average, only about 1% of the active infected hosts are advertised through the fast-flux network. The main reason for this is that IP addresses from the private address space may not be used in DNS answers [105], which is also a drawback to this technique in terms of botnet measurement. Another reason might be that some participating botnet hosts do not meet the bot herder's requirements (e.g. bandwidth or uptime) for a fast-fluxed host.

3.2.5 IRC-BASED MEASUREMENT AND DETECTION

Today, Internet Relay Chat (IRC) still serves as a common command-and-control (C&C) infrastructure for botnet administration. According to the 2010 Symantec Annual Report, 31% of all identified C&C servers in 2009 used IRC as their communication protocol [6]. IRC is a lightweight and robust chat protocol, offering a lot of functionality suitable for controlling even large botnets. An IRC-based C&C design implies one or more IRC channels, either on public IRC networks or on custom (sometimes compromised) servers, where bots report their presence and receive their commands. Typically, newly-launched bots will, as their first action, connect to such a channel and wait for further instructions.

In order to measure bots using an IRC-based communication model, it is first necessary to obtain the information needed to join and participate in the botnet channel. Basic connection information, which involves the IP address and port number of the IRC server, as well as the channel used for controlling the botnet, can be extracted from malware samples of the botnet concerned. Depending on the complexity of the botnet communication mechanism, the login or authentication credentials, and also encryption functionality, may have to be extracted and understood.

As a second step, these IRC channels can be joined and data can be collected. The quality and amount of information obtainable depends heavily on the precautions taken by the bot herder. If the control channel is hosted on a standard (or even public) IRC server, it may be possible to simply record and track usernames in the channel without

further efforts. IRC status messages give additional information about fluctuation inside the botnet. Together with the botnet population, the command set and malicious activities performed can be observed when the communication between bots and botherder takes place over a public channel.

Where the IRC server is hosted by the botherder himself, modified implementations have been observed. These are tailored to the needs of the botnet. For example, they can be a reduced set of commands, just enough to instruct the bots. Often, “classic” commands included in the regular IRC chat protocol are removed and public chat is disabled. Communication is realised via private messages between botherder and bots only, the intention of this being to hide the presence of bots from each other. The protocol may also use encryption. This severely limits the amount of information that can be gathered by residing in the command-and-control channel.

3.2.6 ENUMERATION OF PEER-TO-PEER NETWORKS

Even though IRC is still a prevalent technology for botnet control, other communication schemes have gained in importance. Another common approach used by botnets is to employ a peer-to-peer (P2P) based infrastructure. The basic idea is to create an overlay network with its own addressing scheme and protocol for routeing messages between participants only. Some of the botnets that achieved the most media coverage, e.g. Storm, Waledac and Conficker, used peer-to-peer mechanisms.

In a peer-to-peer botnet, the bots form a loosely-coupled network in which each peer knows only a limited group of other peers. In terms of the network they can be seen as its neighbours. These relationships provide a robust architecture, without the need for centralised servers to administrate the botnet. Consequently, this architecture no longer has a single point of failure and is much more robust against countermeasures such as the shutdown of central command-and-control servers. Each node in this type of network is usually identified by a unique key, combined with additional information like IP address and port number. In the case of the Storm botnet, the Kademlia P2P protocol and the existing Overnet peer-to-peer network were used. In this P2P infrastructure, keys are also used for indexing and searching peers and for routeing messages. Each peer maintains a list of these keys to keep connectivity to the network alive. As only a comparatively small set of peers is known to each peer, information about the whole botnet is distributed within the peer-to-peer network.

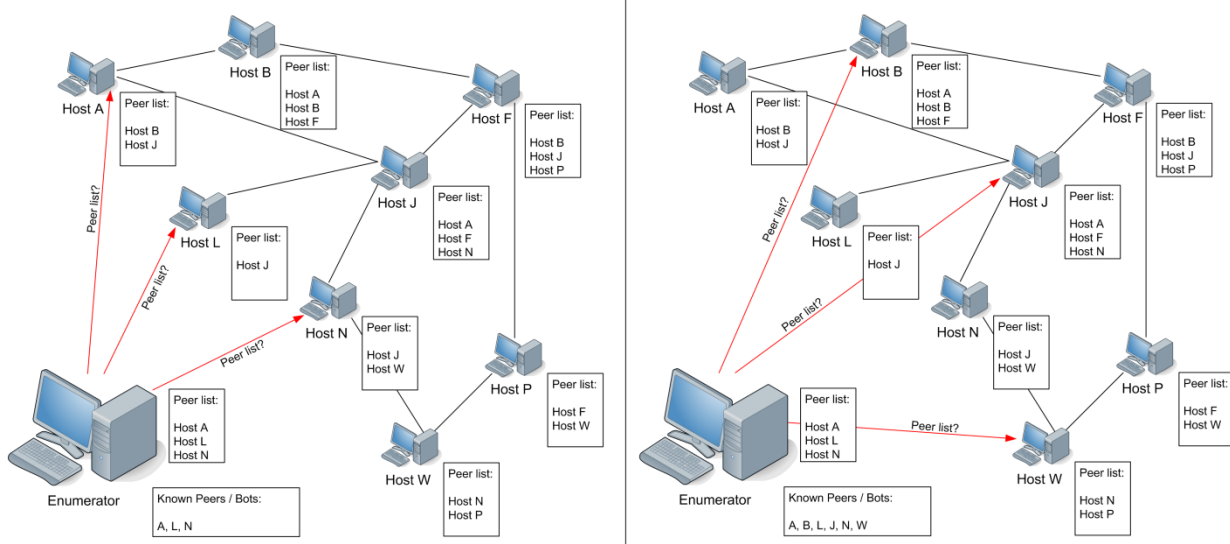


Figure 10: Peer-to-peer botnet enumeration.

Even though the information about the botnet is not available at a central point, the structure of the network can be exploited for measurement purposes. By repeatedly querying peers for their neighbour lists it is possible to enumerate the botnet recursively. The result will ideally be an exhaustive list of active nodes participating in the peer-to-peer network. Of course it is necessary to be able to participate in the botnet in the first place. This is usually achieved by reverse-engineering the communication protocol and creating an implementation of the bot in order to perform the enumeration task.

Example: Holz et al. [106], [107] demonstrated approaches to the systematic infiltration and measurement of a peer-to-peer based botnet by enumerating all peers through the use of crawling. In both cases, their approach depended on a reconstruction of the communication protocol [20], [108], which involved exploitation of design flaws in the cryptography used; participation was achieved by the construction of a drone that was able to communicate over this protocol. This tool enabled them to perform an in-depth analysis of the botnets and also to develop methods that could be used as countermeasures.

Example: Kanich et al. [109], [110] also created a crawler for the Storm botnet, named Stormdrain. Their work focused on dissecting real and active bots from other peers participating in the Overnet peer-to-peer network. By studying the protocol and algorithms inside the bot, various features that were used for checking peers for their authenticity could be identified. They were able to show that the efforts of other parties working inside the Storm botnet were visible and detectable. As an example, Kanich et al. were able to observe the insertion of about 5000 new nodes into the botnet for two hours. These bots did not show the native bot's characteristics and were likely to have been crafted by researchers.

Example: Dittrich and Dietrich [111] give an overview of recent techniques used for discovering botnets. They emphasised the accuracy and stealthiness of various approaches which concentrate on the botnet's quantitative properties, such as footprint and live population. Their work provided a case study containing a long-term enumeration of the Nugache peer-to-peer botnet, with which they could underscore the differences between different approaches.

3.3 OTHER

In this section, two more techniques are explained. Both are auxiliary techniques that support the measurement and detection of botnets.

3.3.1 MALWARE REVERSE ENGINEERING

Reverse engineering generally refers to a technique that aims to recover the functionality and "innards" of a compiled program, without knowledge of its source code. In the context of malware, it can be used to extract information about installation, spreading, communication mechanisms and damage potential. From the information gained, possible countermeasures can be developed, like detection signatures for network traffic based on characteristic byte patterns generated by, or included in, the malware. Reverse engineering additionally allows the classification of malware into families, and in rare cases even the support of investigations into the authors of certain types of malware.

The malware reverse-engineering process can be divided into static and dynamic analyses. Static analysis is performed without actual execution of the binary. In general, the reconstruction of particular functional aspects, or even of the binary's whole architecture, can be the target of this approach. In dynamic analysis, the sample is usually executed in a controlled environment. Monitoring the host system can reveal the malware's behaviour, e.g. changes to the file system, registry and kernel, but also activities like searching for credentials, key logging or network usage/connection attempts.

Malware reverse engineering is not in itself a measurement technique, but it enables possible working methods to be identified and derived from the results obtained.

Examples are current lists of other bots of the botnet, statistics about successful spreading, or command-and-control servers. Also, new botnets can occasionally be identified from analysis of the corresponding samples.

3.3.2 C&C FORENSICS AND ABUSE DESKS

‘C&C Forensics’ applies forensic techniques to command-and-control servers by treating them as a crime scene. This method can be used once control of the targeted server has been gained. Subgroups of analysis can be divided into live analysis, which is performed directly on the running system, and analysis of the extracted hard discs, disc images or virtual machine images if the server has been powered off.

The measurement aspect of this method can be compared to the infiltration of a botnet (see section 3.3.6). The major difference is the mainly retrospective character of this group of techniques, as it focuses on the evaluation of any log files, databases, and logged keystroke files that can be found on the server.

Abuse Desks of Internet Service Providers, for example, are a central source of information about malicious activities, with the scope on the network maintained by the ISP. This is not only limited to malware activities and botnet command-and-control traffic but also includes spam emails, phishing websites or directed protocol attacks based on various communication protocols, such as Secure Shell (SSH). Abuse desks serve primarily as a tool for detecting internal and outbound attacks that originate from the ISP’s network itself. This is because the abuse desk’s main information sources are reports they receive about activities detected by third parties. Further actions can be initiated by the abuse desk. For example notification of customers about infections, which is relevant for the mitigation concept presented in sections 4.1.8 (Walled gardens) and 4.2.3 (Central Incident HelpDesks). Other actions include the start of criminal investigations, which may potentially lead to the seizure of hosted computers and forensics on C&C mechanisms as described above.

3.4 ANALYSIS

After introducing different detection and measurement approaches, this section compares the major characteristics and advantages of the approaches. The goal is to outline the strengths and weaknesses of the techniques in the context of the features defined in section 2.2.

First of all, it is necessary to explain that the threat potential of a botnet depends on the view of the stakeholder it is assessed for. A botnet’s size is only one contributing aspect of its total threat potential and works like a scaling factor. Even a small botnet can inflict severe damage, depending on the individual compromised machines, the infrastructure in

Size is not the only indicator of a botnet’s damage potential. Capabilities, robustness and flexibility are also critical.

which they are embedded and the functionality and quality of implementation included in the bot software. However, the size of a botnet is an indicator of the secondary costs that arise through the effort of cleaning the infected machines.

Further metrics for the threat assessment of a botnet include, for example, the amount and quality of spam email, bandwidth use during DDoS attacks, or the aggressiveness with which credentials and other sensitive information are extracted from the victim computer. Regarding

the botnet itself, the robustness of command-and-control infrastructure, the resulting resilience against countermeasures and the malware's stealth capabilities, are also important factors contributing to a botnet's threat level. Offensive features, e.g. propagation mechanisms and exploits used, also play a role in these considerations.

Independently of the metrics and concrete measurement method applied, it is important to describe every detail of the measurement procedure in order to support and justify the significance of the results obtained. However, this is beyond the scope of this section, whose focus is on approaches for the detection and measurement of botnets. Metrics for the assessment of a botnet's threat potential are discussed in more detail in section 1.2.3 (Attack Potential and Threat Characterisation).

The evaluation will rate the described approaches in the context of the usability of their results. The techniques are therefore interpreted as an input vector for decisions on how to engage the botnet threat. But the monitoring of botnets plays an important part in obtaining an impression of their activities and infrastructures. This is relevant to the preparation of countermeasures. The more information is available, the better the chance of countering the botnet effectively.

3.4.1 GENERAL CHARACTERISTICS

Regarding the general characteristics of a detection and measurement approach, important features are its flexibility to adapt to changes, and generality in terms of botnet types covered.

Honeypots, packet inspection, and the analysis of flow records work independently of a specific botnet infrastructure. All of these approaches are suitable for recognising botnet spreading attempts, typical botnet communication, and the actual effects of malicious functionality. This makes them universal and flexible in their application. The same attributes can be assigned to the submission of samples of malware specimens to dedicated analysis services. The parallel application of several anti-malware products to the same sample can give an initial impression of how

Even a small botnet can inflict severe damage, depending on the individual compromised machines

Passive detection approaches like honeypots, packet inspection and analysis of flows provide good flexibility.

common it is. Running a sample in a sandbox may also result in an overview of the behaviour and contacted domains and IP addresses. This approach is limited only by anti-debugging attempts integrated into the malware. Depending on the malware's level of sophistication, the suspected sample may exhibit different behaviour in a virtualised analysis environment, which most sandboxes are based upon. The reaction time of all of these approaches depends on the detection of new trends in malware in general, because the approaches have to be adapted to these trends. Some of the existing solutions for the above tasks are quite mature and, depending on their configuration, capable of detecting various types of botnet.

The technique described as C&C forensics can yield excellent results but depends heavily on the individual case. The success of this approach usually depends on log files and databases extracted from the infiltrated command-and-control server. In the best case scenario, a history of the botnet can be reconstructed. This includes all stolen data, such as victims' documents and credentials, as well as information about the botnet's size and functionality, if these were maintained by the botmaster. Although a general procedure for the

analysis of captured servers is hard to define, the approach offers moderate flexibility. Similarities between cases can often be exploited if two botnets originate from the same family if, for example, they were generated through a crimeware construction kit. In an ideal case, the analysis will even reveal the botmaster himself, if he has logged into the server using an IP address that can be tracked back to his location.

The group of DNS-based approaches, namely anomaly detection, cache snooping and DNS-based sinkholing, obviously depend on the domain name system being part of botnet infrastructure and communication. This limitation is tolerable because almost all botnets use DNS in some way. Due to the fact that DNS is a fundamental concept of the Internet, the

techniques are mostly independent from botnet infrastructure and so offer a high degree of flexibility. Sinkholing can be easily applied if the DNS registrar in charge of the malicious domain is cooperative. DNS cache snooping can be applied, even without the cooperation of DNS registrars, as this approach can be considered a form of public sources data mining. To gain access to this data, DNS anomaly detection and correlation-based approaches usually need some time for development or adjustment, as well as the cooperation of DNS operators.

In the best case scenario, a history of the botnet can be reconstructed. This includes all stolen data, such as victims' documents and credentials, as well as information about the botnet's size and functionality

Specialized approaches targeting DNS, IRC, or P2P often provide better results than generalized approaches, but lack flexibility.

Spam-based methods and the evaluation of application log files are limited in their generality as they are based on effects caused by botnets. Without the presence of these features, the methods will not produce any results. On the other hand, both approaches are able to measure these effects independently of the concrete botnet, thus giving them good flexibility. Additionally, because these methods operate on data already collected, their application is not visible to botmasters.

To be applicable, the remaining active approaches related to infiltration, IRC monitoring and peer-to-peer enumeration, require certain command-and-control mechanisms to be present in the botnet. Inside their classes they offer flexibility, because they are tailored specifically to them. In the case of IRC monitoring, examples of fully automated generation of botnet-specific monitoring drones have been presented and applied practically. This is possible because the necessary parameters can be directly extracted from network traces and binaries. Peer-to-peer enumeration is less flexible and automatable because the routing mechanism for a specific botnet has to be understood and, if possible, adapted for measurement purposes. The benefits of these specialised approaches, which target certain botnet properties, are better insights into the botnet infrastructure itself.

Anti-virus feedback and sensor networks depend on the product itself, as well as the customer base or, more specifically, on how it is distributed and in which sectors (government, businesses, end-users, etc) the software has been deployed. Because Anti-virus software has to be updated regularly, emerging threats are usually detected only when the relevant detection signatures have been delivered.

It is important to mention that most of the passive approaches are not in any way visible to the botmaster, because they are based on silently duplicated data. This is a significant advantage when supporting ongoing investigations because no suspicion is aroused on the side of botherders.

Most passive detection and measurement approaches are invisible to botmasters.

As for the active techniques, they expose different degrees of visibility as they interact with the botnet one way or another. In the case of DNS cache snooping, it may be detected that the queries leave traces at all monitored DNS servers in the form of cache entries. The visibility of fast-flux tracking again depends on the queried DNS servers. If these are maintained by botherders, they might recognise a pattern in the source IP addresses used for queries. Generally speaking, enumeration and IRC monitoring can attract attention if the implemented agent or drone behaviour differ from a native bot's behaviour, or if malicious commands are not carried out as desired by the botmaster. The sniffing of incoming data on a mirror port of a command-and-control server is silent. By contrast, sinkholing resulting from a shutdown/deregistration of a domain has maximum impact on, and visibility to, the botherder.

3.4.2 QUALITY OF RESULTS

When evaluating the quality of the results achieved with the approaches presented, it is again important to emphasise that size alone, in terms of compromised computers, is not sufficient to characterise the severity of the botnet threat. Nevertheless, a botnet's population is considered to be the major threat indicator. In many publications, only a total number for botnet size is given or cited, without statements about how this number was obtained. Both the detection and measurement aspects and the concrete methodology applied play a significant role in determining the output values. In particular, the methodology used in evaluating the significance of the measurement results should be justified on solid scientific basis. To obtain a full picture of the malware and botnet, it is important to factor in different approaches that complement each other.

Measurement methodologies have to be documented transparently and results have to be evaluated on a solid scientific basis.

Reverse engineering is a favoured approach for evaluating the potential damage a malware specimen can cause. Significant information about its functionality, propagation mechanisms and communication structure may be revealed by in-depth analysis of malware samples. Although it is an auxiliary technique to general detection and measurement, reverse engineering is especially important in the context of botnet research. At this point it must be emphasised that knowledge about the functionality of malware does not automatically assess the threat posed by the related botnet. For example, a botnet with DDoS functionality may target small or big companies with practically the same result, e.g. the inaccessibility of websites or services, but it still produces a different financial impact because bigger companies are likely to have higher business volumes.

Reverse Engineering is a favoured approach for estimating a malware specimen's damage potential.

Both sinkholing and C&C forensics have the potential to offer very reliable information about botnet size. The reason for this is that these techniques allow insight into the botnet and a view of the network from the botnet's perspective. While sinkholing involves active impersonation of the botnet in terms of live incoming connections, forensics are the post-mortem counterpart for the botnet's administrative level.

As mentioned earlier, sinkholing provides live information about attempts to connect to compromised computers and can therefore provide a snapshot of the current botnet population. This can range from a few connections to hundreds of thousands of connection attempts per hour. However, as stated above, counting IP addresses alone is a complicated measure, because various effects distort the association of unique IP addresses and distinct systems.

In a report about the temporary takeover of the Torpig botnet [50], different measurement methods could be applied for 10 days and compared:

- 1,247,642 unique IP addresses.
- The number of observed IP addresses increased almost linearly over time.
- 182,800 hosts estimated by unique identifiers.
- 75% of unique identifiers were detected during the first 2 days.

Sinkholing of botnet traffic provides a view of the botnet's live population.

Taking the unique identifiers as a reference, the IP addresses yielded an overestimate factor of 6.8. This, and the fact that the bot's IP addresses varied over the time, leading to a constant increase in observed addresses, are an indicator of the unreliability of measuring IP addresses only.

Trusting size estimates based on such identifiers can also be open to question questionable, in case the generation algorithm is flawed or has been manipulated by the botmaster. For example, in IRC botnets, bots use nicknames as identifiers. Depending on the botnet, these nicknames are often generated every time the computer is

Taking observed unique IP addresses as the only indicator of a botnet's size usually leads to drastic inaccuracy and often to overestimations.

started and are therefore no more reliable than unique IP addresses. In the case of HTTP-based botnets, the header information can sometimes be used to increase accuracy, for example, by evaluating the User-Agent field. If the botnet uses encrypted communication, and no such unique identifier is available, unique IP addresses remain one of the only ways to derive any information about a botnet's population from the incoming connections. Furthermore, where unique IP addresses are considered, it is important to limit the counting to certain time-frames. For example, many broadband providers assign IP addresses dynamically to their customers with 24 hours validity. The results from measurements in these time-frames, may have more significance than those over longer periods,. It has to be pointed out that measuring in this way provides a view of the live population of the botnet and not the overall number of infected hosts.

C&C forensics is a post-mortem approach which can give a retrospective view of botnet operation processes. In ideal cases, this approach may reveal full databases and repositories used for botnet management, which contain a lot of useful information and provide a high-definition picture of the botnet. This technique may yield the best results in terms of completeness, scope and accuracy, as original data from the

botmaster is used. As an aside, it should be noted that the application of both approaches is accompanied by the temporary or permanent shutdown of the botnet.

With regard to active infiltration approaches, like peer-to-peer based enumeration and IRC monitoring, these methods may yield comparable results to the above approaches. Data is gathered directly from inside the botnet but from the perspective of a bot, and so has limited scope when considering the whole command infrastructure. One advantage of these approaches, when compared to the presented passive methods, is that they monitor data originating directly from the botnet, without the need to first classify malicious and benign information. Furthermore, the accuracy of these techniques depends heavily on the features used for measurement. Counting unique IP addresses over long periods of time can be considered meaningless, as already described above. The benefit of monitoring a botnet from the inside is that activities, as well as updates, can be detected very quickly. This enables changes in the botnet architecture and functionality to be monitored. By applying these approaches over time, a history of the botnet can be created remotely. Other specialised approaches are more prone to failure when changes to the botnet occur. For example, slight modifications in the protocols used can stop enumeration approaches from working.

The tracking of fast-flux networks is useful for an indication of live bots connected to a botnet. However, any measurements regarding the size of the underlying botnet have little or no meaning. Typically, only a small percentage of suitable botnet hosts is used in the fast-flux service network. This was shown, for example by research on the Storm fast-flux network, where only about 1% of all bots were used in fast-flux operations [103]. However, tracking fast-flux networks can be especially useful as input for further mitigation techniques, e.g. blacklisting.

Active enumeration approaches can provide acceptable accuracy regarding a botnet's live population.

DNS cache snooping is useful for a rough overview of how widespread certain malware is; more specifically, of how certain malicious domains across the DNS partitioning of the Internet are queried. As explained in the section about this approach, it can only be performed by one party at a time, because the DNS servers' caches are full when they are accessed for the first time during the lifetime of a cached entry. Multiple parties performing cache snooping in parallel will receive highly inaccurate results. However, no reliable data about a botnet's size or function can be derived from this approach.

The group of passive approaches, including passive honeypots, packet inspection, the analysis of flow records and spam evaluation, depend heavily on their input sources. Hence, the accuracy of measurements is generally difficult to estimate, because these techniques can only evaluate what they actually recognise within their network scope. Packet inspection, the analysis of flow records and DNS-based analysis have to treat

massive amounts of legitimate traffic in order to identify malicious traffic. Therefore, considering their scope, these approaches work inefficiently. The analysis of application or service log files may also lack efficiency because their scope is so general. The analysis of spam is already filtered in advance when being analysed and has a narrowed scope. The same applies to passive honeypots, which are usually operated on network addresses that do not serve production purposes and which are contacted solely by suspicious entities.

In general, if passive approaches are not deployed globally, only a fraction of a whole botnet can be seen [91]. This is especially true for distributed and decentralised botnet approaches, e.g. peer-to-peer based botnets. Additionally, all passive approaches aim at generality in their capabilities and therefore work well when applied as pure detection methods. This makes them good auxiliary techniques when applied in combination with other detection, measurement, and mitigation approaches, e.g. walled gardens and blacklisting. On the other hand, the generality and lack of scope limit accuracy when single botnets are evaluated.

The use of the passive approaches presented as part of early warning systems (EWS) and threat monitors is realistic. Early warning systems usually consist of a variety of different sensor types, whose output is merged using the principles of multi-sensor data fusion. The intention is to derive a higher information value that is easily understandable by humans from the combined results of the sensors. Honeypots and IDS in combination can contribute greatly to EWS, as their main intention is the detection of a variety of threats as well as unknown attack techniques.

passive detection approaches can serve as combined sensors in an early warning system.

