



[Recognising Botnets in Organisations]

Barry Weymes
Number: 662

A thesis submitted to the faculty of
Computer Science, Radboud University
in partial fulfillment of the requirements for the degree of

Master of Science

Eric Verheul, Chair
Erik Poll
Sander Peters (Fox-IT)

Department of Computer Science

Radboud University

August 2012

Copyright © 2012 Barry Weymes
Number: 662

All Rights Reserved

ABSTRACT

[Recognising Botnets in Organisations]

Barry WeymesNumber: 662
Department of Computer Science
Master of Science

Dealing with the raise in botnets is fast becoming one of the major problems in IT. Their adaptable and dangerous nature makes detecting them difficult, if not impossible. In this thesis, we present how botnets function, how they are utilised and most importantly, how to limit their impact. DNS Dynamic Reputations Systems, among others, are an innovative new way to deal with this threat. By indexing individual DNS requests and responses together we can provide a fuller picture of what computer systems on a network are doing and can easily provide information about botnets within the organisation.

The expertise and knowledge presented here comes from the IT security firm Fox-IT in Delft, the Netherlands. The author works full time as a security analyst there, and this rich environment of information in the field of IT security provides a deep insight into the current botnet environment.

Keywords: [Botnets, Organisations, DNS, Honeypot, IDS]

ACKNOWLEDGMENTS

- I would like to thank my parents, whom made my time in the Netherlands possible. They paid my tuition, and giving me the privilege to follow my ambition of getting a Masters degree.
- My dear friend Dave, always gets a mention in my thesis for asking the questions other dont ask. He is also a good man to spitball ideas towards, to help work them out. Also, a high five to Herb, my American friend.
- Fox-IT gave me an internship (and a job) to write this thesis, for that I am thankful. The majority of the technical example information displayed in this thesis is from Fox-IT, I am very happy that this type of information was available to me during my research.
- Within Fox-IT, I'd like to thank Sander Peters for having me, and to the whole Cybercrime team for putting up with me.
- Eric Verheul, my supervisor I would like to thank, and say that I appreciate the effort in helping this thesis come together.
- Last but not least, I'd like to thank the Kerckhoffs institute for its good courses and the knowledge that they have bestowed upon me.

Contents

Table of Contents	iv
1 Research proposal	1
1.1 Research context	1
1.2 Research question	3
1.3 Research objectives	3
1.4 Deliverables	4
2 Timeline	5
2.1 Past	5
2.1.1 Viruses	6
2.1.2 Worms	7
2.1.3 The first botnet: Ghostnet	8
2.2 Present	8
2.2.1 Trojans	8
2.2.2 Rootkits	9
2.2.3 Phishing/Poisoning	10
2.2.4 Spam	12
2.2.5 Bots	13
2.2.6 Advanced persistent threat	18
2.3 Future	23
2.3.1 Phonebots	24
2.3.2 Botclouds	25
2.3.3 PNRP	26
2.3.4 VOIP	27
2.3.5 DNS	29
2.4 Trends	30
2.4.1 Going mobile	30
2.4.2 Increased user awareness	31
2.4.3 Greater risks, greater losses	31
2.4.4 Bigger/better protection applications	32

3	Challenges	34
3.1	Endless amount of vulnerabilities	34
3.2	Sophistication	36
3.3	Law	38
3.4	Trust on the Internet	40
3.5	Users	41
3.6	Cost vrs Risk	42
4	Botnet in depth	44
4.1	Mechanics	44
4.1.1	Stages	45
4.1.2	Uses of a botnet	48
4.1.3	Communication	51
4.2	Organisation	56
4.3	Actors	58
5	Botnet Detection and Prevention	61
5.1	Detection Basics and Malware Tracking	62
5.1.1	Signature based detection	62
5.1.2	Anomaly based detection	63
5.1.3	Zeus/Spyeye Tracker	64
5.2	Malware collection	65
5.2.1	End users	65
5.2.2	Submitted malware	66
5.2.3	Binary analysis	66
5.3	DNS	68
5.3.1	DNS overview	68
5.3.2	Malicious uses of DNS	69
5.3.3	Current Anti-botnet technologies	71
5.4	Honeypots	77
5.4.1	Low vrs High interaction	78
5.4.2	Malware collection	79
5.4.3	Example, The Netherlands	80
5.4.4	Example, Japan	81
5.4.5	Botnet infiltration	82
5.5	IDS/IPS	83
5.5.1	Snort	83
5.5.2	Bro	84
5.5.3	Canary Detector	84
5.6	Firewalls	85
5.6.1	Proxy	86
5.6.2	Filter known malicious traffic	86

5.7	Financial transactions monitoring	87
5.7.1	Torpig/Mebrook	87
5.7.2	Virtual machines	88
5.7.3	Host based tripwire systems	88
5.7.4	Monitoring transactions	89
5.8	Security policy	90
5.8.1	Keeping programs up to date	91
5.8.2	Auditing an organisation	91
5.9	Botnet annihilation	92
5.9.1	Law	93
5.9.2	Technology companies	93
6	Organisations and Botnets	95
6.1	Small organisation/Home	95
6.2	Medium size organisation	97
6.3	Large organisation	105
7	Conclusion	107
7.1	Answering the research question	108
7.2	Discussion	112
A	Zeus logging samples	114
A.1	Banking details	114
A.2	Facebook details	114
B	DNS	115
B.1	DNS Tunnel example 1	115
B.2	DNS Tunnel example 2	115
	Bibliography	116

Chapter 1

Research proposal

1.1 Research context

These days the internet is an integral part of our daily life, it connects millions of people together on a daily basis. As such it has become a critical part of our infrastructure, without it we would be lost.

The trend of increased usage is bound to continue, however so are the increased threats that we are faced with in Internet security. Unfortunately what made the internet so successful: its openness and trusting nature, is something criminals use to conduct illegal activities. According to the International Telecommunication Union, the global cost of these activities is in the billions of dollars annually [40].



Figure 1.1 Current estimated cyber crime losses according to Symantec[2011]

Another report called the Global Economic Crime Survey by PWC [35] claimed that the threat of cybercrime is on the rise. It states that one in four of the responders have been experienced cybercrime once or more in the last 12 months. It seems the threat of cybercrime continues to increasingly harass computer users.

Countering this threat is an uphill struggle faced by computer security researchers and IT security sector, as a whole. Criminals have increased their innovation to conduct ever increasing sophisticated ways of making more money from users and businesses around the world. One such innovation is the botnet.

Later, we will look at all the issues that are faced on the internet in combating botnets. An example of one of these issues is that the Internet does not respect international boundaries or sovereignty, making it harder for the legal system to catch criminals that are in another jurisdiction.

This thesis is about *botnets*. They are arguably the most dangerous and hardest to tackle threats (as we shall see in later chapters). A *bot* is a computer system that is infected with botnet software, it can be controlled remotely by its controller. This controller is typically called a *botmaster*. He controls the bots and commands them to conduct whichever actions he chooses.

Organisations such as large corporations stand to lose a significant amount of money and reputation from botnets as they are an increasingly dangerous threat. Whether it is data loss due to a bot infection within the organisation or attacks from outside by many bots, an organisation must accept the fact that they will be (specifi-

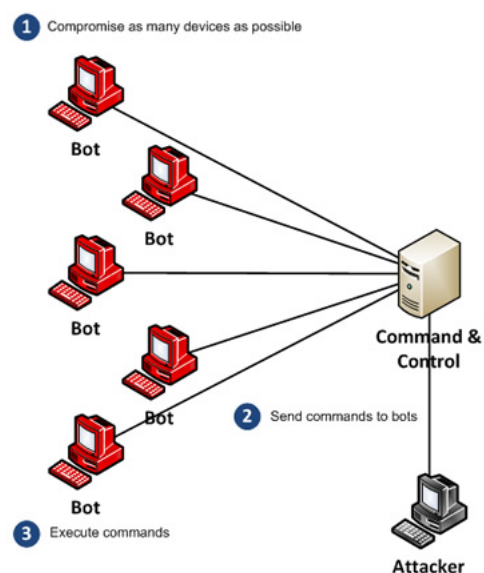


Figure 1.2 A typical botnet setup.

cally) attacked.

1.2 Research question

My research question is **What are the best methods for detecting and analysing modern botnets in organisations?**

This research question specially focuses on the following various sub questions:

- What are the trends of botnets and what are the things we can expect in the future?
- What challenges are currently being faced in defending against botnets?
- How can DNS servers and their data/records be used to detect botnets in organisations?
- How can honeypots be used to detect botnets in organisations?
- What is the best strategy for detecting botnet activity? Are passive or active approaches better?
- Should organisations focus on botnets or other threats, how big is this problem for organisations compared to other security problems?

1.3 Research objectives

The objective of this thesis is to answer the research questions above. This will be done by reviewing the current (scientific) literature on botnets, in particular with relation to DNS, Honeypots, Law and future threats. The authors technical experience will also be used to answer the research questions.

It is hoped that others can benefit from the knowledge that this thesis provides to implement better security measures within their own organisations. The information provided in this thesis

will provide the reader with indepth knowledge on the subject of botnets, and how they the threat of them within an organisation can be migrated.

1.4 Deliverables

The deliverable is this thesis consisting of 7 chapters.

The research proposal will be the first chapter. The second chapter will consist of a timeline to give the reader a history of cybercrime. This timeline will explain the current internet security environment. Next, the future trends of botnets are explained.

In the third chapter, the challenges that are faced in defending against botnets are explored. The endless stream of vulnerabilities that can exploited, the difficulties the law has in prosecuting, as well as the increased sophistication that cyber criminals are developing are all looked at.

Chapter 4 Botnets in depth, will focus on what botnets are. How they infect systems, propagate and how they are operated. Chapter 5 complements the previous chapter by looking at botnet detection and prevention. This is a big chapter that will look into how an organisation can defend against the botnet threat.

This leads us to chapter 6 which will look at 3 different organisations and how botnets specifically effect them. Its will also look at what best defences can be deployed for the different sized organisations. Finally, in chapter 7, the conclusion is dealt with as well as some recommendations for further research in the subject.

Chapter 2

Timeline

The timeline will give the reader an indication of how the cybercrime threat has come about, where it currently is and my predictions for the future. We will look at malware which is short for 'malicious software'. This malware can come in many different forms, as we shall see later. Its focus shall be on answering the sub question: *What are the trends of botnets and what are the things we can expect in the future?*

We set out to discuss the trends at the end of the chapter. These trends will give a general indication of how the cybercrime environment is changing.

2.1 Past

"To know your future you must know your past" - George Santayana

In this section we will look at viruses and worms. Both attack systems without regard to the actual system, their objective is merely to keep propagating. It should be noted however the line between a computer worm and virus is blurring these days.

2.1.1 Viruses

Computer viruses gain their name from the fact that they infect a computer system by attaching onto another program or file to infect the computer. It then tries to propagate. The word virus is commonly used to falsely describe many other types of malware.

The early viruses would infect the boot sector of a floppy disk, the user would normally boot the computer from the floppy disk to use the system. As the infected disk was booting the virus would execute its commands first and therefore was able to perform any actions it wanted without interference. The main route of infection in these early days was by sharing removable media with others.

In the early 90's, macro viruses were a common way of infection. macro viruses would infect a file used by popular software programs, for example Microsoft Word or Excel. These files would then spread from the need to share these documents around the workplace or home environment.

The computer virus became more advanced to keep one step ahead of the defences that had been put in place, such as anti-virus software. These features of viruses include various methods of avoiding detection. Some viruses would change their executable code to make it harder to detect. Others would encrypt themselves so they were not viewable to others trying to study it.

The era of viruses attaching themselves to programs and propagating seems to be over. Increased user awareness, better detection and better security policies such as blocking executable email attachments have stopped all but the most advanced viruses.

2.1.2 Worms

A computer worm is like a computer virus only they are standalone programs designed to replicate over networks. The worm will infect a host mostly via network vulnerabilities. The first worm to cause major disruption was the morris worm [32]. It was before the invention of the firewall making it possible for the worm to spread easily. In fact it spread so rapidly it exhausted itself, it had no other computer systems left to infect.

The world has seen many worm infections that have shaped history. Worms such as sasser, blaster and conficker all lead to massive disruption that has lead to the increased awareness of the threat. The characteristics of a worm infection are these days very easy to defend against by advances made in monitoring internet traffic. Users, businesses, internet service providers and government agencies have all gained significant knowledge on worm infection and are now able to response accordingly.

This increased awareness and better defences has lead cyber criminals to concentrate on making others types of malware to infect machines. Worm infection are mostly a thing of the past. One exception is the Morto RDP worm [16]. It is a worm that spreads via Remote Desktop Protocol (RDP). Morto scans the local network for RDP servers and tries to use weak passwords to gain access to the machine. Its a good example of an old infection method using a relativity new technology: RDP.

Another type of worm is the web worm. This worm propagates solely via vulnerable web servers, using cross-side scripting (XSS). XSS is vulnerability caused by improper input validation with permits executable content to embedded within a web page. The first and most successful web worm was Santy, it exploited a vulnerability in phpBB, a billboard web application. It spread within 3 hours to 50,000 servers using Google to search and was only stopped by them when they filtered all the worm related search queries.

2.1.3 The first botnet: Ghostnet

Ghostnet was a massive botnet that spread to hundreds of countries, infecting mostly high value targets such as embassies and the private offices of the Dalai Lama. It was the first of its kind, focusing on these high value targets only.

The vector of infection was from malicious email attachments that would install a remote access tool (RAT), to allow the attackers remote access to the valuable machines. These attachments were executed because the users were social engineered into thinking they were real attachments.

Once the infected machine was accessed it could be used to do a variety of operations such as file manager, screen capture, keylogger, remote shell, webcam and audio capture, as well as the ability to force the infected host to download and execute additional malware, such as a RAT update. The attacker could also secretly execute programs on the target computer¹.

It is commonly believed that the Chinese was behind this botnet. The majority of targets were from countries that don't enough good diplomatic relations, such as Taiwan and Tibet.

2.2 Present

In this section, I will try detail what the current cybercrime environment looks like. Much of the information comes from my job, at FOX-IT as a security analyst. While working there I have seen lots of threats and below you will see data from actual infections on networks.

2.2.1 Trojans

The trojan horse is named after the wooden horse the Greeks used to infiltrate Troy. It is a normal computer program, however included within this program is another program that executes malicious actions that the users do not desire.

¹<http://www.f-secure.com/weblog/archives/ghostnet.pdf>

A common example of a trojan horse is an unlicensed copy of a well known program such as a game that has a malicious program added to it. This game will then be executed by the user, which will mean the malicious section gets executed as well. Trojans will typically create a backdoor for future access by the criminals. With this access they can then do almost anything they want with the machine.

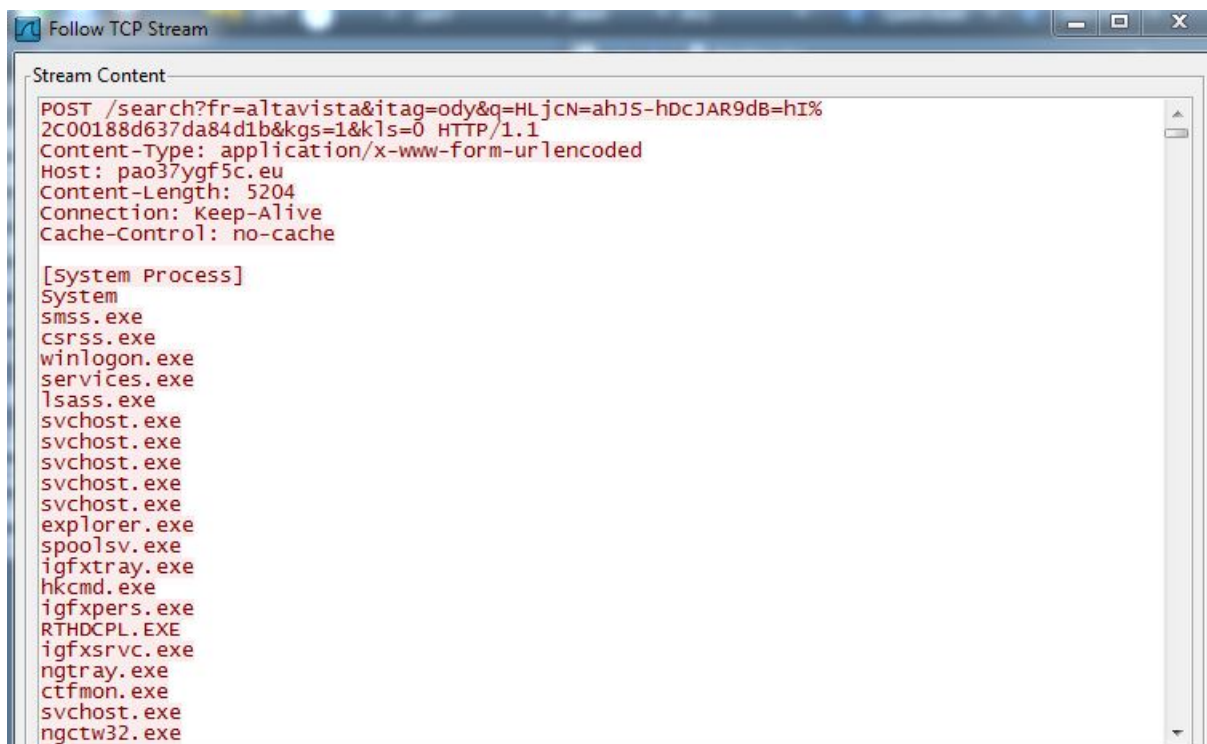
Unlike viruses and worms, Trojans do not reproduce by infecting other files nor do they self-replicate. Trojans must spread through user interaction such as opening an e-mail attachment or downloading and running a file from the Internet.

2.2.2 Rootkits

A rootkit is a type of malware that tries to stay hidden from the operating system and the users of the system. It does this by booting itself first, before the operating system or another program can execute. By booting first, the trojan can do almost anything it wants with the computer system.

Rootkits are notoriously difficult to detect and disinfect because they install themselves into the core of the computer system. There are a number of places it can try and hide. The most common is adapting the Master Boot Record(MBR) so that the rootkit can boot first. Another place the rootkit can hide is within the kernel, by adapting it. As the kernel has the highest security level it can operate within the most trusted part of the operating system. Many currently believe that once a computer is infected, it can be disinfect by reinstalling the operating system. With rootkits this is not the case.

An example of a rootkit infection is Mebroot. This is a particularly difficult infection to defend against as infects the MBR so that it can survive most attempts to disinfect the computer. It also can hook the network drivers, which mean nobody can detect its presence, to bypass the firewall. Thankfully it is easy to detect from within the network as it uses google.com to check for internet connectivity in a very specific way, www.google.com/webhp. It will then start sending data to the



```
Follow TCP Stream
Stream Content
POST /search?fr=altavista&itag=ody&q=HLjcn=ahJS-hDcJAR9dB=hI%
2C00188d637da84d1b&kgs=1&kls=0 HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Host: pao37ygf5c.eu
Content-Length: 5204
Connection: Keep-Alive
Cache-Control: no-cache

[System Process]
System
smss.exe
csrss.exe
winlogon.exe
services.exe
lsass.exe
svchost.exe
svchost.exe
svchost.exe
svchost.exe
svchost.exe
explorer.exe
spoolsv.exe
igfxtray.exe
hkcmd.exe
igfxpers.exe
RTHDCPL.EXE
igfxsrvc.exe
ngtray.exe
ctfmon.exe
svchost.exe
ngctw32.exe
```

Figure 2.1 A Mebroot infection sending a list of processes to the CnC server.

CnC servers, such as processes that are currently running, and will then begin sending encrypted data.

2.2.3 Phishing/Poisoning

Around 2004, malicious websites became another avenue of attack for criminals. Instead of trying to attack the system that the user is using, attacks via malicious webpages focus on tricking the users into providing their credentials. It is a form of social engineering to attack in this way.

Banking websites and social media websites are particularly targeted as they provide legitimate information from careless users who send their log in details to these fake websites. Once the attackers have the credentials they can transfer money from a bank account or introduce the users friends to the phishing site.



Figure 2.2 Example of an email footer from a phishing attempt, note the in email the misspelled word *departament*

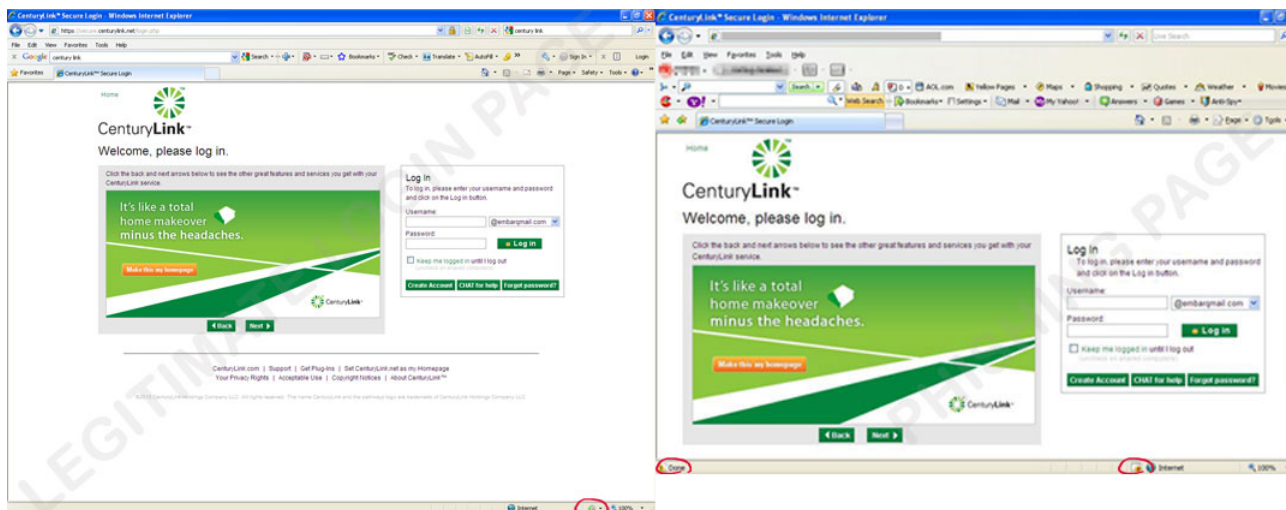


Figure 2.3 Example of a legitimate banking login website, and a phishing website [23]

A typical avenue of attack is via phishing emails. Many users will receive a spoofed email from their bank, for example, with instructions how to prevent their bank account being closed. The users will then open the attachment attached or follow a link from the email.

Another related attack is the poisoning attack, in it a legitimate website is infiltrated with some automated malware such as some malicious JavaScript. When any user visits the legitimate web-

site, the malicious actions are also executed. Its is possible session cookies of a website can then be stolen. This type of attack is very stealthy, the infected webpage can go unnoticed for a long amount of time.

