337 Assignment 6

Due: 2/27/2017

By: Ryan Peters

**Question #1**

Problem statement: An example of a host-based intrusion detection tool is the tripwire program. This is a file integrity checking tool that scans files and directories on the system on a regular basis and notifies the administrator of any changes. It uses a protected database of cryptographic checksums for each file checked and compares this value with that recomputed on each file as it is scanned. It must be configured with a list of files and directories to check and what changes, if any, are permissible to each. It can allow, for example, log files to have new entries appended, but not for existing entries to be changed. What are the advantages and disadvantages of using such a tool? Consider the problem of determining which files should only change rarely, which files may change more often and how, and which change frequently and hence cannot be checked. Consider the amount of work in both the configuration of the program and on the system administrator monitoring the responses generated.

Trip-wire advantages:

* Where file changes consist of only additions to the file, Trip-wire can check for unauthorized modifications to existing entries.
* It is less difficult to monitor a small list of configuration files and critical system programs because the changes in that file can be easily detected.

Trip-wire disadvantages:

* Welcome to the land of false positives
* Only checks where instructed to check, thus not securing entirety of system.
* Requires read-only media protection of the checksum lest attackers perve that as well, rendering all detection attempts useless.

**Question #2**

Problem statement: The question arises as to whether it is possible to develop a program that can analyze a piece of software to determine if it is a virus. Consider that we have a program D that is supposed to be able to do that. That is, for any program P, if we run D (P), the result returned is TRUE (P is a virus) or FALSE (P is not a virus). Now consider the following program:

Program CV :=

{

. . .

main-program :=

{

if D(CV) then goto next:

else infect-executable;

}

next:

}

In the preceding program, infect-executable is a module that scans memory for executable programs and replicates itself in those programs. Determine if D can correctly decide whether CV is a virus.

Assumptions:

* D is only checking active executable files.
* While CV is in memory, it's an active executable.
  + Under these assumptions, D can catch CV. As D is essentially acting in the same way that CV would check a targeted executable file if it's already infected.

**Question #3**

Problem statement:

Consider the following fragment:

legitimate code

if data is Friday the 13th;

crash\_computer();

legitimate code

What type of malware is this?

* A logic bomb!!

**Question #4**

Problem statement: Assume you have found a USB memory stick in your work parking area. What threats might this pose to your work computer should you just plug the memory stick in and examine its contents? In particular, consider whether each of the malware propagation mechanisms we discuss could use such a memory stick for transport. What steps could you take to mitigate these threats, and safely determine the contents of the memory stick?

Threats that are created to the user computer while plug in the memory stick?

* When a memory stick is plugged into the computer which was founded on the parking area may create a variety of threats to the confidentiality, integrity, and availability of the system.
  + The memory stick may transmit:
    - Executable virus - Executable program files are affected by machine executable virus and these program files work with specific operating system and in some cases it is based upon the hardware platform.
    - Macro virus - Files with macro or scripting code are affected by macro viruses and its support effective content in a field of user document types and is translated by an application.
    - Worm - While viewing the memory stick, worm runs automatically and infects other appropriate files as a virus on the system.
    - Trojan horse - Trojan horse is the malicious piece of code that is delivered through the mail or web page or through the USB drives that causes damage to the data or system.
* To mitigate the threats:
  + The user should “scan the memory stick” with appropriate up-to-date anti-virus software.
  + The user could “check the memory stick” in a controlled environment. For example a live-boot Linux or emulation environment.

**Question #5**

Problem statement: Suppose you have a new smartphone and are excited about the range of apps available for it. You read about a really interesting new game that is available for your phone. You do a quick Web search for it, and see that a version is available from one of the free marketplaces. When you download and start to install this app, you are asked to approve the access permissions granted to it. You see that it wants permission to “Send SMS messages” and to “Access your address-book”. Should you be suspicious that a game wants these types of permissions? What threat might the app pose to your smartphone, should you grant these permissions and proceed to install it? What types of malware might it be?

