**Assignment 4**

Question: Prepare a report on following malware using following points and complete the comparison sheets:

1. History of the malware.
2. Author of the malware.
3. What is peculiarity/specialty of the malware.
4. How does the malware the works?
5. Tools used for performing malware attacks and tools used as an counter measure to prevent the attack.
6. Possible subtype under the malware or various versions of the malware. Also write down examples of the malware.
7. Any damage caused by the malware with the associated cost.
8. Any other things you find useful to add.

Malware to be included:

1. Adware
2. Attack Kit
3. Auto Rooter
4. Backdoor/Trapdoor
5. Drive-by-downloads
6. Exploits
7. Flooders (DoS Client)
8. Keyloggers
9. Logic Bombs
10. Macro Virus
11. Malicious Mobile Code
12. Rootkit
13. Spammer Program
14. Worms
15. Zombie, Bot
16. APT
17. Phishing
18. Web shells
19. DoS
20. Payload
21. Ransomware
22. Virus
23. Any other malware missing in the list/latest malware

In this assignment each student is supposed to write his own report (individual assignment) in his own words. Later we will prepare a excel sheet of comparison of all malware using above points (group assignments/class assignment).

Adware

History

Over 10 years ago, a new strain of malware was born and christened Spyware. Through large security holes in XP and Internet Explorer, the pay load entered without user knowledge or consent and the drive by download had come to age. None of the popular security products of the day detected nor remediated the new Spyware threat.

Computers were now deluged with pop up ads making many systems unusable and frequently painfully slow. In early 2003, PC Pitstop’s free online diagnostic began to detect the presence of spyware, and offered a variety of spyware solution including Pest Patrol and Spyware Doctor. Perhaps the largest spyware company of the time was called Gator. Once Gator learned that PC Pitstop detected their product, Gator, drunk with ill gotten advertising cash, sued PC Pitstop and our partner, CompUSA. The lawsuit stated that Gator was not Spyware but rather Adware and they were seeking damages.

At that point, PC Pitstop lost our partnership with CompUSA, and chose to settle with Gator and agreed to stop calling them Spyware.

In August 2004, Microsoft released XP Service Pack 2 which closed many of the security holes in XP and Internet Explorer and eliminated the drive by download. Microsoft’s move reversed Gator’s fortunes, and in 2008, Gator closed their doors for good. Was Gator spyware? Of course there were.

In 2013, the term spyware is rarely heard, but adware is alive and well. There are no more drive by downloads. Adware now enters through security holes in Java and Flash player. Perhaps more frequently, adware enters by bundling its payload with the installers of other software applications. Then they pay the publisher for each install giving the publishers an incentive to deceive users to erroneously install adware.

Gone are the pop up ads, but adware’s motivations are still the same, to monetize our computers and our eyeballs. A good example is a new adware called Conduit that often calls itself Trustworthy Computing. Once Conduit hits pay dirt, it places an unwanted toolbar, and hijacks all your searches in IE, Firefox and Chrome. Worse yet, once it is on the target system, it is extremely difficult to completely remove.

Just like in 2003, almost none of the major security vendors block and detect adware. PC Matic and a handful of smaller security products block the extremely annoying online pest called adware.

Author of the adware

The 2003 *Microsoft Encyclopedia of Security* and some other sources use the term "adware" differently: "any software that installs itself on your system without your knowledge and displays advertisements when the user browses the Internet",[[1]](https://en.wikipedia.org/wiki/Adware" \l "cite_note-MS-Enc-Sec-1) i.e., a form of [malware](https://en.wikipedia.org/wiki/Malware).

Specialty of the adware

Adware, or advertising-supported software, is [software](https://en.wikipedia.org/wiki/Software) that generates revenue for its developer by automatically generating [online advertisements](https://en.wikipedia.org/wiki/Online_advertising) in the user interface of the software or on a screen presented to the user during the installation process. The software may generate two types of revenue: one is for the display of the advertisement and another on a "[pay-per-click](https://en.wikipedia.org/wiki/Pay-per-click)" basis, if the user clicks on the advertisement. The software may implement advertisements in a variety of ways, including a static box display, a banner display, full screen, a [video](https://en.wikipedia.org/wiki/Video), [pop-up ad](https://en.wikipedia.org/wiki/Pop-up_ad) or in some other form.

Working of Adware

When a company chooses to use adware to support its software, the adware is bundled or integrated with the program itself. This allows for advertisements to load automatically. Ad space is bought by other companies who want to market to the users of that program, and the revenue made by selling ad space may be used to cover user and/or development costs. Some companies integrate adware with their program in such a way that the program cannot run unless the ads are being displayed. Any ad blockers must then be disabled in order to use the software. Adware also often collects personal information and tracks users’ browsing habits, which is then sold to third parties.

Tools used for performing malware attacks and tools used as an counter measure to prevent the attack.

There are a number of different tactics that can be used to address different aspects of malicious adware. These methods offer a stronger defense when used in combination.

* Use an ad blocker: In many cases, a free service can be used without viewing ads. Ad blockers prevent advertisements from being displayed in a browser, eliminating the chance of clicking on something malicious while preventing drive-by downloads.
* Pay for premium, ad-free versions of popular services: The easiest way to avoid adware is to pay for the service that you are using. The majority of businesses do not serve ads to premium subscription users which helps ensure that malicious adware will not be able to reach them.
* Buy devices from trusted companies with reputable, built-in security: There have been numerous incidents where people who purchase low-cost Android devices find that adware is installed in core files. These devices are especially susceptible to infection and purchasing one should be avoided at all costs.
* Use an antivirus: Some antivirus programs can be used to block malicious adware, but many of them are not able to discern between the legitimate and the malicious as of yet. Antiviruses may also be blocked by some adware, in which case a more aggressive method should be used.
* Use an adware removal software: When negatively impacted by adware, this type of software will clean up any unwanted programs on the device.

Possible subtype under the malware or various versions of the malware. Also write down examples of the malware.

SpyTrooper

SpyTrooper is a threat roaming the internet. This program is categorized as adware. It is difficult to remove SpyTrooper without advanced security software. The program comes with no uninstall functions and inserts keys into the registry, which implies that we have to remove it manually.

Any damage caused by the malware with the associated cost

Adware is often harmless, but sometimes the adware you inadvertently download can also be malware. It can continuously run in the background, usually without your knowledge, using computer resources such as processor and memory. Your first warning that you have an abundance of malicious software on your computer may be the deterioration of performance.

Attack Kit

History of the attack kit

The term rootkit or root kit originally referred to a maliciously modified set of administrative tools for a [Unix-like](https://en.wikipedia.org/wiki/Unix-like) [operating system](https://en.wikipedia.org/wiki/Operating_system) that granted "[root](https://en.wikipedia.org/wiki/Superuser)" access. If an intruder could replace the standard administrative tools on a system with a rootkit, the intruder could obtain root access over the system whilst simultaneously concealing these activities from the legitimate [system administrator](https://en.wikipedia.org/wiki/System_administrator). These first-generation rootkits were trivial to detect by using tools such as [Tripwire](https://en.wikipedia.org/wiki/Open_Source_Tripwire) that had not been compromised to access the same information. Lane Davis and Steven Dake wrote the earliest known rootkit in 1990 for [Sun Microsystems](https://en.wikipedia.org/wiki/Sun_Microsystems)' [SunOS](https://en.wikipedia.org/wiki/SunOS) UNIX operating system In the lecture he gave upon receiving the [Turing award](https://en.wikipedia.org/wiki/Turing_award) in 1983, [Ken Thompson](https://en.wikipedia.org/wiki/Ken_Thompson) of [Bell Labs](https://en.wikipedia.org/wiki/Bell_Labs), one of the creators of [Unix](https://en.wikipedia.org/wiki/Unix), theorized about subverting the [C compiler](https://en.wikipedia.org/wiki/C_compiler) in a Unix distribution and discussed the exploit. The modified compiler would detect attempts to compile the Unix login command and generate altered code that would accept not only the user's correct password, but an additional "[backdoor](https://en.wikipedia.org/wiki/Backdoor_(computing))" password known to the attacker. Additionally, the compiler would detect attempts to compile a new version of the compiler, and would insert the same exploits into the new compiler. A review of the source code for the login command or the updated compiler would not reveal any malicious code. This exploit was equivalent to a rootkit.

Author of the malware.

A trojan called NTRootkit created by Greg Hoglund in 1999.

What is peculiarity/specialty of the malware.

The acronym ATK stands for Attack Tool Kit. It is a mixture of security scanner and exploiting framework. ATK can be automated. It consists of enumeration and generation of Report. It also supports modularity and necessary changes can be done on the fly.

How does the malware the works

Rootkits work by using a process called modificatio. It consists of changing of user account permissions and security. Usually this is granted by a computer administrator. Modification is often used to make positive changes for improving systems, attackers seeking full control will use modification to grant themselves unlimited access to cause damage.

Tools used for performing malware attacks and tools used as an counter measure to prevent the attack.

A new malware-powered attack kit that stitches up two Trojans and a coinminer to mine for data and Monero was observed while scanning for vulnerable machines from China, Taiwan, Italy, and Hong Kong, and spreading itself over the Internet and local area networks.

