

Threats That Computer Botnets Pose to International Businesses

By

Matthew West

A Research Paper

Submitted to the Faculty of D'Youville College

Department of Business

In partial fulfillment of the requirements for the degree of

Master of Science

In

International Business

December 3, 2008

Copyright 2008 by Matthew West. All rights reserved. No part of this paper or project, unless noted, may be copied or reproduced in any form or by any means without written permission of

Matthew West.

**ABSTRACT**

In the past few years the threat of computer botnet attacks has become an increasing concern among information technology managers. Botnets are networks of computers that have been infected with malicious software that can be remotely controlled by an attacker. They are significant contributors to malicious and criminal activity on the Internet and their attacks often directly target businesses. These attacks can result in the loss of data and service downtime holding the company financially liable. This paper outlines why information technology managers must be aware of the botnet problem and provides a guide for developing security policies that will enable staff to recognize and defend against botnet attacks that target technology services.

**TABLE OF CONTENTS**

	Page
ABSTRACT.....	iii
LIST OF FIGURES .....	vi
INTRODUCTION .....	1
STATEMENT OF PURPOSE .....	6
DEFINITION OF TERMS .....	7
THEORETICAL FRAMEWORK.....	9
LIMITATIONS.....	10
LITERATURE REVIEW .....	11
Botnet Motivations.....	11
Botnet Functions.....	13
Botnet Operating Structures.....	16
Tracking Botnets.....	28
PROCEDURES .....	31
RESULTS .....	32
Case 1 – Web Site Comment Spam .....	29
Case 2 – Suspicious Browser User Agent Strings .....	44
Case 3 – Suspicious Game Server Activity .....	50

**TABLE OF CONTENTS (Continued)**

	Page
DISCUSSION .....	55
Recognizing Malicious Network Activity .....	55
Responding to Malicious Network Activity .....	57
REFERENCES .....	59
FOOTNOTES .....	63

**LIST OF FIGURES**

Figure 1	Distributed Denial of Service Attack .....	14
Figure 2	Drive-by Download .....	19
Figure 3	A Client-to-Server Command and Control Mechanism .....	21
Figure 4	IRC Command and Control .....	22
Figure 5	Web-based Command and Control Connection Log .....	24
Figure 6	Peer-to-Peer Botnet.....	26
Figure 7	Message Board User Registration Form.....	33
Figure 8	Legitimate Web Page Request.....	35
Figure 9	Suspicious Web Page Request.....	36
Figure 10	Spam User Accounts.....	38
Figure 11	Web Server Log Entries for a Malicious User Account .....	39
Figure 12	Remote Hosts Attempting to Create Spam User Accounts .....	40
Figure 13	Remote Hosts Attempting to Login .....	41
Figure 14	Firewall Rules to Block the Malicious Hosts .....	43
Figure 15	User Browser String Patterns.....	45
Figure 16	List of Suspicious Hosts .....	46
Figure 17	Hosts Running Web Servers .....	47
Figure 18	XAMPP Web Page File Request Statistics.....	48
Figure 19	Contents of linka.txt.....	49
Figure 20	Example of Dropped Connection Attempt .....	52
Figure 21	Connection Attempts to UDP Port 12203 and 12204.....	53
Figure 22	Record Connection Request.....	54

## INTRODUCTION

Internet services have become an essential part of everyday life. Businesses worldwide rely on the convenience of Internet connected devices to buy, sell, and communicate. The Internet is easily accessible to anyone with a computer and a network connection allowing users to reach any point on the Internet without regard to national or geographic boundaries. This ease of communication allows a business to be more accessible to customers. However, if a competing business is more easily accessible then the competitive advantage is lost. As a result, businesses are under pressure to adopt Internet-based technologies to remain competitive in their industries.

As companies have begun to implement technologies that were born in the engineering and science sectors, where openness is preferred, a competition has emerged between developing technologies that are convenient to use and developing technologies that are secure. If a technology's security features make it too complicated to use, then that technology will not be adopted. Businesses are under pressure to remain competitive and often adopt technologies before they are mature. The result is a reliance on technologies that are insecure.

Today's Internet is a global array of interconnected computer networks. It was born as the Advanced Research Projects Agency Network (ARPANet) in 1969. The program was funded by the United States Department of Defense as a means to connect the country's large and powerful research computers that were spread across the country. The network was the first to successfully implement the packet switching concept that would allow a robust computer-to-computer network.

ARPANet originally used the 1822 communication protocol, which proved to be inadequate. While reliable, it could not handle multiple concurrent communication transmissions

from the same computer. The X.25 protocol solved this problem and soon the US, Europe, Canada, Hong Kong, and Australia were connected to form the worldwide network infrastructure known today as the Internet.

Although X.25 was an improvement, it needed to be expanded to allow for communication across the network. To solve this problem the email technology that had already existed as a communication mechanism amongst users on the same mainframe was extended to allow communication amongst multiple users on the same network.

Email has changed very little since its early development and works in a way similar to traditional postal mail. Unfortunately, much like postal mail, email can be used to send unsolicited commercial mail. This is more commonly referred to as “spam.” The major difference between postal mail and email is that unlike the traditional mail system, which requires the sender to pay a postage fee for each message that is sent, email carries little cost and the delivery time is significantly faster. Spam continues to be a major flaw of the email service. It is an example of a technology that was designed to be convenient for users while not considering a means to secure it. It has a low entry barrier making it easy for novice computer users to send email messages. Unfortunately, this low entry barrier also allows anyone to send a spam email message as easily as sending a regular email message.

Spam is an attack on a company’s technology resources. It monopolizes a company’s bandwidth and vital communication tools. The costs of productivity losses caused by spam email burden the staff of businesses and Internet service providers. In response, information technology managers must allocate resources by means of equipment and manpower in order to maintain email service and communication quality in their company.

Another example of a service that was designed to be highly functional is the World Wide Web. It was developed with a focus on convenience and made no consideration that users browsing the web could become a target of malicious attacks. The web is based on the Hyper-Text Transfer Protocol (HTTP) that was designed by Tim Berners-Lee in 1989 as a way to transmit and display physics data in a universal and organized fashion (Web History Project, 2003). As public use of the web grew through the 1990s additional features and plugins were added to web browser software to make them more interactive. However, this made user's computers less secure by tying the web browser more closely to the computer's underlying operating system. This technology made the web easier to use while at the same time sacrificing security.

While services like email and the World Wide Web were maturing the Internet was growing at an explosive rate. This growth was the result of a collision of 2 phenomena: Moore's law and Metcalfe's law. Moore's law was coined by Gordon Moore, the co-founder of Intel who noted the advancement in the silicon chip production process. Moore proposed that the speed and power of integrated circuits in computer processors and memory chips would double every 18 months. Rapid advancements in technology contributed to the support of Moore's law. This exponential growth allowed computers to go from gigantic mainframe servers to compact laptops within only a few years.

Metcalfe's law also contributed to the Internet's rapid growth. Robert Metcalfe is the co-creator of Ethernet and coined Metcalfe's law. The law states that every new host, an Internet connected computer that has an assigned Internet protocol (IP) address, that is added to the network dramatically increases the usefulness of the network in terms of material that is available and easily accessible. This means that as the number of Internet users increases, the

usefulness of the network increases and it becomes even more compelling over time for new users to join.

By the middle of the 1990's Moore's law and Metcalf's law were working hand in hand to fuel a rapid progression of technology. The adoption of desktop computers by businesses increased as they became faster, cheaper, and more powerful. These computers were then connected to the Internet making their technology even more useful.

As the Internet continued to grow and the diversity of users increased, society's darker side began to see it as a method for personal gain. Malicious users began to attack remote computers to gain unauthorized access. The benefit of having this remote access was to control the computer's resources and to use them as a gateway to the rest of the computer's network. Thus, making the discovery and access of other hosts on the same network easier. In addition, any activity that originated from a compromised computer would be tracked to that computer and hide the attacker's source.

Attacks by malicious users reveal a fundamental problem with technology: Technology is unable to discern human intentions. It is unable to tell when it is being used for good purposes and when it is being used for evil. So the same technological services that are provided for ethical purposes can be used against their original intention for unethical ends.

An example of a malicious user utilizing technology in an evil manner is if they obtain unauthorized access to a computer remotely. This is often achieved by guessing common user account names and their corresponding passwords. Another method is to observe network traffic in order to extract user account information. The ability to observe the network traffic exists because the technology was developed so that it would be convenient to use and easily adopted.

As the number of remote hosts that an attacker controlled increased, automated procedures were developed to make their management easier. During this time viruses and self-replicating worms began to include code that would automate further infection and control of other systems. The term “botnet” was coined to define infected host computers and the networks they made up (Barford & Yegneswaran, 2006).

Botnets are believed to be responsible for the majority of the problems with the Internet today. While there are inconsistent estimates among security researchers as to the number of computers around the world that are part of a botnet, recent findings by Johns Hopkins University estimates ranges from 1,000 to 10,000 bots for smaller networks and 1 to 10 million bots for larger networks (McMillan, 2007).

The size and scale of today’s botnets has raised concern among information technology professionals who have implemented security policies in an effort to defend against botnet attacks. These security policies can be understood as attempts by information technology experts to conform technology to the human desire to discriminate between ethical and unethical uses.

Unfortunately, malicious users have become very adept at circumventing traditional defenses such as anti-virus software and firewalls. Even encrypted web transactions may not protect sensitive information if the user’s computer has been infected with malicious software. Since technology is forever changing, information technology managers must stay aware of current security trends and develop clear security policies regarding botnet attacks.

## STATEMENT OF PURPOSE

This paper argues that information technology managers must be aware of the botnet problem so that the necessary resources can be allocated in order to avoid attacks and, in the event an attack occurs, allow a faster recovery. It also argues that information technology managers must be aware of the fundamental issues about how humans relate to technology. One issue is that technology must always draw a compromise between being easily accessible and being secure. Every time a company makes itself more accessible for its customers and partners it also makes itself less secure. Technology cannot discriminate between ethical and unethical uses and so humans are forced to develop security policies that can.

In the future technology may mature to the point where a distinction between ethical and unethical use can be made. Until that day competition will drive businesses to adopt technologies prematurely. This paper will not recommend that businesses wait for technologies to mature before implementing them as these technologies allow businesses to be competitive. Instead, this paper will propose policy suggestions that information technology managers can implement to mitigate the risk involved with implementing technologies prematurely. These policies will act as a bridge until security improves and they are no longer necessary.

