Computer Security Seminar — Lecture 2

Orr Dunkelman

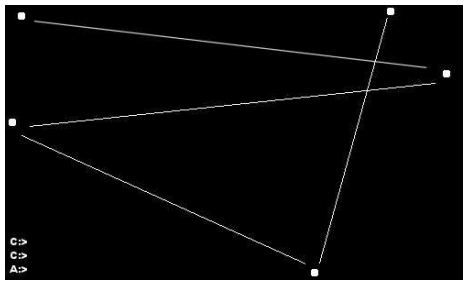
Computer Science Department

28^{th} October, 2012

Outline

Computer Viruses

- One of the eldest malicious threats on computer systems.
- A computer program which infects other programs.
- Once an infected program is loaded, the virus is first executed.
- Usually it loads to memory and infects the "loading mechanism" of the OS.
- Thus, each time another program is loaded, the virus can decide to infect it.
- The same goes to when the infected program "touches" an executable.



Computer Viruses (cont.)

- As computer viruses want to spread themselves, they do not immediately inflict damage to the system.
- After a sufficient amount of infections was achieved, the viruses tended to delete files, or cause other damages.
- Some viruses were quite subtle (causing trouble only a specific date), and some were very “aggressive”.
- And of course, there was Ping-Pong.

The Cat and Mouse Game

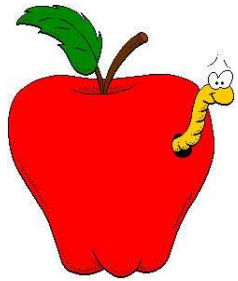
- Anti-virus software was developed very quickly after viruses started spreading.
- The first anti-virus software looked for the code of the virus in files (signature-based).
- As a result, many viruses started "hiding" in various places:
  - The boot record area,
  - Hiding in memory using the "terminate and stay resident" mechanism.
  - The file allocation table,
- Which led the anti-virus software to start looking in these places.

The Cat and Mouse Game (cont.)

- As most anti-virus software were looking for signatures, many viruses started to transform themselves:
  - Inserting "NOP" instructions,
  - Reordering instructions (which do not affect the outcome),
  - Encrypting the code (with each new infection changing the key),
- Which led anti-virus software to start looking for different things...

Computer Worms

- Computer worms are a self-replicating malware.
- Unlike viruses which need “help” by activating their carrying program, worms replicate themselves with no intervention.
- Worms spread themselves by attacking other computers (automatically).
- So once they start spreading, they tend to spread very quickly.



Computer Worms (cont.)

- Most worms are based on using *remote exploits*.
- If the exploit allows for *a remote code execution*, then this exploit can be automated, and grouped with a worm.
- So besides the actual infection, the worm needs a method of propagation.
- This is done either by looking at contacts (users/machines), or just randomly trying.
- Most worms actually use well known security holes, for which a *patch* was issued.
- Which raises the question — why security patches are not installed as quickly as possible.

Window of Vulnerability

- One would expect that once a security issue is identified (e.g., by a report), a patch would be released as quickly as possible.
- Once a patch is released, one would expect the patch to be installed as quickly as possible.
- However, real life data shows something completely different:

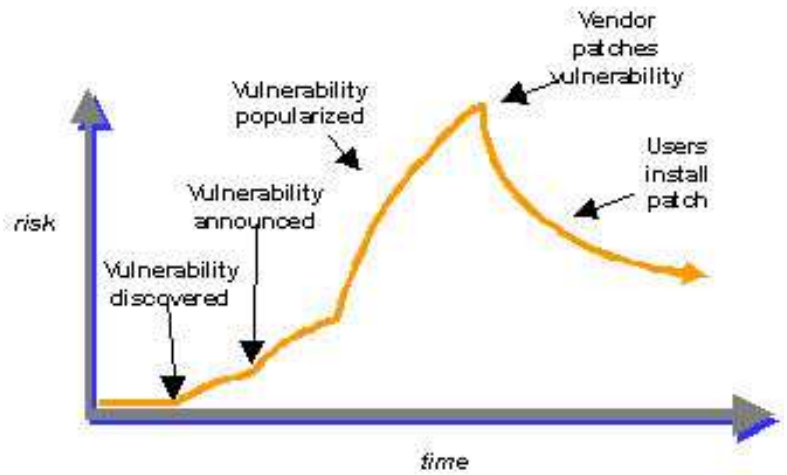


Figure 1

(taken from Schneier's cryptogram, based on data found in Windows of Vulnerability: A Case Study Analysis, Bill Arbaugh. Bill Fithen and John McHugh, 2000).

