ECE 4309

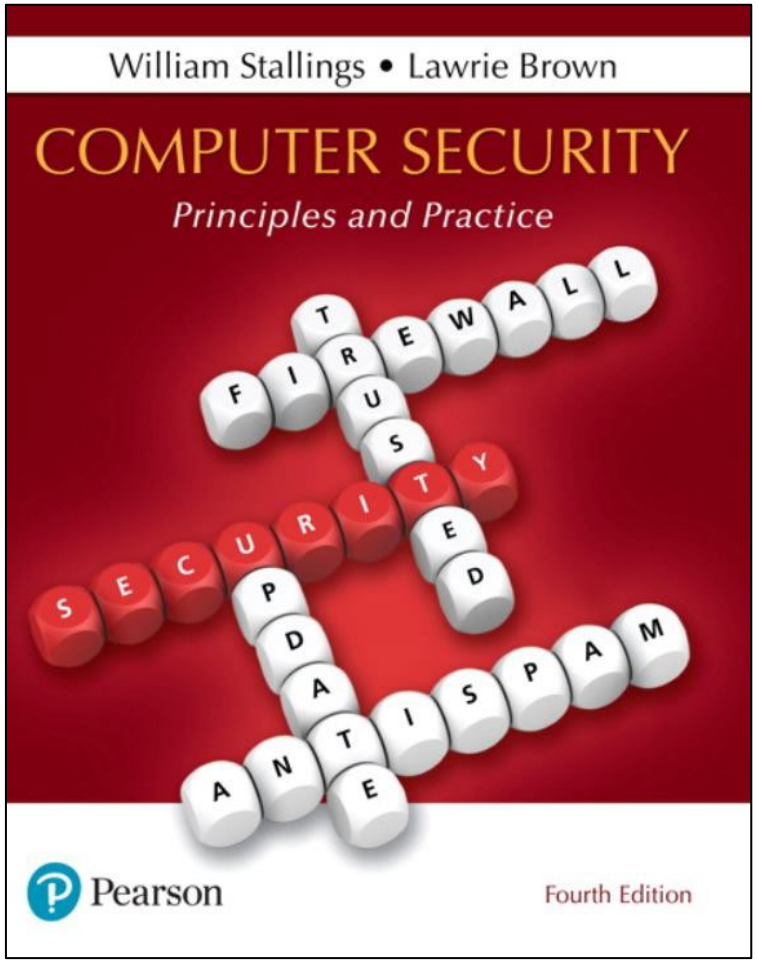
Malwares – part 1

Dr. Valerio Formicola

CalPolyPomona

Computer Security: Principles and Practice

Fourth Edition



Chapter 6

Malicious Software

Malware

Malicious software, or **malware**

NIST 800-83 defines malware as:

"a program that is inserted into a system, usually covertly, with the intent of compromising the confidentiality, integrity, or availability of the victim's data, applications, or operating system or otherwise annoying or disrupting the victim."

Table 6.1 Malware Terminology (1 of 3)

Name	Description
Advanced Persistent Threat (APT)	Cybercrime directed at business and political targets, using a wide variety of intrusion technologies and malware, applied persistently and effectively to specific targets over an extended period, often attributed to state-sponsored organizations.
Adware	Advertising that is integrated into software. It can result in pop-up ads or redirection of a browser to a commercial site.
Attack kit	Set of tools for generating new malware automatically using a variety of supplied propagation and payload mechanisms.
Auto-rooter	Malicious hacker tools used to break into new machines remotely.
Backdoor (trapdoor)	Any mechanism that bypasses a normal security check; it may allow unauthorized access to functionality in a program, or onto a compromised system.
Downloaders	Code that installs other items on a machine that is under attack. It is normally included in the malware code first inserted on to a compromised system to then import a larger malware package.
Drive-by-download	An attack using code on a compromised website that exploits a browser Vulnerability to attack a client system when the site is viewed.
Exploits	Code specific to a single vulnerability or set of vulnerabilities.

Table 6.1 Malware Terminology (2 of 3)

Name	Description
Flooders (DoS client)	Used to generate a large volume of data to attack networked computer systems, by carrying out some form of denial-of-service (DoS) attack.
Keyloggers	Captures keystrokes on a compromised system.
Logic bomb	Code inserted into malware by an intruder. A logic bomb lies dormant until a Predefined condition is met; the code then triggers some payload.
Macro virus	A type of virus that uses macro or scripting code, typically embedded in a Document or document template, and triggered when the document is viewed or edited, to run and replicate itself into other such documents.
Mobile code	Software (e.g., script and macro) that can be shipped unchanged to a heterogeneous collection of platforms and execute with identical semantics.
Rootkit	Set of hacker tools used after attacker has broken into a computer system and gained root-level access.
Spammer programs	Used to send large volumes of unwanted e-mail.
Spyware	Software that collects information from a computer and transmits it to another system by monitoring keystrokes, screen data, and/or network traffic; or by scanning files on the system for sensitive information.

Table 6.1 Malware Terminology (3 of 3)

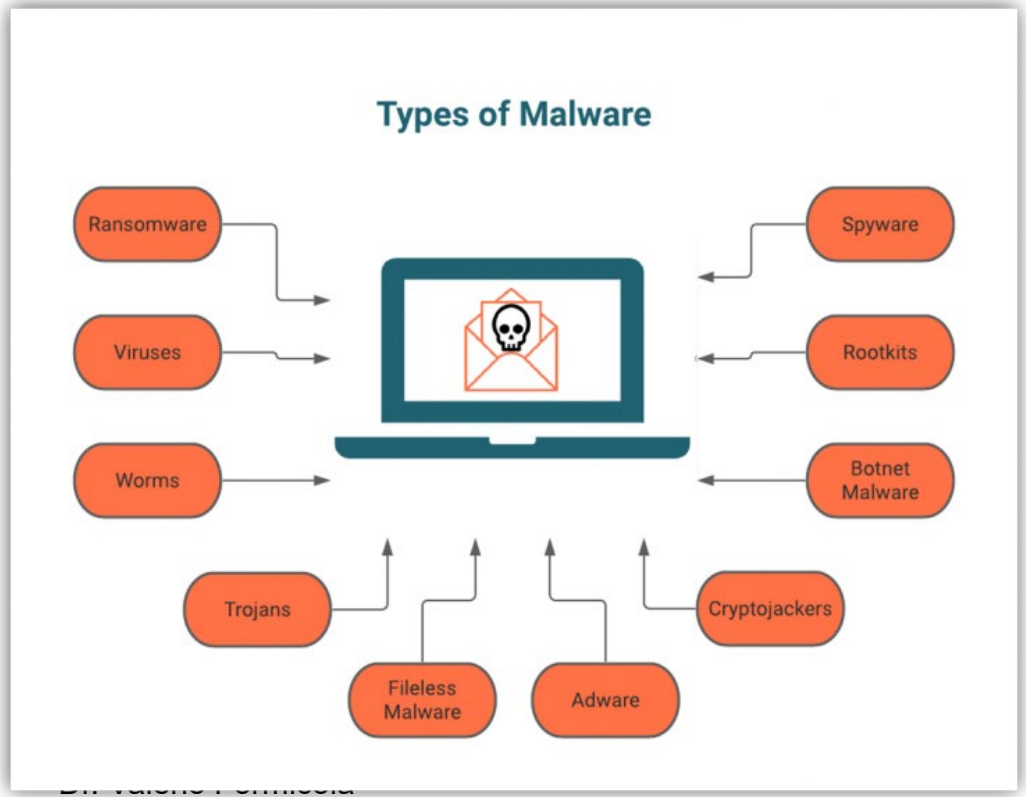
Name	Description
Trojan horse	A computer program that appears to have a useful function, but also has a hidden and potentially malicious function that evades security mechanisms, sometimes by exploiting legitimate authorizations of a system entity that invokes it.
Virus	Malware that, when executed, tries to replicate itself into other executable machine or script code; when it succeeds, the code is said to be infected. When the infected code is executed, the virus also executes.
Worm	A computer program that can run independently and can propagate a complete working version of itself onto other hosts on a network, by exploiting software vulnerabilities in the target system, or using captured authorization credentials.
Zombie, bot	Program installed on an infected machine that is activated to launch attacks on other machines.

Classification of Malware

- Categories used for classification:
  - **Propagation Method:** Based first on how it spreads or propagates to reach the desired targets
  - **Actions on the Target:** Actions or payloads it performs once a target is reached

Additional criteria for classification:

- The need for a host program (parasitic code such as viruses)
- Independent, self-contained programs (worms, trojans, and bots)
- Malware that does not replicate (trojans and spam e-mail)
- Malware that does replicate (viruses and worms)



Types of Malicious Software (Malwares)

- Propagation mechanisms include:
  1. Infection of existing content by viruses that is subsequently spread to other systems (*user-activated replication*)
  2. Exploit of software vulnerabilities by worms or drive-by-downloads to allow the malware to replicate (*malware self-replication*)
  3. Social engineering attacks that convince users to bypass security mechanisms to install Trojans or to respond to phishing attacks (*user-induced replication*)
- Payload actions performed by malware once it reaches a target system can include:
  1. Corruption of system or data files
  2. Theft of service/make the system a zombie agent of attack as part of a botnet
  3. Theft of information from the system/keylogging
  4. Stealthing/hiding its presence on the system

Creation of new malwares: Attack Kits

- Initially the development and deployment of malware required considerable technical skill by software authors
  - The development of virus-creation toolkits in the early 1990s and then more general attack kits in the 2000s greatly assisted in the development and deployment of malware
- Toolkits are often known as "crimeware"
  - Include a variety of propagation mechanisms and payload modules that even novices can deploy
  - Variants that can be generated by attackers using these toolkits creates a significant problem for those defending systems against them
- Examples are:
  - Zeus
  - Angler

Note: *Script kiddies* are hackers that use attack kits or, even with less effort, programs generated by pre-configured kits

Diffusion of malwares

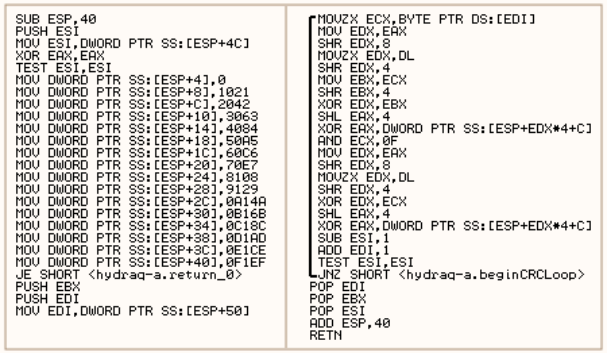
- As more capable and financially funded hackers exist, more and more actors can generate malware
  - Remember: Cyber-Criminals, State-sponsored, hobbyists, Organizations that sell their services to companies and nations

- This has significantly changed the resources available and motivation behind the rise of malware and has led to development of a large underground economy involving the sale of attack kits, access to compromised hosts, and to stolen information

Viruses

CalPolyPomona

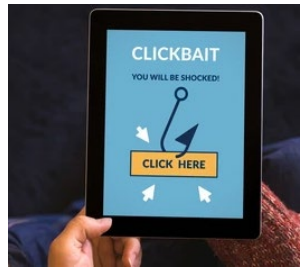
Viruses



```
SUB ESP,40
PUSH ESI
MOV ESI,DWORD PTR SS:[ESP+4C]
XOR EAX,EAX
TEST ESI,ESI
MOV DWORD PTR SS:[ESP+4],0
MOV DWORD PTR SS:[ESP+8],1021
MOV DWORD PTR SS:[ESP+C],2042
MOV DWORD PTR SS:[ESP+10],3063
MOV DWORD PTR SS:[ESP+14],4084
MOV DWORD PTR SS:[ESP+18],50A5
MOV DWORD PTR SS:[ESP+1C],60C6
MOV DWORD PTR SS:[ESP+20],70E7
MOV DWORD PTR SS:[ESP+24],8108
MOV DWORD PTR SS:[ESP+28],9129
MOV DWORD PTR SS:[ESP+2C],0A14A
MOV DWORD PTR SS:[ESP+30],0B16B
MOV DWORD PTR SS:[ESP+34],0C18C
MOV DWORD PTR SS:[ESP+38],0D1AD
MOV DWORD PTR SS:[ESP+3C],0E1CE
MOV DWORD PTR SS:[ESP+40],0F1EF
JE SHORT <hydraq-a.return_0>
PUSH EBX
PUSH EDI
MOV EDI,DWORD PTR SS:[ESP+50]
MOVZX ECX,BYTE PTR DS:[EDI]
MOV EDX,EAX
SHR EDX,8
MOVZX EDX,DL
SHR EDX,4
MOV EBX,ECX
SHR EBX,4
XOR EDX,EBX
SHL EAX,4
XOR EAX,DWORD PTR SS:[ESP+EDX*4+C]
AND ECX,0F
MOV EDX,EAX
SHR EDX,8
MOVZX EDX,DL
SHR EDX,4
XOR EDX,ECX
SHL EAX,4
XOR EAX,DWORD PTR SS:[ESP+EDX*4+C]
SUB ESI,1
ADD EDI,1
TEST ESI,ESI
JNZ SHORT <hydraq-a.beginCRCLoop>
POP EDI
POP EBX
POP ESI
ADD ESP,40
RETN
```