2.2.4 Spam

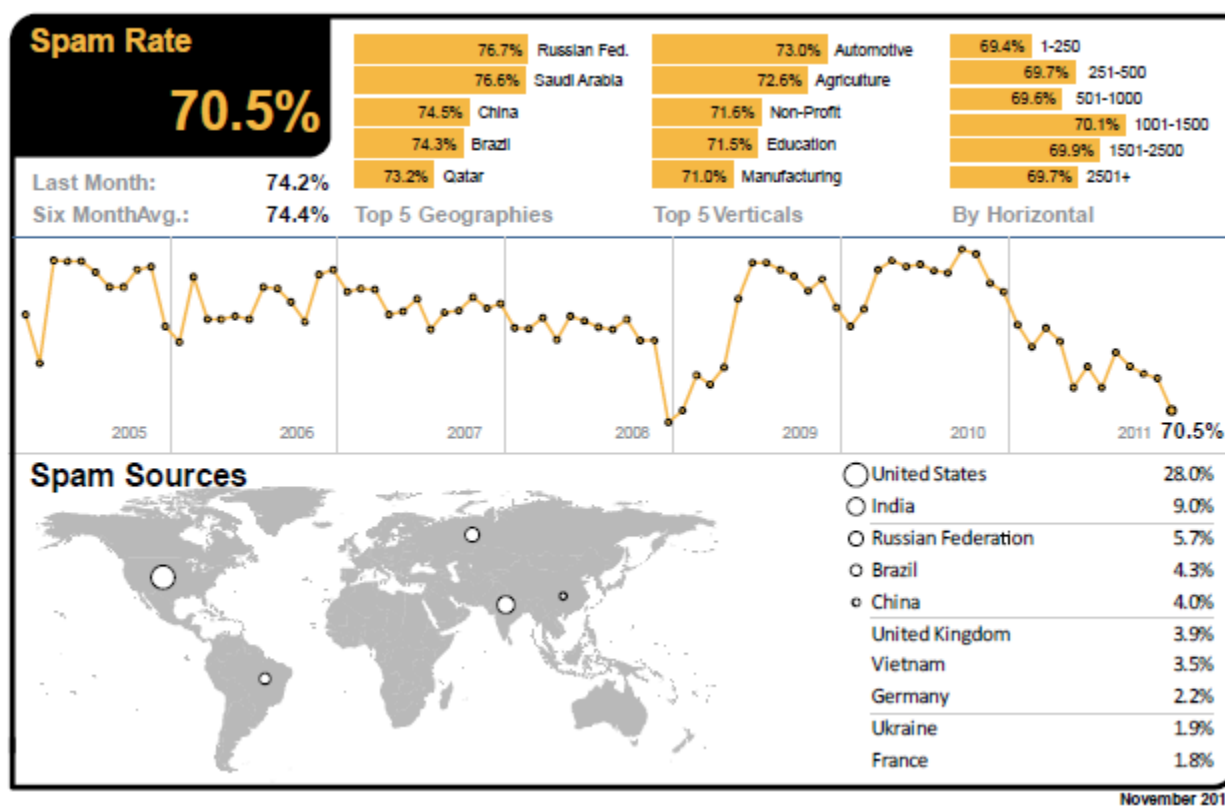


Figure 2.4 Spam levels are now at a record low, Symantec [Nov 2011]

Spam can be defined as "an electronic message that is unsolicited and bulk". Unsolicited meaning that the recipient has not agreed to receive the message in advance. Bulk means that an identical message is sent to a large number of recipients.

Spam emails vary in content, some are used to attempt a phishing attack, others are used to advertise products. It should be noted that in many countries spam is now illegal, however its

volume has remained constant, mostly due to botnets.

Recently a report by Symantec [39] has shown that the level of spam has fallen to a record low. Now only 75% of emails sent are spam, compared to 90% one year ago.

It is believed that this lower level of spam is not because of law enforcement efforts or other attempts to stop spam. Spammers are instead sending more focused spam to specific people or organisations. The increased use of social media networks and other personal information becoming available has allowed the spammers to focus on their targets, instead of spamming arbitrarily.

2.2.5 Bots

All the above malware can lead to the computer system becoming a bot. There are many avenues of attack for cybercriminals to use to gain control of computer systems. Once they are infected they are controlled by the botmaster to conduct:

- DDOS attacks on other systems
- Spamming
- Proxying of a cybercriminals activities to conceal their tracks
- Information stealing
- and many other malicious things

We will revisit bots and botnets in chapter 4. Suffices to say that botnets are a serious threat to anyone online, not just end users. Lets look at currently three of the most dangerous botnets in the world today: Zeus, Spyeye and ZeroAccess.

Zeus

Zeus is a sophisticated banking trojan horse, that is capable of intercepting keystrokes and form data. Its primary purpose is to steal online banking information, so that this information can be used to steal the users money. At its height the botnet is said to have 3.6 million bots in the US alone. Most of the infections happened via driveby download or phishing attacks. A driveby download is the intended downloading of (malicious) software from the internet. It can occur either by clicking on a malicious link, as many users do, or via a specially crafted attack were the user is directed to a website that he/she thinks is legitimate but its in fact malicious.

Zeus is special because it is very professionally build. Its can be purchased by anybody with money and intent. It uses a copy protection mechanism to prevent others copying its code. Once a cybercriminal buys the Zeus software, they can install it on one computer and generate custom binaries for attack. In this way the cybercriminals can adapt their attacks to what suits them. This has give rise to the notion of Do-It-Yourself (DIY) malware. Zeus is also modular, which means that the software can have addons such as chat listeners and VPN support [21].

In May 2011, the source code for Zeus was leaked, probably by the programmer himself. The high profile of Zeus was probably something that was not desirable. The release of this source code was an indication that he was retiring. However many believe this is just a ruse, to keep under the radar. Without all the distractions of publicity, the programmer is believed to have started a premium botnet software program that is only sold to very special clientele [2].

Appendix A shows a real life example of what can be gathered by the Zeus botnet. One Italian girl called Claudio logging into an Italian bank called Inbank and has her credentials stolen. The most common data stolen on this botnet is actually facebook logging credentials.

The leak of the Zeus source code has lead to a massive increase in botnets that use this code. Many new botnets are simply customisations of the Zeus code.

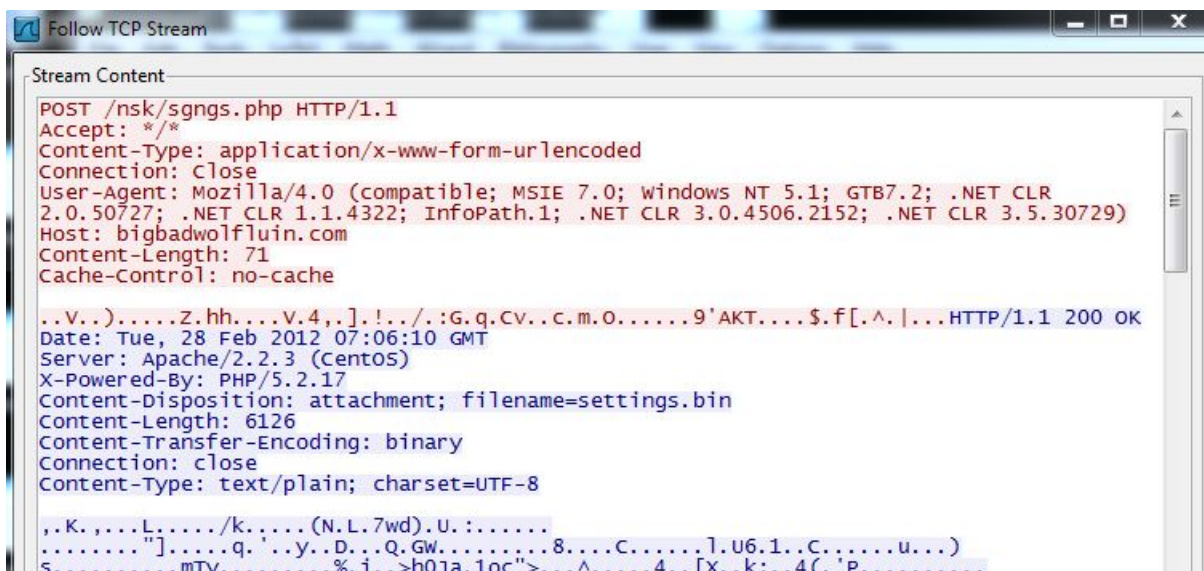
Spyeye



Figure 2.5 A screen shot of a Spyeye binary builder. This will build a fresh version of Spyeye for this owner, to help prevent detection.

The Spyeye botnet is what some consider the successor of Zeus. It has the same characteristics as Zeus, its modular and copy protected, see Figure 2.5. Its development is ongoing. New features are always being added. Spyeye will actually detect a Zeus infection, steal its logs and can in some cases delete the Zeus software itself.

Once a machine is infected it will request the latest configuration bin file, Figure 2.6 shows an example of settings.bin. It is an encrypted and password protected file. Regularly, the file is encrypted with a password not even the malicious installation knows. It must spend time cracking a simple (different) password, it does this to frustrate analysis.



```

Follow TCP Stream
Stream Content
POST /nsk/sgngs.php HTTP/1.1
Accept: */*
Content-Type: application/x-www-form-urlencoded
Connection: close
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; windows NT 5.1; GTB7.2; .NET CLR
2.0.50727; .NET CLR 1.1.4322; InfoPath.1; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)
Host: bigbadwolfluin.com
Content-Length: 71
Cache-Control: no-cache

..V..).....Z.hh...V.4,.]!.../.:G.q.CV..c.m.O.....9'AKT....$.f[.^.|...HTTP/1.1 200 OK
Date: Tue, 28 Feb 2012 07:06:10 GMT
Server: Apache/2.2.3 (CentOS)
X-Powered-By: PHP/5.2.17
Content-Disposition: attachment; filename=settings.bin
Content-Length: 6126
Content-Transfer-Encoding: binary
Connection: close
Content-Type: text/plain; charset=UTF-8

,.K.,...L...../k.....(N.L.7wd).U.:.....
....."].....q.'..y..D...Q.GW.....8....C.....l.U6.1..C.....u...)
s.....mTV.....%.i..>h0Ja.1oc">...^.....4..fX..k:.4(. 'P.....

```

Figure 2.6 The infected machine requests the latest settings file. Note that the malicious communication has **filename=settings.bin** in the response

Once the bin file is decrypted it will execute itself with the new configurations and its add on modules such as a credit card grabber. It will then inject itself into various Windows processes, and hooks the browser traffic so the browser gets the traffic after its decrypted but before its displayed in the browser. This is a very useful function for a banking trojan.

ZeroAccess

ZeroAccess is a sophisticated kernel-mode rootkit that is rapidly becoming one of the most widespread threats in the current malware ecosystem [43]. ZeroAccess has the ability to run on both 32-bit and 64-bit versions of Windows, includes a resilient peer-to-peer command and has a control infrastructure that constantly updates its functionality over time show that ZeroAccess is a modern threat capable of thriving on modern networks and modern operating systems.

Once ZeroAccess is executed by whatever means (Driveby, Exploit kit or social engineering) it will install itself. To gain administrator privileges, it will ask the user to install flash player.

This flash player install is a hacked version that looks real, but is really just linked to a malicious DLL. However when the user installs the Adobe Flash Player it appears to be legitimate. It appears legitimate because it is a legitimate installer, signed by Adobe. The difference with the ZeroAccess version is that the DLL it uses called msimg32.dll is replaced with a malicious version. Windows will load this DLL in preference to the genuine msimg32.dll because Windows looks in the current directory before the system directory when loading DLL's. Very smart.



Figure 2.7 A fake Adobe Flash Player that looks legitimate, but is really just the ZeroAccess trojan

There are 2 possible payloads to ZeroAccess, a click fraud module and a spambot module. The click fraud module can redirect the browsers search results to anything. It gets a regularly updated list of URL and referrer URL's to the redirection. When the spambot payload is downloaded it installs itself, downloads spam templates and target email addresses and sends spam. It has likely rented out a portion of the bots for this.

ZeroAccess is very difficult to get rid of because of its aggressive self defence. An excellent example of this is the bait it sets. A process is created that is monitored by the rootkit. If any application attempts to open this 'bait' process then the rootkit will attack that application. In this way, it can stop any security process that might be in place.

2.2.6 Advanced persistent threat

A commonly used principle in determining security is gauging one-selves security against others security. If you have better security than others, someone will more than likely attack the easier target. That is premise of the whole IT security sector, that supports millions of jobs. If a business is better protected than its rival it can consider itself doing well on security.

Advanced persistent threats (APT) change how we all look at the above security principle, because APT does not consider this basic principle on security. APT is a targeted attack that will focus on one organisation or individual. Put simply, other organisations do not matter in this case because it is a targeted attack.

Definitions of what APT exactly is varies, however here are the basic requirements:

Advanced

- Implies the attacker has a full spectrum of knowledge in intelligence gathering.
- Has access to many malware do-it-yourself kits, or has the capability to build its own specially malware.
- Uses a variety of methods to compromise the target, and has time and patience to attack the target slowly without arousing suspicion.

Persistent

- The attacker wants long term access to the target instead of just short term opportunist possible gains in mind such as financial information.
- Continuously monitors the target to achieve its objective.
- Will try re-establish communication if they lose access to the target.

Threat

- A threat because the attackers are well funded and organised, such as government agencies.
- Actions of the attacker are coordinated human based attacks not mindless automated attacks such as a computer virus.
- They have a specific objective, are skilled and motivated.

Next we will look at the 4 APT botnets that have attracted allot of attention. We could dedicate a whole thesis to these type of threats but we only look at them in brief. All were constructed by a groups that have lots of experience, knowledge and resources. One example of such sophistication is the compromising of trusted private keys of hardware manufacturers, these valuable keys were not used in any obvious botnet that would be discovered out eventually. Instead they were carefully planned to be used in the most effective manner possible, stealthy.



Figure 2.8 Stuxnet, Duqu, Flame and Guass. Will the next APT botnet begin with a 'H'?

Stuxnet

Stuxnet is a computer worm that can be defined as an APT. Its nature is to setup a (covert) network, and is therefore a bot. The bots discovery was partially accidental or lucky, and partially because of the attackers error in attacking systems that were not within its objective. Its is commonly believed that the worms objective is to infect Siemens SCADA systems within Iran, to stop or delay their nuclear weapon capabilities [42].

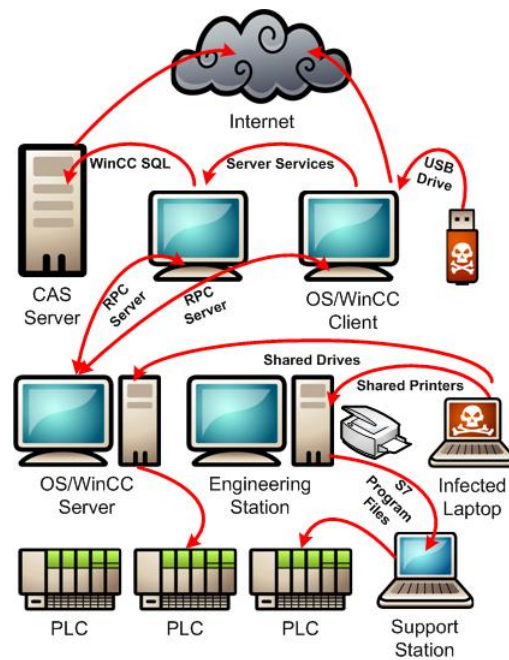


Figure 2.9 How Stuxnet propagates.

Stuxnet infected both the windows computers controlling the SCADA systems as well as the PLC controllers that control the actual machinery. It could provide fake data to the systems controllers. Its slow nature of propagating was designed to avoid detection. Once it had infected a system it would disable the anti virus, contact the CnC for instructions and try find a SCADA system to control².

²<http://www.isssource.com/stuxnet-report-ii-a-worm%E2%80%99s-life>

Duqu

Duqu is Stuxnet's little brother, its functionality is almost similar as it. The real difference is that instead of attacking the SCADA systems it is involved in intelligence gathering that could lead to another Stuxnet [29].

Feature	Stuxnet	Duqu
Modular malware	✓	✓
Kernel driver based rootkit	✓	✓ very similar
Valid digital signature on driver	Realtek, JMicron	C-Media
Injection based on A/V list	✓	✓ seems based on Stux.
Imports based on checksum	✓	✓ different alg.
3 Config files, all encrypted, etc.	✓	✓ almost the same
Keylogger module	?	✓
PLC functionality	✓	✗ (different goal)
Infection through local shares	✓	No proof, but seems so
Exploits	✓	?
0-day exploits	✓	?
DLL injection to system processes	✓	✓
DLL with modules as resources	✓ (many)	✓ (one)
RPC communication	✓	✓
RPC control in LAN	✓	?
RPC Based C&C	✓	?
Port 80/443, TLS based C&C	?	✓
Special "magic" keys, e.g. 790522, AE	✓	✓ lots of similar
Virtual file based access to modules	✓	✓
Usage of LZO lib	?	✓ multiple
Visual C++ payload	✓	✓
UPX compressed payload,	✓	✓
Careful error handling	✓	✓
Deactivation timer	✓	✓
Initial Delay	? Some	✓ 15 mins
Configurable starting in safe mode/dbg	✓	✓ (exactly same mech.)

Figure 2.10 How Duqu and Stuxnet can be compared. They are almost similar expect that Duqu doesn't attack PLC's as Stuxnet does [29].

Both stuxnet and duqu have shown that by lending out trust to digital signatures, we are expose to advanced attacks. Stuxnet had 2 Taiwanese hardware manufacturers sign the drivers used for its install. Duqu additionally used one new Taiwanese hardware manufacturer's signature to as disguise. These signatures were probably allowed to happen because the companies networks were compromised and the mechanism for signing drivers was stolen or copied.

Flame

Flame is a sophisticated attack toolkit, which is a lot more complex than Duqu. It is a backdoor, a Trojan, and it has worm-like features, allowing it to replicate in a local network and on removable media if it is commanded so by its master [30].

Its goal is surveillance, it can be used to record audio, screenshots, keyboard activity and network traffic. The program also records Skype conversations and can turn infected computers into Bluetooth beacons which attempt to download contact information from nearby Bluetooth enabled devices. This data, along with locally stored documents, is sent on to one of several command and control servers that are scattered around the world. The program then awaits further instructions from these servers.

Kaspersky discovered that its target was documents that would be useful to intelligence agencies such as drawing and PDF's. Most notably within the Islamic Republic of Iran.

A really interest part of Flame, is the 'Gadget' module that is used for propagation. One of the way it could propagate was via the local network. It was able to infect a fully patched windows 7 computer. It did this by pretending to be a Windows update server and conducting a collision attack on the real update as it had a weak MD5 hash algorithm protecting it.

Guass

Recently, a new malware called Guass has appeared. Its structure is linked to the previous three botnets. However, its goal is to monitor banking. This is a significant step as it this is the first time such an advanced botnet has looked at banking details.

As of August 2012, Stuxnet, Duqu, Flame and Guass are all voluntarily shutdown because they no longer serve as convert way to achieve their goals. This is not surprising, as once the world was watching the botnet controller couldnt risk the exposé. It still remains to be seen what the next APT botnet is.

2.3 Future

	Botnet	Mini-Botnets
Owner	Same group or individual	Different groups or individuals
Purpose	General (Spam, DDoS, etc)	Specific (Information stealing)
Target	General (any vulnerable host on the world wide net)	Specific (vulnerable hosts within an enterprise network)
Examples	Storm, Conficker	Zeus, SpyEye

Figure 2.11 A comparison between a Botnet and Mini-Botnet [7]

Mini-botnets differ from normal botnets because they are more focused on what they target and are smaller in size [7]. They are therefore more difficult to defend against. These targeted small botnets are designed specifically to not attract attention. Zeus and Spyeeye based bots are good examples of mini-botnets.

The goal of mini-botnets is primarily confidential information stealing. These capabilities are very advanced. It can steal POP and FTP credentials as well as information stored within the system such as X.509 PKI client certificates or Windows Protected Storage. Finally it can intercept HTTP forms, modify and redirect them for malicious man in the middle attacks.

They focus on specific targets to gather this information. An example of this is the Kneber botnet which belongs to the Zeus family. It targeted US based firms and had 75,000 systems under its command. It stole login credentials, social media logins, email account information and online financial information from its victims [26].

2.3.1 Phonebots

The increase in smartphone usage is bound to be noticed by cybercriminals. Smartphones have the capability to use both the GSM phone network and the TCP/IP network at the same time. This leaves an attacker that has control the smartphone two very different ways of connecting to the phone.

In late 2009, jailbroken iPhones were the target of a worm that exploited the fact many people that jail broke into their phones to install an SSH server left the username and password as a default setting. This was the first example of things to come. Increasingly, Smartphones have been the target of cybercriminals because of the amount of information that they contain and the difficulty in which to detect an infection. One simple fact is that smartphones have little anti-virus software, if any. [24]

The android open source smartphone operating system has in recent times become the most popular, this has opened itself up to lots of attention from cybercriminals³. Android's open nature is currently and will in future continue to be one of the root causes of android smartphones being exploited. Making a malicious application and letting Android smartphone users download and run it is trivial. The Android marketplace allows cybercriminals to publish applications without its content being scrutinised before its is allowed on android smartphones. This means the cybercriminals has an easy avenue of attack [44].

Discovering botnet software for the android smartphone is not really a future prediction. It is present day fact. There is already an IRC botnet for android called Foncy⁴. Its masquerades as the 'MADDEN NFL 12' game. Once it is executed it starts an IRC bot and waits for commands. Cybercriminals can do whatever they like with the phone at this point.

³http://news.cnet.com/8301-1023_3-57337389-93/androids-popularity-makes-it-open-target-for-malware-says-study/

⁴http://www.securelist.com/en/blog/208193332/IRC_bot_for_Android

2.3.2 Botclouds

Cloud computing has become a very big deal in recent years. It provides the ability for organisations to use massive scalable applications without having to host its necessary hardware. Dedicated businesses called Cloud Service Providers (CSP) typically offer both refined software services, such as databases and raw computing resources, such as storage or processing power. Customers often use these services following a pay-as-you-go model [10] .

Plenty of businesses around the world use cloud computing, but so do cybercriminals. Instead of spending time building a botnet from infecting systems, cybercriminals can spend money (usually by using an illicitly gained credit card) to buy the same resources they need to conduct their business, for example a DDOS attack. In fact the cloud computing power they buy can be used to its full capacity because it is not trying to hide its existence from users of the infected systems.

What makes botclouds hard to detect is the nature of cloud computing. It is very easy and cheap to start using these resources. It makes deployment of bots much faster. It also means they are easily replaced should they be detected and brought down. Researchers have show how simple this is. They have used 20 bots within the cloud to do a DDOS attack. The attack successfully brought down the test server within 10 seconds. Another example is conducting click fraud, researchers in TU Delft bought 1000 virtual machines and proceeded to do experiments with accessing advertisements from a 1000 different IP addresses. Neither botcloud attacks were detected or shut down by the cloud provider [36]

Even if CSP's become better at botnet detection in the cloud, there will surely be a market in the future for some bullet proof malicious cloud computing from places that have little to no anti-cybercrime legislation or law enforcement.

2.3.3 PNRP

PNRP is the acronym for peer-name resolution protocol. It maps names to IPv6 addresses, like DNS, but using a peer to peer network to publish, resolve and store the records⁵. The use of IPv6 has meant there is limited utility in the current environment however IPv6 is destined to become the defacto standard for communication over the internet in the coming years.

PNRP has two methods of operation, secure and unsecure. Secure names mode uses a private key to sign the request to register a hostname. The hostname can then be checked for authenticity against its public key. Unsecure mode allows hostnames to be made with a public key infrastructure, but these names can then be easily spoofed [15].

PNRP is already in use today. Every Windows product after XP has it enabled by default. Its main current function these days is within the remote assistant application in Windows, to provide a way of connecting to the users computer easily. The protocol generates an unsecured PNRP record and allows the person being the remote assistant to connect to the system.

Anyone can run their own PNRP server to allow systems to query for a host. At present only Microsoft provide this function but once it becomes used more others surely will follow with their own servers.

A botherder could use this protocol for effective communication between the bots and the command and control servers. It would be quite easy, as described below:

- Bots have a shared salt.
- The botmaster uses PNRP to register an hostname which is a hash of the shared salt and a time range.
- Bots calculate the hostname and resolve it.
- Once the time range is over, the hash calculation process is repeated.

⁵<http://carnivore.it/2011/03/27/pnrp>

The main functionality that this provides is an effective way of fast fluxing (which we shall look at later) without using DNS, as is currently. Using PNRP means a hostname can be registered instantly without a central authority.