Trojan Horses

- Trojan horse is a malware which is entered into the system by the user who is tricked to believe the malware does something else.
- The main difference between viruses and Trojan horses is that the Trojan horse is deceiving by design.
- Once the Trojan horse is installed, it can be used to control the user's system: log keyboard strokes, capture the screen, expose sensitive data, and more.

Rootkits — Hide and Seek for the Advanced

- After infecting a machine, the hacker tries to maintain control.
- At the same time, the user wishes to clean his machine, and may have protection software aimed at identifying malware.
- Rootkits are invisibility cloaks that the malware uses to hide itself from the prying eye of these protection mechanisms.
- Rootkits may perform many actions to prevent detection, depending on the level in which they run.

Hiding Rootkits

- Consider a Trojan horse which keeps a log of screen shots.
- It needs to store these screen shots, i.e., the file system must contain the files*.
- A user who asks for the list of files in the area of these screen shots, may find "incriminating" files.
- Hence, the rootkit must monitor all requests to see the list of files, and alter the results according to the operation needed.
- The exact level in which the rootkit is deployed (user space vs. kernel space), determines which mechanisms the rootkit can use to hide itself.

*The Trojan may actually use storage units without "notifying" the OS, but then the OS may re-use these storage units.

Botnets — The Power of the Masses

- Today, many of the malware is written as part of a business model.
- Instead of buying many machines, connecting them to the internet, etc. it is much cheaper to infect them!
- Hence, cyber-crime people just infect many machines, and put them into botnets.
- This botnets are then used by the hacker to attack other systems.

Botnets — Structure

- Most botnets are composed of a few machine types:
  - Zombies — end machines which are infected.
  - Command & Control servers — to publish new versions of the malware, send commands to botnets, etc.
  - Master servers — the ones which control the network.
- Many efforts are put into finding the Command & Control servers, and removing them from the network.
- As a result, many botnets now come with peer-to-peer Command & Control capabilities.

Botnets — Uses

- Once a botnet is deployed, one can use it for many types of attacks:
  - Distributed Denial of Service attacks,
  - Spam distribution,
  - Collecting data (spying),
  - Click frauds,
  - Phishing attempts,
  - Collecting electronic "goodies"
  - Hiding other attacks using various techniques: Fast flux, anonymization, moving rouge servers from one node to another, etc.

Backdoors — Planning Ahead Your Attack

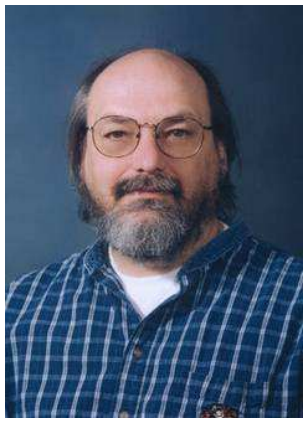
- *Backdoors* are hidden access points left behind by the developer of the system.
- Typically, these backdoors allow the developer to access the system, by setting a special username/password combination.
- Today, after a malware takes over a machine, it installs a backdoor, that allows the hacker to connect to the machine at any time.

Dealing with Backdoors

- Dealing with "classic" backdoors:
  - Obtain the source code, check the source code, and compile on your own machine.
  - Reverse engineer binaries (executables), and look for branches around the login procedures.
- Dealing with backdoors that were installed:
  - Scan your computer (from a different computer), looking for open ports,
  - Do not get infected by malware,
  - Tighten your firewall's definitions (especially if the firewall is external),
  - Protect your network with an IDS.