- Parasitic software fragments: infect programs to be executed, i.e., pieces of instructions that cannot run as an individual program, but they depend on another program (a "host") to be executed
  - Modifies them to include a copy of the virus
  - Replicates and goes on to infect other content
  - Easily spread through network environments
- When attached to an executable program a virus can do anything that the program is permitted to do
  - Executes secretly when the host program is run
  - Also, it targets replication (as biological viruses) by leveraging a "logical" contact between infected and new victim. Logical means that the contact has to happen according to the transmission mechanism of the virus (e.g., USB port, email, Word document, file exchange, etc.)
- In general, specific to operating system, hardware or application
  - Because it's a software and it is coded to exploit a vulnerability on a specific version of systems, applications or libraries, it takes advantage of their details and weaknesses
  - For example, a virus for one specific version of Microsoft Windows or Office

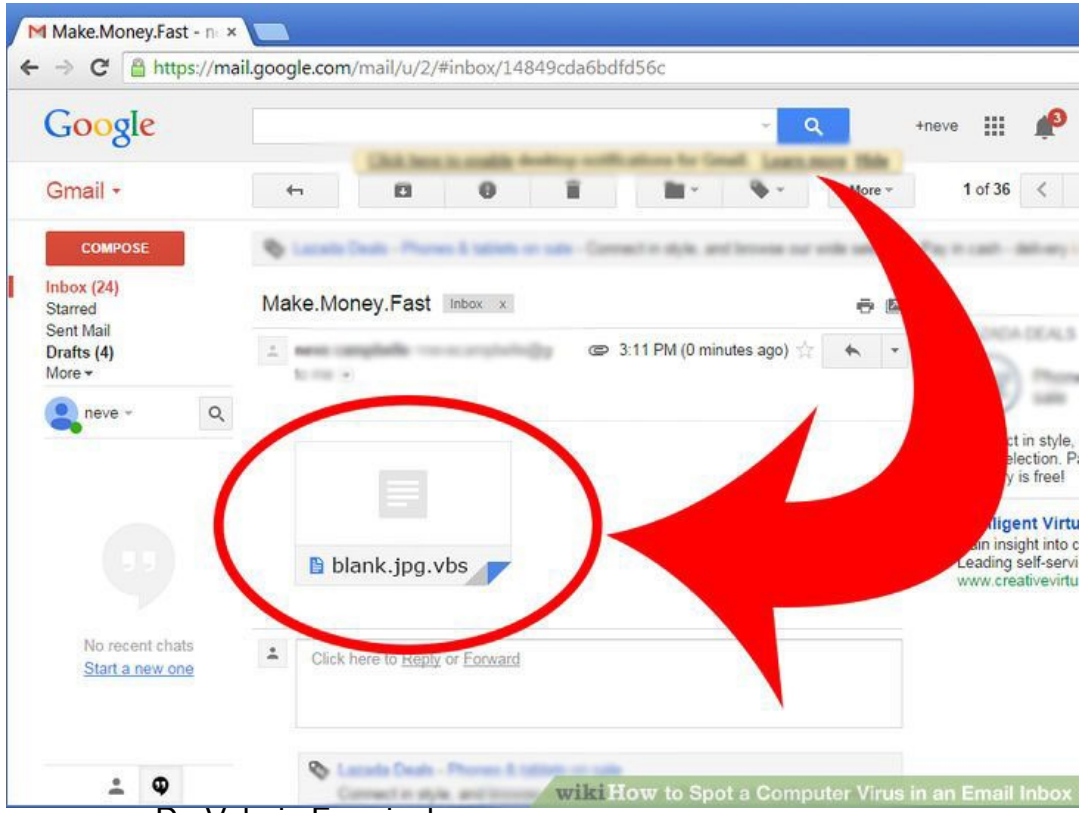
Main characteristic of a virus: user-based activation required



- A computer virus lives within a host, such as a document or executable file, and **requires human interaction to spread.**

Example:
You receive an email (that you're not expecting) with an intriguing (clickbait) title like "Made some changes — please check." Attached to the email is a file with a name like "Updates" — it may be a DOC or EXE file.

- If it's a DOC file, once you download it, you'll be prompted to enable macros (programmed rules that help simplify repetitive tasks). This action triggers the virus.
- If the file is an EXE, downloading it and running it triggers the virus. The virus will then commandeer your computer's resources to copy itself and spread, damaging your devices and files or stealing your personal data.
- If it's an advertisement (*clickbait*), you might click attracted by some clickbait and be redirected to an infected website.



Virus Components

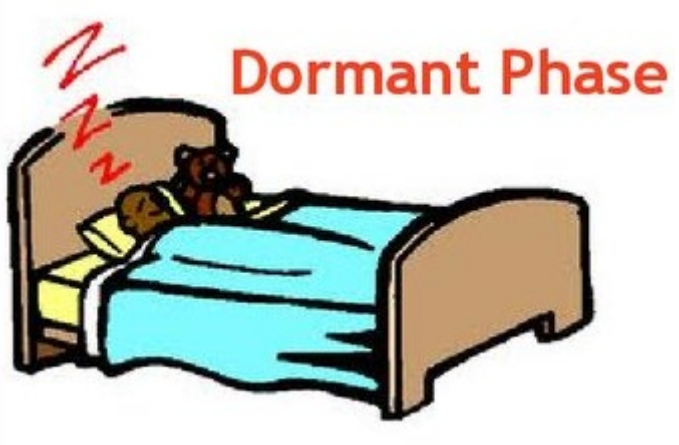
1. **Infection mechanism**
   - Means by which a virus spreads or propagates: Initially they were executable instructions, then more and more scripts for Active contents, like documents (e.g., Word, Excel, PDF macros)
   - Also referred to as the **infection vector**
   - *Usually, contained in a code dedicated to produce the infection of new victims or download the payload* ***(Infector code)***
2. **Payload**
   - What the virus does (besides spreading): Some viruses just copy themselves from one computer to other. Other viruses may steal data or files, permit eavesdropping or unauthorized access, destroy data and cause other consequences. It is also possible for a virus to carry multiple payloads.
   - May involve damage or benign but noticeable activity
   - *Usually, contained in a code dedicated to execute the destructive action on the infected victim* ***(Payload code)***
3. **Trigger**
   - Event or condition that determines when the payload is activated or delivered
   - Sometimes known as a **logic bomb:** e.g., a date (aka **time bomb**), the introduction of another file in the system, some metrics of the system like resources used exceeding some threshold
   - *Usually, contained in a code dedicated to indicate the logic of triggering* ***(Trigger code)***

Examples of actions done by virus payloads (i.e., impact)

- **Data theft:** Particularly common is the theft of sensitive information such as login credentials or financial information through various forms of data breaches.
- **Activity monitoring:** An executed malicious payload may serve to monitor user activity on a computer, this can be done for the purposes of spying, blackmail, or to aggregate consumer behavior which can be sold to advertisers.
- **Displaying advertisements:** Some malicious payloads work to display persistent, unwanted ads such as pop-ups and pop-unders to the victim.
- **Deleting or modifying files**: This is one of the most serious consequences to arise from a malicious payload. Files can be deleted or modified to either affect the behavior of a computer, or even disable the operating system and/or startup processes. For example, some malicious payloads are designed to 'brick' smartphones, meaning they can no longer be turned on or used in any way.
- **Downloading new files:** Some malicious payloads come in very lightweight files that are easy to distribute, but once executed they will trigger the download of a much larger piece of malicious software.
- **Running background processes:** A malicious payload can also be triggered to quietly run processes in the background, such as cryptocurrency mining or data storage.
- **Enable backdoor access:** from externals to a user's computer

Virus Phases (1 of 2)

- Dormant phase
  - Virus is idle
  - Will eventually be activated by some event
  - Not all viruses have this stage
- Triggering phase
  - Virus is activated to perform the function for which it was intended
  - Can be caused by a variety of system events



Virus Phases (2 of 2)

- Propagation phase
  - Virus places a copy of itself into other programs or into certain system areas on the disk
  - May not be identical to the propagating version
  - Each infected program will now contain a clone of the virus which will itself enter a propagation phase
- Execution phase
  - Function is performed
  - May be harmless or damaging

Virus Classifications

There is no unique model,
but most commonly the attributes
are like in the figure

Classification by target, i.e., infected objects

- **Boot sector infector**
  - Infects a master boot record or boot record and spreads when a system is booted from the disk containing the virus. These reside inside the boot sector of a drive or storage media.
- **File infector**
  - Infects files that the operating system or shell considers to be executable
- **Macro virus**
  - Infects files with macro or scripting code that is interpreted by an application
- **Multipartite virus**
  - Infects files in multiple ways

Classification by hiding strategy (camouflage)

- **Encrypted virus**
  - A portion of the virus creates a random encryption key and encrypts the remainder of the virus
  - Key is changed in each replication, so it's hard to detect using regular bit sequences
- **Stealth virus**
  - A form of virus explicitly designed to hide itself from detection by anti-virus software
- **Oligomorphic virus**
  - Oligomorphism is an advanced form of the encryption. It contains a collection of different decryptors, which are randomly chosen for a new victim. In such a way, the decryptor code is not identical in various instances. T
- **Polymorphic virus**
  - A virus that mutates with every infection. For example, add random instructions between more targeted instructions (i.e., the instructions that replicate or execute the payloads)
- **Metamorphic virus**
  - A virus that mutates and rewrites itself completely at each iteration and may cha appearance