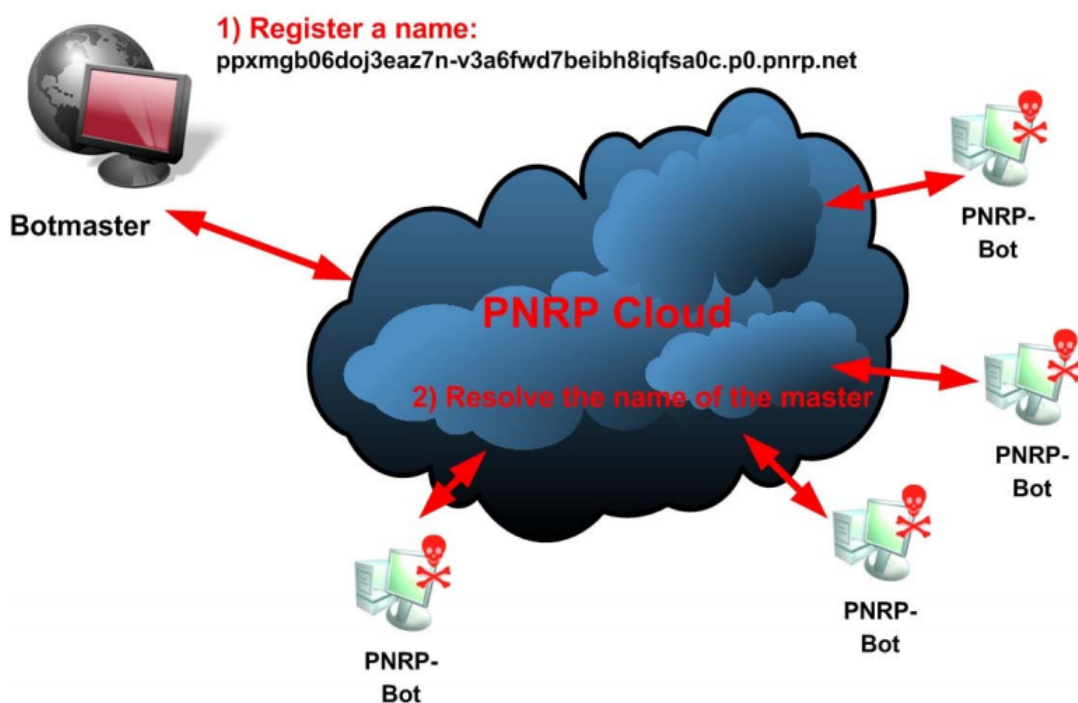


Figure 2.12 How to build a PNRP botnet [15]

2.3.4 VOIP

Skype is the defacto VOIP (Voice Over IP) technology that includes instant messaging functionality. Running over an encrypted P2P (Peer to Peer) network, it is one of the most effective network applications at routing communications through difficult to traverse networks such as NAT or heavily firewalled networks [25]. For this reason it is well suited to use as a communications medium for botnets.

There are many different ways Skype can communicate, one of the most advanced is a method called STUN (Simple Tunneling of UDP over NATs). Its works by tricking the firewalls into

accepting a tunnel to another Skype user by initiating an outbound established UDP connection. As the connection is already established the firewall will let it through. These connections are all coordinated by Skype's supernode control servers.

Skype has an API which means malware writers can develop plugins to interact with it. Botnet communication can then be routed through the already complex and highly effective network to receive/give commands. Skype has a white/blacklist method of allowing plugins to access its API. Any new skype plugin must get authorisation from the user before the plugin is used. However if the system is compromised then bypassing this check would seem trivial.

An early example of this type of threat is the Pykspa worm. It spread via Skype by spamming messages with links [25]. It installed a backdoor so that an attacker can execute commands remotely.

Other VOIP technologies are also at risk. Many of the existing voice communication networks in organisations these days are through VOIP. Internet accessible VOIP servers can and will be attacked in the future to provide attackers with easy money from routing the calls towards expensive toll numbers they can profit from [33].

There are already tools designed to probe VOIP servers, SIP primer and the Sipvicious tool⁶ are examples. Sipvicious can scan for PBX servers with phone extensions that have weak passwords. Then it can abuse these accounts to make phone calls for "free". Most cases so far (eg \$1 Million in one year) of VOIP attacks are from not so technically advanced attackers, one can only wonder what will happen when more advanced attackers join in.

⁶<http://code.google.com/p/sipvicious/>

2.3.5 DNS

DNS stands for Domain Name System. It is one of the most used parts of the internet. Its function is to translate user friendly hostnames into an IP address that a computer can understand. When a user wishes to visit a website for example, the hostname of that website must be translated into an IP address by a DNS server. Without a DNS server, web browsing would be impossible and emails would not be delivered because they wouldn't know which IP address to use. Now that we



Figure 2.13 FBI's explanation of the DNSChanger trojan [31]

know of DNS's importances, lets look at some examples of its malicious use to gauge its future role in the internet. Cybercriminals know that DNS is responsible for what is essentially trust in a DNS server getting the right IP address to a user. Its this trust, that if interfered with can allow cybercriminals to send users to any website they want.

A good example of this is the DNSChanger botnet. Its core function was to redirect users to

malicious DNS server which would send certain queries to malicious websites. In effect, a user wanting to visit an online banking website would be redirected to a malicious copy of that website. Its is also important to note that it would delete itself after it had changed the DNS setting. Leaving users believing they were no longer infected [22]. The FBI in late 2011, brought the botnet down after 3 years of investigations. The DNSChanger botnet shows that attacking the way users trust DNS is a worthwhile method of attack [31].

Intercepting DNS traffic is just one way cybercriminals are using DNS to conduct malicious activities. DNS can also be used as a communications medium to connect to its Command and Control (CnC) server. In Appendix B, there are two examples of how DNS traffic can be used to send other traffic, such as commands or malicious payloads. Within a home/small office network, DNS traffic is rarely ridiculed, so this traffic will probably go unnoticed.

DNS will continues to play an important part in the workings of the Internet. As such, it will continue to be used for malicious means.

2.4 Trends

In this section, we will look at the trends that the rest of this chapter has presented.

2.4.1 Going mobile

Everything is going mobile. Organisations want to give their employees the flexibility to work on the move, while they wait, basically anywhere. The increased prevalence of wireless communication such as 3G, has helped this new trend. With fast communications infrastructure anywhere an employee can move reliably work anywhere.

Also, employees like the idea of being able to work when they want. Waiting for a doctor appointment? Working on ones mobile device while waiting will soon become the new norm.

However, there is this small matter of security.

Mobile VPN's are going to become very popular in the future (to prevent eavesdropping). Allowing an employee to work anywhere with confidential data will lead to quite a lot of security risks. Securing the data on the mobile device will also become an issue that will need to be solved with data encryption. However, VPN's and encryption are process heavy work and will take lots of battery power, but the continuing increase in battery strength and faster processors will quickly help make this mobile office a reality.

2.4.2 Increased user awareness

In the past, user awareness of IT security issues was very low. Most users didn't know what an anti-virus software package was. Nor did they realise how the internet worked or how they might get infected. Progressively, with users education in enterprises increasing the threat of users mistakes leading to malware has been reduced but not eliminated⁷.

The trend of users becoming more aware of their environment, teaching each other about IT security and knowing more about the current threats will lead to a securer computer environment for all the users.

2.4.3 Greater risks, greater losses

Security is not usually the first thing people developing applications on any type think of including. It is usually something that is patched into the application after functionality is almost completed in the product. This leads to greater risk of a security vulnerability, and the constant need to patch in the application. This trend of patching (fix it later) methodology will continue to plague the IT industry for a long time. What we are talking about here is simple economics, if a competitor can bring out a product before another by sacrificing some security and then gains market share. If

⁷<http://www.techrepublic.com/article/increase-user-awareness-to-bolster-security/5543500>

is has succeeded in gaining market share, then it is normal to then think about security. However patching security will always lead to problems as it is directly tied to functionality. More security means less functionality. So the trend of more risks will in the authors opinion continue because firstly it makes economic sense. Lastly, security costs money. Money that some that know little about security will prefer to spend elsewhere.

Next, the trend of greater losses is destined to increase. In recently times data leakage losses at have increases expediently, TJX lost a phenomenal 45 million customer credit card records in 2007⁸ Heartland Payment Systems lost 130 million credit card records⁹. Finally in 2011, Sony's Playstation network was attacked and cybercriminals got away with 70 million customer records¹⁰

In the banking sector, fraud losses show no sign of decreasing or even slowing. According to the Dutch Banking Association (NVB), throughout 2010, the loss whole sector lost EUR 9.8 million. However, in only the first half of 2011 the losses where 11.2 million euro. Clearly there is a disturbing trend of increased losses to fraud, in most cases by botnets.

2.4.4 Bigger/better protection applications

The rise in advanced malware such as Stuxnet and others will lead the IT security to develop newer, faster and more sophisticated security applications for the protection of the users and data. There are many in development but it will always be unclear which will work and make a difference to security.

Some see the need to process more data as a future requirement. For example, many systems that protect networks could in the future become overworked with the sheer volume of traffic that will be seen in the next few years. This inevitably means that more traffic usage will need to

⁸http://news.cnet.com/TJX-says-45.7-million-customer-records-were-compromised/2100-1029_3-6171671.html

⁹http://news.cnet.com/8301-1009_3-10146275-83.html

¹⁰http://news.cnet.com/8301-31021_3-20057577-260.html

analysed for malicious content or attacks.

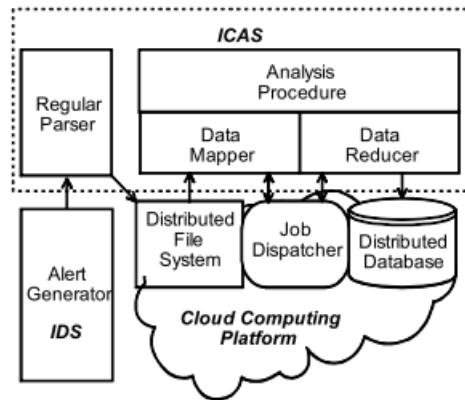


Figure 2.14 IDS Cloud Analysis System (ICAS) [7]

The solution to the need for extra processing capacity, some think is in cloud computing or grid computing. The fact cloud computing can upgrade or degrade its processing capacity is a fundamental advantage to its use as a security application.

Using the Apache Hadoop framework for example, could allow for the instant capacity upgrade needed during peak times for monitoring software such as Snort, from [41]. A system proposed by Chen et al, allows for Snort to be run on top of the Hadoop framework. This would mean a very scalable, fast and robust way of processing traffic to be inspected.

Chapter 3

Challenges

"As soon as we started programming, we found to our surprise that it wasn't as easy to get programs right as we had thought. Debugging had to be discovered. I can remember the exact instant when I realized that a large part of my life from then on was going to be spent in finding mistakes in my own programs."

- Maurice Wilkes discovers debugging, 1949

This chapter is all about the challenges that we face in eliminating the threat of botnets and malware in organisations. We will look at the practical, legal and technical issues that prevent or limit the IT security industry, government and law enforcement from dealing with cybercriminals. The sub question *What challenges are currently being faced in defending against botnets?* is answered here.

3.1 Endless amount of vulnerabilities

Many threats are based on vulnerabilities or bugs in software that can be exploited. After decades of discussion and initiatives on how to reduce this relentless amount of bugs there is still no real

solution on how to solve the problem. The result of the attention is that the security of operating systems and core network protocols has improved significantly. However, Cybercriminals have just moved their focus to the application layer, where it is much harder to secure (or focus on) as there is an enormous number of applications in use.

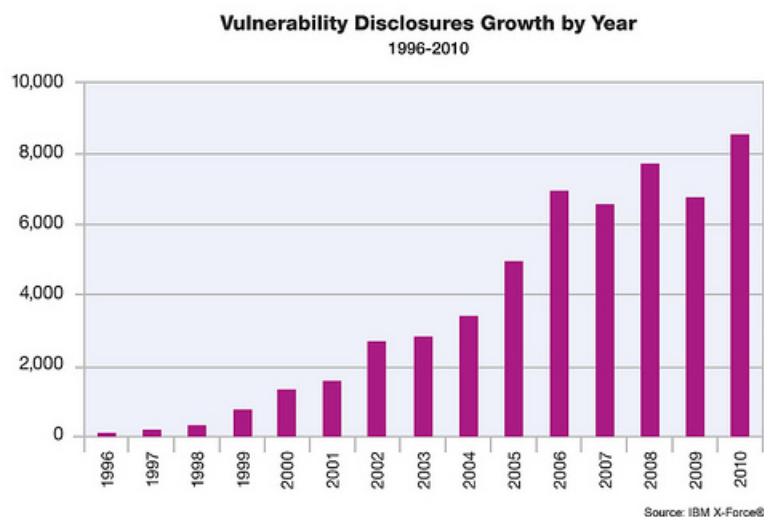
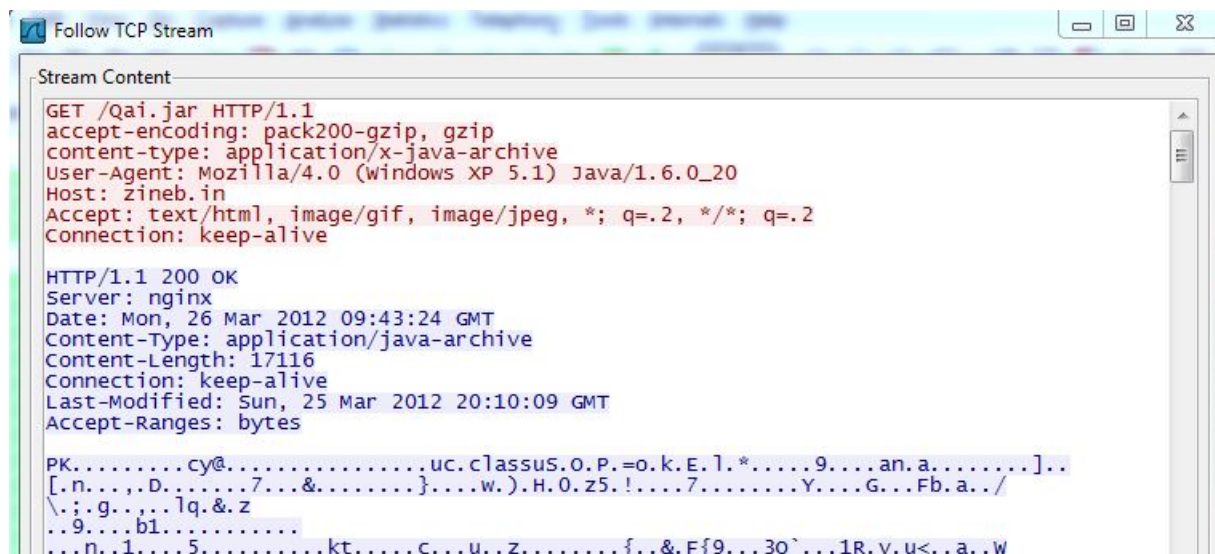


Figure 3.1 The increasing trend of more vulnerabilities [IBM 2011]

In section 2.4.2 (Greater risks, Greater losses), the economic reason for sending out vulnerable software is described. It is simply summarised as the market rewards the product that works well and is in the market first and can gain market share. In Microsoft, in the 90's there was a saying that we "ship it Tuesday and get it right by version 3". This philosophy really reflects the core reason for the endless amount of vulnerabilities in the world today.

An excellent example of a current vulnerability is the Java virtual environment. Cybercriminals will typically gain access to a legitimate website and add some JavaScript so that each user is (unconsciously) redirected to a malicious Java applet. With Java installed on billions of computer systems, a percentage will have an old vulnerable version.

The current version of Java 6 is Java/1.6.0.33, anything less than that (as seen in Figure 3.2) will mean that once the user goes to the blackhole site, the vulnerable Java version will be



```
Follow TCP Stream

Stream Content

GET /Qai.jar HTTP/1.1
accept-encoding: pack200-gzip, gzip
content-type: application/x-java-archive
User-Agent: Mozilla/4.0 (windows XP 5.1) Java/1.6.0_20
Host: zineb.in
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Connection: keep-alive

HTTP/1.1 200 OK
Server: nginx
Date: Mon, 26 Mar 2012 09:43:24 GMT
Content-Type: application/java-archive
Content-Length: 17116
Connection: keep-alive
Last-Modified: Sun, 25 Mar 2012 20:10:09 GMT
Accept-Ranges: bytes

PK.....cy@.....uc.c\lassuS.O.P.=o.k.E.l.*.....9....an.a.....]..
[.n....D.....7...&.....}.w.).H.O.z5.!.....7.....Y....G...Fb.a./
\.:g....lq.&z
..9....b1.....
...n.1....5.....kt....C...u..z.....{..&.F{9...3Q`...1R.v.u<..a..W
```

Figure 3.2 An example of a system with a vulnerable Java version. Note the version of Java/1.6.0_20

exploited, and therefore the computer system. This is just one example of a commonly used vulnerability that Cybercriminals use. There are many others such as Adobe Flash and Reader, Microsoft Office and Internet Explorer.

Making sure vulnerable programs are fully up to date is a difficult problem to solve because the users aren't aware that the versions of the software they use are at risk. Have you checked what version of Java you are running? I checked recently and found that I was using an old vulnerable version. Needless to say, I was a little bit shocked I haven't updated my Java. It will be something I'll be sure to keep an eye on in future.

3.2 Sophistication

There are many good examples of how malware has become more sophisticated, the most obvious one being the progression from simple computer worm to highly advanced botnets as explained in the previous chapter.

The challenges of increased sophistication are huge. The cybercriminal is always one step ahead because it is he or she that makes the next piece of advanced malware that must be defended against. So in that way the IT security industry is always playing catchup.

Lets look at an example of a very advanced botnet family called TDSS. More specifically its latest version called TDL-4. It uses a range of methods to evade signature, heuristic, and proactive detection, and uses encryption to facilitate communication between its bots and the botnet command and control center [37]. It also has a powerful rootkit component that allows it to hide from sight.

Some have called it the 'indestructible' botnet because of its sophisticated nature. It has a custom encrypted communications protocol that allows it to controlled by its command and control servers. This custom communications channel makes detection by network traffic analysis almost impossible.

Another smart feature of this botnet is its ability to detect other botnet software and remove them. This helps the TDL-4 arousing suspicion in the system. Its 'anti-virus' can detect around 20 malicious programs by searching the registry and filenames. It can also blacklist the IP addresses of known command and control servers to prevent any contact to them.

Finally, the TDL-4 botnet masters has a backup way of contacting its bots. Each bot TDL-4 has access to the KAD P2P network for when normal encrypted communication fails or if the botnet master wishes to use it for transferring files such as stolen databases. Having another means of communication via the KAD network means that even if all the TDL-4 command and control servers are put offline the botnet masters can still control their bots. It places network redundancy into the botnet.

The point that is being made is simple, Cybercriminals are making and will continue to make ever increasingly sophisticated malware.

3.3 Law

Cybercriminals activities generate many legal difficulties that, society normally believes will be dealt with, within the framework of the law. However bringing cybercriminals to justice is not as easy as society would hope. Key to the problem is that fact cybercriminals on the internet commit crimes within many jurisdictions. The internet provides a level of anonymity and lack of traceability that favours the illicit activities.

Law enforcement is ill equipped to stem the rapidly rising tide of cybercrime [13]. There are many problems that they face in this international world of cybercriminality. There are barriers to international cooperation, outdated laws that were written before the internet became prevalent and lack of will from government to recognise the issues at large.

Most countries have legislation to deal with cybercrime¹. These laws are described as making it unlawful for anyone to access a computer system or data without permission. Botnet software is never authorised by the users of the system, making it illegal.

The problem with national legislation is that it differs from one nation to another. So one specific malicious action may be unlawful and lawful in another. Even within the European Union (EU) coming up with a concise way of combating cybercrime is difficult. In 2001, the Cybercrime Convention was signed by the EU and other countries. It was the first international treaty on crimes committed via the Internet. Its objectives are to harmonise the the cybercrime laws, provide procedures to investigating cybercrime and allowing speedy international co-operation between nations. To date, the convention has not been widely ratified into law by many of the countries that have signed it.

Another challenge faced is the legality of accessing systems without approval for the propose of notifying the user of an infection or uninstalling some malicious software [22]. The IT security industry is limited to actions within their own computer networks without permission. It is a legal

¹<http://www.cybercrimelaw.net/Cybercrimelaws.html>

grey area as to what is allowed, so the industry proceeds slowly.

Another legal grey area is government involvement in cleaning computer systems without the owners consent. The best example is the Japanese governments development of a "good" virus. Japanese lawmakers must create new legislation allowing for its use because it would appear to violate current Japanese law².

The virus itself has the ability to identify the source of a cyber attack with a high level of accuracy, then replicate itself from computer to computer, cleaning up viruses across the network. However, getting permission to intentionally allow a virus onto a network is a very difficult to get. There are many examples of people of good intent making viruses for good, but failing in their goals³.

One solution to allowing it being used is to limit the virus to Japanese IP addresses, but this will open up legal challenges from users that are using proxy software to pretend they are in Japan, for example.

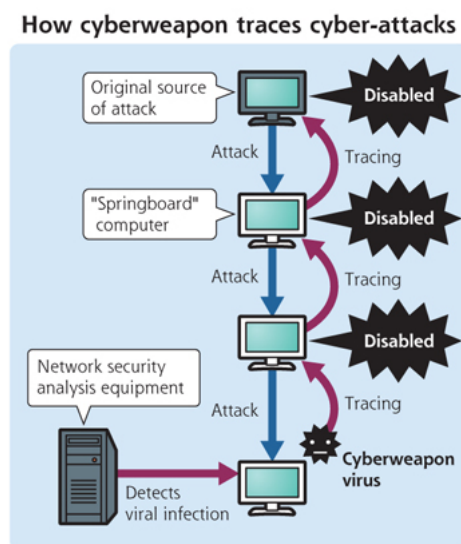


Figure 3.3 Japans "good" virus

²http://www.huffingtonpost.com/2012/01/04/fujitsu-cyberweapon-japan_n_1183462.html

³<http://www.people.frisk-software.com/~bontchev/papers/goodvir.html#SomeBadExamplesofBeneficialViruses>

3.4 Trust on the Internet

Users like to believe they can trust their computer and the internet that they use everyday. The computer is a very sophisticated device and most users have no idea how it works. This uncertainty causes fear. The only way to overcome that this fear for a user is to believe that the computer is always in the users control and wouldn't start doing unexpected things such as email a password to someone.

Trust on the internet also stems from ignorance. Users trust the emails that arrive from friends email accounts that have been compromised. Users trust companies that have been had their website emulated and are fooled into using this copy. This unequivocal trust in the internet is something that the designers of the internet had not taken into account.

The challenge for the IT security industry is to find a way of building trust into the internet. Such a task is not easy. The Diginotar certificate authority hack [27] (which allows cybercriminals to forge certificates for authentication) has shown that placing trust in one place is not a good idea.

A certificate authority's (CA) job is to authenticate users and companies to each other by using electronic certificates. A CA will have its own certificate called the root certificate. This must be heavily guarded as it is a trusted entity. The CA job is to validate that a user or company is who they say they are, and then sign their certificate with the root certificate.

If a cybercriminal has gained the ability to sign any certificate with the trusted root certificate, then any one will trust this new (malicious) certificate. As was shown with the Iranian fake Goggle certificates that were signed, so someone could easily decrypt Gmail traffic. The Diginotar hack showed that it is easier for an attacker to attack a CA than the underlying cryptographic methods that are used to make and sign certificates. In other words, cryptography is so (seemingly) unbreakable these days that its just easier to attack the CA to gain ones objectives.

3.5 Users

"Companies spend millions of dollars on firewalls, encryption and secure access devices, and it's money wasted, because none of these measures address the weakest link in the security chain."

- Kevin Mitnick

The above quote really explains the challenges users present, they are always the weakest link. The best security systems worst flaw will always be humans. An organization's IT assets are ultimately managed and operated by humans [6]. Even though the security of an organisations infrastructure such as mission critical servers is usually handled by trained staff with knowledge in IT security, the responsibility for end of the line security at workstations is usually the users.

There are many famous stories about the stupidity of users. In London, at Liverpool street station random commuters were asked to write down their username and password in exchange for a chocolate bar. An astonishing 70% of people would reveal their password under these circumstances⁴ Family names, pets and football teams were all used by those questioned to provide inspiration for a password, so there can be a high degree of certainly they were real passwords.