3.4.3 REQUIRED EFFORT AND RESOURCES

The question of the effort needed is always connected to the question of the scale on which an approach is implemented. This is especially important for the passive approaches, as this subgroup can only process data that is captured within its network scope. To achieve accurate and exhaustive measurements of botnets, data from the whole Internet, or at least data from the highest tier carriers, would be required.

Centralised detection approaches aimed at generality often face challenges with scalability.

No known centralised, deep-packet inspection approach can handle the real-time analysis of all packet contents on an ISP level. Therefore scalability becomes important, especially for centralised approaches, whose aim is to monitor as generally as possible. Alternatively, in the case of sinkholes, completeness is actually achieved via a centralised approach, because only the desired botnet traffic is routed towards

the sinkhole. Potentially, all existing bots of the botnet are covered by this. In the case of the other monitoring approaches presented, e.g. analysis of flow records or application log files, as well as spam-based methods, while these have better scalability, they are always limited to the area in which they are deployed and may still miss relevant events. Decentralising the detection by using users' machines as sensors, as used in approaches such as evaluating the feedback from anti-virus solutions installed on customers' machines, produces a broad picture. For example, using Microsoft's Malicious Software Removal Tool [94], data from more than 600 million hosts is remotely gathered and evaluated. The approach of filtering data on customers' machines also achieves a massive reduction in the effort required, but creates a dependency on the distributed software itself.

Regarding purely technical effort, most of the presented approaches have moderate requirements. In general, special hardware developed for botnet detection is not necessary. Significant costs may however result from hardware that is capable of providing high capacity with regard to the needs of network traffic handling and computational resources; for example, massive numbers of packets, flow records, or spam mails need to be processed. This is the case especially when results are required in short intervals or even real time. This underlines the problem with these approaches. Identifying botnet traffic among benign, regular traffic is like searching for a needle in 100 million haystacks. For most of the other methods, standard computer hardware suffices. In some cases, it is sufficient to parallelise the processing of data. For example, suspicious files can be examined in parallel by several analysis tools, like anti-virus software and sandboxes. In addition, most of the methods presented work with aggregated data and use sophisticated filtering mechanisms to increase their efficiency. For active approaches that interact with the botnet, it may be necessary to have the ability to change the IP address from where these methods are operated. This is important if operations are discovered by the botmaster, resulting in the blocking of IP addresses or retaliation in the form of a DDoS attack.

Identifying botnet traffic among benign, regular traffic is like searching for a needle in 100 million haystacks

Common to almost all approaches is that they require a deep understanding of system and network details, and these need to be developed and improved. The reason for this is that the methods often exploit the special properties of communication protocols and system architecture, which have to be identified by intense analysis. Therefore, experience and expert knowledge are key requirements for all measurement and detection approaches.

Furthermore, the interpretation of data is often not a trivial task. It is important to note that malware developers constantly adapt to analysts' approaches. This results in an arms race that leads repeatedly to special cases, where all standard approaches fail.

Therefore, frequent adjustments of, and extensions to the existing methods are necessary. For example, honeypots have to be updated with modules for the simulation of recent vulnerabilities. For packet inspection, signatures of current attack vectors have to be integrated and kept up to date.

3.4.4 LIMITATIONS

The most important limitations in botnet detection and measurement relate to legal aspects. In general, approaches are constrained by the conflict resulting from user privacy and personal data protection laws and laws whose aim is to ensure and improve overall Internet security, including aspects like the general availability of Internet connectivity, or ensuring the operation of critical infrastructure. As an example of this, the impact of privacy laws on the applicability of some of the presented approaches is illustrated below.

Some European countries like Switzerland and Sweden treat IP addresses as personal data; others like Ireland do not.

Some European countries like Switzerland [112] and Sweden [113] treat IP addresses as personal data; others like Ireland [115] do not. The main concern in this area is whether or not Internet users can be identified through the combination of a timestamp and an IP address. This is one example of the difficulties that arise from differences in legislature at a national level, complicating the application of several methods in a unified European, or even global, context. For example, the analysis of flow records depends heavily on IP addresses as a feature for processing. The application of deep-packet inspection is even more complicated as an analysis of flows. EU law, EU Directive 2002/58/EC [116] prohibits the

“(22) [...] storage of communications and the related traffic data by persons other than the users or without their consent is not intended to prohibit any automatic, intermediate and transient storage of this information in so far as this takes place for the sole purpose of carrying out the transmission in the electronic communications network and provided that the information is not stored for any period longer than is necessary for the transmission and for traffic management purposes, and that during the period of storage the confidentiality remains guaranteed.”

The processing of personal data is also covered by EU Directive 2002/58/EC [116]:

“(26) The data relating to subscribers processed within electronic communications networks to establish connections and to transmit information contain information on the private life of natural persons and concern the right to respect for their correspondence or concern the legitimate interests of legal persons. Such data may only be stored to the extent that is necessary for the provision of the service for the purpose of billing and for interconnection payments, and for a limited time. Any further processing of such data which the provider of the publicly available electronic communications services may want to perform, for the marketing of electronic communications services or for the provision of value added services, may only be allowed if the subscriber has agreed to this on the basis of accurate and full information given by the provider of the publicly available electronic communications services about the types of further processing it intends to

perform and about the subscriber's right not to give or to withdraw his/her consent to such processing. Traffic data used for marketing communications services or for the provision of value added services should also be erased or made anonymous after the provision of the service. Service providers should always keep subscribers informed of the types of data they are processing and the purposes and duration for which this is done."

The extent of these regulations is framed by Article 15, which introduces some freedom regarding the interpretation of confidentiality and traffic data:

"1. Member States may adopt legislative measures to restrict the scope of the rights and obligations provided for in Article 5, Article 6, Article 8(1), (2), (3) and (4), and Article 9 of this Directive when such restriction constitutes a necessary, appropriate and proportionate measure within a democratic society to safeguard national security (i.e. State security), defence, public security, and the prevention, investigation, detection and prosecution of criminal offences or of unauthorised use of the electronic communication system, as referred to in Article 13(1) of Directive 95/46/EC. To this end, Member States may, inter alia, adopt legislative measures providing for the retention of data for a limited period justified on the grounds laid down in this paragraph. All the measures referred to in this paragraph shall be in accordance with the general principles of Community law, including those referred to in Article 6(1) and (2) of the Treaty on European Union."

Another relevant document in this context is Directive 2006/24/EC [117], targeting the retention of connection data in order to fight organised crime. Article 5, (2) (iii) says that

"(iii) the name and address of the subscriber or registered user to whom an Internet Protocol (IP) address, user ID or telephone number was allocated at the time of the communication;"

are part of the categories to which data retention applies. Retention periods are defined in Article 6:

"Member States shall ensure that the categories of data specified in Article 5 are retained for periods of not less than six months and not more than two years from the date of the communication."

After all, as defined in Article 4, access to retained data shall only be provided to

"[...] competent national authorities in specific cases and in accordance with national law."

In the context of these directives, the issue of whether, for example, packet inspection for the sake of botnet detection and measurement, can or cannot be applied at an ISP level is a complex one and, at this point, exceeds the scope of this report.

Another aspect of the legal issues relates to reverse engineering. In 2008, versions of ZeuS malware were found which contained a licence agreement [118], formally prohibiting the disassembling or analysing of a bot. From a legal point of view, reverse-engineering this malware would violate the copyright of the malware author. Related to this, the Shadowserver [37] has blogged about the legal implications of tracking IRC-based botnets. They came to the conclusion that accessing command-and-control servers is legal, as long as the server is unprotected and publicly available over the Internet. The extraction of passwords for the relevant IRC C&C servers locally, through means such as traffic interception or sandboxing, is also acceptable from their

viewpoint and interpretation of the law, because the operator of the computer with the malware running on it allows traffic interception [119].

For more information on legal issues in the context of botnets, see [2].

Legal limitations apart, some of the presented approaches can be applied only by certain stakeholders or through close cooperation between different stakeholders. For example, for effective and large-scale spam e-mail analysis, access is required to data that is only available to providers of e-mail services, because they can collect massive amounts of spam easily via their internal filter systems. The detection of anomalies in query patterns of domain name queries needs access to the log files or live data of DNS servers themselves. For an effective application, this method would have to be applied by a large number of DNS server operators. On the one hand, the willingness of parties to generally participate in such approaches is very important; on the other, the level of trust between involved parties is also seen by many experts as a crucial factor in the success of botnet mitigation. More on initiatives and cooperation in the botnet context can be found in sections 4.2.4 (Enhance Cooperation between Stakeholders) and 4.3 (Initiatives and Institutions).

Sinkholing and C&C forensics are special cases, because their application is connected with a botnet takedown. This again requires action, usually involving law enforcement and service providers. If, according to their terms and conditions, a service provider detects abuse, they may also take action on their own.

In some cases, the required access may not be obtained at all, due to non-cooperative parties. For example, so-called bullet-proof hosting providers grant their customers leniency in their activities. These services are regularly used by criminals for spamming, as malware distribution hosts or command-and-control servers [120], and they are often located in countries with less restrictive laws, which makes non-cooperative behaviour easier to get away with.

Some of the approaches presented are also affected by their topographic location within the IP address space. Honeypots will encounter different kinds and amounts of traffic, depending on which IP address they are deployed. One reason for this is that botmasters and malware developers use their own blacklists and mechanisms to track and avoid these analysis systems [121]. This may also have an impact on spam-based methods, as e-mail providers might be targeted differently by spam.

3.5 CONCLUSION

In this chapter, various methods of botnet detection and measurement have been discussed. The approaches have been divided into passive and active. Additionally, two further methods that do not fit into these categories have been described. This is followed by an

Only a few current methods offer acceptable measurement accuracy regarding the actual size of botnets.

analysis of the approaches, based on the feature set defined in section 2.2.

It should be noted that the measurement of botnets is very complicated and is influenced by many different factors. Only a few current methods, or those that rely on the special properties of certain botnets, offer acceptable levels of accuracy regarding the actual size of botnets. Because of these findings it is proposed that the threat of botnets is estimated in a broader context, for example by including the severity of damage, both potential and actual, instead of only basing their size on the number of compromised hosts (see section 1.2.3).

The wide availability of broadband access provides even small botnets, consisting of a few hundred to a thousand bots, with the ability to launch significant DDoS attacks [122]. Small, or old and comparably unsophisticated botnets are also able to cause significant damage if they target, for example, critical systems or consist of a few highly connected hosts or those containing information that deserves special protection. These different aspects illustrate that the metrics applied to assessing a botnet threat potential vary considerably and have to be viewed from a specific stakeholder's perspective. Proposals for such metrics are given in section 1.2.3 (Attack Potential and Threat Characterisation). For example, the potential damage of a malware specimen can be estimated through detailed analysis using means such as reverse engineering in combination with the tracking of its activity, giving rise to the practical implications of making it a primary target of further mitigation efforts.

A key characteristic of the group of passive approaches is that they do not interfere with botnets directly and derive their results from observations only. Their major advantage is that they can detect and cover various, and even unknown, types of botnets. In addition, because of their detection character, these methods aim for generality in how they function and use universal features, creating, as a prerequisite, the need to differentiate between benign and malicious traffic or behaviour. The active and more invasive approaches presented specialise mostly in certain instances of botnets, or in a class of botnets in terms of the technology or infrastructure used, e.g. a specific protocol used for command-and-control tasks. Their functionality and scope are tailored to these subgroups, on the one hand offering these approaches the ability to have a stronger focus while, on the other, sacrificing some generality and flexibility.

The results of the analysis from the detection and measurement approaches are summarised in Table 1. This overview table should be interpreted with care, as the colours used indicate only rough trends resulting from the comparisons made between approaches according to the feature set. Because the features themselves have a different weighting in an overall rating and even have a different weighting for distinct approaches, comparisons across columns should be done carefully.

	Packet Inspection	Analysis of Flow Records	DNS-based approaches	Analysis of Spam Records	Analysis of (Application) Log Files	Evaluation of Anti-Virus Software	Honeypots	IRC-based measurement and detection	Enumeration of Peer-to-Peer Networks	Malware Reverse Engineering	Tracking of Fast-Flux networks	DNS Cache Snooping	Sinkholing	Infiltration	C&C Forensics
Generality															
Flexibility															
Reaction time															
Stealthiness															
Completeness															
Accuracy and uncertainty measures															
Scope of considered Data and information															
Timeliness of measured data															
Technical robustness															
Possible interferences															
Technical effort															
Development effort															
Administrative effort															
Required expert knowledge															
Cost factors															
Restricting laws															
Dependency on third parties															
Lack of access to data, systems or infrastructure															
Topographic restrictions															

Interpretation	
	factor with strong negative
	notable negative influence
	average influence / not measurable
	notable positive influence
	factor with strong positive

Table 1: Evaluation of detection and measurement approaches.

4 COUNTERMEASURES AND INITIATIVES

In this chapter, various countermeasures to the botnet threat are presented. They are organised into two sections: technical methods; social and regulatory approaches. Additionally, a short overview is given of existing initiatives and institutions that are active in the mitigation of botnets.

4.1 TECHNICAL COUNTERMEASURES

The countermeasures presented in this section apply at a technical level. Most of them focus on the command-and-control infrastructure of botnets by, for example, filtering botnet-related traffic, sinkholing domains with the assistance of DNS registrars or obtaining the shutdown of malicious servers in data centres.

An important consideration in this context is that all approaches related to the takedown of the command-and-control infrastructure do not disinfect the compromised machines actually participating in the botnet. If they do, the infections that remain can cause severe security issues, as many malware variants contain routines that disable protection mechanisms like anti-virus software or desktop firewalls, or inflict side effects that influence the stability of the host system. Furthermore, if the bots have previously received commands that require another command to stop their actions, instead of a scheduled interval, the possibility exists that this activity may continue until other measures are taken to interrupt it, e.g. rebooting of the victimised computer system. As an example, a takedown at the wrong moment may cause a DDoS attack to continue indefinitely.

The techniques are ranked (from low to high) by the perceived legal complications that may arise when the techniques are applied.

4.1.1 BLACKLISTING

Blacklisting itself is not a direct countermeasure against botnets. Instead, it should be perceived as a supporting process which provides input for further technical means of resistance. The contents of a blacklist are multifaceted and are used by a variety of parties from different sectors. A blacklist may, for example, provide single IP addresses of malicious hosts or whole subnets showing suspicious activities. One use of blacklists is to block all traffic from included addresses. Another application of a blacklist consists of a collection of URLs which can be used by search engines, or within a browser, to filter or mark websites with suspicious or proven malicious contents.

The distribution of a blacklist is usually organised actively with special interfaces and processes. For instance, it can be done via real-time ‘push services’ between ISPs, or

passively in the form of services open to subscription, such as RSS feeds. The information provided by blacklists is used intensively by web browsers to protect users from phishing or infection with malware. Other groups using blacklists are ISPs, email providers, CERTs, and Anti-Virus companies. Furthermore, these corporations typically maintain and distribute such blacklists publicly. In addition, various kinds of honeypots (cp. section 3.2.1) can be used to collect data about infected hosts, creating information that can be introduced into blacklists.

More recent approaches deal with real-time blacklists containing IP addresses that have been identified as infected hosts. This information may then be used, for example by financial institutions, to protect users from fraud by blocking these hosts or denying corresponding logins and transactions if their credentials are likely to be in jeopardy.

Example: Zhang et al. [123] presented the concept of Highly Predictive Blacklists (HPB), a system for dynamically generating blacklists by combining and correlating contributions of DShield.org [124] users. HPBs use a 'link analysis' method similar to Google's PageRank [125], extended with input filtering and a severity metric to rate attack patterns. For evaluation purposes, more than 720 million log entries from DShield contributors, submitted during October to November 2007, were used. Zhang et al. showed that HPB achieve about 20-30% more hit rates (i.e. successful access to a cache entry in relation to all access attempts) compared to the other blacklists offered by DShield.org. This also resulted in improved prediction of new attacks and long-term performance stability.

Example: The Spamhaus Project [126] is an international, non-profit organisation, which aims at the mitigation of spam and other malicious activities. The organisation maintains various real-time lists that assist in identifying and blocking attempts by malicious activities. Examples of lists include: the Spamhaus Block List (SBL), containing a collection of IP addresses from which incoming e-mail should not be accepted; or the Domain Block List (DBL), containing domain names extracted from spam e-mails that can be used for the detection of spam by searching in emails for the presence of such domains.

Example: Abuse.ch [47], [48] maintains a tracking service targeting the command-and-control servers of both the Zeus (Zbot) and SpyEye banking Trojan. Their system is based on the automated analysis of Zeus binaries, which are provided by anti-virus companies, helping to obtain a low false-positive rate. Zeus Tracker lists all identified malicious hosts on a central blacklist that is also distributed as a real-time RSS feed.

4.1.2 DISTRIBUTION OF FAKE/TRACEABLE CREDENTIALS

The distribution of fake credentials is not only a purely technical countermeasure but also targets the botnet's profitability by attacking the underlying business model.

A common botnet application is identity theft. Profit is created by stealing credentials or credit card records. By monitoring a botnet, analysing the functionality of bots and

identifying the target websites where stolen information is submitted, so-called dropzones, an approach to countering this behaviour is to craft false, target-oriented information that can be injected into these dropzones. It is assumed that the submission of fake credentials will generate mistrust between criminals and lower the quality of their stolen information, hence their trade goods. The botmaster's customer might complain about the low quality of the collected data, as will the botmaster about the toolkit used. In other words, fake credentials are used to contaminate the data collected by botnets in order to minimise the profitability of the entire botnet.

In addition to that, these custom-defined credentials can also be used to track involved parties, and where and when they are used. Although this procedure requires a lot of effort across stakeholder groups, it can help to strengthen the cooperation between, for example, financial institutions and law enforcement agencies, that is needed for a successful application. In contrast to simply injecting invalid information, this demands the provision of specially prepared credentials. These credentials might, for instance, include bank account or credit card information, pointing to a monitored bank account [127]. Observation of the account movements can lead to the individuals involved being identified and prosecuted. As such transactions are happening typically at a cross-border level, strong international cooperation is a precondition for the whole process to succeed. Many banks are not willing to cooperate, because even fake accounts have to be filled with real money for transactions to take place.

Example: Ormerod et al. [128] presented a bottom-up approach for mitigating botnets, which aims at the profitability perspective of both the developers of botnet toolkits, like ZeuS and SpyEye, and their respective clients, the botherders. They assume that a cascading effect can be established between involved parties if “chaff” is introduced into a botnet. This could be huge amounts of random data, intended to disrupt databases, or crafted fake (but traceable) credentials aimed at identifying criminal individuals who are abusing them for financial transactions. As a prerequisite, the internal communication structure and submission mechanism of the botnet has to be studied by either reverse-engineering the botnet samples, or concentrating behaviour analysis on file system and network activities.

4.1.3 BGP BLACKHOLING

The Border Gateway Protocol (BGP) is widely used throughout the Internet and has consequently become the predominant technology for decisions on the routing of network traffic. BGP is used to maintain the Internet routing table, which is organised in chunks of IP addresses, and ensures the accessibility of so-called autonomous systems (AS), for instance ISP networks. This table includes information about paths between ASs, and also the shortest paths.

Data provided by the aforementioned blacklisting techniques can be used to change routing policies and “nullroute” malicious hosts to deny traffic from or to them and their networks. Nullrouting describes the process of silently dropping packets

originating from, or destined for, such addresses. Typically, these decisions are made at the ISP level and affect or protect hosts served by a particular ISP. A more comprehensive approach challenges cross-AS and cross-border cooperation, but so far has hardly been implemented.

Blackholing is best suited to botnets with static C&C infrastructure. This allows selective blackholing of the addresses of these botnets' command-and-control servers, preventing bots from communicating with the botmaster's control mechanisms. The approach is comparable to DNS sinkholing, because both protocols are situated on the application layer.

Example: In the last few years, attempts have been made to shut down Internet Service Providers known for hosting malicious software and command-and-control servers. For example, the de-peering of Atrivo/Intercage (AS 27595) in September 2008 [129] resulted in a temporary shutdown of the Ozdok/Mega-D C&C servers, affecting up to 150.000 infected computers and reducing global spam for a short period. However, the botnet operators were able to move their infrastructure to other hosting providers and quickly re-establish their botnet.

4.1.4 DNS-BASED COUNTERMEASURES

This approach is related to the DNS-based measurement techniques described in sections 3.1.3 (DNS-based Approaches), 3.2.1 (Sinkholing), and 3.2.3 (DNS Cache Snooping). Depending on the type of botnet, many malware samples use fixed domain names as identifiers for their underlying C&C infrastructure, which is contacted by compromised hosts.

If a domain name like this can be found to be related to malware, and it has been established that it is used for malicious purposes only, the domain should be shut down by the responsible registrar. The actual act of shutting down a domain still has various dependencies, ranging from a frequently-needed court warrant, involving law enforcement in the country of the registrar, to the willingness of the registrar to cooperate in the case.

In general, this method cannot be applied if a botnet uses a legitimate domain or service to perform its communication. Popular examples observed in the wild include twitter RSS feeds or similar social network accounts with comparable features that are used to propagate commands. These commands are implemented as status messages on a crafted profile, as seen recently by a growing number of botnets like the Mehika Twitter botnet, which originated in Mexico [130].

In September 2010, a bill entitled "Combating Online Infringement and Counterfeits Act" (COICA, S.3804) [131] was introduced to the 111th US Congress by Senator Patrick Leahy. This bill includes passages that empower the Attorney General to bring action by US District Courts against domain names that are connected to the infringement of copyright or other laws in order to protect goods and trademarks. This

would allow ISPs to be ordered to block traffic from and to rogue domains, as well as the seizure of the advertising accounts of those selling advertisements on these sites [132]. This bill was interpreted mainly as a tool for faster actions on behalf of the Digital Millennium Copyright Act (DMCA) [133]. Depending on the case, however, the bill also allows fast-tracked action against maliciously-used domain names, e.g. in the context of botnets. It should be noted that COICA faces strong opposition and is referred to as a censorship law [134], which has prompted open letters from, for example, several law professors, press freedom organisations and the Public Interest Registry. The bill was not passed in the 111th US Congress but is very likely to be reintroduced in the 112th US Congress.

DNS Response Policy Zones (RPZ) have been presented [135] as an initiative of the Internet Systems Consortium towards hardening DNS against malicious exploitation. The idea of DNS RPZ [136] is to intercept potentially malicious activities, while the related malicious domain names are being resolved. For this, recursive DNS resolvers may subscribe to blacklists and use them to monitor queried domain names and act accordingly. When they recognise a listed domain name, they may end the resolving process and refer their findings back to the issuer of the query. With the implementation of BIND 9.8.0 beta 1, Internet Systems Consortium (ISC) has released a version of their DNS server supporting DNS RPZ.

Another DNS-based approach that aims to reduce the secondary effects of malware is the introduction of authenticated DNS traffic, with approaches like DNSSEC [137] or DNSCurve [138]. Signed responses of DNS servers offer the possibility of requiring authentication of DNS responses, thereby shutting out malicious DNS servers [139].

Example: Ramachandran et al. [140] have studied botnet behaviour with regard to the circumvention of blacklists. They pointed out that spam botnets query well-known spam blacklists in order to identify which bots can and cannot serve as valid spammers. Having defined two main characteristics of legitimate mail servers, Ramachandran et al. derived heuristics which were then used in a graph-based approach, suitable for detecting these reconnaissance patterns of botnets. By applying them, Ramachandran et al. were able to identify new spam bots through the queries they sent to blacklists. Furthermore, they found out that such reconnaissance is likely to happen, even across botnets. This means that the bots of one botnet that have already been blacklisted, and can be considered useless for effective spam emailing, perform queries for other bots and even bots from other botnets. This is an example of bots divided into groups according to their capabilities. Finally, Ramachandran et al. proposed countermeasures that target the query behaviour of bots and which could help mitigate the global spam level. For example, blacklist information services may generate custom answers, depending on the IP address or the sender of the query. If the sender's IP address is a known address of a bot or spam sender, the queried IP address can be included in the blacklist.

Example: Antonakakis et al. have presented Notos [141], an analysis system for the dynamic generation of reputation scores for domain names. Their system is based mainly on two information sources:

- Historical DNS information that is passively collected from recursive DNS resolvers operated by ISPs for modelling the benign use of DNS.
- Spam-traps, honeynets, malware analysis services and comparable instruments for modelling the use of DNS by criminals.

