Malware – set of instructions that cause a site’s security policy to be violated.

Different types of malware share many common characteristics.

Defenses focus on these characteristics, so defences apply to many different types of malware.

That is part of reason why security companies that market malware detection and prevention tools refer to them as antivirus tools.

When malware infects a system , it alters memory content or disk files. Thus one defence is to look for such changes.

Some mechanisms use manipulation detection codes(MDC’s) to apply some function to a file to obtain a set of bits called signature block and then protect that block.

If after recomputing signature block ,result differs from stored signature block ,file has changed.

Tripwire is an integrity checker that targets the UNIX environment. The program computes a signature block for each file and stores it in a database.

The signature for each file consists of file attributes (size, owner, protection mode and inode number) and various cryptographic checksums.

When tripwire is executed, it recomputes the signature block and compares the recomputed blocks with blocks stored. If any differ, file reported as being possibly corrupted.

Assumption is file doesn’t contain malicious logic before signature generated.

All integrity based schemes rely on software that if infected may fail to report tampering.

Performance will be affected because encrypting the file, or computing the signature block may take a significant amount of time.

The encrypting key must also be secret ,if not malicious logic can easily alter a signed file without the change being detected.

Antivirus scanners check files for specific viruses and if a virus is present ,either warn the user or attempt to cure infection by removing virus.

Many such agents exist for personal computers , but because each agent must look for particular characteristics or behaviours of virus or set of virus ,cannot detect viruses with characteristics not yet been analysed.

A malware signature is an algorithm that identifies malware. Ideally it should be unique to minimize false positives.

These signatures began as static sequences of bits and grew to include patterns of behaviour.

Containment

Because a user unknowingly executes malicious logic ,that code can access and affect objects within user’s protection domain.

So limiting objects accessible to a process run by a user is obvious protection technique.

This draws on mechanisms for confining information.

Sandboxing

Sandboxes and virtual machines implicitly restrict process rights.

Common implementation – restrict program by modifying it.

How are we going to manage to discover malware and do something about them in software?

We are talking about things that are normally called anti virus systems but call them anti -malware systems.

Malware protection

1)Recognise them

Main method – using signatures and recognising signatures. Idea is I can recognise the program being propagated. It has the same kind of pattern in its code generally speaking ,so whenever I see that pattern say we know this pattern ,we don’t like it, don’t let it in or try and eradicate it from our system. That is the major strategy.

What things look like when they are in primary memory rather than when they are in secondary memory. There are other strategies but primarily we are interesting ourselves in the idea that when something is on hard drive can recognise it as malware even if it’s a virus that is infected ,a non-malicious programme we should be able to find that pattern there.

2)Recognise their side effects

3)Segregate and monitor

Which of the following are reasonable countermeasures to malicious software?

Select one or more:

a)

Gamble. The risk of infection is not 100% if one knows approximately which files not to download, and which sites not to visit and so on, so with a bit of luck one can avoid a malware infection by taking a chance or "gambling".

b)

Have extremely good knowledge about the system, its environment, the software installed and how it is used, and every action that is taken using the system.

"Knowledge is power". In theory, if you know everything about your system; exactly how the hardware was made, how the software was coded, what effects the system's environment may have on it, and all the input and output passing through it, you will have a secure and malware-free system. However, this is not very reasonable.

c)

Remove the cable to the local area network and the Internet, alternatively, disable the wireless network interface card; block or remove all peripheral connections like USB, PS-2, Firewire, e-Sata, etc.

d)

Use an anti-virus, or anti-malware software, that automatically removes all suspicious files on the system.

Anti-malware software is very useful for malware that is already known. However, it does not protect against unknown (e.g. modified version of already existing) malware. Furthermore, it might also remove some tools used by more advanced users, like yourselves, due to the fact that they can be used by malware and/or used individually to attack the local system, or another system.

Feedback

The correct answers are: Remove the cable to the local area network and the Internet, alternatively, disable the wireless network interface card; block or remove all peripheral connections like USB, PS-2, Firewire, e-Sata, etc., Use an anti-virus, or anti-malware software, that automatically removes all suspicious files on the system., Have extremely good knowledge about the system, its environment, the software installed and how it is used, and every action that is taken using the system.