جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

قسم الامن السيبراني
DEPARTMENT OF CYBER SECURITY

SUBJECT:

MALICIOUS CODES

CLASS:

THIRD

LECTURER:

DR. SUHA ALHUSSIENY

LECTURE: (6)

COMPARISON OF ANTI-VIRUS DETECTION TECHNIQUES

WORMS , THE MORRIS WORM

Comparison of Anti-Virus Detection Techniques:

Scanning: Gives precise identification of any viruses that are found. This characteristic makes scanning useful by itself, as well as in conjunction with other anti-virus techniques. But requires an up-to-date database of virus signatures for scanning to be effective. Even assuming that users update their virus databases right away, which isn't the case, there is a delay between the time when a new threat is discovered and when an anti-virus company has a signature update ready.

Static heuristics: Static heuristic analysis detects both known and unknown viruses. But: False positives are a major problem, and a detected virus is neither identified, nor disinfectible except by using generic methods.

Note: In terms of the accuracy of an IDS, there are four possible states for each activity observed. A true positive state is when the IDS identifies an activity as an attack and the activity is actually an attack. A true positive is activity is actually an attack. That is, a false negative is when the IDS fails to catch an attack.

Integrity checkers: Integrity checkers boast high operating speeds and low resource requirements. They detect known and unknown viruses. But: Detection only occurs after a virus has infected the computer, and the source of the infection can't necessarily be pinpointed. An integrity checker can't detect viruses in newly-created files, or ones modified legitimately,

such as through a software update. Ultimately, the user will be called upon to assess whether a change to a file was made legitimately or not. Finally, found viruses can't be identified or disinfected.

Behavior blockers: Known and unknown viruses are detected. But: While a behavior blocker knows which executable is the problem, unlike an integrity checker, it again cannot identify or disinfect the virus.

Emulation: Any viruses found are running in a safe environment. Known and unknown viruses are detected, even new polymorphic viruses. But emulation is slow. The emulator may stop before the virus reveals itself, and even so, precise emulation is very hard to get correct.

Verification, Quarantine, and Disinfection:

The tasks for anti-virus software that lie beyond detection are verification, quarantine, and disinfection. Compared to detection, these three tasks are performed rarely, and can be much slower and more resource-intensive if necessary.

Verification: After the initial detection of a virus occurs, Anti-virus software will often perform a secondary verification. It is performed for two reasons. First, it is used to reduce false positives that might happen by coincidence. Second, verification is used to positively identify the virus.

Quarantine: When a virus is detected in a file, anti-virus software may need to quarantine the infected file, isolating it from the rest of the system. Quarantine is only a temporary measure, and may only be done until the user decides how to handle the file.

Disinfection: It does not mean that an infected system has been restored to its original state, even if the disinfection was successful, there are different ways to do disinfection: Restore infected files from backups. Because everyone meticulously keeps backups of their files, the affected files can be restored to their backed-up state.

Virus Databases and Virus Description Languages

A virus database and virus description languages are crucial components in the field of cybersecurity, particularly in antivirus and anti-malware systems. Here's an overview of both.

Virus Database

- Purpose: A virus database is a comprehensive repository of known malware signatures, behaviors, and other characteristics. Antivirus software uses this database to detect and remove malware on a system.

- Content:

- Signatures: Unique patterns or sequences in the code of known viruses. These can be binary patterns, file hashes, or specific code sequences.

- Behavioral Patterns: Descriptions of how a virus operates, such as file access patterns, network traffic behavior, or changes to system settings.

- Metadata: Information about each virus, including its name, type, origin, known payloads, and potential impacts.

- Update Mechanism: The database needs regular updates to include new threats as they are discovered. Most antivirus programs frequently update their virus databases to stay effective against emerging malware.

Virus Description Languages

- Purpose: Virus description languages are used to create formal descriptions of malware. These languages help security tools analyze and categorize viruses based on their behaviors, signatures, and effects on systems,e.g (YARA:, OpenIOC (Indicator of Compromise, Snort Rules)

Key Features:

- Pattern Matching: These languages allow for complex pattern matching against files, processes, or network traffic, helping to identify known or unknown malware.

- Flexibility: The languages are typically flexible enough to describe both known viruses and emerging threats, allowing for the creation of signatures that can evolve with the threat landscape.

- Automation: Descriptions written in these languages can be automatically used by security tools to scan for malware without manual intervention.

How They Work Together

1. Detection: When a file or behavior on a system matches a signature or description in the virus database (often defined using a virus description language like YARA), the antivirus software flags it as malicious.

2. Analysis: The descriptions in the virus database help cybersecurity professionals understand the nature of the detected malware, its potential impact, and how to remove or mitigate it.

