2FA, Malicious Software

#week4

- **Definition:**
  Two-Factor Authentication (2FA) is a method of confirming a user's identity using **two distinct forms of evidence** from different categories.

- Goal: Add a second layer of defence, so even if one factor (e.g. password) is stolen, unauthorized access is harder.
  - 2FA uses two of these from different categories.
  - MFA (Multi-Factor Authentication) extends this to three or more.

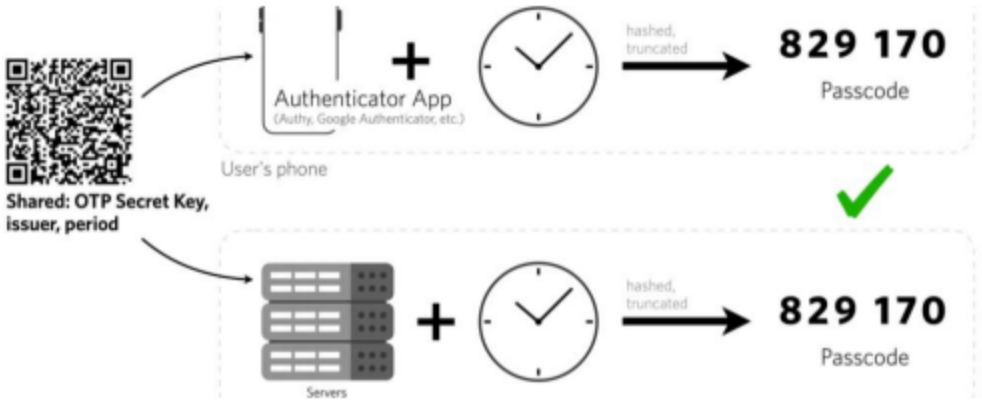
Factor Type	Example	Description
Something you know	Password, PIN, Security question	Knowledge-based
Something you have	Smartphone, Smartcard, Hardware token	Possession-based
Something you are	Fingerprint, Face ID, Retina scan	Inherence-based

1. **SMS-based codes**
   - One-time code sent to registered phone number.
   - Vulnerable to SIM swapping, interception.
2. **App-based (TOTP)**
   - Time-based One-Time Password (TOTP) generated by apps like Google Authenticator, Authy, Microsoft Authenticator.
   - Offline, more secure than SMS.
3. **Push notifications**
   - "Approve login" prompts (e.g. Duo, Okta, Microsoft).
   - Convenient but can lead to *push fatigue* (users approve without checking).
4. **Hardware tokens**
   - Physical devices like YubiKey or RSA SecurID.
   - Provide cryptographically strong authentication.
5. **Biometric 2FA**
   - Fingerprint or facial recognition paired with password/PIN.
   - Privacy and device limitations may apply.

How does TOTP Work

Based on shared secret key between client and server, uses the current time and key to generate a 6 digit code which are valid for short intervals.

Implemented using standards such as RFC6238 and RFC 4226



- Shared Secret + Current Time → HMAC-SHA1 → Truncate → 6-digit Code

The advantages of this is that it is simple and dramatically reduces unauthorised access.

Limitations & Risks

- **Phishing kits** can capture both password *and* one-time code.
- **Man-in-the-middle attacks** may intercept login flow.
- **SIM swapping** makes SMS 2FA unreliable.
- **User fatigue**: repeated prompts → careless approvals.
- **Device loss** may lock out legitimate users.

Best practice: Prefer app-based or hardware-based 2FA; avoid SMS when possible.

Overview of SMS based OTP

- **Concept**: Adds an extra layer of security by requiring a one-time code sent via text message, in addition to a password.
- **Typical Process**:
  - User enters username and password.
  - System sends a 6-digit OTP (one-time passcode) via SMS.
  - User enters OTP to complete authentication.
- **Goal:** Mitigate risks of password compromise (e.g., phishing, credential stuffing).