Trend Micro's Don Ovid Ladores, Michael Jhon Ofiaza, and Gilbert Sison detected the attack kit while it was dropping what looked like random files in the Windows folder of computers who had the 445 port open and ready to be compromised with an SMB exploit targeting the [Windows MS17-010 Server Vulnerability](https://success.trendmicro.com/solution/1121399-ms17-010-smb-remote-code-execution-exploit-appears-on-the-suspicious-connection-logs) already patched back in 2017.

The multi-stage infection process uses what Trend Micro calls Trojan.Win32.INFOSTEAL.ADS to gain an initial foothold after successfully exploiting its victims, a malware strain which will connect to a command-and-control (C&C) server to send its masters info about the infected host and to grab the next malware payloads.

Possible subtype under the malware or various versions of the malware. Also write down examples of the malware

An exploit kit is a programming tool that allows someone who does not have any experience writing software code to create, customize and distribute [malware](https://searchsecurity.techtarget.com/definition/malware). Exploit kits are known by a number of other names, including infection kit, [crimeware](https://searchsecurity.techtarget.com/definition/crimeware) kit, DIY attack kit and malware toolkit.

Exploit kits have graphical application program interfaces ([APIs](https://searchmicroservices.techtarget.com/definition/application-program-interface-API)) that allow non-technical users to manage sophisticated attacks capable of stealing corporate and personal data, orchestrating denial of service ([DoS](https://searchsecurity.techtarget.com/definition/denial-of-service)) exploits or building [botnets](https://searchsecurity.techtarget.com/definition/botnet). Most kits are built by professional programmers who exploit browser and client-side [vulnerabilities](https://whatis.techtarget.com/definition/vulnerability) that have already been publicly disclosed. The kits, which are commercially available on underground discussion forums, can cost as little as $100 or as much as $10,000.

Ironically, the high profits that can be gained by selling crimeware kits have led developers to model their software distribution model after that of legitimate software [vendors](https://whatis.techtarget.com/definition/vendor). Many crimeware kits have clearly defined refund policies, licensing options, digital rights management ([DRM](https://searchcio.techtarget.com/definition/digital-rights-management)) components and customer service.A new malware-powered attack kit that stitches up two Trojans and a coin miner to mine for data and Monero was observed while scanning for vulnerable machines from China, Taiwan, Italy, and Hong Kong, and spreading itself over the Internet and local area networks.

Autorooter

Introduction

Programs designed to automatically scan and attack target computers are called Autorooters.

A successful Autorooter grants a complete control of a target machine, scanning networks for vulnerable machines, gaining unauthorized administrative access and installing backdoors.

What Does an Autorooter Do?

The term autorooter is based on security lingo for successfully cracking and gaining privileged access to a machine. The act, known as "rooting" a system, originates from the name of the administrative account on a Unix box - "root". The "auto" prefix stems from the fact that these devices essentially package, or automate, the cracking process from start to finish. They can be designed to scan a network for vulnerable machines or attack everything they come across. Once a machine is successfully compromised, or rooted, any type of malicious code can be installed and configured: data might be captured (using a tool known as a sniffer), Web pages defaced, servers installed. Some autorooters are finished after sending the results back to the cracker, others may install zombies that await further instructions from the attacker, such as IRC-controlled denial of service slaves.

**Who Uses Autorooters?**

While many automated network scanning tools may be used for legitimate as well as illegitimate purposes, this is not the case with autorooters, which are generally used by crackers, a name attributed to hackers with malicious intentions. What's particularly troubling about autorooters is the skill level of those who most often employ them.

Although, normally written by knowledgeable programmers, autorooters are often used by a type known as "script kiddies". Script kiddies are held in great disdain by the security community because they have very little coding ability but can inflict considerable damage through the use of simple point-and-click hacking tools. Such users rarely understand how or why an exploit works, but they get results because they have the time and resources to scan thousands of machines and are oblivious to the ramifications. A script kiddie needs only to download an autorooter, enter a net block or range of IP addresses and turn it loose. They don't need to know how their malicious tools function, they only know they work. For script kiddies, it's point, click, and crack.

**Why are Autorooters Dangerous?**

In order to understand the threat posed by these dangerous tools, we need to outline the process they replaced. First, crackers would have needed a scanning tool, such as [Nmap](http://www.nmap.org/), to explore a large range of machines. Next, they would review all of this data, looking for specific ports, operating systems or servers to generate a list of potential targets. At this point, an exploit would be needed to compromise the targets remotely. Running the exploit on each machine would yield some successful cracks. Later each device would need to be configured for the cracker's use: malicious code installed, log files tweaked, backdoors setup. To further compromise the new private networks within reach, the process would be repeated.

It wasn't rocket science, but the steps demanded time, patience, organization and a basic understanding of what was taking place. Autorooters changed all of that. A cracker no longer needs to manually scan machines or review port and OS information. The legwork is eliminated. Rather, they merely "aim" the autorooter at a range of potential targets, i.e. 198.162.0.0 - 192.168.255.255. Every network device within that address space will be inspected within a short amount of time. All data will be sorted automatically by the autorooter, which will then launch attacks against all potentially vulnerable machines. Once successful, the autorooter can "clean" the log files to eliminate traces of itself and prepare the system as it was configured to do by the author.

The implications are frightening for professionals and home users alike. Now, a novice cracker can hit dozens of machines in minutes, simply playing the odds. If one in a hundred machines is vulnerable, a relatively quick scan of any address block yields a plethora of victims. Most troubling is the fact that entire networks, if caught without the proper patches, can be harvested by a single autorooter. This can occur because the autorooter, much like a worm, often attacks the surrounding network of its victim. If a firewall is configured improperly, a public server, cracked by an autorooter, can quickly begin attacking machines on the private network.

**Evolution of the Autorooter**Autorooters can be thought of as advanced versions, or combinations, of some rather traditional types of malicious code, such as:

Exploits - Small pieces of code that take advantage of a specific flaw or bug in a piece of software (a vulnerability) to gain privileged access to a machine or network. [Buffer overflows](http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci549024,00.html) are the basis for many exploits.

Rootkit - A tool that allows intruders to illicitly obtain administrator-level access to a computer or computer network. Once a cracker finds a victim, the rootkit is used to perform specific tasks such as erasing logs and installing backdoors. It "prepares" the machine for a cracker to perform view, alter, or steal data, or perform other malicious functions.

Trojan - A compromised or tainted file which, when executed by a privileged user on a target machine, installs malicious code on the machine. Often used to replace valid system binaries and can prep a system similar to a rootkit.

Virus - A piece of malicious code included with a valid file that infects the file and then replicates itself on a target machine. Normally reaches a system by piggybacking a valid file via e-mail or disk. A virus can inflict damage on a target machine by corrupting data, wiping out data, or otherwise making data unavailable.

Worm - An automated program which attempts to spread via the network by infecting every vulnerable host within reach. This is a very popular method of spreading an autorooter.

Autorooters have evolved from each of the above and combines different characteristics and methodologies from each of them. The exploit itself is the foundation, as it is what is manipulated to actually gain access to a target machine. Once access has been gained, the rootkit deploys the malicious payload on the victim. Upon installation, the malicious code can set about inflicting damage. The autorooter can actually "poison" its target and the surrounding network like a virus, trojan or worm. It might be designed to spread via multiple methods such as e-mail, a carefully crafted java-script executed by those who visit an infected server or by simply crawling the network looking for more victims. A programmer could code an autorooter to merely crack the machines it scans, or could very easily add in worm-like propagation techniques, allowing for multi--generational replication. Add to this the fact that targets may be culled from massive, automated scans and the autorooter represents a vicious new breed of cracking tool.

A Super Exploit?

The term "super exploit" refers to an exploit or attack that could, theoretically, traverse the Internet in a very short time. While it might seem far fetched, autorooters are turning this scary notion into a reality. How so? While it is unlikely that one specific attack could affect the majority of machines on the Internet, due in part to the diversity of configurations, if designed to pursue a specific exploit, such as one of several DNS exploits, or if a flaw was exposed in a very popular service, such as the Microsoft Unicode vulnerabilities, an autorooter could spread at a frightening rate. Why? As mentioned above, autorooters can be instructed to propagate. Once "released", they may be very difficult to contain. A single compromised server can rapidly become many, each of which become cracking tools themselves. An exponential rate of return follows.

Adding to this problem is the advanced nature of these programs. Worms such as [Nimda](http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci770982,00.html) were designed to take advantage of multiple exploits and spread via different methods. Thousands of servers can fall victim in hours, clogging Internet traffic and wreaking havoc on networks around the world. While a "super exploit" has not yet been seen, many security professionals fear that it could happen. The appearance of Nimda in September, 2001, gave credence to these fears.

