10

Department of Electrical and Computer Engineering
University of Victoria

SENG 460 – Practice of Information Security and Privacy

Assignment 5

Case Study: Monitoring

Table of Contents

3.2.2 Centralized Architecture

Pony Entertainment is a large company with global operations. Ideally, utilizing one vendor for system architecture throughout the company would allow for a better security architecture overall. In terms of ease of use, a Linux based network system would allow for very rapid detection of intrusions through the use of Netflow, Syslog and front end applications for searching through the information for threats encountered by Syslog. It is recommended that all servers and network infrastructure be updated with a Linux based operating system. The servers and equipment should be housed in central building that come with physical security implementations such as CCTV and onsite security personnel to prevent physical break-ins.

3.2.2.1 Centralized Architecture – Business Challenges Addressed

Migrating current infrastructure to a Linux based operating system for the network and server components can be cost prohibitive. The purpose of having a universal centralized architecture however allows for efficient and fast access to logged data, as well as upgrades to the rule set used for implementing policies for access across the entire network. Implementing this would directly combat two of the three requirements of the new monitoring design which include:

- Notice intrusions before being notified by either the Public or the hackers
- Notice suspicious activity close to real time and use this information to adjust current defences.

Having one standardized system also allows for multiple employees to become proficient in the network architecture, so if an employee leaves Pony Entertainment, it would not be difficult to find replacements who are familiar with the single architecture (i.e. only Linux servers rather than Windows, Linux and UNIX). Standardized and centralized architecture throughout the company will allow for threats to be found and dealt with quickly and more efficiently.

Ultimately, with 50 million records being compromised due to a lack of valid security measures in the past, Pony Entertainment has lost income from the degradation of their public image and security. The cost of investing in a standardized and centralized architecture will help to bring back customer confidence and loyalty.

3.2.3 Capacity Planning

Pony Entertainment maintains records for user ID's, passwords, credit card data and personal information. The amount of capacity required for logging and retaining network data would be based on the amount of good logs and bad logs observed. Good logs are based on a rule set that distinguishes normal network traffic from the bad logs which is traffic that can be considered questionable or threat worthy. Capacity planning should be considered as well as future capabilities for log storage as Pony Entertainment grows.

3.2.6 Reporting and Analysis

Having an efficient way to analyze and report the information obtained as well as provide that information to the decision makers is the final stage in a good monitoring system. The elements of this phase relay on the IT personnel involved who are trained to take the information and make decisions based on the severity of events. Having employees who are then able to summarize their findings and create reports to provide to management and law enforcement is important part of making sure information gets to where it needs to be.

3.2.6.1 Reporting and Analysis – Business Challenges Addressed

The ability to quickly analyze information and provide that information to police, investigators, management, or people of authority, allows for a rapid response to find and minimize the impact of an intrusion should one happen. Decisions cannot be made without the right information presented in the proper context. Pony Entertainment can better understand and focus their resources if informed of threats and law enforcement can better prosecute those who do perpetrate these intrusions, all with the right information. Analysing and Reporting of the information will directly combat the third criteria for creating a monitoring system which is:

- Be able to assist Law Enforcement with prosecuting the attackers.

4.0 Prior to System Implementation

Prior to the implementation of the new monitoring system outlined in 3.0, the current system must be audited in terms of hardware available and current security policies used. These items must be arranged so that they can be incorporated into the new monitoring system. The new system must be implemented in phases, with security policies and normalizing of data to come first to build up the defences using the current technology. Once this is satisfactory, begin implementing new policies and training geared to the new infrastructure required. Once this has been adapted, begin phasing out the old hardware and incorporating the new hardware using the new security policies. Since a normalizing data system would already be in place, implement the new hardware in sections, which can maintain the integrity of the system. Piece by piece add the centralized architecture, then when the final hardware and training is complete, the new monitoring system will be fully functional.