Notos is able to assign reputation scores to previously unseen domain names and has achieved high true-positive (96.8%) and low false-positive (0.38%) rates.

4.1.1.5 DIRECT TAKEDOWN OF COMMAND-AND-CONTROL SERVER

The mitigation technique known as direct takedown or decapitation aims at eliminating the instances of command-and-control servers with which bots are remotely controlled. In order to perform a takedown, a centralised botnet architecture, as often used in IRC- and HTTP-based botnets, is required.

The first step in applying this method is the identification of the target's IP address. Techniques that yield the address are described in detail throughout chapter 4 (Measurement and Detection Techniques). At this point it should be noted that working directly towards the IP address might not be sufficient if the botnet is also using a domain name. In this case, the actions described in section 4.1.3 (DNS-based Countermeasures) should be coordinated in parallel with the actual takedown to avoid the possibility of the botnet reacting by setting up a new command-and-control server under the same domain name but with a different IP in order to regain control of the botnet. This could even lead to a migration of the botnet's services to more resilient ones, complicating future actions against the botnet.

After the IP address has been identified, the next step is to determine the responsible service provider or data centre that is hosting the corresponding server and contact him. As this step involves cooperation between entities that can be situated anywhere in the world, several challenges may arise:

- The most severe problems may result from the non-cooperative behaviour of the hosting provider responsible for the server.
- In some countries, the hosting provider cannot even be ordered to cease operation of the server due to the lack of political control and corresponding laws.
- Different time zones can influence the availability of contact points.
- Possible language issues that might complicate communication in general.

A request for a takedown depends mainly on the terms and conditions of the provider

as well as the laws of the responsible country. Botmasters will generally try to deploy their servers for malicious operations with providers who guarantee their clients anonymity and robustness, if their server is the target of an investigation. Such service providers are usually called bullet-proof hosters. Cloud computing services may also become increasingly interesting for cyber criminals because of their flexibility. Cases of command-and-control servers hosted in Amazon's EC2 service [142] and Google's AppEngine have already been detected in 2009 [143].

Several experts questioned in the survey stated that personal contacts among investigators, service providers, and law enforcement agencies, as well as a high degree of trust, are essential to being able work efficiently when performing takedowns.

As the takedown is performed either by removing the network connectivity or by powering off the server, this can result in data loss, together with the destruction of evidence, e.g. non-persistent processes that only reside in memory and might be used in further investigations of the command-and-control server.

Direct takedowns of command-and-control servers suffer from the same problems as the DNS-based approaches described in section 4.1.3 (DNS-based Countermeasures). Because the infected hosts are not cleaned, the systems are often left vulnerable to further threats, as many types of malware use Anti-AV mechanisms and prevent the operating system from applying patches.

Example: Song et al. [21] presented a detailed analysis of the MegaD spamming botnet. They consequently revealed the management architecture of MegaD, which is divided into Master Servers that issue commands, Template Servers that provide target emails and templates for spam campaigns, Drop Servers that are used to distribute update binaries and SMTP servers that are mainly for testing the bots' ability to successfully send spam and receive status updates. Furthermore, Song et al. tracked the results of a takedown attempt by FireEye [144]. Although the takedown seemed successful at first, the MegaD botherders were able to regain control within three weeks, even expanding the spamming capabilities of the network. Song et al. stated that they believe the botherders used a Pay-Per-Install (PPI) service to update the bot software for new command-and-control servers.

4.1.6 PACKET FILTERING ON NETWORK AND APPLICATION LEVEL

This method is closely related to the measurement technique described in section 3.1.1 (Packet Inspection). It extends the idea of transparent monitoring and detecting to the actual application of further actions, if suspicious activities are recognised. The requirements are broadly similar to those described in section 3.1.1. Either detection signatures are needed to identify malicious traffic patterns or, alternatively, knowledge about IP addresses that are known command-and-control servers has to be integrated.

The filtering can generally be applied at a host, network and ISP level. A typical

component that performs packet filtering at host level is a desktop firewall. Its purpose is to monitor the network activities of all active processes. As the amount of traffic at host level is usually manageable, deep-packet inspection is applicable. Often, user or administrator interaction is required to allow or deny network access for certain applications, if no suitable rules have been specified for them yet. In this case, a pop-up will usually inform the user about the application and the remote host being targeted by the connection attempt.

If the technique is applied at the network level, packet filtering is usually performed by a firewall. Additionally, an intrusion detection system can be enhanced to not only monitor and report events but also automatically take actions, like dropping packets or closing connections, depending on the severity of recognised events, thus promoting the IDS to an intrusion-prevention system. The method also works at flow level, if communication endpoints have been unambiguously identified as malicious by, for example, using blacklists or tracking information for known command-and-control servers.

Packet filtering may also be applied at ISP level. By only inspecting packets originating from its own network, it is already possible to filter all packets that have a source address that does not belong to the address space owned by the ISP. Such spoofed source addresses are usually a sign of malicious activities, e.g. in the case of a Distributed Denial of Service attack. If Deep Packet Inspection is considered, the same issues regarding scalability arise that were already mentioned in section 3.1.1

4.1.7 PORT 25 BLOCKING

'Port blocking' is a preventive measure that can be applied by ISPs to reducing the amount of spam mails traversing their network. As more than 87% of all email is reported as spam [6], mitigation of this threat is desirable. The following approach is based on the assumption that the use of unauthenticated services via port 25, like direct mail exchange (MX) or open relay mail servers, is almost exclusively for spam distribution purposes.

Therefore blocking port 25 at ISP level has been recommended as best practice by the Messaging Anti-Abuse Working Group (MAAWG) since 2005 [145]. Any e-mail submission service should be offered via port 587, as defined in RFC 2476 [146] and use authentication. Port 25 blocking is also recommended as essential best practice, with a high-impact factor, by ETIS (Global IT Association for Telecommunications) [147]. According to ETIS, the spam outputs of Turk Telecom and Telecom Italia were significantly reduced through the introduction of port 25 blocking. Exceptions to general port blocking can be made for accredited, legitimate services like email providers, which can be whitelisted by ISPs.

Example: Schmidt [148] has presented a case study on port 25 blocking. A subnet of 20,000 subscribers to a Multiple Systems Operator (MSO) with a total of 240,000

Internet subscribers was treated with adaptive port 25 blocking, which was applied according to the behaviour of the subscriber's computer. The rules included, for example, the immediate blocking of hosts with more than 40 port 25 connections in any one minute and frequent notification of subscribers close to, or above, a limit of 5 e-mails per minute on port 25. During a two-week test in May 2005, abuse complaints for the subnet decreased to almost zero from nearly 100 per day, and only 5 legitimate high-volume users complained about the regulations. After activating port blocking for the entire 240,000 subscribers, email output dropped by 95% and complaints decreased to single digits from an average of 600 per day.

4.1.8 WALLED GARDENS

The concept of a 'walled garden' has the goal of protecting an ISP's customers and other Internet users from further damage, by intercepting and isolating outgoing connections from a detected infected host. According to [149] the procedure can be divided into three stages: detection, notification and remediation.

For the initial detection of malware on a remote system and to increase its effectiveness, the ISP may use one of the techniques described in chapter 3 (Detection and Measurement Techniques), e.g. honeypots, DNS-based methods that watch for malicious domain names, NetFlow analysis, evaluation of blacklists and feeds, or a combination of such approaches. After an infection has been confirmed for the connection, the user is placed in a walled garden as shown in figure 11. This means that his connectivity is more or less strictly constrained, depending on the ISP's policies, and that a notification is sent to him. It is very important that the remote detection of the infection works reliably, because the the actions performed on the basis of the detection are strong interventions and false-positives are likely to be perceived very negatively. The general idea of a walled garden is to forbid almost all connection attempts by the isolated user, except those to a defined whitelist of malware mitigation services. All other DNS queries are handled with a crafted answer that will lead the customer to a prepared website belonging to the ISP, informing the customer about the detected infection and offering helpful advice, e.g. providing removal instructions, a list of links to malware removal tools, measures to decrease the risk of future infections and a guide to backing up user data safely. Self-remediation is therefore the usual target of this approach, as it is argued that responsibility for the affected systems lies in the hands of the customers.

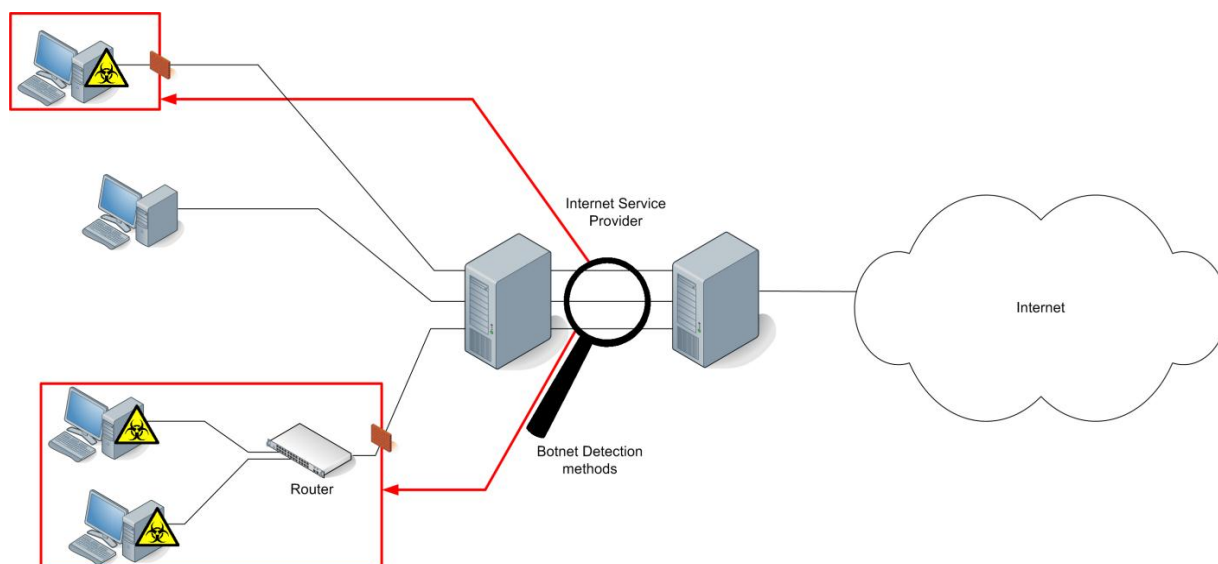


Figure 11: Functioning of walled gardens.

Some nuances concerning the realisation of a walled garden should be considered. Because restricted connectivity is a severe limitation from the customers' point of view, a policy or implementation decision may be made to offer an option to leave the walled garden, based solely on the customers' own judgement. If no disinfection of the system is performed, full connectivity will be reactivated for a certain length of time only. The connection constraints are then renewed as soon as detection of an infection recurs and/or a grace period elapses. It is important to note that limiting Internet access can lead to drastic side effects, if telephony is operated over the same connection. In this case, blocking connections in general may disable the customer's ability to dial emergency calls, which does not appear desirable.

A variation of the walled garden concept is services that prohibit access to potentially dangerous websites by using the blacklists that contain domain names and IP addresses aggregated by initiatives like StopBadware [150] or Spamhaus [126].

4.1.9 PEER-TO-PEER COUNTERMEASURES

Every peer-to-peer based network has to handle information management to do with connectivity and routing. New peers have to be advertised within the network and the information about different peers has to be publicised in the network. Typically, this is achieved by maintaining and exchanging peer lists. These are collections of records that contain the contact information of other peers. A complete overview of the entire address space is dynamically shared throughout the peer-to-peer network, but is not available at a single point.

Countermeasures aimed at peer-to-peer based botnets exploit this concept of peer-lists and their publication mechanisms. One approach is to try to pollute these lists by

inserting invalid connection entries of non-existing peers. Advertising a large number of invalid peers to known peers can result in a loss of overall connectivity and can even be sufficient to pollute and disrupt the whole botnet, as the available memory for peer-lists is limited. In due course, bots will not be able to store information about new peers and drop old but still valid information about existing peers. A comparable effect can be achieved by injecting duplicate connection entries that create ambiguities and influence the botnet's routing capabilities.

Another approach, called Sybil Attack, aims at manipulating the routing in peer-to-peer networks by introducing a large number of crafted and seemingly independent peers that are controlled from a central component. The peers are usually injected as distributed as possible, relative to the topography of the P2P network, in order to have as much impact on routing as possible. The Sybils will continually advertise themselves to the existing original participants of the peer-to-peer network, the bots, answering any route requests to them with identifiers of other Sybils that are of course known to the central component managing them. That way, the routing tables maintained by the bots will be poisoned over time by Sybils, taking over large parts or all of the routing in the P2P network. This position is equivalent to taking away control from a botmaster, because it can be used to inspect and sinkhole traffic or disrupt the functionality of the network.

The application of these techniques, when considered by researchers, should be coordinated with law enforcement. Any such interactions with a botnet may alter the forensics of the "crime scene", leading to modified data, e.g. the measurement number of bots, and therefore victims, which may be relevant if such cases are subject to legal prosecution [151]. Additionally, many of these techniques involve the alteration of traffic without the consent of the owners of victimised computers. Even with good intent, this makes the application questionable [152].

Example: Holz et al. [106] not only presented a method for measuring peer-to-peer botnets, but, after analysing the communication protocol and being able to enumerate all hosts, they established a basis for the evaluation of countermeasures. Assuming that an effective mitigation method requires an attack on the botnet's control infrastructure itself, several theoretical strategies against classical peer-to-peer networks were pursued.

4.1.10 INFILTRATION AND REMOTE DISINFECTION

Infiltration, in the context of botnets, describes the process of finding a way to impersonate the botmaster and obtain control over the infected hosts. As almost all botnet families exhibit differences in their implementation and mode of operation, an infiltration can be considered as a tailored approach for each targeted botnet. The approach can be divided into two stages. The first stage consists of analysing the botnet's communication mechanism. The second stage involves the design and implementation of a tool for the actual infiltration, based on the previous analysis step.

The effort required to perform this technique is largely weighted towards the first stage, as this involves large amounts of reverse engineering of the malware binary (see section 3.3.1 (Reverse Engineering) for more information). The main goal is to spot weaknesses in the botnet's communication protocol, which may serve as attack vectors on which the actual infiltration can be constructed. This usually involves identifying the structure of botnet command messages, cryptographic measures used by the botnet and authorisation mechanisms for verifying the origin of the commands.

If the first stage is successful and the means of infiltrating the botnet have been found, it is possible to emulate the botnet's control mechanism through a customised tool that emulates the protocol. The main goal of this process is to find a way of performing a remote disinfection of infected hosts. For example, integrated in some early IRC-based botnets was a command that issued a local disinfection routine on the zombies, e.g. SDBot's ".remove"-command as documented in [153]. The original intention behind including this command was to allow the destruction of evidence on the compromised system and to wipe traces of the infection. Another way of achieving remote disinfection is to exploit the update functionality present in almost every botnet. In this case, instructing a bot to update itself with a crafted file containing a cleaning routine achieves the desired disinfection.

Applying remote disinfection to bots means fighting fire with fire, in terms of modifying a remote system without having confirmed the willingness of the affected user or obtained legal permission. In any case, activating autonomous disinfection routines is not allowed under the existing law in many countries. In addition, this process always runs the risk of unforeseen side effects, as testing all affected systems and configurations is obviously not possible.

Example: Different prototypes and approaches for the takeover of different botnets have been demonstrated by Leder and Werner [152]. Their research included disinfection strategies for various major botnets and botnet architectures, including the Storm and Conficker botnets.

4.2 REGULATORY AND SOCIAL COUNTERMEASURES

The approaches covered in this section rely less on technical measures but aim at improving the environment needed for botnet countermeasures. This includes end users who are affected by the impact of botnets and how to improve the coordination and courses of action in handling the botnet challenge from an international point of view.

4.2.1 DEDICATED LAWS ON CYBERCRIME

When discussing countermeasures against botnets, it is only natural to take legislation and specialised laws on cybercrime into account. As this is a very dynamic and fast-paced field, and initiatives have been taken with different motivations, existing laws

vary significantly at a global level. A more detailed report on the legal aspects of mitigating botnets will be published by ENISA in Q2 2011 [2].

A case that shows the need for the establishment of laws on cybercrime is the ongoing trial of the botnet developers behind the Mariposa botnet. According to Captain Cesar Lorenzana of the Spanish Civil Guard, “In Spain, it is not a crime to own and operate a botnet or distribute malware. So even if we manage to prove they are using a botnet, we will need to prove they also were stealing identities and other things, and that is where our lines of investigation are focusing right now.” [154]

A directive to advance the fight against cybercrime that requires Member States of the European Union to criminalise the creation of botnets and their use for committing criminal offences was proposed by the European Commission on September 30 under “Proposal for a Directive on attacks against information systems, repealing Framework Decision 2005/222/JHA” [122].

Elements of this proposed Directive that are relevant for the fight against botnets and cybercrime include:

- Illegal access to information systems, illegal interference with information systems and computer data, and illegal interception of the transmission of computer data from or within an information system are treated as criminal offences.
- The use of tools (tools refer to, for example, malicious software, including botnets) for committing the above-mentioned offences is penalised.
- Penalties for committing an offence are increased to a maximum term of imprisonment of at least two years and of at least five years in aggravating circumstances that include the use of botnets.
- Cooperation is enforced by strengthening contact points and requiring that urgent requests are answered within 8 hours.
- The creation of an obligation to collect basic statistical data on cybercrimes.

Another example shows that providers might be forced to change their practices. A change in Swiss law, confirmed in 2007 [156], prohibits the sending of mass advertising (which includes spam) in cases where the sender has not acquired explicit approval from the recipient [157]. Furthermore, service providers are required to actively combat such mass advertising [158] (Chapter 7, Art. 45a, 1). This includes providers having to take some responsibility for disinfecting victimised customer machines that are used to generate spam emails [159] (Art 83, 3). Service providers are even required to intercept connections related to spam email and are allowed to cancel the contract in a repeat case. An additional responsibility for providers is to provide a contact point, such as specialised hotline, to deal with this topic [159] (Art 83, 4).

Other laws on cybercrime might also extend the responsibility of customers. In 2010, the German Federal Supreme Court pronounced a judgement that forced all citizens to operate their wireless networks with encryption enabled at all times. This resulted from a case where someone was accused of copyright violations through file sharing. The individual, operating an unprotected wireless network, argued that the copyright offence happened during his vacation but he was still held responsible, because operating such a device without encryption enabled was considered negligent [160].

It should be emphasised that, given the flexibility of botmasters in changing jurisdictions, the scope of any legislative initiative should not, as far as possible, be restricted to any one country, region or political grouping.

4.2.2 USER AWARENESS RAISING AND SPECIAL TRAINING

Raising user awareness is an approach to mitigating botnets focused on their root causes, namely infected end-users' computers on company networks. The occurrence of infections is often the result of computer usage driven by ignorance of potential infection sources and a lack of caution.

It is therefore reasonable to aim at improving both the end-users' technical knowledge and their sense of social responsibility in terms of secure operation. This includes, among others, the following topics:

- Education in malware-spreading mechanisms can help prevent infections from drive-by infections, included in unsolicited email, caused by linked websites. This also incorporates the handling of removable media, like thumb drives or external data storages, whose origin is not fully trusted.
- Emphasising the importance of keeping systems up to date with patches for installed software protects against known and already fixed vulnerabilities that are exploited by autonomous spreading malware.
- Information about the interpretation of potential symptoms of infection and guidance on how to treat an infection, including the use of anti-malware tools, can help minimise the potential damage from acts of identity theft and financial fraud.
- Well-considered password management can also contribute to reducing damage when an actual incident occurs.

Several campaigns and initiatives are already active in the field of user awareness raising. [161], [150], [1], [162], [163].

A common challenge of this approach is how to reach the actual target audience. IT professionals and users with a vital private interest in computers are easily reached via articles and advertisements in the popular technical press, as well as relevant online sources. In general, this group is expected to already have an enhanced awareness of

security and malware prevention. Employees whose daily work involves significant amounts of interaction with computers may be educated via special security awareness training courses organised by their company. Finally, there are casual users, who are not only especially threatened by malware but are also probably the hardest to reach.

An approach already mentioned that contributes to raising user awareness is ‘walled gardens’, described in section 4.1.7 (Walled Gardens). Another even more direct approach to raising user awareness can be the restitution of stolen credentials and identities in cases where dropzones have been taken down. On the one hand, this helps the affected users restore the integrity of their data; on the other, it also reveals to them that their system or data was compromised and that damage may result from this.

Both approaches may be complicated in terms of the interpretation of the message. Despite the good intentions behind these techniques, end-users may mistakenly “shoot the messenger” i.e. blame the service provider as the bearer of the news about their machine being infected. This may result in damage to the service provider’s reputation, which could act as a disincentive to implementing these measures.

4.2.3 CENTRAL INCIDENT HELP DESK

The idea of a central incident help desk offering consultation on the treatment of bot infections benefits from the assumption that a designated organisation bundling these services can be efficiently subsidised and advertised by a federal institution. It is closely related to the approach to raising user awareness described in section 4.2.2 (User Awareness Raising and Special Training), but extends the awareness by advising individual user victims directly on the process for cleaning their systems.

A pioneer central help desk project in Europe is the German Anti Botnet HelpDesk [161]. The project is led by eco (Association of the German Internet Industry) [164] and the German Federal Office for the Information Security (BSI) [165]. It is funded by the German Federal Ministry of the Interior. The initiative’s goal is to move Germany out of the top ten countries from which botnet-related activities originate. More information on the Anti-Botnet Initiative and other initiatives related to the fight against botnets and cybercrime can be found in section 4.3 (Initiatives and Institutions).

Decentralised approaches to informing customers of infections are employed in, for example, Australia and Japan: The Australian Internet Security Initiative (AISI) [166] started in November 2005. AISI collects data from various sources to identify infections in the Australian Internet. The data is processed on a daily basis and reports are sent to participating ISPs via e-mail. The service is free of charge for participating ISPs, and at the time of writing 80 Australian ISPs benefit from the service. Customer notification and further steps are handled by the ISPs on their own.

In Japan, the Cyber Clean Center (CCC) [163] has an active role in fighting bot infections. The organisation is divided into three working groups. The BOT countermeasure system operations group collects bot samples and cooperates with several Japanese ISPs in informing customers of infections. Effective and efficient analysis techniques are studied by the BOT program analysis group. This group also develops countermeasures and disinfection strategies based on the analysis of botnet samples. Lastly, the BOT infection prevention promotion group works on the publication and promotion of information about threats connected with botnets and provides collections of bot samples to security and anti-malware vendors. CCC does not directly notify users of infections, but participating ISPs may choose to inform their customers by email or letter. See section 4.3 for more information on these initiatives.

4.2.4 ENHANCE COOPERATION BETWEEN STAKEHOLDERS

One key factor in fighting botnets successfully is the extent and effectiveness of cooperation between the stakeholders involved, as the required knowledge, resources, and authority are generally distributed among several parties. For example, security researchers from academia and industry may have information about the key factors of a certain target botnet, e.g. a list of identified command-and-control servers and communication mechanisms, or possible attack vectors and the appropriate tools for detection and measurement. CERTs may have information and evidence about incidents related to the botnet that support investigations. Law enforcement have the power to order or perform countermeasures against the botnet, as described in section 4.1. This indicates that different stakeholders are associated with certain responsibilities and capabilities that lie within their area of authority. To improve botnet mitigation procedures, these role models need to be clarified to support possible cooperation. The method therefore aims to improve and structure the communication management between the stakeholders involved.

One general aspect of behaviour that has been regularly observed is that parties or victims who start botnet countermeasures tend to plan their mitigation attempts in isolation at first and to develop strategies before they start to look for cooperation. In addition, dedicated groups specialising in mitigation often have their own established private communication channels, such as private mailing lists, IRC channels or ticket systems, in order to lessen the risk of being infiltrated. The management of these channels is usually tight, and so the external exchange of information at an international or even global level tends to be complicated. Common issues include a lack of trust towards unknown groups or individuals, often also the absence of a contact point with these groups and a lack of knowledge about useful information and data formats that would enable more efficient processing. A general issue is the lack of time, usually due to heavy internal workloads that leave little room for additional activities like, for example, the strengthening of cooperation.

Regarding data formats for information exchange about computer security incidents,

various standards have emerged from organisations like IEEE and IETF.