User education has long been the most widely acknowledged countermeasure to this threat [17]. If a user can be trained to identify threats, remind and watch over others about these threats then users and organisations networks and data will be much safer.

However the user education has its limits. Even the best trained and knowledgeable individual can be fooled by a sophisticated attack. So the challenges of users will always be there, the only thing that can be done is to migrate the threats to users.

⁴<http://news.bbc.co.uk/2/hi/technology/3639679.stm>

3.6 Cost vrs Risk

"If you spend more on coffee than on IT security, you will be hacked. What's more, you deserve to be hacked."

- Richard Clarke

IT Security cost money. Unfortunately IT security is not something you can easily measure, it doesn't produced anything physical like manufacturing a product. Nor does it show it is working well, because if an organisations security is working well there will no security incidents. When there are no incidents, people are likely to let their guard down.

In a typical (large) organisation, there is a Chief Security Officer. He prepares an annual report on the organisations security to the board of directors. They sign off on it and how the information security policy should work. However, it really depends on the organisation approach to security if it is effective. A security policy that is not respected is worse than have no security policy at all.

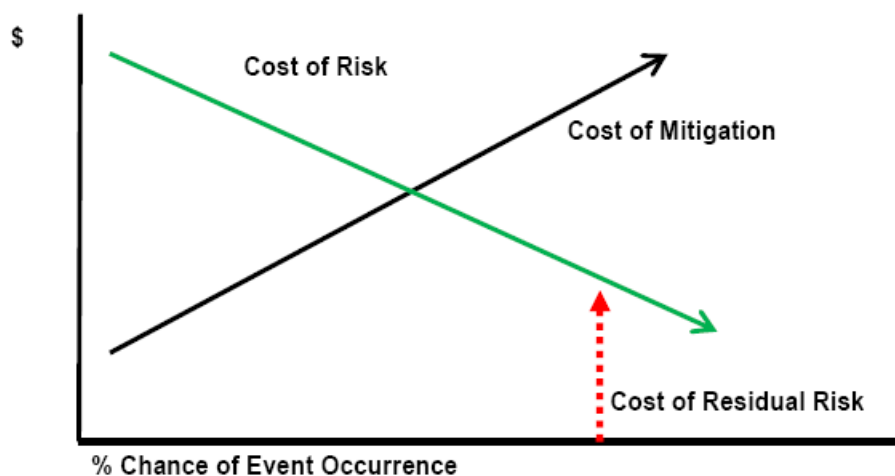


Figure 3.4 Migration Cost Vrs Chance of Incident Occurrence [ERE-Security 2010]

Consider the impact on an organization if it does not adequately mitigate risks. In the end, how an organization approaches security depends on its appetite for risk. A healthy dose of paranoia is

warranted here. After all, the stakes are extremely high⁵.

There is a direct correlation between the cost of security for an organisation and the risks that are migrated by spending money on IT security. The real challenge is coming up with a viable balance between cost and risks that are acceptable.

⁵<http://www.gideonrasmussen.com/article-07.html>

Chapter 4

Botnet in depth

"A botnet is comparable to compulsory military service for windows boxes"

- Bjorn Stromberg

In this section, we look at botnets in depth. Most of the knowledge in this section is from the author himself as well as data from Fox-IT found while working there. This chapter does not answer a specific question, instead it serves to explain many of the elements of botnets had are used in further chapters.

4.1 Mechanics

Let us start by giving a clear definition of what a botnet is. A botnet is a collection of Internet connected computers that have had software secretly installed that allows a remote controller to issue commands. These are typically sent encrypted. All the computers in the botnet (known as Zombies) can be used to send spam emails, or launch DDOS (Distributed Denial of service) attacks on websites or other servers. These computers are compromised, and are being used without their owner's knowledge. The malicious software they run is referred to as 'bot'.

The bigger that botnet the more valuable it is. To make the botnet bigger cybercriminals try to recruit new bots into there botnet. They use increasingly devious methods in trying to do this recruiting. There are three ways a botnet can be formed/recruited:

- **Automated Infection**

The botnet will be commanded to seek out other systems on their network for exploitation. It will do this by focusing on known/unknown vulnerabilities, misconfigured software and systems with weak passwords to spread. The scanning nature of bots on a network resemble the nature of a *worm*.

- **Requiring user actions**

Users can be tricked into installing the bot software via many different methods. Getting the user to install the software makes behaving like a *trojan*.

- **Rent or buy from the Cloud**

This in the newest method of propagation. Cybercriminals will use stolen credit cards (or other illegitimate means) to buy resources on the Cloud. This can be defined as a *Business to Business* (B2B) transaction.

4.1.1 Stages

There are 4 stages to a botnet infection. Firstly the initial infection **(1)** of the botnet software onto the system. Then the botnet software will contract its controller **(2)**, the botmaster. The botmaster will then typically be given commands to execute **(3)**. Lastly, the botmasters commands are executed **(4)**.

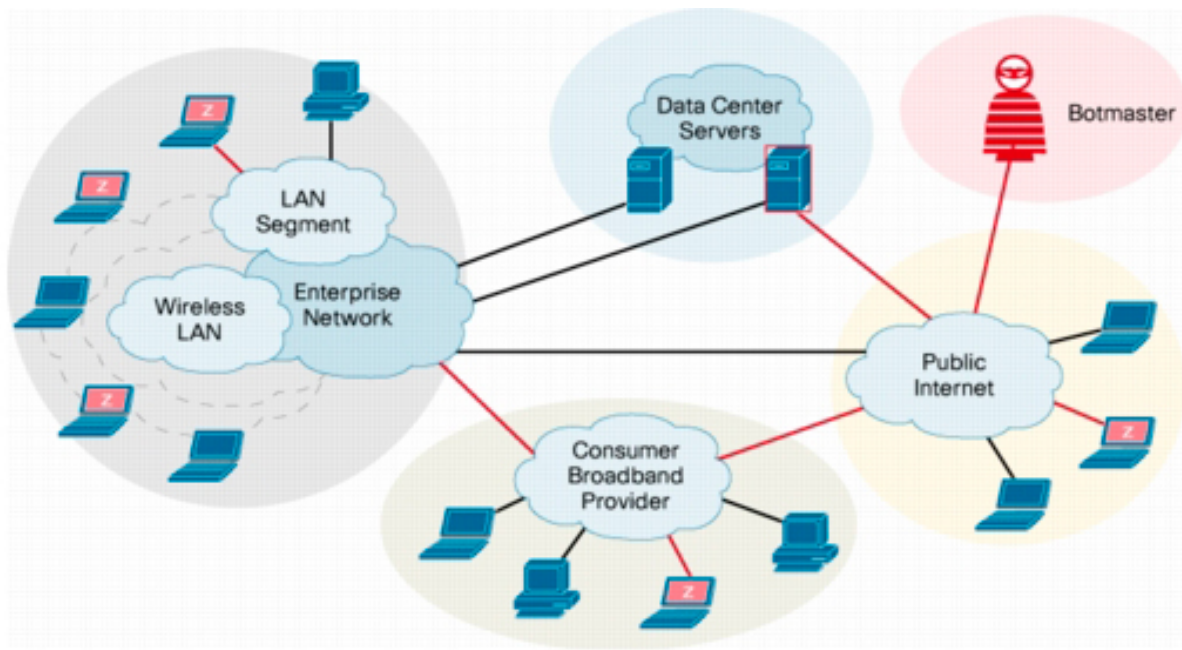


Figure 4.1 A typical Botnet [12]

1. Initial infection

A system can become a bot by many different methods. As previous stated, there are 3 different ways of setting up the botnet. The worm, trojan horse or B2B approaches are all used to make a botnet.

2. Contact botmaster

Communication between the bots and botmaster is an essential part of the botnet. Most attackers would like the ability to rapidly send instructions to bots but also do not want that communication to be detected or the source of the those commands to be revealed [8].

As we shall look at later in section 4.1.3(Communication) there are many different architectures, protocols and methods of communication for the botmaster to use. These are all being further developed to escape detection.



Figure 4.2 The stages of a botnet infection, an illustration of how a compromised system might be used to send spam [Wikipedia <http://en.wikipedia.org/wiki/File:Botnet.svg>]

3. Download payload/instructions

Once the system is infected and within the botnet it waits for commands. These commands all come from the botmaster. The botmaster is effectively in control of the system to do whatever he wishes. Once the commands are sent the botnet will do whatever its actions it has been instructed to do. Typical commands are to download and execute a payload that will provide some function such as listen to keystrokes.

4. Malicious actions

The bots will execute these commands or malicious software its has been instructed to by the botmaster. We shall look at the different commands or uses of the bots in the next section.

4.1.2 Uses of a botnet

Bot software is usually self-sufficient and modular. This means that a bot that can be told to do one action today but can be requisitioned to do a completely different action tomorrow. Making a bot modular allows the botmaster total control of what the bots can do. Any new use that is required can be just downloaded in a new module for the bots. As we can see there are many uses to bots:

1. Distributed Denial-of-Service Attacks

Using the many bots in unison to connect with a specific service such as a webserver can cause the service to crash or deny legitimate users for that service.

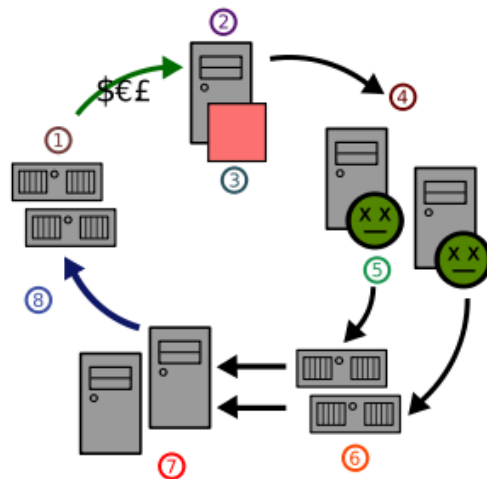


Figure 4.3 The typical lifecycle of spam that originates from a botnet: (1) Spammer's web site (2) Spammer (3) Spamware (4) Infected computers (5) Virus or trojan (6) Mail servers (7) Users (8) Web traffic

2. Spamming

Most of today's e-mail spam is sent by botnet zombies. It allows the spammers to send massive amounts of emails from legitimate IP addresses.

3. Phishing

Botnets can be used to host phishing sites for fraud, allowing them to use various IP addresses. This makes them harder to detect and take down. Once one is taken down another one can be instantly brought online in its place.

4. Identity Theft

With complete control of the system, any valuable information that is stored there can be used by a cybercriminal for identity theft. Typically all this information is packaged into an encrypted file and uploaded to a server for later use.

5. Spreading new malware / botnet growth

Growth of the botnet is important. Locating new systems for infection are another use.

6. Sniffing Traffic / Keylogging

Sniffers or keylogging can be used to retrieve sensitive information like usernames and passwords from the computer system or network [18].

7. Proxy

Anonymity is a very important concern for botmaster. Bots can be used to proxy connections around to prevent anyone tracking where he is.

8. Host Warez and others illegal content

As recently seen in the Megaupload takedown¹, there is a lot of money in controlling and

¹<http://www.bbc.co.uk/news/technology-16642369>

hosting content that users want. Warez (pirated software), pornography, and other illegal content sites can be used to gather advertisement money.

9. Click fraud / Installing Advertisement Addons and Browser Helper Objects (BHOs)

The exploit code may imitate a legitimate Web browser user to click on ads for the sole purpose of generating revenue (or penalizing an advertiser) for a Website on pay-per-click advertising networks (such as Google Adwords) [12].

10. **Scareware** Scareware scares people. Its used as a tactic to extort money from botnet controlled systems. The tactic is to use social engineering to cause shock, anxiety or just a threaten towards to the user to force them to pay money as a fine or to buying some software.

PRS for MUSIC

METROPOLITAN POLICE

Your computer has been locked.

Illegally downloaded music pieces (pirated) have been located on your computer.

By downloading, those music pieces were reproduced, thereby involving a criminal offense under **Section 106 of the Copyright Act**.

The downloading of copyrighted songs via the Internet or music-sharing networks is illegal and is in accordance with **Section 106 of the Copyright Act** subject to a **fine or imprisonment for a penalty of up to 3 years**. Furthermore, possession of illegally downloaded music pieces is punishable under **Section 184 paragraph 3 of the Criminal Code** and may also lead to the **confiscation of the computer**, with which the files were downloaded.

Your IP-Address: _____
Your Hostname: _____

You can be clearly identified by resolving your IP address and the associated hostname.

The pirated material has been encrypted and was moved to a protected folder to prevent further damage.

To unlock your computer and to avoid other legal consequences, you are obligated to pay a release fee of **£50**. Payable through our payment partner Paysafecard. After successful payment, your computer will automatically unlock.

Failure to adhere to this request could involve criminal charges and possible imprisonment.

To perform the payment, enter the acquired Paysafecard code in the designated payment box and press the "Submit" button.

The PRS for MUSIC is legitimized by law - and is in close contact with the legislators and the Metropolitan Police.

Unlock computer

Code: value: £50

Submit

Enter your Paysafecard code using the PIN-pad

1 2 3 4 5 6 7 8
9 0 back

Where to buy Paysafecard

Additional Information

paysafecard is available from 350,000 sales outlets worldwide, in the United Kingdom, exclusively from all PayPoint outlets

1. Ask the merchant for a £50 Paysafecard
2. Receive your Paysafecard code
3. Enter the Paysafecard code using the PIN-pad

paysafecard
pay cash. pay safe.

Figure 4.4 A UK locale scareware notice that appears on an infected system. [2]

A commonly used tactic is convincing the user that their computer has a virus, and that the new anti-virus program they have can clean the virus from the computer, if and only if they

buy the product.

4.1.3 Communication

There are two different architectures that botnets can have: centralised and decentralised. Each have advantages and disadvantages. We will look at real life examples of different botnet architectures and how they are used.

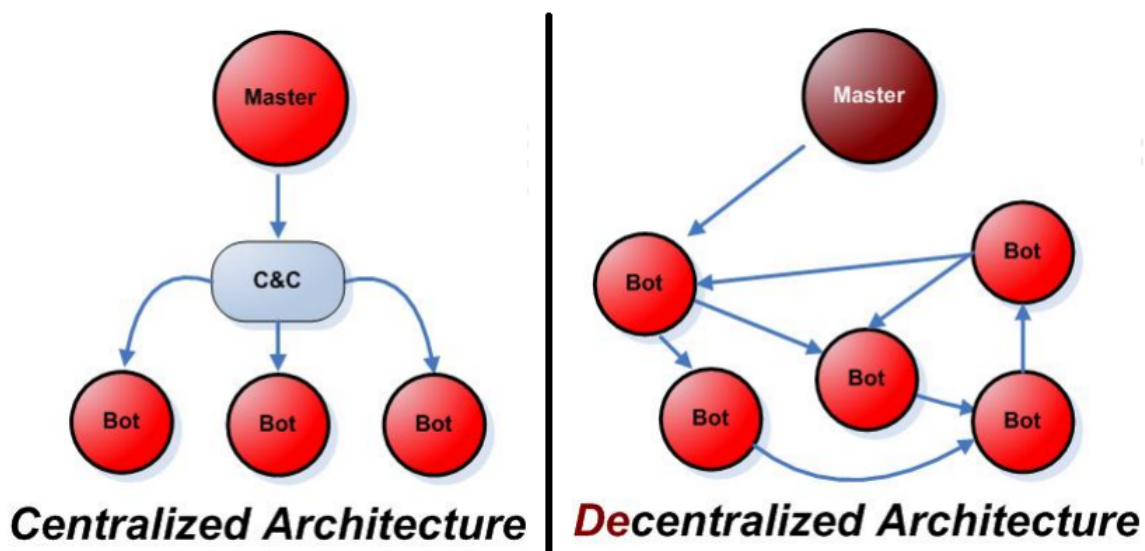


Figure 4.5 The difference between centralised and decentralised architectures [19]

P2P

Peer to Peer (P2P) C&C communication allows the botnet not have one central point of failure. It is a decentralised architecture in which each bot in the network can connect with others to retrieve commands and pass them onto others bots.

The best example of a P2P botnet is the Storm botnet. It was one of the biggest botnets every built. In 2007 there were reports of that the botnet has grown to around a million bots². It was used

²<http://www.networkworld.com/news/2007/080207-black-hat-storm-worms-virulence.html>

mainly to send spam, do DDOS attacks or be rented out to other cybercriminals. Its dominance has seen fallen, but it remains a good example of a P2P botnet.

One disadvantage of having a P2P botnet is the ease of which IT security firms or law enforcement can join the botnet and therefore study it for weaknesses by adding a sensor to it³. Also the sophistication of the network mean there is lots more maintaining of the computer code needed. It is this sophistication that leads to bugs in the code that can be exploited to take the botnet down or in the eyes of another cybercriminal, take over someones elses botnet. A good example of this is Stormfucker⁴, which takes advantage of flaws in Storm's command network. Storm bots only use a four-byte XOR challenge, so communication is trivial to emulate and replace with other commands.

IRC

IRC stands for Internet Relay Chat, it is a protocol designed for real time chat communication. Control over these bots is usually based on sending commands to an IRC channel setup by the attacker, used by then bots. Of course, bot administration requires authentication and authorisation, so that only the owner can use them. It offers a simple method to control hundreds or even thousands of bots at once in a flexible manner.

Very popular in the early years of botnet development, its use has been in decline with better detection methods and policies such as blocking IRC traffic from within a network. Its centralised architecture also made it very vulnerable to actions by law enforcement to stop its use. It is very rare to see an IRC botnet these days, so we will not look at an example botnet.

³http://www.securelist.com/en/blog/654/Lab_Matters_The_threat_from_P2P_botnets

⁴<http://hackaday.com/tag/stormfucker/>

HTTP

Another centralised method of communication is HTTP on port 80(sometimes called the 'always on' port). Its use as a method of botnet communication has been popular because HTTP traffic is very widely used for the internet and it cannot be filtered or blocked easily for fear of obstructing legitimate traffic.

There are two ways the HTTP protocol are most commonly used to communicate, GET and POST methods. GET is a request for a specific piece of data from a webserver. POST is a simple way of sending data to a webserver. Cybercriminals however are known to use GET requests to send data as well.

- An example of a malicious GET request:

```
GET http:freedownload3.comscreenhot4s89_3278.gif?sv=153&tq=gwY92w4AoyB%2B%2BI7heKU4yG0c7N4Fh3ntHooyjSy8N7DhVAVFabOGYh3XCOvqRFhUbTzsn0A0AYJ7InnBPCEdkMLcrpdrwFG37YvSlpVZqR5T%2BCadlHgE%2BOazffjOn%2B%2FdQAEtE6Xzyv9zgbUgDzJFNvKjGG%2Fce41PCSaFtuwMB58hW5%2BUSOmm53F2HZCc88OW4XaiHNNHq%2B3emHkB%2F5ehwsvyyLkcYqR04aVqL7gDWSyKKclDphthrLf5g5w1WQ%2FN9RWzJxtOYlwqV8DucxCOF
```

The above value tq is obviously some sort of encrypted value that is passed to the CnC server via the HTTP GET request.

- An example of a POST request in which the infected client will send data about itself to a CnC server to check into the botnet:

```
POST http:www.smk4mslnwa.c0m.libackupzhl9.phpbn1=11-000-1087_7875768F6522DF69_24&sk1=46B8DB635FA735E0F8B2BB63C479C4ED
```

We see there are two values passed to the CnC server: bn1 and sk1. They could be anything, but the most likely explanation is that they are a unique identifier and some other identifier.

Fast Fluxing Describing HTTP botnet communication would not be complete without explaining what fast fluxing is. The goal of fast-flux is for a fully qualified domain name (such as `www.example.com`) to have multiple (hundreds or even thousands) IP addresses assigned to it. These IP addresses are swapped in and out of flux with extreme frequency, using a combination of round-robin IP addresses and a very short Time-To-Live (TTL) for any given particular DNS Resource Record (RR)⁵.

Below we can compare a normal web browser communicating directly with a typical website against the case of a single-flux service network, where the end user's browser communication is proxied via a redirector called a flux agent. When a victim believes that they are browsing `http://flux.example.com`, their browser is actually communicating with the fast-flux service network redirector which redirects the requests to the target website⁶

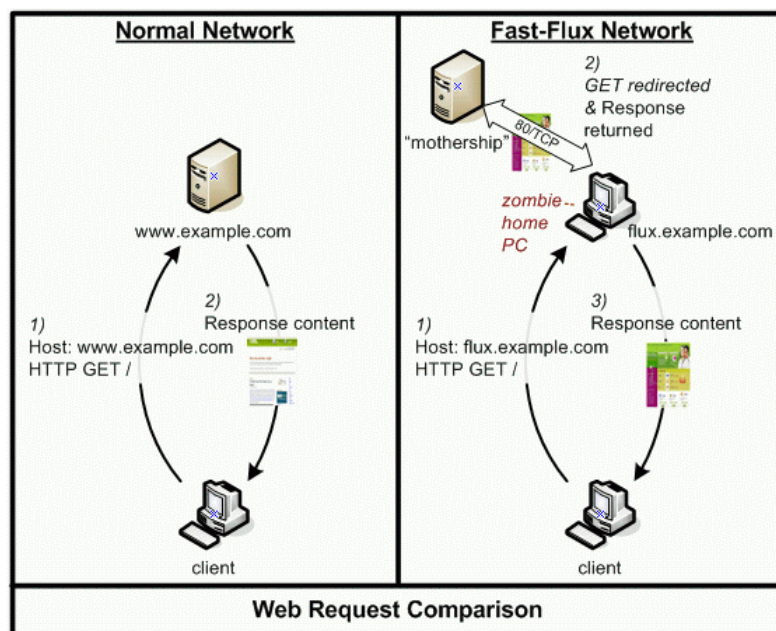


Figure 4.6 The difference between a normal network and a fast flux network [Honeynet Project 2008]

⁵<http://www.honeynet.org/node/132>

⁶<http://www.honeynet.org/node/134>

DNS

DNS as carrier for botnet CnC traffic seems to be getting popular. Concerning its usage as botnet CnC, DNS has not really been seen so far in the wild. Additionally, in typical network environments DNS is usually one of the few protocols (if not the only one) that is allowed to pass without further ado⁷.

One example of a botnet using DNS for CnC communication is Feederbot. Feederbot uses valid DNS syntax for its DNS messages. Messages from the CnC server to the bot are transmitted in the rdata field of a TXT resource record [14]. Using the RC4 stream encryption scheme and encoding the communication into unicode allows the DNS request appear legitimate.

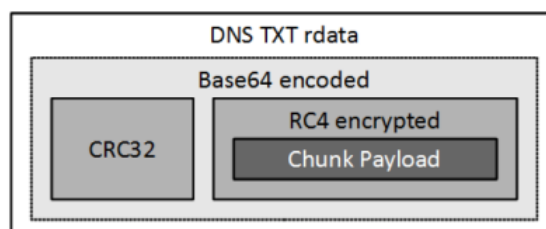


Figure 4.7 Structure of a Feederbot DNS CnC Message Chunk [14]

Others

There are of course other ways of communicating within a botnet. One option would be to write a custom protocol that others don't know about [28]. However this approach might lead to an easily fingerprinting or analysed for weaknesses the signature. As is the case of the YoyoDDos botnet (DDOS specific botnet) which has a custom communication protocol. After analysis, it was determined that it uses a weak obfuscation scheme that appears to be a hard-coded substitution table. So de-obfuscated is trivial.

Another option is to convert botnet communication to image steganography⁸. It is quite simple,

⁷<http://blog.cj2s.de/archives/28-Feederbot-a-bot-using-DNS-as-carrier-for-its-CC.html>

⁸<http://www.irongeek.com/i.php?page=security/steganographic-command-and-control>

modify an image in such way the information one would like to convey is able to be extracted. Place the image online somewhere the bots can download it and let them extract information from it. The bots can of course follow the same process.

