Cybersecurity

Malware: threats and defenses

Mauro Barni
University of Siena

Malicious software

- **NIST SP 800-83** defines malware as follows

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

A jungle of terms, acronyms, jargons

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.

A jungle of terms, acronyms, jargons

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.

A jungle of terms, acronyms, jargons

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.

Malware classification

- **Propagation mean**
  - infection of existing executable or interpreted content (viruses)
  - exploitation of software vulnerabilities either locally or over a network (worms, drive-by-downloads)
  - social engineering attacks that convince users to bypass security mechanisms (Trojans, phishing attacks)
- **Payload**
  - corruption of system or data files;
  - theft of service (zombie agent, botnet)
  - theft of information (logins, passwords, personal details)
  - stealth for subsequent action

Malware evolution

- **From home-made viruses to crimeware**
  - virus creation toolkits (90's)
  - general crimeware (00's)
- **Attack source**
  - single individuals
  - criminal organizations, government agencies
- **APT**
  - *Advanced:* wide variety of intrusion technologies and malware, including the development of custom malware.
  - *Permanent:* Determined application of the attacks over an extended period against the chosen target.
  - *Threats (Targeted):* serious threat to targeted victim

Propagation by infected content (virus)

- A virus is a parasitic software fragment that attaches itself to some existing executable content
- When the infected system comes into contact with an uninfected piece of code (or document), a fresh copy of the virus passes into the new location
- The hidden code executes secretly when the host program is run. The virus has the same access rights of the host program.
- Viruses attacking macros are particularly dangerous

Macro-based viruses

- Platform independent
- Most of the information introduced onto a computer system is in the form of documents rather than programs
- Macro viruses are easily spread (e.g by e-mail)
- Traditional file system access controls are of limited use in preventing their spread, since users are expected to modify them.
- Macro viruses are much easier to write or to modify than traditional executable viruses.

Virus components

- **Infection mechanism**: this is the means by which a virus spreads or propagates, enabling it to replicate. The mechanism is also referred to as the infection vector.
- **Trigger**: the event or condition that determines when the payload is activated or delivered, sometimes known as a *logic bomb*.
- **Payload**: what the virus does, besides spreading. The pay load may involve damage or may involve benign but noticeable activity.

Virus life

- **Dormant phase**: The virus is idle. The virus will eventually be activated by some event. Not all viruses have this stage
- **Propagation phase**: the virus places a copy of itself into other programs or into certain system areas of the disk. The copy may not be identical to the propagating version
- **Triggering phase**: the virus is activated to perform the function for which it was intended.
- **Execution phase:** the function is performed. It may be harmless, such as a message on the screen, or damaging, such as the destruction of programs and files.

Example: Melissa virus

- **Host:** Microsoft Word macro, Document_Open macro command
- **Sequence of operations:**
  1. Disables Macro menu and related security features
  2. Copies itself into the global template file (to infect new documents)
  3. If never opened before, uses Outlook to send 50 copies of the document to first entries in user's address book
  4. Check date and time to trigger paylod: include Simpson's quite into document

Virus classification: based on target

- **Boot sector infector:** Infects a master boot record or boot record and spreads when a system is booted
- **File infector:** Infects files that the operating system or shell consider 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. Typically, a multipartite virus is capable of infecting multiple types of files, so virus eradication must deal with all of the possible sites of infection.

Virus classification: concealment strategy

- **Encrypted virus**: A portion of the virus creates a random encryption key and encrypts the remainder of the virus (payload). The key is stored with the virus
- **Stealthy virus**: The entire virus, not just the payload, is hidden. It may use code mutation or compression
- **Polymorphic virus**: A virus that creates copies during replication that are functionally equivalent but have distinctly different bit patterns, in order to defeat programs that scan for viruses.
- **Metamorphic virus**: The difference with respect to polymorphic viruses is that a metamorphic virus rewrites itself completely at each iteration, using multiple transformation techniques. Metamorphic viruses may change their behavior as well as their appearance.

Propagation by vulnerability exploit (worms)

- A worm is a program that actively seeks out more machines to infect, and then each infected machine serves as an automated launching pad for attacks on other machines
- Infection mechanisms include
  - Electronic mail or instant messenger facility
  - Filesharing
  - Remote execution capability
  - Remote file access or transfer capability
  - Remote login capability

Life of worms

- A worm's life goes through through the same phases of viruses: dormant, propagation, triggering, and execution
- Main difference is propagation
  - Scanning
  - Copying

Scanning approaches

- **Random:** a compromised host probes random addresses in the IP address space
- **Hit-List**: The attacker compiles a long list of potential vulnerable machines and begins infecting machines on the list. Each infected machine is provided with a portion of the list to scan.
- **Topological**: it uses information contained on an infected victim machine to find more hosts to scan
- **Local subnet**: it uses the subnet address structure to infect other hosts.

