Malicious Code

Thierry Sans
Action
- performs unsolicited operations on the system

- **Rabbit** exhausts the hardware resources of a system until failure
- **Backdoor** allows an attacker to take control of the system bypassing authorization mechanisms
- **Spyware** collects information
- **Spamware** uses the system to send spam
- **Ransomware** restricts access to system’s data and resources and demands for a ransom
- **Adware** renders unsolicited advertisement
Dissimulation
- avoid detection by anti-malware programs

**Rootkit** hides the existence of malicious activities
Infection
- penetrate a system and spread to others

Replication
- copy itself to spread

- **Virus** contaminates existing executable programs
- **Worm** exploits a service’s vulnerability

Subterfuge
- based on user’s credulity

- **Trojan Horse** tricks the user to execute the malicious code
Control
- activate the malicious code

• **Backdoor** communicates with command & control servers allowing an attacker to control the virus

• **Logic Bomb** activates the malicious code when certain conditions are met on the system
The history of malicious code
Chronology

- 70's - The era of the first self-replicating programs
- 80's - The era of maturity and first pandemics
- 90's - The era of self-modifying virus and macro viruses
- 00's - The era of Trojan horses and internet worms
- 10's - The era of cyber-warfare viruses
70's - The era of the first self-replicating programs
The era of the first self-replicating programs (70's)

**ANIMAL** (a popular game)
- Replication through the filesystem
- No effect

**Creeper** (and **Reaper**) on Tenex OS (Arpanet)
- Replication through a modem and copied itself to the remote system
- Displays the message
  
  I'M THE CREEPER : CATCH ME IF YOU CAN

**The Rabbit** program
- Replication through the filesystem
- Reduces system performance till crashing
Anatomy of a Virus

A *virus* can be

- a malicious code embedded in an existing program and replicates itself by infecting other programs through the filesystem or the network
- a program that exists by itself and replicates through the filesystem or network

**Infection vector**
how the virus penetrate
the system

**The payload**
what the virus does
80's - The era of maturity and first pandemics
1987 - the beginning of pandemics

**Jerusalem (MS-DOS)**
- Destroys all executable files on infected machines upon every occurrence of Friday the 13th

**SCA (Amiga)**
- Displays a text every 15th boot
- 40% of the Amiga owners were infected

**Christmas Tree EXEC (IBM/PC)**
- Displays a snow flow animation
- Paralyzed several international computer networks in December 1987
The first anti-virus softwares (end of 80's)

**Virus scanner** (detection)
- Signature based -
  Using a signature database of existing viruses
- Behavior based
  Looking for suspicious code patterns that can be used by viruses

**Virus removal tools** (sanitation)
- Cleaning the memory and the filesystem
Avoiding detection

**Cascade** (1987)

- The virus encrypts itself with a cryptographic key and changes this key when replicating itself

✓ Each instance of the virus does not look the same

⇒ This is the emergence of **polymorphic viruses**
90's - The era of self-modifying virus and macros viruses
The era of self-modifying virus (90's)

The Chameleon family (1990)

Ply (1996)

- DOS 16-bit based complicated polymorphic virus with built-in permutation engine
Anatomy of a “polymorphic” virus

A **polymorphic virus** mutates when replicating (but keeps the original algorithm intact)

- By using cryptography
- By injecting garbage code
- By doing permutations within certain instructions or block of instructions
- By using code obfuscation technique

How to detect it?

➡ By detecting code patterns used for the self-modification
Macro Viruses

A **macro virus** is written in scripting languages used by some office applications (can be cross-platform)

- Written in VBS, embedded in a MS-office document, activated when the document is open (autoload function)

**Concept** (1995)

**Melissa** (1999)

- March 26 1999, Melissa shut down e-mail systems that got clogged with infected e-mails
00's - The era of Trojan horses and internet worms
Anatomy of a Trojan horse

A **Trojan horse** is a program that disguise itself as a legitimate program or file.