Reflections on Trusting Trust

- What happens when the developer wishes to install a backdoor, but must supply the source code?
- If the software is a "delicate" one, it may be reviewed by many.
- The idea is to "hide" in plain site — namely, the compiler.
- When receiving the Turing award, Ken Thompson delivered a speech describing how he installed a backdoor into the Unix' login process.



Reflections on Trusting Trust (cont.)

- Ken knew that the login code was under a lot of review.
- Hence, he made sure that the compiler would identify login programs, and install a backdoor to that login program.
- However, the source of the compiler is also available for review.
- The solution is to install a "tainted" compiler.
- Namely, when the compiler identifies that it compiles a compiler, it adds the "backdoor installer" into the compiler.
- And Voilá.

Good Surprises — Easter Eggs

- Sometimes, developers get bored.
- They hide easter eggs — undocumented small “bonuses” in various applications.
- These bonuses range from games inside games, till small “inside jokes”.
- For example, search for the word “recursion” in google.

Anti-Viruses (and Related Software)

- Since the development of viruses, anti-viruses tried to detect (and sometimes disinfect).
- Such solutions can be separated into
  - Signature-based vs. Behavior-based,
  - On-access vs. Scan-based,
- Today, these solutions also deal with worms, trojan horses, etc.
- Some of them also try to fight adware.

Firewalls

- Firewalls are mechanisms to separate networks.
- The firewall is installed on a choke point between the two (or more) networks, and filter packets.
- The idea is to identify packets which stem from malicious activity and stop them, before they reach their destination.
- There are a few types of firewalls, depending on how much they analyze packets:
  - Stateless packet filters,
  - Statefull packet filters,
  - Deep packet inspection,

Firewalls (cont.)

- The stateless packet filters look at each packet at a time.
- Hence, their filtering capabilities are quite limited, but they tend to be fast.
- The stateful packet filters look at the session, and decide whether the session is legitimate.
- Deep packet inspection also keeps track of the information flowing inside the session.

Proxy Servers

- A different type of network isolation solution is the proxy server.
- Instead of a direct connection between the two end parties, one of them is using a middleman.
- This proxy server takes part in the communication and it can see and analyze it as a whole.
- As a result, the proxy server can identify very complex threats.
- For example, the proxy server can analyze all HTTP communication, identifying malicious sites or content.
- The main disadvantage of the proxy is the fact it is required to take part in the communication, and thus, it needs to be designed for each new protocol.

Intrusion Detection Systems

- Intrusion detection systems look for suspicious activity in the network.
- They search for attack patterns which are more complicated than a single connection.
- For example, they can monitor the internal network to see if the network behaves "incorrectly".
- These systems can either be signature-based or behavior-based (and most use both methods).
- Some IDSes also serve as an Intrusion Prevention System (which identifies the intrusion in an online manner and mitigate the attack).

Intrusion Detection Systems

- Honeypots and honeynets are traps for hackers.
- These are virtual machines which seem to be weak (i.e., unprotected or unpatched systems).
- These machines are not supposed to be accessed, so whomever is accessing them, is probably not one of the "good guys".
- By monitoring the hackers' activity, one can learn about the actual attacks in the "wild".

Rootkit Detection

- The detection of rootkits is very tricky.
- If the software operates inside a compromised environment, how can it tell whether the system is intact?
- The first option is of course to boot the system, and start the scanning from a clean state.
- However, many machines cannot be booted.
- In this case, the detection is much harder.

Rootkit Detection — Distinguishing Reality from Dreams

- Once you run in a compromised system, you cannot trust whatever the OS tells you.
- However, the OS gives you many mechanisms to obtain the same results.
- For example, there are several ways to obtain the list of files in a directory. In the case of Windows:
  - Calling FindFirstFile and FindNextFile (using Kernel32.dll),
  - Access the kernel directly,
  - Accessing the harddrive directly.
- If the answers of the different mechanisms differs, then someone is “tweaking” the results.

Rootkit Detection — Distinguishing Reality from Dreams (cont.)

- Other rootkit detection methods:
  - Identify the hooks installed by the rootkit,
  - Measure the time needed for various actions,
  - Use trusted neighbors to measure the system from the outside,