Note: some viruses might use a combination of the techniques above to hide

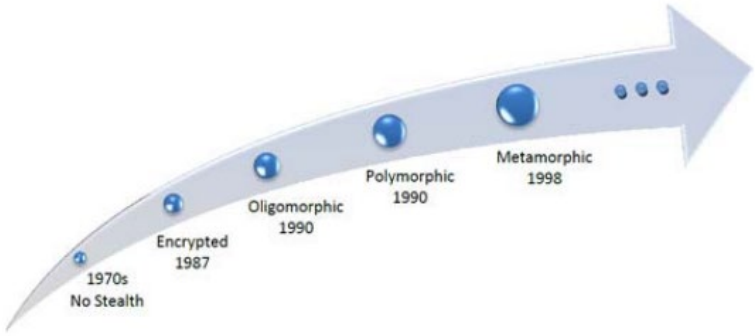


Figure 1: Evolution timeline of Camouflage techniques appearance in malware

Dr

Encrypted Virus

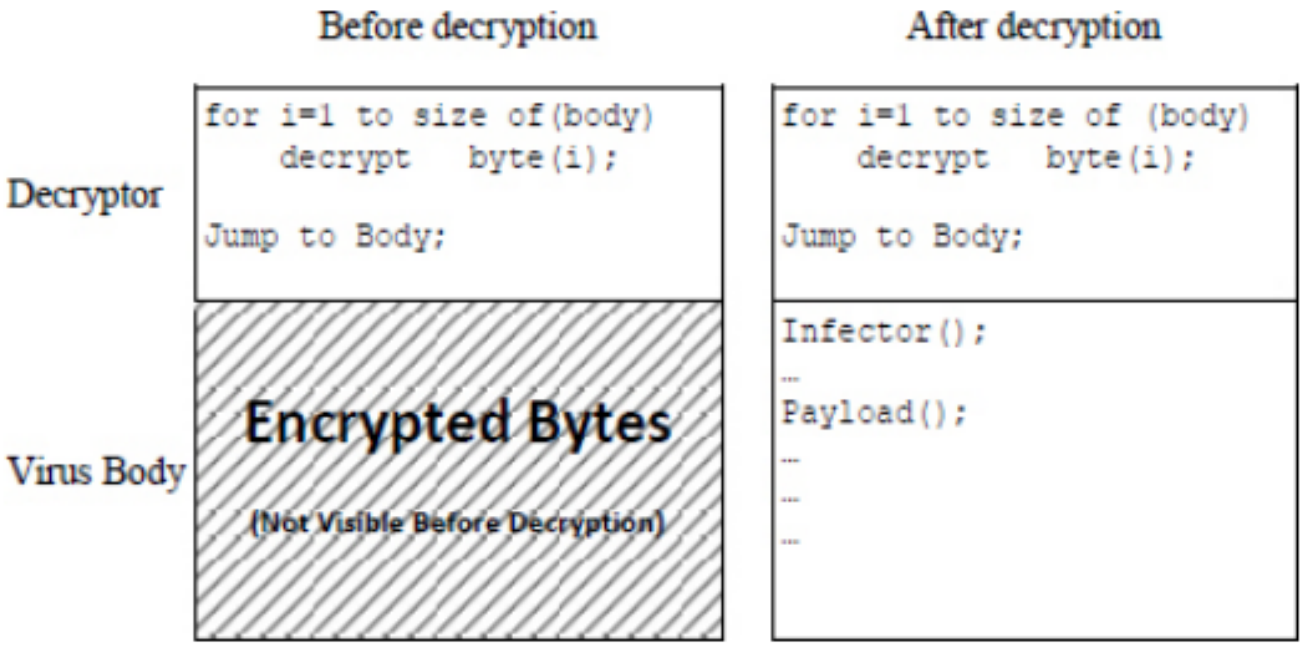


Figure 2: Structure of encrypted virus

Oligomorphic virus

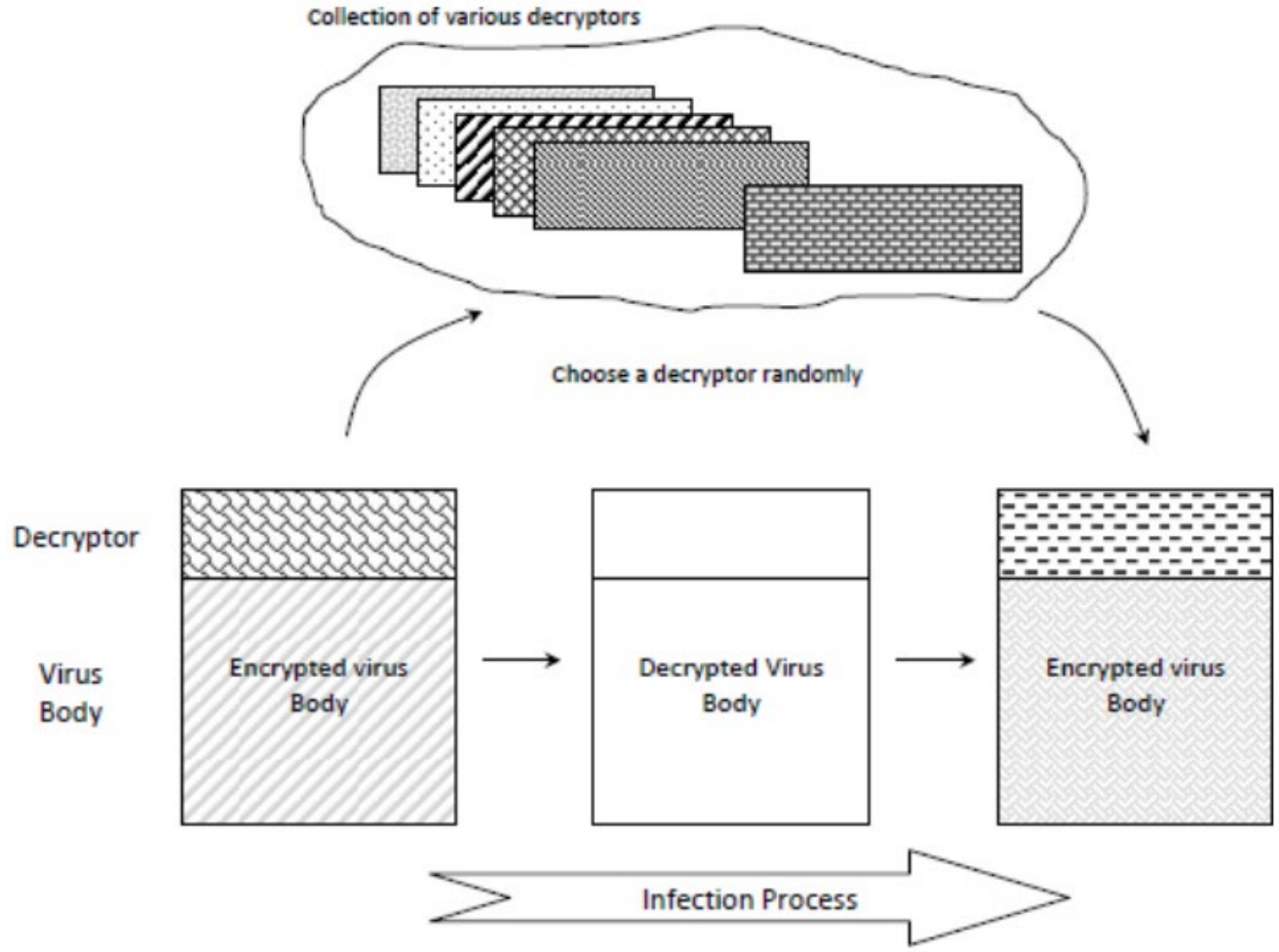
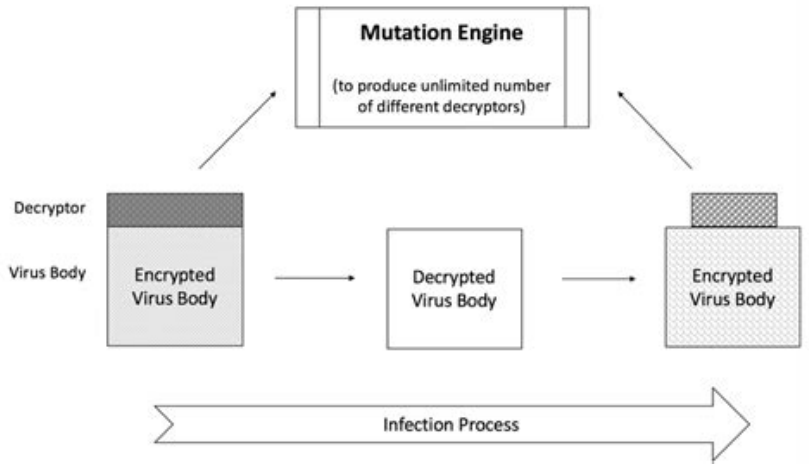
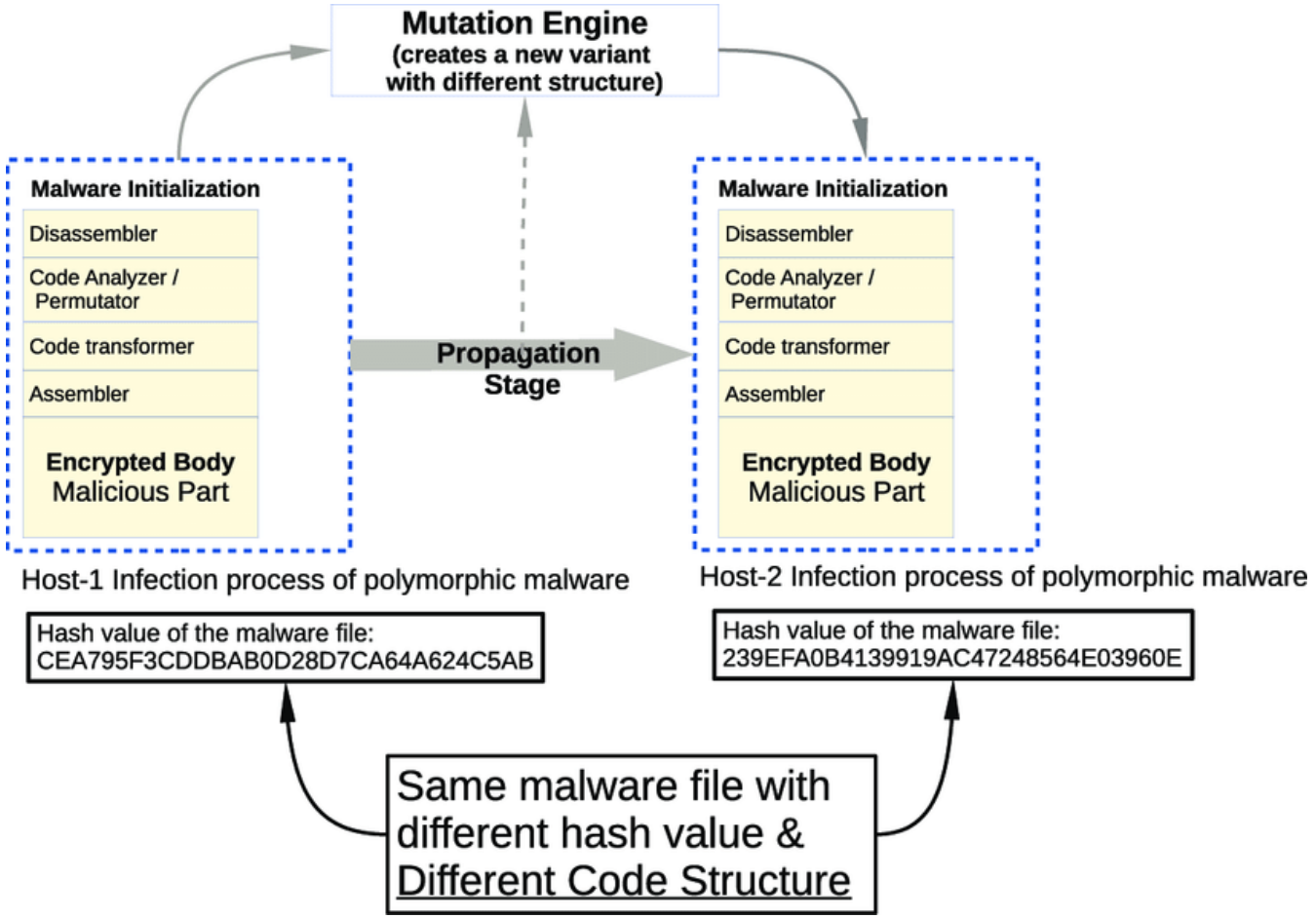