Many administrators and security experts unhappily recall September 18, 2001, the day Nimda exploded. I noticed the first log entries twenty minutes after I sat down for the morning. Within hours, hundreds of different machines were hitting my servers every few minutes. Months later, administrators continued to see Nimda HTTP requests every day. While not an autorooter in the strictest sense, Nimda produced similar results. As a worm, its main goal was to spread. But imagine Nimda built with an even more malicious intent, perhaps the goal of silently monitoring network traffic, or waiting to participate in a distributed denial of service attack, directed to begin by scanning your network. Something to keep in mind is that these tools are relatively new and raw - in some ways they're still just an idea. At the moment - any of the readily available autorooters are likely basic and clunky -perhaps just a step above traditional rootkits. And the "features" vary, since they were likely tailored to the needs of whoever wrote it. But in the future, that might not be the case. It would be trivial to code the capabilities to spread onto surrounding networks - as well as virtually any other post-crack activity. So, there is a very good chance we will see more advanced versions, with different traits, slipping into the public sector. As Nimda illustrated, when that happens, security administrators will be well-advised to have adequate defences in place.

**The Defense**

The best methods to protect oneself from autorooters are rooted (no pun intended) in basic security strategies, such as:

* Firewalls - A properly configured firewall filtering both inbound and outbound traffic can work wonders. These devices can be used to prevent and contain the spread of an autorooter.
* Intrusion detection systems - Although they cannot directly prevent the spread of an autorooter, they can alert an IT staff of both incoming and outgoing autorooter signatures. This warning time can help to minimize damages, identify and disconnect infected machines, and patch systems.
* Disabled services - Removing and disabling unnecessary services is a crucial defense. Since most autorooters spread via compromised network services, this simple step is very effective. If a machine is not be used as a server - disable everything.
* Anti-virus techniques - Workstation users need to employ standard antivirus measures since autorooters can spread via e-mail and Web scripts. Modern anti-virus programs can be used to detect and prevent such attempts.
* Updated patches - Another simple, but critical requirement. The success of autorooters depends on wide spread vulnerabilities. If a system is regularly patched, it's unlikely that an autorooter using anything greater than a zero-day exploit will succeed. This step is even more important for network administrators. If the machines under your control are vulnerable to a specific exploit, an autorooter could rip through the network in seconds.
* Incident response - If one of your machines is hit, please act responsibly and remove it from the network. The longer it remains active and on-line, the more machines it can in turn attack.

Backdoor /Trapdoor

A computer trapdoor, also known as a back door, provides a secret -- or at least undocumented -- method of gaining access to an application, operating system or online service. Programmers write trapdoors into programs for a variety of reasons. Left in place, trapdoors can facilitate a range of activities from benign troubleshooting to illegal access.

**Legitimate Use**

Programmers typically don't create and retain trapdoors with malicious intent. They leave them in place for legitimate testing or debugging purposes, or to give service technicians emergency access to a system. Weaknesses in design logic also can introduce trapdoors into program code inadvertently and innocently. Many software developers include undocumented trapdoor passwords, which they use for maintenance or unspecified purposes. Software companies rarely acknowledge the presence of trapdoors and trapdoor passwords in proprietary software -- software whose source code is not distributed publicly -- but users sometimes expose them.

## Security Vulnerability

Because trapdoors allow anyone with knowledge of them to circumvent normal security procedures, unscrupulous individuals can exploit them for nefarious purposes. Software vendors may expect and hope that trapdoors and trapdoor passwords remain secret, but as users become more technically savvy, they become increasingly likely to discover them, accidentally or intentionally, and thus to create security vulnerabilities. Some users exploit trapdoors or disclose them so others can exploit them, rather than report the presence of such vulnerabilities to the developer of the software that contains them.

Drive by download

Drive-by download means two things, each concerning the unintended [download](https://en.wikipedia.org/wiki/Download) of [computer](https://en.wikipedia.org/wiki/Computer) [software](https://en.wikipedia.org/wiki/Software) from the [Internet](https://en.wikipedia.org/wiki/Internet):

1. Downloads which a person has authorized but without understanding the consequences (e.g. downloads which install an unknown or counterfeit [executable program](https://en.wikipedia.org/wiki/Executable_program), [ActiveX](https://en.wikipedia.org/wiki/ActiveX) component, or [Java](https://en.wikipedia.org/wiki/Java_(software_platform)) applet) automatically.
2. Any [download](https://en.wikipedia.org/wiki/Download) that happens without a person's knowledge, often a [computer virus](https://en.wikipedia.org/wiki/Computer_virus), [spyware](https://en.wikipedia.org/wiki/Spyware), [malware](https://en.wikipedia.org/wiki/Malware), or [crimeware](https://en.wikipedia.org/wiki/Crimeware).

Drive-by downloads may happen when visiting a [website](https://en.wikipedia.org/wiki/Website), opening an e-mail attachment or clicking a link, or clicking on a deceptive pop-up window:[[2]](https://en.wikipedia.org/wiki/Drive-by_download" \l "cite_note-2) by clicking on the window in the mistaken belief that, for example, an error report from the computer's operating system itself is being acknowledged or a seemingly innocuous advertisement pop-up is being dismissed. In such cases, the "supplier" may claim that the user "consented" to the download, although the user was in fact unaware of having started an unwanted or malicious software download. Similarly if a person is visiting a site with malicious content, the person may become victim to a drive-by download attack. That is, the malicious content may be able to exploit [vulnerabilities](https://en.wikipedia.org/wiki/Vulnerability_(computing)) in the browser or plugins to run malicious code without the user’s knowledge.[[3]](https://en.wikipedia.org/wiki/Drive-by_download" \l "cite_note-:0-3)

A drive-by install (or installation) is a similar event. It refers to [installation](https://en.wikipedia.org/wiki/Installation_(computer_programs)) rather than download (though sometimes the two terms are used interchangeably).

## Process

When creating a drive-by download, an attacker must first create their malicious content to perform the attack. With the rise in exploit packs that contain the vulnerabilities needed to carry out drive-by download attacks, the skill level needed to perform this attack has been reduced. [[3]](https://en.wikipedia.org/wiki/Drive-by_download" \l "cite_note-:0-3)

The next step is to host the malicious content that the attacker wishes to distribute. One option is for the attacker to host the malicious content on their own server. However, because of the difficulty in directing users to a new page, it may also be hosted on a compromised legitimate website, or a legitimate website unknowingly distributing the attackers content through a [third party service](https://en.wikipedia.org/wiki/Third-party_software_component) (e.g. an advertisement). When the content is loaded by the client, the attacker will analyze the [fingerprint](https://en.wikipedia.org/wiki/Device_fingerprint) of the client in order to tailor the code to exploit vulnerabilities specific to that client. [[4]](https://en.wikipedia.org/wiki/Drive-by_download" \l "cite_note-:1-4)

Finally, the attacker exploits the necessary vulnerabilities to launch the drive-by download attack. Drive-by downloads usually use one of two strategies. The first strategy is exploiting [API](https://en.wikipedia.org/wiki/Application_programming_interface) calls for various [plugins](https://en.wikipedia.org/wiki/Plug-in_(computing)). For example, the DownloadAndInstall API of the Sina [ActiveX](https://en.wikipedia.org/wiki/ActiveX) component did not properly check its parameters and allowed the downloading and execution of arbitrary files from the internet. The second strategy involves writing [shellcode](https://en.wikipedia.org/wiki/Shellcode) to memory, and then exploiting vulnerabilities in the web browser or plugin to divert the control flow of the program to the shell code. [[4]](https://en.wikipedia.org/wiki/Drive-by_download" \l "cite_note-:1-4) After the shellcode has been executed, the attacker can perform further malicious activities. This often involves downloading and installing [malware](https://en.wikipedia.org/wiki/Malware), but can be anything, including stealing information to send back to the attacker.[[3]](https://en.wikipedia.org/wiki/Drive-by_download" \l "cite_note-:0-3)

The attacker may also take measures to prevent detection throughout the attack. One method is to rely on the [obfuscation](https://en.wikipedia.org/wiki/Obfuscation_(software)) of the malicious code. This can be done through the use of [IFrames](https://en.wikipedia.org/wiki/Iframes). [[3]](https://en.wikipedia.org/wiki/Drive-by_download" \l "cite_note-:0-3) Another method is to encrypt the malicious code to prevent detection. Generally the attacker encrypts the malicious code into a [ciphertext](https://en.wikipedia.org/wiki/Ciphertext), then includes the decryption method after the ciphertext.[[4]](https://en.wikipedia.org/wiki/Drive-by_download" \l "cite_note-:1-4)

## Detection

Detection of drive-by download attacks is an active area of research. Some methods of detection involve [anomaly detection](https://en.wikipedia.org/wiki/Anomaly_detection), which tracks for state changes on a user’s computer system while the user visits a webpage. This involves monitoring the user’s computer system for anomalous changes when a web page is rendered. Other methods of detection include detecting when malicious code (shellcode) is written to memory by an attacker’s exploit. Another detection method is to make run-time environments that allow [JavaScript](https://en.wikipedia.org/wiki/JavaScript) code to run and track its behavior while it runs. Other detection methods include examining contents of HTML pages to identify features that can be used to identify malicious web pages, and using characteristics of web servers to determine if a page is malicious.[[3]](https://en.wikipedia.org/wiki/Drive-by_download" \l "cite_note-:0-3) Some antivirus tools use static [signatures](software) to match patterns of malicious scripts, although these are not very effective because of obfuscation techniques. Detection is also possible by using low-interaction or high-interaction [honeyclients](honeypot).[[4]](https://en.wikipedia.org/wiki/Drive-by_download" \l "cite_note-:1-4)

Drive-by downloads can also be prevented from occurring by using script-blockers such as [NoScript](https://en.wikipedia.org/wiki/NoScript), which can easily be added into browsers such as Firefox. Using such a script-blocker, the user can disable all the scripts in a given webpage, and then selectively re-enable individual scripts on a one-by-one basis in order to determine which ones are truly necessary for webpage functionality. In this way, a whitelist of acceptable scripts can quickly be developed, which in turn facilitates convenient, safe, and efficient browsing of other websites. Such script-blocking also saves time and bandwidth (hence money) for the user, because scripts that load advertising (especially targeted ads) and invade the user's privacy (via tracking and profiling) no longer constitute a personal drain on the user and his/her resources.

