

MALWARE & VIRUSES

Originally designed to “harmlessly” track user browsing habits, malware has quickly become as intrusive and damaging as even the worst of viruses. Left unchecked and unmanaged over time, malware can adversely effect computer performance, pilfer your most sensitive data and even cripple entire networks – much like viruses.

Your best defense against malware is the diligent and persistent use of various malware scanners and firewalls. Often times, more than one scanner will need to be utilized to achieve satisfactory results.

For example, even when Windows Defender (which has proven to be rather “buggy”) is deployed – running a malware scan using Lavasoft’s AdAware will, more often than not, detect malware that managed to go undetected. Same holds true for Spybot and other malware scanners.

Then, just when you thought you had a handle on things – you learn that what you thought was an anti-malware or anti-virus program turns out to actually be malware and/or a virus – much like the prolific and unshakable “AntiVirus2009”.

Most “free” scanners are designed to detect and delete malware only after it is already on your PC – paying the annual \$30-\$60 upgrade fees will enable the “shields” designed to prevent malware from actually getting onto your PC.

Viruses are designed to deceive, steal, damage and cripple computers of any kind and can only be thwarted with fervent diligence and persistence. Forget to renew your subscription, update your virus pattern or temporarily disable your shield and you’ll be putting yourself at great risk.

Once you get a virus, installing anti-virus software will do little to help. You’ll need to employ the use of a remote virus scan to eradicate any viruses before you can even get the new virus software installed. Because it is difficult and not advised to run more than one anti-virus application, one local virus application combined with frequent remote scans are your best insurance policy.

That was my simple overview of malware and viruses, which can be attributed to approximately 15%-20% of the service calls I perform. An extensive library of information is available on the following pages.

WRITTEN BY: Marcum Patrick – Montrose Data Protection & Recovery Solutions - 970.209.3967

www.970data.com

info@970data.com

MDPRSrev120908

This information can be seen at Wikipedia – specifically <http://en.wikipedia.org/wiki/Malware>

Malware, a portmanteau from the words malicious and software, is software designed to infiltrate or damage a computer system without the owner's informed consent. The expression is a general term used by computer professionals to mean a variety of forms of hostile, intrusive, or annoying software or program code. Many computer users are unfamiliar with the term, and often use "computer virus" for all types of malware, including true viruses.

Software is considered malware based on the perceived intent of the creator rather than any particular features. Malware includes computer viruses, worms, trojan horses, most rootkits, spyware, dishonest adware, crimeware and other malicious and unwanted software. In law, malware is sometimes known as a computer contaminant, for instance in the legal codes of several American states, including California and West Virginia. Malware is not the same as defective software, that is, software which has a legitimate purpose but contains harmful bugs.

Preliminary results from Symantec published in 2008 suggested that "the release rate of malicious code and other unwanted programs may be exceeding that of legitimate software applications." According to F-Secure, "As much malware [was] produced in 2007 as in the previous 20 years altogether." Malware's most common pathway from criminals to users is through the Internet, by email and the World Wide Web.

Purposes

Many early infectious programs, including the first Internet Worm and a number of MS-DOS viruses, were written as experiments or pranks generally intended to be harmless or merely annoying rather than to cause serious damage to computers. In some cases the perpetrator did not realize how much harm their creations could do. Young programmers learning about viruses and the techniques used to write them only to prove that they could or to see how far it could spread. As late as 1999, widespread viruses such as the Melissa virus appear to have been written chiefly as pranks.

Hostile intent related to vandalism can be found in programs designed to cause harm or data loss. Many DOS viruses, and the Windows ExploreZip worm, were designed to destroy files on a hard disk, or to corrupt the file system by writing junk data. Network-borne worms such as the 2001 Code Red worm or the Ramen worm fall into the same category. Designed to vandalize web pages, these worms may seem like the online equivalent to graffiti tagging, with the author's alias or affinity group appearing everywhere the worm goes.