We have already discussed the use of the Skype protocol for communicating with bots in chapter 3.

4.2 Organisation

Like any money-driven market, botnet developers operate like a legitimate business: they take advantage of the economic benefits of cooperation, trade, and development processes, and quality. Recently, botnets have begun to use common software quality practices such as lifecycle management tools, peer reviews, object orientation, and modularity. Botnet developers are selling their software and infection vectors, providing documentation and support, as well as collecting feedback and requirements from customers [11].

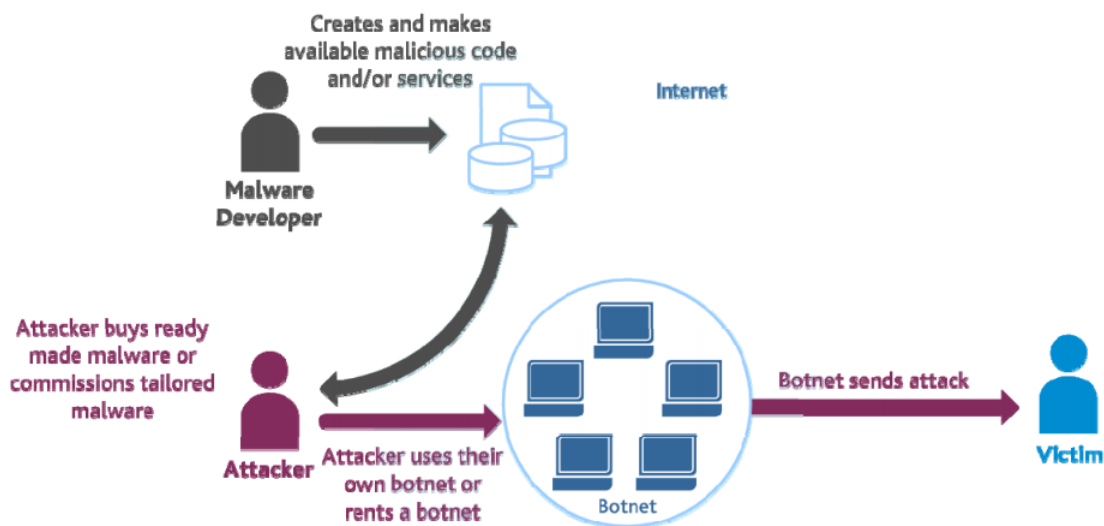


Figure 4.8 The lifecycle of botnet attacks

It is called the underground economy. The old system was that the malicious software writer

was also the controller of the botnet it created. Following this system, a whole in-depth specialised economy in which each participant has a specific role developed. This new system involves many different types of people. Each loosely working together to compromise computer systems, steal information and profit from that information.

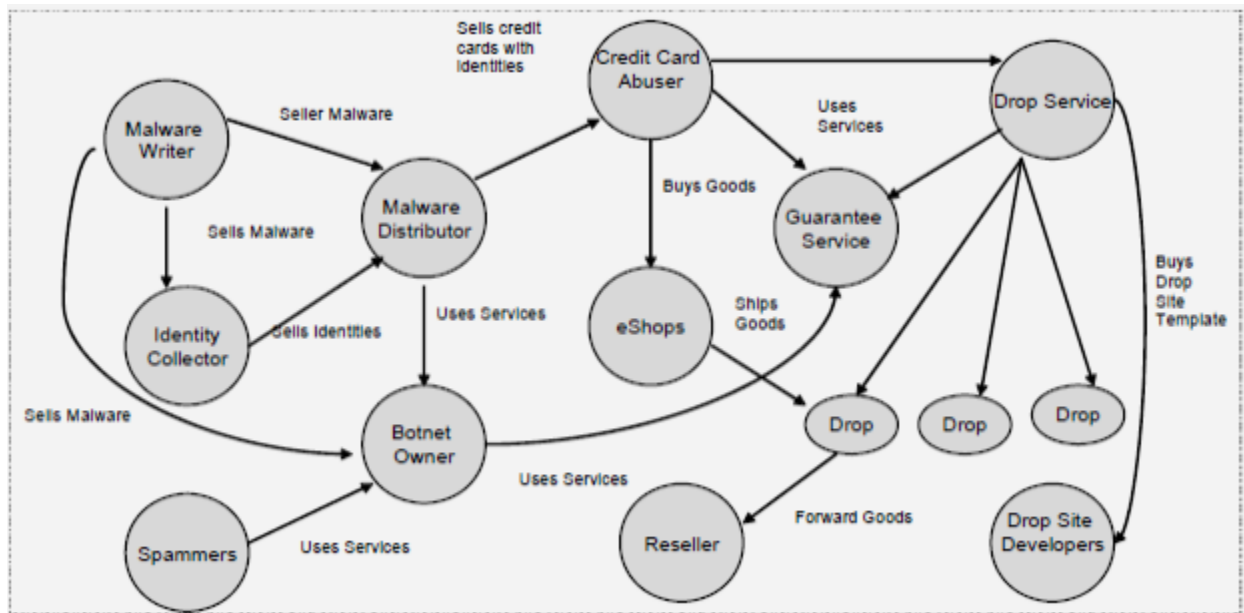


Figure 4.9 Division of labour in the malware underground economy
Visibility of malware vs. malicious intent [40]

The malware writers will develop new vectors of attack and package them into software that can be sold/licensed primarily through Internet Relay Chat (IRC) channels, underground bulletin boards, and online forums. Cybercriminals will use them to get the newest software that they need to attack systems. Once the cybercriminals have gained access to systems and valuable information that they can profit from they send the money they get to a money mule controller to be cleaned. These mules specialize in turning this illegally acquired information into money, be it from stolen credit cards or identity theft. Stolen credit card information, for example, may be used to make purchases for people known as "drops." These drops, in turn, post the acquired merchandise on eBay or sell it immediately for cash [40]. The money mule will send the money back to the

cybercriminal minus his commission, of course.

One malicious organisation that has recently become known is the 'factory' of Stuxnet, Duqu, Flame and the Guass APT malware. There is little know about them, except for the they are highly skilled and patient in their approach towards their goals.

4.3 Actors

There are many different actors that need to be considered when looking at botnets. Each have a roles to play in the lifecycle of botnet infections. The users get infected, the malicious actors control the infections, the vendors try to stop the infections, ISP's allow vulnerable users onto the Internet and governments try legislate the issues around botnets. Security decisions by one player have consequences on many other players. For example, individual users may face financial expenses to protect their machines against malware but not experience the costs of failing to do so as these costs are incurred by other users who are the targets of malware and/or spam sent from the infected machine.

Lets look at each actor in detail:

End users/Organisations End users or the organisations that they work for are always the target of attack. The onious however, is on the user to protect their computer system from malicious software by keeping a vigilant eye on what they do on the internet. An organisation itself cannot protect itself without the help of the end users (employees). A malware infection can have a significant impact to both the user and the organisation they work for, in the event their work computer is infected while they use it.

Personal computer users spend a significant money every year on anti-virus software and fire-wall software to protect themselves from online threats. They have in effect created an industry that is designed to protect them.

Malicious actors Cybercriminals. The bad guys. The ones that are always one step ahead. There are many different ways to describe the malicious actors on the internet. Their goal is always the same, insert malicious software onto system. Then exploit them for profit. There are a wide variety of these malicious users:

- **Security researchers** that seek out problems in systems to improve their security. Technically their actions are malicious even though they mean well.
- A wide variety of people from **organised crime**.
- Some malicious require specialised services such as bullet-proof hosting, these **malicious service providers** allow the underground economy to grow.
- **Insiders** that have experience with the systems they wish to compromise.
- **Script-kiddies** that try to replicate a simple attack to other systems in an automated way.
- Others without real malicious intent such as **Amateurs** seeking fame not profit but still conduct malicious actions.

Vendors There are two types of IT security vendors: hardware and software. Hardware vendors are mainly concerned with selling to organisations, heavy duty equipment such as firewalls or VPN modules, that can handle massive amount of data. Software vendors sell such products as anti-virus software and personal firewalls to both personal users and organisations.

They are stakeholders in the catch up game of protecting users against the latest threats. They profit from the need for protection from online threats.

ISP's Internet Service Providers (ISP) provide the internet to end users, organisations and malicious users. The main problem they face is spam. Every spam email must use the infrastructure that the ISP's have built. According to Message Labs, a security service provider, the overall proportion of spam intercepted in 2007 was around 84.6 percent of the total number of emails [40]. Some ISPs have started blocking port 25 to reduce outgoing spam messages, with limited success.

They are in the unfortunate position of controlling the gateways to the internet. This puts them in a unique position to counter threats (such as using a DNS sinkhole, which we will look at in the next chapter). They are also in a dangerous position of potentially censoring users traffic, which would be illegal in many countries.

Government Finally, governments have multiple roles to play in the malicious software environment. They (their employees) are targeted by attacks that could potentially lead to such events as data loss.

They also have a part to play in enforcement both at the law enforcement level and the legislative level. Law enforcement tries to catch the cybercriminals. Government legislation tries to empower law enforcement and govern online threats.

Chapter 5

Botnet Detection and Prevention

"The only truly secure system is one that is powered off, cast in a block of concrete and sealed in a lead-lined room with armed guards"

- Gene Spafford

This chapter will explain the many options available for detecting botnets. We will look at how malware (used to create bots) is collected, analysed and tracked. How DNS traffic can be used to detect botnets. Also, using honeypots we can infiltrate botnets or simply wait to detect attacks as they occur. Software and hardware such as IDS's and firewalls can limit a botnets impact on a organisation. In the end of this chapter, we will look at how security policies can be use to migrate the risk of botnets, and how audits can close open security holes. Finally, we will present how a botnet infection, once it is detected can be eradicated.

This chapter will among other things, look at two sub research questions:

- *How can DNS servers and their data/records be used to detect botnets in organisations?*
- *How can honeypots be used to detect botnets in organisations?*

5.1 Detection Basics and Malware Tracking

In this section, we will look at the basics of how to detect malicious activity. There are two ways, Signature based and Anomaly based. We will also look at how infections are tracked worldwide and how the increased collaboration is leading to some interesting ideas on how to prevent malware infections and their communications.

5.1.1 Signature based detection

A signature based system is simple, a signature is a detailed collection of information that comprises of all the details of an infection or malicious communication. This can be the any type of information such as filenames, a specific sequence of bits, a URL, anything that is specific to something an IT security professional would like to catch. With this information, a detection engine such as Snort can tell that this signature matches on traffic it has analysed. This will more than likely mean there is an infection on the same network as Snort is protecting.

```
alert tcp any any -> any any
(msg:"ZBOT settings.bin retrieve";
flow:established,from_server; content:"filename=settings.bin"; http_header;
classtype:trojan-activity)
```

Here is a simple snort signature how to catch settings.bin (from Figure 2.6) which is one of the filenames that Spyeeye uses for its configuration files. Lets look at it in detail: The *alert* keyword means that snort should trigger an alert if it sees this traffic. It should only look at TCP traffic. It will look at *any* IP address on *any* port going to *any* IP address on *any* port. Once it triggers it will log it as "ZBOT settings.bin retrieve". There must be a full TCP handshake or *flow*. The traffic must have the content filename=settings.bin in the *http header* only. Finally this alert is classified as *trojan activity*.

Anti-Virus software

Anti-virus software is based on the signature based detection system, but they are sometimes also include an anomaly based detection engine, which we will look at in the next section. Anti-virus companies are always a step behind the malware developers. They must wait for the malware to be released into the wild before they can make a signature for it.

An anti-virus company will use many different methods to get a malicious executable, so that they can analyse it and make a signature from it. Honeypots, user submitting websites such as virustotal.com, IT security researchers and companies all provide this vital information to them.

5.1.2 Anomaly based detection

Many anti-virus packages include a heuristics engine for detecting anomalies in the computer system they protect. This allows the anti-virus company to gather information on new malware or new strains of older malware.

Heuristics is the use of different methodologies, technologies, tricks, rules or techniques to make an educated guess as to whether a file is infected by malware or not ¹. It has extensive knowledge about malware. If a file starts jumping around reading important system files or starts to write itself to another folder then the heuristics engine will flag it as suspicious.

Unfortunately, heuristics isn't perfect. The Melissa virus for example, was completely missed by anti-virus heuristic engines. Another problem with heuristics is how bad things can go wrong with them. A file that has been deleted (or quarantined) because the heuristics engine thinks its malicious and in reality it is not, can have disastrous consequences for an organisation. Just look at the example of a web administrator that 'lost' his website because of a bad configuration of his anti virus that used a heuristics engine.

¹<http://www.thehackademy.net/madchat/vxdevl/papers/avers/paper141.pdf>

5.1.3 Zeus/Spyeye Tracker

One way to find information on botnet traffic in the wild is to use a malware tracking website. They consists of a list of known Zeus and SpyEye CnC domains and IP addresses. To produce these lists, the tracking website employs automated systems, called Trackers, which analyse various feeds and identify malicious CnC domains, IPs, and servers. These malicious communication points are updated on an ongoing basis, and are displayed on the tracking websites: <https://zeustracker.abuse.ch> and <https://spyeyetracker.abuse.ch>.

These websites are very useful, and can be referenced when investigating suspicious network traffic for IP addresses that are probably malicious. Any computer connecting to these IP addresses can be flagged for investigation, and should be a suspected as a bot.



Figure 5.1 From Spyeye Tracker, 443 Spyeye CnC servers detected December 2011 [1]

These tracking websites have become quite a nuisance to botnet controllers². These websites are a regular target for the DDOS attack, as they threaten the very existence of their botnets. Spyeye botnets have even tried to disrupt this tracking by adding legitimate domains to the botnets configurations files.

²<http://blogs.rsa.com/rsafarl/spyeye-botmasters-fight-back---targeting-swiss-security-sites-spyeye-tracker/>

This disruption is by means of sending that all the credentials collected by the SpyEye bots, including screenshots, username and password combinations, and stolen certificates and cookies, to port 443 of the legitimate websites, such as google.com. In this way the data on the tracking website can be tainted.

5.2 Malware collection

This section is about malware collection, and how organisations can collect malware to help protect themselves.

5.2.1 End users

The best place collect a sample of malware will always be the nearest to its intended target, the end users. Let's take anti-virus software, most anti-virus companies develop free versions of their software. Both parties gain from this arrangement, the user is protected by the software and the companies have their software installed in millions of computer systems. All these systems also have a heuristic engine³ (within the anti-virus software) as well to generate generic signatures. They can be used to identify new viruses or variants of existing viruses by looking for known malicious code, or slight variations of such code, in files. Statistics and other information is then sent back to the anti-virus company. The anti-virus companies gain up to date signatures for their free and paid customers from both customers data.

Of course, anti-virus software is not a perfect system. It can be detected on a computer and attacked. Its detection engine can allow be outsmarted with methods such as polymorphism and malware that loads at boot time. That is why user education is important. Many users do not see the obvious signs that their computer is infected, such as the computer slowing down or that their

³<http://www.thehackademy.net/madchat/vxdevl/papers/avers/paper141.pdf>

homepage is now something different that what it was before. When these signs of infection are there, the user should be prompt in trying to deal with the issue.

5.2.2 Submitted malware

Another way of malware (signature) collection is from online scanning websites that sandbox submitted files for analysis. Anyone that wishes to know whether a file is malicious or not can submit it to one of these online sandboxes and get the results. An obvious example of this is Virustotal⁴.

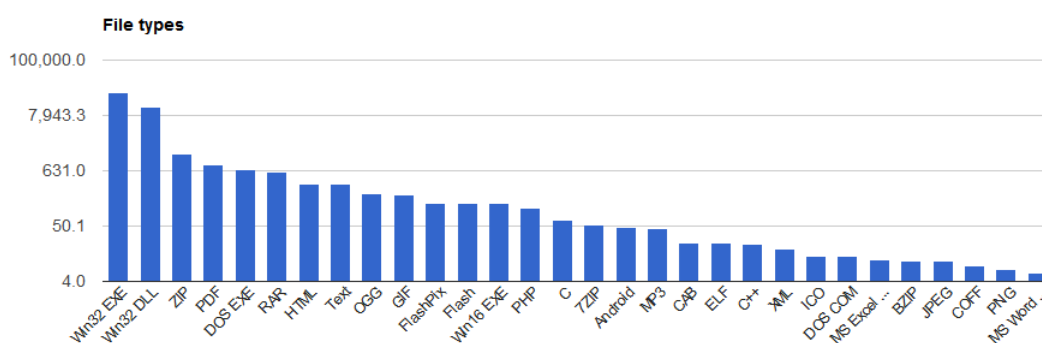


Figure 5.2 An incredible amount of files sent to Virustotal in 7days

Virustotal scans the files with 43 different up to date anti-virus software packages. The results that are returned are never typically a 100% detection rate. However if no packages find a problem with the file; there can only be two conclusions, either it is malware free, or it has some malware that has not been detected yet (0-day vulnerability). It is very rarely the latter.

5.2.3 Binary analysis

It is common practice for IT security professionals to use sandbox's in their work. There are lots of advantages to sandbox's. They are able to execute the (malicious) code in a safe environment.

⁴<https://www.virustotal.com/>

A detailed log of how the code is executed can then be shown to the sandbox user to determine if the code is in fact malicious. Importantly, the domains or IP addresses the malware is trying to connect to can be used elsewhere, such as a blacklist,

Cuckoo⁵ is just one of the many sandbox's available. After feeding it an executable file, this open source sandbox server can trace the performed relevant win32 API calls, dump files and network connections created, as well as trace the assembly instructions that the possible malware executes. More importantly it can be used to automatically make signatures or reports for submitted files, which might be gathered automatically from another system such as a proxy server. This could possibly be a very powerful (automated) way of protecting an organisation from malicious software.

⁵<http://www.cuckoo.org/>

5.3 DNS

In this section we will talk about DNS, how it works, how botnets use it and how to defend an organisation by using it data. We will choose to explain DNS here for consistency and ease of reading, as there are many complex parts of DNS that will be referenced later in the section.

5.3.1 DNS overview

Domain name system(DNS) is one of the core services on the Internet. This system is in place to organize and identify domains. It is used to map these domains into IP addresses, something machines can understand. Similar to looking up someone in the phonebook, DNS is used to translate a domain such as `www.google.com` to the relevant IP address such as `74.125.19.147`. With that information the computer system can connect to the systems at that domain and communicate with them.

DNS records for domains are used to store these IP addresses. A DNS record can have many different options, here are some of the main ones:

1. MX Record, tells where to direct mail when its being sent to the domain.
2. TXT Record, contain arbitrary text that can be used by other protocols to provide other services. Also, as we shall see it can be used for malicious communication and tunnelling.
3. CNAME Record, links a domain name to another canonical domain name.
4. A Record, are the main records used to tell which IP address to send data to.
5. NS Record, hold information on the DNS server for that domain.

Botnets rely on DNS to support their CnC communication infrastructures. Botnet controllers typically change the domain they use with either a fast flux network (were one domain changes its

DNS record every 5 minutes) or by using a Domain Generation Algorithms (DGA) to automatically generate a new domain for the botnet to use as a basis of communication.

5.3.2 Malicious uses of DNS

As DNS is the one of the main elements of the Internet, bad guys cant help but use it. However, they're uses of DNS are not normal and therefore this is a weakness can be used against them. The goal for any malicious software is to have DNS agility, which is obtained by using new domains daily to frustrate detection efforts. Lets looks at some of the ways malicious actors use DNS services:

1. Drop zones
2. DNS spoofing
3. (Fast) Fake advertising sites
4. CnC communication

A dropzone is a publicly writable directory on a server on the Internet that serves as an exchange point for stolen data. Massive amounts of confidential data are stored there. To avoid detection the server typically changes the DNS records of the IP address that it uses. The botnet controller can still simply login to the server to retrieve the data, and in the hope he can remain anonymous.

Some botnets use DNS spoofing to trick the DNS server accepting a domain's IP address as something they control. Many bots can be told to flood one DNS server with the same incorrect information to poison the records. Another different way to do this is to override the Windows host file. This is a feature of the Simda bots, it changes the DNS resolution of a website URL to a particular IP address. This way, what looks like a simple search with Bing in the network logs becomes a bot check in ⁶. Such as <http://www.bing.com/chrome/report.html?9oC7s=%9B%EE...>

⁶<http://www.microsoft.com/security/portal/Threat/Encyclopedia/Entry.aspx?Name=Backdoor:Win32/Simda.F>

Very smart.

The DNS infrastructure can be used by botnet masters to redirect unsuspecting victims to infected websites, alter user searches, replace ads, block legit anti-virus software and promote fake security products. They can earned millions of dollars displaying false advertisements and redirecting users to wrong websites, using the CNAME part of DNS records.

Lastly, DNS can be used as a tunnel to get information to and from the bots in a very covert way. DNS cannot be blocked (as it is a core service), therefore bots can encode data or send their requests over a DNS session. An example is seen in figure 5.3, the bot sends a request for a DNS TXT to a sub-domain that holds its request and the malicious DNS server will respond with its command. Alternatively, the bot could send data in tiny chunks in requests to the malicious DNS server as requests, and it would send an acknowledging response.

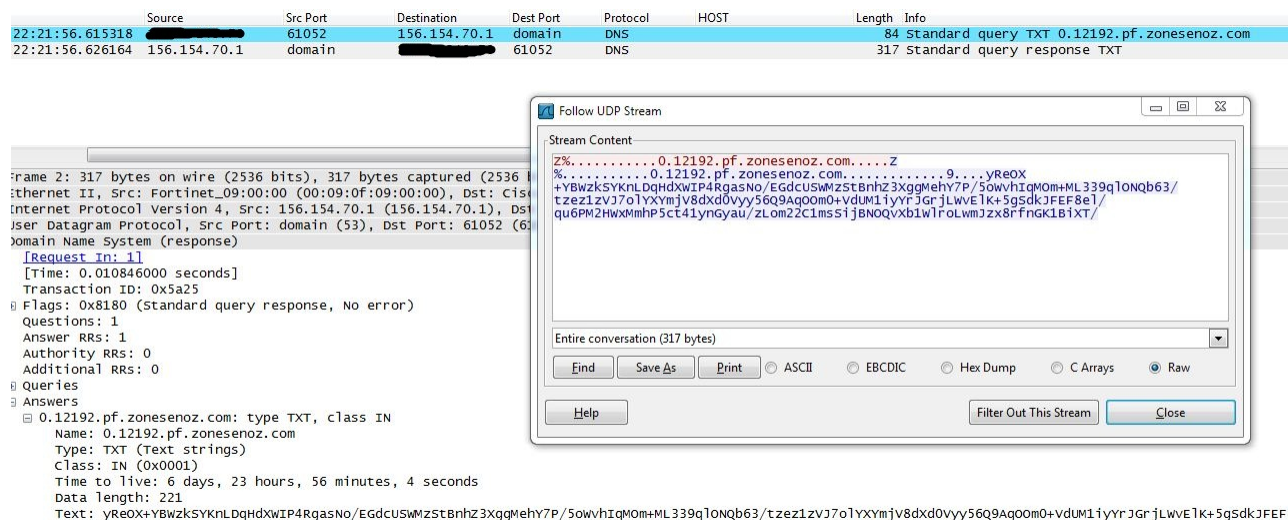


Figure 5.3 Example of botnet communication via DNS. Note, the very long TXT record

5.3.3 Current Anti-botnet technologies

In this subsection, we will look at how DNS can be used to detect botnets in organisations. DNS monitoring is passive, scalable, and doesn't require deep-packet inspection (DPI) to work. It can also keep track of multiple botnets simultaneously from a central location, the DNS servers. It is however limited to only working on botnets that use DNS in its communication infrastructure, for example some P2P networks wouldn't use DNS.

Blacklisting

Blacklists are collections of malicious information such as malicious domains which should be blocked. Used by many organisations to stop the spread of malicious software or communication, these lists are provided by IT security companies or organisations. The most typical type of blacklist is a DNS-based Blackhole List (DNSBL).

