

Botnets

An In-depth Analysis

Matthew Boyd
Undergraduate Student
College of Information Sciences and Technology
The Pennsylvania State University
University Park, PA 16802
5/7/2008

Botnets have become a major force on the internet. As a tool for attacks, botnets can be a devastating force, able to render remote networks unusable. As a tool for profit, botnets have become a lucrative investment and a gateway for organized crime online. This report provides an overview of botnets including how they work, how they have evolved, and how they can be prevented from infecting and attacking networks.

Table of Contents

Introduction	3
Anatomy of a Botnet	3
Communication Techniques	4
Common Spreading Techniques	4
Botnet Lifecycle	5
Botnet Uses and Strategies.....	6
Ongoing Research	8
Evolution of Botnets	8
Early Progress	8
The First Widespread Botnets.....	9
Enhancements	10
Recent Botnets.....	10
The Future of Botnets	11
Prevention, Detection, and Mitigation	12
Prevention	12
Detection	13
Mitigation.....	14
Conclusion.....	16
Works Cited.....	18

Introduction

It only took a few minutes to take down 10 million blogs and online communities on May 2, 2006. Six Apart, the company that owns blogging sites LiveJournal and Typepad, noticed an increase in network congestion that afternoon hours which began bogging down their sites. However, after about 15 minutes servers stopped responding completely to requests. Huge amounts of data choked the network. Unfortunately, the network traffic increase was not associated with a furious rush of new visitors. Instead, Six Apart had become a victim of a Distributed Denial of Service (DDoS) attack, which had been coordinated by an army of computers around the world unwillingly participating in a botnet.

Six Apart was actually caught in the middle of attack on one of its customer's blogs: BlueSecurity.com. Blue Security had provoked the assault in part because it provided a do-not-spam registry called Blue Frog, which provided software to essentially reverse-attack junk e-mail sources targeting Blue Frog subscribers. The ethics of the Blue Frog service were debatable, and the attack was retaliation from some of the biggest spammers in the world. After two weeks of fierce and repeated attacks on the Blue Frog service and any Internet Service Provider hosting it, Blue Security discontinued the service and the company went out of business. The sheer volume of the botnet attack was practically unstoppable (Berinato 1).

Since the beginning of the 2000s, botnets have become an increasingly powerful, complex, and dangerous force on the internet. Created by installing malicious remote control applications on millions of vulnerable computers worldwide, botnets are capable of generating attacks as well as revenue for their owners by stealing the computing resources of their victims. This document analyzes the anatomy and history of botnets in-depth and also looks at steps toward prevention, defense, and mitigation against what is arguably one of the biggest threats to the internet today.

Anatomy of a Botnet

Botnets have continued to increase in complexity over the years as the popularity of the internet has exploded and computers have reached the hands of millions of users across the world. Combined, there is an extraordinary amount of idle computing power available that is ripe with possibilities for an attacker. Attackers typically create and use botnets for several common reasons, including profit.

Botnets can vary in the way they control and communicate across networks as well as what their intended purpose is. However, the general principle behind a botnet is this: "A botnet is an army of compromised machines, also known as 'zombies,' that are under the command and control of a single 'botmaster.'" (Cisco Systems, Inc)

Botnets work by infecting machines on the internet with software that allows the owner of the botnet to remotely control and command it. There are several key players involved in a botnet:

- **Attacker Machine**

The attacker is the host that is trying to infect another machine or machines. It will attempt to exploit a weakness in a remote host in order to install a bot application. It may also send commands to bots (Puri 4).

- **Victim Machine**
Sometimes referred to as zombie or botclient, the victim machine is where bot software is installed. In order to install the bot application, an attacker must first find a way to compromise the system. This can be done through application or operating system exploits as well as with viruses or worms that an unsuspecting user executes (Puri 4).
- **Bot**
A bot is an application installed on a victim machine that connects it to the botnet. After a machine is compromised, the payload of the attack is the bot application. When a bot is installed, it usually changes several settings within the system so that it is run automatically whenever the computer is started. It will also connect to a communications channel such as an IRC server in order to wait for commands (Puri 3).
- **Control and Command Channel**
This is the communications method used to send out commands to zombies. The most common method for communicating between bot controllers and zombies is through Internet Relay Chat (IRC) servers. Once a machine is infected, the bot application will typically join a private IRC channel and wait for commands from a channel operator (Puri 4).

Communication Techniques

The most common method for botnets to communicate is the Internet Relay Chat (IRC) protocol. This protocol was designed for instant messaging using client-server architecture. IRC servers host both public and private chat rooms known as channels. Participants within a channel can have one of two modes of access. In operator mode, the user has elevated privileges that allow them to control the room. In user mode, the user is simply a participant in the room and cannot control room settings.

Typically, bots or zombies will enter a channel on an IRC server specified by the bot owner. The bot will have user privileges while a bot master, or controller, will have operator privileges.

Bots will listen for command messages sent across the channel by an operator (Robot Wars - How Botnets Work).