## DEFINITION OF TERMS

*Blacklist (Spam).* A list of IP addresses that have been identified as sources of spam email (Malkin & LaQuey Parker, 1993).

*Bot.* The term is derived from the word “robot” and is referred to in this paper as a computer that has been infected with a program that grants control of its resources to someone remotely.

*Botnet.* A network of bot computers (Barford & Yegneswaran, 2006).

*Domain Name System (DNS).* The service that translates and resolves human readable domain names into IP addresses (Malkin & LaQuey Parker, 1993).

*Exploit.* A computer program written to take advantage of a software vulnerability (Barford & Yegneswaran, 2006).

*Fast-Flux.* A DNS technique typically used by botnets to change the DNS servers in order to maintain communication with the botnet operator (Honeynet Project, 2007).

*Firewall.* A network device that separates one network from another and controls what services can communicate between networks.

*Hacker.* An individual who accesses a remote computer without authorization (Malkin & LaQuey Parker, 1993). The terms “hacker”, “attacker”, “criminal”, “cyber criminal” are used interchangeably.

*Internet protocol (IP) address.* A 32-bit set of numbers (in the case of IPv4) used to uniquely identify computers connected to a network or the Internet (Malkin & LaQuey Parker, 1993).

*Internet Relay Chat (IRC)*. An Internet chat protocol that can handle group communication in discussion channels and also person-to-person communication via its private messaging function (Malkin & LaQuey Parker, 1993).

*Malware*. Software with a malicious purpose. This term is used interchangeably with “trojan,” “worm,” “adware,” and “spyware.” (Ianeli & Hackworth, 2005).

*Peer-to-peer*. A method of distributing data and communication across a network without requiring a centralized server (Wang, Sparks, & Zou, 2007).

*Server*. A computer whose function is to provide access to files and other services over a network (Malkin & LaQuey Parker, 1993).

*Spam*. Unsolicited email that is typically sent in bulk (Brodsky & Brodsky, 2007).

*Trojan*. See Malware.

*Virus*. See Malware.

*Web proxy*. See Firewall.

*Worm*. See Malware.

## **THEORETICAL FRAMEWORK**

Information technology is a critical link in the various interrelated activities that are common among large firms and is seen as one of the dimensions for maximizing corporate value creation. This is represented by Porter's value chain model that is used by firms to create a competitive advantage against other businesses.

Information technology is labeled as a support activity on the value chain model. The underlying primary value chain activities all rely on information technology to operate efficiently. Any threats to a firm's information technology sector directly threaten any underlying support activities.

Information technology supports a firm's other value chain activities. The proper operation of a firm's technology is vital to its survival. The botnet problem discussed in this paper targets a firm's technology operations. This results in a compromise of other value chain activities weakening a firm's competitive advantage.

## LIMITATIONS

Humans have always been faced with the problem of making technology either easily accessible or secure. Allowing technology to be easily accessible for all users means that it will be easily accessible for both good and evil users. Conversely, if technology is made secure, then convenience must be sacrificed.

Discussing the depths of how humans use technology is beyond the limits of this paper. It is unknown whether one day humans will resolve the competition between convenience and security. This paper will focus on the botnet problem as a moment in the evolution of technology and how their impact will be felt by businesses worldwide.

## **LITERATURE REVIEW**

This literature review will analyze the current research that has been published about botnets. It will first identify the motivations behind building and operating botnets and how these motivations have evolved over time. Next, it will discuss the functions that a botnet can perform and how these functions impact a business. Then, it will analyze the operating structures of botnets. Finally, it will discuss the current research that exists on how to track and disable botnets.

### **Botnet Motivations**

The prime motivators and abilities of cyber criminals have evolved significantly from the early days of computing. Sharing software and information were once the primary motivators for underground hackers. Computer viruses and worms were simply a step used to gain control over another computer. The authors of this malicious software, also known as malware, took pride in bragging about their accomplishments to others in chat rooms as a way to show off their technical superiority. The impact on victims and organizations was a disruption of service resulting in loss of productivity and occasionally a loss in revenue.

As Internet users began to do their shopping and banking online the nature of malware shifted from disrupting service to exploiting these technologies for financial gain. Malware is now designed to steal sensitive information such as credit card numbers, social security numbers, and passwords and send the information to the botnet owner. The botnet owner, also known as a botmaster, uses the information for further attacks or sells it to other criminals. Other criminals use the information for many nefarious activities including identity theft.

Today, the primary motivation for operating a botnet is the monetary income that can be earned from sending spam email and installing advertising and marketing software on infected computers. Ferris Research (2007) has found that email spam costs businesses over \$100 billion a year worldwide, \$35 billion specifically in the U.S. alone. The main components of this are losses in productivity from users inspecting and deleting spam messages in their email inbox folders, searching for legitimate email messages that were marked as a false positive for spam, and the costs associated with the administrative spam fighting operations.

The second most popular source of income for online criminals is the installation of advertising software, known as adware. According to Webroot Software Inc (2006) the distribution of adware is a \$2 billion per year industry. Many of these software companies, such as TopConverting, Gamma-Cash, and 180solutions, offer monetary incentives for installing their software (Krebs, 2006). Botmasters participate in affiliate programs and receive a small payment each time they install this software on a compromised computer.

Advertising software companies have been criticized for funding botnets. They have been accused of not doing enough to prevent their products from being installed unknowingly by users (Krebs, 2006). To avoid liability adware companies state in their installation agreements that the installation of their software without the consent of the computer's owner is strictly forbidden (Krebs, 2006). They threaten that if these terms are violated they will terminate affiliate programs and remaining account balances will not be paid (Krebs, 2006). To work around this botmasters spread installations over time to avoid suspicion (Krebs, 2006).

### **Botnet Functions**

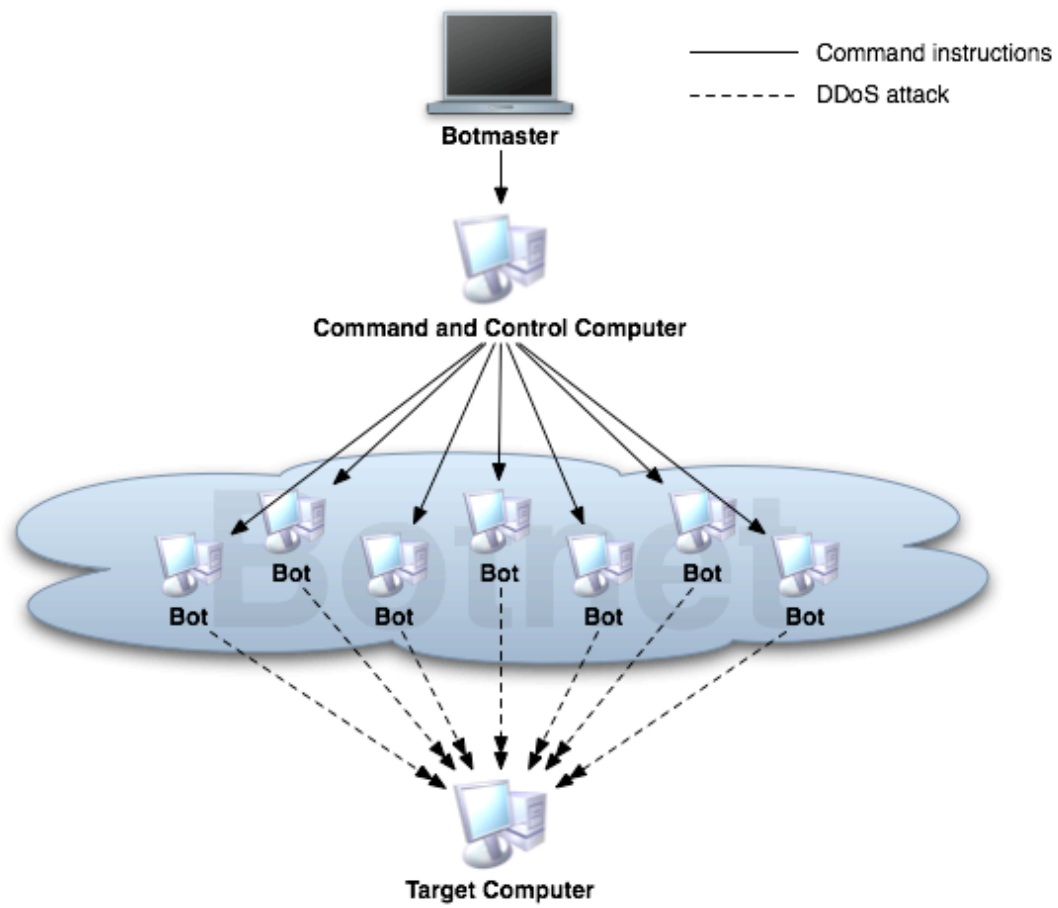
Botnets can perform several functions. We will discuss the three most common botnet uses.

#### *Denial of service attacks.*

The Internet is comprised of finite resources that attackers can exploit. A computer is limited to the number of requests that it can handle and there is a limit to how much information can be transmitted across a network at one time. A criminal can attack a company's technology infrastructure by exhausting these computer and network resources with illegitimate requests. Simon, Agarwal, and Maltz (2007) identified that growing bot networks have given hackers enormous computing and bandwidth power. This power can be used to unleash attacks against a remote host computer or network. Statistically, the most common type of malicious attack is a distributed denial of service (DDoS) attack (Figure 1). This attack method occurs when several malicious client computers saturate a single target computer with requests in an attempt to make its resources unavailable to legitimate client requests (Handley & Rescorla, 2006). These attacks often aim to demonstrate power but are also politically and, more commonly, financially motivated. Botmasters have attempted to extort money from businesses by attacking the business until they pay the botmaster to stop (Ianeli & Hackworth, 2005).

The CSI/FBI Computer Crime and Security Survey found that DDoS attacks cost businesses an average of almost \$3 million in 2006 (CSI/FBI, 2006). This creates a major concern for information technology professionals that are responsible for maintaining the reliability and consistency of their business's information services. If these services are interrupted the resulting losses to a company can be irreversible.

Figure 1. Distributed Denial of Service Attack



Source: Riverhead Networks

*Email spam.*

Email server software is often included in botnet malware because of the profits that spam campaigns generate (Ianeli & Hackworth, 2005). Using several computers to send messages is more effective than a single machine because they are difficult to trace and block once discovered. Sandvine Intelligent Business Networks (2004) has stated that 80% of all spam circulating the Internet is generated by botnets.