- Malware samples. The IEEE Industry Connections Security Group (ICSG) [167] has published a XML schema called Malware Meta-Data Exchange Format (MMDEF) to facilitate the exchange of malware samples. The format has been adopted by various major computer security organisations.
- In the context of computer incident handling, the IETF has developed the Incident Object Description and Exchange Format (IODEF) [168]. IODEF is XML-based and contains several classes that are needed for a detailed description of a computer incident.
- The Malware Attribute Enumeration and Characterisation Language (MAEC) [169] aims to provide a framework for encoding and communicating information about malware. The language is based on attributes such as behaviours, artefacts and attack patterns, which describe malware unambiguously.
- The Abuse Reporting Format (ARF) [170] was standardised by the IETF as a format to make the reporting of spam easier. The main goal is to improve the generation and interpretation of automated abuse reports. X-ARF [171] is an extension to ARF that is also capable of capturing non-messaging related incidents, for example network or phishing attacks.

Cross-party collaboration and the global consolidation of data records help to obtain a more detailed view of the international botnet situation and recent developments. As botnets are distributed across all countries, this collaboration is a necessary step. Cloud-based approaches can provide efficient processing of data records like blacklists and spam records.

Example: The Italian Chapter of The Honeynet Project [172] has created the Dorothy Framework as an open botnet analysis framework that can be used for automatic tracing and visualising activities. The project aims to provide relevant information automatically to ISPs and law enforcement in order to stimulate a mitigation strategy. Dorothy collects malware samples with a honeypot, executes them in a sandbox and extracts information that can be used to create a drone that is then inserted into the botnet. Data from the monitored botnet is gathered in real time and presented in a web interface. Dorothy is restricted to IRC-based botnets in its current version.

4.3 INITIATIVES AND INSTITUTIONS

With the growing importance of the global botnet threat, several cooperative initiatives were started at a national and international level in order to concentrate activities. The primary goals of these groups include the building and maintenance of trusted relationships between different organisations to speed up operations, making it easier for the process of critical information exchange to support investigations and knowledge transfer, and to improve the competences of all the parties involved.

As these competences vary (both from a professional point of view and a legislative perspective) between the different parties, collaboration and coordination are clearly desirable. Some of the initiatives are summarised in the following section. A more comprehensive overview is given in [173].

4.3.1 NATIONAL INITIATIVES

First, two initiatives of European Member States (Germany, Netherlands) are listed in this section, followed by initiatives from other countries (Australia, Japan and South Korea).

GERMANY

The German Anti Botnet HelpDesk [161] is a project led by Eco (Association of the German Internet Industry) [164] in cooperation with the German Federal Office for Information Security (BSI) [165]. Financial support is provided by the German Federal Ministry of the Interior, providing 2 million Euros for the first year of operation. Eco guarantees continuous operation of the initiative for another year. The biggest share of the funding is being used to install a call centre for supporting users in cleaning their systems.

The initiative aims to move Germany out of the top ten countries from which botnet-related activities originate, which was also the reason for starting the project. The work of the Anti Botnet HelpDesk [161] is guided by the approaches of Australia, Japan, and Korea. Operational activities focus on ISPs' customers and are divided into three steps:

1. Victims are identified indirectly through spam traps and honeypots (cp. section 3.1.4 (Analysis of Spam Records) and section 3.1.6 (Honeypots)).
2. Users identified receive a notification via their ISP. At the time of writing, six ISPs are taking part in this process. Between them, they have 23 million customers and account for almost the entire German broadband market [174]. Notification is given in many different ways, by e-mail, (snail)mail, or through a walled garden. It contains general information about how to handle infections from malware and includes links to removal software. An anonymous ticket number, which can be used in step 3, is also included.

3. Users are offered help interactively in case they were unsuccessful in removing the malware themselves. For this, a central support centre has been set up. After receiving the ticket, users can call a hotline to get first-level support for advice on removing the infection. If this still does not succeed, the customer is forwarded to a second-level support specialist for further and even more detailed instructions. The second-level support also aims to improve the first-level support procedure through experience gained in cases handled by first-level support, as well as their own.

Challenges to this project are Germany's rather strict privacy and data protection laws. As a consequence, no monitoring of connections takes place and only indirect detection approaches, such as honeypots and spam traps, are deployed. In addition to the operation of the centralised call centre, a ticket system for anonymization has been designed. This is intended to ensure the privacy of identified users and to enable tickets to be allocated to customers for presentation to their responsible ISPs.

This project is a cooperation between a growing number of participants in Germany, consisting of ISPs, email and social networks service providers, and IT-security vendors.

NETHERLANDS

In July 2009, 14 Dutch ISPs agreed to unite in the fight against botnets [175]. These ISPs represent almost all the consumer Internet connections in the Netherlands, accounting for 98% of the market. One of the initiators of this alliance was the Dutch Telecom Regulatory Authority (OPTA) [176]. Whilst the agreement features no provisions on how mitigation will take place, the approaches defined include the exchange of information on infected systems and best practice between participating ISPs, a customer notification service, and, as a protective measure, constraints on identified hosts' Internet access.

The National Infrastructure against Cybercrime (NICC) [177] is a Dutch programme, founded in 2006, with a focus on improving the resilience of critical infrastructure against threats originating from cybercrime. Although NICC is not fighting crime directly, it supports the parties involved in their efforts to improve the safety of IT-related work processes. The initiative serves as a link between the parties, making resources available and encouraging information exchange. Topics for information sharing are incidents, vulnerabilities and good practice. Communication is organised in a trusted public-private environment, where all participants are known explicitly by name. Data sharing happens, depending on the required level of confidentiality. Cooperation and levels of trust are improved by regular meetings involving representatives of stakeholders in different sectors and the government. Apart from these meetings, important topics of concern to parties in all sectors are organised, such as a working group for "Secure Internet Banking" that comprises banks, ISPs and security vendors. An NICC project manager cites a major benefit of the programme as

the exchange of knowledge and building of trust between the parties, which has led to a social network that is now protecting the technical network. In addition, the government has a better insight into security processes, which improves the awareness of the available information and requirements and so results in more efficient policy initiatives.

AUSTRALIA

In 2005, the Australian Communications and Media Authority (ACMA) [178] started the Australian Internet Security Initiative (AISI) [166] with the intention of reducing the number of infected computers in Australia connected to the Internet. This initiative is connected with the Internet Industry Code of Good Practice [179], which describes a guideline of actions for implementation by participating Internet Service Providers in cooperation with CERT Australia [180]. The programme is voluntary, but well received. At the time of writing, the number of participating ISPs has increased from 6 in 2005 to 100 [166].

The main idea of this AISI is to raise awareness of the overall malware situation by performing centrally organised, remote identification of infected devices and notifying the responsible network providers, e.g. Internet Service Providers or business IT administrators. AISI sends out [166] daily reports to the participating parties and serves as a cornerstone of the project. Depending on network providers' implementation details, further actions are taken, from forwarding infection information, and giving advice on removing the malware, to constraining the Internet connectivity of the infected system in order to protect the owner and others from further harm.

Furthermore, any incidents that give rise to suspicions of criminal activity will be reported to the government agencies responsible.

JAPAN

Japanese efforts in the fight against botnets are concentrated in the national Cyber Clean Center (CCC) [163], and began in 2006. A steering committee, consisting of Japan's Ministry of Internal Affairs and Communications and its Ministry of Economy, Trade, and Industry, organises its activities. Several institutions and companies from various stakeholder groups are working together, including more than 70 Japanese ISPs, between them responsible for about 90% of all Internet users. Participants are divided into three internal groups:

- The Telecom Information Sharing and Analysis Centre (ISAC) Japan [181], responsible for the Bot countermeasure system operation group.
- JP-CERT [182], responsible for the botnet program analysis group.
- The Information Technology Promotion Agency (IPA) [183], which works on raising user awareness and is responsible for the botnet infection prevention promotion group.

The overall procedure is comparable to Australia's approach, as CCC also centrally sends information about detected infections to the participating ISPs. It is also more oriented towards actual infections, because notifications are customised for certain specimens of malware and specialised disinfection information, and tools for removal of these specimens are distributed.

SOUTH KOREA

As a reaction to significant DDoS attacks against their country [184] and reports indicating high infection rates among South Korean computers, the Korean Internet Security Agency (KISA) [185] and the Korean CERT (KRCERT) [186] started an extensive anti-botnet campaign. The approach consists of three parts:

- Infected machines are remotely detected by various means. This includes mainly specialised DNS servers, which monitor suspicious queries and connections. Further data comes from malware analysis and reports of Intrusion Detection Systems.
- KRCERT performs extensive botnet monitoring and mitigation by using a centralised DNS management service [187]. This way, domain names that have been confirmed as serving malicious purposes can be easily sinkholed. For this, crafted DNS Resource Records containing the domain name and IP addresses used by the sinkhole, instead of the malicious server, are distributed to DNS servers operated, for example, by ISPs.
- To complement the mitigation efforts, cooperation between KRCERT, ISPs and IT security vendors is sought in notifying end-users of infections and providing them with removal tools for cleaning up their computer systems.

In addition to these efforts, Korea has set up the E-Call Center 118 [188], a free emergency hotline for handling Internet incidents. The call centre operatives are trained to give advice on the removal of malware, to deal with, or identify, spam emails, and to answer questions on privacy and Internet technology in general.

4.3.2 INTERNATIONAL INITIATIVES

In this section, selected globally-oriented anti-botnet initiatives and projects are presented.

ITU BOTNET MITIGATION TOOLKIT

In 2007, the ITU [189] started the development of a Botnet Mitigation Toolkit. This toolkit characterises the botnet threat in general and provides recommendations for attacking the problem at various levels. It is divided into three political, technical and social aspects of the problem.

- The section on policy addresses the legal situation, promotes more widespread competence in handling cybercrime, promotes cooperation among stakeholder

groups and provides insights into the balance between user privacy and overall security.

- In the technical section, a general overview on botnet detection and investigation is given, the role of ISPs, domain name registries and registrars is outlined and the role of financial institutions in building capacity in botnet mitigation is addressed.
- Finally, the social measures include starting broad-based targeting campaigns to raise user awareness and using visual media to make them more accessible. These will also address making security software easier to obtain and install, and creating a mindset for performing regular software updates.

ITU-T CYBEX

The standardisation of a global approach for a framework, currently being worked on by the ITU-T, is the Cybersecurity Information Exchange Framework (CYBEX) [190]. The framework focuses on bringing communication between different cybersecurity organisations up to the same level, eliminating errors caused by a lack of understanding and the advance of automation. The information is structured and aggregated according to areas of interest, like knowledge accumulation or incident handling. Organisations will be identified by a unique Object Identifier (OID), making services and information sources easier to find.

OECD – WORKING PARTY ON INFORMATION SECURITY AND PRIVACY (WPISP) - VOLUNTEERS' GROUP ON BOTNETS

WPISP [191] is an exchange platform working in an international context and supported by the OECD Secretariat within the Directorate for Science, Technology and Industry. Topics of interest include Malware, Cybersecurity Policies and Critical Information Infrastructure Protection (CIIP). WPISP started a special group in 2010, focusing on botnets.

This group analysed the importance and position of ISPs in botnet mitigation and how governments can help them enhance the stability and security of the Internet. The results of this research identified a global trend in this direction, with national efforts taking place similar to those described in the previous section. Furthermore, the results of the work of ongoing Public-Private-Partnerships (PPPs) are interpreted as promising.

As part of the WPISP efforts, a study on the role of ISPs in botnet mitigation [192] has been published by van Eeten et al. Key findings are:

- ISPs have a very important strategic position in the fight against botnets, because they connect their customers to the Internet and therefore enable their computers involuntarily to communicate with botnets.
- ISPs have differing performances in botnet mitigation efforts. In similar market conditions with comparable customer bases, notable variations in infection populations have been identified.
- The degree of software piracy in a country can probably be correlated with potential malware infection rates, because illegal software versions may be modified with malware before their release to the Internet and used as spreading vectors.

4.3.3 TARGETED BOTNET MITIGATION WORKGROUPS

In this section, three examples of workgroups that targeted single, unusual botnets are presented. All three are examples of how excellent cooperation led to the final shutdown of the large Conficker, Mariposa, and Waledac botnets.

CONFICKER WORKING GROUP

The Conficker Working Group (CWG) [49] started its fight against the Internet threat created by the Conficker malware in late 2008. CWG consisted of various security researchers from, for example, Microsoft, ICANN, domain registry operators, anti-virus vendors and academic researchers [193]. Their resources were bundled into a coordinated countermeasure, leading to cooperation between several international institutions and organisations from both the public and private sectors.

Because of the remarkable success achieved by the CWG, it has been perceived as a role model. Therefore an independent “Lessons Learned” study [193] was conducted on the work of CWG, funded by the US Department of Homeland Security. Apart from giving a historical view on the events around Conficker, the document contains valuable information about best practice in cooperative botnet mitigation, together with a comprehensive list of recommendations for future groups.

MARIPOSA WORKING GROUP

After the discovery of the Mariposa botnet in May 2009, a working group was set up by Defence Intelligence, Panda Security, Neustar, Directi, the Georgia Tech Information Security Center and other anonymous security researchers [194].

First, the botnet was shut down in a coordinated strike against the botnet command-and-control infrastructure. Following this, the botmasters eventually regained control by bribing a person from a domain registry [195]. Because one of the botmasters used

his home Internet account, he was identified and arrested, as were two other botmasters and, later, also the developer of the bot malware that was used [196].

OPERATION B49 (WALEDAC WORKING GROUP)

Operation B49 [197] was a 10-month effort against the Waledac botnet and the name of the activities of a working group including, among others, the Shadowserver Foundation, the Vienna University of Technology, University of Mannheim, University of Bonn and the University of Washington. With the information about malicious domain names collected, Microsoft was able to take legal action in the form of a restraining order, without alerting those who registered the domain names. A final shutdown was made possible through cooperation with China CERT, which helped to take down further domain names that were not affected by the court order.

4.3.4 OTHER PUBLIC PRIVATE PARTNERSHIPS (PPPs)

In this section, more Public Private Partnerships with a European focus are presented.

EUROPEAN PUBLIC PRIVATE PARTNERSHIP FOR RESILIENCE (EP3R)

The European Public Private Partnership for Resilience (EP3R) [198] is a policy initiative on Critical Information Infrastructure Protection and was adopted by the European Commission in March 2009. In the initial phase the priorities are the Internet's communication systems and fixed and mobile telecommunication services.

The aim of EP3R is to create a governance framework in an EU-context for strengthening the security and resilience of critical infrastructure by including representatives of relevant stakeholders from the public and private sectors. The programme focuses on prevention and preparation as well as on enhancing trust and cooperation between participants. The creation of a common understanding of the conflict between market orientation and security will be supported through intense information sharing and a stocktake of best practice.

While some initiatives and approaches exist at a national level, EP3R will harmonise practices at an EU-level to enable long-term benefits to accrue. EP3R is therefore intended, where possible, as a complement to existing efforts and to provide the necessary flexibility for the continuous improvement of procedures.

DEMONS

A project from the Seventh Framework Programme (FP7) [199] is Decentralised, cooperative and privacy-preserving MONitoring for trustworthiness (DEMONS) [200]. Its objective is to provide an infrastructure suitable for the detection, reporting and mitigation of incidents on a global level. Involved stakeholders are ISPs, the network equipment industry and research institutes. In the course of the work, legal obstacles

that may arise from privacy regulations will be examined and accounted for.

4.4 ANALYSIS

In this section, the approaches presented to counter the botnet threat are compared to each other and analysed, using the feature set defined in section 2.2 (Applied Features for Analysis).

At first, it is important to note that several of these approaches are connected to techniques described in chapter 4 (Measurement and Detection Techniques) because, in many cases, the relevant targets have to be identified before the appropriate countermeasures can be enlisted.

A general observation is that, although many approaches are well engineered, their application is heavily constrained by different laws and the differences in legislation and its interpretation across borders, reducing their effectiveness by introducing delays that affect reaction time or prohibiting their implementation.

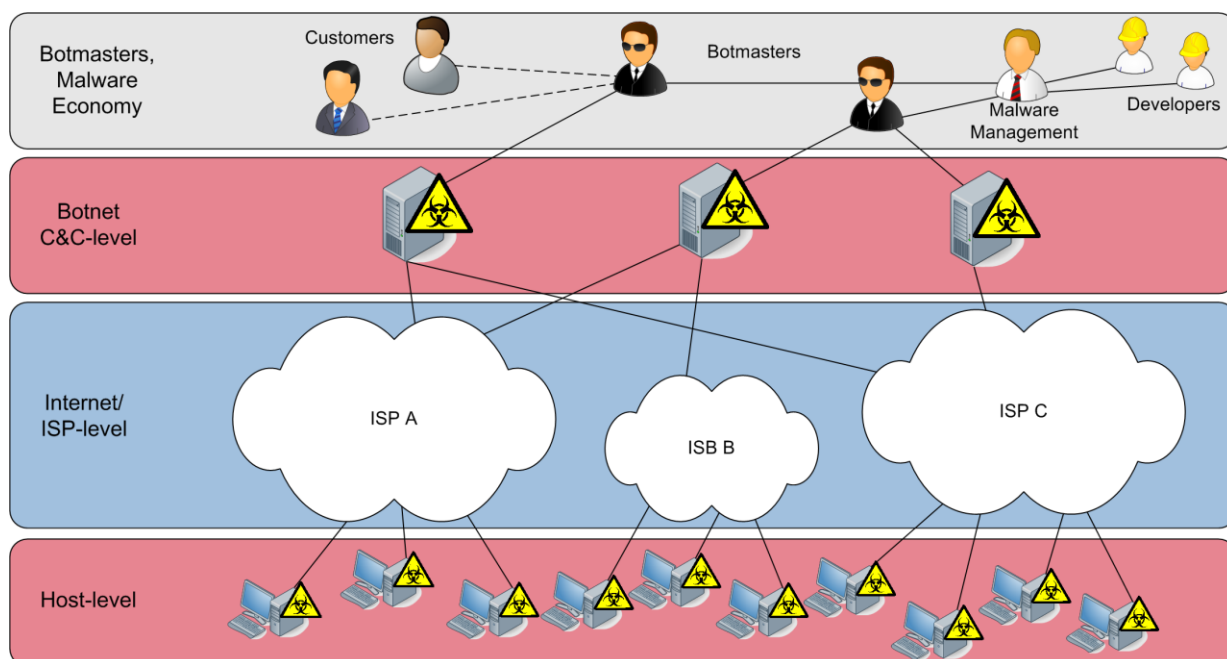


Figure 12: Layers of botnet operation.

All countermeasures operate at different levels of the communication infrastructure. While some approaches may target command-and-control servers directly, like takedowns or DNS-sinkholing, and so work against the upper tier of the technical infrastructure, other principles, like walled gardens, may include the infected hosts and therefore aim at the root cause (cp. figure 12). Others also target the criminals behind botnets directly, for example through legislation on cybercrime and regulations.

4.4.1 GENERAL CHARACTERISTICS

The group of countermeasures that are based on changes to the connectivity, such as blackholing, sinkholing, and packet filtering, offer good generality, as they do not depend on the special properties of certain botnets or types of botnets. This group's approaches use information provided by detection and measurement methods.

A good example of interplay between network-based approaches is the application of blacklists as an input source for packet filtering. Blacklists usually contain information about communication endpoints, identified through IP addresses or domain names that serve malicious purposes like command-and-control servers, or servers used for malware distribution. By incorporating real-time blacklists into packet filtering, communication with these IP addresses can be interrupted before a connection is established, preventing further damage. Alternatively, the blacklists may be analysed to determine the origin of malicious servers from their AS number. Notifying the service provider in charge can result in actions against the malicious entities. If it can be proved that a service provider is systematically supporting malicious activities, its upstream service providers may consider de-peering with it. This would result in a general loss of Internet connectivity, effectively blackholing the entire network. Examples of events like this have been observed over the last few years, e.g. in the cases of hosting providers McColo and Interchange/Atrivo in 2008.

By incorporating real-time blacklists into packet filtering, communication with these IP addresses can be interrupted before a connection is established

Other blacklists include information about hosts that were probably infected and identified by techniques like honeypots or spam traps. Results from these detection techniques may also serve as input for ISPs that apply preventive measures to their customers, e.g. walled gardens. The reaction time of these network-based approaches depends on the detection methods used as input sources. Once information is included, changes will become effective almost immediately. Nevertheless, blacklists, if used to filter communications, need to be chosen very carefully, as some of their listing and escalation rules are quite aggressive and can lead to false-positives.

One general consideration regarding the DNS system is to strengthen the process of domain registration. This could include stricter requirements for the validation of personal identity. However, its effectiveness can be questioned because many criminals use fake identities for these purposes and for interactions with public authorities. Another interesting development regarding DNS will be the integration and establishment of DNSSEC [137] or DNSCurve [138], which allows the signing of

One general consideration regarding the DNS system is to strengthen the process of domain registration. This could include stricter requirements for the validation of personal identity.

domains and provides additional measures against botnet-related criminal activities such as phishing.

The group of countermeasures that target special types, or infrastructures, of botnets are naturally less general, but they profit from this trade-off because their effectiveness increases as a result of their specialisation.

Taking down command-and-control servers is a viable approach which can often be applied because the majority of botnets have a centralised architecture that could consist of a single or several C&C servers. For example, in the case of the Bredolab shutdown in October 2010 by the Dutch National High Tech Crime Team, major parts of a whole malware family were disabled at once. In total, 143 servers were simultaneously seized and disconnected from the Internet [204]. Many of these servers were responsible for smaller portions of the botnet, which were rented or sold as parts to botmasters. Shortly after the takedown, FireEye [205] announced that they had identified that at least one C&C server related to Bredolab was still online. Symantec explained in their December report on the State of Spam & Phishing [206] that the Bredolab takedown had an influence on the general downward trend of spam volume observed. In January 2011, Avira reported that they had observed an increase in activity related to Bredolab, which was the spamming of fake Facebook mails [207]. Taken together, these reports show that the takedown was an important success, but also that even a single remaining server can leave room for the botnet to recover (and even become more resilient).

Takedowns can be applied only against centralised elements of botnets.

Other successful takedowns have targeted the command-and-control infrastructure indirectly by modifying DNS answers, and so sinkholing domain names. Examples of this sinkholing approach are the December 2009 Mariposa botnet and the February 2010 Waledac botnet takedowns. In the case of Mariposa, takedown activities started in May 2009, according to Defence Intelligence [194], with the setting-up of a tracking system for the botnet. In the months that followed, cooperation with the Spanish Guardia Civil was established in order to finally redirect the botnet's DNS system. The reaction time of this approach is significantly influenced by legal processes. According to the law in most countries, service and hosting providers cannot allow immediate access to suspicious servers by researchers and security companies that are active in investigations against botnets. The same is true of a change of DNS records. In both cases, law enforcement has to be involved. A more detailed discussion on this topic is provided in the following report on legal aspects in the fight against botnets [2].

Even a single remaining server can leave room for the botnet to recover (and even become more resilient)

Approaches that target peer-to-peer communication infrastructures can be helpful in botnet investigations if, and only if, this type of protocol is used. This limits these

approaches in comparison to others. Nevertheless, the technique is useful for supplementing an ongoing mitigation process, because the communication infrastructure can be partly or fully disabled and thereby take control away from botmasters who are trying to save their bots through migration to another communication protocol. This has already been applied to multiple botnets. In the case of Waledac, for example, the P2P routing was actively disrupted during the takedown efforts,

The approaches of the group of regulatory and social countermeasures aim for a larger scale than the other countermeasures described. By design they are very general, in order to achieve high impact, and their main goal is to contribute to overall security rather than target individual botnets. On the other hand, the reaction time of these approaches is rather slow since, for example, raising users' awareness of security- and malware-related topics is an ongoing process that, to be effective, requires ongoing coverage.

The harmonised creation and adoption of dedicated laws against cybercrime has to be supported by political motivation at an international level. In this context it is important to support efforts to understand the overall threat potential of malware and botnets in order to clarify the current situation and identify the

Laws on cybercrime need global orientation and have to provide flexibility in order to face the fast-paced evolution of cybercrime.

need for efficient countermeasures. The main intention of the dedicated laws on cybercrime is to create a framework that facilitates the targeting of botnets by operations against them. This makes them fundamental to any method, and the design of a legal framework has to aim at providing generality. Because decisions made for legislation are of high relevance and create obligations, their design requires foresight and is more complex than their implementation. Processes connected with the definition of laws have a slow reaction time compared to the evolution of cybercrime, emphasising that these laws have to be created both with sufficient foresight and with the flexibility to adapt. Even more important than laws on cybercrime themselves is the willingness of nation states to cooperate.