**Exploit (computer security)**

An **exploit** (from the English verb *to exploit*, meaning "to use something to one’s own advantage") is a piece of [software](https://en.wikipedia.org/wiki/Software), a chunk of data, or a sequence of commands that takes advantage of a [bug](https://en.wikipedia.org/wiki/Software_bug) or [vulnerability](https://en.wikipedia.org/wiki/Vulnerability_(computer_science)) to cause unintended or unanticipated behavior to occur on computer software, hardware, or something electronic (usually computerized). Such behavior frequently includes things like gaining control of a computer system, allowing [privilege escalation](https://en.wikipedia.org/wiki/Privilege_escalation), or a [denial-of-service (DoS or related DDoS) attack](https://en.wikipedia.org/wiki/Denial-of-service_attack).

## Classification

## There are several methods of classifying exploits. The most common is by how the exploit communicates to the vulnerable software.A remote exploit[[1]](https://en.wikipedia.org/wiki/Exploit_(computer_security)" \l "cite_note-1) works over a network and exploits the security vulnerability without any prior access to the vulnerable system.

## A local exploit[[2]](https://en.wikipedia.org/wiki/Exploit_(computer_security)" \l "cite_note-2) requires prior access to the vulnerable system and usually increases the privileges of the person running the exploit past those granted by the system administrator. Exploits against client applications also exist, usually consisting of modified servers that send an exploit if accessed with a client application.

Exploits against client applications may also require some interaction with the user and thus may be used in combination with the [social engineering](https://en.wikipedia.org/wiki/Social_engineering_(security)) method. Another classification is by the action against the vulnerable system; unauthorized data access, arbitrary code execution, and denial of service are examples.

Many exploits are designed to provide superuser-level access to a computer system. However, it is also possible to use several exploits, first to gain low-level access, then to escalate privileges repeatedly until one reaches the highest administrative level (often called "root").

After an exploit is made known to the authors of the affected software, the vulnerability is often fixed through a patch and the exploit becomes unusable. That is the reason why some [black hat](https://en.wikipedia.org/wiki/Black_hat_hacking) [hackers](https://en.wikipedia.org/wiki/Hacker_(computer_security)) as well as military or intelligence agencies hackers do not publish their exploits but keep them private.

Exploits unknown to everyone but the people that found and developed them are referred to as [zero day](https://en.wikipedia.org/wiki/Zero-day_(computing)) exploits.

### Types

Exploitations are commonly categorized by the type of vulnerability they exploit (see [vulnerabilities](https://en.wikipedia.org/wiki/Vulnerability_(computing)) for a list), whether they are local/remote and the result of running the exploit (e.g. [EoP](https://en.wikipedia.org/wiki/Elevation_of_Privilege_(computing)), [DoS](https://en.wikipedia.org/wiki/Denial_of_Service_(computing)), [spoofing](https://en.wikipedia.org/wiki/Spoofing_attack)).

### Pivoting

Pivoting refers to a method used by [penetration testers](https://en.wikipedia.org/wiki/Penetration_test) that uses the compromised system to attack other systems on the same network to avoid restrictions such as [firewall](https://en.wikipedia.org/wiki/Firewall_(computing)) configurations, which may prohibit direct access to all machines. For example, if an attacker compromises a web server on a corporate network, the attacker can then use the compromised web server to attack other systems on the network. These types of attacks are often called multi-layered attacks. Pivoting is also known as island hopping.

Pivoting can further be distinguished into [proxy](https://en.wikipedia.org/wiki/Proxy_server) pivoting and [VPN](https://en.wikipedia.org/wiki/VPN) pivoting. Proxy pivoting generally describes the practice of channeling traffic through a compromised target using a proxy payload on the machine and launching attacks from the computer.[[5]](https://en.wikipedia.org/wiki/Exploit_(computer_security)" \l "cite_note-5) This type of pivoting is restricted to certain [TCP](https://en.wikipedia.org/wiki/Transmission_Control_Protocol) and [UDP](https://en.wikipedia.org/wiki/User_Datagram_Protocol) ports that are supported by the proxy.

VPN pivoting enables the attacker to create an encrypted layer to tunnel into the compromised machine to route any network traffic through that target machine, for example, to run a vulnerability scan on the internal network through the compromised machine, effectively giving the attacker full network access as if they were behind the firewall.

Typically, the proxy or VPN applications enabling pivoting are executed on the target computer as the [payload (software)](https://en.wikipedia.org/wiki/Payload_(software)) of an exploit.

Pivoting is usually done by infiltrating a part of a network infrastructure (as an example, a vulnerable printer or thermostat) and using a scanner to find other devices connected to attack them. By attacking a vulnerable piece of networking, an attacker could infect most or all of a network and gain complete control.

Flooders (DoS Client)

Flooding is a [Denial of Service](https://www.webopedia.com/TERM/D/DOS.html) (DoS) attack that is designed to bring a [network](https://www.webopedia.com/TERM/N/network.html) or service down by flooding it with large amounts of [traffic](https://www.webopedia.com/TERM/T/traffic.html). Flood attacks occur when a network or service becomes so weighed down with [packets](https://www.webopedia.com/TERM/P/packet.html) initiating incomplete connection requests that it can no longer process genuine connection requests. By flooding a [server](https://www.webopedia.com/TERM/S/server.html) or [host](https://www.webopedia.com/TERM/H/host.html) with connections that cannot be completed, the flood attack eventually fills the host’s [memory](https://www.webopedia.com/TERM/M/memory.html) [buffer](https://www.webopedia.com/TERM/B/buffer.html). Once this buffer is full no further connections can be made, and the result is a Denial of Service.

Keylogger:

A keylogger, sometimes called a keystroke logger or system monitor, is a type of surveillance technology used to monitor and record each keystroke typed on a specific computer's [keyboard](https://whatis.techtarget.com/definition/keyboard). Keylogger software is also available for use on smartphones, such as Apple's iPhone and Android devices.

Keyloggers are often used as a [spyware](https://searchsecurity.techtarget.com/definition/spyware) tool by cybercriminals to steal [personally identifiable information (PII)](https://searchfinancialsecurity.techtarget.com/definition/personally-identifiable-information), login credentials and sensitive enterprise data. Keylogger recorders may also be used by employers to observe employees' computer activities, parents to supervise their children's internet usage, users to track possible unauthorized activity on their devices or law enforcement agencies to analyze incidents involving computer use. These uses are considered ethical or appropriate in varying degrees.

### Types of keyloggers

A hardware-based keylogger is a small device that serves as a connector between the computer keyboard and the computer. The device is designed to resemble an ordinary keyboard [PS/2 connector](https://whatis.techtarget.com/definition/PS-2-connector), part of the computer cabling or a [USB](https://whatis.techtarget.com/definition/USB-Universal-Serial-Bus-USB-30-SuperSpeed-USB) adaptor, making it relatively easy for someone who wants to monitor a user's behavior to hide such a device.

Most workstation keyboards also plug into the back of the computer, keeping the connections out of the user's line of sight. A hardware keylogger may also come in the form of a module that is installed inside the keyboard itself. When the user types on the keyboard, the keylogger collects each keystroke and saves it as text in its own miniature [hard drive](https://searchstorage.techtarget.com/definition/hard-disk-drive), which may have a [memory](https://searchstorage.techtarget.com/definition/memory-card) capacity of up to several gigabytes. The person who installed the keylogger must later return and physically remove the device in order to access the information that has been gathered. There are also wireless keylogger [sniffers](https://searchnetworking.techtarget.com/definition/sniffer) that can intercept and decrypt data packets being transferred between a wireless keyboard and its receiver.

A keylogging software program Bottom of Form does not require physical access to the user's computer for installation. It can be downloaded on purpose by someone who wants to monitor activity on a particular computer, or it can be [malware](https://searchsecurity.techtarget.com/definition/malware) downloaded unwittingly and executed as part of a [rootkit](https://searchsecurity.techtarget.com/definition/rootkit) or [remote administration Trojan (RAT)](https://searchsecurity.techtarget.com/definition/RAT-remote-access-Trojan). The rootkit can launch and operate stealthily in order to evade manual detection or [antivirus](https://searchsecurity.techtarget.com/definition/antivirus-software) scans.