However, since the rise of widespread broadband Internet access, malicious software has come to be designed for a profit motive, either more or less legal (forced advertising) or criminal. For instance, since 2003, the majority of widespread viruses and worms have been designed to take control of users' computers for black-market exploitation.[citation needed] Infected "zombie computers" are used to send email spam, to host contraband data such as child pornography[7], or to engage in distributed denial-of-service attacks as a form of extortion.

Another strictly for-profit category of malware has emerged in spyware -- programs designed to monitor users' web browsing, display unsolicited advertisements, or redirect affiliate marketing revenues to the spyware creator. Spyware programs do not spread like viruses; they are generally installed by exploiting security holes or are packaged with user-installed software, such as peer-to-peer applications. It is not uncommon for spyware and advertising programs to install so many processes that the infected machine becomes unusable, defeating the intention of the attack.

Infectious malware: viruses and worms

Main articles: Computer virus and Computer worm

The best-known types of malware, viruses and worms, are known for the manner in which they spread, rather than any other particular behavior. The term computer virus is used for a program which has infected some executable software and which causes that software, when run, to spread the virus to other executable software. Viruses may also contain a payload which performs other actions, often malicious. A worm, on the other hand, is a program which actively transmits itself over a network to infect other computers. It too may carry a payload.

These definitions lead to the observation that a virus requires user intervention to spread, whereas a worm spreads automatically. Using this distinction, infections transmitted by email or Microsoft Word documents, which rely on the recipient opening a file or email to infect the system, would be classified as viruses rather than worms. Some writers in the trade and popular press appear to misunderstand this distinction, and use the terms interchangeably.

Capsule history of viruses and worms

Before Internet access became widespread, viruses spread on personal computers by infecting programs or the executable boot sectors of floppy disks. By inserting a copy of itself into the machine code instructions in these executables, a virus causes itself to be run whenever the program is run or the disk is booted. Early computer viruses were written for the Apple II and Macintosh, but they became more widespread with the dominance of the IBM PC and MS-DOS system. Executable-infecting viruses are dependent on users exchanging software or boot floppies, so they spread heavily in computer hobbyist circles.

The first worms, network-borne infectious programs, originated not on personal computers, but on multitasking Unix systems. The first well-known worm was the Internet Worm of 1988, which infected SunOS and VAX BSD systems. Unlike a virus, this worm did not insert itself into other programs. Instead, it exploited security holes in network server programs and started itself running as a separate process. This same behavior is used by today's worms as well.

With the rise of the Microsoft Windows platform in the 1990s, and the flexible macro systems of its applications, it became possible to write infectious code in the macro language of Microsoft Word and similar programs. These macro viruses infect documents and templates rather than applications, but rely on the fact that macros in a Word document are a form of executable code.

Today, worms are most commonly written for the Windows OS, although a small number are also written for Linux and Unix systems. Worms today work in the same basic way as 1988's Internet Worm: they scan the network for computers with vulnerable network services, break in to those computers, and copy themselves over. Worm outbreaks have become a cyclical plague for both home users and businesses, eclipsed recently in terms of damage by spyware.

Concealment: Trojan horses, rootkits, and backdoors

Trojan horses: For a malicious program to accomplish its goals, it must be able to do so without being shut down, or deleted by the user or administrator of the computer it's running on. Concealment can also help get the malware installed in the first place. When a malicious program is disguised as something innocuous or desirable, users may be tempted to install it without knowing what it does. This is the technique of the Trojan horse or trojan.

Broadly speaking, a Trojan horse is any program that invites the user to run it, but conceals a harmful or malicious payload. The payload may take effect immediately and can lead to many undesirable effects, such as deleting all the user's files, or more commonly it may install further harmful software into the user's system to serve the creator's longer-term goals. Trojan horses known as droppers are used to start off a worm outbreak, by injecting the worm into users' local networks.

One of the most common ways that spyware is distributed is as a Trojan horse, bundled with a piece of desirable software that the user downloads from the Internet. When the user installs the software, the spyware is installed alongside. Spyware authors who attempt to act in a legal fashion may include an end-user license agreement which states the behavior of the spyware in loose terms, and which the users are unlikely to read or understand.