Common Spreading Techniques

Botnet creators use several common tools and techniques to increase their zombie numbers. Once a victim machine has been infected, many bot applications will try to invade other machines both within the local network and on the internet using automated methods:

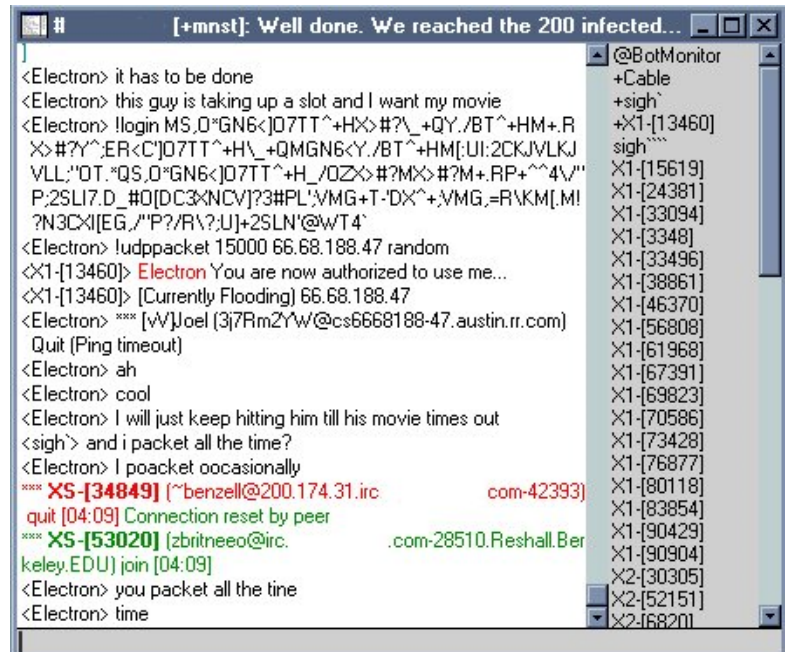


Figure 1A botnet owner issues commands to bots in an IRC channel

Scanning and Exploiting

Some bot applications scan the local network for potential victims. Tools and commands can be used to gather a list of other hosts on the network. A port scanner may be used to find hosts with open ports that are related to common services. Using a combination of this information, bot applications can begin attempts to exploit vulnerabilities of services running on other hosts. Once an exploit is successful, the bot application can be installed on the compromised host (Schiller, Binkley and Harley 42).

Sniffing

In addition to trying to look for exploits in other network clients, bot applications can be programmed to sniff network traffic for usernames and passwords to other resources on the network. This is especially useful if the password is sent in clear text. The bot can then utilize the user credentials to gain access to another network resource and infect it (Schiller, Binkley and Harley 42).

Password-Guessing and Reusing

In some scenarios, bots will simply try to guess the password of common usernames such as 'Administrator' in order to gain access to other machines connected to the network. Bot applications can guess passwords at a rate that is much faster than humans can type, allowing them to try a large number of passwords in a small amount of time. This is why it is important to choose strong passwords for all user accounts and, if possible, enable policies that lock out a user after a certain number of attempts in a given time period (Schiller, Binkley and Harley 185).

In some cases, a bot application may be able to retrieve a file containing the encrypted or hashed credentials of users on the local machine. From there, password cracking tools can sometimes retrieve the unencrypted password of user accounts. These passwords can then be used to try and gain access to other machines on the network (Schiller, Binkley and Harley 43).

Social Engineering

Bots such as those in the Global Threat (GT) Bot family can use social engineering tactics to trick the victim into visiting a malicious website or running a malicious executable capable of infecting their machine. Spam e-mails containing malicious links or file attachments can be sent to harvested e-mail addresses. The messages may be disguised as legitimate messages from friends, businesses, or other identities of significance. In some cases, these e-mails have even claimed to contain important security information or files that the user needs to install. However, if the user clicks on the link or opens the attachment, they are infected with a bot application and are now participating in a botnet (Schiller, Binkley and Harley 8).

Botnet Lifecycle

Botnets share a common lifecycle that is dependent on the clients participating within the botnet. Each botclient will go through a similar process:

- **Exploitation**

First, the vulnerable client is infected with bot software so that it can be controlled remotely. The exploit may be a result of an unpatched vulnerability or malicious code that an unsuspecting user executed on the local machine (Schiller, Binkley and Harley 31).

- **Rallying**
Typically, once the botclient is infected, it will notify a botherder that it is now a participant in the botnet. This is done using the botnet owner's communications method of choice. However, most botnets currently use IRC channels for communication and control (Schiller, Binkley and Harley 37).
- **Security and Evasion**
Once the machine has been infected, bot software may try to take measures to cover its tracks and remain an active host. The botclient may try to download additional modules that work to prevent the user from knowing that their machine is compromised or it could install itself as a rootkit hidden deep within the Operating System. To further prevent detection, many botnets try to disable common anti-virus software installed on the machine. In some cases, malicious code may try to hijack and quietly subdue the antivirus software so that user believes it is still running properly (Schiller, Binkley and Harley 37).
- **Receiving Commands**
For the majority of its life, a botnet client will run quietly on a machine and wait for commands from a botherder or botnet owner. Botclients will typically listen for commands on an IRC channel. When a command is sent out across the channel, the botclient will process and execute it (Schiller, Binkley and Harley 41).
- **Death**
When a bot owner can no longer participate with one of its clients, it is no longer considered a part of the botnet. This can happen for several reasons, including detection or removal of the malicious bot software by the victim. If the botherder suspects that their cover has been blown, they may take steps to cover their tracks by executing a command that tells the bot software to erase itself. In some cases, security and antivirus logs may be cleared as well to hide any evidence that the machine has been compromised or tampered with (Schiller, Binkley and Harley 62).