"Conficker" Worm Report

What it is, How it works, Why it was effective and Worldwide damage

Conficker is a worm which was first detected in November 2008. A worm is a piece of malicious code that replicates itself in order to spread its code to other computers. Conficker spread to computers through various revisions of its code, which it upgraded through a network connection. Microsoft website has the following figure to describe how it Conficker works[2]:

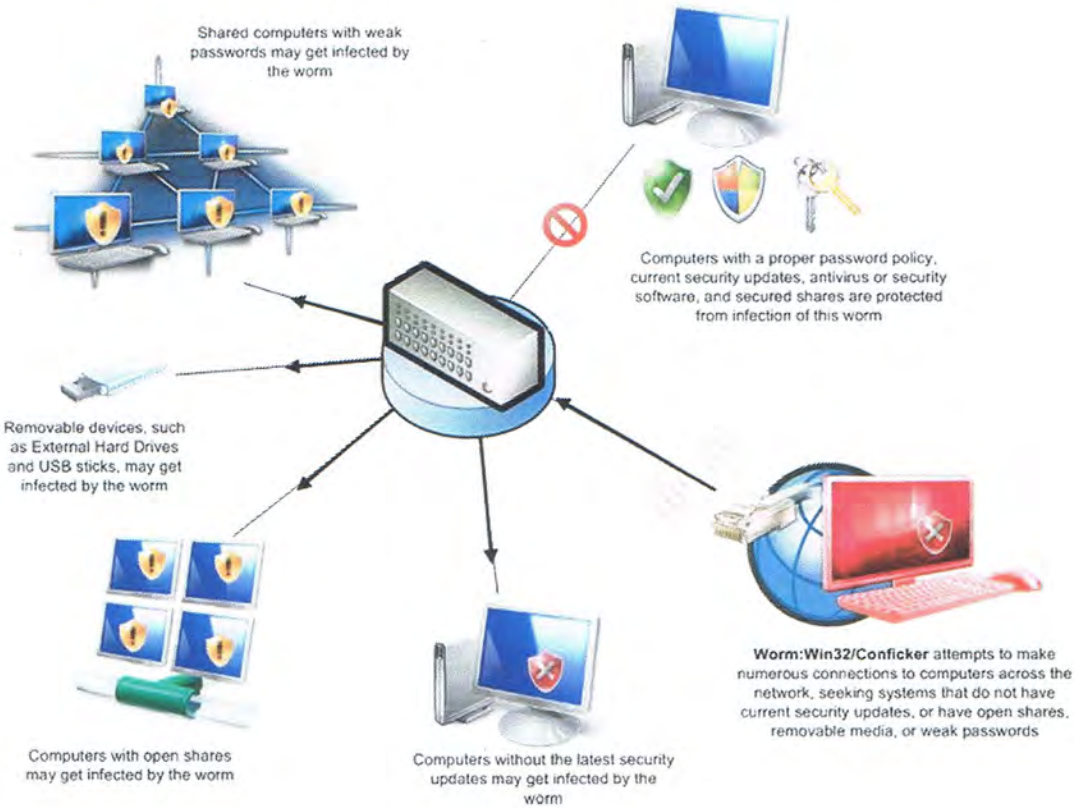

Figure 2: How Conficker Works

Conficker was effective because it wasn't a typical worm. I wasn't just exploiting a network hole in the Windows Operating System. Conficker came equipped with a sophisticated method of cracking administrator passwords. This ability made it particularly difficult to remove. It also copied itself to USB drives so that it could spread itself even if online functionality was removed.

[2] http://www.microsoft.com/security/pc-security/conficker.aspx#EKE, February 2014

Conficker's ability to update itself was another reason for its effectiveness. The following table shows the variants of Conficker, the dates each variant was detected, and various methods it was able to update and defended itself.

Table 1: Conficker Worm Variants[3]

Variant	Detection date	Infection vectors	Update propagation	Self-defense	End action

Conficker A	2008-11-21	• NetBIOS
		  • Exploits MS08-067 vulnerability in Server service[27]
			• HTTP pull
			  • Downloads from trafficconverter.biz
			  • Downloads daily from any of 250 pseudorandom domains over 5 TLDs[32]
				None	• Updates self to Conficker B, C or D[31]

Conficker B	2008-12-29	• NetBIOS
		  • Exploits MS08-067 vulnerability in Server service[27]
		  • Dictionary attack on ADMIN$ shares[32]
		• Removable media
		  • Creates DLL-based AutoRun trojan on attached removable drives[17]
			• HTTP pull
			  • Downloads daily from any of 250 pseudorandom domains over 8 TLDs[32]
			• NetBIOS push
			  • Patches MS08-067 to open reinfection backdoor in Server service[33][34]
				• Blocks certain DNS lookups
				• Disables AutoUpdate
					• Updates self to Conficker C or D[31]