Figure 3: Structure and mechanism of oligomorphic virus

Polymorphic virus





Metamorphic virus

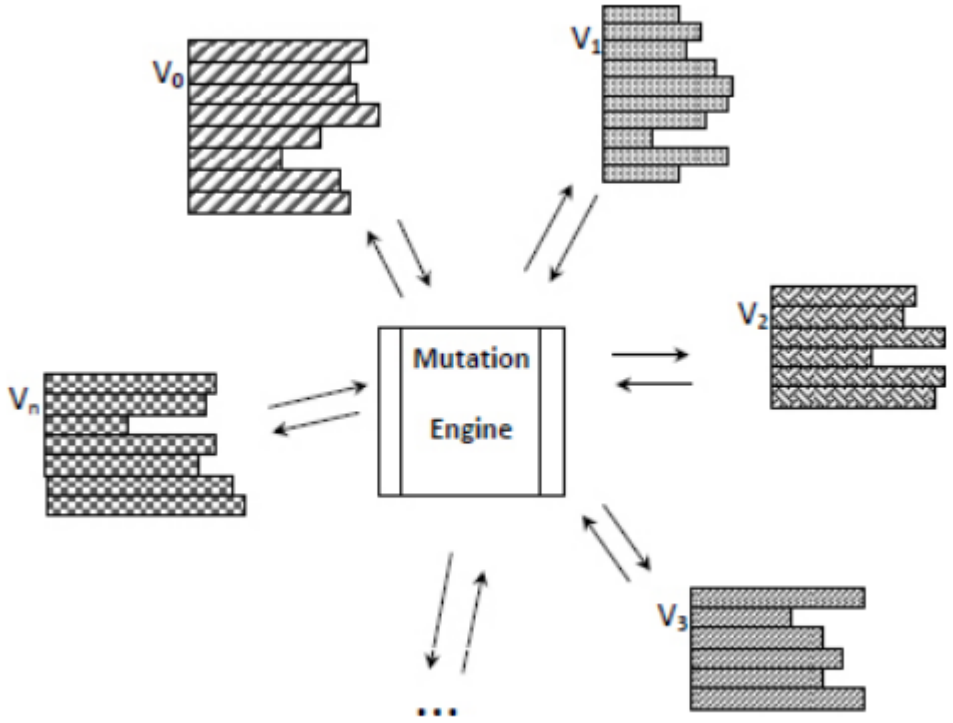


Figure 5: Metamorphic virus propagation scheme

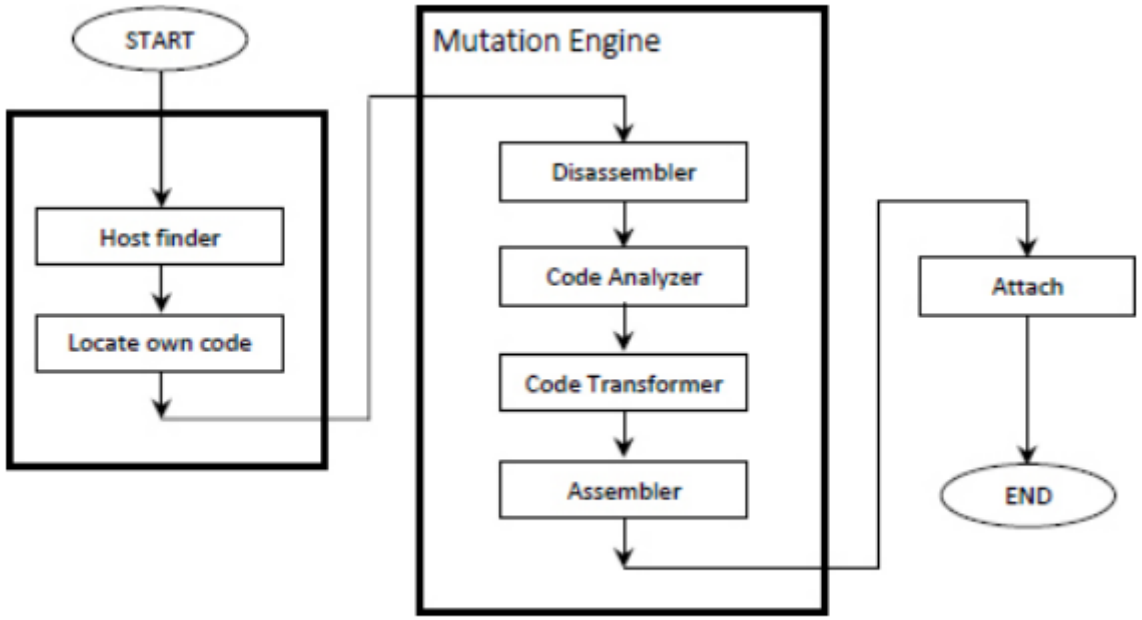
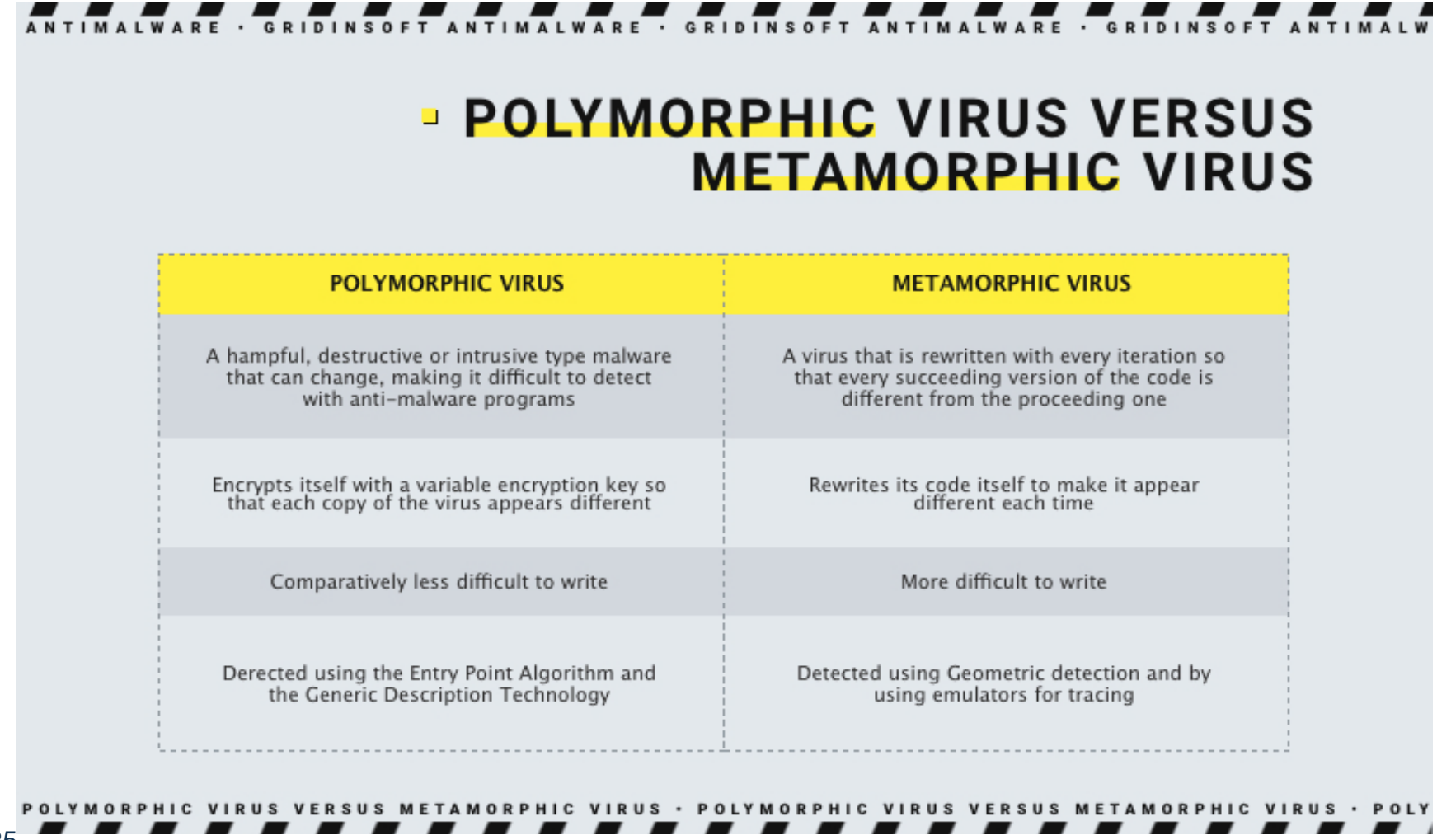


Figure 6: Structure of replicator and mutation engine in metamorphic virus

Metamorphic virus



Stealth virus

- A computer virus that avoids detection by hiding itself after infecting the machine. The stealth virus may hide in different locations such as **boot sectors (e.g., Master Boot Sector MBR),** various files, and **undetectable computer areas**, effectively tricking anti-virus software
- In order to avoid detection, stealth viruses also self-modify in the following ways:
  - Code Modification: The stealth virus changes the code and virus signature of each infected file.
  - Encryption: The stealth virus encrypts data via simple encryption and uses a different encryption key for each infected file.

Table 3: Two versions of W32.Evol

Binary Code Sequence	Assembly Code	
C7060F000055	mov	[esi], 5500000Fh
C746048BEC5151	mov	[esi+0004],5151EC8Bh
String Signature:	C7060F000055C746048BEC5151	

Binary Code Sequence	Assembly Code	
BF0F000055	mov	edi,5500000Fh
893E	mov	[esi],edi
5F	pop	edi
52	push	edx
B640	mov	dh,40
BA8BEC5151	mov	edx,5151EC8Bh
53	push	ebx
8BDA	mov	ebx,edx
895E04	mov	[esi+0004],ebx
String Signature:	BF0F000055893E5F52B640BA8BEC5151538BDA895E04	

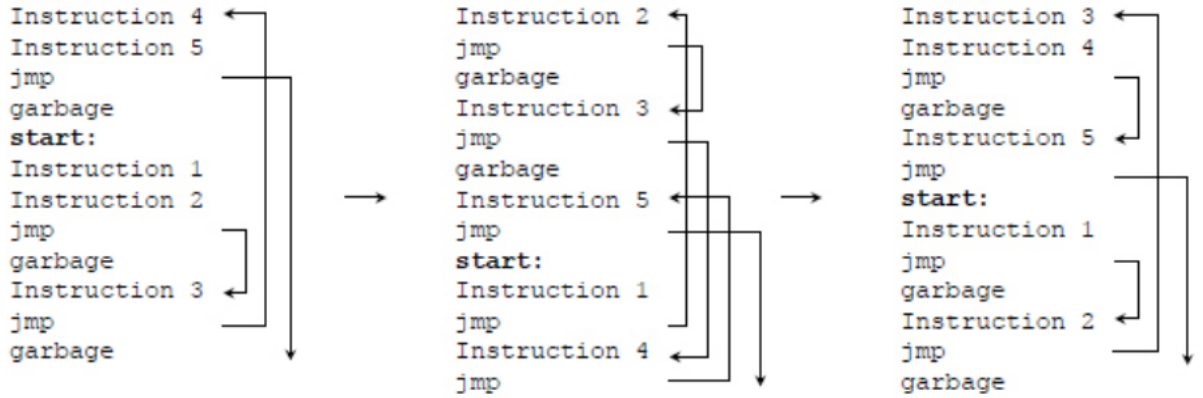
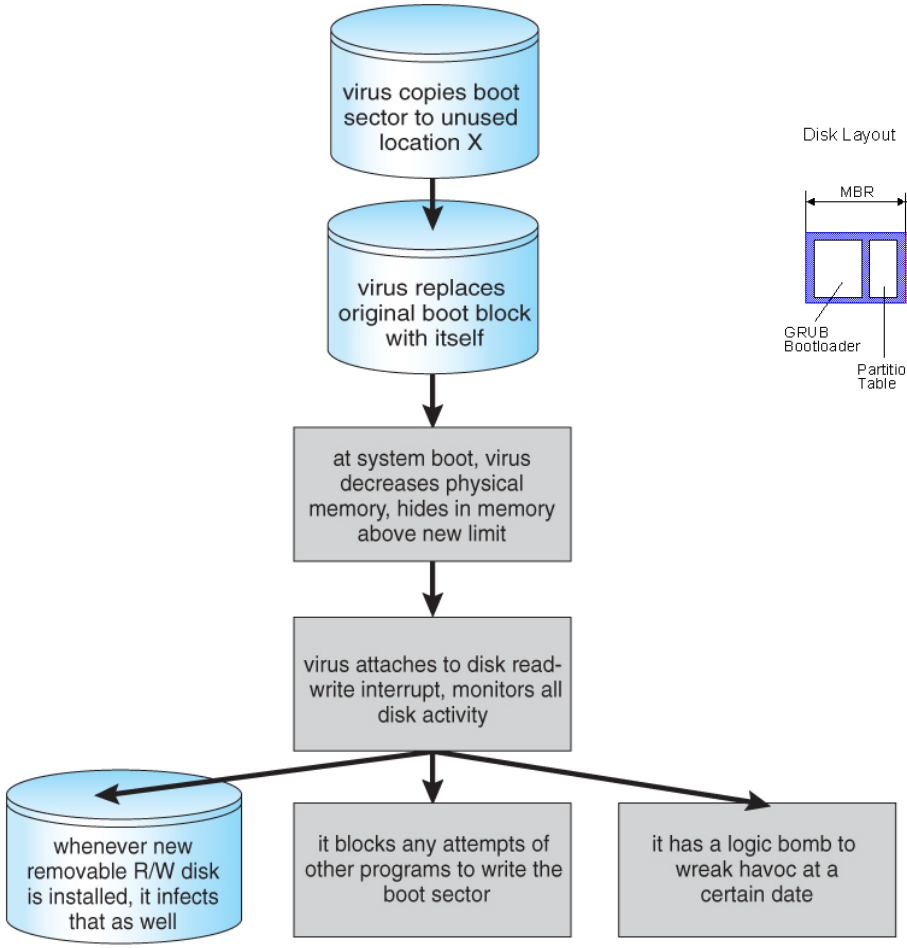
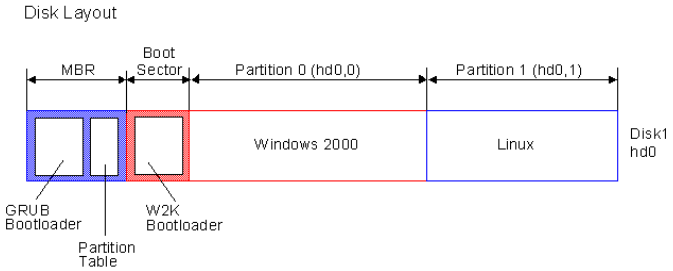
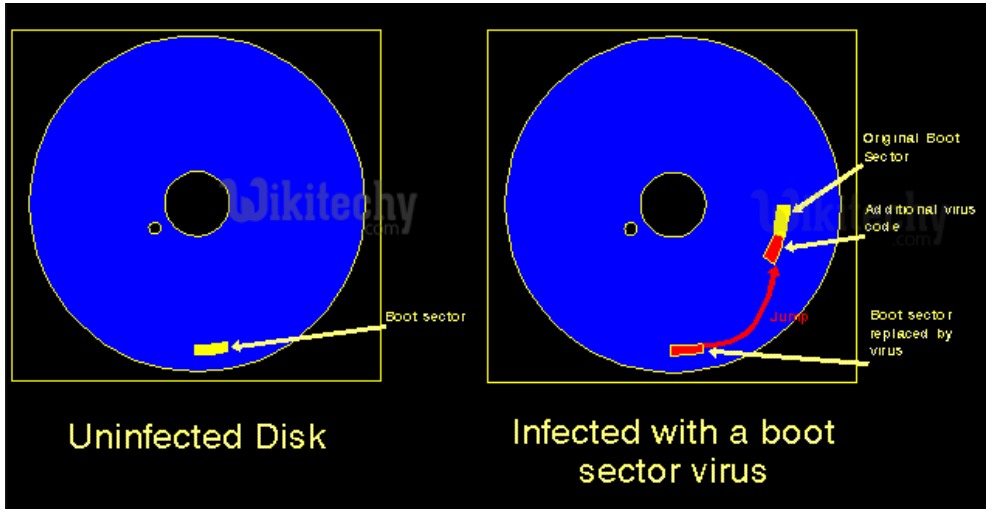


Figure 7: Code transposition in Zperm virus

Example: infecting and hiding in the boot sector of disk








```
.text:100015C9
.text:100015C9
.text:100015C9 0F B6 84 0D 74 FF FF FF
.text:100015D1 33 D2
.text:100015D3 6A 3A
.text:100015D5 5F
.text:100015D6 F7 F7
.text:100015D8 41
.text:100015D9 8A 82 4C FF 00 10
.text:100015D9
.text:100015D9
.text:100015D9
.text:100015DF 88 44 0D AF
.text:100015E3 83 F9 3C
.text:100015E6 72 E1
.text:100015E8 8D 85 68 FA FF FF
.text:100015EE 50
.text:100015EF 8D 85 68 FE FF FF
.text:100015F5 50
.text:100015F6 E8 DA FC FF FF
.text:100015FB A3 F8 F8 01 10
.text:10001600 3B C6
.text:10001602 0F 8C 8D 02 00 00
.text:10001608 6A 04
.text:1000160A 8D 8D 2E FC FF FF
.text:10001610 5F
.text:10001611 8B D6
```

```
loc_100015C9:                           ; CODE XREF: overwrite_mbr_func+13D↓j
        movzx   eax, [ebp+ecx+random_bytes]
        xor     edx, edx
        push    3Ah
        pop     edi
        div     edi
        inc     ecx
        mov     al, byte ptr ds:a123456789abcdefghjklmnpq[edx]
                                        ; DATA XREF: sub_100166F0+1DF↓r
                                        ; sub_100166F0:loc_100168FC↓r
                                        ; "123456789ABCDEFGHJKLMNPQRSTUVWXYZabcdef"...
        mov     [ebp+ecx+var_51], al
        cmp     ecx, 3Ch
        jb      short loc_100015C9
        lea     eax, [ebp+mbr_sector]
        push    eax             ; Dst
        lea     eax, [ebp+FileName]
        push    eax             ; lpFileName
        call    read_bytes_From_file
        mov     function_check, eax
        cmp     eax, esi
        jl      loc_10001895
        push    4
        lea     ecx, [ebp+LBA]
        pop     edi
        mov     edx, esi
```