Botnet Uses and Strategies

Infecting machines across the world with control mechanisms would be meaningless without some kind of incentive for the attacker. Unfortunately, bots are used for network attacks as well as making profit. Recently, more botnets are being designed to help their owners make profit instead of simply denying services.

Denial of Service

One of the first uses of botnets was in Denial of Service (DoS) attacks. These attacks seek to overload a target, such as a website or server, until it is completely unresponsive to legitimate traffic. Specifically, botnets most often perform a DoS attack by commanding zombies from all over the internet to send huge amounts of network traffic to a target. This is known as a Distributed Denial of Server (DDoS) attack. Attackers use several techniques to perform these attacks.

TCP SYN Flood

In a TCP SYN Flood attack, the attacker commands zombies to send SYN messages, or the first part of the TCP handshake, without responding to follow-up requests. This attack may also be combined with IP spoofing so that follow-up requests are sent to another host. As thousands of SYN messages are sent to the server, the target's connection table fills up and large amounts of

memory must be allocated to each connection attempt until it exceeds the timeout period. Eventually, the server may not be able to handle any more SYN requests and therefore may start dropping traffic (Schiller, Binkley and Harley 48).

UDP Flood

Using the UDP protocol, an attacker may attempt to overwhelm a server by having botnet zombies simultaneously sent small UDP packets to the target. These packets are sometimes sent on random open ports. Once the packets reach the target, it must allocate resources to handle them. If enough packets reach the target, it can cause other traffic to be dropped (Schiller, Binkley and Harley 49).

Smurfing

Smurf attacks involve sending an overwhelming number of ICMP ping requests to a broadcast address while spoofing the source IP address. The spoofed source IP address is often the target of the attack since each host in the broadcast range will attempt to reply to the source address that was specified. This can overwhelm the target with requests. However, there are newer techniques to filter this spoofed, or illegitimate, traffic at the border of the network, preventing this kind of attack from being effective (Schiller, Binkley and Harley 49).

Scams and Identity Theft

There are several ways that an attacker can leverage the processing power of a botnet to their advantage. However, money can often be a strong motivator for botnet creator. Many botnet owners have tried using their powerful resource to turn a profit using several common approaches:

- **Spam Attack**
Many botnets are used to distribute spam across the internet to unsuspecting recipients. Instead of trying to send spam from several large e-mail servers that can easily be blacklisted by network administrators, each bot in a network may act as an e-mail server with a unique address, sending large amounts of bulk e-mail across the internet to seemingly random addresses. However, bots can also be used to harvest the e-mail addresses located in address books on the infected machine. This allows the attacker to create an enormous list of e-mail addresses to send spam to (Bächer, Holz and Kötter).
- **Keylogging Attack**
Some botnets are organized to log keystrokes on each infected machine in order to steal sensitive information from users. In some cases, some bots can also filter logged keystrokes for keywords that may indicate a username and password sequence has been entered. For example, the keyword “Ebay” could indicate that someone is typing the address to Ebay and is about to sign in with their credentials. This can be more effective than sniffing network traffic for user credentials since it bypasses any encryption that web sites or services are using (Bächer, Holz and Kötter).
- **Click Fraud Attack**
Online advertising programs that offer site owners compensation for advertisement clicks have also fallen victim to botnet attacks. By programmatically commanding infected machines to click on an advertisement on a web page, an attacker can generate thousands of unique, but illegitimate, clicks instantly for a substantial profit (Bächer, Holz and Kötter).

- **Poll and Game Manipulation Attack**

Botnets have also targeted online polls and games in order to manipulate the outcomes. Many online polls base unique votes on IP address. However, since each infected machine in a botnet may have a unique address, the attacker can command each machine to vote. Many games also track points or statistics by unique IP address, allowing botnets yet another way to skew results (Bächer, Holz and Kötter).

Recruiting

The only way to rapidly create a massive botnet is by continuing to infect new victim machines. Botnets may scan for potential victims within the local network of a victim machine in order to spread. Bot software may include network traffic sniffers, port scanners, and other network analysis tools to look for other vulnerable systems on a subnet (Schiller, Binkley and Harley 42).

Ongoing Research

As botnets continue to evolve and become more complex, security researchers must find ways to analyze them in an effort to better understand how they work and, more importantly, how they might be stopped. One of the most popular methods for tracking and analyzing botnets is the honeypot. A honeypot is a network host set up by security researchers and disguised as a vulnerable machine that attackers might target. Once the honeypot has been compromised by an attacker, security researchers can analyze the bot in order to get information about the attack and its source (Honeypots).