Conficker C	2009-02-20	• NetBIOS
		  • Exploits MS08-067 vulnerability in Server service[27]
		  • Dictionary attack on ADMIN$ shares[32]
		• Removable media
		  • Creates DLL-based AutoRun trojan on attached removable drives[17]
			• HTTP pull
			  • Downloads daily from 500 of 50,000 pseudorandom domains over 8 TLDs per day[27]
			• NetBIOS push
			  • Patches MS08-067 to open reinfection backdoor in Server service[33][34]
			  • Creates named pipe to receive URL from remote host, then downloads from URL
				• Blocks certain DNS lookups
				• Disables AutoUpdate
					• Updates self to Conficker D[31]

Conficker D	2009-03-04	None	• HTTP pull
			  • Downloads daily from any 500 of 50,000 pseudorandom domains over 110 TLDs[30]
			• P2P push/pull
			  • Uses custom protocol to scan for infected peers via UDP, then transfer via TCP[35]
				• Blocks certain DNS lookups[35]
				  • Does an in-memory patch of DNSAPI.DLL to block lookups of anti-malware related web sites[36]
				• Disables Safe Mode[38]
				• Disables AutoUpdate
				• Kills anti-malware
				  • Scans for and terminates processes with names of anti-malware, patch or diagnostic utilities at one-second intervals[37]
					• Downloads and installs Conficker E[31]

Conficker E	2009-04-07	• NetBIOS
		  • Exploits MS08-067 vulnerability in Server service[39]
			• NetBIOS push
			  • Patches MS08-067 to open reinfection backdoor in Server service
			• P2P push/pull
			  • Uses custom protocol to scan for infected peers via UDP, then transfer via TCP[34]
				• Blocks certain DNS lookups
				• Disables AutoUpdate
				• Kills anti-malware
				  • Scans for and terminates processes with names of anti-malware, patch or diagnostic utilities at one-second intervals[39]
					• Updates local copy of Conficker C to Conficker D[40]
					• Downloads and installs malware payload
					  • Waledac spambot[38]
					  • SpyProtect 2009 scareware[41]
					• Removes self on 3 May 2009 (but leaves remaining copy of Conficker D)[42]

[3] http://en.wikipedia.org/wiki/Conficker, February 2014

Conficker is estimated to have infected between 9 million to 15 million computers at its height in January 2009. The following graph shows the various geographical locations that the C variant of Conficker worm's infection was felt.

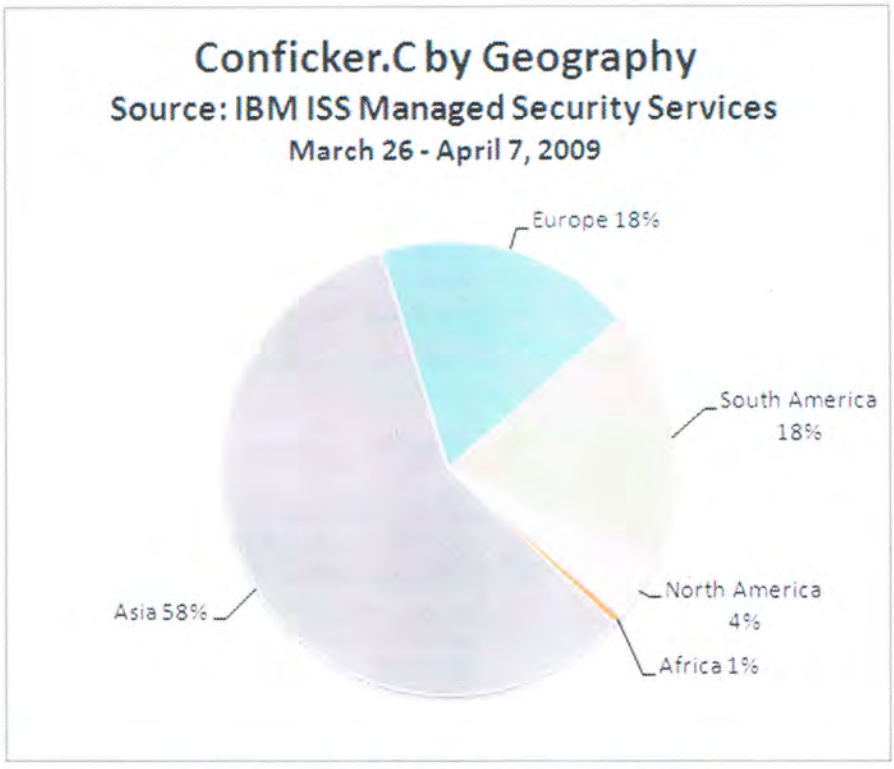

Figure 3: Conficker Infection by Geographic Location[4]