The Security Model Assumption

- SMS is assumed to provide a “separate channel” from the internet.
- It relies on the **telecommunications network** as a trusted intermediary.
- The problem: this trust model is outdated — the mobile network and SIM system were never designed for high-security authentication.

Man-in-the-Middle (MitM) Phishing

- **Mechanism**: Attacker tricks a user into logging in on a fake website.
- The site proxies credentials and the SMS code to the legitimate site in real time.
- Tools like **Evilginx** or **Modlishka** can automate this.
- **Impact**: Both password and OTP are stolen during the session.
- ***Observation****:* SMS codes authenticate possession of a number, not identity of the session origin.

SIM Swapping

- **Mechanism**: Attacker convinces a mobile carrier to transfer the victim's phone number to a SIM card they control.
- **Result**: All SMS messages (including OTPs) are sent to the attacker.
- **Attack surface**: Social engineering of telecom staff, weak carrier identity verification.
- **Impact**: Full account takeover even if the password is strong.
- *Example:* In 2019, Twitter CEO Jack Dorsey's account was compromised via SIM swapping — attackers gained control over his phone number and intercepted authentication codes.

SS7 Network Exploits

- **Signaling System 7 (SS7)** underpins SMS delivery.
- Vulnerabilities allow attackers to intercept SMS traffic at the telecom level.
- Exploitation has been demonstrated repeatedly since 2014.

Aspect	Advantage	Weakness
Accessibility	Works on any phone, no app needed	Relies on insecure SMS protocol
User adoption	Simple and familiar	High risk from phishing and SIM fraud
Cost	Cheap to deploy	Expensive to mitigate breaches

Critical Assessment

- SMS-based 2FA **does improve** over password-only systems, particularly against automated attacks.
- However, **it provides a false sense of strong protection**:
  - Vulnerable to human and infrastructural manipulation.
  - Inadequate against targeted attacks.
- **Recommendation:** Use SMS 2FA only as a temporary or fallback mechanism; promote migration to app-based or hardware-based authentication.

Malicious Software

Malicious software comes in many shapes and sizes and can be categorised into several groups.

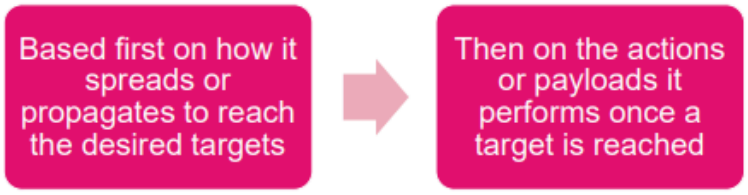
Category	Term	Description	Key Concept
Self-Replicating	**Virus**	Malware that attaches to an executable file and attempts to **replicate itself** to other files when the infected code is run.	Requires a **host program** and **user execution**.
	Worm	A complete, independent program that can **run and propagate by itself** across a network without needing to attach to another program.	Self-contained and **self-propagating** over a network.
Deception & Concealment	**Trojan Horse**	Malicious software disguised as, or embedded within, a legitimate program. It often exploits the permissions of the invoking entity.	Relies on **deception** to gain initial access/execution.
	Backdoor (Trapdoor)	A hidden mechanism inserted by an intruder or developer to **bypass normal security checks** and allow unauthorized access or functionality.	A **secret bypass** for access.
	Rootkit	A set of hacker tools used to **conceal the presence** of a compromise and maintain **privileged access** (root/admin) to a system.	**Stealth** and **privileged persistence**.