Jump instruction in MBR

Petya malware, 2016

Macro and scripts viruses

Macro and Scripting Viruses

- NISTIR 7298 defines a macro virus as:
  - "a virus that attaches itself to documents and uses the macro programming capabilities of the document's application to execute and propagate"
- Macro viruses infect scripting code used to support active content in a variety of user document types
- Are threatening for a number of reasons:
  - Is platform independent
  - Infect documents, not executable portions of code
  - Are easily spread
  - Because they infect user documents rather than system programs, traditional file system access controls are of limited use in preventing their spread, since users are expected to modify them
  - Are much easier to write or to modify than traditional executable viruses

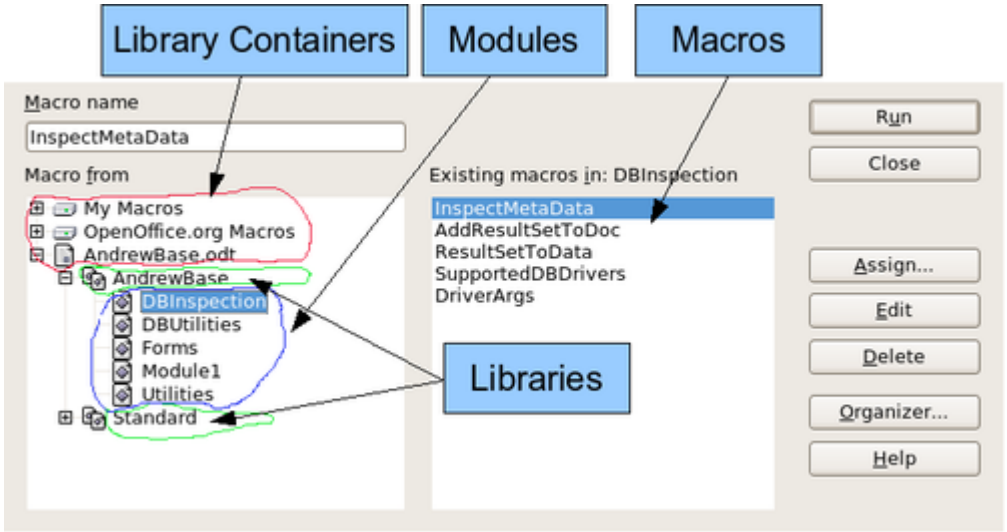
Most often, vendors disable Macros, scripts executions and Active content by default. So any macro is not executed. However, you risk to limit important functionalities of your application (Ms Word, Excel, etc.)

Why MACROs and scripts as a virus

- Viruses dominated the malware scene in earlier years due to lack of user authentication and access controls on personal computer systems at that time.
  - This enabled a virus to infect any executable content on the system.
  - The significant quantity of programs shared on floppy disk also enabled its easy, if somewhat slow, spread.
- The inclusion of tighter access controls on modern operating systems significantly hinders the ease of infection of such traditional, machine executable code, viruses. This resulted in the development of macro viruses that exploit the active content supported by some documents types, such as Microsoft Word or Excel files, or Adobe PDF documents. Such documents are easily modified and shared by users as part of their normal system use, and are not protected by the same access controls as programs.

Dr. Valerio Formicola

Example: a macro




```
REM  *****  BASIC  *****

Sub send_email(sAttachmentURL as string, sEmailAddress as string, sSubject as string, sMessageBody as string)

 Dim eMailer as Object
 Dim eMailClient as Object
 Dim eMessage as Object

        eMailer = createUnoService( "com.sun.star.system.SimpleSystemMail" )
        eMailClient = eMailer.querySimpleMailClient()
        eMessage = eMailClient.createSimpleMailMessage()
                eMessage.setRecipient(sMailAddress)
                eMessage.setSubject(sSubject)
        eMailClient.sendSimpleMailMessage( eMessage, 0 ) 'if you want to handle the sending manually in the mail client software
'       eMailClient.sendSimpleMailMessage ( eMessage,com.sun.star.system.SimpleMailClientFlags.NO_USER_INTERFACE ) 'Silent sendir
End Sub
```

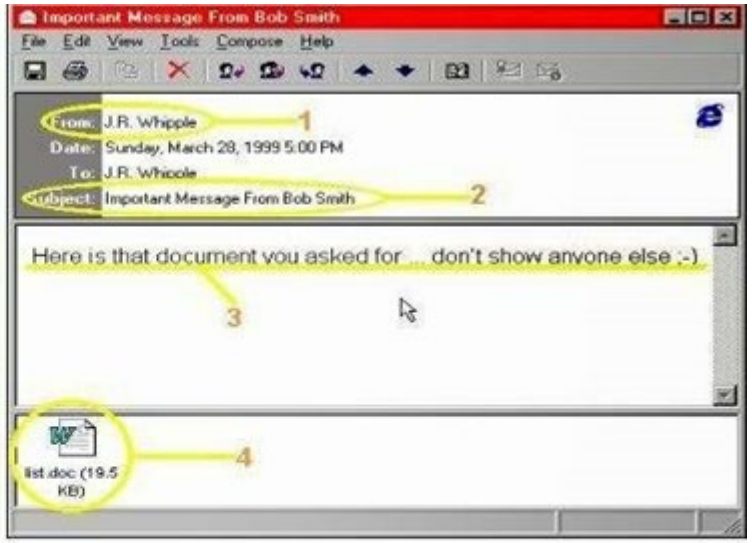
OpenOffice.org macro

- With a MACRO you can: open files, create new files, delete files, execute media files, send emails, etc.
- The depend on the application interpreting the macro (i.e., scripts for Ms Office, OpenOffice, Adobe PDF, …)
- Executed at the moment indicated in the macro: e.g., when you open a document, create a new document, close a document, edit the document, etc.

Example: Melissa virus

- *Date of Attack* – **March 26, 1999**
- *Attacker* – **30 year old David Smith**
- *Victims* – **thousands of Microsoft Word 97 and Word 2000 email users**
- *Damage* - **$80 million**

David L. Smith



This is what you'll see if you receive the Melissa Virus

1. It may be from someone you know.
2. The subject line will read, "Important Message From" and the name may be someone you know.
3. The body of the mail will read, "Here is that document you asked for don't show anyone else ;-)"
4. The attachment, where this nasty little bug lives, has the name of list.doc.

It has characteristics of Viruses, Worms and Trojans

The Virus part is because of users required to **open the document attached to emails**
(following versions just needed to open the email)

Figure 6.1 Melissa Macro Virus Pseudo-code

```
macro Document_Open
    disable Macro menu and some macro security features
    if called from a user document
       copy macro code into Normal template file
    else
       copy macro code into user document being opened
    end if
    if registry key "Melissa" not present
       if Outlook is email client
          for first 50 addresses in address book
              send email to that address
              with currently infected document attached
          end for
       end if
       create registry key "Melissa"
    end if
    if minute in hour equals day of month
       insert text into document being opened
    end if
end macro
```

- Installs itself as the ``open'' macro and copies itself into the Normal template so that any files that are opened are infected
- Then invokes mail program and sends copies to names in address book
  - On PC spread was through mail

Impact:

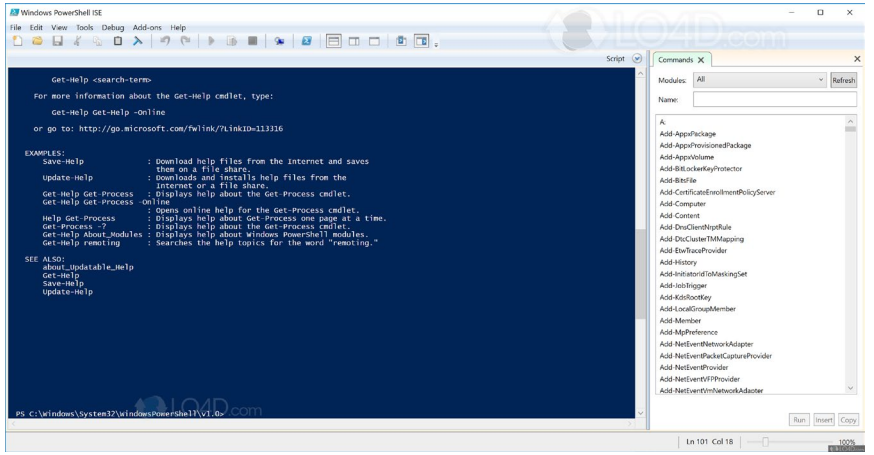
- Overloaded E-mail servers
- Forced companies to stop their email servers
- Only for Microsoft systems
- MacOS could store but not execute

Macros and scripts nowadays: PowerShell target

- These attacks can happen locally or remotely via a network connection, and they often leverage built-in tools like Windows PowerShell and Visual Basic, or Bash and Python on Linux systems. Even macros like those built into Microsoft's Office Suite can be leveraged by attackers.

Ms Powershell

PowerShell, the built-in Windows scripting language, is a popular target for malicious actors because of the powerful capabilities it provides. PowerShell allows remote and local execution, network access, and many other capabilities. In addition, since it is available by default on Windows systems and is often not carefully monitored, attackers can leverage it in many different ways, including for fileless malware attacks where PowerShell scripts are executed locally once a browser or plug-in is compromised.



Defenses against PowerShell attacks include using Constrained Language Mode, which limits sensitive commands in PowerShell, and using Windows Defender's built-in Application Control tool or AppLocker to validate scripts and to limit which modules and plug-ins can be run. It is also a good idea to turn on logging for PowerShell as well as Windows command-line auditing.

As a defender, enabling logging is one of the most important things you can do to make incident response easier. Make sure you consider whether you should have command-line and PowerShell logging turned on to allow you to detect attacks like those we discuss here.

Macros on Microsoft Office, Linux, MacOS

- Many Windows systems have Microsoft Office installed, and Microsoft Office macros written in **Visual Basic for Applications (VBA)** are another target for attackers. Although macro viruses are no longer as common as they once were, macros embedded in Office documents and similar functionality in other applications are potential targets for attackers, and if new vulnerabilities are discovered in Office, the popularity of macro viruses could increase.
- **Linux** systems are also targeted. Attackers may leverage common languages and tools like **Python** (it is ubiquitous in the Linux line of distributions and is available for installation on Windows machines), Perl, and **Bash** as part of their attack process. Languages like these can be used to create persistent remote access using bind or reverse shells, as well as a multitude of other useful exploit tools. **Metasploit**, a popular exploit tool, includes rootkits that leverage each of these languages.
- Preventing use of built-in or preexisting tools like programming languages and shells can be difficult because they are an important part of how users interact with and use the systems they exist on. That makes security that prevents attackers from gaining access to the systems through vulnerabilities, compromised accounts, and other means one of the most important layers of defense.

MacOS Python script: https://www.intego.com/mac-security-blog/beware-dangerous-macro-malware-ahead/

Example: writing Python malware https://hackmag.com/coding/python-malware/

Worms

CalPolyPomona

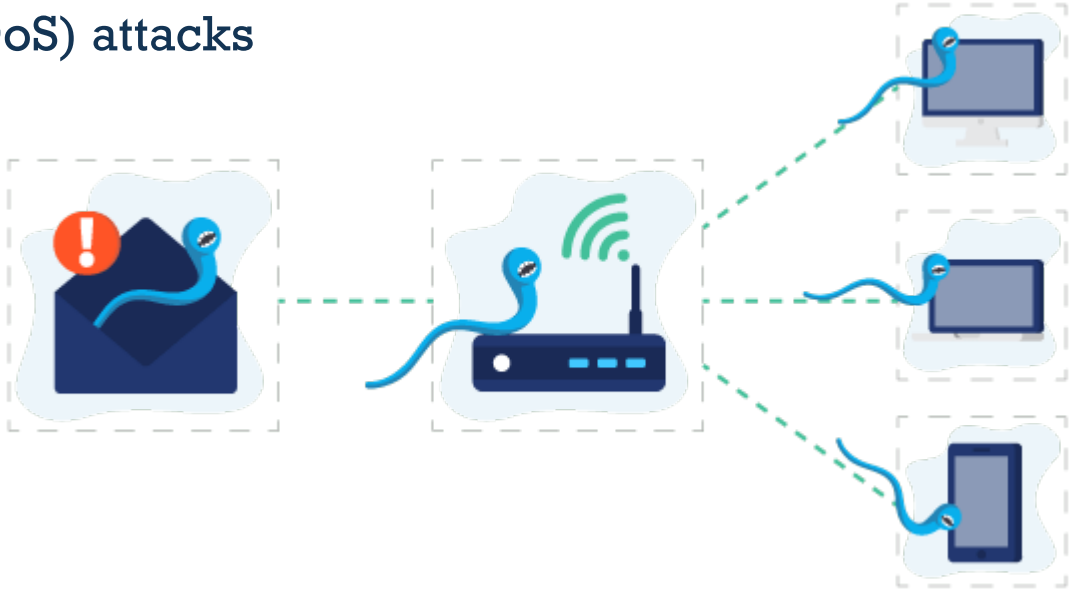
Worms

Program that actively seeks out more machines to infect and each infected machine serves as an automated launching pad for attacks on other machines

- Exploits software vulnerabilities in client or server programs
- Upon activation the worm may replicate and propagate again
- Usually carries some form of payload
- First known implementation was done in Xerox Palo Alto Labs in the early 1980s

