Flame and other viruses:

tools or weapons?

Luis J. Dominguez Perez

Malware

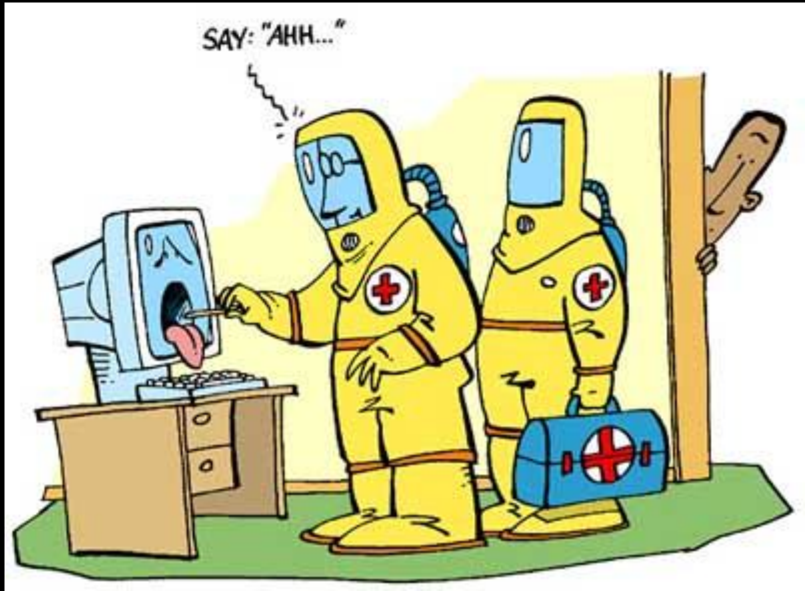
Short for malicious software. It’s a software created to disrupt the computer normal operation, to gather sensitive information, or to gain access to private computer systems.

Malware as a category

Malware is a category of software that includes: computer virus, worms, trojan horses, spyware, adware, most rootkits, and some other malicious programs.

Computer virus

Is a small piece of software that piggybacks on real programs or documents to exploit some bug on the program or reading program with the aim to destroy information.



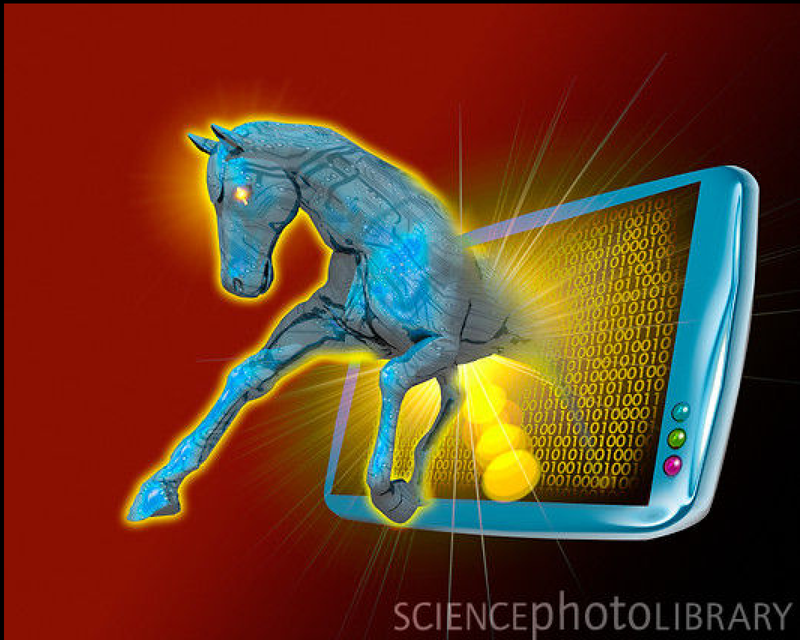
Worm

A computer program designed to replicate itself and propagate. They cause severe network traffic issues when using in large scale.

The computer becomes a zombie computer if the worm is taking most of its resources. They can grant remote access to an attacker.

Trojan

A computer program that pretends to be some other. They can replicate, steal information, or harm the computer system. They usually grant remote access to the sender.



Botnet

Is a set of compromised computers with some malware, which calls-home an awaits further instructions to launch a remote attack to either any computer, or a specific target.

The bots (computers) are also called zombies as they have no brain (they don't attack until they got instructed to do it so).

Spyware

Is a hidden program that collects information from the user, by either reading files, by capturing keystrokes, or some other technique.

On the other hand, they can also be used by the owners of the computer resources for legitime supervision purposes.