A common keylogger program typically consists of two files that get installed in the same directory: a [dynamic link library (DLL)](https://searchwindowsserver.techtarget.com/definition/dynamic-link-library-DLL) file that does all the recording and an [executable](https://searchsecurity.techtarget.com/definition/executable) file that installs the DLL file and triggers it to work. The keylogger program records each keystroke the user types and uploads the information over the internet periodically to whoever installed the program. There are many other ways that keylogging software can be designed to monitor keystrokes, including hooking keyboard APIs to another application, malicious script injection or memory injection.

Some keylogging programs may include functionality for recording user data besides keystrokes, such as capturing anything that has been copied to the clipboard and taking screenshots of the user's screen or a single application.

### Detection, prevention and removal

As there are various types of keyloggers that use different techniques, no single detection or removal method is considered the most effective.

Antikeylogger software is designed specifically to scan for software-based keyloggers, by comparing the files on a computer against a keylogger signature base or a checklist of common keylogger attributes. Using an antikeylogger can be more effective than using an antivirus or antispyware program, as the latter may identify a keylogger as a legitimate program instead of spyware.

Depending on the technique the antispyware application uses, it can possibly locate and disable keylogger software with lower privileges than it has. Use of a network monitor will ensure the user is notified each time an application tries to make a network connection, giving a security team the opportunity to stop any possible keylogger activity.[Application whitelisting](https://searchsecurity.techtarget.com/definition/application-whitelisting) can also be used to allow only documented, authorized programs to run on a system.

While visual inspection can be used to identify hardware keyloggers, it is impractical and time-consuming to implement on a large scale. System cages that prevent access to or tampering with USB and PS/2 ports can be added to the user's desktop setup. Extra precautions include using a [security token](https://searchsecurity.techtarget.com/definition/security-token) as part of t[wo-factor authentication (2FA)](https://searchsecurity.techtarget.com/definition/two-factor-authentication) to ensure an attacker cannot use a stolen password alone to log in to a user's account, or using an [onscreen keyboard](https://whatis.techtarget.com/definition/soft-keyboard-onscreen-keyboard-or-software-keyboard) and [voice-to-text](https://searchunifiedcommunications.techtarget.com/definition/voice-to-text) software to circumvent using a physical keyboard.

A keylogger, sometimes called a keystroke logger or system monitor, is a type of surveillance technology used to monitor and record each keystroke typed on a specific computer's [keyboard](https://whatis.techtarget.com/definition/keyboard). Keylogger software is also available for use on smartphones, such as Apple's iPhone and Android devices.

Keyloggers are often used as a [spyware](https://searchsecurity.techtarget.com/definition/spyware) tool by cybercriminals to steal [personally identifiable information (PII)](https://searchfinancialsecurity.techtarget.com/definition/personally-identifiable-information), login credentials and sensitive enterprise data. Keylogger recorders may also be used by employers to observe employees' computer activities, parents to supervise their children's internet usage, users to track possible unauthorized activity on their devices or law enforcement agencies to analyze incidents involving computer use. These uses are considered ethical or appropriate in varying degrees.

Logic bomb:

A logic bomb is a piece of code intentionally inserted into a software system that will set off a malicious function when specified conditions are met. For example, a programmer may hide a piece of code that starts deleting files (such as a salary database trigger), should he or she ever be terminated from the company.

Software that is inherently malicious, such as viruses and worms, often contain logic bombs that execute a certain payload at a pre-defined time or when some other condition is met. This technique can be used by a virus or worm to gain momentum and spread before being noticed. Some viruses attack their host systems on specific dates, such as April Fools' Day.Trojans and other computer viruses that activate on certain dates are often called "time bombs".

To be considered a logic bomb, the payload should be unwanted and unknown to the user of the software. As an example, trial programs with code that disables certain functionality after a set time are not normally regarded as logic bombs.

## Successful logic bombs

* In June 2006 Roger Duronio, a [system administrator](https://en.wikipedia.org/wiki/System_administrator) for [UBS](https://en.wikipedia.org/wiki/UBS_AG), was charged with using a logic bomb to damage the company's computer network, and with securities fraud for his failed plan to drive down the company's stock with activation of the logic bomb. Duronio was later convicted and sentenced to 8 years and 1 month in prison, as well as a $3.1 million restitution to UBS.[[5]](https://en.wikipedia.org/wiki/Logic_bomb" \l "cite_note-5)
* On 20 March 2013, in an attack launched against [South Korea](https://en.wikipedia.org/wiki/South_Korea), a logic bomb struck machines and "wiped the hard drives and master boot records of at least three banks and two media companies simultaneously."[Symantec](https://en.wikipedia.org/wiki/Symantec) reported that the malware also contained a component that was capable of wiping Linux machines.
* On 19 July 2019, David Tinley, a contract employee, pleaded guilty for programming logic bombs within the software he created for Siemens Corporation.The software was intentionally made to malfunction after a certain amount of time, requiring the company to hire him to fix it for a fee. The logic bombs went undetected for two years, but was then discovered while he was out of town and had to hand over the administrative password to his software.[[11]](https://en.wikipedia.org/wiki/Logic_bomb" \l "cite_note-11)

## Attempted logic bombs

* In February 2000, Tony Xiaotong, indicted before a [grand jury](https://en.wikipedia.org/wiki/Grand_jury), was accused of planting a logic bomb during his employment as a programmer and securities trader at [Deutsche Morgan Grenfell](https://en.wikipedia.org/wiki/Morgan,_Grenfell_%26_Co.). The bomb, planted in 1996, had a trigger date of 20 July 2000, but was discovered by other programmers in the company. Removing and cleaning up after the bomb allegedly took several months.[[12]](https://en.wikipedia.org/wiki/Logic_bomb" \l "cite_note-12)
* On 2 October 2003 Yung-Hsun Lin, also known as Andy Lin, changed code on a server at [Medco Health Solutions](https://en.wikipedia.org/wiki/Medco_Health_Solutions) Inc.'s [Fair Lawn, New Jersey](https://en.wikipedia.org/wiki/Fair_Lawn,_New_Jersey) headquarters, where he was employed as a Unix administrator, creating a logic bomb set to go off on his birthday in 2004. It failed to work due to a programming error, so Lin corrected the error and reset it to go off on his next birthday, but it was discovered and disabled by a Medco computer systems administrator a few months before the trigger date. Lin pleaded guilty and was sentenced to 30 months in jail in a federal prison in addition to $81,200 in [restitution](https://en.wikipedia.org/wiki/Restitution). The charges held a maximum sentence of 10 years and a fine of US$250,000.[[13]](https://en.wikipedia.org/wiki/Logic_bomb" \l "cite_note-13)[[14]](https://en.wikipedia.org/wiki/Logic_bomb" \l "cite_note-14)
* On 29 October 2008 a logic bomb was discovered at American mortgage giant [Fannie Mae](https://en.wikipedia.org/wiki/Fannie_Mae). The bomb was planted by Rajendrasinh Babubhai Makwana, an IT contractor who worked at Fannie Mae's [Urbana, Maryland](https://en.wikipedia.org/wiki/Urbana,_Maryland) facility. The bomb was set to activate on 31 January 2009 and could have wiped all of Fannie Mae's 4000 servers. Makwana had been terminated around 1:00pm on 24 October 2008 and managed to plant the bomb before his network access was revoked. Makwana was indicted in a Maryland court on 27 January 2009 for unauthorized computer access,[[15]](https://en.wikipedia.org/wiki/Logic_bomb" \l "cite_note-15)[[16]](https://en.wikipedia.org/wiki/Logic_bomb" \l "cite_note-16) convicted on 4 October 2010, and sentenced to 41 months in prison on 17 December 2010.[[17]](https://en.wikipedia.org/wiki/Logic_bomb" \l "cite_note-17)
* In October 2009, Douglas Duchak was terminated from his job as data analyst at the Colorado Springs Operations Center (CSOC) of the U.S. [Transportation Security Administration](https://en.wikipedia.org/wiki/Transportation_Security_Administration). Surveillance cameras captured images of Duchak entering the facility after hours and loading a logic bomb onto a CSOC server that stored data from the U.S. Marshals. In January 2011, Duchak was sentenced to two years in prison, $60,587 in fines, and three years on probation.[[18]](https://en.wikipedia.org/wiki/Logic_bomb" \l "cite_note-18) At his sentencing, Duchak tearfully apologized as his lawyer noted that at the time of the incident, Duchak's wife was pregnant with their second child. The judge at the sentencing mentioned that this logic bomb planting "incident was an anomaly in an otherwise untarnished work history.

Macro virus

A macro virus is a computer virus written in the same macro language used for software programs, including Microsoft Excel or word processors such as Microsoft Word. When a macro virus infects a software application, it causes a sequence of actions to begin automatically when the application is opened.

Since a macro virus centers on an application and not an operating system, it typically can infect any computer running any operating system, even those running MacOS and Linux.

### How macro viruses work

Macro viruses work by embedding malicious code in the macros that are associated with documents, spreadsheets and other data files, causing the malicious programs to run as soon as the documents are opened. Typically, macro [malware](https://searchsecurity.techtarget.com/definition/malware) is transmitted through [phishing emails](https://searchsecurity.techtarget.com/definition/phishing) containing malicious attachments. The macro virus spreads quickly as users share infected documents.