Honeypots can generally be grouped into two different categories:

- **Low Interaction**

Honeypots that limit what an attacker can do with a system are considered low interaction. The host or network services within the honeypot are normally emulated so that no device is actually infected. This method is primarily used for malware collection.

- **High Interaction**

When actual services and software are used and system compromises are not simulated the honeypot is considered high interaction. In this case, a system is closely monitored while an attacker compromises and infects the system. These honeypots allow security researchers to get a better understanding of how attackers and malware are modifying a system in order to use it to their advantage. It can also help researchers discover new, stealthier exploits (Honeypots).

Evolution of Botnets

In only two decades botnets have gone from simple scripts that demonstrated primitive artificial intelligence and automation to a dangerous and devastating force on the internet. The same technologies that were originally intended for good have become more intelligent, resilient, and malicious than ever.

Early Progress

The first bot was created by a pioneering IRC server operator in 1989. Unlike the bots and botnets of today, this bot application had a very simple task: it played Hunt the Wumpus with IRC users connected to the server.

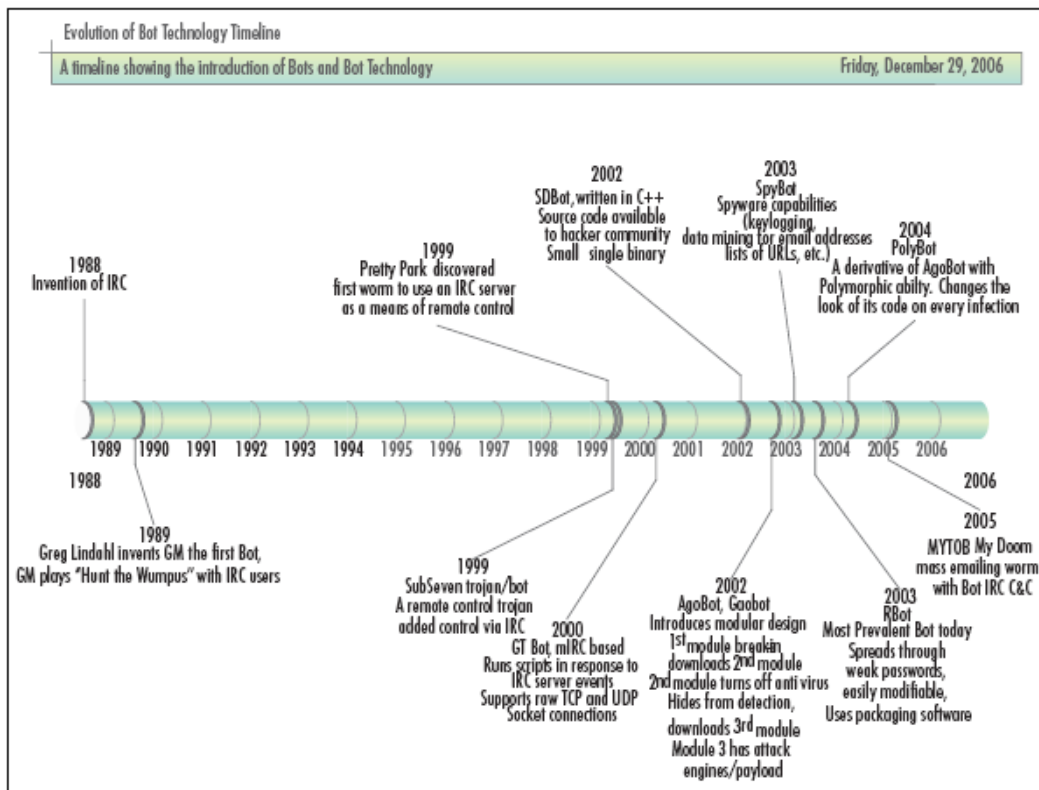


Figure 2 A timeline of botnet history and evolution.

Early on, IRC operators found that bots had a lot of possible uses. They could act and perform operations as users in an IRC channel 24 hours a day. This allowed operators to maintain control of their IRC channels, even when they were sleeping.

Despite the use of bots on IRC channels in the early and mid 1990s, the first botclient application was not discovered until 1999. Called PrettyPark, this bot application had the functionality of many of today's botclients. It was capable of retrieving system information, e-mail addresses, and user credentials and could redirect traffic, launch DoS attacks, and complete file transfers. Most importantly, the bot was able to update itself and act as an IRC client (Schiller, Binkley and Harley 9).

Another bot application appeared in 1999 that made more progress towards powerful, large-scale botnets. The bot application SubSeven featured remote control capabilities through a bot on an IRC channel. The SubSeven application was capable of stealing passwords, logging keystrokes, and giving administrative rights to the bot operator. Because the creator of SubSeven marketed it as remote administration tool, it was also considered a Trojan disguised to trick users into installing it (Schiller, Binkley and Harley 9).