3. Updates: As new viruses are discovered, their characteristics are described using virus description languages, and these descriptions are added to the virus database.

In summary, the virus database contains the data necessary for identifying malware, while virus description languages provide the tools to describe and detect that malware in a structured, automated manner.

Worms

A worm is a program that can replicate itself and send copies from computer to computer across network connections. Upon arrival, the worm may be activated to replicate and propagate again. In addition to propagation, the worm usually performs some unwanted function. An e-mail virus has some of the characteristics of a worm because it propagates itself from system to system. we can still classify it as a virus because it uses a document modified to contain viral macro content and requires human action. A worm actively seeks out more machines to infect and each machine that is infected serves as an automated launching pad for attacks on other machines.

The first known worm implementation was done in in the early 1980s. Network worm programs use network connections to spread from system to system. Once active within a system, a network worm can behave as a computer virus or bacteria, or it could implant Trojan horse programs or perform any number of disruptive or destructive actions.

To replicate itself, a network worm uses some sort of network vehicle.

Examples include the following:

1. Electronic mail facility: A worm mails a copy of itself to other systems, so that its code is run when the e-mail or an attachment is received or viewed.

2. Remote execution capability: A worm executes a copy of itself on another system, either using an explicit remote execution facility or by exploiting a program flaw in a network service to subvert its operations.

Note: Remote code execution is the ability an attacker has to access someone else's computing device and make changes, no matter where the device is geographically located.

3. Remote login capability: A worm logs onto a remote system as a user and then use commands to copy itself from one system to the other, where it then executes.

The new copy of the worm program is then run on the remote system where, in addition to any functions that it performs at that system, it continues to spread in the same fashion.

A network worm exhibits the same characteristics as a computer virus: (a dormant phase, a propagation phase, a triggering phase, and an execution phase)

The propagation phase generally performs the following functions:

1. Search for other systems to infect by examining host tables or similar repositories of remote system addresses.

Note: Remote address is the IP Address/host name of the remote computer to which the connection is connected.

2. Establish a connection with a remote system.

3. Copy itself to the remote system and cause the copy to be run.

Note: A remote computer is a computer to which a user does not have physical access, but which he or she can access or manipulate via some kind of computer network. Remote desktop software allows a person to control a remote computer from another computer.

The Morris Worm

Until the current generation of worms, the best known was the worm released onto the Internet by Robert Morris in 1988 .The Morris worm was designed to spread on UNIX systems and used a number of different techniques for propagation.

When a copy began execution, its first task was to discover other hosts known to this host that would allow entry from this host.The worm performed this task by examining a variety of lists and tables, including system tables that declared which other machines were trusted by this host, users' mail forwarding files, tables by which users gave themselves permission for access to remote accounts, and from a program that reported the status of network connections.

For each discovered host, the worm tried a number of methods for gaining access:

1. It attempted to log on to a remote host as a legitimate user. In this method, the worm first attempted to crack the local password file and then used the discovered passwords and corresponding user IDs. The assumption was that many users would use the same password on different systems. To obtain the passwords, the worm ran a password-cracking program that tried

a) Each user's account name and simple permutations of it

b) A list of 432 built-in passwords that Morris thought to be likely candidates2

c) All the words in the local system dictionary

2. It exploited a bug in the UNIX finger protocol, which reports the whereabouts of a remote user.

3. It exploited a trapdoor in the debug option of the remote process that receives and sends mail.

If any of these attacks succeeded, the worm achieved communication with the operating system command interpreter. It then sent this interpreter a short bootstrap program, issued a command to execute that program, and then logged off. The bootstrap program then called back the parent program and downloaded the remainder of the worm. The new worm was then executed.

H.W/

1. Which antivirus technique requires an up-to-date virus signature database to be effective?
2. What is a major drawback of static heuristic analysis?
3. Which technique can detect both known and unknown viruses but cannot identify or disinfect them?
4. Integrity checkers primarily detect viruses by:
5. Which method can detect known and unknown viruses but is slow due to safe environment execution?
6. Which task in antivirus software isolates an infected file to prevent further spread?
7. What is the main purpose of verification in antivirus systems?
8. Disinfection may not fully restore a system because:
9. A virus database stores:
10. Virus description languages such as YARA are mainly used for:
11. What is a key feature of virus description languages?
12. Which detection phase involves comparing observed activity to stored virus patterns?

13. A worm differs from a virus mainly because it:

14. Which of the following best describes a network worm?

15. Which of the following was the first known worm to spread across UNIX systems?

16. One propagation method used by the Morris Worm was:

17. Which UNIX protocol bug was exploited by the Morris Worm?

18. What did the Morris Worm use to find remote hosts to infect?

19. What was one reason the Morris Worm spread widely?

20. Which antivirus method executes code in a controlled environment to observe its behavior?