Once an infected macro is executed, it will typically infect every other document on a user's computer. Some macro viruses cause irregularities in text documents, such as inserting or deleting words. Other macro malware accesses email accounts and sends out copies of infected files to all of the users' contacts, who then open and access these files because they come from trusted sources.

### Examples of macro viruses

The first macro virus to be spread in the wild through Microsoft Word was Concept, which was discovered in July 1995. The virus was accidentally included on a CD-ROM called "Microsoft Compatibility Test" shipped by Microsoft to hundreds of corporations in August 1995.

Concept spread to computers in infected Word documents with a .doc extension attached to email messages. From there, the virus infected the English version of Word 6.0 or Word 95 documents that had been saved using the command Save As. Concept did not carry out any damaging actions in affected computers; it just displayed a message on screen when it infected a document.

Another classic example of a macro virus was the [Melissa virus](https://searchsecurity.techtarget.com/definition/Melissa-virus) first found in March 1999. Melissa was a macro virus that was distributed as an email attachment and spread quickly across the globe. The subject of the email indicated that the message contained a file that the user had requested. When the user opened the attachment, the virus infected the user's computer and spread to other email messages using macros in Microsoft Word 97, Microsoft Word 2000 files, as well as Microsoft Excel and Outlook.

Although Melissa didn't destroy files or other resources, it caused Microsoft to shut down all incoming email on March 26, 1999, and it disrupted over 1 million email accounts worldwide and cost businesses an estimated $80 million.

Discovered in September 1995, the Nuclear macro virus was similar to Concept, but because the malicious macros in Nuclear were designated ExecuteOnly, they were encrypted by Word and couldn't be viewed or edited, though they were visible in the macro list. The message carried by the Nuclear virus was only displayed on the last page of a document when it was printed, but only if it was printed during the last four seconds of any minute.

The message that the Nuclear macro virus added to documents was, "And finally I would like to say: STOP ALL FRENCH NUCLEAR TESTING IN THE PACIFIC." This virus would also cause error messages to appear when a user selected from File > Print or File > Save As.

First observed in 2014, Hancitor (also known as Chanitor) was a macro-based malware downloader hidden in Word documents that were delivered via phishing email. The main purpose of Hancitor was to download malicious [payloads](https://searchsecurity.techtarget.com/definition/payload) such as banking [Trojans](https://searchsecurity.techtarget.com/definition/Trojan-horse) and ransomware on contaminated machines.

The Dridex banking malware and Locky [ransomware](https://searchsecurity.techtarget.com/definition/ransomware) also made use of malicious macros to hijack systems. And the Rovnix Trojan used macros embedded in Microsoft Word documents to infect computers and steal data. Vawtrak was another Trojan that was distributed to victims in malicious Word documents and could take screenshots, hijack webcams and log keystrokes. Rovnix and Vawtrak typically targeted financial organizations.

Until recently, most macro viruses targeted the Windows operating system, but in early 2017 MacDownloader, the first Word macro virus for Apple's MacOS, was found in the wild. [Hackers](https://searchsecurity.techtarget.com/definition/hacker) were using malicious macros in Word documents to install malware on Mac computers to steal users' data. Once the malware is installed, hackers can access browser history logs, monitor webcams, as well as steal password and encryption keys.

One malicious Word file carrying the MacDownloader macro virus was titled "U.S. Allies and Rivals Digest Trump's Victory -- Carnegie Endowment for International Peace.docm."

### Preventing macro viruses

Since macro viruses are usually spread in application files that have been shared across the internet by email, especially in phishing email, macro virus defenses include strategies for scanning inbound email attachments, preventing users from opening dubious files and preventing macros from running at all when documents are opened.

Some techniques for preventing the spread of macro viruses include:

* Use a spam filter. The fewer phishing email messages users get in their inboxes, the less chance their computers will be infected by malware.
* Use a strong antivirus program. Antivirus software will warn users when they attempt to download suspicious files or open harmful links.
* Ensure that computers are running current software versions and that all security patches are installed.
* Do not open attachments from unknown senders.
* Do not open attachments in suspicious email messages, even when they appear to be from known senders.
* Activate the macro security function on Microsoft Word and Excel, and exercise extreme caution about enabling macros.
* Disable macro scripts entirely.

### Removing a macro virus

It is important to remove all files infected with a macro virus to prevent it from spreading.

The first step in removing macro malware is to reboot the infected computer in [Safe Mode](https://whatis.techtarget.com/definition/safe-mode).

Deleting all temporary files will help speed up virus scanning, as well as freeing up disk space and removing any malware-infected temp files.

Finally, do a virus scan of the infected computer. If a real-time antivirus program is already running on the machine, use a different, on-demand scanner to run a macro malware check because the running antivirus program may not have detected the malware. In this case, use the on-demand scanner first followed by a full scan using the real-time antivirus. This should detect and quarantine any macro malware found on the system.

Malicious Mobile Code :

Malicious Mobile Code: What You Need to Know. Author: Eric Everson, MBA, MSIT-SE

The thought of someone hacking into your mobile phone to steal your personal data added to the growing number of mobile threats sounds bad enough, then you come across the industry term “Malicious Mobile Code” and it makes downloading any new mobile app a scary process.

So it sounds like scary stuff, but what is Malicious Mobile Code (MMC) REALLY? If you follow my journal, you know that I’m always knuckle-deep in this kind of stuff, and as a result I’ve lost many good computers and mobile handsets along the way. As threatening as the words may sound, MMC is really an industry catchall phrase that refers to any code that can hinder the operation of a mobile application or device.

Building software is kind of like building a house of cards in that each layer depends on the next to function properly. In software (just as in the house of cards) if you remove or otherwise tamper with a key component it can often corrupt the entire structure. MMC most often attempts to do this very thing by injecting faulty code into a key operating component of your mobile software or Mobile Operating System (MOPS).

Though mobile devices are steadily becoming more sophisticated with added computing power, the reality is that MOPS remain highly vulnerable to such MMC attacks. This is why third-party mobile security software is becoming so important to have on your mobile device. Many of the mobile security solutions on the market today block the MMC similar to antivirus software for a computer. Additionally, the demand for mobile app-driven handsets is significantly on the rise which is putting many users at greater risk.

Often consumers on the most popular app retail portals assume that anything they download to their handsets should be safe. Despite best efforts however, many risky apps from those with harmful embedded source code to those masquerading as legitimate financial services apps are making their way to unsuspecting mobile users.

This issue has created new demand for services like MyMobiSafe Verified, the first service of its kind that offers a formal review and validation of new mobile apps across every platform (iPhone, Android, BlackBerry, Symbian, Java, Orange, and all others). By creating an environment where developers and the mobile community alike are looking for the confidence of the MyMobiSafe Verified mark, this creates a significant hurdle for unwanted Malicious Mobile Code in the market.

MMC can range from the simplest corrupt code to the worst mobile viruses, yet the phrase and acronym remains as an industry catch all. As a software engineer and one with substantial frontline experience with this kind of code, my words of wisdom are to be cautious of anything that you are loading onto your handset. If it is free, remember that old adage that suggests “nothing good comes free.” In too many cases of mobile apps, free means that there is something else behind the curtains. Start looking for verified apps before you buy them as they will often display an industry-wide recognizable logo. Finally, remember that not all MMC is created equal, in many cases damage is not permanent and can often be repaired by a professional.

Rootkit

### What Is a Rootkit?

A rootkit is a clandestine computer program designed to provide continued privileged access to a computer while actively hiding its presence. The term rootkit is a connection of the two words "root" and "kit." Originally, a rootkit was a collection of tools that enabled administrator-level access to a computer or network. Root refers to the Admin account on Unix and Linux systems, and kit refers to the software components that implement the tool. Today rootkits are generally associated with malware – such as Trojans, [worms](https://www.veracode.com/security/computer-worm), viruses – that conceal their existence and actions from users and other system processes.

What Can a Rootkit Do?

A rootkit allows someone to maintain command and control over a computer without the computer user/owner knowing about it. Once a rootkit has been installed, the controller of the rootkit has the ability to remotely execute files and change system configurations on the host machine. A rootkit on an infected computer can also access log files and spy on the legitimate computer owner’s usage.

Rootkit Detection

It is difficult to detect rootkits. There are no commercial products available that can find and remove all known and unknown rootkits. There are various ways to look for a rootkit on an infected machine. Detection methods include behavioral-based methods (e.g., looking for strange behavior on a computer system), signature scanning and memory dump analysis. Often, the only option to remove a rootkit is to completely rebuild the compromised system.

Rootkit Protection

Many rootkits penetrate computer systems by piggybacking with software you trust or with a virus. You can safeguard your system from rootkits by ensuring it is kept patched against known vulnerabilities. This includes patches of your OS, applications and up-to-date virus definitions. Don't accept files or open email file attachments from unknown sources. Be careful when installing software and carefully read the end-user license agreements.