Should you be suspicious that a game wants these types of permissions?

* Yesh!

What threat might the app pose to your smartphone?

* Possible malware:
  + Trojan
  + Worm
  + Virus

Should you grant these permissions and proceed to install it?

* Yesh! Because who cares if your phone is infected? You got candy to crush!!
  + Actually, you should stop the install and investigate alternative download mirrors and reviews on those mirrors to see if there are any known threats at the used sites.

**Question #6**

Problem statement: List the types of attacks on a personal computer that each of a (host-based) personal firewall, and anti-virus software, can help you protect against. Which of these countermeasures would help block the spread of macro viruses spread using email attachments? Which would block the use of backdoors on the system?

* I shamelessly looked up blockable malware for firewall/anti-virus defense tools.
* Needless to say, there are a lot, so here is a list of the top level general malware types.

1. Trojan
2. Virus
3. Worms
4. Adware
5. Auto-rooter
6. Downloaders
7. Macro Virus
8. Logic Bomb

* Each of the listed malware threats are simply the most generalized category of threats that a firewall/anti-virus combo could reasonably combat. Each threat encompasses a broader domain of threats that generally fail to defeat combined firewall/anti-virus defense.

Which of these countermeasures would help block the spread of macro viruses spread using email attachments?

* A combination of firewall and anti-virus would be the ideal defense.
* The firewall with statefull-packet-inspection can help contain macro virus propagation by monitoring for abnormal amounts of traffic containing identical message contents.
* The anti-virus would monitor for abnormal behavior and flag then quarantine the source.

Which would block the use of backdoors on the system?

* The firewall!

**Question #7**

Problem statement: Research and list any three malwares that occurred in 2017. If possible, classify them as virus, worm or Trojan. Provide technical details of how the attack occurred, what vulnerability was exploited, what was the impact of the attack. Briefly provide countermeasures that could have possibly prevented these attacks. Provide proper references and citations.

from:

<https://www.totaldefense.com/security-blog/the-worst-malware-of-2017>

Disclaimer: I'm a lazy piece of tish... sorry.

**WannaCry**

This piece of ransomware emerged in May 2017. It infected more than 400,000 machines and caused trouble for Spanish telco Telefonica and the U.K. National Health Service, among others. Very few of its victims actually paid the ransom.

WannaCry had several notable capabilities. First, it exploited outdated and unpatched operating systems. Most of its victims were running Microsoft Windows 7, which was originally released in 2009, or the even older Windows XP. All of them were also missing the patch for a flaw in the pivotal Server Message Block (SMB) protocol, which WannaCry took advantage of to spread.

Another novel feature of WannaCry was its mysterious inclusion of a kill switch. A security researcher was able to shut down the threat by registering a specific URL listed in WannaCry’s code, according to Wired.

Ransomware has loomed large n 2017.Ransomware has loomed large n 2017.

**NotPetya**

Following in WannaCry’s footsteps, NotPetya emerged in the summer of 2017. It targeted the same Windows weakness – known as EternalBlue – as WannaCry but added a few twists of its own.

Most importantly, it included a password-harvesting feature that enabled it to steal credentials from the local filesystems and memories of machines as it jumped between them, according to Forbes. It also lacked the convenient kill switch inexplicably contained in WannaCry.

**Bad Rabbit**

The latest of the bunch, Bad Rabbit emerged in October 2017 in Russia and Ukraine. Unlike WannaCry and NotPetya, it didn’t take advantage of EternalBlue.

However, it was designed to spread through SMB (like WannaCry), and it could harvest usernames and passwords (like NotPetya). One of its defining features was its ability to infect devices by tricking their users to install a fake update to Adobe Flash Player, according to Vice.

This attack angle has nothing to do with the many flaws in Flash documented over the years. Instead, it is designed to compromise the machines of visitors to specific sites – a process sometimes called drive-by downloading.