Category	Term	Description	Key Concept
Triggered Malice	**Logic Bomb**	A dormant program that lies in wait until a **predefined condition** is met (e.g., a specific date, file deletion, or user action) before executing its malicious function.	Malicious code with a **time or event-based trigger**.
System Compromise	**Exploit**	Code specifically written to take advantage of a single known **vulnerability** or set of security flaws in software or hardware.	A **tool** to leverage a specific **flaw**.
	Keylogger	Software or hardware that **records every keystroke** made on a compromised computer system, capturing sensitive information like passwords.	Captures **input data** directly.
Attack Infrastructure	**Zombie / Bot**	An infected machine that is activated to **launch attacks** on other machines, typically as part of a **botnet** (a network of compromised devices).	A **controlled participant** in a larger attack network.
Delivery & Installation	**Downloader**	A program, often sent via email, whose primary purpose is to **install other malicious items** onto a compromised machine.	Primary job is **secondary payload delivery**.
	Auto-rooter	A tool used by malicious hackers to **break into new machines remotely** and automatically gain administrative (root) access.	Automates the **initial unauthorized access**.
	Kit (Virus Generator)	A set of tools designed for the **automatic generation** of new, often customized, viruses.	A **creation toolkit** for malware.

Category	Term	Description	Key Concept
Monetization & Tracking	**Spyware**	Software that **collects information** about a user's computer activity and transmits it to another entity without the user's knowledge.	**Secret data collection**.
	Adware	Software that displays or downloads **unwanted advertising**, often resulting in pop-ups or browser redirection, sometimes integrated with spyware.	**Unwanted commercial intrusion**.
Denial of Service	**Flooder (DoS/DDoS Tool)**	A tool used to attack networked computer systems by generating a massive volume of traffic, designed to **overwhelm resources** and cause a **denial-of-service (DoS)**.	Focuses on **resource exhaustion**.
	Spammer Programs	Used to send extremely **large volumes of unsolicited e-mail** (spam), often as part of a phishing or malware distribution campaign.	Focuses on **high-volume communication**.
Platform-Agnostic	**Mobile Code**	Software (e.g., scripts, macros) designed to be **platform-independent** (heterogeneous) and execute with the same semantics across different operating systems or platforms.	**Portable code** for multi-platform infection.

You can see there are a lot of catagories, to simplifier this further they are often split into two main catagories, based on how they propogate

- Can be classified into two broad categories:



- Propagation mechanisms:
  - Include infection of existing executable or interpreted content by viruses that is subsequently spread to other system
  - Exploit of software vulnerabilities either locally or over a network by worms or drive-by-downloads to allow the malware to replicate
  - Social engineering attacks that convince users to bypass security mechanisms to install trojans or to respond to phishing attacks

- Earlier approaches to malware classification distinguished between:
  - Those that need a host program, being parasitic code such as viruses
  - Those that are independent, self-contained programs run on the system such as worms, trojans, and bots
- Another distinction used was:
  - Malware that does not replicate, such as trojans and spam e-mail
  - Malware that does, including viruses and worms
- Payload actions performed by malware once it reaches a target system can include:
  - Corruption of system or data files
  - Theft of service in order to make the system a zombie agent of attack as part of a botnet
  - Theft of information from the system, especially of logins, passwords, or other personal details by keylogging or spyware programs
  - Stealthing where the malware hides its presence on the system from attempts to detect and block it
- Blended attack
  - Uses multiple methods of infection or propagation to maximize the speed of contagion and the severity of the attack

Attack kits

- Initially the development and deployment of malware required considerable technical skill by software authors
- This changed with the development of virus-creation toolkits in the early 1990s and more general attack kits in the 2000s
  - These toolkits are often known as *crimeware*
  - Include a variety of propagation mechanisms and payload modules that even novices can combine, select, and deploy
  - Can easily be customized with the latest discovered vulnerabilities in order to exploit the window of opportunity between the publication of a weakness and the deployment of patches to close it
  - These kits greatly enlarged the population of attackers able to deploy malware

Attack sources

- Another significant malware development over the last couple of decades is the change from attackers being individuals to more organised and dangerous attack sources