[Static analysis](https://www.veracode.com/products/binary-static-analysis-sast-1) can detect backdoors and other malicious insertions such as rootkits. Enterprise developers as well as IT departments buying ready-made software can scan their applications to detect threats including "special" and "hidden-credential" backdoors.

Well-Known Rootkit Examples

Lane Davis and Steven Dake - wrote the earliest known rootkit in the early 1990s.

NTRootkit – one of the first malicious rootkits targeted at Windows OS.

HackerDefender – this early Trojan altered/augmented the OS at a very low level of functions calls.

Machiavelli - the first rootkit targeting Mac OS X appeared in 2009. This rootkit creates hidden system calls and kernel threads.

Greek wiretapping – in 2004/05, intruders installed a rootkit that targeted Ericsson's AXE PBX.

Zeus, first identified in July 2007, is a Trojan horse that steals banking information by man-in-the-browser keystroke logging and form grabbing.

Stuxnet - the first known rootkit for industrial control systems

Flame - a computer malware discovered in 2012 that attacks computers running Windows OS. It can record audio, screenshots, keyboard activity and network traffic.

Ransomware:

History of the Ransomware

According to Becker's Hospital Review, the first known ransomware attack occurred in 1989 and targeted the healthcare industry. 28 years later, the healthcare industry remains a top target for ransomware attacks. PC CYBORG advisory from 1989. ... This ransomware attack became known as the AIDS Trojan, or the PC Cyborg

Author of Ransomware

Encrypting ransomware. The first known malware extortionattack, the "AIDS Trojan" written by Joseph Popp in 1989, had a design failure so severe it was not necessary to pay the extortionist at all.

How does the malware the works?

Most ransomware is delivered via email that appears to be legitimate, enticing you to click a link or download an attachment that delivers the malicious software. Ransomware is also delivered via drive-by-download attacks on compromised or malicious websites. Some ransomware attacks have even been sent using social media messaging.Generic ransomware is rarely individually targeted, but rather a “shotgun” approach where attackers acquire lists of emails or compromised websites and blast out ransomware. Given the number of attackers out there, it will be likely that if you get hit multiple times, it will be by a different attacker. Whether or not the ransom is paid, keep in mind that attackers will always try extracting useful data from a compromised machine. Assume all sensitive data on the machine was compromised, which could include usernames & passwords for internal or web resources, payment information, email addresses of contacts, and more.

Types of ransomware

There are three main types of ransomware, ranging in severity from mildly off-putting to Cuban Missile Crisis dangerous.

Scareware Scareware, as it turns out, is not that scary. It includes rogue security software and tech support scams. You might receive a pop-up message claiming that malware was discovered and the only way to get rid of it is to pay up. If you do nothing, you’ll likely continue to be bombarded with pop-ups, but your files are essentially safe. A legitimate cybersecurity software program would not solicit customers in this way. If you don’t already have this company’s software on your computer, then they would not be monitoring you for ransomware infection. If you do have security software, you wouldn’t need to pay to have the infection removed—you’ve already paid for the software to do that very job.

Screen lockers: Upgrade to terror alert orange for these guys. When lock-screen ransomware gets on your computer, it means you’re frozen out of your PC entirely. Upon starting up your computer, a full-size window will appear, often accompanied by an official-looking FBI or US Department of Justice seal saying illegal activity has been detected on your computer and you must pay a fine. However, the FBI would not freeze you out of your computer or demand payment for illegal activity. If they suspected you of piracy, child pornography, or other cybercrimes, they would go through the appropriate legal channels.

Encrypting ransomware : This is the truly nasty stuff. These are the guys who snatch up your files and encrypt them, demanding payment in order to decrypt and redeliver. The reason why this type of ransomware is so dangerous is because once cybercriminals get ahold of your files, no security software or system restore can return them to you. Unless you pay the ransom—for the most part, they’re gone. And even if you do pay up, there’s no guarantee the cybercriminals will give you those files back.

Payload

History of the payload

A payload refers to the component of a computer virus that executes a malicious activity. Apart from the speed in which a virus spreads, the threat level of a virus is calculated by the damages it causes. Viruses with more powerful payloads tend to be more harmful.

Although not all viruses carry a payload, a few payloads are considered extremely dangerous. Some of the examples of payloads are data destruction, offensive messages and the delivery of spam emails through the infected user's account.

How do malicious payloads harm their victims?

Some typical examples of the way malicious payloads cause damage:

• Data theft: Particularly common is the theft of sensitive information such as login credentials or financial information through various forms of data breaches.

• Activity monitoring: An executed malicious payload may serve to monitor user activity on a computer, this can be done for the purposes of spying, blackmail, or to aggregate consumer behavior which can be sold to advertisers.

• Displaying advertisements: Some malicious payloads work to display persistent, unwanted ads such as pop-ups and pop-unders to the victim.

• Deleting or modifying files: This is one of the most serious consequences to arise from a malicious payload. Files can be deleted or modified to either affect the behavior of a computer, or even disable the operating system and/or startup processes. For example some malicious payloads are designed to ‘brick’ smartphones, meaning they can no longer be turned on or used in any way.

• Downloading new files: Some malicious payloads come in very lightweight files that are easy to distribute, but once executed they will trigger the download of a much larger piece of malicious software.

• Running background processes: A malicious payload can also be triggered to quietly run processes in the background, such as cryptocurrency mining or data storage.

DOS

History

DOS was initially released on the IBM System360 and later used by other platforms independently. These were not the same operating system but instead meant they were for use with disks, often booting from the disk. ... Digital Research founder Gary Kildall refused, and IBM withdrew

Denial-of-Service (DoS) attack is an attack meant to shut down a machine or network, making it inaccessible to its intended users. DoS attacks accomplish this by flooding the target with traffic, or sending it information that triggers a crash. In both instances, the DoS attack deprives legitimate users (i.e. employees, members, or account holders) of the service or resource they expected.

Victims of DoS attacks often target web servers of high-profile organizations such as banking, commerce, and media companies, or government and trade organizations. Though DoS attacks do not typically result in the theft or loss of significant information or other assets, they can cost the victim a great deal of time and money to handle.

There are two general methods of DoS attacks: flooding services or crashing services. Flood attacks occur when the system receives too much traffic for the server to buffer, causing them to slow down and eventually stop.

Web shells

A web-shell is a malicious script used by an attacker with the intent to escalate and maintain persistent access on an already compromised web application. A web-shell itself cannot attack or exploit a remote vulnerability, so it is always the second step of an attack.An attacker can take advantage of common vulnerabilities such as SQL injection, remote file inclusion (RFI), FTP, or even use cross-site scripting (XSS) as part of a social engineering attack in order to upload the malicious script. The common functionality includes but is not limited to shell command execution, code execution, database enumeration and file management.

Persistent Remote Access

A web-shell usually contains a backdoor which allows an attacker to remotely access and possibly, control a server at any time. This would save the attacker the inconvenience of having to exploit a vulnerability each time access to the compromised server is required.

An attacker might also choose to fix the vulnerability themselves, in order to ensure that no one else will exploit that vulnerability. This way the attacker can keep a low-profile and avoid any interaction with an administrator, while still obtaining the same result.

It is also worth mentioning that several popular web shells use password authentication and other techniques to ensure that only the attacker uploading the web-shell has access to it. Such techniques include locking down the script to a specific custom HTTP header, specific cookie values, specific IP addresses, or a combination of these techniques. Most web shells also contain code to identify and block search engines from listing the shell and, as a consequence, blacklisting the domain or server the web application is hosted on – in other words, stealth is key.

Privilege Escalation

Unless a server is misconfigured, the web shell will be running under the web server’s user permissions, which are (or, at least, should be) limited. Using a web-shell, an attacker can attempt to perform privilege escalation attacks by exploiting local vulnerabilities on the system in order to assume root privileges, which, in Linux and other UNIX-based operating systems is the ‘super-user’.

With access to the root account, the attacker can essentially do anything on the system including installing software, changing permissions, adding and removing users, stealing passwords, reading emails and more.

Pivoting and Launching Attacks

A web-shell can be used for pivoting inside or outside a network. The attacker might want to monitor (sniff) the network traffic on the system, scan the internal network to discover live hosts, and enumerate firewalls and routers within the network.

This process can take days, even months, predominantly because an attacker typically seeks to keep a low profile, and draw the least amount of attention possible. Once an attacker has persistent access, they can patiently make their moves.

The compromised system can also be used to attack or scan targets that reside outside the network. This adds an additional layer of “anonymity” to the attacker since they are using a 3rd party system to launch an attack. A step further would be to pivot (tunnel) through multiple systems to make it almost impossible to trace an attack back to its source.

Zombie

Another use of web-shells is to make servers part of a botnet. A botnet is a network of compromised systems that an attacker would control, either to use themselves, or to lease to other criminals. The web-shell or backdoor is connected to a command and control (C&C) server from which it can take commands on what instructions to execute.

This setup is commonly used in distributed-denial-of-service (DDoS) attacks, which require expansive amounts of bandwidth. In this case, the attacker does not have any interest in harming, or stealing anything off-of the system upon which the web shell was deployed. Instead, they will simply use its resources for whenever is needed.