- In most cases, Trojan horses replicate themselves through emails.
The big stars among trojan horses

VBS/Loveletter ILOVEYOU (2000)
- Caused 5.5 to 10 billion dollars in damage

Sobig (2002)
- Sobig.F set a record in sheer volume of e-mails

MyDoom (2002)
- Broke the record set by Sobig.F
Anatomy of a worm

A **worm** exploits a security flaw (often of a network service) to infect the machine and replicates itself through the network

- Very fast infection (does not need the user to be activated)
- Has a payload as well (more or less harmful)
Factors

- The wide adoption of internet
- The global network is a good medium for virus pandemics
- The multiplication of internet applications and services
- Fast publication of program vulnerabilities
- Slow release of corrective patches
- Slower adoption of these patches (not automatic)
**Code-Red (2001)**

- Exploits a security flaw (buffer overflow) of Microsoft IIS web server (MS01-033) patched one month earlier
- In few days, 359 000 machines infected

**Nimda (2001)**

- Exploits another security flaw of MS-IIS
- The Internet's most widespread worm so far (the most part of the infection was done in 22min)

**Klez (2001)**

- Exploits a security flaw of Microsoft Internet Explorer layout engine used by Outlook and IE
- Infection through email attachment however the user does not have to open this attachment to get infected
**SQL-Slammer** (also called **Sapphire**) (2002)

- Exploits a security flaw in MS-SQL servers for which a patch had been released six months earlier (MS02-039)
- Infected 75,000 machines in 10 minutes causing a massive denial of service and dramatically slowed down global Internet traffic

**Sasser** (2002)

- Exploiting a buffer overflow of Microsoft LSASS on Windows 2000 and XP systems
- Many companies had to shut down their services
**Blaster** (also known as **Lovesan**) (2003)
- Exploits a security flaw in DCOM-RPC services on Windows 2000 and XP
- Was supposed to do SYN flood on August 15, 2003 against port 80 of windowsupdate.com

**Welchia** (also known as **Nachia**) (2003)
- Exploits the same security flaw than Blaster
- Corrects the security flaw by patching the system
**Conficker** (2008)

- Exploits a security flaw in NetBIOS
- Disables auto-update
- Embeds a dictionary password cracker and a backdoor to turn the machine into a “bot”
- Believed to be originated from Ukraine and/or Russia
10's - The era of cyber-warfare malware & Ransomware
The first cyber-warfare virus

**W32.Dozer** (July 2009)

- A virus that created a botnet dedicated to perform a DDoS attack South Korea and US government website on July 4th
- Believed to be originated from China and/or North Korea
**Stuxnet** (Sept 2010)
- A very sophisticated virus that targets SCADA systems (supervisory control and data acquisition)
- Believed that it took down 4000 nuclear centrifuges in Iran
- Believed to be originated from the USA and Israel

**Flame** also called **Skywiper** (May 2012)
- An espionage virus that embeds sophisticated spywares
- Believed to be originated from the USA (Olympic Games defense program)
Another trend - Ransomware

**Reveton** (2012)
- Displays a message from the law enforcement agency saying that you have pirated software and child pornography on your machine
- Ask you to pay a fine using a prepaid cash service

**CryptoLocker** (2013)
- Encrypt specific files on your machine with a 2048 RSA key
- Ask you to pay a ransom with Bitcoins

“Ransomware attacks grew by 500% in 2013 and turned vicious”

source: Symantec Internet Security Threat Report 2014
... and it turned vicious

**WannaCry** and **Petya** (2017)

- Use a vulnerability found in the NSA hacking toolkit leak
- Researcher have found a "kill switch"
- Paralyzed hospitals in UK and trains in Germany
Late 10's - the emergence of IoT malware and Cryptominers

**Mirai** (2016)
- Infects IoT devices
- Most powerful DDoS attacks to date

**Coinhive** (2018)
- Javascript embedded in website (either legitimately or not) and popular malware as well
Modern Malicious Code