Main characteristic of Worms: fast self-replicating and host independence

- A computer worm is a type of malware that can automatically propagate or self-replicate without human interaction, enabling its spread to other computers across a network.
- **Worms** spread from computer to computer and can move and operate independently. A worm's ability to send out hundreds or thousands of copies of itself is one of its biggest dangers.
- Typical uses of worms by hackers (payloads):
  - Launching distributed denial of service (DDoS) attacks
  - Conducting ransomware attacks
  - Stealing sensitive data
  - Dropping other malware
  - Consuming bandwidth
  - Deleting files
  - Overloading networks



Worm replication and activation strategies

via Applications

- **E-mail or instant messenger (IM) facility**
  - Worm e-mails a copy of itself to other systems
  - Sends itself as an attachment via an instant message service, like IRC
- **P2P applications**
  - through P2P applications on network

via System

- **Remote execution capability**
  - Worm executes a copy of itself on another system (remote control of another instance)
- **Remote file access or transfer capability**
  - Worm uses a remote file access or transfer service to copy itself from one system to the other (remote replication)
- **Remote login capability**
  - Worm logs onto a remote system as a user and then uses commands to copy itself from one system to the other
- **Physical file sharing**
  - Creates a copy of itself or infects a file as a virus on removable media (e.g., USB drives)

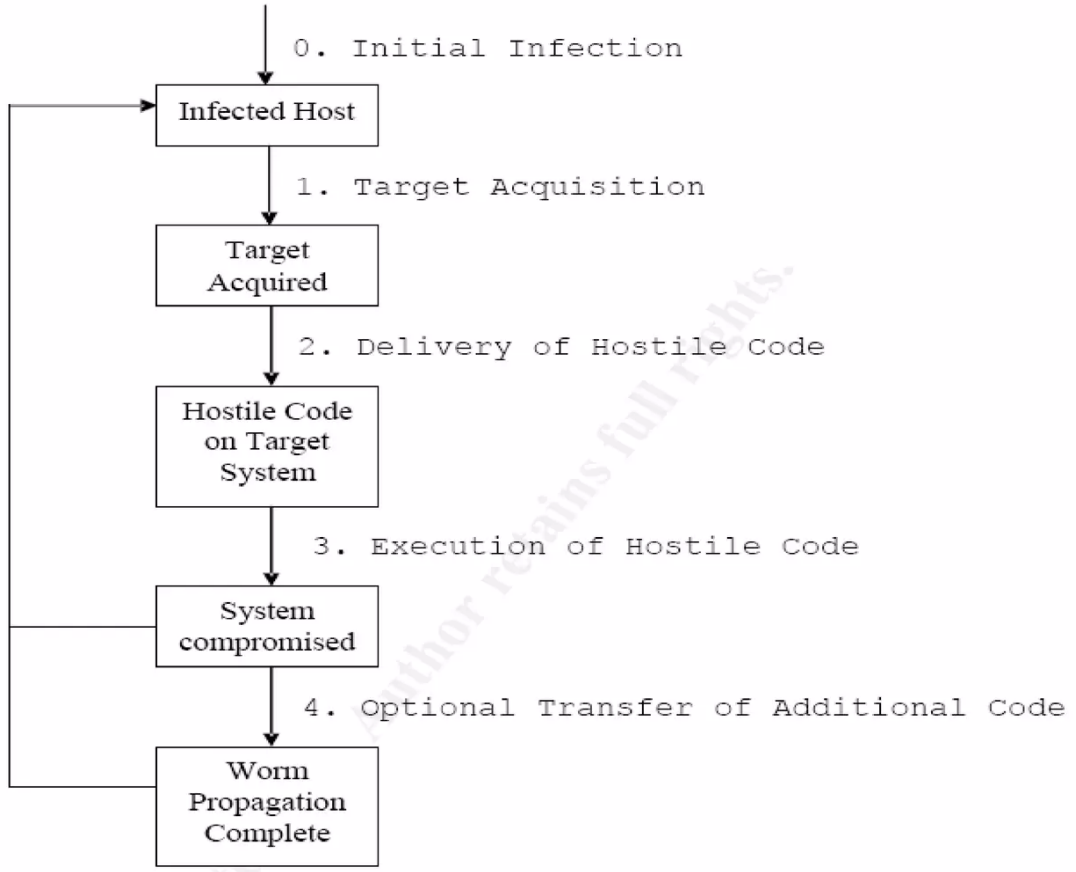
Danger of worms

- It's similar to a virus but way faster to replicate because doesn't need human intervention and is based on software vulnerabilities
- Due to fast replication, it's used in a broader range of attacks, including crashing systems through self-replication, downloading malicious applications, and providing hackers with backdoor access to equipment.
- Worms can also be hard to remediate. Because they spread automatically and quickly, it can take a lot of time and effort to eradicate a worm outbreak from the environment and fully recover. When a worm spreads inside a data storage environment, for example, it can take months to completely clean it up. Even when a worm doesn't have a malicious payload that does damage, it poses a serious nuisance for IT managers who have to dedicate valuable resources to navigate the incident response process.

Target Discovery (1 of 2)

First objective of a worm is to determine the victims. Hence, they use different strategies

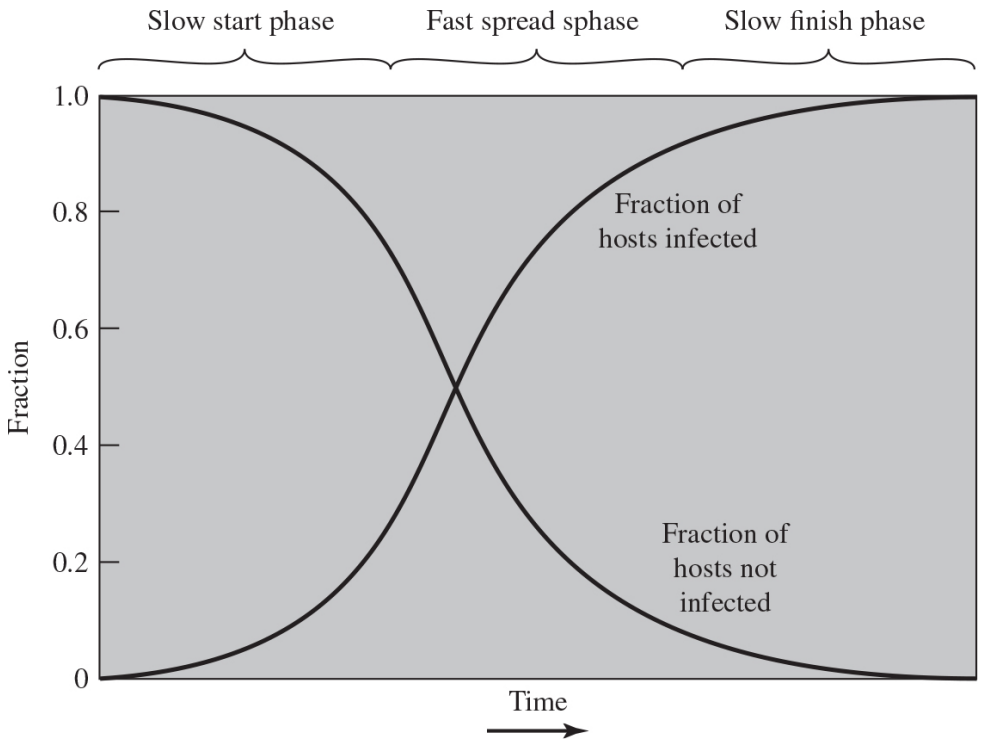
- **Scanning (or fingerprinting)**
  - First function in the propagation phase for a network worm
  - Searches for other systems to infect
- **Random**
  - Each compromised host probes random addresses in the IP address space using a different seed
  - This produces a high volume of Internet traffic which may cause generalized disruption even before the actual attack is launched



Target Discovery (2 of 2)

- **Hit-list**
  - The attacker first compiles a long list of potential vulnerable machines
  - Once the list is compiled the attacker begins infecting machines on the list
  - Each infected machine is provided with a portion of the list to scan
  - This results in a very short scanning period which may make it difficult to detect that infection is taking place
- **Topological**
  - This method uses information contained on an infected victim machine to find more hosts to scan
- **Local subnet**
  - If a host can be infected behind a firewall that host then looks for targets in its own local network
  - The host uses the subnet address structure to find other hosts that would otherwise be protected by the firewall

Figure 6.2 Worm Propagation Model



$$dI(t)dt=\beta I(t)\ S\ (t)$$

where

I(t)=number of individuals infected as of time *t*

S(t)=number of susceptible individuals (susceptible to infection but not yet infected) at time *t*

β=infection rate

N=size of the population, N=I(t)+S(t)

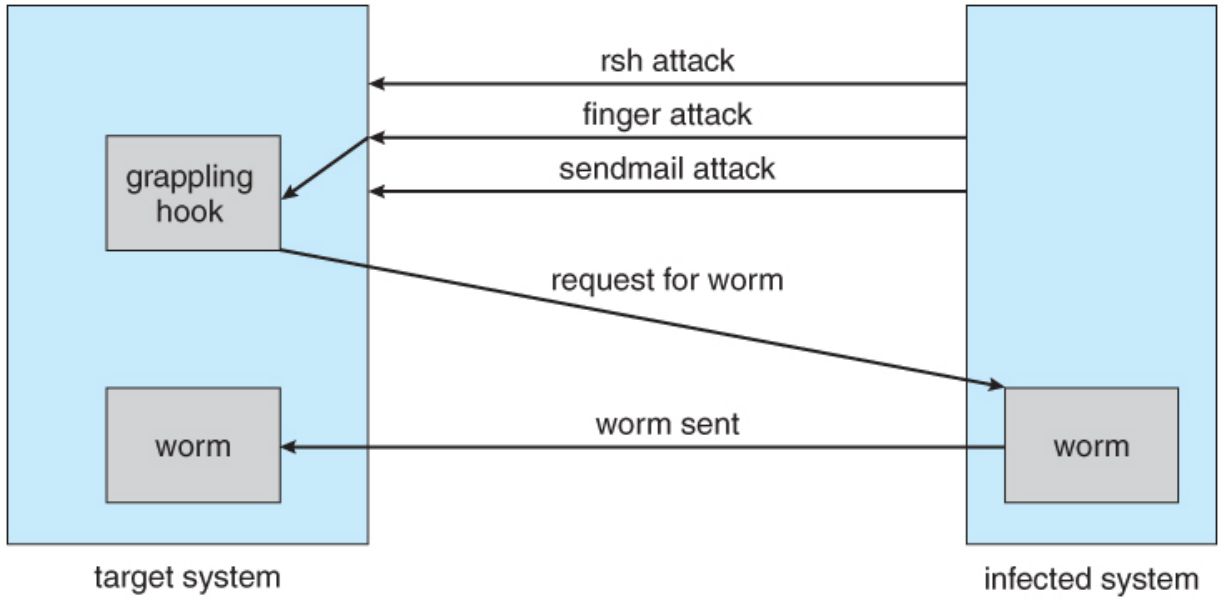
Viruses Vs Worms

Virus	Worm
• Requires a host • Triggered by human interaction • Often arrives through an infected file or program (file-infector)	• Spreads independently • Doesn't require human interaction • Often arrives through a software vulnerability

Example: Morris Worm

- Earliest significant worm infection
- Released by Robert Morris in 1988
- Designed to spread on UNIX systems.
- Part 1: A small program called a grappling hook is deposited on the target using 3 vulnerabilities:
  - Attempted to crack local password file to use login/password to logon to other systems
  - Exploited a bug in the finger protocol which reports the whereabouts of a remote user
  - Exploited a trapdoor in the debug option of the remote process that receives and sends mail
- Part 2: Successful attacks achieved communication with the operating system command interpreter
  - Sent interpreter a bootstrap program to copy worm over

- Born on November 8, 1965
- A Harvard Graduate and attended graduate school at Cornell
- First person convicted under the new "Computer Fraud and Abuse Act".