DNSBL don't store static lists on devices to monitor, the devices that use the list simply send a request to the DNSBL provider to ask if a certain domain or IP address is on the list or not. So if the domain google.com needs to be checked then the DNSBL request will be google.com.somelist.org. The response will state if it should be blocked or not.

In an attempt to evade domain name blacklisting, botnet masters now make very aggressive use of DNS agility, to nullify its effect, the most common being fast flux networks. The static nature of the list means that it can't keep up with these dynamic communications. However, static lists still serve a purpose. Previously discovered malicious communication can be found and reported.

DNS Sinkhole

The idea of a DNS Sinkhole is that if you know about some bad domain names that malware might use or that users might try to visit, you can create an authoritative "Zone" files on your organisation's DNS server. This allows you to return any result for any domain name or host you want to [20].

The obvious choice for this traffic would be sending it to a server that logs and reports the systems on the network that connect to such malicious domains and investigate.

There are benefits and limitations to DNS sinkholes. They are very easy to setup and administer. Open source software can be used in combination with free malicious domain lists, to provide a solution without massive capital investment. The one big limitation is that all DNS outbound traffic must be routed through the organisations DNS server, however this is a best practice and should already be in place. The perimeter firewall should also block all DNS queries that don't originate from the authorised DNS servers.

Lets look at how a DNS sinkhole works in practice. In figure 5.4, we can see the client is sent a malicious domain to resolve by email, but this happen in other ways such as a website link, or if it has malware installed. The client asks the organisations DNS server for an IP addresses of that domain, however the DNS server will reply with an internal IP address to a logging machine instead of the malicious domain. Sometimes, instead of redirecting the traffic to the logging machine, the response will simply be to the client itself (127.0.0.1) to prevent it from making a malicious communication outside the organisation. Once the incident is logged, the client can be investigated and disinfected if needed.

The Korea Information Security Agency (KISA) has shown that DNS sinkholes can be an effective weapon against botnets. South Korea is one of the most Internet connected countries in the world. This level of connectivity brings with it botnet traffic that needs to be contained. Using DNS sinkholes at a country level has meant that they could tackle this issue [3].

By using Honeypots, malware analysis, incident professionals and other sources, they built a regularly updated DNS file for the the South Korean ISP's to use. All the traffic that was directed towards these domains was then sinkholed.

Sinkholes are also a major source of intelligence for the IT security industry. If a bot is receiving commands from its CnC server but all these commands are monitored by an IT security

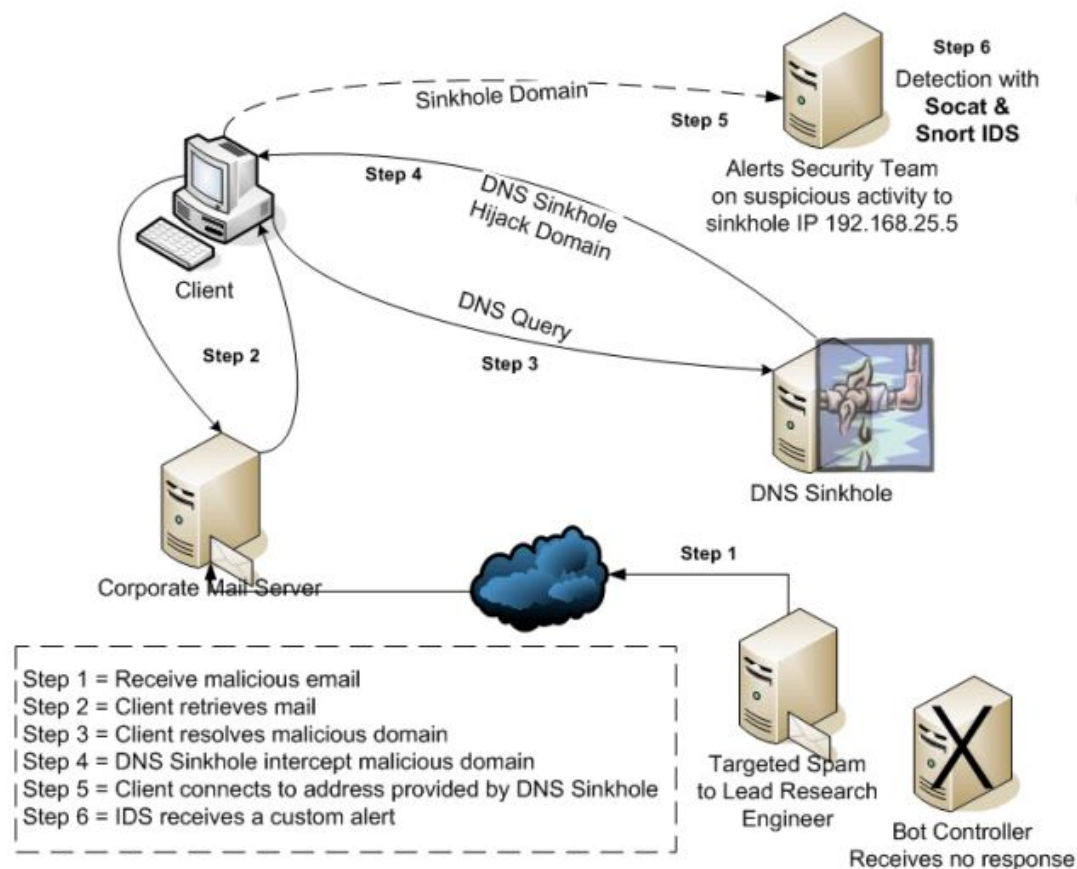


Figure 5.4 An advanced DNS Sinkhole [20]

company then, there will be lots of knowledge gathered without the bot owners knowing. The traffic could be completely blocked or let through and logged.

DNS tunnel prevention

People are constantly developing new ways to tunnel using DNS. The one way that this risk can be mitigated however is to force systems to only use a certain DNS server, that can then forward the DNS requests for them. This allows for easy analysis of the DNS traffic from one central location. If a client is making 10 DNS requests for the same domain every second, then there is a very good chance it's a DNS tunnel.

A typical DNS tunnel will use the TXT record of a domain to ask for its next command or to send data. Appendix B contains an example of a DNS tunnel made made by a user, and Figure 5.3 also has a botnet DNS tunnel example. Both can be easily detected as they generate a massive amount of TXT record requests and responses. There are legitimate uses for the TXT record (such as SPF, which prevents email fraud) but not with massive amounts of requests and responses need by a tunnel.

Dynamic reputation systems

A dynamic reputation system for DNS is a collection of DNS data (and other data) that is used to provide a score for a new domain. The goal is to automatically assign a low reputation score to a domain that is involved in malicious activities, such as malware spreading, phishing, and spam campaigns. Conversely, assign a high reputation score to domains that are used for legitimate purposes [4]. Notos and Kopsis are such systems, and can be used to identify previously unknown malware domain names several weeks before they appeared in blacklists. Notos is used to monitor a local RDNS (recursive DNS) servers, and Kopsis is used to monitor the authoritative name servers.

Network resources used for malicious and fraudulent activities inevitably have distinct network characteristics because of their need to evade security countermeasures. By identifying and measuring these features, Notos can assign appropriate reputation scores.

An example of how Notos works:

- A device on the network looks up a domain, lets say ilovekittens.com.
- The authoritative DNS response comes from a free DDNS provider in China.
- The domain name points to a residential, DHCP assigned IP address, in Tehran.
- By looking up our database of DNS requests and responses we can see, that there are 8 other domain names that have pointed to that particular IP address over the last 30 days.

- Five of those domain names have been referenced within previously captured malware or blacklisted as CnC servers.

The point here is that one single DNS request by a device on a network will generally never be interesting. However if that request is checked against a dynamic list of other DNS requests then some interesting information can be taken from it. Such as its a new domain, and its the first time anyone has requested it, even that fact alone should raise suspicions.

From the attackers point of view, domain names are the key to DNS agility. Domain names are significantly cheaper than IPv4 addresses; so malicious actors tend to reuse address space with new domain names.

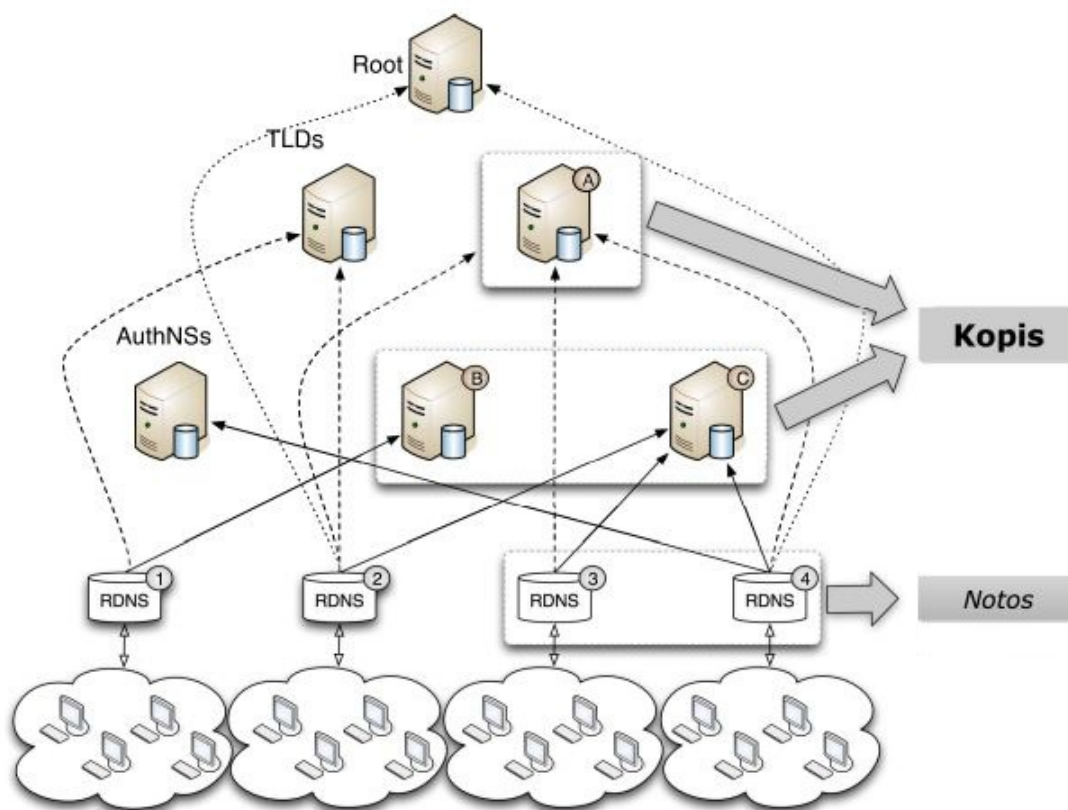


Figure 5.5 The difference between Notos (which monitors recursive DNS resolvers) and Kopsis (which monitors authoritative name servers [5])

Kopis works a little differently, it analyses the streams of DNS queries and responses at AuthNS or TLD servers from which are extracted statistical features such as the diversity in the network locations of the RDNS servers that query a domain name, the level of "popularity" of the querying RDNS servers and the reputation of the IP space into which the domain name resolves.

The system leverages the global visibility obtained by monitoring DNS traffic at the upper levels of the DNS hierarchy, and can detect malware-related domains based on DNS resolution patterns.

For example, if a new domain is suddenly requested from lots of RDNS servers then it will get a lower reputation score. There are three different features that are used in combination to give the new domain its score:

- Requester Diversity
- Requester Profile
- Resolved-IPs Reputation

Requester Diversity maps the requesters (RDNS servers) and gets their country code and autonomous system (AS) numbers, called such as RIPE or APNIC. Using this location information, statistics such as average or standard deviation are computed. The reason this is useful is that the distribution of the machines on the Internet that query malicious domain names are on average different from the distribution of IP addresses that query legitimate domains. Malware related domain names will have a diverse pool of IP addresses looking them up in a systematic way (i.e., multiple contiguous days) [5]

Requester Profile is used to add weight to the RDNS servers that service a large client population. By giving an ISP network with a large user base a higher weight, the score reflexes the reality that smaller networks such as an academic network usually have better protection.

Resolved-IPs Reputation simply tries to recognise a malicious domain from its historical records. If the domain points to an IP address space that is known to host lots of malicious activities then it will get a lower score. While an IP address that is hosted on a well known, well run network, malicious activity is less likely.

Some of the real work applications for this technology are impressive. Such as during the researchers work they labelled 225,429 unique RRs requests which corresponded to 28,915 unique domain names. 1,598 domain names were labelled as legitimate and 27,317 domain names were labelled as malware-related. That basically means that 15 out of 16 new domains are malicious, quite an interesting statistic.

5.4 Honeypots

To learn which vulnerabilities adversaries are using, we could install a computer system on a network that has no other purpose other than to wait for traffic that shouldn't occur. If the system then receives any traffic towards it, that traffic should be considered malicious. Detecting this malicious traffic can help us to discover the attackers methods. The system we use to detect these attacks is called a honeypot. More precisely, a honeypot is an information resource whose value lies in unauthorised or illicit use of that resource [34]. It can be considered an intrusion detection system (IDS), which is detailed in the next section, however they are given their own section.

A good example of a honeypot is the Kippo SSH honeypot. It is a very simple idea, once there is a SSH connection on port 22 (the typical SSH port), Kippo will pretend to be the SSH service. Its a type of virtual jail that only allow the intruder conduct certain actions, such as download malicious binaries or scripts. However when they try execute the scripts, it will give them an excuse such as Segmentation fault. This honeypot can be used to detect new malware as well as give an early indicator of a breach in security. As it serves no purpose this honeypot should have

no connection towards it. However, if there is a scan for hosts on the (internal) network that have SSH running, the source of the connection (the bot/infected system) can be identified.

5.4.1 Low vrs High interaction

	Low-interaction honeypot	High-interaction honeypot
Degree of Involvement	Low	High
Real Operating System	No	Yes
Risk	Low	High
Information Gathering	Connections	All
Compromised Wished	No	Yes
Knowledge to Run	Low	High
Knowledge to Develop	Low	Mid-High
Maintenance Time	Low	Very High

Figure 5.6 The differences between a high and low interaction honeypot

Traditionally honeypots can be put in two groups: low and high interaction. Low interaction honeypots only emulate the services that are present on a real computer. They are cheap, easy to setup and maintain, unfortunately the information they receive is not in depth and cannot be used for analysis. It simply a log of malicious connections.

High interaction is a real system. Using an actual system brings its challenges, as its expensive to maintain and setup. The big advantage of this type of system though is that actual attacks can be monitored and analysed.

There is a third group of honeypot, a medium interaction honeypot. The kippo SSH honeypot is a good example of this type. Medium interaction means that the service is emulated, but is still interactive. Attackers can interact with the emulated service, thinking its a real service.

5.4.2 Malware collection

There are many types of honeypots, all typical services such as FTP, SSH, HTTP and network shares can be used to make a honeypot, either through emulation or by providing the actual service but it must be monitored and protected it. Whichever type of honeypot it is, one of its jobs is to collect malware so that it can be analysed, for information such as which domain it will try connect to.

Most honeypots spend most of there time waiting in the hope it will be contacted. Client honeypots are different though, they actively search for malware. They are used mostly in detecting web server based malware, such as blackholes. There is two systems for this type of honeypot, a controlling server and some high interaction clients that visit the websites they are instructed to visit. After every webpage visit, the client checks its file system is identical from when it started so that its integrity has not been compromised. If there are new files on the client, the server is instructed, given a copy of the malware and the activity is investigated. These type of honeypot could be used by organisations as another way to make sure their webserver arent used to host malware. It can also be used to test new URL's being queried within the organisation, to test if the websites are malicious or not.

Next we will look at 2 examples of countries that have brought development and real application to honeypot systems: The Netherlands and Japan.

5.4.3 Example, The Netherlands

Surfnet, The Netherlands research network that provides ICT infrastructure for the Dutch university system is heavily involved in honeypots. They have contributed to the Dionaea honeypot that emulates vulnerabilities to gain a copy of the malware that was targeting the system.

They also have developed a distributed honeypot system called Surfids⁷. It contains sensors that collect the data that is sent at it, and redirects over it over a VPN towards a central honeypot system. The central honeypot system can contain many different types of honeypots such as Dionaea to learn the maximum amount of information for the attacks. Sensors can be placed anywhere within an organisation all over the world to forward malicious traffic, and this can give the organisation lots of coverage without the complexities of supporting systems everywhere.

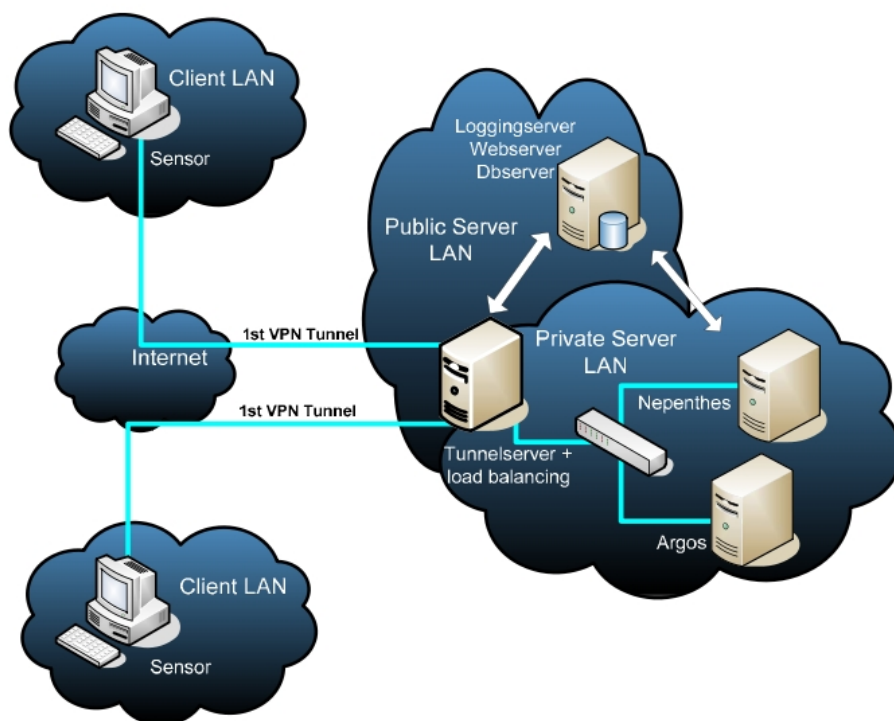


Figure 5.7 Surfids, a distributed honeypot system

⁷<http://ids.surfnet.nl>

Another Dutch organisation that uses honeypots is the National Cyber Security Center (NCSC). They use it to keep track of malicious activity targeting the country and issue warnings to the general public about their status. They also sponsor new honeypot related research such as a visualization tool to aggregate the data that systems such as the malicious traffic Surfids receives.

5.4.4 Example, Japan

Japan has long been a country of technological ingenuity, and honeypots are no different. One such invention that uses honeypots is the Cyber Clean Center initiative⁸, that warns Japanese computer users that they have been infected and helps them recover their computer to a healthy state.

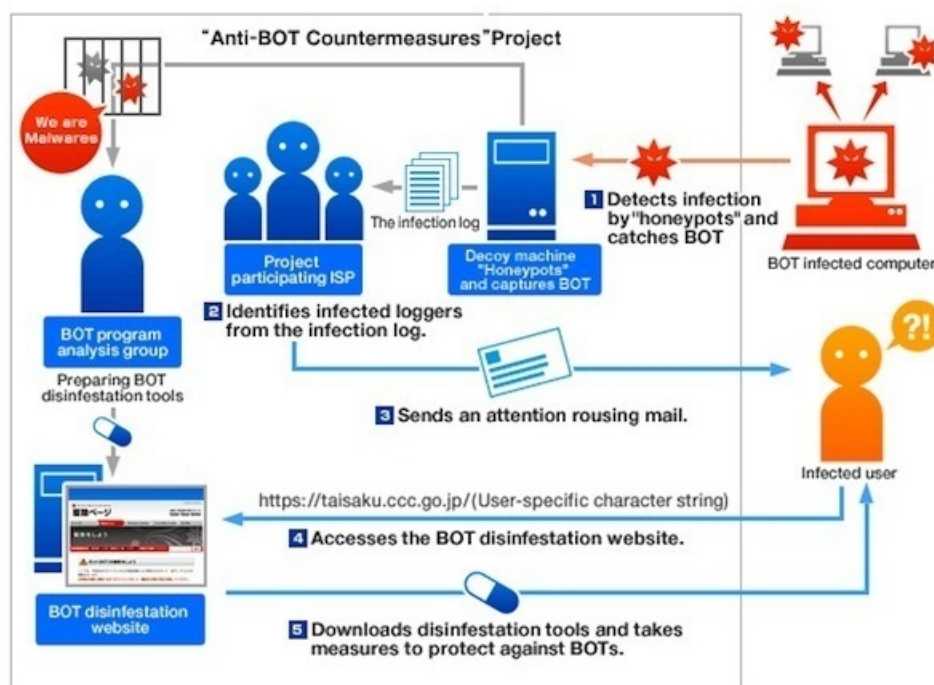


Figure 5.8 Japan's Cyber Clean Center initiative uses "honeypot" machines

They use the word BOT to describe malware that can lead to system to being compromised. They use honeypots to capture malware, most of which is Japan specific. With a copy of the

⁸https://www.ccc.go.jp/en_ccc/

malware, they can then execute it in a controlled manner such as a virtual machine and gather information on its communication mechanisms.

With detailed information on how the malware communicates, they can then actively search out other infected machines with the help of the Japanese ISP's. They then send users an alert mail, and tell them to visit a disinfection website. As they have a copy of the malware, the Cyber Clean Center can also make a reliable and specific disinfection tool for the public to use.

Japan's Cyber Clean Center is not the only organisation that is creating inventive solutions to cybercrime issues. The National Institute of Information and Communications Technology (NIST) in Japan has developed an interesting large scale network honeypot that looks for computers that are trying to connect to IP addresses that are not in use⁹. This idea comes from a honeypot called arpd. It can be used to reply to unused local IP address space. However, the Japanese have taken this idea to a massive scale. Connections to unused address space are typically a sign of malware trying to spread within an organisation.

5.4.5 Botnet infiltration

While analysis of malware can give certain information about the botnet, it cant tell anyone if the botnet is active. Nor can it tell what the botnets commands are if it were to be connected. Honeypot can also be used to infiltrate a botnet to gather information from within, such where the CnC servers are. Infiltrating a botnet is the best methodology to being able to get a whole map of its structure. P2P botnets are easier to infiltrate than botnets that use HTTP for communication because of the nature of P2P networks.

⁹<http://www.diginfo.tv/v/12-0116-r-en.php>

5.5 IDS/IPS

IDS/IPS stands for Intrusion Detection/Prevention System. The difference between the two is an IDS will detect a threat and report it, but an IPS will attempt to block the attack while it happens. An IPS is rare these days as letting any system dynamically block traffic can lead to connection problems. No system is perfect, so any false positives would block legitimate traffic, which is something many organisations take very seriously. This section will deal mostly with IDS's as they are regularly used in organisations.

An IDS's job is to detect attacks, and report them to be investigated. A good, well maintained IDS will have very few false positives and a high true positive rate. This authors main function within Fox-IT is working with IDS's.

The basic premise of detecting threats is once we have detailed information on a malicious activity we can tell the IDS to look for the activity. This can be done in a signature based or anomaly based way, which we looked at in the start of this chapter.

5.5.1 Snort

Snort is an open source IPS/IDS. It can run on both modes, the majority actual use of which is IDS mode. It has the ability to perform real time traffic analysis and packet logging on network traffic. Snort can perform protocol analysis and content searching/matching. It can be used to detect a variety of attacks and probes, such as buffer overflows, stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts¹⁰

It consists of three main parts: A packet sniffer to reliably retrieve the network traffic, a packet logger to write the traffic to disk and a signature engine used to match rules against this traffic to recognize attacks.