In terms of botnets that are globally organised, only global efforts can provide a satisfactory response and achieve significant success in the long run. It is therefore important to extend the scope of activities to the level of global initiatives, or to at least implement interfaces that facilitate cooperation with other international organisations. Nevertheless, creating efficient structures in an EU context is an important step in improving capabilities for fighting botnets and also raises the level of protection. This also allows the scope to be extended to a more global perspective, using experience gained in an EU context.

The latest efforts in this direction at an EU level are the "Proposal for a Directive on attacks against information systems, repealing Framework Decision 2005/222/JHA"

[155] and the “EU Internal Security Strategy in Action” [208]. Both propose that cooperation should be strengthened but that there should be a strong focus on the EU.

Although countermeasures are, by default, not stealthy from the point of view of the botmaster, once they are applied at least the preparation of countermeasures can benefit from coordination as well. For example, the coordination of investigations against a certain botnet can help to reduce the noise level because profiling, for example, is not performed by multiple independent teams but just one, providing the others with a complete package of information. Because of this, stealthiness is not considered in this analysis as a feature for countermeasures.

4.4.2 QUALITY OF RESULTS

To evaluate the effectiveness of botnet countermeasures, a definition of success has to be given first. For this analysis, three measures are considered:

- The usability of the botnet in terms of the botmaster’s ability to access the C&C infrastructure and to command the bots.
- The number of functioning of bots. This specifically targets techniques leading to the disinfection of computer systems.
- The availability of revenue streams for botmasters. As money is the main motivation for cyber criminals, making it difficult for them to extract money from their botnets is also a measure of success.

Disrupting a botnet by disabling its command-and-control infrastructure can be seen as a measure of success. In this case, the botnet becomes unusable to the botmasters. Nevertheless, an important aspect of this approach is that the hosts remain infected. Over time, if anti-malware software is used on hosts, only a proportion of computer systems will get cleaned. A number of hosts will still be infected with malware, influencing the systems’ stability and behaviour. A good example of this situation is Conficker. Although the botnet is considered useless to the botmasters, because of continued sinkholing of C&C domain names, around 5 million hosts infected with Conficker.A+B are identified on a daily basis through their connection attempts made using unique IP addresses. Conficker.B disables various system functions related to security and update mechanisms. This means the remaining hosts may be vulnerable to attacks discovered after the initial infection with Conficker. As the results of a botnet disruption produce an almost immediate effect, this form of success is easier to recognise.

A takedown of C&C infrastructure does not solve the problem of infected hosts.

Evaluation of the success of mitigation approaches should include the cleaning of victimised computers.

When regarding the number of cleaned infections as a measure of success, several issues arise. Approaches that achieve this kind of success mostly work progressively, instead of being the result of a one-time hit, like a botnet takedown. An exception to this assumption is remote disinfection, which may include functionality for reporting the success of malware removal back to a central benign server, similar to some anti-virus products. Alternatively, remote disinfection is affected by factors that drastically constrain the applicability of this approach, as explained later in this analysis. In addition, some systems may suffer from reinfections, or an insufficient patch state that leave the system vulnerable to attacks from other malware. In general, methods for the measurement of botnets provide only limited accuracy, as explained in chapter 3 (Detection and Measurement Techniques) and so make it hard to measure in-depth success.

Good examples of techniques that are very successful at disabling the usability for botmasters are direct takedowns of command-and-control servers, DNS sinkholing and BGP blackholing, because they take control away immediately from the botmasters. All of these techniques target single botnets. If the preliminary investigations are complete in terms of servers or domain names, and if the botnet has no back-up communication channel, the botnet is finally terminated. Apart from successful shutdowns, there are examples of an incomplete takedown enabling the botmasters to regain control. On November 6 2009, the Mega-D/Ozdok spamming botnet was shut down by FireEye, Inc. [144] in an action coordinated by ISPs, registrars and other institutions. Three days later, all spam originating from the botnet stopped. The botnet recovered quickly after the initial takedown and on November 22 reached its previous spamming capabilities.

These events can be looked at using a medical analogy to illustrate the effectiveness of C&C takedowns. By not finishing a course of antibiotics, not only might the bacterial illness not be fully healed (comparable to an incomplete takedown) but the bacteria might recover and become more resistant against future doses of antibiotics (increased resilience as the answer to a takedown attempt).

Even one undetected or ignored, command-and-control server may be enough to enable control of the botnet to be regained, as the whole structure might be reached through this single server, providing the possibility of reconstructing a resilient architecture again. Another problem arises from the usual malware functionality of downloading arbitrary executables to the already infected systems. This enables a compromised computer to be integrated into more than one botnet, or to migrate easily to other botnets. Moreover, as described in section 1.3 (Motivation and Usage of Botnets), pay-per-install is a service offered in this context by botmasters and can be used to recruit machines for other botnets.

Even one undetected or uncovered C&C server may give the botmaster a chance to regain control of the botnet.

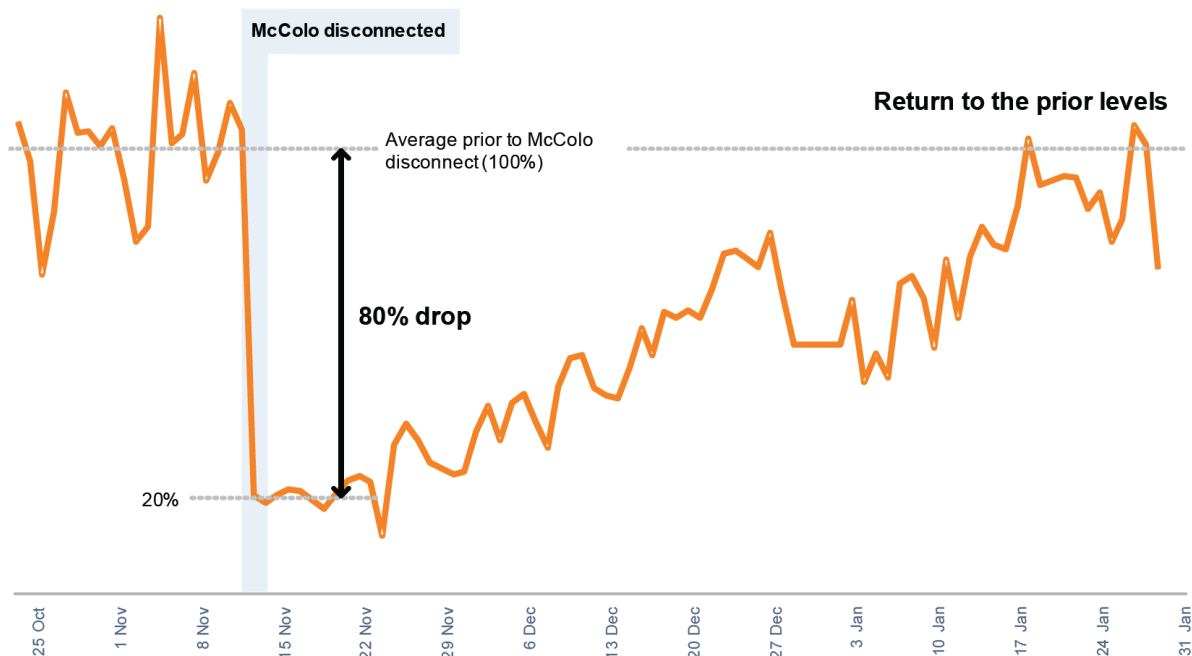


Figure 13: Temporary impact of the shutdown of hosting provider McColo on spam e-mail (Diagram by MessageLabs, Symantec Hosted Services). [209]

In the last few years, apart from takedowns of botnet command-and-control servers, some hosting providers with proven connections with malware activities have been targeted directly. The example most recognised in this context is the takedown of the bullet-proof hosting provider, McColo, in November 2008. After its upstream providers, Global Crossing and Hurricane Electric, disconnected McColo from the Internet, the global volume of spam e-mail fell temporarily by approximately two thirds.

A year later, the average global spam e-mail rate nearly doubled, as compared to the peak before the McColo shutdown

The effect did not last long, however because, in the long run, the responsible botmasters were able to recover from the loss of their infrastructure from the disconnection (cp. figure 13). A year later, the average global spam e-mail rate nearly doubled, as compared to the peak before the McColo shutdown [210]. The main reason for this is that, after the shutdown of one bullet-proof hosting provider, the botmasters extended their command-and-control infrastructures across several hosting providers to ensure the stability of their networks.

This implies that, only longterm actions that are coordinated at international level can produce effective countermeasures. Viewing the actions

Only longterm actions that are coordinated at international level can produce effective countermeasures

from this perspective clarifies the fact that the principle of action and reaction definitely applies to countermeasures against botnets; every action taken leads to a cybercriminal reaction that makes mitigation under the given circumstances harder. Clearly, taking no action is not an option, but the cases examined show the need for a coordinated long-term strategy. Further examples of shutdowns of hosting providers are the cases of Atrivo/InterCage [211], Real Host [212], Troyak [120], and Pricewert/Triple Fiber Network (3FN) [213].

Both Infiltration and remote disinfection target end users' systems with the objective of an automated clean-up. Infiltration is usually achieved by reverse-engineering the command-and-control protocol. Remote disinfection requires knowledge about the infection routines and file systems changes, which can also be acquired through reverse-engineering techniques. Once an intruder targeting a botnet is able to inject its own commands, a crafted executable can be loaded on to the compromised machines, initiating a clean-up, or informing the user about the infection. This technique has to be employed for a period of time, because only a fraction of infected hosts are connected to the botnet at any one moment. If the technique is applied for long enough, and the botmaster is not able to react to this countermeasure, its success rate should increase over time, extending to almost all bots. The robustness of this method is an important concern, as it is almost impossible to guarantee that remote systems will be successfully cleaned. Side effects may appear, e.g. systems might become unstable. This may pose enormous risks if the affected system performs critical tasks. Compromised hosts situated in a hospital [214] or in air traffic control [215] would count as extreme examples.

The manipulation of remote computer systems is prohibited by law in almost all countries, rendering this approach only applicable in theory in the current circumstances

Nevertheless, the manipulation of remote computer systems is prohibited by law in almost all countries, rendering this approach only applicable in theory in the current circumstances. In the recent takedown of the Bredolab botnet, a variation of this approach was applied. Users received a notice about their infection when they restarted their computer after the takedown [204], [216].

Countermeasures that work by influencing botnets' peer-to-peer infrastructure are limited in their effectiveness. The techniques presented aim mostly at progressively hindering more traffic or functionality of the network, until there is complete disruption. They have to be applied constantly over time to have an effect. However, they are a viable complement to other countermeasures and can be used to block the communication channels, for example during a shutdown of C&C servers or domains, as was performed during the Waledac takedown [217]. It is important to note that uncoordinated attempts, using these techniques, may interfere with each other and can result in a change to, or a hardening of, the botnet communication protocol.

Blacklisting and packet filtering can also be considered as auxiliary techniques. They can help to identify infected hosts, but they have no immediate effect on the number of infections, other than to prevent new ones. Nonetheless, these techniques can hinder botmasters in the process of committing crimes against their victims and stop them from extracting money. For example, even if a computer system has been infected, requests may not be accepted by a service provider for financial processing, or the bot may be prevented from connecting to the command-and-control server because malicious contents were identified in the related traffic.

Walled gardens depend heavily on the detection methods used to supplement the process. Because this technique is performed at the network infrastructure level by Internet Service Providers, it can reach actual end-users. If notifications are sent to the users via media other than e-mail or browser redirections, e.g. letters or phone calls, a higher level of trust can be established in the authenticity of the sender. The correct detection of infections has to be ensured, as errors may result in the service being discredited. While notifications are important, and can raise awareness, many end-users are not capable of performing disinfections on their own. Therefore, for full effectiveness, users must be supported in the disinfection process, not just notified. This can be achieved by providing the appropriate tools, extensive information and guidance, and advice on consulting specialised computer services or dedicated services like a telephone help desk, as featured in the German Anti Botnet Initiative [161].

Walled gardens and notification systems are effective because they reach the end-user. User guidance can improve the cleaning rate.

Rating the effectiveness of regulatory and social countermeasures is complicated due to the fact that, while these techniques improve overall security and efficiency, an immediate impact on botnets cannot be measured directly. Dedicated laws on cybercrime are in any case essential, because the framework provided by them is a foundation for any actions taken in fighting criminal activities related to botnets.

An important issue to be addressed is to find a practical balance between privacy and data protection laws on the one hand and the freedom of action enjoyed by law enforcement agencies and on the other hand. Certain conflicts arise from the fact that in terms of botnets, end-users' machines are compromised and used for criminal activities. A practical balance must be found that enables law enforcement to investigate botnets effectively while, at the same time, protecting users' privacy.

Cooperation between different stakeholders is essential for successful botnet mitigation.

It is also important to clarify what is and what is not allowed in each jurisdiction as an enabler of

cross-border co-operation. For more information on the legal issues identified in the context of botnets, see [2].

Some sources indicate the importance of this approach to the effectiveness of stakeholder cooperation. For example, Jeff Williams of Microsoft commented on the coordinated operations against Waledac and the experience gained [217], as follows

“These actions demonstrate how critical the incredible cooperation of stakeholders and experts all around the world is to success. Look for more efforts like these as we work together to take a stand against botnets and make the internet safer and more secure for everyone.”

A press release of the FBI [218], regarding international cooperation during investigations against criminals using the Zeus botnet for financial fraud also underscores the importance of cooperation:

“No one country, no one company, and no one agency can stop cybercrime. The only way to do that is by standing together. For ultimately, we all face the same threat. Together, the FBI and its international partners can and will find better ways to safeguard our systems, minimize these attacks, and stop those who would do us harm.”

And further:

“The multi-agency partnership, including support from Internet security researchers, gave law enforcement the opportunity to gather intelligence about this scheme and significantly disrupt the activities of cyber criminals and money mules who took part in these crimes.”

4.4.3 REQUIRED EFFORT AND RESOURCES

The application of countermeasures is related to various aspects of effort and the need for resources to ensure effective and efficient operations. According to the feature set defined in section 2.2 (Applied Features for Analysis), five major categories can be distinguished: technical, development, and administrative effort, and resources in forms of knowledge and finance. Approaches that are very demanding in respect of these categories are evaluated as follows.

In the case of technical hardware and software requirements, it can be observed that most approaches place only low demands on them. One reason for this is that some are an extension to existing systems, like, for example, when incorporating blacklists for filtering spam e-mail or DNS requests. In general, when considering technical requirements for the application of a technique, the scale on which it is implemented plays an important role. For example, packet filtering at the ISP level can meet an enormous need for computing capacity, especially if real-time, deep-packet inspection of the entire traffic is required. In the case of Europe’s (current) largest provider in terms of bandwidth, Leaseweb, currently more than 700 Gbit/s would have to be handled [219]. Such a large amount of data can be only processed through a massive parallelisation of analysis, if at all possible. Distributing the load by moving the packet analysis to the

Technical requirements for most approaches are low in terms of hardware and software

edges of the network architecture, e.g. to the borders of, or inside, company networks, or even at host level, makes this approach viable. For most of the other approaches, technical requirements are negligible because they build on aggregated information already gathered by earlier detection and measurement steps.

Development effort can have a high impact on the presented approaches. Customised solutions are needed in the case of specialised approaches that target certain botnets or classes of command-and-control infrastructure.

Most approaches require significant development efforts and customisation.

For example, when applying such countermeasures to peer-to-peer-based botnets, the C&C protocol has to be reverse-engineered and re-implemented in a custom client to influence the network, exploiting the properties of the protocol. Depending on the case, infiltrating a botnet in order to impersonate the botmaster can require be related to even more development effort. In addition to understanding the command-and-control mechanism, flaws in the mechanism that can be exploited have to be identified. When considering the use of automated disinfection, significant testing of the disinfection routines is involved. As well as these specialised approaches, the implementation of the walled garden concept requires extensive planning and testing, as a high degree of automation and a low false-positive rate are required. This also applies to the development of guidelines for telephone support from a centralised user help desk.

Another important dependency for the effective application of the presented approaches is the knowledge and expertise needed. Overall, the application of countermeasures requires less technical expertise than the operation of detection and measurement techniques. The highest level of skill is required for both specialised approaches mentioned in the preceding paragraph.

Combined expert knowledge in the technical, legal and social sector is a key requirement for botnet countermeasures.

The reverse-engineering of compiled code and network protocols, that are almost certainly encrypted, requires in-depth knowledge of system-level architecture. In contrast to the purely technical expertise necessary for the successful employment of these methods, other approaches require different kinds of knowledge and experience. For example, knowledge of administrative processes and legal options is helpful when taking down C&C servers and domain names that are used for malicious purposes. The quality of service and advice offered by help desks does not only depend on the technical understanding of the telephone operatives but also includes their social and didactic skills.

The effort connected with planning and administration has a dramatic impact on several of the approaches.:

- The implementation of an approach can depend heavily on the target environment into which it is integrated. In this context, compliance has to be ensured with existing frameworks, like internal security and privacy policies, and the law.
- The documentation of processes and interfaces is important for ensuring that the operation is both stable and flexible .
- The tracking and management of individual cases have to be carried out, for example when working with walled gardens and help desks. This is necessary for improving reaction times, as infections caused by related malicious software can usually be handled using comparable methods.
- Staff management, including continuous further training, may be necessary. This is needed in the case of help desks, for example, so that the effectiveness of helping users can be maintained.

Compliance has to be ensured with existing frameworks, like internal security and privacy policies, and the law

The creation of dedicated laws against cybercrime is a complex process that involves various parties and advisors. Communications and actions between participants have to be coordinated in the case of large synchronised actions against botnets, such as takedowns.

Examples of interactions influenced by administrative processes are legitimate requests from law enforcement to service providers to de-register domain names or power off servers. Because botmasters want reliable command-and-control architectures, this can easily involve hosting providers from a range of different countries, which affect and complicate the proceedings. To overcome these challenges, communication and information exchange at an international level have to be further improved. A key factor is structured and secure information exchange. As outlined in section 4.3 (Initiatives and Institutions), appropriate standards are currently in development and will hopefully make the processes easier.

International cooperation and information sharing have to be further improved.

As for financial resources, the duration and scope of application of approaches is a major factor, primarily for socially oriented countermeasures. For example, the launching of campaigns and the placement of advertisements aimed at raising users' awareness of the importance of securing their

Socially-based countermeasures have the highest financial costs among the presented approaches.

own computer systems have to be broad based, in order to reach as many as possible, and over a significant period of time.

The operation of a dedicated incident help desk demands various financial efforts, including, for example, continuous training of telephone operatives on new threats, infrastructure maintenance, incident case management or sending letters. Technical countermeasures may also require continuous efforts. In the mitigation of Conficker, up to 500 domains are registered on a daily basis to stop infected hosts from communicating with possible command-and-control servers. In these cases, cooperation with domain registrars is important for handling both the cost and organisation of the processes involved.

4.4.4 LIMITATIONS

To begin with, the most significant current limitations on countering the botnet threat are legal ones, which influence the applicability and reaction time of various approaches. The inclusion of law enforcement as early as possible in the process of investigating botnets and planning countermeasures can drastically improve the outcome of actions and their response time limits.

The complexity that arises from the differences in laws at a national level complicates planning and coordination

Botnets are a global problem and have to be seen from this viewpoint. The complexity that arises from the differences in laws at a national level complicates planning and coordination, because unforeseen delays, due to the availability of warrants, have an impact on reaction time. This is relevant in cases where botnets are changing their command-and-control infrastructure regularly, for example by migrating to servers or hosting providers in other countries, thereby evading practically every seizure attempt by minimising their presence in one place. The existence of bullet-proof hosting providers further benefits this behaviour.

As regards the impact of laws in the context of botnet countermeasures, a general tension is discernible between users' privacy and protection on the one hand and improvement of the overall security level on the other. In the case of infiltration and remote disinfection, it is obvious that these cannot be performed without breaking the law, because data on a remote system is modified without the explicit permission of the owner. Performing countermeasures at a peer-to-peer level may also be affected by legal considerations, because it involves interaction with remote machines and the manipulation of data or whole networks. Another approach in question is the application of sinkholing to a malicious domain name. Incoming data will probably contain sensitive data that has to be treated accordingly. Using this data for contacting the victims directly and informing them about the infection of their system can cause distress. Such reporting is often misinterpreted, ending in disbelief, accusations of unauthorised access being used to obtain data from the victims' computers, or no

reaction at all. The same applies to walled gardens and the associated messages. The objective of user awareness campaigns over the last few years has been to sensitise users to potential fraudulent messages in emails and websites. The level of suspicion created by these approaches may produce negative outcomes.

Cooperation between representatives of different stakeholder groups may be affected, or limited, by levels of confidentiality, e.g. regarding their customer data or details restricted by internal policies. Confidentiality is important in any case, in order to protect the collected information from those criminals intending to infiltrate working groups. The topographic distribution of stakeholders may also introduce delays in progressing activities as a result of timezone mismatches .

4.5 CONCLUSION

In this chapter, a selection of current countermeasures to the botnet threat has been discussed. The presentation of countermeasures is divided into two groups, one focusing on technically-oriented methods, the other on regulatory and social countermeasures.

Many of the technical countermeasures build on detection and measurement approaches that were discussed in chapter 3 (Detection and Measurement Techniques). The analysis shows that effective methods exist and that these are already being applied against botnets. It has been further shown that success is limited by different factors, including administrative and development efforts, the need for more global coordination during actions against certain botnets, and legal considerations affecting the reaction time or even application of countermeasures. Whilst a reduction in development effort is unlikely, administrative efforts may be reduced, partially through the use of standards for communication and information exchange. Appropriate standards (cp. section 4.2.4 Enhance Cooperation between Stakeholders) have already been defined and are ready to be implemented. Legal considerations are covered by the design of cybercrime-dedicated laws.

Existing legal frameworks have a drastic impact on the applicability and performance of some technical countermeasures. Even where the law permits the use of a methodology, the reaction time may increase due to dependencies on administrative processes, leaving enough time for botmasters to evade countermeasures simply by migrating their command-and-control infrastructure.

Again, it is important to note that the disruption of command-and-control infrastructure, for example through a takedown of C&C servers, may disable communication between server and bots, rendering the botnet useless to the botmaster. However, this does not solve the problem of infected machines, which remain partially unstable, and have disabled defence mechanisms such as AV software or system updates. Interruption may also cause the last command to be repeated, as long as the infected computer is running and is connected to the network, for example a DDoS command with a

separate command for stopping the attack. In general, the success of C&C disruption can be measured via indirect indicators like spam e-mail output (with the abovementioned limitations on their accuracy). Measuring the success of disinfections is only possible using relative numbers because, as outlined in chapter 3, there is no generally reliable way of measuring botnet populations. In any case, the cleaning of users' computer systems has more significance, because it reduces the attack surface for botnets and malicious activities in general.

The results of the analysis of countermeasures are summarised in Table 2. The overview table should be interpreted with care, as the colours indicate only rough trends, resulting from the comparisons made between approaches according to feature set. Technical and social countermeasures in particular are listed in the same table, but should not be compared directly. Comparisons across columns should also be made with caution, because the features themselves have different weights in the overall rating and even have different weights for distinct approaches.

	Packet Filtering on Network and Application Level	Direct Takedown of C&C Server (Sinkholing)	DNS-based Countermeasures	BGP Blackholing	Distribution of Fake Credentials	Blacklisting	User Awareness Raising	Dedicated Laws on Cybercrime	Infiltration and Remote Disinfection	Peer-to-Peer Counter Measures	Port 25 Blocking	Walled Gardens	Enhance Cooperation between Stakeholders	Central Incident Helpdesks	Dedicated Training
Generality															
Flexibility															
Reaction time															
Coverage and Success Rate															
Durability															
Measurable Effect															
Scope of considered Data and information															
(Technical) robustness															
Technical effort															
Development effort															
Administrative effort															
Required expert knowledge															
Cost factors															
Restricting laws															
Dependency on third parties															
Lack of access to data, systems or infrastructure															
Topographic restrictions															

Interpretation	
	factor with strong negative influence
	notable negative influence
	average influence / not measurable
	notable positive influence
	factor with strong positive influence

Table 2: Evaluation of countermeasures.