  These include politically motivated attackers, criminals, organised crime, organisations that sell their services to companies and nations, and national government agencies
- This has significantly changed the resources available and motivation behind the rise of malware leading to development of a large underground economy involving the sale of attack kits, access to compromised hosts, and to stolen information

Advanced persistent threat (apt)

- Have risen to prominence in recent years
- A well-resourced, persistent application of a wide variety of intrusion technologies and malware to selected targets, usually business or political
- APTs differ from other types of attack by their careful target selection, and persistent, often stealthy, intrusion efforts over extended periods
  - Aurora, RSA, APT1, and Stuxnet are often cited as examples
- Named as a result of these characteristics:
  - Advanced
    - The individual components may not necessarily be technically advanced, but are carefully selected to suit the chosen
  - Persistent
    - Determined application of the attacks over an extended period against the chosen target in order to maximize the chance of success
  - Threats
    - Threats to the selected targets as a result of the organised, capable, and well-funded attackers intent to compromise the specifically chosen targets

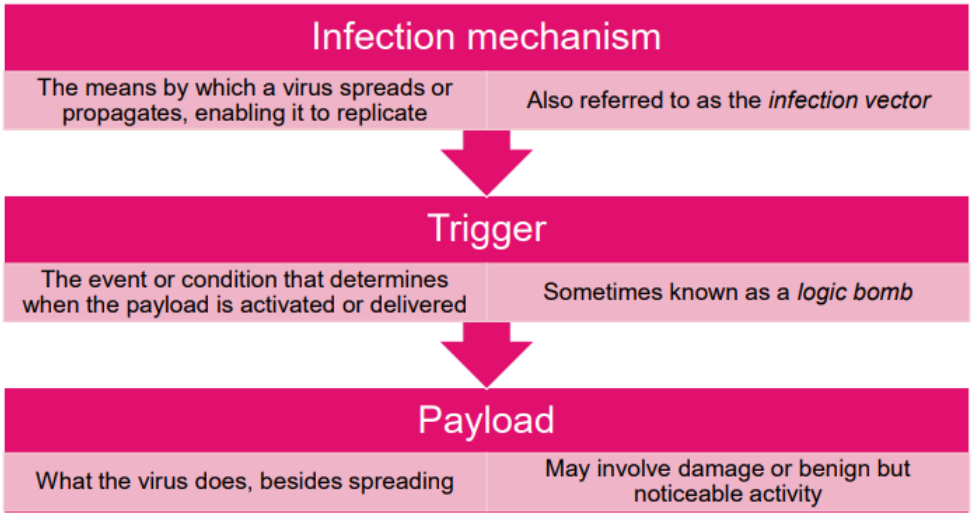
Viruses

Viruses

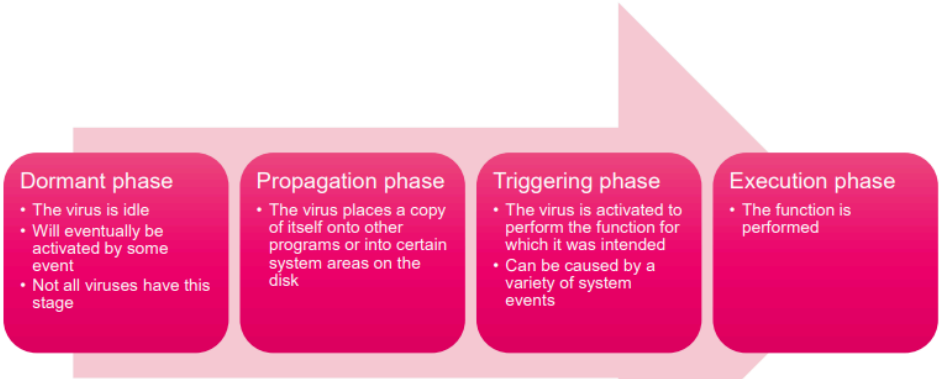
- Parasitic software fragments that attach themselves to some existing executable content
- Can “infect” other programs or any type of executable content and modify them
- The modification includes injecting the original code with a routine to make copies of the virus code, which can then go on to infect other content
- One reason viruses dominated the malware scene in earlier years was the lack of user authentication and access controls on personal computer systems