Rootkits: Once a malicious program is installed on a system, it is often useful to the creator if it stays concealed. The same is true when a human attacker breaks into a computer directly. Techniques known as rootkits allow this concealment, by modifying the host operating system so that the malware is hidden from the user. Rootkits can prevent a malicious process from being visible in the system's list of processes, or keep its files from being read. Originally, a rootkit was a set of tools installed by a human attacker on a Unix system where the attacker had gained administrator (root) access. Today, the term is used more generally for concealment routines in a malicious program.

Some malicious programs contain routines to defend against removal: not merely to hide themselves, but to repel attempts to remove them. An early example of this behavior is recorded in the Jargon File tale of a pair of programs infesting a Xerox CP-V timesharing system:

Each ghost-job would detect the fact that the other had been killed, and would start a new copy of the recently slain program within a few milliseconds. The only way to kill both ghosts was to kill them simultaneously (very difficult) or to deliberately crash the system. Similar techniques are used by some modern malware, wherein the malware starts a number of processes which monitor one another and restart any process which is killed off by the operator.

Backdoors: A backdoor is a method of bypassing normal authentication procedures. Once a system has been compromised (by one of the above methods, or in some other way), one or more backdoors may be installed, in order to allow the attacker access in the future. The idea has often been suggested that computer manufacturers preinstall backdoors on their systems to provide technical support for customers, but this has never been reliably verified. Crackers typically use backdoors to secure remote access to a computer, while attempting to remain hidden from casual inspection. To install backdoors crackers may use Trojan horses, worms, or other methods.

Malware for profit: spyware, botnets, keystroke loggers, and dialers

During the 1980s and 1990s, it was usually taken for granted that malicious programs were created as a form of vandalism or prank (although some viruses were spread only to discourage users from illegal software exchange.) More recently, the greater share of malware programs have been written with a financial or profit motive in mind. This can be taken as the malware authors' choice to monetize their control over infected systems: to turn that control into a source of revenue.

Since 2003 or so, the most costly form of malware in terms of time and money spent in recovery has been the broad category known as spyware. [citation needed] Spyware programs are commercially produced for the purpose of gathering information about computer users, showing them pop-up ads, or altering web-browser behavior for the financial benefit of the spyware creator. For instance, some spyware programs redirect search engine results to paid advertisements. Others, often called "stealware" by the media, overwrite affiliate marketing codes so that revenue goes to the spyware creator rather than the intended recipient.

Spyware programs are sometimes installed as Trojan horses of one sort or another. They differ in that their creators present themselves openly as businesses, for instance by selling advertising space on the pop-ups created by the malware. Most such programs present the user with an end-user license agreement which purportedly protects the creator from prosecution under computer contaminant laws. However, spyware EULAs have not yet been upheld in court.

Another way that financially-motivated malware creators can profit from their infections is to directly use the infected computers to do work for the creator. Spammer viruses, such as the Sobig and Mydoom virus families, are commissioned by e-mail spam gangs. The infected computers are used as proxies to send out spam messages. The advantage to spammers of using infected computers is that they are available in large supply (thanks to the virus) and they provide anonymity, protecting the spammer from prosecution. Spammers have also used infected PCs to target anti-spam organizations with distributed denial-of-service attacks.

In order to coordinate the activity of many infected computers, attackers have used coordinating systems known as botnets. In a botnet, the malware or malbot logs in to an Internet Relay Chat channel or other chat system. The attacker can then give instructions to all the infected systems simultaneously. Botnets can also be used to push upgraded malware to the infected systems, keeping them resistant to anti-virus software or other security measures.

Lastly, it is possible for a malware creator to profit by simply stealing from the person whose computer is infected. Some malware programs install a key logger, which copies down the user's keystrokes when entering a password, credit card number, or other information that may be useful to the creator. This is then transmitted to the malware creator automatically, enabling credit card fraud and other theft. Similarly, malware may copy the CD key or password for online games, allowing the creator to steal accounts or virtual items.

Another way of stealing money from the infected PC owner is to take control of the modem and dial an expensive toll call. Dialer (or porn dialer) software dials up a premium-rate telephone number such as a U.S. "900 number" and leave the line open, charging the toll to the infected user.