5 RECOMMENDATIONS FOR GOOD PRACTICE

In this chapter, recommendations are made on good practice when fighting the botnet threat. The recommendations build on the techniques analysed for detection, measurement, and countermeasures against botnets. First, an integrated approach is presented, involving recommendations for all affected stakeholder groups. Following this, specific recommendations for stakeholder groups are summarised.

5.1 ASSUMED THREAT PICTURE

The threat picture is assumed as follows. Malware and botnets are recognised as a global problem and therefore reside in a complex system with many dependencies. Millions of computer systems are infected with often multiple types of malware. These computer systems are organised in several hundreds of different botnets. For the average user, the identification of an infection is assumed to happen reactively. This means that malware does not come to the user's attention until it is experienced personally. The reaction time for malware detection, even with anti-malware software installed, is sometimes so long that infections are likely to occur because of the failure to install critical or recent product updates. Malware starts communicating with its command-and-control servers almost immediately, which means that breaches of important information like credentials and identities may happen instantly. This information is abused by botmasters in different ways, mainly for financial gain. Botmasters use various mechanisms for anonymization and are in general hard to identify.

5.2 INTEGRATED APPROACH AGAINST THE BOTNET THREAT

Botnets are a serious threat that can only be handled through cooperation and the pooled efforts of all affected stakeholders. In this section, an approach to minimising the botnet threat is proposed (cp. figure 14). It is split into three categories:

- Mitigation of existing botnets and infections.
- Preventive measures for aggravating the acquisition of new bots and growth of botnets.
- Approaches that target botnet usability, as seen from the botherders' perspective.

Because enhanced cooperation is fundamental to all areas of mitigation, it has not been explicitly shown in the diagram. All recommendations regarding botnets are aimed at the status quo.



Figure 14: Integrated approach against the botnet threat.

5.2.1 MITIGATE EXISTING BOTNETS

On the technical side, a two-fold approach is suggested to reduce the threat of existing botnets. According to the success measures presented in section 4.4 (Analysis of Countermeasures) and metrics defined in section 1.2.3 (Attack Potential and Threat Characterisation), both infected systems and C&C infrastructures, together with their responsible individuals, have to be targeted.

Mitigate existing botnets

Reduce number of infected systems

Fight C&C infrastructure

Reliable method for the detection of infections

Analysis of C&C infrastructures

Host-level anti-MW software

Netw.-level ISPs

Information sharing, tracking of botnets

Cleaning of systems

Takedown of C&C and arrest of botmasters

ARREST CRIMINALS BEHIND BOTNETS

In 2010, there were several successful coordinated actions against major botnets, e.g. the cases of the Mariposa botnet, leading to 4 arrests, the Bredolab botnet shutdown, that included 143 C&C servers, or the identification of a crime ring consisting of almost 100 individuals in total, connected with a Zeus botnet that was used for organised financial fraud and extracted 70\$ million from its victims.

It is important to continue these efforts and disable botnets quickly, arrest their operators and also ensure that botnet shutdowns are final. If a botnet is reactivated, it is likely that it will metastasise and harden its infrastructure, e.g. by migrating C&C servers to bullet-proof hosters or by using highly dynamic C&C mechanisms, making future shutdown efforts much more difficult.

IMPROVE MALWARE ANALYSIS TOOLS

To handle the evolution of malware, the efficiency of malware analysis has to be increased. The complexity of malware specimens is evolving rapidly. A higher throughput in automated analysis has to be achieved and, at the same time, to enable the prioritisation of targets, the analysis has to become more precise with regard to the threat potential and communication mechanisms involved. It is also essential that the results are provided to the relevant working groups, which involves international-level cooperation in the fight against botnets.

ONGOING INFORMATION SHARING

Tools and frameworks for sharing information, intelligence and incident data have to be established as proposed, for example, by EC COM(2010) 673 [208]. Effective botnet mitigation involves different stakeholder groups and is performed across operational boundaries. The installation of one or more reputable information-sharing frameworks could bridge the existing gaps and support the investigations. In this context, the pool of trusted people has to be widened carefully. For example, more inclusion of researchers working in these areas can help bundle resources and avoid instances like the parallel investigations of the Storm botnet.

Overall, this would integrate research more effectively into the investigation process. The fact that exclusive information is often seen as a business asset, has to be addressed if cybercrime is to be fought effectively. This could be achieved with, for example, financial incentives to researchers in combination with confidentiality certificates, which that ensure credit for the research results goes to those researchers responsible.

HARMONISATION OF LAWS AGAINST CYBERCRIME AT AN INTERNATIONAL LEVEL

At an international level, challenges originate from the lack of harmonisation of laws, penalties and crime definition and, in addition, a lack of guidance on the interpretation of legislation in different jurisdictions. This acts as a disincentive to defensive measures which is examined in more detail in [2]. The harmonisation of laws on cybercrime at a European level is provided by the EU Directive 2005/222/JHA [155], and further measures are recommended in the European Commission's communication EC COM(2010) 673 [208]. To create an efficient framework within which law enforcement can operate, swift implementation of these recommendations at a national level is desirable. Legal investigations have to take place across country borders and governments should do their best to support global cooperation.

REDUCTION OF EXISTING INFECTIONS

While efforts against botmasters and their C&C infrastructures continue, the number of infected systems has to be reduced. This requires an understanding of responsibilities with regard to botnets. On the one hand, operating an infected computer system exposes the owners to various risks that can lead to personal damage. On the other, the infected systems also pose risks to other users. A general understanding of this situation and the civic responsibility it implies, has to be conveyed to owners of computer systems. While securing a computer system is a non-trivial task, solutions to raising the security level are widely available, many of them even free of charge. The same applies to guidance on security topics and possible precautions. Infections are not easy to detect, and so users must be supported in doing so.

ISP-BASED DETECTION AND NOTIFICATION

Internet Service Providers are clearly not responsible for infections on their customers' systems, but since they provide network access to them they are in an excellent position to identify infections. This is also true of hosting providers who own servers that are used for malicious purposes, for example for botnet command-and-control. Government incentives or regulations may be created to integrate ISPs more fully in the mitigation process, e.g. through funding for the operation of notification services for their customers or the implementation of walled gardens. These techniques are already deployed in some countries such as in Germany, with the Anti-Botnet Initiative [161], UK with ISP Virgin Media, using a notification system based on postal letters [220], Japan with the Cyber Clean Center [163], Australia with their Australian Internet Security Initiative or US ISP Comcast with their Constant Guard Security Program

[221]. These approaches are still in the early stages of their application and their development and success should be kept under close review because they may serve as good examples for related projects in other Member States.

5.2.2 PREVENT NEW INFECTIONS

Prevent new infections

It is important, not only to cover reactive measures, but to prevent new infections, in order to both protect users and weaken the position of botmasters.

Slow down botnet spreading through early detection

Protect systems
User awareness

Analysis of structures and patterns

Identification of vulnerabilities

Identification of C&C and comm. patterns

Exploit discovery and information sharing

Application of preventive measures

Responsible operation, patching of systems

SLOW DOWN BOTNET SPREADING

To prevent the expansion of botnets, it is necessary to be able to detect botnet spreading and communication as early as possible. This enables communication with control entities to be blocked, effectively disrupting the bootstrapping process and hindering the download of further malware executables or commands. To make this possible, early warning is essential and so the availability of sensors has to be extended. New detection methods that complement the approaches presented in chapter 3 (Detection and Measurement Techniques)

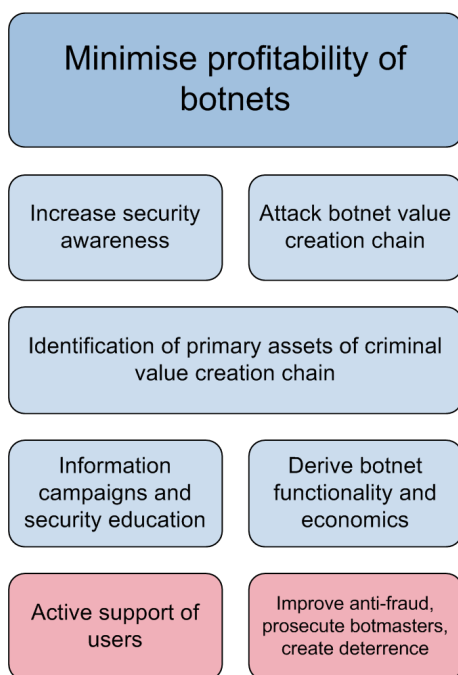
have to be invented and deployed. For example, there are more sophisticated honeypots that can detect a wider range of attacks, or are able to actively harvest websites for exploits and malware. As previously described, tools for faster analysis of specific malware have to be created. Of especial interest are more automated tools for fast reverse-engineering and interpretation of the results collected. Early detection and identification also benefit the mitigation efforts described in the previous section. This leads to faster cycles of investigation, disrupting the operation of botnets and increasing botmasters' efforts.

VULNERABILITY MANAGEMENT AND SYSTEM PROTECTION

Because exploiting vulnerabilities in software and systems is a common vector for the spreading of malware, this circumstance has to be addressed more intensively by vendors and users alike. To avoid vulnerabilities in the first place, initiatives promoting secure software engineering can help to improve the quality of software development and reduce the number of software errors [222]. Apart from software audits, network penetration tests may indicate critical flaws in the design and application of software. Additionally, many vendors already offer financial rewards for identification of vulnerabilities in their software, a potentially useful incentive for professional vulnerability researchers to disclose information to them. While the regular provision of patches underpins the secure operation of computer systems, maintenance and awareness on the side of users is also important. Companies that operate outdated or

unpatched software need to improve their software update and patch management practices.

5.2.3 MINIMISE PROFITABILITY OF BOTNETS AND CYBERCRIME



The third category of recommendations is less technically oriented and aims at reducing the profitability of botnets and cybercrime. At the current time, the economic patterns of the cybercrime business, with malware developers and botnet operators separated, make it very hard to prevent the evolution of malware. The developers are able to hide in the background, providing the tools that are then used for generating financial gains through malicious activities like identity theft and financial fraud, distribution of spam e-mails or denial of service attacks carried out by botmasters.

ENGAGE MALWARE VALUE CREATION SCHEMES

To fully take on the threat posed by botnets, it is not only bots and command-and-control infrastructures that have to be attacked. A particular problem is that botmasters have a choice of various means to use as C&C. They can switch easily from one technique

to another. The most effective way to attack them is to target different levels in their processes and concentrate on those elements that do not offer many backup strategies. This suggests this could serve as a supporting strategy when targeting the economic processes used by the criminals and hence their revenue streams. Their economic motivation can be lowered if their revenue streams are cut off and the banking accounts they use frozen. A key element in this context is to address the frequent use of irreversible money transfers provided by financial services like Western Union or virtual currencies such as WebMoney or Liberty Reserve, which are known to have been used in money transactions connected to cybercrime [223].

SECURITY AWARENESS

Raising security awareness on a broad scale can help reduce the number of potential targets for botnet operators. Only if users understand the threats they are exposed to and the mechanisms used to generate profits, can a fundamental change be achieved. The education of end-users should cover various security-related topics. Cautious use of the Internet and handling of personal information can help to reduce the risk of malware infections, abuse and fraud. For example, the recruitment strategy of botmasters in money laundering should be explained. Offers, typically received via email, of earning

money easily from home with minimum effort are clearly suspect and have to be recognised as such [224].

RESPONSIBLE HANDLING OF INFECTIONS

When an infection is detected, handling it responsibly is important, as most malware integrates itself deep into the computer system. This can lead to situations which the average computer user will often underestimate and also find difficult to handle, and so they should seek professional support if unsure about the procedure. In this context, financial funding allocated for these purposes could be considered. This could also be extended to offer financial incentives to end-users via their ISPs, if they keep their system clean for a certain amount of time, e.g. on an annual basis.

IMPROVEMENT OF ANTI-FRAUD

The second direction which is proposed for reducing the profitability of botnets aims at improvements in anti-fraud mechanisms and the targeting the criminals behind botnets. The rapid developments of electronic payment, with all their advantages, have also introduced certain risks to the integrity of customers' accounts. To protect users against fraud even better, financial institutions can strengthen their cooperation with security-oriented service providers. An example for a new protection scheme might be the use of real-time blocklists that show infected machines and the consequent interception of banking operations between the hosts and banking services.

PROSECUTION AND DETERRENCE

Prosecution of cyber criminals is very important because it stops the miscreant individuals behind botnets simply moving on to their next botnet after losing their command-and-control position. Prosecution also has to come down harder on criminal suspects on the customer side of botnets, e.g. those who rent botnets or use botnet services for their criminal purposes.

With the presence of various versions of bot source code easily available in the Internet, it is important to discourage potential malware users and remind them of the illegality of using this malware to compromise foreign computer systems. Prosecutions help to illustrate that using malware is not a minor criminal act but a serious contravention of multiple laws, and may deter interested parties from starting their own botnets.

5.3 RECOMMENDATIONS FOR STAKEHOLDER GROUPS

After presenting a selection of combined recommendations for good practice against botnets, this section features recommendations targeted at key stakeholder groups.

5.3.1 RECOMMENDATIONS FOR REGULATORS AND LAW ENFORCEMENT

Governments and regulators in general are in a unique position to define frameworks in which actions against botnets take place. In the case of botnet mitigation, time is a critical factor in effective and successful operations. Currently, investigations and mitigation attempts are heavily impacted by legal aspects, not only about what is permitted but also about the time taken to implement measures. In order to improve actions against botnets, roles, responsibilities, and the rights of involved stakeholder groups have to be defined. In the first place, this includes the harmonisation of laws and legal processes related to cybercrime at an international level. A major benefit of these efforts could be the simplification of cross-border cooperation across borders and between stakeholder groups, including bringing academia and industry into investigations. As a minimum, clear guidance should be given on differences in legislation across borders in order to facilitate botnet defence activities in multiple jurisdictions.

DEFINITION OF RELEVANT INFORMATION

Information about incidents and the results of investigations into botnets are widely distributed when looked at from a global perspective. Furthermore, information often contains more details than are relevant for involved parties when shared. Alternatively, information may contain details whose treatment is restricted by data protection law, such as IP addresses in some EU countries. Occasionally, work is needlessly repeated by independent groups that are not aware of each other's activities: this may even result in investigations being obstructed.

Intelligent information sharing can help to resolve these issues. Law enforcement should explore the amount of available information and define its needs in order to receive the most suitable results from cooperating partners. Subsequently, after receiving and using the information, law enforcement agencies should give their source of information feedback on the usefulness, or share results where possible in the interest of confidentiality. Two-way communication can enable appreciation of the contributed information to be shown and further information sharing encouraged.

Because the willingness to share information is mainly influenced by the trust that exists between the parties involved and legal issues (again) regarding privacy and data protection, regulators and governments are in a position to enable high-level information-sharing frameworks that are able to deal with the above-mentioned aspects.

DISCUSSION OF POWERS AND CAPABILITIES FOR STAKEHOLDERS

Law enforcement agencies and institutions like CERTs, ISPs, abuse teams, or other stakeholders from the public and private sectors, should receive further clarification on the capabilities required for fighting botnets. Activities like monitoring, studying and mitigating botnets have to be aligned with laws on privacy and data protection. These

laws mostly originate from the pre-botnet era and have to be reviewed and reinterpreted in the light of this new threat. Regulators should initiate and lead the discussion on how to redesign these laws in the context of botnet mitigation.

ENISA could also play a role in providing EU-wide information on legally permissible activities in the fight against botnets and in supporting dialogue and consensus on the practical interpretation of laws in the member states. This is particularly important, where conflicts exist between data protection and legal or commercial requirements when protecting customers from security threats.

CREATION AND PUBLICATION OF CENTRAL CONTACT POINTS IN ORGANISATIONS

Botmasters benefit from the global availability of Internet services and can move their command-and-control entities quickly, achieving high reliability for their malicious networks and hiding behind anonymity services, operating from anywhere. Because the botnet threat is globally distributed, and in order to reduce reaction time, it is important that central contact points in organisations are available for contact immediately at any time.

5.3.2 RECOMMENDATIONS FOR INTERNET SERVICE PROVIDERS

Internet Service Providers enable Internet access for their customers. This puts them in the position where they control all the traffic flowing through them. At first glance, this seems like an ideal position for the detection of botnets. But of course, their customers are protected by privacy laws and, for example, in some jurisdictions, analysis of traffic content is only allowed in cases where there is concrete suspicion. In addition, the business segment of Internet Service Providers is driven by the same strong competition for customers as any other segment.

As a result, the individual reputation of service providers has an impact on their customers' choices, while considerations regarding their being a messenger for bad news (in this case an infection of the customer's computer system) need to be taken into account. Nevertheless, ISPs are in a unique position vis-à-vis botnets. This conclusion is supported by the OECD Working Paper on the role of Internet Service Providers in botnet mitigation [225]. A key finding of the study is that, globally, the networks of 50 ISPs account for half of all bots of all infected machines worldwide.

IDENTIFICATION AND NOTIFICATION OF CUSTOMERS WITH MALICIOUS HOSTS

To support mitigation, a two-stage approach for the identification and assessment of infected hosts can be implemented in compliance with privacy criteria such as privacy of communication. By implementing passive detection techniques like honeypots and spam traps, ISPs are able to identify hosts with obvious malicious behaviour. When an ISP's sensors are attacked, the ISP is one communication endpoint and the attacking host the other. Because the sensor is part of the communication, and therefore included in privacy considerations, this enables the traffic and application of further actions to be evaluated. In a second step, they can correlate the collected data with

their own IP address assignments and check if these addresses belong to one of their customers. In this case, they can notify their customer. ISPs should still define in their EULA that they are authorised to send their customers notifications, in case they have detected a possible infection. This approach is already used by some ISPs and is recommended for implementation as a default. As attack and defence trends depend on each other, it can still be assumed that botnet developers will react to these measures. Nevertheless, if this approach proves effective, it may lead to a reduction in side effects like spam and the automated spreading attempts of malware.

CREATION OF INCENTIVES FOR CUSTOMER NOTIFICATIONS

Deploying this technique means additional effort for the ISPs without giving them any notable benefits. Because service providers are faced with market pressure, they should be supported through incentives or funding in order to implement this approach to botnet mitigation.

5.3.3 RECOMMENDATIONS FOR RESEARCHERS

The understanding of malware and botnets is essential for the development of effective countermeasures. Therefore, ongoing research is important for the fight against botnets. The further development of analysis approaches is a foundation for the successful investigation of botnets. Detection approaches, leading to the identification and assessment of infections, should be continued with increased efforts. To enable this, the work of researchers should be supported with funding and incentives.

SOCIAL RESPONSIBILITY

Science should always be complemented by a sense of responsibility for society. It is recommended that researchers who are active in the field of botnets should play an active role in increasing cross-sector cooperation and information sharing. For example, researchers should try to support law enforcement when useful results have been obtained.

INFORMATION MANAGEMENT AND SHARING

It is a general observed that information is increasingly seen as a business asset and therefore treated as an economic benefit. Instead of encouraging a market which allows the selling of information, incentives should be created to support researchers, valuing their work and motivating their efforts in order to achieve results that can help mitigate the botnet threat more efficiently. One way of realising this is to include researchers in the existing and upcoming information-sharing frameworks for botnet defence and to provide funding for projects that fit the context of the mitigation of certain botnets. Scientific investigations into recent malware mechanisms have value for both researchers and law enforcement. New techniques related to the analysis and

detection of botnets are constantly needed, especially those that can also be effective in helping the mitigation process.

PUBLICATION OF RESULTS

When publishing the results of their work, researchers should concentrate less on approaches that aim purely at measurement of a botnet's size. If a botnet does not provide a mechanism for the unique identification of the participating bots that can be used, any measurements are likely to be very inaccurate, due to the effects of the characteristics of Internet addressing mechanisms. Even unique bot identifiers may be inaccurate, for example due to collisions created by the generation algorithm, or due to changes made over time to the host system's setup, or simply through re-infection. Because these identifiers are provided through means created by the malware creator, they cannot be trusted by default.

In any event, the methodology applied to measurement should be explained transparently and in detail, with analysis of uncertainty to enable other researchers to validate the results.

RESEARCH FOCUS

The research focus should shift markedly away from measuring the size of botnets towards evaluating the impact of specific botnets and malware that pose significant threats to society (see section 1.2.3 Attack Potential and Threat Characterisation). A thorough analysis of malicious functionality is essential, especially the development of generic approaches for fast identification, categorisation, and the handling of important aspects, including:

- C&C infrastructure and protocol.
- Weaknesses and attack vectors in the malware and C&C infrastructure usable for mitigation.
- Spreading potential with respect to server-side mechanisms.
- Complexity of applied design concepts like cryptography, hiding techniques, code obfuscation in the malware.

As a general consideration, research should also focus on techniques which can be implemented in large-scale operational environments subject to typical cost constraints.

5.3.4 RECOMMENDATIONS FOR END-USERS AND COMPANIES

The majority of bots that are active in botnets consist of computer systems owned by end-users. According to Eurostat data for 2010, 22% of Internet users in the European Union encountered infections from malware within the last 12 months [51]. A commonly held view is that end-users are responsible for their equipment and should therefore be concerned about possible malware infections on their systems. Whilst most published

figures on the size of botnets may be overestimates, due to extrapolation, the true numbers will still remain large enough for it to be concluded that the task of removing malicious software from computers would be asking too much of the average user.

CREATION OF AWARENESS FOR SECURITY RESPONSIBILITY

To reduce the number of potential victims of botmasters, user awareness for security responsibility has to be increased. Users have to fully understand that they will become both a victim and, unconsciously, an accomplice to a crime when their computer is unintentionally infected with malware. From an ethical and social point of view, protecting a computer from malware is a civic duty, which may include patching the system and running preventive tools like Anti-Malware solutions and desktop firewalls to control network connections. According to Eurostat, 60% of Internet users in the European Union are using security software for the protection of their computer or data malware [51]. This number must be increased. IT security responsibility should be addressed with public campaigns comparable to those against drink-driving or smoking.

SUPPORT END-USERS IN HANDLING INCIDENTS

In the EC Communication COM(2010) 673, empowering and protecting citizens is one action proposed to combat cybercrime [208] (objective 3, action 2). This Communication proposes creating contact points for easy reporting of cybercrime incidents in order to support prosecutions when borderers are arrested. Users should be encouraged to take this step if they encounter an incident. It is further recommended that end-users be encouraged to seek support to help clean their systems. The first attempts to provide dedicated incident help desks and centrally-provided information about possible precautions have already emerged, e.g. the German Anti-Botnet Initiative [161]. With the growing availability of such services, users will still be responsible for accepting the help offered and for more actively caring about the security of their data and systems.

CONTINUATION OF SECURITY EFFORTS IN COMPANIES

Nowadays, every company's processes are driven by significant amounts of IT. According to Eurostat in 2009, only 5% of enterprises in the European Union faced the destruction or corruption of data due to a malicious software infection or unauthorised access [51]. But botnets and malware still pose a threat to the economy. Rick Wesson of the Conficker Working Group stated that many of the Fortune 1000 companies were hit by infections of the Conficker Worm [226]. Looking back, Conficker did not in the end cause widespread damage, but the situation could quite easily have been much more severe.

Security breaches not only result in the loss of operability or money, but the publication of incidents can also have a negative impact on the reputation of a well-known company. To address this issue, it is advisable to have in place, and follow, security

policies adapted to the size and specific needs of an organisation. Training employees in order to raise their IT competence and security awareness is a further measure that will reduce the risk of infections and avoid the resulting costs of re-establishing the computer systems' integrity. This not only raises the level of security in a company but also benefits the users on a personal level.

6 FUTURE TRENDS

This chapter provides a short overview of predictions for possible future trends in botnets. For the most part, these predictions are derived from observations of recent developments and input from contributing experts.

FURTHER COMMERCIALISATION OF MALWARE INDUSTRY

It is very likely that the existing economic patterns in the business of malicious software will become even clearer. The latest example of this is the assumed takeover of the ZeuS banking Trojan construction kit by the competing malware product SpyEye [227]. According to the article, one reason for this merger might be the recent arrests of botmasters who used the ZeuS Trojan for financial gain to the tune of around several million Euros. The sale of ZeuS could therefore be the ZeuS malware author's attempt to cover his tracks as a precaution against being arrested. The rivalry between these two construction kits has been well documented. Before the merger, recent versions of SpyEye integrated a feature that would perform a check on the presence of a ZeuS infection in a newly-compromised system, remove this Trojan and then replace it with itself [228]. It is assumed from these observations that the malware economy will increasingly adopt the principles of a market economy.