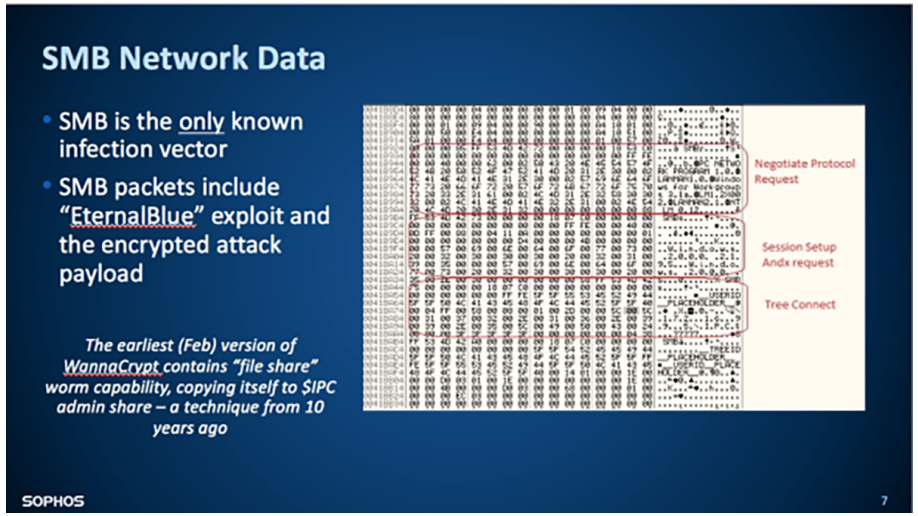
Recent Worm Attacks

Considering Melissa, the propagation part is the same of worms

Melissa	1998	E-mail worm First to include virus, worm and Trojan in one package
Code Red	July 2001	Exploited Microsoft IIS bug (Microsoft Internet Information Server, a web server) Probes random IP addresses Consumes significant Internet capacity when active
Code Red II	August 2001	Also targeted Microsoft IIS Installs a backdoor for access
Nimda	September 2001	Had worm, virus and mobile code characteristics Spread using e-mail, Windows shares, Web servers, Web clients, backdoors
SQL Slammer	Early 2003	Exploited a buffer overflow vulnerability in SQL server compact and spread rapidly
Sobig.F	Late 2003	Exploited open proxy servers to turn infected machines into spam engines
Mydoom	2004	Mass-mailing e-mail worm Installed a backdoor in infected machines
Warezov	2006	Creates executables in system directories Sends itself as an e-mail attachment Can disable security related products
Conficker (Downadup)	November 2008	Exploits a Windows buffer overflow vulnerability Most widespread infection since S QL Slammer
Stuxnet	2010	Restricted rate of spread to reduce chance of detection Targeted industrial control systems

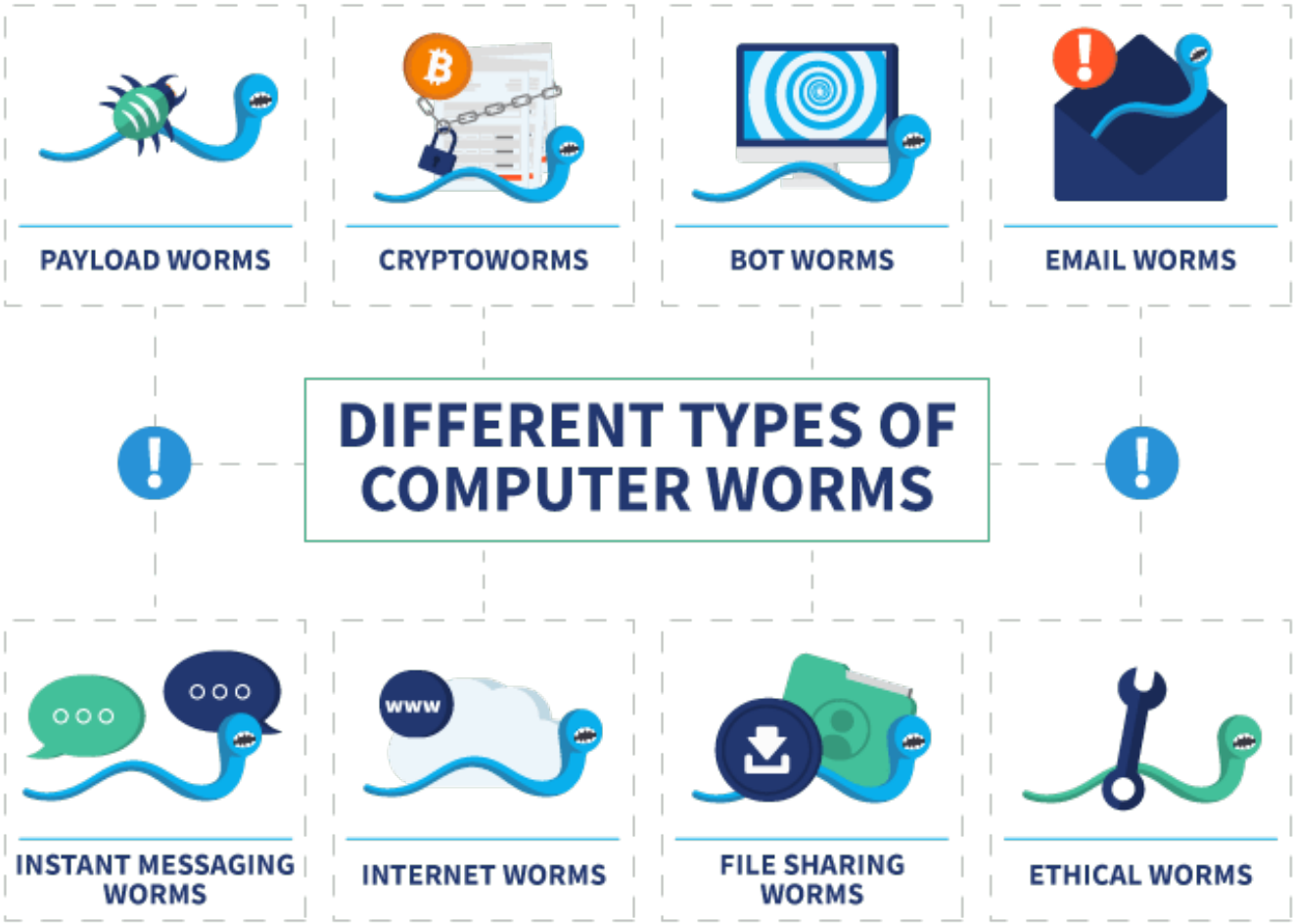
Example: WannaCry

- Ransomware attack in May 2017 that spread extremely fast over a period of hours to days, infecting hundreds of thousands of systems belonging to both public and private organizations in more than 150 countries
- It spread as a worm by aggressively scanning both local and random remote networks, attempting to exploit a vulnerability in the SMB file sharing service on unpatched Windows systems
- This rapid spread was only slowed by the accidental activation of a "kill-switch" domain by a UK security researcher
- Once installed on infected systems, it also encrypted files, demanding a ransom payment to recover them



State of Worm Technology

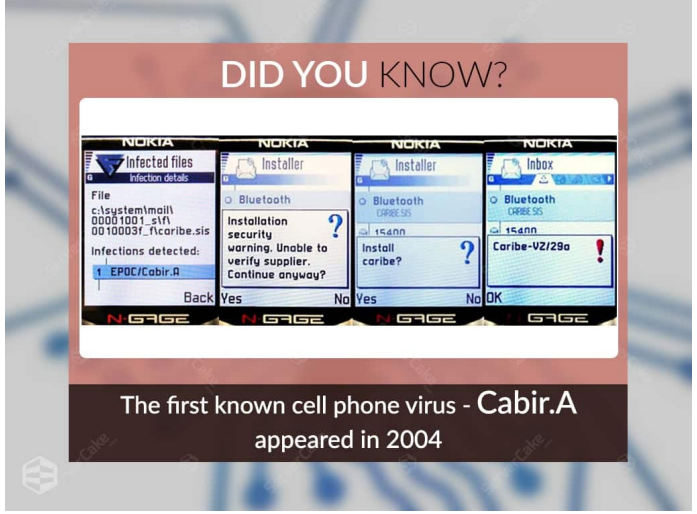
- Multiplatform
- Multi-exploit
- Ultrafast spreading
- Polymorphic
- Metamorphic
- Zero-Day exploits



Mobile Code

- NIST SP 800-28 defines mobile code as
  - "programs that can be shipped unchanged to a heterogeneous collection of platforms and executed with identical semantics"
- Transmitted from a remote system to a local system and then executed on the local system
  - Malicious mobile code is malware that is obtained from remote servers, transferred across a network, and then downloaded on to your computer. This type of code can be transmitted through interactive Web applications such as ActiveX controls, Flash animation, or JavaScript.
- Often acts as a mechanism for a virus, worm, or Trojan horse
- Takes advantage of vulnerabilities to perform its own exploits
- Most common ways of using mobile code for malicious operations on local system are:
  - Cross-site scripting
  - Interactive and dynamic Web sites
  - E-mail attachments
  - Downloads from untrusted sites or of untrusted software

Mobile Phone Worms



- First discovery was Cabir worm in 2004
- Then Lasco and CommWarrior in 2005
- Communicate through **Bluetooth** wireless connections or **MMS**
- **Target is the smartphone**
- Can completely disable the phone, delete data on the phone, or force the device to send costly messages
- CommWarrior replicates by means of Bluetooth to other phones, sends itself as an MMS file to contacts and as an auto reply to incoming text messages

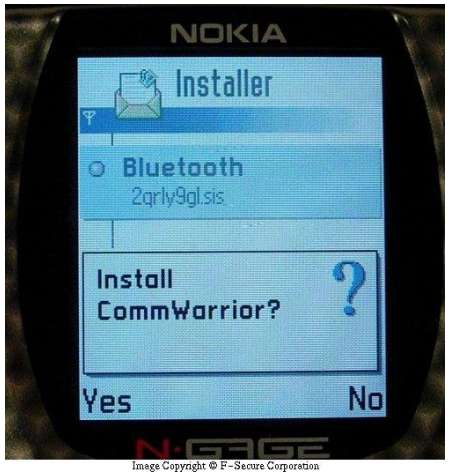
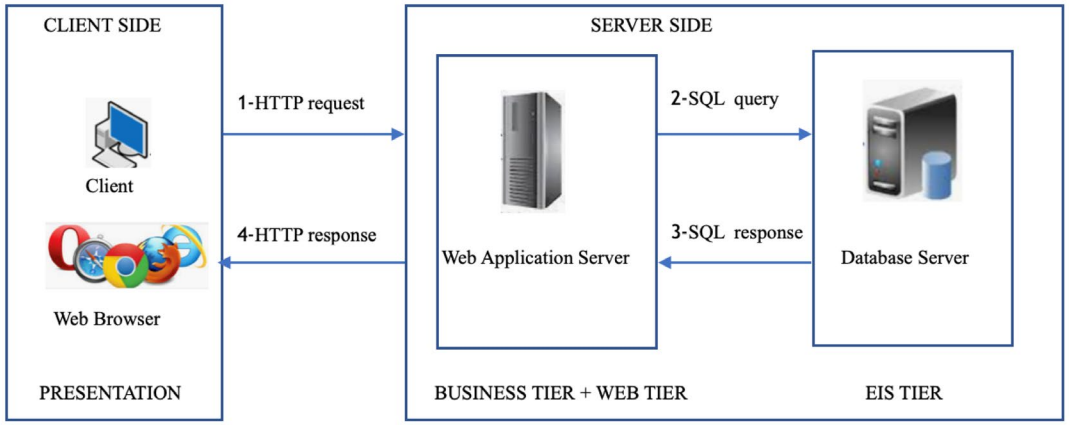
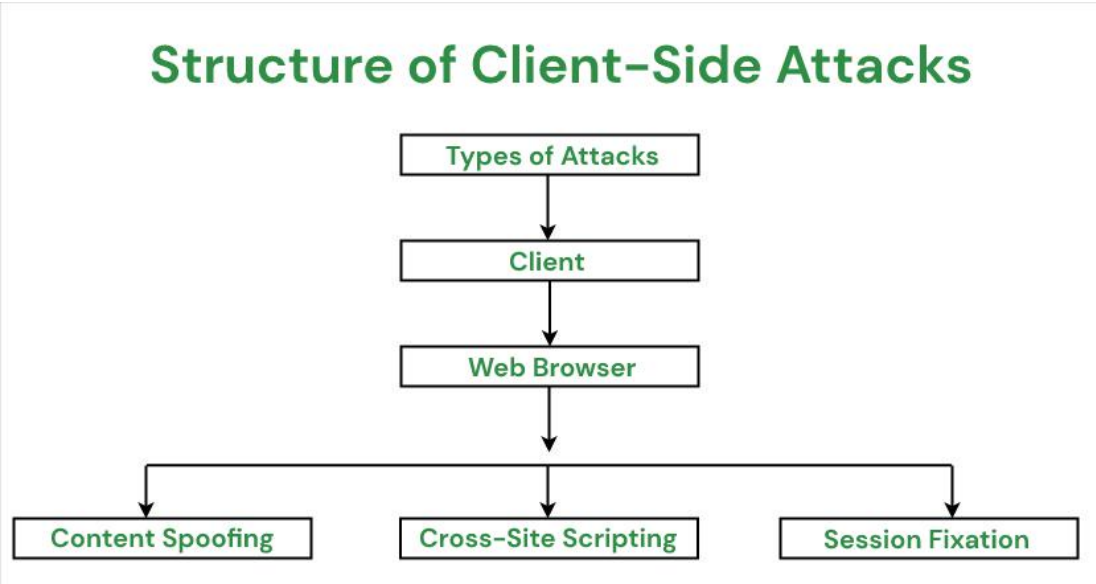


Image Copyright © F-Secure Corporation

Client-side attacks

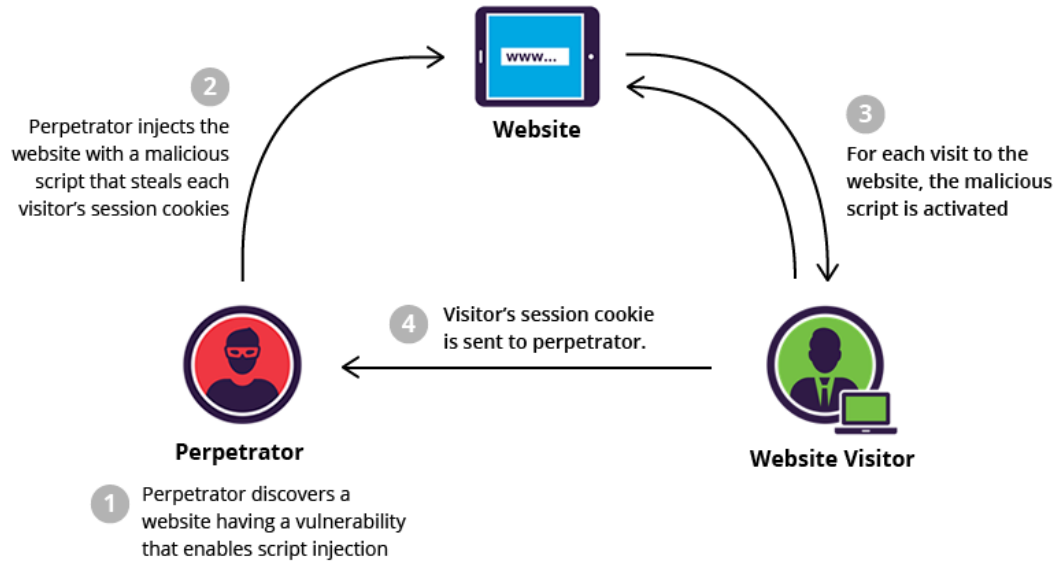
- Browser side applications are frequently a complex combination of custom HTML, CSS, and JavaScript, leveraging numerous third-party libraries that are both served by the custom application, and frequently integrated with third-party services that supply their own custom code and libraries into the same client-side application. All this runs in the customer's browser in the wild, rather than on application owner controlled, managed, and secured servers. Browser applications frequently interact with numerous servers, not just the original server hosting the server application and serving the core elements of the client-side JavaScript application to the user's browser.
- This results in numerous risks for client-side code that are very different from the server-side applications.