¹⁰<http://www.snort.org/snort>

Lets look at an easy example of how Snort works. The Aldi bot is botnet family that uses a very specific user agent tag (Aldi Bot FTW! :D) while it communicates over HTTP. Using this information, we can make a rule to look for this type of traffic. Once any traffic with this user agent is found, an alert is generated and logged for investigation.

5.5.2 Bro

Bro is a different type of IDS, it focuses on protocols such as SSH, HTTP and FTP. If someone tries to misuse one of these protocols Bro will detect this. A good example of misuse of a protocol would be using HTTP over a non standard port (not port 80 or 443). These would be a good way of getting information out from within an organisation because many networks do not appreciate that the bad guys don't play by the rules. Bro would detect that the HTTP protocol is being used somewhere else that other than the usual port and would trigger an alert.

Bro can also be used to detect multiple stages attacks. A multi stage attack usually consists of four parts: Reconnaissance, Intrusion, Privilege Escalation, and Goal. If there is a record of the reconnaissance, then when the intrusion occurs, Bro can correlate these events together to get a complete picture of the attack. It keeps extensive information on application-layer state of the network it monitors. Every connection to and from the network is stored in Bro, this database allows for very powerful network analysis.

5.5.3 Canary Detector

Another approach to detecting botnet traffic is the using a canary detector. The canary detector takes into account that some botnets traffic can be very quiet and stealthy and can therefore cannot be easily detected with IDS's such as Snort or Bro. This is where the canary detector is useful.

The idea is to track network traffic by the usage of destination atoms, the logical collections of destination addresses. These atoms are then measured for individual users and across the set of

users, and once they are scrutinised they can provide information on which destination atom stands out. In botnet communication, bots must check in periodically and by tracking these connections over time destinations, that are persistent can be flagged [11].

The amount of destination atoms that a normal end user has is small and stable, but when the machine the user uses is infected with malware, this behaviour changes. The periodic connection to the botnet controller stands out because the users connections are random in nature.

	Burst alarm	p-alarm	c-alarm
(long) DDoS attack	♦	♦	
DDoS attack	♦		♦
Scanning worm	♦		
IRC botnet	♦	♦	♦
Stealthy botnet		♦	

Figure 5.9 How a Canary Detect can find different types of botnets, c-alarms are local and p-alarms are globally detected throughout the group of sensors. [11]

5.6 Firewalls

A firewall is a system designed to prevent unauthorized access to or from a private network. It is a network gatekeeper that can allow or deny traffic to and from the network, by using a collection of information that is allowed. Firewall are not an absolute defence against malicious traffic but they do help contain threats.

Firewalls can be placed in lots of different devices. Host based firewall software can block connections on the machine is it installed on. Network devices such as routers can block connections with a firewall that don't meet a certain security criteria. Lastly, a dedicated firewall can protect the main point of access to the Internet.

5.6.1 Proxy

A proxy server is a type of firewall that intercepts all messages entering and leaving the network. It effectively hides the true network addresses of the computers on the network. A firewall is commonly used in organisations to keep control of the traffic that is on the network.

A typical proxy example is a HTTP proxy. All web traffic must go through this HTTP proxy to send and receive data. The proxy will inspect the network packets to try block malicious traffic based on a list of malicious IP addresses, domains, ports, protocols and other network attributes.

In organisations where there are many computer systems, the single point of access that a firewall will protect toward the Internet can be invaluable. When a certain malicious file is used to infect computers, the proxy can be set to simply block files with these names from entering or leaving the network. A good example of this, would be a file (field.jar) from a popular blackhole exploit kit. Field.jar would not usually be the name of a legitimate file so blocking these files wouldn't impact users, but might save them from being infected.

5.6.2 Filter known malicious traffic

One notable success of the firewall has been against worm propagation. Many of the network worms in the wild use a certain port to communicate and spread. By blocking these ports the worm can't spread from network to network. A good example of this is the Slammer worm that sends 376 bytes to UDP port 1434, once that port was blocked the slammer worm could not continue to propagate.

Firewalls use a variety of techniques to block malicious access. Blacklists and whitelists are one way of filtering out known malicious traffic. However, as we have already explained they have limited utility as they can become obsolete, but they can quickly prevent known malware from getting into the network.

At the router level, Spamhaus and Zeustracker have collaborated and are releasing a special list

of botnet CnC IP addresses that can be used by routers to block traffic using the Border Gateway Protocol. This list is automatically updated once a new CnC server is discovered.

5.7 Financial transactions monitoring

When malware defences fail and botnets control computers, the botnet try generate money for the botnet controller. Spam, DDOS and other ways of making money are used, but the most effective and popular method is to steal from a compromised computer users bank account. High profile botnets such as Zeus, Torpig/Mebrook and Spyeeye are specifically developed to commit banking fraud [2].

Lets look at how malware, in this case Mebrook gets banking details, to provide you with an insight into what financial transactions monitoring tries to stop, and the basics of how it works.

5.7.1 Torpig/Mebrook

Mebrook, as we saw in the section (2.2.2) on rootkits is a particularly dangerous botnet to become a part of. Its rootkits element make it very difficult to get rid of and detect, the exception being that when analysing the network traffic anyone can detect one of mebrooks flaws, its need to check for internet connectivity is a very specific way.

Once a computer is infected with mebrook, the payload it executes is a password stealing/bank fraud module called torpig. It will copy saved passwords and wait for the user to visit one of 300 banking websites. Once the user starts to conduct their internet banking, torpig will hook the browser process and control it into doing whatever it wishes [38].

Users will be redirected to a page were they must enter their password, PIN or other credentials. Once they have done this, torpig will use this information to send money to its controller via a money mule.

5.7.2 Virtual machines

Virtualisation is the technique of hiding the physical characteristics of computing resources from the way in which others systems, applications, and end users interact with those resources¹¹. A virtual machine is a machine that has been completely virtualised, and is merely a set of file to hosted an actual machine.

One way to prevent any systems on the network from infecting each other with a bot, is not have them physically be connected at all. By having an organisation use virtual machines, isolation can be obtained. One virtual machine can not communicate with another without explicit permission because there is no physical link between them.

An example of an attack that could be prevented by virtual machines is Flame's sophisticated Windows update attack that infected machines on the same subnet. If every host has its own private network segment this could not happen.

5.7.3 Host based tripwire systems

Antivirus's work on signature, Host based tripwire systems don't. They look for the methodological attack that can bypass antivirus scanners and encryption techniques¹². If a system is infected it will do something it usually doesn't do. By placing a tripwire, an infected system can be flagged as doing something outside the bounds of what it used to do.

Stopping APT's is a very tough task. One last defence against them is the tripwire, which is a piece of software installed on a host. A tripwire actively traces irregularities in the computers memory and hard drive. These irregularities are the first indication that something is wrong. If any file suddenly changes its permissions for example, the tripwire will detect it and flag it for investigation.

¹¹<http://www.cse.psu.edu/~tjaeger/cse497b-s07/slides/cse497b-lecture-26-virtualmachine.pdf>

¹²<https://www.fox-it.com/en/files/2012/08/Fox-Files-2-aug-2012-EN.pdf>

By having a safe baseline of movement the system can make, a tripwire can detect an APT's actions and stop them before any harm can be made. However directing a tripwire system is not an easy task. Making a good baseline can be difficult, as there are many different processes and files that are used in a system. Once something is detected outside the baseline it can also be difficult to determine what it is, and if it malicious or not.

Tripwire systems are certainly a good idea for protecting high valuable assets, such as databases and executives computers. They can give easy of mind to the IT security officer, that if everything goes wrong with every other security technology, at least the tripwire will catch the malware.

5.7.4 Monitoring transactions

The thing about torpig, zeus and all the other malware that tries to steal our money, is that they are automating and manipulating a process that is human. This is very difficult, some might say impossible to get absolutely correct. The modules that malware uses to steal money are automated scripts that go from state to state, action to action before finishing. The fact that automating a user's action perfectly is so difficult makes it possible to detect.

One such system is Fox-IT's DetACT¹³. It is a way for banks to detect when a customers computer defences havent worked and are trying to transfer money via an automated script. An easy example of how an automated script will not behave like a user is filling in online forms. A user might take a minute or two to fill out and the script will take just 1 second. An automated script wants to conduct its fraud in the fastest way possible before the user might notice what is going on. These anomalies are what financial transactions monitoring are mostly about.

In the above image, financial transactions are analysed for fraudulent transactions. This data is checked against known fraud data, intelligence reports and an anomaly detection engine. If anything suspicious detected it is reported and analysed further.

¹³<https://www.fox-it.com/en/products/detact-for-online-banking/>

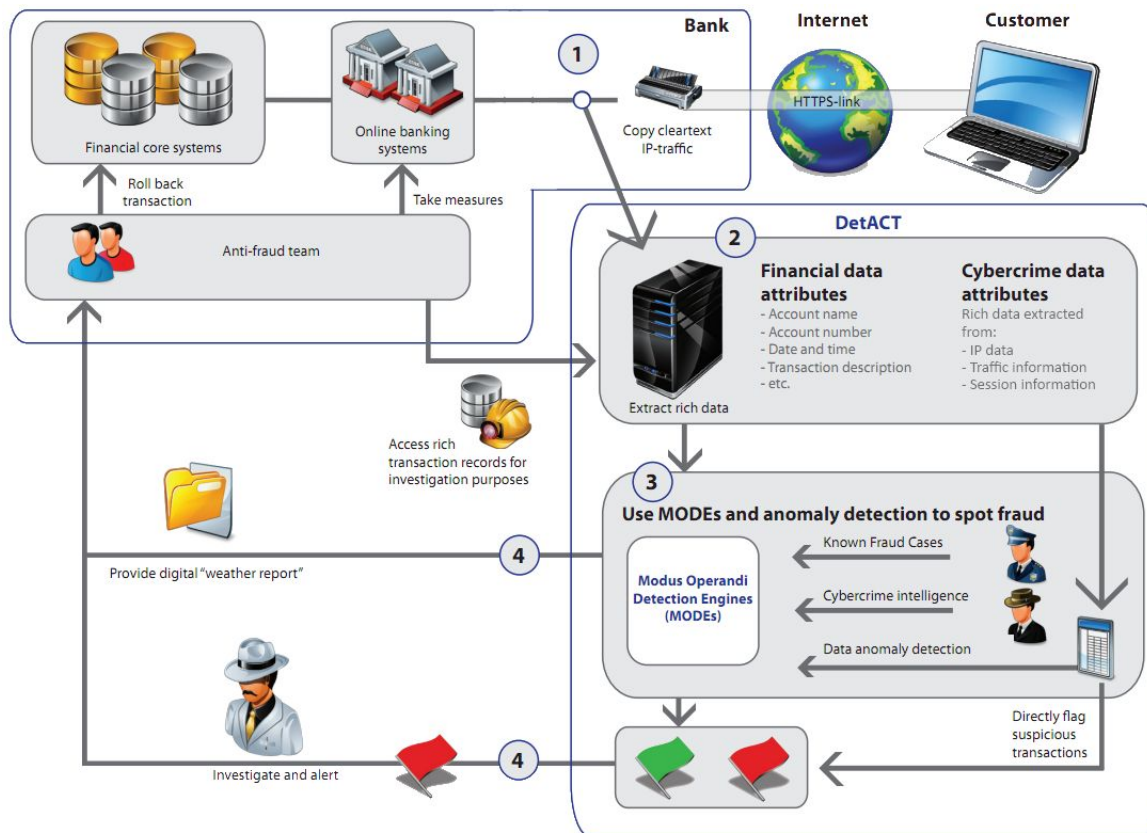


Figure 5.10 How Fox-IT's DetACT works

5.8 Security policy

Security policies define "security" for a system or site. They may be implied policies defined by the common consensus of the community, or they may be informal policies whose interpretations are defined by the community [9]. With a good security policy, the threat of malware can be migrated within an organisation.

A good security policy is one that is easily understood and easy to follow. It should not be taxing on the users but should allow them the flexibility to adjust if needed. Here are a couple of rules that would be in a typical security policy for an organisation:

- Give users the least amount of access they need to do their job. Also known as the 'Principle

of least privilege’.

- Always sign out of a computer while away.
- Give each document you write a security level such as ‘Confidential’
- Keep all software up to date, we will elaborate on this one later.
- All data (hard drives, usb sticks, etc) leaving the organisation must be encrypted
- The list goes on....

5.8.1 Keeping programs up to date

As we have seen previously, keeping software up to date on a system is not easy. Most security policies will state that this must happen however, because an old version of software makes the system more vulnerable.

One way an organisation can migrate this risk is to centralise the ability to determine what software is installed on a certain computer. All systems would have a script installed that indexes the programs installed on the systems and reports it. This way, if a users has not undated the software (Java, for example) on the system the organisation can know and remind the user to update it.

5.8.2 Auditing an organisation

An audit is the process of maintaining the security policy is adhered to. An auditor will check if the computers on the network all adhere to the policy but testing the systems for vulnerabilities. His job involves thinking like an attacker to try gain access to systems that should be restricted.

This process of attacking oneself, can lead to the discovering of many vulnerabilities in the organisation.

5.9 Botnet annihilation

When botnets get too big, they attract attention. When they attract attention, law enforcement and technology companies find it in their interest to tackle the specific problem. There are many examples of botnet annihilation in recent months, and the one thing they have in common is the high profile nature and enormous size of these botnets that were taken down.

P2P botnets are easier to infiltrate but harder to take down because P2P networks are built with redundancy in mind, HTTP botnets are harder to infiltrate because each bot only knows its CnC server not the other bots on the network. With these facts in mind, Law enforcement and technology companies infiltrate by adding honeypots to the botnet so they can gather intelligence on how they work. They also register domains that bots would use and sinkhole them so that they can gather reliable information on a botnets size.

Infiltration is the first step towards annihilating a botnet, however botnet controllers do not make either easy. Botnet controllers regularly test their bots to see if they are really under their control, such as issuing a command to send a malicious payload to a server they control and check if it is received. They do this because technology firms are especially prone to conducting activities that are considered illegal and therefore they cannot take part in such actions, so they block or limit outbound connections.

Once a botnets communication structure is under attack from organisations trying to take it down, Botnet controllers usually try sending a command to the bots to change its CnC servers. This can result in a smaller botnet still under malicious control. Its very difficult to take absolute control of a botnet because of the way botnets are structured.

5.9.1 Law

Law enforcement such as the FBI, and the new European Cybercrime Centre (EC3) are all involved in tackling botnets from a criminal perspective. Organised crime groups, terrorist groups and other criminals have been quick to exploit the ever growing use of technology. Law enforcement is simply reacting to this change.

While the FBI infiltrated and annihilated the DNSChanger botnet, they didnt stop their. They tracked the owners of the botnet to Estonia so they could be arrested by Estonian authorities. DNSChanger was a particularity difficult and sensitive botnet to take down because the criminals malware had changed the infected computers DNS server addresses to rogue DNS servers that they controlled. The FBI took control of these DNS servers and not take them down because users would then potentially be without a DNS server, which would mean no internet connectivity.

Another botnet annihilation success story is the Dutch high tech crime unit's take down of the Bredolab botnet. They seized control of 143 command and control servers rented within the Netherlands. This is a great example of how jurisdiction matters because the botnet controller had made the mistake of placing his CnC servers in a country that is clearly serious about its cybercrime prevention.

5.9.2 Technology companies

There are many examples hows technology companies have helped in the fight against cybercrime. We will look at 2 different companies attempts to take down botnets:

Kaspersky, the anti virus vendor last year attempted to take down the Kelihos botnet. Kelihos botnet was a P2P botnet that used a custom protocol to communicate. Kaspersky used the fact that the criminals had made their own protocol which would always have some vulnerabilities. They reverse-engineered the malware and wrote their own code to infiltrate and sinkhole the botnet away from the criminals. The sinkholed botnet was then abandoned by the criminals because they had

lost control of it.

Microsoft, the vendor from which the Windows operating system comes, don't take kindly to criminals attacking its product or its customers. They recently tried to take down a Zeus botnet that had its CnC servers in the US. This gave the American company jurisdiction to seek a court order to disable the botnet.

What all the above examples have in common, is that it wasn't just one company or law enforcement agency that stopped the botnet. It was the cooperation, intelligence sharing, communication and trust between international organisations that helped stop the spread of these botnets. The IT security community is usually the first to tell the rest of the world about a new botnet, a good example being Stuxnet was found by a Ukrainian IT security researcher.

Chapter 6

Organisations and Botnets

"The achievements of an organization are the results of the combined effort of each individual."

- Vince Lombardi

Every organisation is different, some are small and local, some are huge and international. All have to deal with the threat of botnets within their organisations. This chapter we will look at 3 types of organisation and show what they can do, to deal with botnets.

6.1 Small organisation/Home

Small organisations and home users are the most vulnerable because they have limited knowledge and resources to deal with botnets. Keeping them safe is the hardest task at hand. However thankfully the sheer volume of small businesses and home users means that bigger businesses have made products to suit them, that wouldn't cost the much money. Small organisations don't see the need to spend money on malware protection because they think that it wouldn't affect them.

Best practices

Implementing best practices is a way a small organisations can deal with malware. Here are some best practices that can make a difference in a small organisation or home environment.

Making sure all the software on the systems is patched up to date is a critical best practice these days. Vulnerabilities can be easily exploited to gain control of the organisations systems. Considerate the software that is used by the Internet or files that come from it:

1. Operating systems such as Windows
2. Internet browsers such as Internet explorer
3. Plugins such as Java
4. Document editors such as PDF readers or Microsoft word

If these steps is taken, an organisation will be 'ahead of the crowd' in terms of limiting its vulnerability to malware.

The second best practice that must be followed is education, it can really make a difference. With some simple education on the current type of threats that can effect the organisation, the users can prevent malware gaining a foothold. With awareness, the users can know what to look out for and can prevent malware infecting the computer.

Lastly, Anti-virus is essential for protecting the computer when the user does something they shouldn't have. Its obviously not an absolute solution but it can prevent the most trivial of malware infecting the computer.

Focus on importance

Not everybody in an organisation is important, some people or computers are much more important than others. The most important entity should get the priority of defence. Having best practices

(above) for everyone is good, but focusing on where the worst place an attack could happen is much smarter.

Evaluating such importance is the first step. With a list of important entities, more focus on security can be applied to them, such as having a stricter password policies or focusing on what software needs to be kept updated.

Every organisation, big or small will have this list of important entities. Also, an important person in a small organisation is no different than a big organisation, they are still important and need to be treated as such.

6.2 Medium size organisation

Medium size organisations don't have lots of money to use to solve problems such as botnets, so they must commit to a balance between cost and efficiency. There are many good proven technologies that can be used within the organisation to mitigate the risk of a botnet infection and the possible loss of data, reputation and money. It seems obvious but the medium size organisations should have already implemented what the above small organisation are doing, such as anti-virus and education just to start with.

Firstly, however there must be a person (or a team of people) in charge of the IT security within the organisation. If the organisation deals with high risk entities such as sensitive data, there should be extra resources available to deal with these risks.

Lets take a closer look at what best suits a medium size business in dealing with botnets.

Firewall

Every organisation should have a firewall, or multiple. Employees computers should have local software firewall installed as a last line of defence against intrusions into their computers.

Hardware firewalls that are placed between the organisations internal network and the internet are one of the most important network defences an organisation can have. Without them any traffic from anyone on the internet could pass freely to systems within the organisations network; the possibilities of malicious (automated) traffic getting onto the network would be a enormous. Therefore the risk would be far to much for an organisation to accept.

A firewall also works the other way round, in keeping traffic that shouldn't be allowed out to the internet. While nobody should be able to connect to the databases from the internet, for example, nobody should also be able to connect from the databases to the internet. Firewall secure both directions of traffic.

Proxy

There are two main ways for malware to enter an organisation, web traffic and emails. There are of course, other avenues of attack but these are the main ones. Protecting what web traffic comes and goes from the organisation is an important job, and it is the job of the (web) proxy.

The proxy can filter out bad traffic that is undesirable, stop malicious jar files from getting past it, for example. Something that a firewall cannot do, because it must let web traffic though because all web traffic looks the same to it. If the proxy detects something it doesn't like, it simply resets the malicious connection. Once this is done, there is no hope of the malicious traffic succeeding in its goals.

IDS

An IDS is essential in dealing with botnets, within an organisation. Some manner of network monitoring must be in place to find weaknesses and highlight problems within the organisation. A properly placed IDS should be located within the organisations firewalls, not outside them. Placing the IDS outside the firewall, within the DMZ, for example will reduce the efficiency of the system

because it can't 'see' all the traffic, but most importantly nor can it determine which computer inside the firewall or proxy the malicious connection came from.

The logs of an IDS are useless without someone skilled to look at them. They should be reviewed in an organised manner, and escalated if an incident is deemed important or critical. With some tuning to suit the network, an IDS can be a great tool for discovering malware within the network. An early indicator to the malware problem or issue is better than no indicator at all whatsoever.

Encryption

It almost doesn't matter which type of organisation it is, encryption is important. Every organisation has data it wishes to keep confidential. Encryption allows them to do just that. A good place to start is the employees computers, they should all have encrypted hard drives. Data on there is very valuable and without encryption; if someone takes the computer they can simply read the contents of the hard drive or even worse reset the password and gain total access to the users computer.

The need for encryption does not stop at employees computers however, databases and servers should also be encrypted. Physical access to a server, in effect means total access to that servers data without encryption.

Another important consideration should be the client data that is held by the organisation, these should have the most defences attached to it because loss of this data would probably mean litigation and loss of reputation.

Recently Blizzard, a massive online gaming company was compromised.¹ The data that the attacker took was very valuable. It included password reset information, email addresses and other security related information. The breach did not however haemorrhage its users passwords. Blizzard was lucky enough that they sensibly stored these passwords, by using Secure Remote

¹<http://www.bbc.com/news/technology-19207276>

Password(SRP) protocol.

SRP is a secure password-based authentication and key-exchange protocol. It is effectively a way of storing passwords for authentication without storing them in plaintext. Passwords are converted into an 'verifier' for each user and only this verifier is stored. Once the user sends his password to be authenticated the server will check if this password is the same as the verifier. It is a zero-knowledge password proof. Any attacker that steals the verifier cannot determine the password without a brute force search.

Using technology like SRP, cannot help stop breaches in security but it can help prevent contain them if and when they occur.

Virtual networks

Virtually splitting up an organisations network, make sense. It adds flexibility and security. Virtual networks can be deployed and changed within moments without having new cables or switches. Traffic also be dynamically partitioned, for example the web server VLAN (Virtual LAN) can be seperated from the mail server VLAN even though they are on the same physical network.

Virtual private networks are also an important addition to an organisations security, they allow people to securely connect to organisation internal data without being inside the organisations network. VPN's also allow multiple offices to be (virtually and) securely connected together.

Mail server

Emails are one of the biggest vectors of attack for malware. The most obvious example of this is an untrusted email attachment, see below. A reasonable defence against these emails is for making all emails go though the organisations mail servers. Port 25 should be blocked at the external firewall except for the mail server. Any traffic on port 25, going directly towards the Internet is most probably spam and shouldn't be allowed through. By forcing all emails coming into and

out of the organisation through the mail server, this traffic can be then be checked for malicious content.

```
Subject: United Parcel Service notification #28815969
MIME-Version: 1.0
Content-Type: multipart/mixed;
  boundary="-----9C53D2B06885B84D"
-----9C53D2B06885B84D
Content-Type: text/plain; charset=iso-8859-1
Content-Transfer-Encoding: 7bit

Dear customer.

The parcel was sent your home address.
And it will arrive within 6 business day.

More information and the tracking number are attached in document below.

Thank you.
. 1994-2011 United Parcel Service of America, Inc.

-----9C53D2B06885B84D
Content-Type: application/zip; name="UPS_document.zip"
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="UPS_document.zip"

UESDBBQAAAAIAFmkBkF712w0mXIAAACAAAQAAAUVBTIGRVY3VtZW50LmV4Ze39B9gVRbY/
Cq/OOcfdERVZGJQgZhrUUJSMOZDKBRGQJJhgpkXtKVHG7OgoiHHGNKODYgQDGNqc4yCmNuCC
61bv16Azc/5nzm+73nu/Z67od/dxV21a1XVqrV+q9Le7+AFwAAASy6MAW6Hzk8f+O8/D5FL
z+/Q4VbpsQ1upwY+tshwiqnTu0vdNmX8tFHhdBkzavLkKTO6jB7XZdrMYV0mTO7Sb9CwLsdM
```

Figure 6.1 A trivial example of an email that would be stopped by the mail server. Note the attachment called `UPS_document.zip` that is clearly malicious because it makes no reference to an actual delivery or user.