EMERGING BOTNET USE

Economic models of botnet renting and reselling through pay-per-install services may also become even more prevalent. These allow the rapid creation of new botnets that are produced by larger botnets. The new botnets are then independent from the botnet responsible for their initial infection. An advantage of this approach to botmasters is that the risk of detection can be further reduced. Noisy scanning and spreading behaviour can be avoided, which benefits the operation of smaller and specialised botnets.

The availability of bots for sale in an open market opens the botnet market to technically less versatile botmasters. It may also serve as an entry point for parties with political interests, using botnets as an instrument of political influence or terrorism.

In the last few years there has been a trend for botnets to be used in a political context. Multiple attacks were observed against governmental websites and national economic interests. Prominent examples are the attacks in 2007 against Estonia [229] [41], in 2008 against Georgia [230], or in 2009 against South Korea [184]. In the case of South Korea, the victimised hosts used to perform the attack were also instructed to download another malware binary. This additional binary was programmed to overwrite the hard drive of the infected host with the message "memory of independence day", followed by an endless loop of the character "U", which has a binary equivalent of "01010101", in order to render the infected PC useless [184].

In these politically motivated attacks, the usage of botnets has to be clearly separated

from the phenomenon of hacktivism [231]. Generally in hacktivism, sympathisers voluntarily contribute their own resources to actions. Recent examples of this phenomenon are the attacks organised by Anonymous in the course of the WikiLeaks case. For instance, financial institutions which stopped accepting money transfers to WikiLeaks were targeted in coordinated attacks with DDoS tools like Low Orbit Ion Cannon. It should be noted however that although they have characteristics in common, these do not match the definition of a botnet since participation is entirely voluntary [232].

MALWARE AND BOTNETS ON MOBILE PHONES

A prediction related to the growing availability of mobile Internet access concerns the increasing probability that smartphones will be compromised on a large scale. Smartphones are becoming attractive to criminals because of the devices' increasing computing capacity and ability to connect to the Internet. Some reports on the targeting of phones by malicious software appeared already in 2001 [233], [234]. Examples of the automatic spreading of malware between mobile phones appeared in 2009 [235]. The spreading of the worm was supported by access to personal contact data stored on infected phones. However, user interaction was still needed to enable this worm to spread.

Among other factors, the following make mobile devices interesting for malware developers and botmasters:

- Users have a higher level of trust of messages that originate from people they have personal relationships with. This is a fact that was already observed earlier with the spreading through social networks of e-mail worms and malware and may make it easier for malware to spread.
- Although mobile devices provide less bandwidth and less reliable connectivity than regular computer systems, they are gradually assuming the role of a personal information system. This makes them attractive for information and identity theft and fraud.
- Apart from telephony, Smartphones usually have multiple network interfaces such as WiFi or Bluetooth. These are additional potential spreading vectors.
- The deployment of updates and patches on mobile devices is not as straightforward as the experience users have with their computers. This may result in less well-patched mobile devices and increase the durability of vulnerabilities and exploits.
- Malware detection and defence mechanisms for mobile devices are not as widely deployed as equivalent solutions for computers. This is a problem that other types of embedded devices with Internet connectivity may face sooner or later.

More information on smartphone security can be obtained from a report published by ENISA [236].

IPv6

According to the Number Resource Organisation (NRO), in October 2010 fewer than five percent of the possible IPv4 addresses remained unallocated. This is interpreted as a warning sign that the address space of IPv4 will be exhausted in the near future [237]. With the predicted large-scale adoption of the new Internet Protocol IPv6, this technology will also become more and more important for cyber criminals. Scenarios like covert channels for command-and-control communication can be expected [238], [239], which, for example, exploit the translation of IP addresses between version 6 and 4 to bypass security measures. The protocol stack of IPv6 also provides new vectors for protocol attacks that could be used in DoS and DDoS [240], [241]. Although IPv6 introduces IPsec as a mandatory feature to increase the security of communication, this does not affect the application level on which many attacks currently used for the spreading of malware are performed. The future of Network Address Translation (NAT) for IPv6 networks is still under discussion. NAT, which was intended to be a tool for coping with the depletion of IPv4 address space, provided a basic side-effect security for end-users against earlier common attacks based on hosts being directly targeted via IP addresses. A benefit of this introduction is that randomly scanning the IP address space for possible victims becomes no longer feasible due to the size of the address space. Yet approaches for the discovery of hosts connected to a network are still possible [240].

ADVANCES OF BOTNET INFRASTRUCTURE AND DISTRIBUTION

Another prediction is based on observations of the variety and speed with which new samples of malware specimen appear. These high appearance rates of new versions are possible through the concepts of polymorphism and metamorphism. A trend observed in the last few years is that these code changes are performed only on malicious servers, thus hiding the exact modification mechanisms from anti-malware researchers. Server-side generated malware may depend on parameters uniquely present in the host system that is requesting it. If the malware checks for the presence of these parameters before its execution, the malware sample is personalised to the targeted host. This has a dramatic impact on automated detection approaches, because the malware can only be examined in the presence of the system parameters for which it was generated. A comparable approach for the evasion of Intrusion Detection Systems is given in [242].

New developments in command-and-control infrastructures are also to be expected. Cloud hosting and services may be abused as a new tool for creating temporary channels and C&C domains. The same is true for Web 2.0 content and service providers, as initial experimental examples have shown for Twitter or Facebook [243]. Emerging web technologies such as HTML5 introduce new content tags that are likely

to bypass existing blacklists that have not been updated. The concept of WebSockets has the potential to be abused for malicious purposes, as shown in [244]. These includes issues that may arise from devices lacking compatibility with proxy attacks, exposing the systems behind NAT to the Internet. To a limited extent, botnets can be created entirely in JavaScript, and so independent from the underlying operating system [245].

MORE COMPLEX AND POTENTIALLY DANGEROUS MALWARE

Connected with the latter issue is the question of malware capabilities. That malware will further evolve to carry out its aims and conceal its activities can be regarded more as fact than prediction. The use of multiple zero-day vulnerabilities and, even more importantly, functionality, to manipulate industrial components in Stuxnet has shown that threats originating from malware can increase enormously within a short period. A major side-effect of the Stuxnet appearance is the demonstration to other malevolent parties of the possibility of attacks on industrial infrastructure, opening the door for these ideas to be copied.

APPEARANCE OF WHITE WORMS

Last, but not least, countermeasures against botnets may become more radical. The long-discussed principle of so-called “white worms”, which are autonomously spreading programs aimed at cleaning up infected systems, has yet to produce its first representative. In the recent Bredolab shutdown by the Dutch High Tech Crime Team [204], infected users, on connecting to the Internet, were forwarded to dedicated web pages that informed them about their infection. Using the botnet to inform users can be interpreted as a first step in this direction. Unauthorised installation and execution of code on remote machines, as would be necessary for a white worm, still remain illegal, as described in section 4.1.10 (Infiltration and Remote Disinfection).

7 APPENDIX

7.1 ABBREVIATIONS

AS – Autonomous System

AV - Anti-Virus

BGP – Border Gateway Protocol

BSI - Bundesamt für Sicherheit in der Informationstechnik (German Federal Office for Information Security)

CDN – Content Delivery Network

CERT – Computer Emergency Response Team

CIIP - Critical Information Infrastructure Protection

C&C – Command-and-Control

CPNI – Centre for the Protection of National Infrastructure

CSIRT – Computer Security Incident Response Team

CYBEX – Cybersecurity Information Exchange Framework

DEMONS - DEcentralised, cooperative and privacy-preserving MONitoring for trustworthiness

DHCP – Dynamic Host Configuration Protocol

DHT – Distributed Hash Table

DGA – Domain Generation Algorithm

DNS / DDNS – Domain Name System

DoS / DDoS – (Distributed) Denial-of-Service

DPI – Deep Packet Inspection

ENISA – European Network and Information Security Agency

EP3R – European Public Private Partnership for Resilience

FFSN – Fast-Flux Service Networks

HPB – Highly Predictive Blacklist

HTTP – Hyper-Text Transfer Protocol

IDS / HIDS / NIDS / IPS – Intrusion Detection System (Host-based, Network-based), Intrusion Prevention System

IP – Internet Protocol, often also used as an abbreviation for an Internet Protocol

address.

IPA - Information Technology Promotion Agency

IRC – Internet Relay Chat

ISAC - Telecom Information Sharing and Analysis Centre Japan

ISC – Internet Systems Consortium

ISP – Internet Service Provider

KISA - Korean Internet Security Agency

LE – Law Enforcement

LEA – Law Enforcement Agency

P2P – Peer-to-Peer

MD5 – Message Digest version 5

MELANI – Melde- und Analysestelle Informationssicherung (Swiss Reporting and Analysis Centre for Information Assurance)

MSO – Multi Systems Operator

NAT – Network Address Translation

NEISAS - National and European Information Sharing and Alerting System

NICC – National Infrastructure against Cybercrime

NRO - Number Resource Organization

OID – Object Identifier

RFC – Request For Comments (IETF standard proposal documents)

SMTP – Simple Mail Transfer Protocol

TCP – Transmission Control Protocol

TLD – Top-Level Domain

TTL - Time-To-Live

WAN – Wide Area Network

8 REFERENCES

All online references last visited on 14.02.2011.

- [1] ITU Study on the Financial Aspects of Network Security: Malware and Spam. ICT Applications and Cybersecurity Division, Policies and Strategies Department, ITU Telecommunication Development Sector, 2008.
- [2] Legal Issues in Botnet Mitigation. ENISA Report, to appear, 2011.
- [3] 10 Hard Questions on Botnet Mitigation. ENISA Report, to appear, 2011.
- [4] Year-end malware stats from AV-Test, 2011. [Online]
<http://sunbeltblog.blogspot.com/2011/01/updated-virus-stats-from-av-test.html>
<http://www.av-test.org/>
- [5] 2007 Malware Report: The Economic Impact of Viruses, Spyware, Adware, Botnets, and Other Malicious Code. Computer Economics, 2007.
- [6] Symantec Global Internet Security Threat Report: Trends for 2009 (Volume XV). Symantec Corp., 2010.
- [7] Kaspersky Security Bulletin 2009: Malware Evolution 2009.
- [8] Malware: Fighting Malicious Code. Skoudis E., Zeltser L., 2003.
- [9] Eggheads.org - eggdrop development. [Online] <http://www.eggheads.org/>
- [10] The DoS Project's "trinoo" distributed denial of service attack tool. Dittrich, D. University of Washington, 1999.
- [11] The "Stacheldraht" distributed denial of service attack tool. Dittrich, D. University of Washington, 1999.
- [12] An analysis of the "Shaft" distributed denial of service tool. Dietrich, S., Long, N. Dittrich, D. 2000.
- [13] TFN2K - An Analysis. Barlow, J., Thrower W. AXENT Security Team, 2000.
- [14] The history of worm-like programs. Darby, T., Schmidt, C., 2000. [Online]
<http://www.snowplow.org/tom/worm/history.html>
- [15] BO2K. Cult of the Dead Cow. [Online]. www.bo2k.com/
- [16] SubSeven 2.3.1. SubSevenCrew, [Online]. www.subseven.org
- [17] The Botnet Chronicles - The Botnet Chronicles. Ferguson, R. Trend Micro Whitepaper, 2010.
- [18] On the Analysis of the Zeus Botnet Crimeware Toolkit. Binsalleeh, H., Ormerod, T., Boukhtouta, A., Sinha, P., Youssef, A., Debbabi, M., Wang, L. In: Proceedings of the 8th Annual Conference on Privacy, Security and Trust, PST'2010, 2010.

- [19] An Analysis of Conficker's Logic and Rendezvous Points. Porras P., Saidi, H., Yegneswaran, V. Technical Report, SRI International, 2009.
- [20] The waledac protocol: The how and why. Sinclair, G., Nunnery, C., Kang, B.B.-H. In: Proceedings of the 4th International Conference on Malicious and Unwanted Software (MALWARE), 2009.
- [21] Insights from the Inside: A View of Botnet Management from Infiltration. Cho, C.Y., Caballero, J., Grier, C., Paxson, V., Song, D. In: Proceedings of the 3rd Usenix Workshop on Large-Scale Exploits and Emergent Threats (LEET'10), 2010.
- [22] Spamalytics: An Empirical Analysis of Spam Marketing Conversion. Kanich, C., Kreibich, C., Levchenko, K., Enright, B., Voelker, G., Paxson, V., Savage, S. In: Proceedings of the 15th ACM Conference on Computer and Communications Security (CCS), 2008.
- [23] Peer-to-Peer Botnets: Overview and Case Study. Grizzard, J. B., Sharma, V., Nunnery, C., Kang, B., Dagon, D. In: Proceedings of the First Workshop on Hot Topics in Understanding Botnets (HotBots'07), 2007.
- [24] Taking over the Torpig botnet- My botnet is your botnet. Project Website, Computer Security Group, UC Santa Barbara.
<http://www.cs.ucsb.edu/~seclab/projects/torpig/>
- [25] Call Centers for Computer Criminals. Krebs, B. Krebs on Security, 2010. [Online] <http://krebsonsecurity.com/2010/04/call-centers-for-computer-criminals/>
- [26] Virus Scanners for Virus Authors. Krebs, B. Krebs on Security, 2009. [Online] <http://krebsonsecurity.com/2009/12/virus-scanners-for-virus-authors/>
- [27] Botnet Kit And Service Offered To Non-Techies. CyberInsecure, 2008.
<http://cyberinsecure.com/botnet-kit-and-service-offered-to-non-techies>
- [28] The biggest cloud on the planet is owned by ... the crooks. Mullins, R. Network World, [Online] 2010 <http://www.networkworld.com/community/node/58829>
- [29] Involuntary Computing: Hacking the Cloud. Murphy, M., Stout L., Goasguen S. In: Proceedings of the 2nd IEEE Cloud Computing Conference, 2010.
- [30] Learning more about the underground economy: a case-study of keyloggers and dropzones. Holz, T., Engelberth, M., Freiling, F. In: Proceedings of the 14th European conference on Research in computer security (ESORICS'09), 2009.
- [31] M86 Security, Spam Statistics. [Online] http://www.m86security.com/labs/spam_statistics.asp
- [32] Ferris Research. [Online] www.ferris.com
- [33] ClickForensics, Inc. [Online] www.clickforensics.com
- [34] Anchor Intelligence, Inc. [Online] www.anchorintelligence.com

- [35] Cyberextortion: An Overview of Distributed Denial of Service Attacks against Online Gaming Companies. Paulson, R. A., Weber, J. E. In: Issues in Information Systems, Vol. VII, No. 1-2, 2006.
- [36] Low Orbit Ion Cannon - An open source network stress tool for Windows. NewEraCracker. [Online] <https://github.com/NewEraCracker/LOIC/downloads>
- [37] ShadowServer Foundation. [Online] <http://www.shadowserver.org>
- [38] BlackEnergy competitor – The 'Darkness' DDoS Bot. DiMino A. M. ShadowServer, 2010. [Online] <http://www.shadowserver.org/wiki/pmwiki.php/Calendar/20101205>
- [39] BlackEnergy DDoS Bot Analysis. Nazario J. Technical Report, Arbor Networks, 2007.
- [40] Politically Motivated Denial of Service Attacks. Nazario, J. In: The Virtual Battlefield: Perspectives on Cyber Warfare (pp. 163-181), IOS Press, 2009.
- [41] International Cyber Incidents: Legal Considerations. Tikk, E., Kaska, K., & Vihul, L. In: CCD COE Publications, 2010.
- [42] Tracking GhostNet: Investigating a Cyber Espionage Network. Information Warfare Monitor, 2009.
- [43] Shadows in the Cloud: An investigation into cyber espionage 2.0. Information Warfare Monitor and Shadowserver Foundation, 2010.
- [44] Is Stuxnet the 'best' malware ever? Keizer, G. Computerworld, 2010. http://www.computerworld.com/s/article/9185919/Is_Stuxnet_the_best_malware_ever?
- [45] W32.Stuxnet Dossier v1.3. Symantec Corp., 2010.
- [46] My botnet is bigger than yours (maybe, better than yours): why size estimates remain challenging. Rajab, M. A., Zarfoss, J., Monroe, F., Terzis, A. In: Proceedings of the First Workshop on Hot Topics in Understanding Botnets (HotBots'07), 2007.
- [47] abuse.ch ZeuS Tracker [Online] <https://zeustracker.abuse.ch>
- [48] abuse.ch SpyEye Tracker [Online] <https://spyeyetracker.abuse.ch/>
- [49] Conficker Working Group [Online] <http://www.confickerworkinggroup.org>
- [50] Your botnet is my botnet: analysis of a botnet takeover. Stone-Gross, B., Cova, M., Cavallaro, L., Gilbert, B., Szydlowski, M., Kemmerer, R., Kruegel, C., Vigna, G. In: Proceedings of the 16th ACM conference on Computer and communications security (CCS '09), 2009.
- [51] Eurostat, Special module 2010: Internet Security.
- [52] Microsoft Security Intelligence Report, Volume 9 (January 1st - June 30th 2010). Microsoft, 2010. [Online] <http://www.microsoft.com/security/sir>

[53] Global Energy Industry Hit In “Night Dragon” Attacks. Kurtz, G. McAfee Labs Blog Central, 2011. [Online] <http://blogs.mcafee.com/corporate/cto/global-energy-industry-hit-in-night-dragon-attacks>

[54] Large-scale Monitoring of Broadband Internet Infrastructures (LOBSTER). [Online] <http://www.ist-lobster.org>

[55] An algorithm for anomaly-based botnet detection. Binkley, J. R., Singh, S. In: Proceedings of the 2nd conference on Steps to Reducing Unwanted Traffic on the Internet (SRUTI'06), 2006.

[56] BotHunter: detecting malware infection through IDS-driven dialog correlation. Gu, G., Porras, P., Yegneswaran, V., Fong, M., Lee, W. In: Proceedings of the 16th USENIX Security Symposium on USENIX Security Symposium (SS'07), 2007.

[57] Snort. An open source network intrusion prevention and detection system (IDS/IPS). SourceFire. [Online] <http://www.snort.org/>

[58] SRI Honeynet and BotHunter Malware Analysis - Automatic Summary Analysis Table. SRI [Online] <http://wasp.csl.sri.com/Honeynet/>

[59] FloMA: Pointers and Software for Flow network assessment. Switch [Online] <http://www.switch.ch/network/projects/completed/TF-NGN/floma/software.html>

[60] Using machine learning techniques to identify botnet traffic. Livadas C., Walsh, R., Lapsley, D., Strayer, T. In: Proceedings of the 31st IEEE Conference on Local Computer Networks, 2006.

[61] A Proposed Framework for P2P Botnet Detection. Zeidanloo, H. R., Manaf, A., Ahmad, R., Zamani, M., Chaeikar, S. In: IACSIT International Journal of Engineering and Technology, Vol.2, No.2, 2010.

[62] Traffic aggregation for malware detection. Yen, T.-F., Reiter, M. K. . In: Proceedings of the 5th international conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA '08), 2008.

[63] Binary Codes Capable of Correcting Deletions, Insertions and Reversals. Levenshtein, V. In: Soviet Physics Doklady 10, 8, 1966.

[64] Towards automated detection of peer-to-peer botnets: on the limits of local approaches. Jelasity, M., Bilicki, V. In: Proceedings of the 2nd USENIX conference on Large-scale exploits and emergent threats: botnets, spyware, worms, and more (LEET'09), 2009.

[65] Network monitoring using traffic dispersion graphs (tdgs). Iliofotou, M., Pappu P., Faloutsos, M., Mitzenmacher M., Singh S., Varghese G. In: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement (ICM'07), 2007.

[66] Passive DNS Replication. Weimer, F. In. Proceedings of the 1st Conference on Computer Security Incidents, 2005.

- [67] Measuring the Perpetrators and Funders of Typosquatting. Moore, T., Edelman, B. In: 14th International Conference on Financial Cryptography, (TC'10), 2010.
- [68] Botnet Detection by Monitoring Group Activities in DNS Traffic. Choi, H., Lee H., Lee H., Kim H. In: Proceedings of the 7th IEEE International Conference on Computer and Information Technology (CIT '07), 2007.
- [69] Identifying Botnets Using Anomaly Detection Techniques Applied to DNS Traffic. Villamarin-Salomon, R., Brustoloni, J.C. In: Proceedings of the 5th IEEE Consumer Communications and Networking Conference (CCNC'08), 2008.
- [70] Analyzing DNS activities of bot processes. Morales, J.A., Al-Bataineh, A., Shouhuai, Xu, Sandhu, R. In: Proceedings of the 4th International Conference on Malicious and Unwanted Software (MALWARE), 2009.
- [71] Traffic Analysis on Mass Mailing Worm and DNS/SMTP. Musashi, Y., Sugitani, K., Matsuba, R. In: Proceedings of the 19th IPSJ SIGNotes Computer Security, 2002.
- [72] Honeytokens: The Other Honeypot. Spitzner, L. Symantec Connect [Online] <http://www.symantec.com/connect/articles/honeytokens-other-honeypot>
- [73] Spamcraft: An Inside Look At Spam Campaign Orchestration. Kreibich, C., Kanich, C., Levchenko, K., Enright, B., Voelker, G., Paxson, V., Savage, S. In: Proceedings of the 2nd USENIX Workshop on Large-scale Exploits and Emergent Threats (LEET '09), 2009.
- [74] Estimating Botnet Populations from Attack Traffic. Hao S., Feamster, N. 2008.
- [75] Characterizing botnets from email spam records. Zhuang, L., Dunagan, J., Simon, D. R., Wang, H. J., Tygar, J. D. In: Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats (LEET'08), 2008.
- [76] A methodology for anomaly and botnet detection and characterisation from application logs. Linari, A., Buckley, O., Duce, D., Mitchell, F., Morris, S. 2010.
- [77] Containing Conficker - To Tame A Malware. Leder, F., Werner, T. Honey Project, Know Your Enemy series, 2009.
- [78] Honeypots: Tracking Hackers. Spitzer, L. Addison-Wesley Professional, 2002.
- [79] Dionaea Honeypot [Online] <http://dionaea.carnivore.it/>
- [80] Nepenthes Honeypot [Online] <http://nepenthes.carnivore.it/>
- [81] mwcollected – malware collecting daemon [Online] <http://code.mwcollect.org/projects/show/mwcollectd>
- [82] Honeytrap – a low-interaction honeypot system [Online] <http://honeytrap.carnivore.it/>
- [83] Honeyd Virtual Honeypot [Online] <http://www.honeyd.org>

- [84] GQ: Realizing a System to Catch Worms in a Quarter Million Places. Cui, W., Paxson, V., Weaver, N. C. ICSI technical report TR-06-004, 2006.
- [85] Team Cymru Darknet Statistics. [Online]
<http://www.cymru.com/Reach/darknet.html>
- [86] CAIDA Internet Data -- Realtime Monitors [Online]
<http://www.caida.org/data/realtime/>
- [87] HoneyNet-based Botnet Scan Traffic Analysis. Zhichun Li, Anup Goyal, and Yan Chen. Invited book chapter for "Botnet Detection: Countering the Largest Security Threat", Springer, 2007.
- [88] Measurement and Analysis of Autonomous Spreading Malware in a University Environment. Goebel, J., Holz, T., Willems, C. In: Proceedings of the 4th international conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA '07), 2007.
- [89] GFI Sandbox (former CWSandbox) [Online]
<http://www.sunbeltsoftware.com/Malware-Research-Analysis-Tools/Sunbelt-CWSandbox/>
- [90] Overbeck, C. R. F. Efficient Observation of Botnets. Master's thesis, RWTH Aachen, University, 2007.
- [91] HoneyPot traces forensics: the observation viewpoint matters. Pham, V., Dacier, M. In: Proceedings of the 3rd International Conference on Network and System Security (NSS'09), 2009.
- [92] To catch a predator: a natural language approach for eliciting malicious payloads. Small, S., Mason, J., Monroe, F., Provos, N., Stubblefield, A. In: Proceedings of the 17th conference on Security symposium (SS'08). 2008.
- [93] The antivirus weather forecast: cloudy. Mashevsky, Y. SecureList 2010. [Online]
http://www.securelist.com/en/analysis/204792140/The_antivirus_weather_forecast_cloudy
- [94] Microsoft: Malicious Software Removal Tool [Online]
<http://www.microsoft.com/security/malwareremove>
- [95] VirusTotal - Free Online Virus, Malware and URL Scanner [Online]
<http://www.virustotal.com>
- [96] Jotti's malware scan [Online] <http://virusscan.jotti.org/>
- [97] Anubis: Analyzing Unknown Binaries [Online] <http://anubis.iseclab.org/>
- [98] NORMAN SandBox Information Center [Online]
http://www.norman.com/security_center/security_tools/