Adware

Is a program that renders an advertisement not requested by the user, or by the content generator. In some cases, they add themselves to webpages to generate some revenue. They are usually spyware that masquerade themselves as an ad. For example, Wikipedia has no ads.

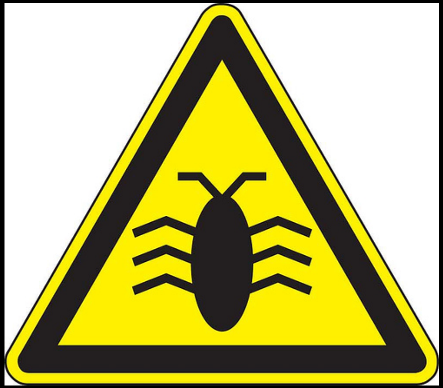
Rootkit

Is a stealthy malicious software that runs with privileged access. In a compromised computer, an attacker inserts a rootkit that hides itself maintaining privileged access to the computer, so that it can use it latter.

A note on malware

A buggy program doing bad thing is not considered a malware, but can be exploited to do some malware actions.

The majority of the malware tends to be either a trojan or a regular computer virus.



Vulnerability

A weakness in the system, that exploited, reduces the assurance of the computer system.

A programming language or library difficult to use has a tendency to provide vulnerabilities.

A program with escalated privileges are common targets. Other programs with access to the previous are also a target.

Why we have vulnerabilities

- Software
  - insufficient testing
  - lack of audit
- Network
  - plain network traffic
  - design issues with the network
- People
  - inadequate training
  - security oblivious
- System management
  - lack of audit
  - no security policy
  - human resource management

Zero-day vulnerabilities

Is an attack over a publicity unknown vulnerability. There is no known way to defend against it, unless it follows an old pattern.

Hackers use zero-day vulnerabilities to enter remote or local systems, gaining privileges on the target system to launch an attack.

Who is susceptible to zero-day?

Every single computer system possibly has a zero-day vulnerability.

The complex the system, the most expensive is to pay for one.

Every malware enters a system via a zero-day vulnerability. Modern spies use their own unknown zero-day vulnerabilities to spy alien or national systems.

Counter measures

The orange book of Trusted Computer System Evaluation Criteria estates several levels of security.

- D - Minimal protection system
- C - Discretionary protection
- B - Mandatory protection
- A - Verified protection.

Level D system

System that have been evaluated but have failed to provide security

Level C system

- Discretionary protection
  - Discretionary security protection: User Authentication, Separation of users and data, DAC - Discretionary access control, system documentation
  - Controlled Access protection: finely grained user authentication and DAC, audit trails, resource isolation, accountability of login process.

Level B system

- Mandatory protection
  - B1 - Labeled security protection: informal security model, data sensitivity labels, Mandatory Access Control over selected objects, design specifications and verification
  - B2 - Structured protection: DAC and MAC enforced to all objects, strict management controls, trusted administration and operator segregation
  - B3 - Security domains: exclude unnecessary code not essential, design for minimize complexity, audit relevant alerts, security administration roles defined, automated IDS/IPS

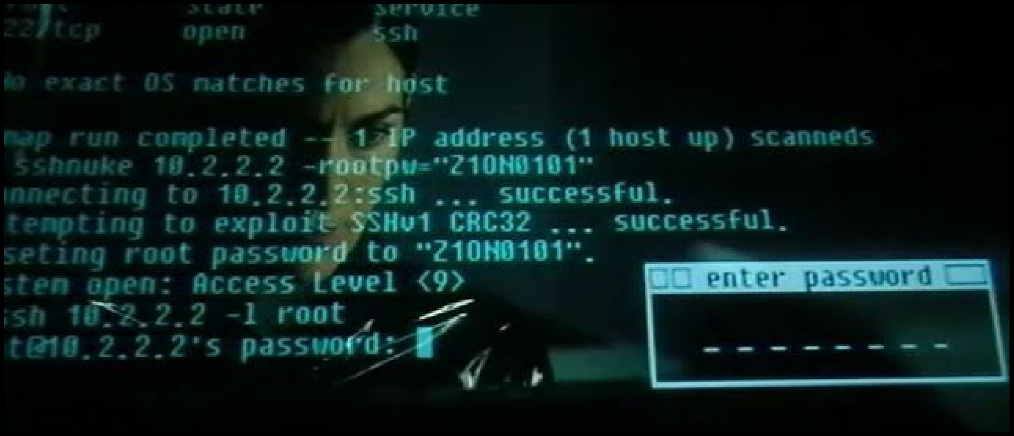
Level A

- Verified protection
  - A1 - Verified design: formal design and verification procedures, formal security management roles
  - Beyond A1: Self-protection systems, trusted design operated by trustable parties only, testing procedures complete from top-to-down including lower-levels specifications.