There are many ways a mail server can help prevent malicious content getting into the organisation. Firstly, the mail server should have a spam blacklist to prevent most spam getting through. The mail server should also scan for executables to prevent obvious spread of malware. Lastly, for the most ambitious organisation, they could take all the URL's seen in emails and run them through a client honeypot to make sure the URL's are not blackholes.

Patch management

Within an organisation, patch management is a serious problem. Keeping every piece of software on every system up to date is a monumental task. There is also the added complication of consistency, patching a critical piece of software can have unintended consequences. Only recently, a major bank in the UK, the Royal Bank of Scotland had to pay customers for the outage that was caused by the upgrading of its critical systems².

Depending on the size of the organisation, keeping track of system grows exponentially. Some centralised control of employees computers maybe the answer to keeping track of the installed programs on the computers. In other cases, using a network analysis tool may yield reliable results. There are many things about a host on a network that can be gauged by analysing the traffic on a network.

One simple example is to keep track of old Java versions. Its easy to record the user agent's that are seen on the network and determine what version of Java is it. Below is an example of such detection. Any Java version that is not the latest version is logged and can be processed into a list or even an alert to upgrade the vulnerable software.

```
alert tcp any any -> any any (msg:"Vulnerable Java version";  
flow:to_server,established; content:"User-Agent|3A|";  
content:"Java/"; content:!/1.6.0_33"; content:!/1.7.0_05";  
classtype:web-application-activity;)
```

Security policy

All the above technology needs to be specified somewhere. That document is the security policy. Every organisation has one, in some shape or form. The document will contain all the controls that

²<http://www.bbc.co.uk/news/business-18575932>

are in place in the business.

This security policy will be written by the chief security officer and must be approved by higher management such as the board of directors. A security policy will include everything about the organisations IT security efforts³.

There are three types of types of rules that go into a security policy:

- A *policy* is typically a document that outlines specific requirements or rules that must be met. In the information/network security realm, policies are usually point-specific, covering a single area. For example, an "Acceptable Use" policy would cover the rules and regulations for appropriate use of the computing facilities.
- A *standard* is typically collections of system-specific or procedural-specific requirements that must be met by everyone. For example, you might have a standard that describes how to harden a Linux server for placement on an external (DMZ) network. People must follow this standard exactly if they wish to install a Linux server on an external network segment.
- A *guideline* is typically a collection of system specific or procedural specific "suggestions" for best practice. They are not requirements to be met, but are strongly recommended. Effective security policies make frequent references to standards and guidelines that exist within an organization.

The SANS Institute, which is a large information security resource, recommends a IT security policy be broken down to the following sections:

- *Audit Security Policy*. This policy defines what must be done when conducting audits and risk assessments to ensure integrity with the organisation.
- *Computer Security Policy*. This policy defines software, encryption and passwords that must be used as well as how to recover from a disaster.

³<http://www.sans.org/security-resources/policies/>

- *Desktop Security Policy.* A simple policy about having a clean desk and how to catch social engineering attempts.
- *Email Security Policy.* Every thing to do with email is set out here, such as how the organisation has the right to read the emails an employee sends.
- *HIPAA Security Policy.* If the organisation is involved in the healthcare information it must adhere to extra security constants such as provide an audit trail for patient's stored information.
- *Internet Security Policy.* Anti-virus, remote access and digital signatures are defined in this policy.
- *Mobile Security Policy.* The organisations mobile phones used by employees have their own policy.
- *Network Security Policy.* Everything network related is included in this policy, such as the minimum security setting on all the routers.
- *Physical Security Policy* All procedures to do with visitors and contractors is defined here.
- *Server Security Policy.* This policy details server security procedures and the minimum levels that should be implemented on all server within the organisations.
- *Wireless Security Policy.* All wireless communication security is defined here, such as the type of protocol that must be used e.g. no WAP protocol shall be used.

An important detail related to botnets in organisations is incident response and reporting. There is a policy for this of course; all botnet infections need to be reported and analysed to prevent them happening again. Such policy might include, making sure the organisations anti-virus signatures are updated to include all the previously infected malware from within the organisation.

6.3 Large organisation

Everything that is in a small and medium organisation will be implemented in a large organisation, the only difference will be that it is on a much larger scale. Large organisations still have anti-virus, firewalls and security policies, the difference is that they must keep track of the vast amount of information that all these parts of the organisation generate.

Centralised monitoring

A large organisation will have an Information Security Team or maybe even a whole department dedicated to IT security. They must direct security policy and deal with incidents from thousands of employees. This group will typically have its own center called a Security Operations Center(SOC) for monitoring the incidents that are generated within the organisation. All security related information can be correlated at one time from for a centralised place for logging, reporting and analysing security incidents.

This size of organisation will also need to monitor its network segments in one or more places to make sure everything in the networks is working precisely as it should. A network operation center (NOC) is similar to the SOC, but its work deals with dealing with the network infrastructure itself. This centralised control point allows for easy resolution of issues because the expertise to resolve the issue is in one place.

Targeted attacks

Large organisations hold lots of money and responsibility, this makes them a prime target for attack. Cybercriminals will specifically target these organisations because they are rich and there is a good possibility of success. In a large organisations, there will always be a bigger attack surface (for example, more computers systems; more chance of finding a vulnerability) to attack.

However large organisations have more resources to deal with the botnet threat. They could use the network's central structure to place honeypots/sensors around their network in an attempt to catch any malware that tries to spread within the network. The use of network address space that is not used is a common tactic to using honeypots in this nature. Nobody can tell if an address space is used or not, when malware tries to connect to a network segment not in use the source can be flagged as suspicious.

Another technology a large organisation could use to prevent targeted attacks is tripwires. An expensive but very powerful tool as explained in Chapter 5, tripwires can detect anomalies on hosts they are installed on and configured for. By focusing and therefore protecting what is most important to an organisation with a tripwire allows them extra security where it really matters.

Loss of data/damage to reputation

Botnets within a large organisation can be used to exfiltrate valuable data that can damage the organisation's reputation. While this can happen to any sized organisation, large ones have more data to protect and therefore have a tougher job at doing so.

An organisation's best defence against the loss of data is to implement all the detailed procedures above. Using encrypted databases and segmenting networks into virtual networks will tighten security.

Furthermore, audits conducted will report on areas that are vulnerable such as old web servers. These reports need to be acknowledged and acted upon.

Chapter 7

Conclusion

"In war you will generally find that the enemy has at any time three courses of action open to him. Of those three, he will invariably choose the fourth."

- Helmuth Von Moltke

In this last chapter we will answer the research question: What are the best methods for detecting and analysing modern botnets in organisations? We have looked at many different methods to detecting botnets in organisations. We have also looked at how these botnet can be analysed, from the malware they used to infect systems to the network traffic they use to communicate. We have also looked at the fact that malware makers is always ahead of the race, and IT security is always catching up.

In summary, botnets cannot be stopped completely, they can only be contained and dealt with as they arise. Cybercriminals always come up with new techniques to conduct their criminal activities. We have looked at many examples of botnets (see below) and how each type of botnet can be detected within an organisation with such technologies such as snort and tripwire.

Examples of botnets:

- Ghostnet, page 8
- Mebroot, page 9
- Zeus, page 14
- Spyeye, page 15
- ZeroAccess, page 16
- Stuxnet, Duqu, Flame and Guass, page 19-22
- DNSChanger. page 29
- Storm, page 51
- Feederbot, page 55
- Aldibot, page 85

7.1 Answering the research question

My research question asked what are the best methods are detecting botnets, the answer is simple: All of them. Defence in depth is not a new idea, its a long used military strategy adopted by the IT security industry. It seeks to delay rather than prevent the advance of an cyber attacker, buying time to investigate the issues and come up with solution or detection method.

Certain organisations should presume different defence strategies. Small organisations should focus on getting a good baseline security and focusing on the critical important parts of the organisation. While larger organisations have so many computer systems to protect, they must consider-

ate on their critical systems with more complex security measures such as a tripwire or honeypot system.

What challenges are currently being faced in defending against botnets?

In chapter 3, we looked at the challenges; at the constant and endless stream of vulnerabilities that plague our computer systems. Cybercriminals are using using these vulnerabilities to compromise organisations around the world, in an increasingly sophisticated manner. This sophistication makes defence against botnets more and more difficult everyday.

Law enforcement is active in fighting cybercrime, but their utility is limited. Cybercrime doesn't respect national borders nor does it care about accountability of it actions on others. Cross border cooperation is rare because of the lack of will to address the global issue of botnets. The lack of trust on the internet is also partially the reason for the growth of botnets and cybercrime because cybercriminals have free rein to attack computer systems from the comfort of their homes. It is very difficult to trust anything coming from the Internet these days.

Users are also to blame for the lack of security on computer systems, in fact they are the weakest link in the whole chain of security that is built to protect them. Lastly, organisations must strike a balance between the risk of botnets infecting their computer systems and cost that is involved dealing with this problem. In other words, they can build a wall against the threat but a wall can be gotten around somehow, never mind how tall it is.

How can DNS servers and their data/records be used to detect botnets in organisations?

The question is an interesting one. DNS servers and their information are an untapped resource against botnets in many organisations. This is especially the case of ISP's and large organisations

because they have a tremendous amount of data that can be used to track botnets.

In Chapter 5, we look at a proactive approach called DNS dynamic reputation systems. Both are systems to gauge if a new domain seen on the network was malicious or not. One of the systems was for the local reserve DNS servers traffic typically within an organisation called Notos. The other was a top level domain focused solution called Kopsis.

DNS dynamic reputation systems are a recent technology that have proven to detect botnets even before they can propagate and become a large sized botnet. It introduces an effective method of detecting botnets both large and small, as they both adhere to the same principle: IP addresses are more expensive to replace compared to domains.

Another way DNS can be used to protect an organisation is blacklists, these can be IP address or domain name focused. By detecting or blocking these malicious communications, the botnets within the organisations can be found.

Lastly, DNS sinkholes can be used to stop known malicious DNS requests within a network. The idea is to give a special response to a DNS request of a malicious domain. The response will be send back a different DNS response (usually loopback) so that the malicious communication can never happen.

How can honeypots be used to detect botnets in organisations?

Honeypots are more prevalent than one might think, the reason not many people hear about them is because they are never advertised because that would negate the idea and the purpose of the honeypot. A honeypots job is to keep low key and because it only want to detect traffic that is malicious and not legitimate.

A distributed honeypot system called surfids is a great way for an organisation to detect botnet propagation attempts across its network. By setting up sensors across the network that forward traffic to a central honeypot system, the detection of malware on the network can be done on a

large scale easily.

The SSH honeypot, a popular one among IT security professionals; it can simulate a real SSH server and can provide an actual emulated shell for an attacker to interact with. The attacker will more than likely be using a botnet to implement its brute forcing of the password on the systems it tries. By setting up a honeypot like this, an attackers time can be wasted and focused away from actual legitimate systems, as well as the connections and actions of the attacker can be logged for analysis.

What is the best strategy for detecting botnet activity? Are passive or active approaches better?

The best strategy depends on the size of the organisation and the expertise within it. If the organisation is small, its best strategy should be to invest in training staff and having a good anti-virus. Bigger organisations are more likely to focus on IT security themselves, however they may out-source the sophisticated and time consuming tasks to a 3rd party such as maintaining an IDS on their network.

A passive approach consists of data that is gathered solely through observation¹. These approaches include IDS's, honeypots and DNS traffic analysis. Taking the passive approach towards detecting botnet activity brings with it the advantage that no traffic is manipulated. There is no risk of any traffic being intercepted and then manipulated, no risk that a core business process will be blocked or shutdown. However, the passive approach can only work when it has traffic to observe or is directed at it; this is its disadvantage.

To actively dictate what comes in and out of the network must be done with care and intelligence. Business processes could be effected by the decision to block traffic with an IPS, or a DNS

¹http://www.enisa.europa.eu/activities/Resilience-and-CIIP/networks-and-services-resilience/botnets/botnets-measurement-detection-disinfection-and-defence/at_download/fullReport

sinkhole. The consequences are therefore unknown, so care must be taken. The information that can be gauged from these technologies are much greater because the traffic can be directed at them. Also, when infiltrating a botnet the value is bought with the end of the whole malicious network.

Should organisations focus on botnets or other threats, how big is this problem for organisations compared to other security problems?

We have already seen in Chapter 2 (Timeline) how malware is constantly evolving, botnets are merely the next step in this evolution. However, we can clearly see that botnets are the center of all malicious activities on the Internet today. They are a very powerful asset for a cybercriminal to have.

In Chapter 4(Botnets in depth), we showed that botnets are complex networks, that are very difficult to detect and annihilate. Focus should be set towards this threat because it has such a broad spectrum of issues that will effect any organisation.

7.2 Discussion

Throughout this thesis, we have argued that botnets are a serious threat to organisations. They are constantly a menace to users whom don't understand them, to IT staff whom must protect against them and to the whole IT security industry that are always playing catch up.

Cybercriminals have an easy job, there will always be vulnerable software on the internet for them to profit from. This is a fact. There will always be social engineering to trick people. The best anyone can do is make themselves as safe as they can, by understanding botnets and thinking about the problem.

Recommendations / Future work

Most systems such as anti-viruses and firewalls are well developed systems that cannot be improved. On the other hand, new technologies such as honeypots, DNS sinkholes and dynamic reputation systems have a chance to greatly improve security in the future. They need to be constantly improved upon however. There is little incentive for organisations to invest in these new technologies because they are unproven in their eyes. This is why they must be improved.

The recommendation of this thesis is to keep the status quo of firewalls and IDS's but look into the vast amount of new detection technology that is available these days. Some of these many even stop an organisation from catastrophe.

Appendix A

Zeus logging samples

A.1 Banking details

[Censored]

A.2 Facebook details

[Censored]

Appendix B

DNS

B.1 DNS Tunnel example 1

[Censored]

B.2 DNS Tunnel example 2

[Censored]

Bibliography

- [1] Abuse.ch. Spyeye tracker. <https://spyeyetracker.abuse.ch/index.php>, Accessed December 12th, 2011.
- [2] abuse.ch. The swiss security blog. <http://www.abuse.ch>, Accessed November 23rd, 2011.
- [3] Korea Information Security Agency. Botnet c&c handling with dns sinkhole. www.cert.org/archive/pdf/BotSinkhole_KrCERTCC.pdf, Accessed November 23rd, 2011.
- [4] Manos Antonakakis, Roberto Perdisci, David Dagon, Wenke Lee, and Nick Feamster. Building a dynamic reputation system for dns. In *Proceedings of the 19th USENIX conference on Security*, USENIX Security'10, pages 18–18, Berkeley, CA, USA, 2010. USENIX Association.
- [5] Manos Antonakakis, Roberto Perdisci, Wenke Lee, Nikolaos Vasiloglou II, and David Dagon. Detecting malware domains at the upper dns hierarchy. In *USENIX Security Symposium*, 2011.
- [6] I. Arce. The weakest link revisited [information security]. *Security Privacy, IEEE*, 1(2):72 – 76, mar-apr 2003.

- [7] Information Systems Security Association. Information loss in enterprise networks: Mini-botnet. <http://www.issa.org/Library/Journals/2011/February/Al-Bataineh&%20White-Mini-Botnets%20Role.pdf>, Accessed November 23rd, 2011.
- [8] Michael Bailey, Evan Cooke, Farnam Jahanian, Yunjing Xu, and Manish Karir. A survey of botnet technology and defenses. In *Proceedings of the 2009 Cybersecurity Applications & Technology Conference for Homeland Security*, pages 299–304, Washington, DC, USA, 2009. IEEE Computer Society http://www.eecs.umich.edu/~mibailey/publications/catch09_botnets_final.pdf.
- [9] Matt Bishop. *Computer Security: Art and Science*. Addison-Wesley, December 2002.
- [10] F Brazier. Botclouds the future of cloud-based botnets? <http://homepage.tudelft.nl/68x7e/Papers/botclouds.pdf>, Accessed November 23rd, 2011.
- [11] J Chandrashekar. The dark cloud: Understanding and defending against botnets and stealthy malware. *Intel technology Journal* <http://download.intel.com/technology/itj/2009/v13i2/pdfs/ITJ9.2.9-Cloud.pdf>, 13:130–147, Accessed November 23rd, 2011.
- [12] Cisco. Botnets: The new threat landscape. http://www.cisco.com/en/US/solutions/collateral/ns340/ns394/ns171/ns441/networking_solutions_whitepaper0900aecd8072a537.pdf, Accessed January 20th, 2012.
- [13] Team Cymru. Cybercrime: An epidemic. *Queue*, 4(9):24–35, November 2006.
- [14] Christian J. Dietrich, Christian Rossow, Felix C. Freiling, Herbert Bos, Maarten van Steen, and Norbert Pohlmann. On Botnets that use DNS for Command and Control. In *Proceedings of European Conference on Computer Network Defense - EC2ND*, 2011.
- [15] ERNW. Exploring novel ways in building botnets. http://www.ernw.de/content/e7/e181/e1623/ERNW_Novel_ways_to_build_botnets_ger.pdf, Accessed November 23rd, 2011.

- [16] F-Secure. Windows remote desktop worm "morto" spreading. <http://www.f-secure.com/weblog/archives/00002227.html>, Accessed December 12th, 2011.
- [17] Stefan Gorling. The Myth of User Education. *Proceedings of the 16th Virus Bulletin International Conference, 2006* <http://www.gorling.com/files/texts/StefanGorlingVB2006.pdf>.
- [18] Thorsten Holz. Know your enemy: Tracking botnets. <http://www.honeynet.org/book/export/html/50>, Accessed November 23rd, 2011.
- [19] SANS Institute. Intrusion detection and response - leveraging next generation firewall technology. http://www.sans.org/reading_room/whitepapers/firewalls/intrusion-detection-response-leveraging-generation-firewall-technology_33053, Accessed January 21th, 2012.
- [20] SANS Institute. Dns sinkhole. http://www.sans.org/reading_room/whitepapers/dns/dns-sinkhole_33523, Accessed November 23rd, 2011.
- [21] Kaspersky. Zeus banking trojan report. <http://www.secureworks.com/research/threats/zeus/>, Accessed December 12th, 2011.
- [22] Kaspersky. The top-10 of 2011: An explosive year in security. http://www.kaspersky.com/images/Article_The%20top%2010%20security%20stories%20of%202011_ENG-10-136658.pdf, Accessed January 13th, 2012.
- [23] Trend Labs. Phishing pages pose as secure login pages. <http://blog.trendmicro.com/phishing-pages-pose-as-secure-login-pages/>, Accessed December 12th, 2011.
- [24] Collin Mulliner and Jean-Pierre Seifert. Rise of the iBots: Owning a telco network. In *Proceedings of the 5th IEEE International Conference on Malicious and Unwanted Software (Malware)* http://www.mulliner.org/collin/academic/publications/ibots_malware10_mulliner_seifert.pdf, Nancy, France, October 2010.

- [25] Antonio Nappa, Aristide Fattori, Marco Balduzzi, Matteo Dell'Amico, and Lorenzo Cavallaro. Take a deep breath: a stealthy, resilient and cost-effective botnet using skype. In *Proceedings of the 7th international conference on Detection of intrusions and malware, and vulnerability assessment* <http://iseclab.org/papers/skypebot.pdf>, DIMVA'10, pages 81–100, Berlin, Heidelberg, 2010. Springer-Verlag.
- [26] Netwitness. Netwitness discovers massive zeus compromise. <http://www.netwitness.com/about/press-releases/2010-netwitness-discovers-massive-zeus-compromise>, Accessed January 5th, 2012.
- [27] European Network and information Security Agency. Operation black tulip. <http://www.enisa.europa.eu/media/news-items/operation-black-tulip/view>, Accessed May 6th, 2012.
- [28] Arbor networks. Yoyoddos: A new family of ddos bots. <http://ddos.arbornetworks.com/2010/08/yoyoddos-a-new-family-of-ddos-bots/>, Accessed November 23rd, 2011.
- [29] Laboratory of Cryptography and System Security (CrySyS). Duqu: A stuxnet-like malware found in the wild. <http://www.crysys.hu/publications/files/bencsathPBF11duqu.pdf>, Accessed August 14th, 2012.
- [30] Laboratory of Cryptography and System Security (CrySyS). skywiper (a.k.a. flame a.k.a. flamer): A complex malware for targeted attacks. <http://www.crysys.hu/skywiper/skywiper.pdf>, Accessed August 14th, 2012.
- [31] Federal Bureau of Investigations. Operation ghost click, international cyber ring that infected millions of computers dismantled. http://www.fbi.gov/news/stories/2011/november/malware_110911/malware_110911, Accessed November 23rd, 2011.
- [32] Hilarie Orman. The morris worm: A fifteen-year perspective. *IEEE Security and Privacy*, 1:35–43, 2003.

- [33] HoneyNet Project. Voip security. <http://www.honeynet.org/files/HPW2011>, Accessed January 5th, 2012.
- [34] The HoneyNet Project. *Know Your Enemy : Learning about Security Threats (2nd Edition)*. Addison-Wesley Professional, May 2004.
- [35] PWC. Cybercrime: protecting against the growing threat. http://http://www.pwc.com/en_GX/gx/economic-crime-survey/assets/GECS_GLOBAL_REPORT.pdf, Accessed December 12th, 2011.
- [36] New Scientist. Botclouds: a cyberattacker's dream. <http://www.newscientist.com/article/mg21028175.500-botclouds-a-cyberattackers-dream.html>.
- [37] Securelist. Tdl4, top bot. http://www.securelist.com/en/analysis/204792180/TDL4_Top_Bot, Accessed November 23rd, 2011.
- [38] Brett Stone-Gross, Marco Cova, Bob Gilbert, Richard Kemmerer, Christopher Kruegel, and Giovanni Vigna. Analysis of a botnet takeover. *IEEE Security and Privacy* <http://www.cs.ucsb.edu/~seclab/projects/torpig/torpig.pdf>.
- [39] Symantec. Symantec's november intelligence report. http://www.symanteccloud.com/mlireport/SYMCINT_2011_11_November_FINAL-en.pdf, Accessed December 12th, 2011.
- [40] International Telecommunication Union. Itu study on the financial aspects of network security: Malware and spam. <http://www.itu.int/ITU-D/cyb/cybersecurity/docs/itu-study-financial-aspects-of-malware-and-spam.pdf>, Accessed November 23rd, 2011.
- [41] Chen Wei-Yu, Kuo Wen-Chieh, and Wang Yao-Tsung. Building ids log analysis system on novel grid computing architecture. pages https://trac.nchc.org.tw/grid/raw--attachment/wiki/deliverable09/Building_IDS_Log_Analysis_System_on_Novel_Grid_Computing.pdf, 2011.

-
- [42] wired.com. How digital detectives deciphered stuxnet, the most menacing malware in history. <http://m.wired.com/threatlevel/2011/07/how-digital-detectives-deciphered-stuxnet/all/1>, Accessed November 23rd, 2011.
- [43] James Wyke. Zeroaccess. <http://sophosnews.files.wordpress.com/2012/04/zeroaccess2.pdf>, Accessed August 14th, 2012.
- [44] Cui Xiang, Fang Binxing, Yin Lihua, Liu Xiaoyi, and Zang Tianning. Andbot: towards advanced mobile botnets. In *Proceedings of the 4th USENIX conference on Large-scale exploits and emergent threats*, LEET'11, pages 11–11, Berkeley, CA, USA, 2011. USENIX Association.