The First Widespread Botnets

In 2002, a Russian programmer going by the internet handle 'sd' created a new botclient in C++ that was capable of exploiting critical vulnerabilities in order to infect victim machines. Originally intended for DoS attacks, SDBot was capable of exploiting vulnerabilities in common Windows Services including NetBios, DCOM, MS RPC, and UPNP. It also targeted WebDav and MSSQL server vulnerabilities as well as Cisco router vulnerability. In addition to the vulnerabilities that could be exploited to infect machines, SDbot could also exploit backdoors that were found in other Trojans and botclients such as SubSeven (Schiller, Binkley and Harley 10).

What was most revolutionary about SDBot is that its creator published the source code to the bot software on a website. This made the software available to the masses for further development and customization. It also made it possible for less technical or experienced hackers to create their own botnet with relatively simple modifications to the source code, resulting in countless variants being created and distributed (Schiller, Binkley and Harley 10).

Since 2002, SDBot's variants have continued to adapt to attempts by software developers to end the spread of the botclient. A 2006 report found that SDBot.ftp.worm was one of the most frequently detected viruses. Additionally, a report released by Microsoft in 2006 stated that their Malicious Software Removal Tool had detected 678,000 computers infected with SDBot (Schiller, Binkley and Harley 10).

By 2003, bots were increasingly transforming from a method for attacking networks to a way of collecting data on their victims. A successor of SDbot, known as Spybot and also referred to as Milkit, included spyware capabilities. Spybot could collect web form data, e-mail addresses, and even web browser histories. Spybot spread through file-sharing applications, system vulnerabilities, and backdoors in other bot applications. Like SDbot, its customizability helped with the spread of variants that included additional capabilities. The offspring bot applications of Spybot were capable of port scanning, logging keystrokes, capturing the contents of the Windows clipboard, taking screenshots, and killing security processes such as antivirus. Like SDBot, variants of this bot application can still be found in the wild today (Schiller, Binkley and Harley 13).

Enhancements

The latest botnets continue to introduce new techniques for infecting and controlling victim machines without detection. A 2006 report found that there were approximately 1.9 million computers infected with the RBot family of bot applications. This bot was one of the first to use software package encryption tools to further protect itself from prying eyes (Schiller, Binkley and Harley 15). Encryption software such as MoleBox is capable of encrypting application files in order to prevent third parties from disassembling them (MoleBox Pro).

Yet another step in bot evolution occurred in 2004 when Polybot appeared. This was one of the first bot applications to utilize polymorphism in an attempt to evade detection. Polybot morphs itself upon every infection by hiding its compiled code within envelope code that re-encrypts itself every time it is executed (Schiller, Binkley and Harley 15).

Recent Botnets

On January 19, 2007 a seemingly typical e-mail virus began to circulate with a malicious link claiming to be a news story about a deadly storm. However, the e-mail was actually part of a strategy to create one of the most massive self-replicating botnets ever. The Storm botnet is now one of the largest botnets on the internet. A report from Symantec estimated that Storm peaked in July 2007 with 1.7 million infected machines (Acohidio and Swartz).

Storm is revolutionary in that it uses peer-to-peer communications instead of IRC to command and control botclients. Storm is also incredibly resilient. The command server can move across compromised systems so that if a server is shut down the botnet can still be controlled. The botnet is also very aggressive when it comes to probing or blocking attempts. If a host is detected probing the botnet, a DoS attack can automatically be initiated against that host. Storm continues

to be one of the largest botnets on the internet with several hundred thousand victim machines participating in it (Acohidio and Swartz).

The Future of Botnets

As long as there is money to be made and weaknesses to exploit, botnet creators are likely to progressively improve their spreading techniques while also finding new uses for their infected clients. Recent capabilities found in newer botnets are an indication that this may only be the beginning.

Ransomware

Newer bot components have been discovered that could potentially hold user data hostage. A component known as GpCoder encrypts user files so that they are inaccessible and then sends the user a message telling them how to buy a decryption application so they can regain access to their files. Although the encryption performed by current versions of GpCoder can be reversed with repair tools provided by antivirus software vendors, future versions could see stronger encryption and more complex techniques. The result could be thousands of users with files that are held for ransom (Schiller, Binkley and Harley 15).

Illegal Content Sharing

Newer botclients are being used to distribute illegal content such as pirated software, movies, music, and games through victim machines. Even child pornography has been shared in this method, which could potentially leave the owner of the machine facing criminal charges for a crime they did not commit.

On December 16, 2004, police raided the home of 16-year-old Matt Bandy's family and alleged that child pornography had been uploaded to a Yahoo Group from the Bandy's computer. Matt Bandy admitted to using a Yahoo Group for looking at adult images but not child pornography. As it turned out, the Brandy's computer was infected with over 200 viruses and malware applications, several of which allowed remote control of the computer. Matt was arrested and charged with nine counts of a class 2 felony. Eventually, Matt's charges were reduced to a misdemeanor because the defense proved that it was likely the child pornography found on the computer was put there by an intruder, but not before facing up to 90 years in prison and two years of court battles that cost the family their life savings (Bernstein 1).