Propagation models

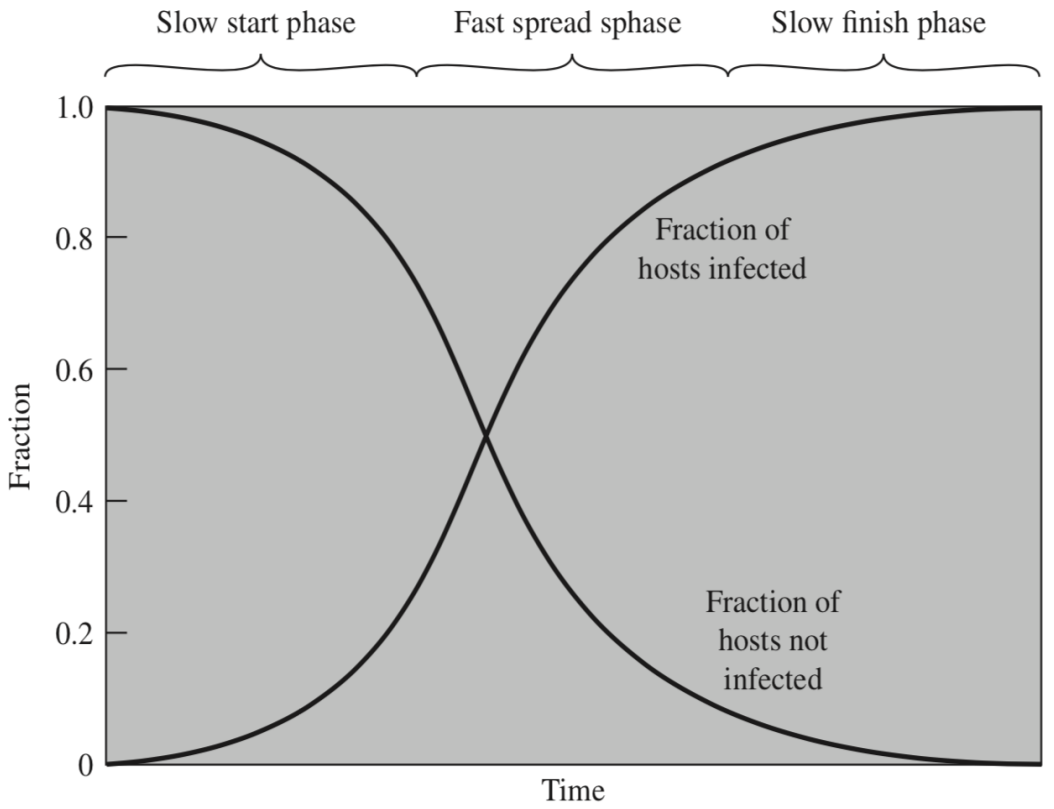
Propagation models used for biological infections can be applied

$$\frac{di(t)}{dt} = \beta i(t)[N - i(t)]$$

$$i(t) = num.of\ infected\ nodes$$

$$N = overall\ number\ of\ nodes$$

$$\beta = infection\ rate$$



Example: Morris worm

- First worm released in 1988
- Attacked UNIX systems
- Propagation mechanism
  - try to log on a remote host by cracking the password file of the infected host
    - User's account name and simple permutations
    - A list of 432 built-in passwords thought to be likely candidates
    - All the words in the local system dictionary
- Communication with UNIX shell to run a program that calls back the infecting host to download the entire worm.

A long history of worms

- Melissa (1998)
- Code Red I and II (2001)
- SQL Slammer (2003)
- Sobig.F (2003)
- Conficker (2008)
- Stuxnet (2010)
  - First example of cyberwarfare?
- WannaCry (ransomware, 2017)

Most commonly exploited vulnerabilities

- Mobile code: Java applets, ActiveX, JavaScript …
- Mobile phone worms: Bluetooth, MMS
- Drive-by-download:
  - wait for user to visit the web-page (plus Adobe Flash Player, Oracle Java)
  - infect target websites
  - malvertising
- Clickjacking (UI redress attack)

Highly sophisticated versions

- Multiplatform
- Multiexploit
- Zero-day exploit
- Ultrafast spreading
- Polymorphic, metamorphic
- Oftebn transport vehicles for other attacks

Propagation by social engineering

- Use social engineering tools to convince the users to assist malware to compromise their own system
  - SPAM-based
  - Trojans

SPAM-based malware propagation

- Continuous race of arms
- Unsolicited advertisement
- Malware propagation
  - attached documents
  - Trojan's code
  - phishing attack

