System Security, Spring 2010 - Exercise 8

Exercise 1: Definition

i) Worm: A computer worm is a piece of software that can self-replicate without the need of any other executable code thus it is non parasitic. To propagate it will normally use the computer network and send copies of itself to the connected systems and most of the time this will happen without the user being involved since the worm will use vulnerabilities in systems and / or applications. A worm will often include malicious payload that gets executed on the infected machine before it infects further systems.

ii) Virus: A virus relies on the existence of other executable code which makes is a parasitic malware. There are various subtypes of viruses where the classification often relates to their propagation vector. i.e. boot sector viruses that infect the boot sector of floppy disks or hard drives to ensure that they get run when the system starts.

iii) Logic bomb: A logic bomb consists of a condition and the payload that may be hidden in a legitimate application. The condition may be a special date or even something that can be triggered remotely. To be classified as a logic bomb the payload has to be unwanted and unknown to the user.

iv) Backdoor: A backdoor is a mechanism which enables to bypass a security check. The backdoor may also have been implemented for a legitimate reason e.g. to give access to an administrator or to simplify debugging during development.

v) Trojan Horse: A Trojan horse is a software that appears useful to the user but instead enables an attacker to gain access to the system. Once the Trojan horse is installed on the system and the attacker has access (mostly remote) to the system it can be controlled and abused by the attacker.

vi) Adware: The category Adware contains software that does (download and) display advertisement to the user once it has been installed on the system. The Adware may be bundled with legitimate software and installed without the user having consent. Additionally it may be impossible to remove the software without removing the legitimate software as well or the Adware will even persist on the machine when the legitimate software is uninstalled.

vii) Scareware: Scareware is software that gets sold to a user by unethical marketing such as shocking or making him believe that there's an imminent threat that can only be averted when the user buys a software. The sold software is often of limited or no use and may contain malicious code as well.

viii) Rootkit: Rootkits are designed to allow full access to a non privileged user and to hide the activities of an attacker or even sometimes pseudo legitimate software (e.g. USB Sticks with fingerprint reader [4] or DRM protectors [1]). In rare cases a rootkit can actually have a legitimate purpose (e.g. in emulation software or honeypots to detect attempted break-ins or tampering).

ix) Dropper: A dropper is in fact an installer for malware. The malware can for instance be contained in the dropper in an encrypted form or get downloaded to receive the newest still undetectable malware to bypass scanners.

x) Zip bomb: A zip bomb is an archive that is constructed such that the unpacking of it requires many system resources (i.e. CPU time, memory, drive space). A zip bomb can for example be used to crash an anti-virus software such that ensuing viruses cannot be detected anymore.

To some extent all of those categories can indeed be counted as malware since the user has normally not agreed to install them on the computer and they may be a threat to the system or even provide access to a malicious attacker, this does also apply to the pseudo legitimate rootkits since they can potentially be abused to hide malware. However Adware to which the user has consented and that does not contain any malicious code should not be counted as malware since the user has the choice to decide what he want to do.

Sources: Various Wikipedia entries

Exercise 2: Threats

a) On execution of the virus "Virus.Win32.Induc.a" it will check if a Delphi programming environment (versions 4 to 7) is installed on the computer. If the virus finds such an environment it will back up the `SysConst.dcu` file and instead modify the source code `SysConst.pas` to include his infection routine and then compile it. Any subsequent software that is built on the system using the infected `SysConst.dcu` will therefore be infected by the virus.

   The potential damage of the virus is very low since it does not contain any malicious code except for the spreading of the virus.

b) There are at least two possible ways to prevent an infection since the virus will in the first place only replicate if one of the stated Delphi versions is installed and secondly will not infect the `SysConst.dcu` file if the backup file `SysConst.bak` exists thus creating an empty `SysConst.bak` file will prevent an infection.

c) Since the virus creates the backup file `SysConst.bak` one can easily see if the system is infected. Any executable that was built using an infected Delphi version will include the virus code that can be spotted quite easily as well.

d) To clean the system the infected `SysConst.dcu` has to be replaced with the original file which is the `SysConst.bak` file. Afterwards an empty `SysConst.bak` should be created to prevent a new infection. To get completely rid of the virus one would also have to check all executables that were compiled with one of the mentioned Delphi versions.