Brodsky & Brodsky (2007) identified that the current attempts to detect active spam mailing campaigns are only effective if a large number of messages are sent from a single computer. Once an email server identifies a computer as a source of spam it adds the computer to a blacklist. The blacklist is shared by email servers across the Internet and is updated regularly. This method is ineffective against botnet-generated spam because it is unable to keep up with the increasingly large volume of hosts. By the time email servers receive the updated spam blacklist, the campaign has completed (Brodsky & Brodsky, 2007).

*Data theft.*

Rajab, Zarfoss, Monroe, and Terzis (2006) identified that another major botnet activity is the theft of private information that is stored on company computers. This poses an immediate danger to any business as computer systems contain valuable information about the users or business ventures they support. Even when the existence and value of information is not clear to a system's users, botnet operators know where it is located and how to extract it.

Ianeli and Hackworth (2005) discovered that data theft and information gathering has become big business for botnet operators. Business computers often contain sensitive data or trade secrets. Even information that is not stored on the computer's internal storage is subject to theft as transmitted data can be captured with a keyboard logger or network traffic analyzer. This

data includes email passwords and vital financial information. The email address book, credit card numbers, and other critical data can be used for extortion or sold for profit.

The CSI/FBI Computer Crime and Security Survey found that on average the theft of proprietary information and unauthorized data access cost U.S. businesses over \$17 million in 2006 (CSI/FBI, 2006). This data includes credit card numbers, social security numbers, and passwords to online web sites. This information holds a high value and is often sold or used in future computer crimes (Ianeli & Hackworth, 2005).

### **Botnet Operating Structures**

Computer security researchers have classified botnet operations into two categories: the initial infection and the command and control mechanism. The initial infection is the step where the computer is first compromised with malware. The command and control mechanism joins the infected computer to the botnet and the botmaster can begin issuing operating commands.

#### *Initial Infection*

Rajab et al. (2006) found that botnet operators use the same techniques to compromise and gain control of remote computers that were used by authors of email viruses and self-replicating worms. These methods are mentioned below. A botnet operator has an incentive to control as many computers as possible. They must constantly collect and infect new computers to maintain and grow the botnet's effectiveness. Computer users, network operators, and ISPs regularly discover and disconnect infected computers forcing a continuum.

Ianeli & Hackworth (2005) identified that the infection process is simple and requires only minimal technical skill. Information on how to compromise and infect computers is shared among underground communities in IRC chat rooms. Additionally, there are many online

resources that document the inner workings of the protocols and systems. This makes it simple for a novice attacker to learn the tools and techniques necessary to build a functioning botnet.

Ianeli & Hackworth (2005) explained that all infection methods share a similar process in which a computer downloads and executes the malware. The only difference among them is the method used to trigger the malware download. There are several ways an infection can be initiated, but the most common will be discussed:

*Deception.* Ianeli & Hackworth (2005) identified that this method requires user interaction. This is achieved by convincing the user to download and execute a file under the assumption that it is safe. This is sometimes referred to as social engineering. The deception method is commonly used in spam email and browser plugin installation requests.

*Vulnerability exploitation.* In this method Ianeli & Hackworth (2005) identified that a computer becomes infected when a security vulnerability in the existing software is exploited. This triggers the download and installation of malware without the user's knowledge or consent. The computer must first be scanned to determine if a security vulnerability exists and if that vulnerability is exploitable. Bots regularly scan to find other vulnerable computers on their network. If a vulnerable computer is found, then an attempt will be made to automatically exploit and infect it.

*Malicious web servers.* A malicious web server is a web server that hosts a web site that contains malware. This scenario is becoming a more common method of infection and is a combination of the previous two techniques. When a user visits a malicious web site the malware will attempt to exploit security vulnerabilities in their web browser (Figure 2). If the exploit attempt is successful, then the malware on the web server will trigger a download of the botnet

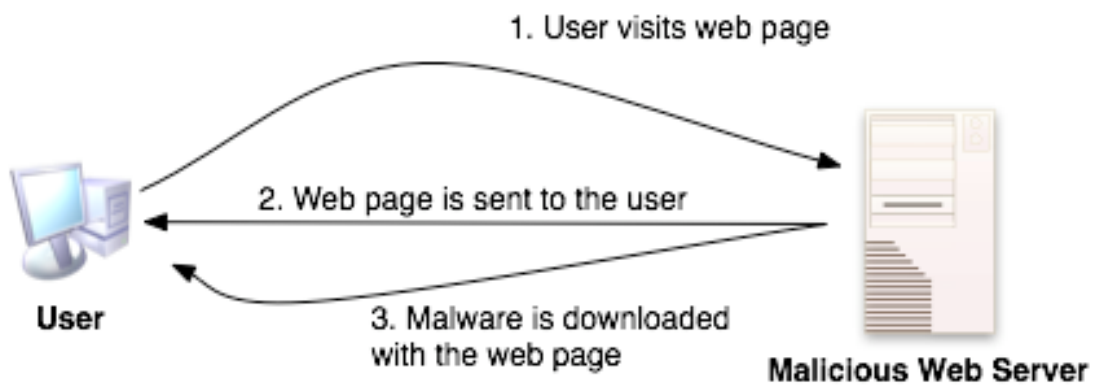
malware. This is also referred to as drive-by downloading because, unlike the deception method, the installation is done quietly in the background without the user's knowledge or consent.

Provos, McNamee, Mavrommatis, Wang, & Modadugu (2007) have identified malicious web servers as far more dangerous than other infection techniques. The team used a honeypot computer that would crawl across the Internet, visiting thousands of web sites. They would then analyze the honeypot to see which websites were able to modify its state.

Unlike traditional infection techniques that use a push-based approach and require an external trigger, web-based infections follow a pull-based model to increase a botnet's population. In a push-based infection a remote attacker must have direct access to a computer in order to gain control over it. In a pull-based infection a client makes the initial request to a public web site. This renders network firewalls and web proxies ineffective.

It is becoming more difficult to avoid drive-by downloads. In the past, these malicious web pages were only found on underground web sites like gambling or adult-oriented sites. However, using the same techniques as Provos et al, Trend Micro (2008) has discovered malware bundled in advertisements on several popular news, sports, and social networking web sites. Provos, Mavrommatis, Abu-Rajab, and Monroe (2008) found that this was accomplished by embedding hidden frames in web pages. These frames contained instructions to initiate a drive-by download when a user visited the web page.

Figure 2. Drive-by Download.



Source: The HoneyNet Project

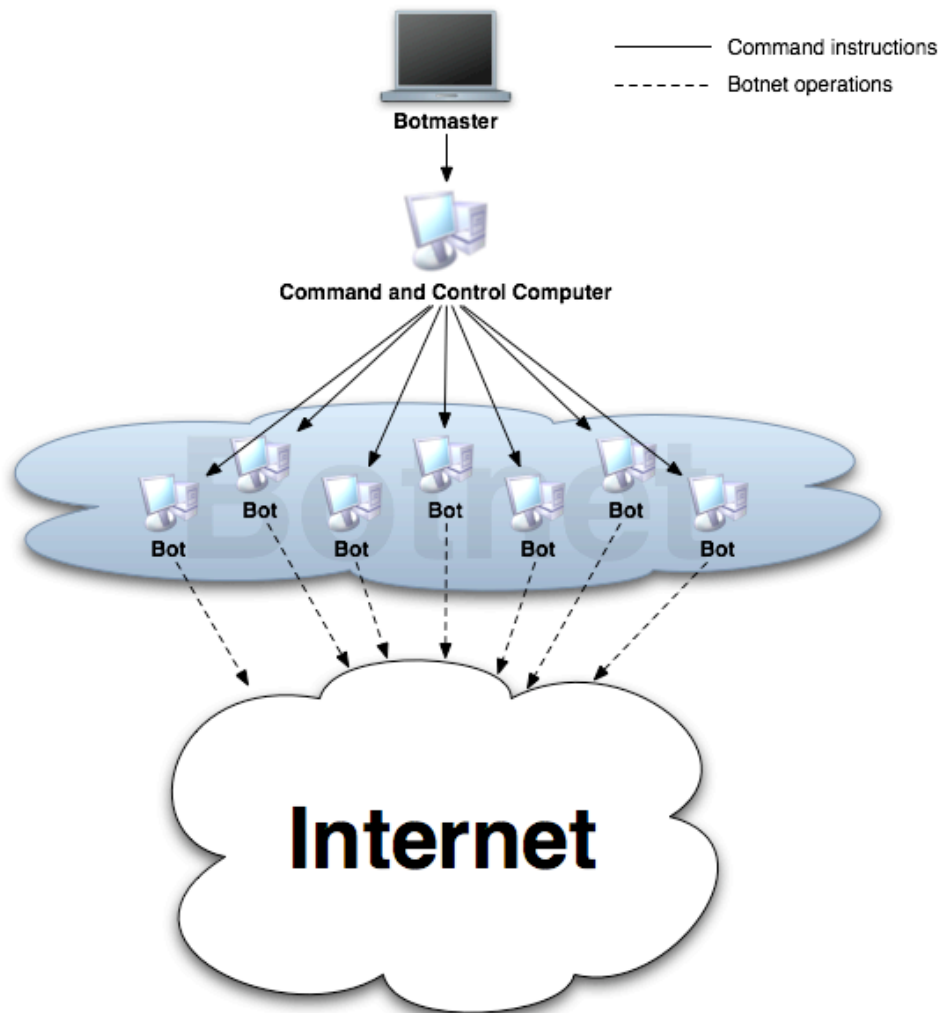
### *Command and Control (C&C) Mechanisms*

The abilities and characteristics of a botnet are dependant on their communication structure. A mechanism must exist for a botmaster to control and communicate with their bots. This is accomplished with the use of a command and control (C&C) server, which all compromised computers connect to in order to receive commands (Figure 3). Thanks to exhaustive research of the inner workings of botnets information technology professionals now have a better understanding of their operating procedures.

*Internet Relay Chat (IRC)*. IRC has existed since the early 1990s and was designed as a flexible protocol that allowed multiple forms of communication, including server-client and peer-to-peer (Rajab et al., 2006). Analysis by Goebel and Holz (2007) found botnet operators commonly choose IRC-based C&C communication architecture when operating a botnet. They believe this is due to the protocol's simplicity and robustness.