Silly example on movies

In the matrix sequel, Trinity is seen abusing a SSH vulnerability to gain access to the power grid of the matrix.

Though we can't log in to the matrix in real-life, this hack was used for real elsewhere.



```
        State          Service
22/tcp  open           ssh

o exact OS matches for host

map run completed -- 1 IP address (1 host up) scanneds
Sshnuke 10.2.2.2 -rootpw="Z10N0101"
nnecting to 10.2.2.2:ssh ... successful.
tempting to exploit SSHv1 CRC32 ... successful.
seting root password to "Z10N0101".
stem open: Access Level <9>
sh 10.2.2.2 -l root
t@10.2.2.2's password:
```

The history of malware

sort of

The start of the malware

- The notion for virus was designed by John von Neumann in 1949.
- The first self-replicating automata was created by Veith Risak in 1972, which was written in assembler and rans in SIEMENS 4004/35
- In 1984, Fred Cohen baptized the self-replicating programs as virus. In 1987 he created the first virus detector.
- Also, in 1984, J. B. Gunn described the practical use of viruses.

Famous historical malware

- 1970. *Creeper* appeared in ARPANET, infecting PDP-10 computer
- 1981. *Elk Cloner*, first virus "in the wild". AppleDOS 3.3 using a floppy disk.
- 1986. *Brain boot sector* virus (first IBM virus)
- 1987. *Jerusalem*, attacking every Friday 13th (main attack in May 13th 1988).
- 1988. *Morris*, buffer overrun on Unix.
- 1992. *Michelangelo*, mass psychosis before attacking.

Famous historical malware

- 1995. *Concept*, first macro virus attached on Microsoft word documents (trojan).
- 1996. *Ply*, polymorphic virus.
- 1999. *Happy99* worm. Attack to emails.
- 1999. *Melissa* worm. Deleting MS Office documents
- 1999. *Kak*, javascript worm.
- 2000. *ILOVEYOU* worm, attack on deception
- 2001. *Anna Kournikova*, attack on deception

Famous historical malware

- 2002. *Beast* trojan with back door
- 2003. *SQL slammer* worm
- 2004. *MyDoom*, fastest spreading worm ever.
- 2004. *Sasser* worm, reboot PCs
- 2007. *Storm*, created its own botnet
- 2007. *Zeus*, keylogger Trojan
- 2009. The start of the *cyber-war* era
- 2010. *Stuxnet* worm, SCADA systems
- 2012. *Flammer* worm...

Stuxnet

Originally of unknown source, was a Windows worm that attacked Siemens equipment with the purpose of spying and hijacking industries.

It targeted the Iranian industry soon after concerns of enriching Uranium for military purposes.

http://pastebin.com/a1DeRyFN

Stuxnet

A year after the initial attack, around 1,000 centrifuges less in the electricity facility of Natanz were reported to be working.

The worm modified the operational speed from very fast to slow, and viceversa.

It is currently known as a prototype between some US and Israeli government agencies that went "accidentally" in-the-wild.

Infection

The worm affected Windows equipment with the Siemens software installed.

Then, it attacked PLC systems having the variable-frequency drives configuration

In some memory block it changed the speed of the device at random times, also installing a rootkit to hide himself, and to hide the changes from the monitoring.

Duqu

In 2011, an identical worm was released into-the-wild but as a spyware to catch keystrokes and some other sensitive information.

Both worms are believed to have been created in 2007. Some other variants are out there.

Flame

the basics

Flame

Attacking Windows machines, it was used for cyber espionage in Middle Eastern countries.

It takes screenshots, keystrokes, network activity, it can spy Skype conversations, and use bluetooth connection to gather contact information from nearby devices.

It was remotely ordered auto-destruction once it was discovered.

Installing the attack

Basically, the flame worm used a vulnerability in a windows service used for some of its customers, and by abusing an error in the the configuration of the Windows Servers, allowing an attacker to fake a Windows Update.

Microsoft Security Advisory (2718704)

Unauthorized digital certificates could allow spoofing.

Active attacks using unauthorized digital certificates derived from a Microsoft Certificate Authority. An unauthorized certificate could be used to spoof content, perform phishing attacks, or perform man-in-the-middle attacks. This issue affects **all supported releases** of Microsoft Windows.