Trojan horses

- Convince users to run useful programs performing unintended, malicious operations
- Three models:
  - Intended functionality plus malicious activity
  - Modified intended functionality to incorporate malicious activity
  - Completely replace intended functionality with malicious one
- Mobile phones: threat comes form apps
  - Android
  - IOS: jail-broken phone, legitimate apps developed with infected development systems

Malware payload: system corruption

- Data destruction
  - Ransomware (Chernobyl, Wannacry)
- Physical damage
  - BIOS code, controllers, Ex: Iranian nuclear plants
- Logic bombs

Maklware payload: attack agent

- Malware subverts the computational and network resources of the infected system for use by the attacker
- A system infected in this way is called a bot (or zombie, or drone). Bots can form a botnet controlled by the attacker
- Bots can be used for
  - DDoS
  - Spamming
  - Sniffing
  - Keylogging
  - Malware spreading
  - Fake clicks, manipulation of polls, games …
- Remote control of infected system

Malware payload: information theft

- Credential theft, keyloggers, spyware
  - Observing the user typing his credential into the system bypasses defenses based on hashing and encryption
  - spyware does not limit its observation to the keybard
- Phishing, identity theft
  - redirection to fake websites, on-line forms … usually spread by means of SPAM
  - spear phishing
- Espionage
  - industrial secret information
  - configuration files for furtehr attacks
  - Wikileaks (Snowden case)

Malware payload: stealthing (backdoors)

- In this case the malware hides its presence and stays there to provide an inner anchor point for a subsequent attack.
- Backdoors (also called trapdoors)
  - Initially devised for debug or recovery reasons, can provide an access point to hack a systems
  - In networked systems may assume the form of programs listening to non-standard ports (used by WannaCry)
- Rootkit: hidden programs run with system administrator privileges
  - persistent vs memory-based
  - user-mode vs kernel-mode
  - Focus here to remain stealthy while at the same time carrying out the malicious activity

Countermeasures (anti-virus)

- Two different and complementary approaches
  - Prevention
  - Threat mitigation
- **Prevention**
  - prevent propagation and damage by
    - access control
    - removal of vulnerabilities (patches)
    - awareness
- **Keep in mind: 100% prevention is not achievable**

Threat mitigation

- Threat mitigation goes through the following steps:
  - Malware detection
  - Malware identification
  - Malware removal
- If removal (or even identification) fails: then infected files or system must be discarded and a clean back up installed

Threat mitigation: requirements

- **Generality**: be able to handle a wide variety of attacks
- **Timeliness**: respond quickly so as to limit damage
- **Resiliency**: be resistant to evasion techniques employed by attackers
- **Minimal denial-of-service**: minimal reduction in capacity or service due to the countermeasure
- **Transparency**: The countermeasure should not require modification to existing resources
- **Global and local coverage**: be able to deal with attack sources both from outside and inside the enterprise network

Host-based scanners and anti-virus

- Most common approach
- 4 anti-virus classes
  - *simple scanners:* look for specific signatures (possibly with wildcards
    - limited to known malware
  - *heuristic scanners:* look for code fragments or checksum
    - append a checksum of encrypted hash for subsequent control
  - *Activity scanners:* look for actions rather then signatures
    - in the end malware must perform some actions
  - *hybrid, full feature protection*

Specific countermeasures (hostbased)

- Sandbox analysis
  - observe malware and or suspect code in a sandbox to detect malware and signatures
  - escape mechanisms: idle time, sandbox detection, logic bomb
- Continuous activity monitoring
  - Attempts to open, view, delete, and/or modify files
  - Attempts to format disk drives and other unrecoverable disk operations
  - Modifications to the logic of executable files or macros
  - Modification of critical system settings, such as start-up settings
  - Scripting of e-mail and instant messaging clients to send executable content
  - Initiation of network communications

Specific countermeasures (host based)

- Spyware (and stealthy malware) detection
  - specific software needed due to the specific characteristics of stealthy malware
- Rootkit detection
  - Particularly difficult since rootkits interferes with and alters operating system functionalities
  - file integrity check: compares different system scans (API, administrator calls)
  - Once detected: reinstall OS

Perimeter scanning approaches

- Located in firewalls or IDS (Intrusion Detection System)
  - More general view
  - only signature based detection is possible
- Ingress monitoring
  - prevent entering the system
  - honeypots, messages sent to unused IP addresses
- Egress monitoring
  - more related to detection of infected systems
  - detection of bots, botnes, spammers
- Distributed intelligence

References

- W. Stallings, L. Brown, "*Computer security: principles and practices",* Pearson, 4-th edition. Chapter 6
- Lectures notes (these slides)