Cases similar to Matt Bandy's are likely to appear more often in the future as botnets are used more to obscure the identity of people committing illegal activities and sharing illegal content online. The Serv-U ftp server allows botherders to use botclients as storage servers for illegal content distribution. Data is stored in hidden directories on the computer and the ftp server process is disguised as Windows Explorer to avoid detection (Schiller, Binkley and Harley 15).

New Spamming Mechanisms

Botnets are likely to continue to be used for spamming as it can bring large profits to botnet owners. However, attackers are likely to spread spam messages in newer ways besides just e-mail. For example, several recent bots have been found sending Spam for Instant Messaging (SPIM). Like e-mail spam, SPIM can be disguised as a message from a friend while actually containing phishing attacks or links to malicious files. However, most instant messaging clients have no way of filtering or detecting spam messages, increasing the likelihood that the messages will reach a victim (Schiller, Binkley and Harley 16).

Larger Attacks and Digital Terrorism

As botnets continue to evolve, their attacks may also become more coordinated and more devastating. An August 2007 article from Wired Magazine hypothesizes that clusters of botnets from around the world could be coordinated to attack and take down the information infrastructure of the United States by targeting key web and e-mail servers across the country. Figure 3 shows how an attack on US network infrastructure may look from a world view (Robb 1).

The capabilities of botnets make them an ideal tool for terrorizing and attacking rival countries at an international level. Furthermore, the source of these attacks can be obscured so that the blame cannot be placed squarely on any government or terrorist group. The future may hold attacks like the one proposed in Wired Magazine if computer and network security is not taken seriously by governments, businesses, and individuals in an effort to stop or prevent botnets.



Figure 3 A map that graphically shows hypothetical attack on key US network targets.

Prevention, Detection, and Mitigation

Although botnets pose a great risk to networks around the world, network security researchers have found ways to protect networks from botnets and mitigate the threat level and impact of an attack. However, in order to begin bringing the world of botnets to its knees, network administrators need to take measures to reasonably protect their systems from infection first and foremost. This may involve installing network security appliances, keeping systems updated with the latest patches and hot fixes, and installing security software on systems connected to the network. Network users also need to be educated about common techniques used to trick them into becoming an unwilling participant in a botnet and what is at stake.

Prevention

Being proactive about network and computer security is an important part of thwarting a botnet. By taking certain measures to protect machines, the risk of networked computers becoming infected with bot software is greatly reduced.

Patches, Updates, and Antivirus

The easiest and most critical way to protect computers from bot infections as well as other malware is to make sure that Operating System updates are automatically downloaded and installed. This is because bots routinely use known exploits and vulnerabilities to compromise

and infect a system (Puri 10). Additionally, application updates are also important as they address and resolve vulnerabilities that exist within specific applications. Vulnerabilities in specific applications have also been exploited by malware to compromise machines. Software such as Secunia's Personal Software Inspector and Network Software Inspector can scan systems for applications that have known vulnerabilities and need updated (Secunia Personal Software Inspector).

Antivirus software is another key component of preventing bot infections. However, it is not infallible and must be routinely updated in order to be effective. Specifically, antivirus software that scans e-mail attachments and file downloads automatically as they arrive is most effective (Schiller, Binkley and Harley 431).

Firewalls

Firewalls are also another way to prevent networks and their hosts from becoming infected with bots. Inbound ports commonly used to compromise and infect machines should be blocked or at least limited to subnets. Software firewalls are also useful for protecting internal hosts from other hosts that may be compromised, both inside and outside of the network (Schiller, Binkley and Harley 431).

Education and User Intervention

Arguably, the most difficult method of preventing infection is in the hands of the user. Many infections can be avoided if users exercise caution when browsing the internet and checking e-mail. However, users must first have at least a basic understanding of what threats exist and how they should be avoided. Teaching users to be aware of common scams and phishing techniques is important. Training them to avoid executing unknown e-mail attachments and downloads is also necessary for an effective defense. Users also need to understand why strong passwords should be used for all of their accounts, especially local user accounts since some bots spread by guessing weak user passwords (Schiller, Binkley and Harley 432).

In large or enterprise networks where hundreds of hosts and users exist, depending on every user to be responsible, knowledgeable, and aware of bot and malware techniques is impossible. In this case, limiting user rights on a system may be the only way to effectively prevent bot infections. Prohibiting users from installing applications on their own can prevent compromises due to unpatched vulnerabilities. Furthermore, limiting user privileges can stop executables from accessing or modifying system files since they are being run without administrative rights (Schiller, Binkley and Harley 427).

Detection

In many cases, botnet infections and attacks are unavoidable. However, network users and administrators must be aware of compromises and attacks in order to remediate them. Several tools and techniques exist for detecting botnet infections when preventative measures have failed. There are also tools that help network administrators detect when they've become a victim of an attack. Note that this section only covers several of the most common ways to detect botnets.

Logs

Not only are system logs useful for troubleshooting system issues, they are invaluable for detecting bot infections and infection attempts. Security logs can be an indicator that an infected

host has attempted to access a machine. Particularly, large numbers of failure attempts for logins, policy changes, and privileged permissions may be an indication that something is trying to gain control of the host. This is why it is important to ensure that logging is enforced on each host (Schiller, Binkley and Harley 433).