About the advisory

In essence, the update revoked a couple of certificates from the Terminal Services Licensing service.

The Terminal service authorizes a computer belonging to the system to use the computer resources of the server: interface, Office, etc.

This allows dumb terminals on low-end or old equipment, or to share a software license.

Why the revoke?

This certificates were only intended to authorize terminal clients, however, they were linked to the same root certificate used to sign, let's say Windows Updates.

In essence, anybody who had a Terminal license could only sign a client certificate in its network, but due to a mistake, these certificates have the for-code flag set up. (So, we can sign updates).

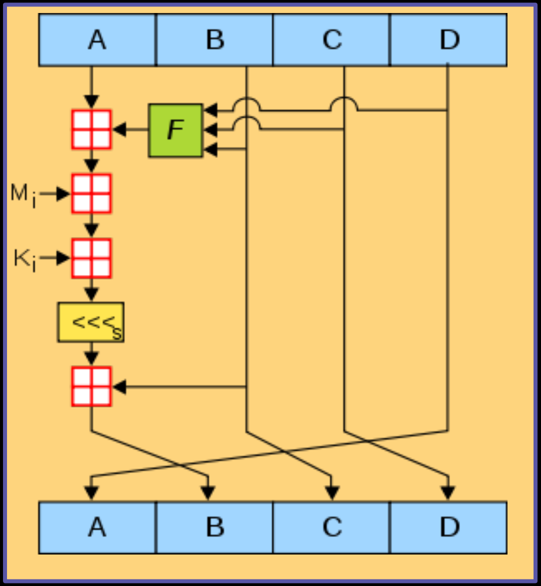
Update on the 2718704 advisory

The Flame malware used a cryptographic collision attack in combination with the terminal server licensing service certificates to sign code as if it came from Microsoft. However, code-signing without performing a collision is also possible.

The attack with collision was possible since the 2009 certificate was using an MD5 hash!

MD5 collision attacks

1996. "The presented attack does not yet threaten practical applications of MD5, but it comes rather close ... in the future MD5 should no longer be implemented...where a collision-resistant hash function is required."



Why the collision?

The Terminal Client License certificates *do* work signing code for Windows version up to Vista.

Current windows version would complain on the certificate extensions of the Terminal server... so, they created a MD5 collision on the certificate to produce one without the extensions!

//so, they can validate the malware as a valid Microsoft software in Windows 7

The cost of the collision

In 2008, a group of scientist found a collision in one day using their PS3 cluster.

Any government agency with enough funds could make a certificate with a collision... but who?

Now, the rest of the old Microsoft certificates will expire in 2020... maybe worth trying!

Details

We now switch to Alex Sotirov's presentation

Analyzing the MD5 collision in Flame

Also, have a look at: http://blog.didierstevens.com/2012/06/04/flame-before-and-after-kb2718704/

Impact of Flame in IT security

Detecting malware

Why is it changing IT security?

Basically, the Flame worm has about two years around on the net, undetected!

Once detected, it killed itself trying to avoid analysis, but some security firm managed to get a copy.

We need *new ways* to detect malware!

Time-to-live

During the Vietnam-USA war, a soldier had an expectancy of life of less than 15 seconds when jumping from a helicopter under fire (in practice they had more if you are not the first copters).

Current threats are malware on the network. How long does an unprotected computer would stay safe before receiving a successful malware attack?

Time before an attack/infection

- In 2003, an unprotected windows computer would stay clean for 40 minutes.
- In 2004, it was reduced to 20 minutes.
- In 2008, a Windows XP would stay safe for 16 minutes (that's the time you have to download and install an antivirus)

Windows 7 is a way better product, it can survive between 40 and 200 minutes (Linux between 400 and 1,400, no data for a Mac)

The important thing

However, that's the time for detectable attacks!

A sophisticated attack like the flame worm was recording conversations in Russia, Iran, Israel, Egypt, Pakistan, Afghanistan, and other countries without anyone noticing it.

This is the most sophisticated piece of malware to date, and we don't know who did it

The obscure thing

Obama ordered to increase attacks on Iran's computers related to nuclear facilities, and Uranium enrichment.

It is known now that the Stuxnet worm that destroyed facilities in Iran and the Flame worm share some code.

A new worm have been found collecting AutoCAD files and chinese emails.

National security

- In the past, Iranian scientist have been killed or have died in obscure circumstances.
- It is known Israel did some of these murders since they feel treated by Iranian forces.
- Same deal with the US and China.

Some think they are protecting their own interests, some think they are pro-active.