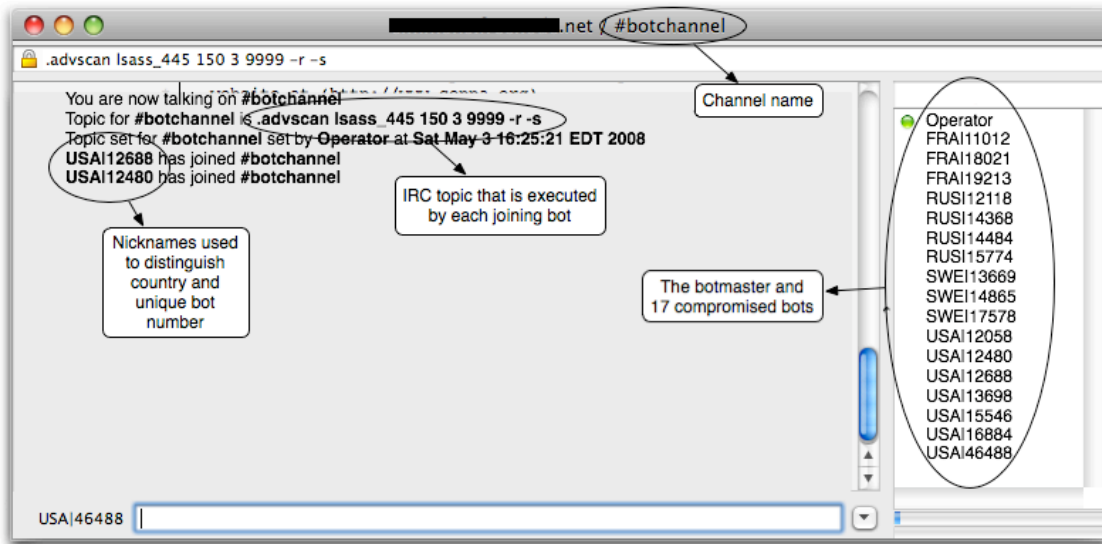
Rajab et al., (2006) studied the communication process and found the C&C structure to be simplistic. The Clickbot.A and Phatbot malware binaries both used IRC as their C&C mechanism (Daswani & Stoppelman, 2007; Stewart, 2004). Immediately after malware infection the bot connects to a waiting IRC server and joins a predetermined chat channel. The channel's topic is a command that the bot executes and then sits idle waiting for further instructions from the botmaster (Figure 4).

Figure 3. A Client-to-Server Command and Control Mechanism.



Source: The HoneyNet Project

Figure 4. IRC Command and Control



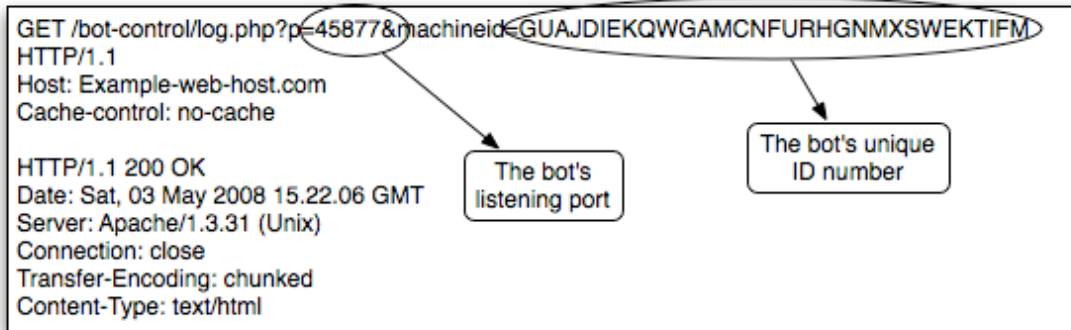
Source: Ianeli & Hackworth

Ianeli & Hackworth (2005) identified that a bot's IRC nickname is a source of information about the compromised computer. The botmaster uses this information to index and organize the bots when coordinating attacks. The Phatbot malware formulated the nickname using the host country name and a unique identification number (Stewart, 2004). The bot stays connected forever until the botmaster, or an administrator that has discovered it, shuts it down.

*Web-based.* Ianeli and Hackworth (2005) discussed how botmasters have begun to use the HTTP protocol as a means of controlling botnets. The main difference between the ways in which a web-based C&C mechanism operates from the IRC model is that instead of command instructions being pushed to the bot from the operator in IRC, the bot downloads and executes a file from the web server that contains commands. Botmasters are beginning to prefer this method because free web space is easily available. Web-based C&C also requires very little setup and configuration time. A computer that is infected with this type of malware first sends its information to the server by querying a web-based script. The query string that is sent to the script contains basic information about the bot (Figure 5). This helps the botnet operator keep an accurate index of the machines in their control. This is illustrated in figure 5 below. The highlighted variable represents the port that the compromised client is listening on.

Once the bot has identified itself to the server it will then request to download a small file with commands to execute. This request is repeated infinitely at 5-second intervals (Ianeli & Hackworth, 2005). The bot stays connected forever until the botmaster, or an administrator that has discovered it, shuts it down.

Figure 5. Web-based Command and Control Connection Log

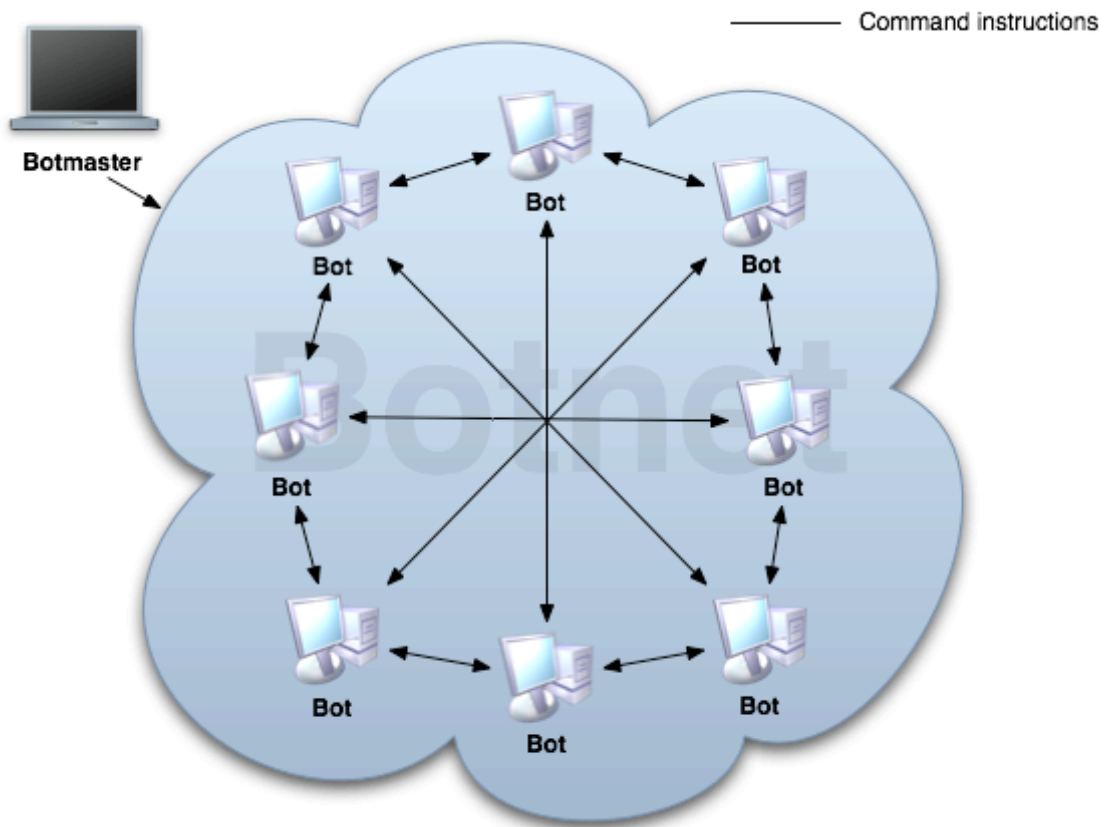


Source: Ianeli & Hackworth

*Fast-Flux DNS.* Maintaining a connection between the botmaster and their bots is crucial to botnet operations. Bots have the DNS address of the C&C server built in. DNS is a service that translates domain names to IP addresses because people can more easily remember names rather than numbers. More importantly, DNS allows bot owners to change the location of the C&C server if it is discovered and shut down. While this is helpful, it is not ideal. Changes to DNS records take several hours to update across the Internet. The HoneyNet Project (2007) has identified a new practice that botnet owners have begun to implement to solve this problem. They are using a service called fast-flux DNS. Fast-flux is a DNS strategy that allows a server's location on the Internet to change rapidly based on the domain name it uses. Fast-flux makes it difficult for administrators to track and disable botnets. This practice has not yet seen wide adoption.

*Peer-to-peer.* A client-to-server connection is very simple and easy to implement. However, it does have a few major drawbacks. The first drawback is that it does not scale well with a large number of clients. The second drawback is that the server is the single failure point for the network. A peer-to-peer network decentralizes the responsibilities of a server to all the peers that are connected in the network. Each peer performs the functions of both a server that serves data and a client that receives data (Figure 6).

Figure 6. Peer-to-Peer Botnet



Source: Wang et al.

Botnets have recently begun to implement peer-to-peer (P2P) networking capabilities in order to decentralize from a central C&C server making the botnet more difficult to disable. Wang, Sparks, & Zou (2007) discussed that the effort to adopt P2P technologies has seen mixed results. Their work mentions attempts (e.g. Phatbot, Slapper, and Storm) to recycle code from WASTE, an open source P2P protocol. In his analysis of the Phatbot trojan, Stewart (2004) noted that the failed adoption thus far has stemmed from WASTE's inability to scale well across large networks.

The challenge botmasters have faced in implementing P2P protocols has been to find an effective indexing method that securely and efficiently monitors and controls the botnet. Perriot & Szor (2003) discovered that the developer of the Slapper worm stored the complete botnet list on each bot. Not only did this generate a large amount of network traffic as the bot list constantly updated but it granted anyone who discovered a bot access to the full list. This made it easy to disable the botnet or for another attacker to steal it.