Network Monitoring and Intrusion Detection Systems

In some cases, the best way to detect botnet infections or attacks is to monitor network traffic for suspicious activity. An Intrusion Detection System (IDS) is an ideal way to monitor network packets for DoS attacks, port scans, and other network anomalies. Using what are known as signatures to detect traffic anomalies, IDS can find botnet attacks or infections and send out alerts to network administrators (Schiller, Binkley and Harley 156).

In general, there are two IDS types: Host-based and Network-based. A Host-based IDS is designed to monitor the activity of individual systems on the internal network. The Host-based IDS will monitor network hosts for abnormal activity such as attack attempts on other machines and can also watch systems for significant changes. These systems can be run on a host-by-host basis, but large networks tend to use an enterprise-class appliance for Host-based detection. With a Network-based IDS, traffic is monitored more closely at the border of the network rather than between internal hosts. This makes it useful for detecting Dos and Port scan attacks. It can also monitor outgoing traffic that may be abnormal. For example, large amounts of traffic sent out from a host on port 25 may indicate an infected machine is distributing spam (Schiller, Binkley and Harley 156).

One popular lightweight Network-based IDS is Snort. This IDS is free and open-source so that network administrators and programmers alike can take advantage of its usefulness and customize it to accurately meet their needs. Snort uses rules, or signatures, to handle detection and alerts. There is a common syntax for each rule. For example, a rule written by Joe Stewart is below:

```
alert tcp any any -> any any (msg:"Agobot/Phatbot Infection
Successful"; flow:established; content:"221 Goodbye, have a good
infection |3a 29 2e 0d 0a|"; dsize:40; classtype:trojan-activity;
reference:url,www.lurhq.com/phatbot.html; sid:1000075; rev:1;)
```

This alert tells Snort to send an alert whenever an incoming packet from any port and address contains the signature content “221 Goodbye, have a good infection |3a 29 2e 0d 0a|” when there is an established connection (Schiller, Binkley and Harley 171).

Mitigation

At some point, the inevitable may happen and botnet activity may be discovered on the network. In this situation, how network administrators react can often determine how successful they are at ridding the network of a botnet infection or attack. This section covers common procedures that administrators should consider when they discover an infected machine on the network or when they have become the victim in a large-scale attack.

Reacting to a Botnet Infection

Disconnect

In the event that a network host is found to be infected with bot software, the host should immediately be disconnected from the network. This will help prevent the infection from spreading to other network hosts while also preventing any more data on the host from being sent back to the botnet owner. It will also protect other systems on the internet from potential attack or infection (Puri 11).

Backup

Once the host is off the network and cannot harm other hosts, it is best to retain a complete backup of the entire contents of the hard drive for forensic and investigative purposes. To do this, the physical hard drive can be removed from the system and replaced with a spare (Schiller, Binkley and Harley 434). However, if this is not an option there are several disk imaging tools available that can create a snapshot of the contents of the hard drive. One free example is DriveImage XML, which can create a sector-by-sector compressed image of a hard drive (DriveImage XML).

Clean

After a backup has been created, the computer can be cleaned or rebuilt. Antivirus software, root kit cleaners, and antispyware software can all be helpful in cleaning a system. However, they are not completely foolproof, especially with many of the newer and more complex bots. Instead, it is probably easier and safer to completely reformat the hard drive and reinstall the operating system and user software. Before restoring any backup files, be sure to scan them for viruses and malware. Also make sure the operating system, antivirus, and other software are up-to-date and patched to minimize the possibility of re-infection (Schiller, Binkley and Harley 435).

Report

Depending on what information that was stored or transmitted on the infected machine, reporting a compromise is one of the most important aspects of damage control with respect to botnets and computer malware. If the compromised machine contained private customer data, such as social security or credit card numbers, the company may need to alert customers of a security breach. In 2007, the United States Senate introduced several bills mandating that businesses and government agencies notify consumers and law enforcement in the event that personal data may have been compromised. Otherwise, they could face several penalties. This is also why it is important to make a backup of the hard drive contents (Bosworth 1).

Additionally, botnets and other illegal internet activity can be reported to the Internet Crime Complaint Center (IC3). IC3 is a partnership between the Federal Bureau of Investigation, National White Collar Crime Center, and Bureau of Justice Assistance in an effort to provide the public with a way to file online cybercrime complaints (Internet Crime Complaint Center).

Any users of the infected computer should also be advised to assume that their usernames and passwords to any and all online services may be compromised. They should change their passwords as quickly as possible to prevent attackers from stealing their online identities (Puri 11).

Reacting to Attacks

As the introduction of this report details, stopping DDoS attacks from botnets is difficult, even for robust networks. The amount of network traffic that botnets can deliver to a target is simply overwhelming. However, there are a few techniques that network administrators can use to try and fend off a DDoS attack.

Egress and Ingress

In order to block spoofed or unwanted IP addresses, Ingress and Egress filtering can be used on routers at the border of the network. Like firewall rules, this filtering technique can block incoming and outgoing traffic that meets certain criteria. Egress filtering can examine packet headers that are leaving a particular subnet to see if they have a valid address. In the event that the packet is invalid, it is dropped with little overhead or network degradation (DDoS Mediation Action List).