Virus Structure

- A computer virus and many contemporary types of malware includes one or more variants of each of these components:



- During its lifetime, a typical virus goes through the following four phases:



Goldsmiths

```
   program V :=

{goto main;
   1234567;

   subroutine infect-executable :=
       {loop:
       file := get-random-executable-file;
       if (first-line-of-file = 1234567)
           then goto loop
           else prepend V to file; }

   subroutine do-damage :=
       {whatever damage is to be done}

   subroutine trigger-pulled :=
       {return true if some condition holds}

main:    main-program :=
       {infect-executable;
       if trigger-pulled then do-damage;
       goto next;}

next:

}
```

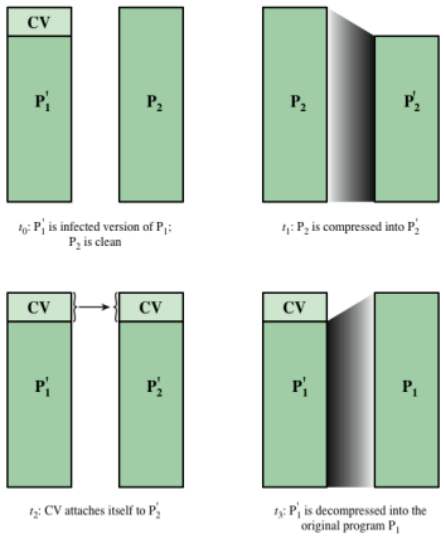
```
   program CV :=

{goto main;
   01234567;

   subroutine infect-executable :=
           {loop:
               file := get-random-executable-file;
           if (first-line-of-file = 01234567) then goto loop;
(1)        compress file;
(2)        prepend CV to file;
       }

main: main-program :=
           {if ask-permission then infect-executable;
(3)        uncompress rest-of-file;
(4)        run uncompressed file;}
       }
```

Virus Vulnerability

A **virus** initially infects a single program. Once this infected program executes, the virus can potentially spread to other executable files, depending on that program's permissions.

While the only guaranteed way to stop infection is complete prevention (blocking the virus from entering the system at all), this is almost impossible because a virus can **hide** inside any external software. Consequently, nearly every system is vulnerable.

A key way to limit the **spread of viruses** is to deny standard users the permission to modify or change existing programs on the system.



Virus Classification by target

Boot sector infector

Infects a master boot record or boot record and spreads when a system is booted from the disk containing the virus

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

Virus classification by concealment strategy

- Includes the following categories:

Encrypted virus

- Portion of the virus creates a random encryption key and encrypts the remainder of the virus
- When an infected program is invoked, the virus uses the stored random key to decrypt the virus
- When the virus replicates, a different random key is selected
- Because the bulk of the virus is encrypted with a different key for each instance, there is no constant bit pattern to observe

Stealth virus

- A form of virus explicitly designed to hide itself from detection by antivirus software
- The entire virus, not just a payload is hidden

Polymorphic virus

- A virus that mutates with every infection, making detection by the "signature" of the virus impossible

Metamorphic virus

- Mutates with every infection
- Rewrites itself completely at each iteration, increasing the difficulty of detection
- May change their behavior as well as their appearance

Macro and scripting viruses

Macro viruses infect scripting code used to support active content in a variety of user document types

Threatening for a number of reasons:

- A macro virus is platform independent
- Macro viruses infect documents, not executable portions of code
- Macro viruses are easily spread, as the documents they exploit are shared in normal use
- Because macro viruses infect user documents rather than system programs, traditional file system access controls are of limited use in preventing their spread