### *Defending The Botnet*

Bolliger and Kaufmann (2004) found that some botnets have implemented a layer of security as a defensive measure to prevent other attackers from stealing bots and merging them with their botnet. In these implementations some authentication takes place prior to establishing a connection between the bot and the IRC server. This is done to verify the identity of the bot, IRC server, and botmaster. First, the bot will authenticate itself with the IRC server using a password. The password is built into the bot and is sent to the server openly, without any encryption (Rajab et al., 2006). Second, the bot will need to authenticate itself with the specific chat channel on the IRC server using another password. Finally, the last step of authentication occurs when the botmaster attempts to obtain control over each bot (Rajab et al., 2006).

### Tracking Botnets

As the botnet problem escalates computer security experts have begun to develop ways to not only dismantle them but to also monitor them in order to gather intelligence that might prove useful in future research. The main benefit of tracking botnet activity is that it allows computer security researchers a direct observation of malicious Internet activity. Also, these observations give a researcher insight into the attackers that create botnets, their profiles and motivations.

Detecting malicious activity on a network is difficult. The attacker can hide their presence on a machine and only become active under certain conditions. In the past, botnet tracking efforts have focused on actively monitoring a sample group of IRC-based botnets in limited numbers. Nazario (2004) discussed the limitations of this approach both for IRC-based botnets and botnets based on HTTP that have recently shown a growing prevalence.

Antivirus companies Symantec (2008) and Trend Micro (2008) as well as Google (2007) have also taken an interest in tracking botnet malware to discover new techniques that might pose a danger to their customers and users. Some vendors publish their findings but this information is not always enough to effectively track botnets.

Botnet tracking is often complementary to honeypot use where new malware samples, including bots and Trojans, are collected and analyzed. The HoneyNet Project has published a few papers that discuss their techniques in tracking botnets that are IRC based and implement malicious web servers as an infection vector. In 2005 the HoneyNet Project (2005) implemented a honeypot that was made up of an unpatched version of Windows XP and attached to a dial-in network of a German ISP. On average, the honeypot would take less than 10 minutes to be exploited by an automated attack. The shortest time took only a few seconds and occurred when

the SDBot malware compromised the honeypot shortly after it was plugged into the network (Honeynet Project, 2005).

After a honeypot was compromised the Honeynet team would disconnect the honeypot and analyze it to record the requests to join IRC servers that the installed malware sent out. This data was then used to reconnect to the botnet using a standalone client. This allowed researchers to manually send commands and record the responses from the IRC server. One problem that was outlined with this approach is that some botnets used modified IRC servers that a normal client could not connect to.

In order to observe communication between the bot and the custom IRC server the Honeynet wrote their own custom IRC client called a “drone.” An IRC drone is designed with the intelligence to mimic a real bot so that it will be included by the botmaster and take part in their operations. It uses a template scheme to advertise their hardware as being similar to a typical desktop computer (Rajab et al., 2006). Drones also respond to commands in a way to mimic their emulated state since a bot will respond to a command based on its current state. For example, the server’s response will be different if it is currently executing a scanning operation than it would be if it were sitting idle. They also have the ability to filter unnecessary data that is collected and enough to not be suspected and ignored by botmasters (Rajab et al., 2006).

After a botnet is discovered and any informative data is gathered it is shut down. Disconnecting the C&C server and disrupting communication between the botmaster and the bots accomplishes this. The use of fast-flux domain names that change the C&C server’s location rapidly and peer-to-peer communication makes shutting down botnets more difficult.

### **Conclusion**

The financial motivations published in previous research show botnets as a growth industry. Researchers expect botnet attacks to have dramatic future implications for global businesses. The tension that exists between developing technology that is easily accessible for all users and developing technology that is secure will have to relax. Research has shown a movement to resolve this tension and create technology that is both secure and convenient to use.

## PROCEDURES

This paper will present case studies of three different real world malicious attacks that target a production server. Each case will perform a qualitative analysis from a sample of normal activity collected from the server. The server runs the CentOS 4 Linux operating system on medium powered hardware. It provides web site hosting, email, and file storage services for ten users. The server is publicly accessible and operates on a high-speed academic network. It runs the latest version of the Apache<sup>1</sup> web server software with the default plugins and modules.

Two administrators are responsible for managing the system. They connect to the machine remotely via a secure shell protocol called SSH.<sup>2</sup> SSH allows an administrator to securely manage a server from a remote location without needing physical access. The system's users are able to upload and manage files for their websites using a secure file transfer method called SFTP.<sup>3</sup> SFTP allows users to securely transfer files to and from a remote server without requiring physical access.<sup>2</sup> Finally, the server uses a software-based firewall that allows incoming connections access to specific, pre-determined services and blocks all other connection attempts.

Several of the Unix command line utilities were used to analyze the data collected in the cases that follow. The Nmap<sup>4</sup> network exploration and security auditing utility was used to gather information about specific hosts. Additionally, Microsoft Excel was used to create visual charts and Mac OS X's built in screen capture tool was used to capture application snap shots.

## RESULTS

### Case 1 – Web Site Comment Spam

A CentOS 4 Linux web server is located on an academic network. It serves several web sites, one of which is a fan site for a music group. The site provides visitors with content about the band and contains a message board. The message board requires users to register before they can post comments.

The registration process contains a step that requires the new user to complete a challenge-response test. To complete the test a user must read and correctly enter an alphanumeric code displayed on the screen. The code is displayed as a picture that non-humans, specifically bots, cannot decipher. This prevents accounts from being created via automated processes, including those issued by a botnet.

After the user completes and submits the registration form the web site will send the user an email containing their account information. The new user account remains inactive until the message board operator authorizes it.

One morning the server administrator was checking the server logs and noticed that several new user registration email messages had been rejected by their email providers. Worried that the server may have been placed on an email spam blacklist he investigated the situation further. He found that the fan site's normally low volume message board had recently had several hundred new users sign up. This activity was not normal for the web site and the administrator found it suspicious.

Figure 7. Message Board User Registration Form

Register

http://.../forums/profile.php?mode=register&agreed

Register Search

Dredg Online Forum Index » Register

The time now is Wed Oct 22, 2008 4:17 pm

Registration Information

Items marked with a \* are required unless stated otherwise.

Username: \*

E-mail address: \*

Password: \*

Confirm password: \*

If you are visually impaired or cannot otherwise read this code please contact the [Administrator](#) for help.

gPUG8v

Confirmation code: \*  
Enter the code exactly as you see it.  
The code is case sensitive and zero has a diagonal line through it.

Profile Information

This information will be publicly viewable

ICQ Number:

AIM Address:

MSN Messenger:

A sudden burst of users is not necessarily a sign of an attack. It is common for a web site's message board to experience a sudden burst of visitors and new user sign ups if a popular website, such as Fark.com or Digg.com, refers its visitors there by placing a link on its site. Sites like Fark.com and Digg.com allow visitors to rank the popularity of the links they display. If a link becomes very popular then the linked site will receive a large number of new visitors and user account sign ups. This activity is similar to activity observed during botnet spam campaigns.

An administrator can observe where visitors have originated from and determine how they found the site by checking the web server log. The log contains the information about the web page request including the referring site. If a visitor is legitimate and came from a popular site, such as Fark.com or Digg.com, the site's address would appear in the log. The example in Figure 8 demonstrates a web page visitor that was referred to the site from a search on Yahoo.com.

In this case the log entries associated with the user signups did not list any referring web sites such as Fark.com or Digg.com (Figure 9). A web page request does not contain a referrer when a visitor types the web site address into their browser instead of clicking a link from another site. Since it is unlikely that several hundred visitors each knew the address of the message board web page the administrator believed that legitimate users did not create the new accounts. Instead, he believed they were created using an automated process. The person who executed the process likely obtained the message board's address by searching Google or other search engines for a list of websites running the message board software.

Figure 8. Legitimate Web Page Request

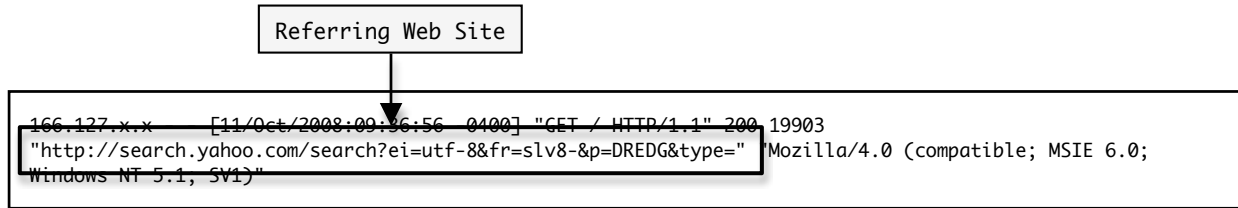
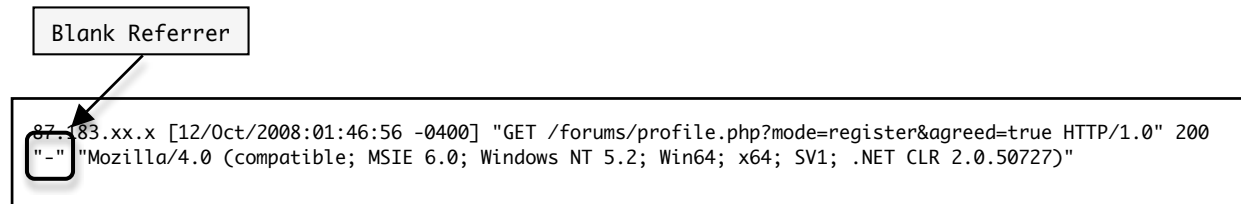


Figure 9. Suspicious Web Page Request



The user accounts appeared to have been created by an automated method from many different hosts. The process was able to successfully complete the image verification challenge-response test during the registration but was frustrated by the web site operator's manual activation process. The registration's challenge-response test that was designed to prevent automated methods from creating user accounts had failed to prevent the attack. The function was out of date and defeated by optical character recognition technology that had developed the ability to decipher the verification code.

The administrator determined that many of the accounts had been created with the intent to post spam material. Several of the account profiles contained affiliate links to adult and prescription drug websites. Figure 10 shows a screen capture of some of the users in the message board's user account database table.

The administrator browsed through his web server logs to gather more information. The server was configured to log all web page requests. He found that for each attempt to create a user account there were three log entries (Figure 11). The log entries showed whoever or whatever created the account made three separate connections. The first two connections were to establish a session with the message board software. The third and final connection was used to submit the user account registration.