Data-stealing malware

Data-stealing malware is a web threat that divests victims of personal and proprietary information with the intent of monetizing stolen data through direct use or underground distribution. Content security threats that fall under this umbrella include keyloggers, screen scrapers, spyware, adware, backdoors, and bots. The term does not refer to activities such as spam, phishing, DNS poisoning, SEO abuse, etc. However, when these threats result in file download or direct installation, as most hybrid attacks do, files that act as agents to proxy information will fall into the data-stealing malware category.

Characteristics of data-stealing malware

Does not leave traces of the event

The malware is typically stored in the local cache which is routinely flushed

The malware may be installed via a drive-by-download process

The website hosting the malware as well as the malware is generally temporary or rogue

Frequently changes and extends its functions

It is difficult for antivirus software to detect final payload attributes due to the combinations of malware components

The malware uses multiple file encryption levels

Malware kits sold via underground forums are able to generate different files on-the-fly

Thwarts Intrusion Detection Systems (IDS) after successful installation

There are no perceivable network anomalies

The malware hides in web traffic

The malware is stealthier in terms of traffic and resource use

Thwarts disk encryption

Data is stolen during decryption and display

The malware can monitor keystrokes and passwords

Thwarts Data Loss Prevention (DLP)

Leakage protection hinges on metadata tagging, not everything is tagged

Miscreants can use encryption to port data

Examples of data-stealing malware

LegMir, spyware that steals personal information such as account names and passwords related to online games

Qhost, a Trojan that modifies the HOSTS file to point to a different DNS server when banking sites are accessed then opens a spoofed login page to steal login credentials for those financial institutions

Bancos, an info stealer that waits for the user to access banking websites then spoofs pages of the bank website to steal sensitive information

Gator, spyware that covertly monitors web-surfing habits, uploads data to a server for analysis then serves targeted pop-up ads

Data-stealing malware incidents

Eleven people were implicated in a massive identity theft and computer fraud scheme targeting nine U.S. retailers (BJ's Wholesale Club, TJX, DSW Shoe, OfficeMax, Barnes & Noble, Boston Market, Sports Authority and Forever 21). Over 40 million credit and debit card numbers were stolen.

A Trojan horse program stole more than 1.6 million records belonging to several hundred thousand people from Monster Worldwide Inc's job search service. The data was used by cybercriminals to craft phishing emails targeted at Monster.com users to plant additional malware on users' PCs.

Customers of Hannaford Bros. Co, a supermarket chain based in Maine, were victims of a data security breach involving the potential compromise of 4.2 million debit and credit cards. The company was hit by several class-action law suits.

Vulnerability to malware

In this context, as throughout, it should be borne in mind that the "system" under attack may be of various types, e.g. a single computer and operating system, a network or an application.

Various factors make a system more vulnerable to malware:

Homogeneity – e.g. when all computers in a network run the same OS, if you can hack that OS, you can break into any computer running it.

Defects – most systems containing errors which may be exploited by malware.

Unconfirmed code – code from a floppy disk, CD-ROM or USB device may be executed without the user's agreement.

Over-privileged users – some systems allow all users to modify their internal structures.

Over-privileged code – most popular systems allow code executed by a user all rights of that user.

An often cited cause of vulnerability of networks is homogeneity or software monoculture. In particular, Microsoft Windows has such a large share of the market that concentrating on it will enable a cracker to subvert a large number of systems. Introducing inhomogeneity purely for the sake of robustness would however bring high costs in terms of training and maintenance.

Most systems contain bugs which may be exploited by malware. A typical example is the buffer overrun, in which an interface designed to store data in a small area of memory allows the caller to supply more data than will fit. This extra data then overwrites the interface's own structure. In this way malware can force the system to execute malicious code, by replacing legitimate code with its own payload.

Originally, PCs had to be booted from floppy disks, and until recently it was common for this to be the default boot device. This meant that a corrupt floppy disk could subvert the computer during booting, and the same applies to CDs. Although that is now less common, it is still possible to forget that one has changed the default, and rare that a BIOS makes one confirm a boot from removable media.