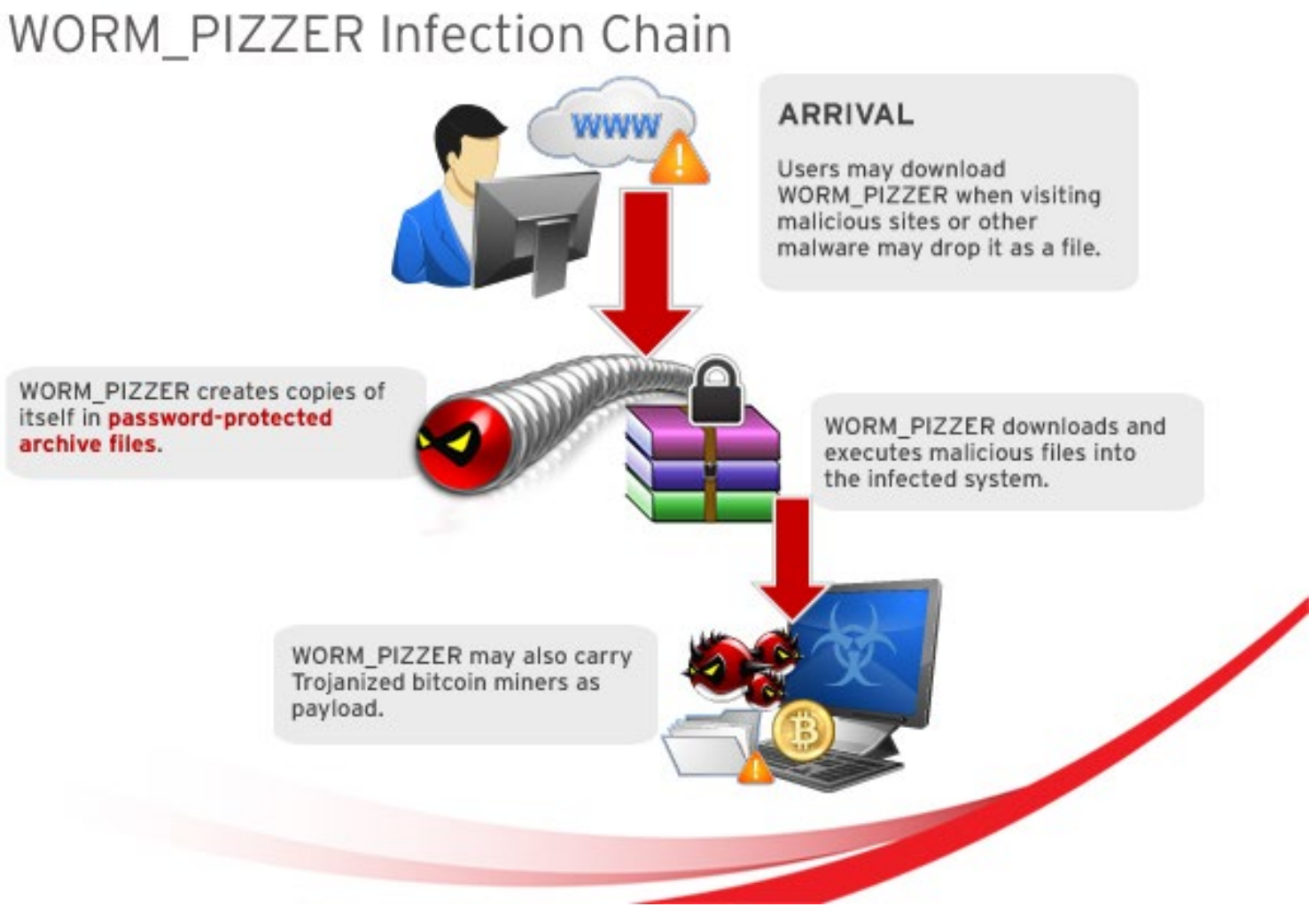


Drive-By-Downloads

- Exploits browser and plugin vulnerabilities so when the user views a webpage controlled by the attacker, it contains code that exploits the bug to download and install malware on the system without the user's knowledge or consent
- In most cases the malware does not actively propagate as a worm does
- Spreads when users visit the malicious Web page
- Exploits vulnerabilities of clients while using Adobe Flash Player and Oracle Java plugins

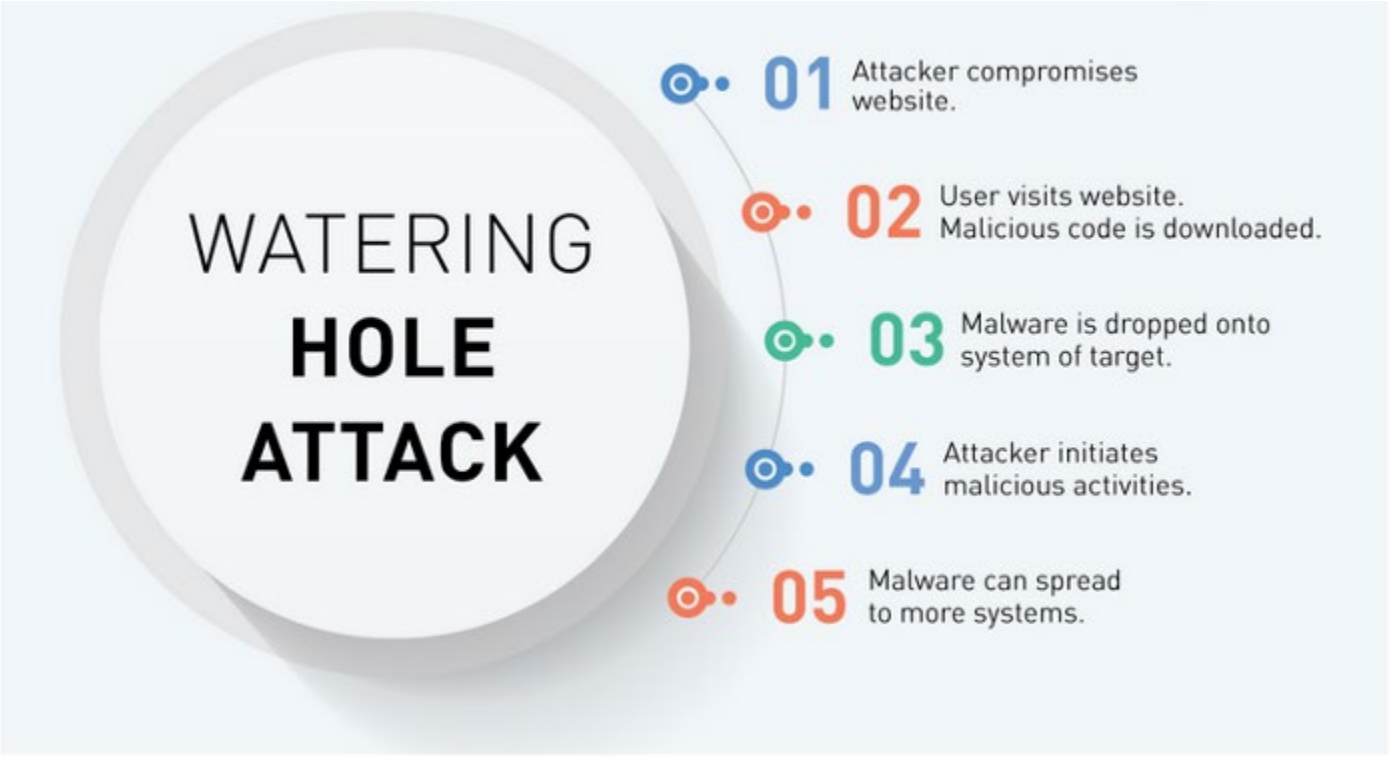


Example


WORM_PIZZER Infection Chain
WWW
ARRIVAL
Users may download WORM_PIZZER when visiting malicious sites or other malware may drop it as a file.
WORM_PIZZER creates copies of itself in password-protected archive files.
WORM_PIZZER downloads and executes malicious files into the infected system.
WORM_PIZZER may also carry Trojanized bitcoin miners as payload.

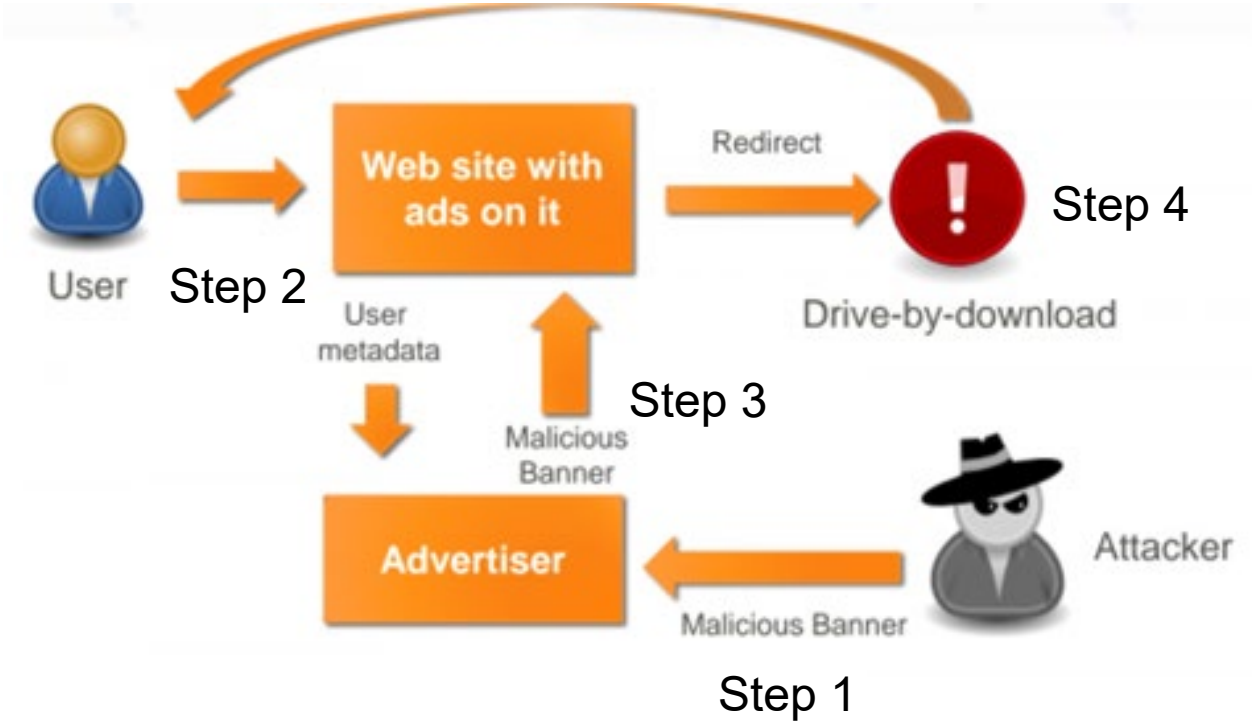
Watering-Hole Attacks

- A variant of drive-by-download used in highly targeted attacks
- The attacker researches their intended victims to identify websites they are likely to visit, then scans these sites to identify those with vulnerabilities that allow their compromise
- They then wait for one of their intended victims to visit one of the compromised sites
- Attack code may even be written so that it will only infect systems belonging to **the target organization** and take no action for other visitors to the site
- This greatly increases the likelihood of the site compromise remaining undetected



Malvertising

- Places **malware on websites without actually compromising them**
- The attacker pays for advertisements that are highly likely to be placed on their intended target websites and incorporate malware in them
- Using these malicious ads, attackers can infect visitors to sites displaying them
- The malware code may be dynamically generated to either reduce the chance of detection or to only infect specific systems
- Has grown rapidly in recent years because they are easy to place on desired websites with few questions asked and are hard to track
- Attackers can place these ads for as little as a few hours, when they expect their intended victims could be browsing the targeted websites, greatly reducing their visibility



Clickjacking

- Also known as a ***user-interface (UI) redress attack***
- Using a similar technique, keystrokes can also be hijacked
  - A user can be led to believe they are typing in the password to their email or bank account, but are instead typing into an invisible frame controlled by the attacker