Once the administrator understood the pattern used to create the accounts he parsed the log files and calculated the number of remote hosts involved and how many accounts they each created (Figure 12). Since the message board was low-volume, he assumed that any inactive account was malicious. He found that over the past ten days there had been 380 accounts created from 208 unique IP addresses. Also, several of the malicious hosts had attempted to log into their inactive accounts (Figure 13).

Figure 10. Spam User Accounts

```

root@ ~ -- ssh -- 91
mysql> SELECT username,user_regdate,user_email,user_website,user_from FROM forums1.users WHERE user_active = '0' AND user_website != '' LIMIT 25;
+-----+-----+-----+-----+-----+
| username      | user_regdate | user_email      | user_website      | user_from      |
+-----+-----+-----+-----+-----+
| MypeHogeAneno | 1224486816  | terulopametar@gmail.com | http://sexvideo.free-site-host.com/map.html | Trinidad and Tobago |
| Sitchiltbaiff | 1224465628  | nasulllopcip@gmail.com  | http://neither-field.com/content/projects/tbe/1/index.html | Angola |
| futballsd     | 1224467825  | futbooll@evdo1x.ru      | http://fut-ball.ru/ | Poccия |
| inhitteWeesiago | 1224469219  | disc@iballit-pharmacy.com | http://abollit-discounts-meds.com | levitra walwart pharmacy |
| AnnelPtoott   | 1224470381  | lypedvedanielieir@gmail.com | http://groups.google.com.ua/group/nikitatester/web | United States |
| Friednerrn    | 1224472413  | pleasyday@gawab.com     | http://uwsaeer.5x.ta/map.html | Iraq |
| BuyViagraGeneric | 1224478711  | kikoceax@gmail.com      | http://www.buyviagra.com | Namibia |
| Jahn          | 1224481746  | johntt@gawab.com        | http://www.baidu.com | US |
| A0Dawayspomona | 1224487287  | bcVeyGotVoire@2008blogger.com | http://2008octobersbest8.net | Taiwan |
| evocixavaxy   | 1224490214  | gangenuathe@gmail.com   | http://links.imeem.com/23RxEVHbU | Venezuela |
| detroit-marathon | 1224495374  | plutarchbpierre@gmail.com | http://detroit-marathon.977nb.com/encyclopedia-of-life.html | Spain |
| abinkinalsy   | 1224496538  | asdf3s22@mail.ru       | http://bbs.keyhole.com/ubb/showprofile.php?Cat=0 | Angola |
| Neonetwy      | 1224496661  | pcarl2085@gmail.com     | http://www.google.com | UK |
| Antalssname   | 1223869537  | aqgaffgagdf@gawab.com   | http://www.teen-angel.nm.ru | Poccия |
| symnervesee   | 1223878788  | pantinica@gmail.com     | http://medicamentos-generico.com/comprar_viagra.html | Spain |
| Arcammbianna  | 1223871850  | wepffsoolley@gmail.com  | http://google.com | Bahrain |
| JexBeryCrexarthy | 1223878672  | albertser@1.ua          | http://hi5.com/friend/profile/displayJournal.do?viewself=true | Eritrea |
| Creriliabemia | 1223883851  | uevyowpeortr@gmail.com  | http://diortradio.edu | Estonia |
| Arennaceengirl | 1223885971  | link@rusasporno.com     | http://rusasporno.com | USA |
| kath-and-kia   | 1223886875  | kathandkim909@gmail.com | http://kath-and-kim.freehostia.com/fantasti-cc.html | Spain |
| mik247        | 1223892550  | anjelinamail@gmail.com  | http://vaocilicvmmk.100webspace.com/ | Bahrain |
| ViagraBUY     | 1223901801  | topshop@besthotels4u.info | http://www.cratekings.com/forum/members/buyingviagra.html | VIAGRA ONLINE NO PRESCRIPTION NEEDED |
| VladimirIIIjr | 1223905237  | vl.a.d.oir.i.ii.jr@gmail.com | http://vtraden1.ru/ | Poccия |
| richard-garriott | 1223909891  | richardgarriott9@gmail.com | http://richard-garriott.freehostia.com/olga-kurylenko.html | Italy |
| BertypeLaurce | 1223918684  | wizwhetty@gmail.com     | http://nemeaianonch.812webpages.com | Croatia |
+-----+-----+-----+-----+-----+
25 rows in set (0.00 sec)

mysql>

```

Figure 11. Web Server Log Entries for a Malicious User Account

```
1 87.183.86.xx1 [12/Oct/2008:01:46:56 -0400] "GET /forums/profile.php?mode=register&agreed=true HTTP/1.0"
200 "-"
2 87.183.86.xx1 [12/Oct/2008:01:46:57 -0400] "GET
/forums/profile.php?mode=confirm&id=6b82330fada18859f9db8fd9ae6eebd1&sid=acc22f472dec3cd89fe0817cd913166f
HTTP/1.0" 200 "-"
3 87.183.86.xx1 [12/Oct/2008:01:46:58 -0400] "POST /forums/profile.php?sid=acc22f472dec3cd89fe0817cd913166f
HTTP/1.0" 200 "http://www.*****.com/forums/profile.php?mode=register&agreed=true"
```

Figure 12. Remote Hosts Attempting to Create Spam User Accounts

```
[root@bu**** ~]# grep "POST /forums/profile.php" access_log | cut -d " " -f 1 | sort | uniq -c \  
| awk '{ print $2 "\t" $1 " times" }'  
78.107.221.xxx 1 times  
78.110.175.xxx 10 times  
78.157.142.xxx 4 times  
78.157.143.xxx 4 times  
78.157.143.xxx 1 times  
78.157.143.xxx 10 times  
...
```

Figure 13. Remote Hosts Attempting to Login

```
[root@bu**** ~]# for i in `grep "POST /forums/profile.php" access_log | cut -d " " -f 1`; do \
  grep $i access_log | grep "POST /forums/login.php" | cut -d " " -f 1 | uniq -c \
  echo \
done | sort | uniq -c | awk '{ print $3 "\t" $2 " login attempts" }' | sort -g
61.7.223.xxx      6 login attempts
77.241.45.xxx    1 login attempts
78.157.143.xxx   19 login attempts
78.26.179.xxx    1 login attempts
79.143.176.xxx   27 login attempts
85.141.234.xxx   3 login attempts
85.17.111.xxx    1 login attempts
85.17.167.xxx    63 login attempts
...
```

The administrator now had a list of offending hosts and could determine the appropriate response. He removed the malicious user accounts from the message board's database. Then, he used the server's firewall to block the offending hosts from making any further incoming connections. (Figure 14).

This campaign took advantage of security weakness in the message board software's image verification code system. The administrator requested that the web site operator correct this flaw by installing an up-to-date challenge-response human verification plugin. This halted further spam accounts from being created.

The attack shown in this case is a clear-cut example of a coordinated botnet attack against a networked server. It is also an example of what a small botnet can accomplish. Had this been a higher traffic web site the attack would have been more difficult to detect because it used few resources and followed the same pattern as legitimate usage.

Figure 14. Firewall Rules to Block the Malicious Hosts

```
[root@bu**** ~]# grep "POST /forums/profile.php" access_log | cut -d " " -f 1 | sort | uniq | \  
  awk '{ print "ssiptables -A INPUT -s " $1 " -j DROP" }'  
iptables -A INPUT -s 194.165.42.xx7 -j DROP  
iptables -A INPUT -s 194.165.42.xx5 -j DROP  
iptables -A INPUT -s 194.165.42.xx8 -j DROP  
iptables -A INPUT -s 194.165.42.xx4 -j DROP  
iptables -A INPUT -s 194.165.42.xx9 -j DROP  
...
```

## Case 2 – Suspicious Browser User Agent Strings

While observing his Apache web server logs an administrator notices some unusual web browser user agent strings. These strings identify a visitor's web browser software and are received and recorded by the web server when the visitor makes a web page request. User agent strings can easily be falsified so finding unusual strings is not that unusual.

The administrator analyzed the user agent strings being used to determine patterns from the visitor's traffic (Figure 15). A pattern showed the majority of visitor's browsers were outdated versions of Internet Explorer.

Parsing the web server log file to gathered the host IP addresses of the requests containing suspicious browser user agent strings (Figure 16).

To gather some basic information about the activity on his network, the administrator scanned several hosts looking for those running public web servers (Figure 17). The choice to scan for only web services was made to quickly locate servers with publicly accessible web sites. Several of the servers had web services running but were not correctly configured.

One of the correctly configured hosts was a Windows server located in New York. The server was running a default installation of XAMPP, an Apache distribution that is preconfigured to work with popular web site software. The web server statistics were publicly available for viewing. They contained a list of the most requested files during the past week (Figure 18).

The server log statistics showed the second most requested file in the past week was a large text file named "linka.txt". The file contained 2416 unique IP addresses with a seemingly random port (Figure 19). 26 of the IPs in the list appeared in the administrator's web server logs during the past 30 days.