In some systems, non-administrator users are over-privileged by design, in the sense that they are allowed to modify internal structures of the system. In some environments, users are over-privileged because they have been inappropriately granted administrator or equivalent status. This is a primarily a configuration decision, but on Microsoft Windows systems the default configuration is to over-privilege the user. This situation exists due to decisions made by Microsoft to prioritize compatibility with older systems above security configuration in newer systems[citation needed] and because typical applications were developed without the under-privileged users in mind. As privilege escalation exploits have increased this priority is shifting for the release of Microsoft Windows Vista. As a result, many existing applications that require excess privilege (over-privileged code) may have compatibility problems with Vista. However, Vista's User Account Control feature attempts to remedy applications not designed for under-privileged users through virtualization, acting as a crutch to resolve the privileged access problem inherent in legacy applications.

Malware, running as over-privileged code, can use this privilege to subvert the system. Almost all currently popular operating systems, and also many scripting applications allow code too many privileges, usually in the sense that when a user executes code, the system allows that code all rights of that user. This makes users vulnerable to malware in the form of e-mail attachments, which may or may not be disguised.

Given this state of affairs, users are warned only to open attachments they trust, and to be wary of code received from untrusted sources. It is also common for operating systems to be designed so that device drivers need escalated privileges, while they are supplied by more and more hardware manufacturers, some of whom may be unreliable.

Eliminating over-privileged code

Over-privileged code dates from the time when most programs were either delivered with a computer or written in-house, and repairing it would at a stroke render most anti-virus software almost redundant. It would, however, have appreciable consequences for the user interface and system management.

The system would have to maintain privilege profiles, and know which to apply for each user and program. In the case of newly installed software, an administrator would need to set up default profiles for the new code.

Eliminating vulnerability to rogue device drivers is probably harder than for arbitrary rogue executables. Two techniques, used in VMS, that can help are memory mapping only the registers of the device in question and a system interface associating the driver with interrupts from the device.

Other approaches are:

Various forms of virtualization, allowing the code unlimited access only to virtual resources

Various forms of sandbox or jail

The security functions of Java, in java.security

Such approaches, however, if not fully integrated with the operating system, would reduplicate effort and not be universally applied, both of which would be detrimental to security.

Anti-malware programs

As malware attacks become more frequent, attention has begun to shift from viruses and spyware protection, to malware protection, and programs have been developed to specifically combat them.

Anti-malware programs can combat malware in two ways:

1. They can provide real time protection against the installation of malware software on a computer. This type of spyware protection works the same way as that of anti-virus protection in that the anti-malware software scans all incoming network data for malware software and blocks any threats it comes across.

2. Anti-malware software programs can be used solely for detection and removal of malware software that has already been installed onto a computer. This type of malware protection is normally much easier to use and more popular[citation needed]. This type of anti-malware software scans the contents of the windows registry, operating system files, and installed programs on a computer and will provide a list of any threats found, allowing the user to choose what which files to delete or keep, or compare this list to a list of known malware components, removing files which match.

Real-time protection from malware works identically to real-time anti-virus protection: the software scans disk files at download time, and blocks the activity of components known to represent malware. In some cases, it may also intercept attempts to install start-up items or to modify browser settings. Because many malware components are installed as a result of browser exploits or user error, using security software (some of which are anti-malware, though many are not) to "sandbox" browsers (essentially babysit the user and their browser) can also be effective to help restrict any damage done.