Ingress filtering has functionality that is very similar to Egress filtering except it monitors incoming packets. This is especially important when fending off incoming DDoS attacks. Using IP and subnet tables, network traffic can be dropped at the border of the network to stop attacks from reaching internal hosts (DDoS Mediation Action List).

Move the Target

In some cases, the scope of a DDoS attack can be reduced by determining the target and moving it. This may involve updating the IP addresses and DNS names of target systems. Removing the routes to the old IP address may also be necessary. This will result in attack packets being sent to an invalid IP address and hopefully allow the service as well as the network to recover. However, the attacker may detect the change and retarget the service. Because of this, it may be necessary to repeatedly and periodically change the IP addresses and DNS names of the target systems until the source of the attacks stops (IBM Internet Security Systems 4).

Conclusion

The impact that botnets have had on networks and the internet since their inception really demonstrates one of the greatest threats to computer systems in a digital age. If previous and current botnet strategies are any indication, the problem is likely to get worse before it gets better. The complexity of botnets continues to increase. New methods for intrusion, infection, and evasion are making it increasingly difficult to track and detect bots. However, researchers are working hard to monitor botnet trends, sometimes going as far as intentionally hosting participant bots in order to understand how they work.

Many have equated the mentality of botnet creators and owners to the gangs of Chicago in the 1920s. Botnets have become a method for committing organized crime online. The ruthless style of Al Capone and his henchmen is evident: Botnets will find any way they can to spread and make money and they will attack anyone or anything that gets in their way.

There are several ways that individuals and groups can lessen the threat of botnets. Keeping systems up-to-date, installing firewalls, and making users aware of common social engineering strategies are a few of the ways individuals as well as network administrators can prevent their computers from becoming part of a botnet. This could suffocate botnets by eliminating the

resources they need to become massive. However, preventative efforts must be adopted by the masses to be effective. Law enforcement must also step up and address the botnet issue worldwide. Governments need to coordinate in an effort to find, apprehend, and punish botnet creators. The war on botnets has just begun.

Works Cited

- Acohido, Byron and Jon Swartz. "Botnet scams are exploding; Typically, 40% of computers hooked up to Internet are infected." USA Today 17 March 2008: 1B.
- Bächer, Paul, et al. "Know your Enemy: Tracking Botnets." 13 March 2005. The HoneyNet Project. 1 April 2008 <<http://www.honeynet.org/papers/bots/>>.
- Berinato, Scott. "Attack of the Bots." November 2006. Wired Magazine. 4 May 2008 <<http://www.wired.com/wired/archive/14.11/botnet.html>>.
- Bernstein, Jonathan. "The Matt Bandy Story." 2006. Justice4Matt.com. 5 May 2008 <<http://www.justice4matt.com/MattsStory.html>>.
- Bosworth, Martin. "Senate Moves Ahead with Data Breach Bills." 4 May 2007. Consumer Affairs. 5 May 2008 <http://www.consumeraffairs.com/news04/2007/05/senate_data.html>.
- Cisco Systems, Inc. "Botnets: The New Threat Landscape White Paper." Cisco Systems. 30 April 2008 <http://www.cisco.com/en/US/solutions/collateral/ns340/ns394/ns171/ns441/networking_solutions_whitepaper0900aecd8072a537.html>.
- "DDos Mediation Action List." 3 April 2000. Computer Incident Advisory Capability. 5 May 2008 <<http://www.ciac.org/ciac/bulletins/k-032.shtml>>.
- "DriveImage XML." Runtime Software. 5 May 2008 <<http://www.runtime.org/driveimage-xml.htm>>.
- "Honeypots." Shadowserver. 6 May 2008 <<http://www.shadowserver.org/wiki/pmwiki.php?n=Information.Honeypots>>.
- IBM Internet Security Systems. "Distributed Denial of Service Attack Tools." 1 March 2000. Internet Security Systems. 5 May 2008 <<http://documents.iss.net/whitepapers/ddos.pdf>>.
- Internet Crime Complaint Center. 5 May 2008 <<http://www.ic3.gov/>>.
- MoleBox Pro. 5 May 2008 <<http://www.molebox.com/>>.
- Puri, Ramneek. "Bots & Botnet: An Overview." GSEC Practical Assignment. 2003.
- Robb, John. "When Bots Attack." 23 August 2007. Wired Magazine. 5 May 2008 <http://www.wired.com/politics/security/magazine/15-09/ff_estonia_bots>.
- "Robot Wars - How Botnets Work." 20 October 2005. WindowSecurity.com. 30 April 2008 <<http://www.windowsecurity.com/articles/Robot-Wars-How-Botnets-Work.html>>.
- Schiller, Craig, Jim Binkley and David Harley. Botnets: The Killer Web App. Syngress Publishing, 2007.
- "Secunia Personal Software Inspector." 2008. Secunia. 5 May 2008 <<https://psi.secunia.com/>>.