Figure 15. User Browser String Patterns

```
[root@bu**** ~]# grep "POST /forums/profile.php" access_log | cut -d " " -f 12-40 | sort \  
| uniq -c | sort -gr  
6 "Mozilla/4.0 (compatible; MSIE 5.5; Windows 98)"  
6 "Mozilla/4.0 (compatible; MSIE 5.0; Windows 2000) Opera 6.0 [en]"  
6 "Mozilla/1.22 (compatible; MSIE 2.0; Windows 95)"  
5 "Mozilla/4.7 (compatible; OffByOne; Windows 2000) Webster Pro V3.4"  
5 "Mozilla/4.0 (compatible; MSIE 6.0; Windows 98; Win 9x 4.90)"  
...
```

Figure 16. List of Hosts Using Out Dated Browsers

```
[root@bu**** ~]# grep "Mozilla/4.0 (compatible; MSIE 6.0; Windows 98; Win 9x 4.90)" access_log > list.txt
```

Figure 17. Hosts Running Web Servers

```
[root@bu**** ~]# for i in `list.txt`; do nmap -p 80 $i ; done
Interesting ports on 58.211.75.xxx:
PORT      STATE SERVICE
80/tcp    open  http

Interesting ports on 58.57.60.xxx:
PORT      STATE SERVICE
80/tcp    open  http

Interesting ports on 60.10.134.xxx:
PORT      STATE SERVICE
80/tcp    closed http

Interesting ports on 61.178.185.xxx:
PORT      STATE SERVICE
80/tcp    open  http

Interesting ports on 78.136.117.xxx:
PORT      STATE SERVICE
80/tcp    open  http
...
```

Figure 18. XAMPP Web Page File Request Statistics

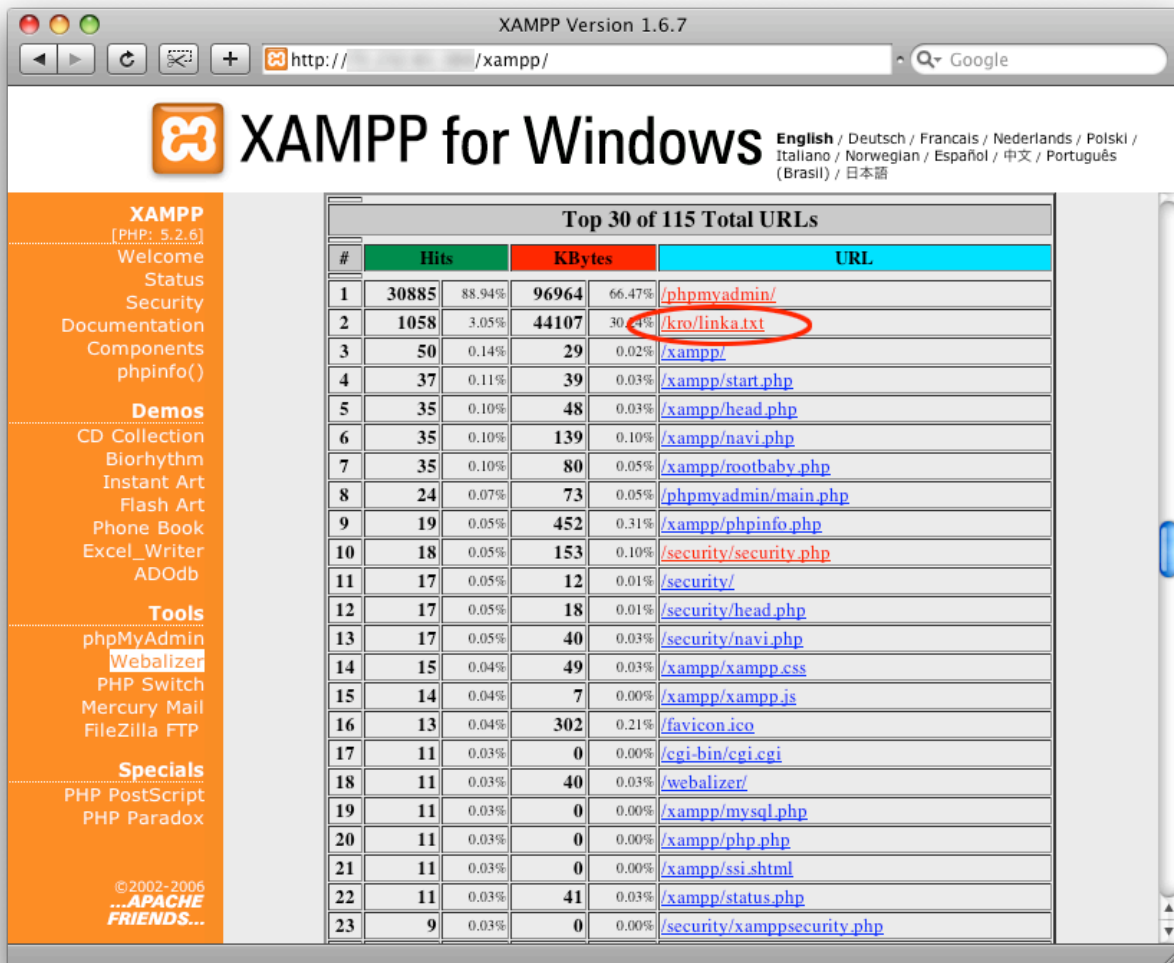
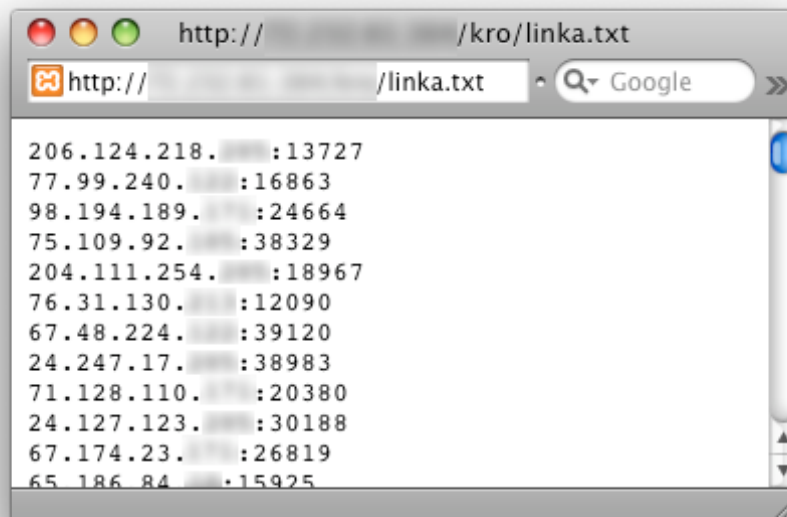


Figure 19. Contents of linka.txt



The administrator was unable to determine the purpose of the IP address list. It followed patterns similar to lists of infected bots but there was no further evidence to support the suspicion. The file could possibly have been a list of proxy servers as some of the ports were common listening ports for the SOCKS proxy service. A full port scan against the server showed no suspicious listening ports, including proxy services. At this point the administrator stopped his investigation to avoid possible illegal intrusion issues and alerted the server's hosting provider of his findings.

The activity shown in this case is an example of a grey area in security analysis. The findings were similar to patterns found in lists used by botnets to index bots however a firm conclusion about the activity could not be drawn. The list was likely evidence of some form of a distributed attack among many computers but without more information the administrator could not make a firm conclusion.

### **Case 3 – Suspicious Game Server Activity**

A server running the CentOS 4.7 Linux operating system on an academic network has firewall software that is configured to only allow incoming connections on a few specified ports and drop connections from all others. The administrator noticed that every day the server's log reports showed several dropped incoming connections from many different source IP addresses. The connection attempts were to the same ports: UDP 12203 and 12204. He recognized these ports as the listening ports for a multiplayer game that the server had hosted in the past. The administrator found it suspicious that multiple clients were attempting to connect even though the game had been turned off several months ago.

The administrator logged onto the machine to investigate the activity. The server's firewall software recorded each blocked connection attempt in a log file (Figure 20). A quick script was written to calculate the number of connection attempts (Figure 21).

The logs showed exactly 6300 connection attempts over the past 47. The connections had originated from 351 different remote hosts. Opening a listening connection to record the information sent during the connection attempts does not provide definitive information about the intention of the connections (Figure 22).

The administrator was unable to determine the intention of the connection attempts. They could have been legitimate connections by game clients searching for an active server based on an outdated game server list. The connection attempts could also have been illegitimate connections by a custom program attempting to find an active server and exploit it.

This case is another example of a grey area in security analysis. However, unlike the previous case the activity appeared to be legitimate. This made it difficult for the administrator to determine the intention of the activity and conclude whether it was normal or malicious.

Figure 20. Example of Dropped Connection Attempt

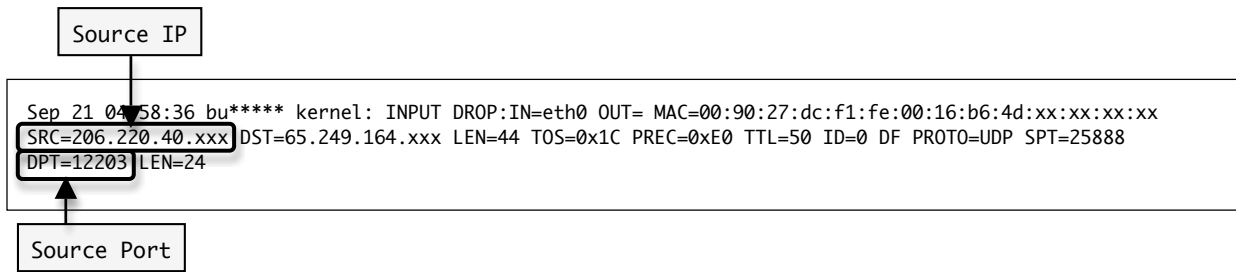


Figure 21. Connection Attempts to UDP Port 12203 and 12204

```
[root@bu**** ~]# grep -e 'DPT=1220[3|4]' messages.log | grep "PROTO=UDP" | awk '{ gsub("SRC=", "", $10);  
print $10 }' | sort | uniq -c | sort -gr  
...  
1 201.132.144.xx2  
1 201.132.144.xx0  
1 201.132.143.xx6  
1 201.132.143.xx1  
...
```

Figure 22. Record Connection Request

```
[root@bu**** ~]# nc -u -l 12203  
???getstatus
```

## DISCUSSION

The case studies in this paper are examples of technological attacks that target a firm's information technology infrastructure. Porter's value chain model labels information technology as a support activity. Since the underlying primary value chain activities all rely on information technology to operate effectively, the attacks in these cases are threats against a firm's underlying support activities and ultimately the entire firm. It is critical for information technology professionals to make sure the technology infrastructure and the containing services are available and are reliable.

Providing the best information technology services requires policies that include methods for recognizing illegitimate activity. Security policies will help staff recognize and react to an attack and perform the necessary steps to recover if necessary. As demonstrated in the previous cases, the problem that information technology professionals experience is trying to determine whether network traffic is legitimate or illegitimate. If network traffic is found to be illegitimate then the appropriate action can be taken. Unfortunately, there are so many different types of network attacks that firm policies are difficult to set. Instead, information technology professionals and administrators need to develop policies that administer security in a casuistic manner.

### **Recognizing Malicious Network Activity**

The first step of a security policy should provide details about how to determine if network activity is suspicious. To make this determination staff must be able to notice abnormal patterns in system usage. First, a baseline of normal network activity must be established. Performing regular log audits and monitoring bandwidth usage can accomplish this. Taking time

to familiarize with network and server log characteristics will give staff the ability to know which services use which resources under normal operating conditions. Observing deviations in network traffic and unique entries in system logs will highlight abnormal activity.