Academic research on malware: a brief overview

The notion of a self-reproducing computer program can be traced back to 1949 when John von Neumann presented lectures that encompassed the theory and organization of complicated automata. Neumann showed that in theory a program could reproduce itself. This constituted a plausibility result in computability theory. Fred Cohen experimented with computer viruses and confirmed Neumann's postulate. He also investigated other properties of malware (detectability, self-obfuscating programs that used rudimentary encryption that he called "evolutionary", and so on). His 1988 doctoral dissertation was on the subject of computer viruses.[13] Cohen's faculty advisor, Leonard Adleman (the A in RSA) presented a rigorous proof that, in the general case, algorithmically determining whether a virus is or is not present is Turing undecidable. This problem must not be mistaken for that of determining, within a broad class of programs, that a virus is not present; this problem differs in that it does not require the ability to recognize all viruses. Adleman's proof is perhaps the deepest result in malware computability theory to date and it relies on Cantor's diagonal argument as well as the halting problem. Ironically, it was later shown by Young and Yung that Adleman's work in cryptography is ideal in constructing a virus that is highly resistant to reverse-engineering by presenting the notion of a cryptovirus. A cryptovirus is a virus that contains and uses a public key and randomly generated symmetric cipher initialization vector (IV) and session key (SK). In the cryptoviral extortion attack, the virus hybrid encrypts plaintext data on the victim's machine using the randomly generated IV and SK. The IV+SK are then encrypted using the virus writer's public key. In theory the victim must negotiate with the virus writer to get the IV+SK back in order to decrypt the ciphertext (assuming there are no backups). Analysis of the virus reveals the public key, not the IV and SK needed for decryption, or the private key needed to recover the IV and SK. This result was the first to show that computational complexity theory can be used to devise malware that is robust against reverse-engineering.

Another growing area of computer virus research is to mathematically model the infection behavior of worms using models such as Lotka–Volterra equations, which has been applied in the study of biological virus. Various virus propagation scenarios have been studied by researchers such as propagation of computer virus, fighting virus with virus like predator codes, effectiveness of patching etc.

Grayware

Grayware (or greyware) is a general term sometimes used as a classification for applications that behave in a manner that is annoying or undesirable, and yet less serious or troublesome than malware. Grayware encompasses spyware, adware, dialers, joke programs, remote access tools, and any other unwelcome files and programs apart from viruses that are designed to harm the performance of computers on your network. The term has been in use since at least as early as September 2004.

Grayware refers to applications or files that are not classified as viruses or trojan horse programs, but can still negatively affect the performance of the computers on your network and introduce significant security risks to your organization. Often grayware performs a variety of undesired actions such as irritating users with pop-up windows, tracking user habits and unnecessarily exposing computer vulnerabilities to attack.

Spyware is software that installs components on a computer for the purpose of recording Web surfing habits (primarily for marketing purposes). Spyware sends this information to its author or to other interested parties when the computer is online. Spyware often downloads with items identified as 'free downloads' and does not notify the user of its existence or ask for permission to install the components. The information spyware components gather can include user keystrokes, which means that private information such as login names, passwords, and credit card numbers are vulnerable to theft. Spyware gathers data, such as account user names, passwords, credit card numbers, and other confidential information, and transmits it to third parties. Adware is software that displays advertising banners on Web browsers such as Internet Explorer and Mozilla Firefox. While not categorized as malware, many users consider adware invasive. Adware programs often create unwanted effects on a system, such as annoying popup ads and the general degradation in either network connection or system performance. Adware programs are typically installed as separate programs that are bundled with certain free software. Many users inadvertently agree to installing adware by accepting the End User License Agreement (EULA) on the free software. Adware are also often installed in tandem with spyware programs. Both programs feed off of each other's functionalities - spyware programs profile users' Internet behavior, while adware programs display targeted ads that correspond to the gathered user profile.

Web and spam

If an intruder can gain access to a website, it can be hijacked with a single HTML element. The World Wide Web is a criminals' preferred pathway for spreading malware. Today's web threats use combinations of malware to create infection chains. About one in ten Web pages may contain malicious code.

Wikis and blogs

Innocuous wikis and blogs are not immune to hijacking. It has been reported that the German edition of Wikipedia has recently been used as an attempt to vector infection. Through a form of social engineering, users with ill intent have added links to web pages that contain malicious software with the claim that the web page would provide detections and remedies, when in fact it was a lure to infect.

Targeted SMTP threats

Targeted SMTP threats also represent an emerging attack vector through which malware is propagated. As users adapt to widespread spam attacks, cybercriminals distribute crimeware to target one specific organization or industry, often for financial gain.

HTTP and FTP

Infections via "drive-by" download are spread through the Web over HTTP and FTP when resources containing spurious keywords are indexed by legitimate search engines, as well as when JavaScript is surreptitiously added to legitimate websites and advertising networks.