Thierry Sans
How to create a new malware? 3 step process

1. **Create the malware’s payload**

2. Make the malware undetectable

3. Spread the malware
The malware payload

• take control of the victim’s device turning it into a **zombie/bot**

• act as a **spam relay** or **DDoS relay**

• steal **personal information**
  including passwords, credit card numbers, banking credentials

• **click bot**: generating web traffic

• **Cryptominer**

• **Ransomware**

• … and so on
Remote Administration Tool/ Trojan (RAT)

Basically a **remote administration tool** with

- stealth features
- and specific functionalities such as:
  - camera controller
  - hardware destroyer
  - password / credit card loggers
  - ... and so on
Buy a RAT as a COTS*

Some RAT Builders

- **Zeus** (2007) initially $700, now open source
- **DarkComet** (2008), open source
- **BlackShades** (2010) can now be purchased from an official company $49 - $56

* Commercial Off-The-Shelf
Are we done yet?

SHA256: 85ff1fc03614302ae5bf779b454b163364a1c051f925dbb360e8fcfa12fc6a3
File name: DarkCometRAT531.zip
Detection ratio: 42 / 50
Analysis date: 2014-04-27 11:56:13 UTC (5 hours, 26 minutes ago)

VirusTotal Analysis

<table>
<thead>
<tr>
<th>Antivirus</th>
<th>Result</th>
<th>Update</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVG</td>
<td>Generic23.AVVP.dropper</td>
<td>20140427</td>
</tr>
<tr>
<td>Ad-Aware</td>
<td>Trojan.Generic.KDV.388330</td>
<td>20140427</td>
</tr>
<tr>
<td>Agritum</td>
<td>HackTool.Binderluc8D13KnW4U</td>
<td>20140427</td>
</tr>
<tr>
<td>AntiVir</td>
<td>SPR/Binder.bs.1</td>
<td>20140426</td>
</tr>
<tr>
<td>Antiy-AVL</td>
<td>HackTool/Win32.Binder</td>
<td>20140427</td>
</tr>
<tr>
<td>Avast</td>
<td>Win32.Malware-gen</td>
<td>20140427</td>
</tr>
</tbody>
</table>
How to create a new malware? 3 step process

1. Create the malware’s payload

2. Make the malware undetectable a.k.a packing a malware

3. Spread the malware
How antiviruses detect malware? 2 techniques

1. **Static Analysis**
   - Scan program comparing it to a collection of signatures

2. **Dynamic Analysis**
   - Run program in a sandbox and infer from its behavior
Malware Crypter

- Encryption
- Code obfuscation
- Stealth mode to detect and bypass sandbox
**Buy a Crypter as a COTS**

Some available Crypters

- **Byte Crypter** $35 for 3 months, $60 for lifetime
- **Datascrambler** $20 for 3 months, $40 for a year
- **BlackShades Crypter** from an official company $60 for 3 months, $100 for a year
How to create a new malware? 3 step process

1. Create the malware’s payload
2. Make the malware undetectable

3. Spread the malware
Spread the malware using social engineering

➤ Trick people to download and install the malware
- tutorial about hacking that makes you install the malware
- video/chat player to access exclusive content or talk to exclusive people
- pirated software on P2P networks
Spread the malware using through a webpage

- Exploit a browser/plugin vulnerability to automatically download and install the malware on the victim’s device.
Exploits bundle and other services

1. **Exploit bundle** : $25/day, $400/month, up to $3,000
   ➡ program to embed into a webpage

2. **Bulletproof host** : $15–250 per month
   ➡ hosting service to bypass any kind of IP filtering anti-spam, anti-virus, anti-malware, law enforcement, search engine anti-malware service and so on

3. **Traffic** : $4–10 per 1,000 unique hits
   ➡ attract people to visit the infected webpage

**Installs** $12 – $550 per 1,000 infections
   ➡ use a spreading service also called Pay-Per-Install (PPI)
Conclusion

Creating a malware, making it undetectable and spreading it would normally be difficult and require a good deal of expertise.

However, the cyber underground market makes this process accessible to the mass given a small amount of money.