When an administrator observes abnormal activity they will then need to make a determination of whether the abnormal activity is malicious. To do this they must be educated on the types of attacks that can occur. This knowledge can be obtained by following security publications that document malicious Internet activity.

Knowledge can also be obtained by running in-house data collection utilities. Deploying a honeypot is an effective data collection method. A honeypot is a stand-alone network service that mimics a real service to attract activity. The honeypot is not part of production and does not provide any legitimate services so any activity that it records is assumed to be malicious. As an aside, caution should be taken when implementing a honeypot. Some researchers have found that if a botnet operator notices that their networks are being probed some may turn against you, often with a DDoS attack. Running a honeypot may not be feasible for some administrators. Instead, following research published by botnet experts can be equally informative.

Data gathered from a honeypot or by using other collection techniques allows administrators to follow botnet trends. Having an administrator with accurate and current knowledge of botnet activities allows them to adapt their security policies and stay current with trends. The administrator can also share any new information with the research community. All this information allows information technology staff to better protect a company's technology services.

### **Responding to Malicious Network Activity**

Once malicious activity is identified on a network a security policy should define the appropriate responses to eradicate it. The type of responses can be broken into two areas: proactive responses and retroactive responses.

Proactive responses aim to prevent damage to a company's network. These can include intrusion detection software that catches attempts to gain unauthorized access to network services. Another option is DDoS prevention at the level of the network service provider. Deep packet inspection software can examine network traffic for spam or virus software.

To prevent exploits of web site software utilities that verify page requests are from legitimate users, such as image verification challenge-response tests mentioned in case 1, can be used to prevent automated attacks. Checking web page referrers can also prevent attacks, however these can be spoofed. Also, human moderation can be required for certain tasks.

Retroactive responses are implemented only after security breaches occur. One simple way to stop malicious hosts from connecting to the network is to implement firewall rules that will block the illegitimate traffic. If the traffic generated by the attack requires greater resources to block then the organization's network service provider should be contacted to assist in the defense. Stopping the traffic at the level of the service provider reduces the load the company's network infrastructure. It may also be necessary to contact the authorities with any information that has been gathered, depending on the severity of the attack or if sensitive data has been compromised. A manual clean up process may also be necessary to reverse any mess created during the attack.

### **Conclusion**

The botnet problem has identified critical weak points in computer security. They pose one of the most severe threats to a business's technology infrastructure. As computer use becomes more prominent in businesses and households throughout the world, particularly in developing countries, cyber criminals will develop ways to harness and exploit their power for illegal use. A proactive defense to these attacks lies in raising the awareness towards Internet security for information technology managers. Managers must develop firm security policies to prevent critical losses if an attack occurs. These security policies must draw a compromise to achieve a system that is both secure and willing to be adopted by users.

Botnets are a growth industry and have attracted an increasing number of malicious software developers looking to earn a quick buck. This complicates the problem and adds pressure on security researchers and security software vendors to develop new defenses against professional Internet criminals.

While some cyber criminals are becoming more skillful in the malware they are developing, the policies that organizations implement prior to an attack can mitigate this threat. Organizations that develop, deploy, monitor, and test security tools throughout their network and implement information security policies that govern these devices, will be better able to avoid compromises and, in the event of an attack, a faster recovery. The diligent, informed, and responsible information technology managers are the most effective defense against these attacks.

**REFERENCES**

- Barford, P. & Yegneswaran, V. (2006) An inside look at botnets. Retrieved October 14, 2007 from <[http://pages.cs.wisc.edu/~pb/botnets\\_final.pdf](http://pages.cs.wisc.edu/~pb/botnets_final.pdf)>
- Brodsky, A. & Brodsky, D. (2007). A distributed content independent method for spam detection. Retrieved October 14, 2007 from <<http://soyuz.acs.uwinnipeg.ca/~abrodsky/papers/trinity-hb07.pdf>>
- CSI/FBI. (2006). CSI/FBI computer crime and security survey. Retrieved April 30, 2008 from <[http://i.cmpnet.com/gocsi/db\\_area/pdfs/fbi/FBI2006.pdf](http://i.cmpnet.com/gocsi/db_area/pdfs/fbi/FBI2006.pdf)>
- Daswani, N., Stoppelman, M. (April 10, 2007). The anatomy of Clickbot.A. Retrieved October 21, 2007 from <[http://www.usenix.org/events/hotbots07/tech/full\\_papers/daswani/daswani.pdf](http://www.usenix.org/events/hotbots07/tech/full_papers/daswani/daswani.pdf)>
- Ferris Research (2007), Industry statistics. Retrieved April 15, 2008 from <<http://www.ferris.com/research-library/industry-statistics/>>
- Goebel, J., & Holz, T. (April 10, 2007). Rishi: Identify bot contaminated hosts by IRC nickname evaluation. Retrieved October 21, 2007 from <[http://www.usenix.org/events/hotbots07/tech/full\\_papers/goebel/goebel.pdf](http://www.usenix.org/events/hotbots07/tech/full_papers/goebel/goebel.pdf)>
- Handley, E. & Rescorla, E. (November 2006). Internet denial-of-service considerations. *Request for Comments: 4732, Network Working Group*. Retrieved February 23, 2008 from <<http://tools.ietf.org/html/rfc4732>>
- Ianeli, N. & Hackworth, A. (December 1, 2005). Botnets as a vehicle for online crime. Retrieved October 21, 2007 from <<http://www.cert.org/archive/pdf/Botnets.pdf>>
- Krebs, B. (February 19, 2006). Invasion of the computer snatchers. *Washington Post*. Retrieved February 13, 2008 from <<http://www.washingtonpost.com/wp->

[dyn/content/article/2006/02/14/AR2006021401342\\_pf.html](http://dyn/content/article/2006/02/14/AR2006021401342_pf.html)>

- Malkin, G. & LaQuey Parker, T. (January 1993). Internet users' glossary. *Request for Comments: 1392, Network Working Group*. Retrieved February 21, 2008 from <http://www.faqs.org/rfcs/rfc1392.html>>
- McMillan, R. (October 21, 2007). Storm worm now just a squall. Retrieved October 21, 2007 from <http://www.pcworld.com/printable/article/id,138721/printable.html>>
- Nazario, Dr. J. (May 2004). The zombie roundup: Understanding, detecting, and disrupting botnets. Retrieved April 24, 2008 from <http://www.eecs.umich.edu/~emcooke/pubs/botnets-sruti05.pdf>>
- Perriot, F. & Szor, P. (October 2003). An analysis of the Slapper worm exploit. Retrieved February 20, 2008 from <http://www.symantec.com/avcenter/reference/analysis.slapper.worm.pdf>>
- Provos, N., McNamee, D., Mavrommatis, P., Wang, K., & Modadugu, N. (April 10, 2007). The ghost in the browser. Retrieved February 19, 2008 from [http://www.usenix.org/event/hotbots07/tech/full\\_papers/provos/provos.pdf](http://www.usenix.org/event/hotbots07/tech/full_papers/provos/provos.pdf)>
- Provos, N., Mavrommatis, P., Abu-Rajab, M., & Monroe, F. (February 4, 2008). All your iframe are point to us. Retrieved February 19, 2008 from <http://research.google.com/archive/provos-2008a.pdf>>
- Rajab, M., Zarfoss, J., Monroe, F., & Terzis, A. (October 2006). A multifaceted approach to understanding the botnet phenomenon. Retrieved October 14, 2007 from <http://www.imconf.net/imc-2006/papers/p4-rajab.pdf>>

Sandvine Intelligent Business Networks. (June 2004). Trend analysis: Spam trojans and their impact on broadband service providers. Retrieved October 14, 2007 from

<[http://www.sandvine.com/solutions/resource\\_library.asp](http://www.sandvine.com/solutions/resource_library.asp)>

Simon, D., Agarwal, S., Maltz, D. (April 10, 2007). AS-based accountability as a cost-effective DDoS defense. Retrieved October 21, 2007 from

<[http://www.usenix.org/event/hotbots07/tech/full\\_papers/simon/simon.pdf](http://www.usenix.org/event/hotbots07/tech/full_papers/simon/simon.pdf)>

Stewart, J. (March 15, 2004). Phatbot trojan analysis. Retrieved February 21, 2008 from

<<http://www.secureworks.com/research/threats/phantbot/>>

The HoneyNet Project. (July 13, 2007). Known your enemy: Fast flux service networks.

Retrieved October 2, 2007 from <<http://honeynet.org/papers/ff/>>

The HoneyNet Project. (November 7, 2007). Known your enemy: Behind the scenes of malicious web servers. Retrieved January 20, 2008 from <<http://honeynet.org/papers/wek/>>

The HoneyNet Project. (March 13, 2005). Know your enemy: Tracking botnets. Retrieved January 20, 2008 from <<http://honeynet.org/papers/wek/>>

Trend Micro. (November 2007). Taxonomy of botnet threats. Retrieved February 24, 2008 from

<<http://us.trendmicro.com/us/threats/enterprise/security-library/white-paper-listing/>>

Trend Micro. (February 2008). Trend Micro 2007 threat report and forecast. Retrieved February

24, 2008 from <<http://trendmicro.mediaroom.com/index.php?s=65&item=163>>

Wang, P., Sparks, S., & Zou, C., (April 10, 2007). An advanced hybrid peer-to-peer botnet.

Retrieved October 21, 2007 from

<[http://www.usenix.org/event/hotbots07/tech/full\\_papers/grizzard/](http://www.usenix.org/event/hotbots07/tech/full_papers/grizzard/)>

Webroot Software, Inc. (2006). From viruses to spyware: In the malware trenches with

small and medium-sized businesses. Retrieved April 20, 2008 from

<[http://www.webroot.com/shared/pdf/wp\\_SMBtrenches.pdf](http://www.webroot.com/shared/pdf/wp_SMBtrenches.pdf)>

Web History Project. (2003). History day abstracts. Retrieved April 20, 2008 from

<<http://1997.webhistory.org/historyday/abstracts.html>>

## FOOTNOTES

<sup>1</sup> More information about the Apache web server can be obtained by visiting

<<http://httpd.apache.org>>.

<sup>2,3</sup> More information about SSH and SFTP can be obtained by visiting

<<http://www.openssh.com>>.

<sup>4</sup> More information about Nmap can be obtained by visiting <<http://nmap.org>>.