Phishing

Phishing is a cyber attack that uses disguised email as a weapon. The goal is to trick the email recipient into believing that the message is something they want or need — a request from their bank, for instance, or a note from someone in their company — and to click a link or download an attachment.

What really distinguishes phishing is the form the message takes: the attackers masquerade as a trusted entity of some kind, often a real or plausibly real person, or a company the victim might do business with. It's one of the oldest types of cyberattacks, dating back to the 1990s, and it's still one of the most widespread and pernicious, with phishing messages and techniques becoming increasingly sophisticated.

What is a phishing kit?

The availability of phishing kits makes it easy for cyber criminals, even those with minimal technical skills, to launch phishing campaigns. A phishing kit bundles phishing website resources and tools that need only be installed on a server. Once installed, all the attacker needs to do is send out emails to potential victims. Phishing kits as well as mailing lists are available on the dark web. A couple of sites, Phishtank and OpenPhish, keep crowd-sourced lists of known phishing kits.

The Duo Labs report, Phish in a Barrel, includes an analysis of phishing kit reuse. Of the 3,200 phishing kits that Duo discovered, 900 (27 percent) were found on more than one host. That number might actually be higher, however. “Why don’t we see a higher percentage of kit reuse? Perhaps because we were measuring based on the SHA1 hash of the kit contents. A single change to just one file in the kit would appear as two separate kits even when they are otherwise identical,” said Jordan Wright, a senior R&D engineer at Duo and the report’s author.

Types of phishing

If there's a common denominator among phishing attacks, it's the disguise. The attackers spoof their email address so it looks like it's coming from someone else, set up fake websites that look like ones the victim trusts, and use foreign character sets to disguise URLs.

That said, there are a variety of techniques that fall under the umbrella of phishing. There are a couple of different ways to break attacks down into categories. One is by the purpose of the phishing attempt. Generally, a phishing campaign tries to get the victim to do one of two things:

• Hand over sensitive information. These messages aim to trick the user into revealing important data — often a username and password that the attacker can use to breach a system or account. The classic version of this scam involves sending out an email tailored to look like a message from a major bank; by spamming out the message to millions of people, the attackers ensure that at least some of the recipients will be customers of that bank. The victim clicks on a link in the message and is taken to a malicious site designed to resemble the bank's webpage, and then hopefully enters their username and password. The attacker can now access the victim's account.

• Download malware. Like a lot of spam, these types of phishing emails aim to get the victim to infect their own computer with malware. Often the messages are "soft targeted" — they might be sent to an HR staffer with an attachment that purports to be a job seeker's resume, for instance. These attachments are often .zip files, or Microsoft Office documents with malicious embedded code. The most common form of malicious code is ransomware — in 2017 it was estimated that 93 percent of phishing emails contained ransomware attachments. There are also several different ways that phishing emails can be targeted. As we noted, sometimes they aren't targeted at all; emails are sent to millions of potential victims to try to trick them into logging in to fake versions of very popular websites. Vade Secure has tallied the most popular brands that hackers use in their phishing attempts (see infographic below). Other times, attackers might send "soft targeted" emails at someone playing a particular role in an organization, even if they don't know anything about them personally.

But some phishing attacks aim to get login information from, or infect the computers of, specific people. Attackers dedicate much more energy to tricking those victims, who have been selected because the potential rewards are quite high.

Advanced persistent threat (APT)

An advanced persistent threat (APT) is a prolonged and targeted cyberattack in which an intruder gains access to a network and remains undetected for an period of time. The intention of an APT attack is usually to monitor network activity and steal data rather than to cause damage to the network or organization. APT attacks typically target organizations in sectors such as national defense, manufacturing and the financial industry, as those companies deal with high-value information, including intellectual property, military plans, and other data from governments and enterprise organizations The goal of most APT attacks is to achieve and maintain ongoing access to the targeted network rather than to get in and out as quickly as possible. Because a great deal of effort and resources usually go into carrying out APT attacks, hackers typically target high-value targets, such as nation-states and large corporations, with the ultimate goal of stealing information over a long period of time.

To gain access, APT groups often use advanced attack methods, including advanced exploits of zero-day vulnerabilities, as well as highly-targeted spear phishing and other social engineering techniques. To maintain access to the targeted network without being discovered, threat actors use advanced methods, including continuously rewriting malicious code to avoid detection and other sophisticated evasion techniques. Some APTs are so complex that they require full-time administrators to maintain the compromised systems and software in the targeted network

How an APT attack works

Attackers executing APTs typically take the following sequential approach to gain and maintain ongoing access to a target:

• Gain access: APT groups gain access to a target by targeting systems through the internet, via spear phishing emails or via an application vulnerability with the intention of leveraging any access by inserting malicious software into the target.

• Establish a foothold: After gaining access to the target, threat actors use their access to do further reconnaissance, as well as to begin exploiting the malware they've installed to create networks of backdoors and tunnels that they can use to move around unnoticed. APTs may use advanced malware techniques such as code rewriting to cover their tracks.

• Gain even greater access: Once inside the targeted network, APT actors may use such methods as password cracking to gain administrative rights so they can control more of the system and get even deeper levels of access.

• Move laterally: Once threat actors have breached their target systems, including gaining administrator rights, they can then move around the enterprise network at will. Additionally, they can attempt to access other servers, as well as other secure areas of the network.

• Stage the attack: At this point, the hackers centralize, encrypt and compress the data so they can exfiltrate it.

• Take the data: The attackers harvest the data and transfer it to their own system.

• Remain until they're detected: The cybercriminals can repeat this process for long periods of time until they're detected, or they can create a backdoor so they can access the system again at some point

Zombie, Bot

News about internet crimes often mentions "bots", "zombies", and "botnets". It's not hard to figure out from the context that these are computer or network security threats. But what exactly are they, how do they work, and what kind of damage can they cause?

A bot, short for "robot", is a type of software application or script that performs automated tasks on command. Bad bots perform malicious tasks that allow an attacker to remotely take control over an affected computer. Once infected, these machines may also be referred to as zombies.

Although taking over one computer is useful, the real value to a criminal comes from collecting huge numbers of zombie computers and networking them so they can all be controlled at once to perform large-scale malicious acts. This type of network is known as a "botnet"

How Botnets can impact you?

Often, the cyber criminal will seek to infect and control thousands, tens of thousands or even millions of computers – so that the cyber criminal can act as the master of a large ‘zombie network’ – or ‘bot-network’ – that is capable of delivering a Distributed Denial of Service (DDoS) attack, a large-scale spam campaign or other types of cyber attack.

In some cases, cyber criminals will establish a large network of zombie machines and then sell access to the zombie network to other criminals – either on a rental basis or as an outright sale. Spammers may rent or buy a network in order to operate a large-scale spam campaign.

Worms

A worm virus is a malicious, self-replicating program that can spread throughout a network without human assistance.

Worms cause damage similar to viruses, exploiting holes in security software and potentially stealing sensitive information, corrupting files and installing a back door for remote access to the system, among other issues. Worms often utilize large amounts of memory and bandwidth, so affected servers, networks and individual systems are often overloaded and stop responding. But worms are not viruses. Viruses need a host computer or operating system. The worm program operates alone. The worm is often transmitted via file-sharing networks, information-transport features, email attachments or by clicking links to malicious websites. Once downloaded, the worm takes advantage of a weakness in its target system or tricks a user into executing it. Some worms have a phishing component that entices users to run the malicious code. Internet worms are often designed to exploit new security issues, and search for systems that haven’t installed current software or operating system security updates.

How do computer worms work?

Worms can be transmitted by exploiting software vulnerabilities or computer worms could arrive as attachments in spam emails. Once opened, these files could provide a link to a malicious website or automatically download the computer worm. Once it’s installed, the worm silently goes to work and infects the machine without the user’s knowledge.

Worms can modify and delete files, and they can even inject additional malicious software onto a computer. Sometimes a computer worm’s purpose is only to make copies of itself over and over — depleting system resources, such as hard drive space or bandwidth, by overloading a shared network. In addition to wreaking havoc on a computer’s resources, worms can also steal data, install a backdoor, and allow a hacker to gain control over a computer and its system settings. Computer worms can exploit network configuration errors (for example, to copy themselves onto a fully accessible disk) or exploit loopholes in operating system and application security. Many worms will use more than one method in order to spread copies via networks.

Spammer Program

XRumer is a piece of software made for spamming online forums and comment sections. The program is able to bypass security techniques commonly used by many forums and blogs to deter automated spam, such as account registration, client detection, many forms of CAPTCHAs, and e-mail activation before posting.

In addition, the software can avoid the suspicions of forumadministrators by first registering to make a post in the form of a question which mentions the spam product before registering another account to post a spam link which mentions the product. The side effect of these innocent-looking posts is that helpful forum visitors may search on a search engine (e.g. Google) for the product and hemselves post a link to help out, thus bolstering the product's Google ranking without falling afoul of forum posting policies. The software is also capable of avoiding detection by making posts in off-topic, spam and overflow sections of forums thus attempting to keep its activities in high activity low content areas of the targeted forum. However, there are other platforms used to spam to which includes website comment spam.