- [99] UMTS Network Planning, Optimization, and Inter-Operation with GSM. Rahnema, M. John Wiley & Sons, 2008.
- [100] Modelling botnet propagation using time zones. Dagon, D., Zhou, C., Lee, W. In: Proceedings of the 13th Annual Symposium on Network and Distributed System Security (NDSS'06), 2006.
- [101] DNS Cache Snooping. Grangeia, L. Research Paper, 2004.
- [102] Peeking through the cloud: DNS-based estimation and its applications. Rajab, M. A., Monroe, F., Terzis, A., Provos, N. In: Proceedings of the 6th international conference on Applied cryptography and network security (ACNS'08), 2008.
- [103] As the Net Churns: Fast-Flux Botnet Observations. Holz, T., Nazario, J. In: Proceedings of the 3rd International Conference on Malicious and Unwanted Software (MALWARE'08), 2008.
- [104] Active Threat Level Analysis System (ATLAS), Arbor Networks. [Online] <http://atlas.arbor.net/>
- [105] RFC 1918 - Address Allocation for Private Internets, 1996.
- [106] Measurements and mitigation of peer-to-peer-based botnets: a case study on storm worm. Holz, T., Steiner, M., Dahl, F., Biersack, E., Freiling, F. In: Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats (LEET'08), 2008.
- [107] Walowdac - Analysis of a Peer-to-Peer Botnet. Stock, B., Goebel, J., Engelberth, M., Freiling, F., Holz, T. European Conference on Computer Network Defense (EC2ND), 2009.
- [108] Speaking waledac. Leder, F. 2009. [Online] <http://www.honeynet.org/node/348>
- [109] The heisenbot uncertainty problem: challenges in separating bots from chaff. Kanich, C., Levchenko, K., Enright, B., Voelker, G. M., Savage S. In: Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats (LEET'08), 2008.
- [110] Storm: When researchers collide. Enright, B., Voelker, G., Savage, S., Kanich, C., Levchenko, K. In: USENIX; login, vol. 33(4), 2008.
- [111] Discovery techniques for P2P botnets. Dittrich, D., Dietrich, S. Stevens CS Technical Report 2008-4, 2008. Revised April 2009.
- [112] Federal Supreme Court decision regarding Logistep AG. Swiss Federal Data Protection and Information Commissioner (FDPIC), 2010. [Online] <http://www.edoeb.admin.ch/aktuell/01688/index.html?lang=en>

[113] Swedish court: IP addresses are personal data. European Digital Rights (EDRI), 2009. [Online] <http://www.edri.org/edri-gram/number7.13/sweden-ip-addresses-personal-data>

[114] French Court says an IP address is not enough for a user's identification. European Digital Rights (EDRI), 2010. [Online] <http://www.edri.org/edri-gram/number8.4/french-court-ip-address>

[115] EMI Records & Ors -v- Eircom Ltd. Judgement of the Irish High Court, [2010] IEHC 108. [Online] <http://www.courts.ie/Judgments.nsf/09859e7a3f34669680256ef3004a27de/7e52f4a2660d8840802577070035082f?OpenDocument>

[116] Directive 2002/58/EC of the European Parliament and of the Council of 12 July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector (Directive on privacy and electronic communications)

[117] Directive 2006/24/EC of the European Parliament and of the Council of 15 March 2006 on the retention of data generated or processed in connection with the provision of publicly available electronic communications services or of public communications networks and amending Directive 2002/58/EC.

[118] Copyright Violations in the Underground. Murchu, L.O. Symantec, 2008. [Online] <http://www.symantec.com/connect/blogs/copyright-violations-underground>

[119] Laws regarding botsnooping. Shadowserver Foundation, 2006. [Online] <http://www.shadowserver.org/wiki/pmwiki.php/Calendar/20060315>

[120] RSA Online Fraud Report - March 2010, RSA 2010.

[121] Vulnerability in Public Malware Sandbox Analysis Systems. Yoshioka, K., Hosobuchi, Y., Orii, T., Matsumoto T. In: Proceedings of the 10th IEEE/IPSJ International Symposium on Applications and the Internet, 2010.

[122] Communication from the Commission of the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions: Volume 1: i2010 — Annual Information Society Report 2009. Benchmarking 2010: Trends and main achievements.

[123] Highly Predictive Blacklists. Zhang, J., Porras, P., Ullrich, J. In: Proceedings of the 17th USENIX Security Symposium on USENIX Security Symposium (SS'08), 2008.

[124] DShield service, Internet Storm Center. [Online] <http://www.dshield.org>

[125] The anatomy of a large-scale hypertextual Web search engine. Brin, S., Page, L. In: Proceedings of the 7th international conference on World Wide Web 7 (WWW7), 1998.

[126] The Spamhaus Project. [Online] <http://www.spamhaus.org/>

- [127] Fraud detection using honeypot data tracking. Catlett, S. K., Xu He. WIPO Patent Application WO/2009/055785, 2009.
- [128] Defaming Botnet Toolkits: A Bottom-Up Approach to Mitigating the Threat. Ormerod, T., Wang, L., Debbabi, M., Youssef, A., Binsalleeh, H., Boukhtouta, A., Sinha, P. In: Proceedings of the 4th International Conference on Emerging Security Information, Systems and Technologies (SECURWARE 2010), 2010.
- [129] Lessons in Botnets: The After-Effects of ISP Takedowns. Shipp, A. RSA 2010 Conference.
- [130] Discerning Relationships: The Mexican Botnet Connection. Romera, R. Trend Micro, 2010.
- [131] S.3804: Combating Inline Infringement and Counterfeits Act. [Online] <http://www.govtrack.us/congress/bill.xpd?bill=s111-3804>
- [132] Battling rogue websites with new regulations. Carnes, R. Fierce Telecom, 2011. [Online] <http://www.fiercetelecom.com/story/battling-rogue-websites-new-regulations/2011-01-23>
- [133] Digital Millennium Copyright Act to amend title 17, United States Code, to implement the World Intellectual Property Organization Copyright Treaty and Performances and Phonograms Treaty, and for other purposes. Enacted by 105th United States Congress, 1998.
- [134] The COICA Internet Censorship and Copyright Bill. Electronic Frontier Foundation. [Online] <https://www.eff.org/coica>
- [135] Taking Back the DNS. Vixie, P. Internet Systems Consortium. Blog, 2010. [Online] <http://www.isc.org/community/blog/201007/taking-back-dns-0>
- [136] DNS Response Policy Zones (DNS RPZ), Draft 3. Vixie, P., Schryver, V. ISC Technical Note Series, 2010.
- [137] DNSSEC: DNS Security Extensions [Online] <http://www.dnssec.net>
- [138] DNSCurve: Usable security for DNS [Online] <http://dnscurve.org>
- [139] Good practices guide for deploying DNSSEC. Saragiotis, P. ENISA Technical Report, 2010. [Online] <http://www.enisa.europa.eu/act/res/technologies/tech/gpgdnssecp>
- [140] Revealing botnet membership using DNSBL counter-intelligence. Ramachandran, A., Feamster, N., Dagon, D. In: Proceedings of the 2nd conference on Steps to Reducing Unwanted Traffic on the Internet - Volume 2 (SRUTI'06), 2006.
- [141] Building a Dynamic Reputation System for DNS. Antonakakis, M., Perdisci, R., Dagon, D., Lee, W., Feamster, N. In: Proceedings of the 19th USENIX Security Symposium on USENIX Security Symposium (SS'10), 2010.

- [142] Zeus “in-the-cloud” . Ferrer, M. C. CA Community Blog [Online] <http://community.ca.com/blogs/securityadvisor/archive/2009/12/09/zeus-in-the-cloud.aspx>
- [143] Malicious Google AppEngine Used as a CnC. Nazario, J. Arbor Networks Security Blog. [Online] <http://asert.arbornetworks.com/2009/11/malicious-google-appengine-used-as-a-cnc>
- [144] FireEye [Online] <http://www.fireeye.com>
- [145] Managing Port 25 for Residential or Dynamic IP Space Benefits of Adoption and Risks of Inaction. MAAWG Recommendation, 2005.
- [146] RFC 2476 - Message Submission, 1998.
- [147] Best Practices in Anti-SPAM, Advisory document for the ETIS community. ETIS, 2010.
- [148] Dynamic port 25 blocking to control spam zombies. Schmidt, J. In: Proceedings of the 3rd Conference on Email and Anti-Spam (CEAS’06), 2006.
- [149] Common Best Practices for Mitigating Large Scale Bot Infections in Residential Networks. Mody, N., O’Reirdan, M., Masiello, S, Zebek, J. MAAWG, 2009.
- [150] StopBadware [Online] <http://stopbadware.org>
- [151] A Case Study in Ethical Decision Making Regarding Remote Mitigation of Botnets . Dittrich, D., Leder, F., Werner, T. Lecture Notes in Computer Science, Volume 6054/2010, 2010.
- [152] Proactive Botnet Countermeasures – An Offensive Approach. Leder, F., Werner, T., Martini, P. Proceedings of the 1st CCD COE Conference on Cyber Warfare, 2009.
- [153] SDbot command reference [Online] http://www.stanford.edu/~stinson/paper_notes/bots/bot_refs/sdbot_commandref.html
- [154] ‘Mariposa’ Botnet Authors May Avoid Jail Time. Krebs, B. Krebs on Security, [Online] 2010. <http://krebsonsecurity.com/2010/03/mariposa-botnet-authors-may-avoid-jail-time>
- [155] Proposal for a Directive on attacks against information systems, repealing Framework Decision 2005/222/JHA. MEMO/10/463, 2010.
- [156] Amendment to the Swiss Law on Telecommunications, 2007. [Online] <http://www.bakom.admin.ch/dokumentation/gesetzgebung/00512/00871/index.html?lang=en>
- [157] Swiss Federal Law on Unfair Competition of 19 December 1986 (Status as at 1 April 2007).
- [158] Swiss Federal Law on Telecommunications (Status as at 30 April 2007).

- [159] Swiss Ordinance on Telecommunications Services (OTS) of 9 March 2007 (Status as at 1 January 2010).
- [160] Court decision I ZR 121/08 of German Federal Court, 2010.
- [161] German Anti-Botnet Initiative [Online] <http://www.botfrei.de>
- [162] Staysafeonline (National Cyber Security Alliance) [Online] <http://www.staysafeonline.org>
- [163] Cyber Clean Center Japan [Online] <http://www.ccc.go.jp>
- [164] eco (Association of the German Internet Industry) [Online] <http://www.eco.de>
- [165] Federal Office for Information Security (BSI) [Online] <http://www.bsi.de>
- [166] The Australian Internet Security Initiative (AISI) [Online] http://www.acma.gov.au/scripts/nc.dll?WEB/STANDARD/1001/pc=PC_310317
- [167] IEEE Industry Connections Security Group [Online] <http://standards.ieee.org/develop/indconn/icsg/>
- [168] RFC 5070 - The Incident Object Description Exchange Format (IODEF), 2007.
- [169] Malware Attribute Enumeration and Characterization (MAEC) - A Standard Language for Attribute-Based Malware Characterization. [Online] <http://maec.mitre.org>
- [170] RFC 5965 - An Extensible Format for Email Feedback Reports (ARF), 2010.
- [171] x-arf: Network Abuse Reporting 2.0 [Online] <http://www.x-arf.org>
- [172] Italian chapter of the Honeynet Research Alliance [Online] <http://www.honeynet.it/>
- [173] Incentives and Barriers to Information Sharing. ENISA Report on Good Practices, 2010.
- [174] The German Anti-Botnet Initiative. Karge, S. OECD Workshop on the Role of Intermediaries in Strengthening Cybersecurity, 2010.
- [175] Learning from the Dutch - ISPs to co-operate against botnets. Virgo, P. When IT Meets Politics, ComputerWeekly.com Blog, 2009. [Online] <http://www.computerweekly.com/blogs/when-it-meets-politics/2009/09/learning-from-the-dutch---isps.html>
- [176] The Independent Post and Telecommunications Authority of the Netherlands (OPTA). [Online] <http://www.opta.nl/nl/>
- [177] Dutch National Infrastructure against Cybercrime (NICC) [Online] <http://www.samentagencybercrime.nl/>
- [178] Australian Communications and Media Authority. [Online] <http://www.acma.gov.au/>

- [179] Internet Industry Code of Practice. Internet Service Providers Voluntary Code of Practice for Industry Self-Regulation in the Area of Cyber Security, 2010.
- [180] CERT Australia. [Online] <http://www.cert.gov.au/>
- [181] Telecom Information Sharing and Analysis Centre Japan (ISAC). [Online] <https://www.telecom-isac.jp/>
- [182] Japan Computer Emergency Response Team Coordination Center. [Online] <http://www.jpcert.or.jp/>
- [183] Information Technology Promotion Agency, Japan (IPA). [Online] <http://www.ipa.go.jp/>
- [184] PCs Used in Korean DDoS Attacks May Self Destruct, Krebs, B. Washington Post Security Fix Blog, 2009. [Online] http://voices.washingtonpost.com/securityfix/2009/07/pcs_used_in_korean_ddos_attack.html
- [185] Korean Internet Security Agency (KISA). [Online] <http://www.kisa.or.kr/>
- [186] Korean CERT (KRCERT). [Online] <http://www.krcert.or.kr/>
- [187] Botnet C&C Handling with DNS Sinkhole. Jeong, H. C., Korea Information Security Agency, 2010.
- [188] Internet troubles in Korea? E-call center 118 is there to help. Van Horenbeeck, M. Technet Blog, 2010 [Online] <http://blogs.technet.com/b/ecostrat/archive/2010/09/17/internet-troubles-in-korea-e-call-center-118-is-there-to-help.aspx>
- [189] ITU Botnet Mitigation Toolkit. ITU Telecommunication Development Sector, Policies and Strategies Department, ICT Applications and Cybersecurity Division, 2008.
- [190] CYBEX: the cybersecurity information exchange framework (x.1500). Rutkowski, A., Kadobayashi, Y., Furey, I., Rajnovic, D., Martin, R., Takahashi, T., Schultz, C., Reid, G., Schudel, G., Hird, M., Adegbite, S. ACM SIGCOMM Computer Communication Review, Volume 40 Issue 5, 2010.
- [191] Working Party on Information Security and Privacy (WPISP) , Directorate for Science, Technology and Industry, Organisation for Economic Co-operation and Development (OECD). [Online] http://www.oecd.org/departement/0,3355,en_2649_34255_1_1_1_1_1,00.html
- [192] The Role of Internet Service Providers in Botnet Mitigation - An Empirical Analysis Based on Spam Data. Van Eeten, M., Bauer, J.M., Asghari, H., Tabatabaie, S. OECD, STI Working Paper, 2010.
- [193] Conficker Working Group: Lessons Learned. The Rendon Group, 2011.

- [194] DefenceIntelligence [Online] <http://www.defintel.com>
- [195] Another Botnet Gets Dismantled, But This Time With Arrests. Jackson Higgins, K. Tech Center: Insider Threat, Darkreading Newsblog, 2010. [Online] <http://www.darkreading.com/insider-threat/167801100/security/application-security/223101396/index.html>
- [196] Mariposa Botnet Briefing. Davis, C. Presentation for DefenceIntelligence.
- [197] Taking Down a Botnet. Amorosi, D. Infosecurity, 2010. [Online] <http://www.infosecurity-us.com/view/10063/taking-down-a-botnet/>
- [198] European Public-Private Partnership for Resilience (EP3R). [Online] <http://www.enisa.europa.eu>
- [199] European Commission CORDIS. Seventh Framework Programme (FP7). [Online] <http://cordis.europa.eu/fp7>
- [200] DEcentralized, cooperative and privacy-preserving MONitoring for trustworthiness (DEMONS) [Online] <http://fp7-demons.org>
- [201] Swiss Reporting and Analysis Centre for Information Assurance (MELANI) [Online] <http://www.melani.admin.ch>
- [202] UK Centre for the Protection of National Infrastructure (CPNI) [Online] <http://www.cpni.gov.uk>
- [203] National and European Information Sharing and Alerting System (NEISAS) [Online] <http://www.neisas.eu>
- [204] Dutch National Crime Squad announces takedown of dangerous botnet. Dutch Public Prosecution Service [Online], 2010. http://www.om.nl/actueel/nieuws-en/@154338/dutch_national_crime/
- [205] Bredolab - Severely Injured but not dead. FireEye Malware Intelligence Blog, 2010 [Online] <http://blog.fireeye.com/research/2010/10/bredolab-severely-injured-but-not-dead.html>
- [206] State of Spam & Fishing - A Monthly Report. Issue December 2010. Symantec, 2010.
- [207] Bredolab Malware spammed via fake Facebook Mails. Avira – TechBlog, 2011. [Online] <http://techblog.avira.com/2011/01/19/bredolab-malware-spammed-via-fake-facebook-mails/en/>
- [208] Communication from the Commission to the European Parliament and the Council. The EU Internal Security Strategy in Action: Five steps towards a more secure Europe. COM(2010) 673, 2010.
- [209] MessageLabs Intelligence: 2009 Annual Security Report. Symantec Hosted Services, 2009.

- [210] The McColo Effect: One Year Later. Masiello, S. McAfee Blog Central, 2009. [Online] <http://blogs.mcafee.com/mcafee-labs/the-mccolo-effect-one-year-later>
- [211] Atrivo – Cybercrime USA. Technical Report by HostExploit, 2008.
- [212] Zeus botnet's Real Host cut off from the internet. Ashford, W. ComputerWeekly, 2009. [Online] <http://www.computerweekly.com/Articles/2009/08/04/237165/Zeus-botnets-Real-Host-cut-off-from-the-internet.htm>
- [213] FTC Shuts Down Notorious Rogue Internet Service Provider, 3FN Service Specializes in Hosting Spam-Spewing Botnets, Phishing Web sites, Child Pornography, and Other Illegal, Malicious Web Content. Federal Trade Commission, 2009. [Online] <http://www.ftc.gov/opa/2009/06/3fn.shtm>
- [214] Cambridge hospital cleans up after mystery malware infection. Leyden, J. The Register, 2009. [Online] http://www.theregister.co.uk/2009/06/03/hospital_malware_outbreak/
- [215] El ordenador de Spanair que anotaba los fallos en los aviones tenía virus. El País, 2010. [Online] http://www.elpais.com/articulo/espana/ordenador/Spanair/anotaba/fallos/aviones/tenia/virus/elpepuesp/20100820elpepinac_11/Tes
- [216] Bredolab Infection Warning Page. Dutch National Alerting Service, 2010 [Online] <http://www.waarschuwingsdienst.nl/Risicos/Virussen+en+malware/Ontmanteling+Bredolab.html>
- [217] What we know (and learned) from the Waledac takedown. Microsoft Malware Protection Center Blog, 2010. [Online] <http://blogs.technet.com/b/mmpc/archive/2010/03/15/what-we-know-and-learned-from-the-waledac-takedown.aspx>
- [218] International Cooperation Disrupts Multi-Country Cyber Theft Ring. National Press Release of FBI National Press Office, 2010. [Online] <http://www.fbi.gov/news/pressrel/press-releases/international-cooperation-disrupts-multi-country-cyber-theft-ring>
- [219] LeaseWeb replaces Cisco CRS-1 core router with Juniper MX-960 3D. The Hosting News, 2010. [Online] <http://www.thehostingnews.com/leaseweb-replaces-cisco-crs-1-core-router-with-juniper-mx-960-3d.html>
- [220] Virgin Media to warn malware-infected customers. Williams, C. The Register, 2010 [Online] http://www.theregister.co.uk/2010/08/16/vm_malware
- [221] Constant Guard. comcast.net Security. [Online] <http://security.comcast.net/constantguard>
- [222] ENISA Clearinghouse for Incident Handling Tools - Software Auditing [Online] <http://www.enisa.europa.eu/act/cert/support/chiht/proactive-tools/software-auditing>

[223] The Underground Economy Brief – The PPI Model in the Underground Economy. Team Cymru, 2010.

[224] Money mules explained. Bank safe online, 2010. [Online]
http://www.banksafeonline.org.uk/moneymule_explained.html

[225] The Role of Internet Service Providers in Botnet Mitigation: An Empirical Analysis Based on Spam (Data DSTI/DOC(2010)5). Van Eeten, M., Bauer, J. M., Asghari, H., Tabatabaie, S. STI Working Paper Series of OECD Directorate for Science, technology and Industry, 2010.

[226] Conficker still infecting 50,000 PCs per day. McMillan, R. ComputerWorld, 2009. [Online]
http://www.computerworld.com/s/article/9133363/Conficker_still_infecting_50_000_PC_s_per_day

[227] SpyEye v. ZeuS Rivalry Ends in Quiet Merger. Krebs, B. Krebs on Security, 2010. [Online] <http://krebsonsecurity.com/2010/10/spyeye-v-zeus-rivalry-ends-in-quiet-merger/>

[228] SpyEye Bot versus Zeus Bot. Coogan, P. Symantec Connect, 2010. [Online]
<http://www.symantec.com/connect/blogs/spyeye-bot-versus-zeus-bot>

[229] Analysis of the 2007 Cyber Attacks Against Estonia from the Information Warfare Perspective. Ottis, R. In: Proceedings of the 7th European Conference on Information Warfare, 2008.

[230] Coordinated Russia vs Georgia cyber attack in progress. Danchev, D. ZDNet Zero Day Blog, 2008. [Online] <http://www.zdnet.com/blog/security/coordinated-russia-vs-georgia-cyber-attack-in-progress/1670>

[231] Hacktivism and the Future of Political Participation. Samuel, A. W. Dissertation, Harvard university, 2004.

[232] ENISA statement on WikiLeaks events. ENISA, 2011. [Online]
<http://www.enisa.europa.eu/media/news-items/enisa-statement-on-wikileaks-events>

[233] WW-12 -- Stupid Browser Phone Tricks. Japan Inc, 2001. [Online]
<http://www.japaninc.com/ww12>

[234] ニュース・ウォッチ モバイルにメール・ウイルス携帯電話
ドコモのiモードで2月発生 | コミュニケーション日経 | 日経BPサービス記事検索.
Nikkei Business Publications, 2001 [Online]
http://bizboard.nikkeibp.co.jp/kijiken/summary/20010305/NCC0337H_428183a.html.

[235] Q & A on "Sexy View" SMS worm. Hyppönen, M. F-Secure Blog, 2009. [Online]
<http://www.f-secure.com/weblog/archives/00001732.html>

- [236] Smartphones: Information security risks, opportunities and recommendations for users. Report by ENISA, 2010.
- [237] Remaining IPv4 Address Space Drops Below 5% - IPv6 adoption at critical phase. Number Resource Organization (NRO), 2010. [Online] <http://www.nro.net/media/remaining-ipv4-address-below-5.html>
- [238] Malware Tunneling in IPv6. US-CERT, 2005
- [239] Botnet Feature Advancement and Zeus Tweaking. Ollmann, G. Damballa Blog Post, 2009 [Online] <http://blog.damballa.com/?p=438>
- [240] Recent advances in IPv6 insecurities. Heuse, M. Presentation at 27th Chaos Communication Congress, 2010.
- [241] The Future of Threats and Threat Technologies - How the Landscape Is Changing. Trend Micro Report, 2009.
- [242] Context-keyed Payload Encoding: Fighting the Next Generation of IDS. Glynos, D. A. 1st AthCon IT Security Conference, 2010.
- [243] Web 2.0 Botnet Evolution – KOOBFACE Revisited. Baltazar, J. Trend Micro Research paper, 2010.
- [244] Transparent Proxies: Threat or Menace? Huang, L., Chen, E. Y., Barth, A., Rescola, E., Jackson, C., 2010.
- [245] Attacking with HTML5. Kuppan, L. Blackhat Abu Dhabi, 2010.