Sources: [2, 3]

Exercise 3: Removal

a) Most well known anti-virus companies do also provide malware-removal software. For specific, well analyzed and widespread malware this might be a software tool that can be used while the infected system is running or in the more general version a bootable rescue CD that scans the system and removes the malware.

b) To be sure that a malware infection gets removed properly one cannot use the malware removal software while the system is still running. Thus a second system or a live CD system, that does not execute any code of the infected system, has to be used to scan and remove any known infections, however since the malware might already have installed additional malicious software that is still unknown and undetectable not all malware might get removed completely such that restoring a clean backup could be the only way to be sure that all malware was removed.

Exercise 4: Reasons

a) Sure enough it will be harder to write malware for a closed system / application that does not provide any documentation, at least before a developer or attacker decides to reverse engineer it and gets an idea on how the interesting stuff works. However one has to see that a system without a proper documentation most likely won't get the popularity of a system with adequate documentation, since there won't be any additional features that get developed by others, and thus attract less attackers which is also a statement made in the article.

   It will however not be impossible to write malware, e.g. using fuzzy testing to find a way to remotely crash a system, and since there's no documentation one has to rely on the manufacturer to properly close any vulnerability that may come to light without knowing the posed security risk. Without documentation it is indeed nearly impossible to verify that the manufacturer has designed the system such that it is working as supposed since no one, except for the manufacturer, really knows what supposed means.

b) Even if we assume that the statement is completely true it does not necessarily hold in both directions and there will be systems / applications that have a proper documentation but still are highly secure. Additionally a system with a proper documentation will be used and verified, when we look at the long term usage, by multiple developers such that existing deficiencies will probably get redesigned and refactored making it in fact a more secure system as time goes by.

   However it might be true that when a system / application gets released there might still be serious miss behaviors that were not spotted during development but can be exploited more easily due to the available documentation. Furthermore if a security relevant bug is found and the corresponding patch is released the documentation can be used to understand the cause of the security risk and thus the writing of malware that targets systems that are not yet patched.

Exercise 5: Prevention and Protection

Besides the usual updating of the operating system as well as applications one should minimize the risk of being infected by hardening or disabling any unused services, applications and other components that could be a possible target of malware. Furthermore the principle of least privileges where users and software have the lowest possible security privileges to lessen the potential impact of malware should be used. An additional mechanism is also Trusted computing where only trustworthy applications are allowed to run. Every application is verified before it gets executed thus preventing malware to run.

One might also want to deploy gateways that filter out known malware using traditional signature based scanning of files or even more sophisticated behavior analysis to detect unknown malware. Some of those mechanisms may also be implemented locally by using adequate software.

Additionally to detect an already infected machine Intrusion Prevention Systems could monitor the network traffic and notify an administrator or even isolate any machines whose behavior is suspicious.

References

[1] Markus Hansen. DRM-Desaster: Das Sony BMG-Rootkit. *Datenschutz und Datensicherheit - DuD*, 30(2):95–97, Februar 2006. ISSN 1614-0702. doi: 10.1007/s11623-006-0026-4. URL `http://www.springerlink.com/index/10.1007/s11623-006-0026-4`.

[2] Nick Hodges. Frequently Asked Questions about the W32/Induc-A Virus (Compile-A-Virus). URL `http://edn.embarcadero.com/article/39851`.

[3] F-Secure Labs. News from the Lab: 0wn1ng Delphi, . URL `http://www.f-secure.com/weblog/archives/00001752.html`.

[4] F-Secure Labs. News from the Lab: Double Whammy! Another Sony Case (And it's Not BioShock), . URL `http://www.f-secure.com/weblog/archives/archive-082007.html#00001263`.