According to Cyber Security Institute who has performed various economic impact studies on malware infected computers[4]:

"Any analysis of the true impact of Conficker must also factor in the (wasted) time, resources, and energies of the cyber-community, governments, companies and individuals. Extrapolating out from studies on the average cost of similar past attacks, the total economic cost of this worm (including the cost of efforts to combat the worm, the cost of purchasing counter-measure software) could be as high as $9.1 billion. Even using the single, outlying data source that suggests a much more limited scope of infection (<200,000 —vastly less than all other sources suggest—the cost of this virus is still roughly $200 million dollars."

Based on this evaluation, the economic impact was estimated to be $9.1 billion dollars.

[4] http://www.zdnet.com/blog/security/confickers-estimated-economic-cost-9-1-billion/3207 February 2014

Three Major Worms prior to Conficker

Although many worms existed in various forms prior to Conficker, three which can be somewhat related to them are now discussed.

Sasser

Sasser spread by exploiting a buffer overflow in the component known as LSASS (Local Security Authority Subsystem Service) on the affected Windows operating systems. It is estimated that Sasser infected tens of millions of computers after its first detection in 2004. The economic damage worldwide caused by Sasser is estimated to be $14.8 and $18.1 billion dollars[5].

myDoom

As of January 2004, myDoom was the fastest spreading email worm of all time. It spread "via e-mail, appearing as a transmission error, with subject lines including "Error", "Mail Delivery System", "Test" or "Mail Transaction Failed" in different languages, including English and French. The mail contains an attachment that, if executed, resends the worm to e-mail addresses found in local files such as a user's address book. It also copies itself to the "shared folder" of peer-to-peer file-sharing application KaZaA in an attempt to spread that way."[6]

MyDoom affected around 500,000 computers worldwide and is estimated to have caused $4 billion dollars in damage[7].

SoBig

The Sobig Worm was a computer worm that infected millions of Internet-connected, Microsoft Windows computers in August 2003. Sobig had many different variants which all spread through email. It was unique in that it masqueraded as something other than malware. The Sobig.F worm reactivated itself on March 10, 2013. On March 14 the same year, Microsoft announced that they will pay $250,000 for information leading to the arrest of the creator of the Sobig worm. To date, the perpetrator has not been caught.[8]

According to the mi2g Intelligence Unit, soBig overtook Klez, Love Bug and Yaha in the league table of most economically damaging malware tracked within the SIPS database, having recorded $14.62 billion of economic damages worldwide measured in terms of lost productivity as of August 2003.[9]

---

[5] http://www.mi2g.com/cgi/mi2g/frameset.php?pageid=http%3A//www.mi2g.com/cgi/mi2g/press/100504.php, February 2014

[6] http://en.wikipedia.org/wiki/Mydoom, February 2014

[7] http://www.scmagazine.com/mydoom-rings-up-4-billion-in-damage/article/30787/, February 2014

[8] http://en.wikipedia.org/wiki/Sobig, February 2014

[9] http://www.net-security.org/secworld.php?id=1635, February 2014

Is Conficker the last of the Big Worms?

Conficker is not the last of the big worms. Although the most prevalent worms after Conficker have been worms that directly target institutions (such as Stuxnet with the SCADA systems in Iran). The world has not yet been infected to the same degree. Although I imagine Iran would argue that the economic impact of Stuxnet was immense, globally the world has not felt this as with Conficker.

The bottom line is that humanity is in love with the internet, with information. We keep making our online presence known by entering our entire lives online into sites such as a Facebook, online banking and local computers such as the one I am writing this assignment on. All of this requires programs and hardware to store this information.

So long as people keep creating software, hardware and networking environments for the public to enter their information, there will be people trying to poke holes in these programs, hardware and networks. People will keep attempting to exploit and gather as much information as they can, as information is the new currency of our age.

All it takes is a single vulnerability to be made, and a single vulnerability to be detected to create the next Conficker.