Worms

Similar to viruses but they don't attach to prexistig files

- A program that actively seeks out more machines to infect
  - Upon activation, the worm may replicate and propagate again
- To replicate itself, a worm uses some means to access remote systems:
  - Electronic mail or instant messenger facility
  - File sharing
  - Remote execution capability
  - Remote file access or transfer capability
  - Remote login capability

A worm typically uses the same phases as a computer virus:

- Dormant
- Propagation
- Triggering
- Execution

The propagation phase generally performs the following functions:

- Search for appropriate access mechanisms to other systems to infect by examining host tables, address books, buddy lists, trusted peers, and other similar repositories of remote system access details
- Use the access mechanisms found to transfer a copy of itself to the remote system and cause the copy to be run

- Scanning/fingerprinting
  - The function in the propagation phase for a network worm to search for other systems to infect
- Worm network scanning strategies:
  - Random
    - Each compromised host probes random addresses in the IP address space, using a different seed
    - Produces a high volume of Internet traffic, which may cause generalized disruption even before the actual attack is lunched
  - 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
    - Uses information contained on an infected victim machine to find more hosts to scan
  - Local subnet
    - If a host is 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

THese scans are normally easy to see on the network, meaning a firewall can pick it up. herefore they often attack sleep attack sleep o so on, to avoid detection

Worm Technology

Multiplatform
- Newer worms can attack a variety of platforms

Multi-exploit
- New worms penetrate systems in a variety of ways, using exploits against Web servers, browsers, e-mail, file sharing, and other network-based applications, or via shared media

Ultrafast spreading
- Exploit various techniques to optimize the rate of spread of a worm to maximize its likelihood of locating as many vulnerable machines as possible in a short time period

Polymorphic
- To evade detection, skip past filters, and foil real-time analysis, each copy of the worm has new code generated on the fly using functionally equivalent instructions and encryption techniques

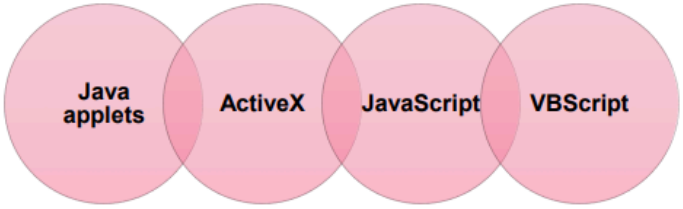
Metamorphic
- In addition to changing their appearance, metamorphic worms have a repertoire of behavior patterns that are unleashed at different stages of propagation

Transport vehicles
- Because worms can rapidly compromise a large number of systems, they are ideal for spreading a wide variety of malicious payloads

Zero-day exploit
- To achieve maximum surprise and distribution, a worm should exploit an unknown vulnerability that is only discovered by the general network community when the worm is launched

Mobile code

- Refers to programs that can be shipped unchanged to a heterogeneous collection of platforms and execute with identical semantics
- Transmitted from a remote system to a local system and then executed on the local system without the user's explicit instruction
- Often acts as a mechanism for a virus, worm, or Trojan horse to be transmitted to the user's workstation
- Popular vehicles for mobile code include:



Client-side vulnerabilities and Drive-by-downloads

- Drive-by-download
  - Exploits browser vulnerabilities so that when the user views a Web page controlled by the attacker, it contains code that exploits the browser bug to download and install malware on the system without the user's knowledge or consent
  - Does not actively propagate as a worm does, but rather waits for unsuspecting users to visit the malicious Web page in order to spread to their systems
- Watering-hole attacks are a variant of this used in highly targeted attacks
  - The attacker researches their intended victims to identify Web sites they are likely to visit and then scans theses sites to identify those with vulnerabilities that allow their compromise with a drive-by-download attack
- Malvertising is another technique used to place malware on Web sites without actually compromising them
  - The attacker pays for advertisements that are highly likely to be placed on their intended target Web sites, and which incorporate malware in them

Clickjacking

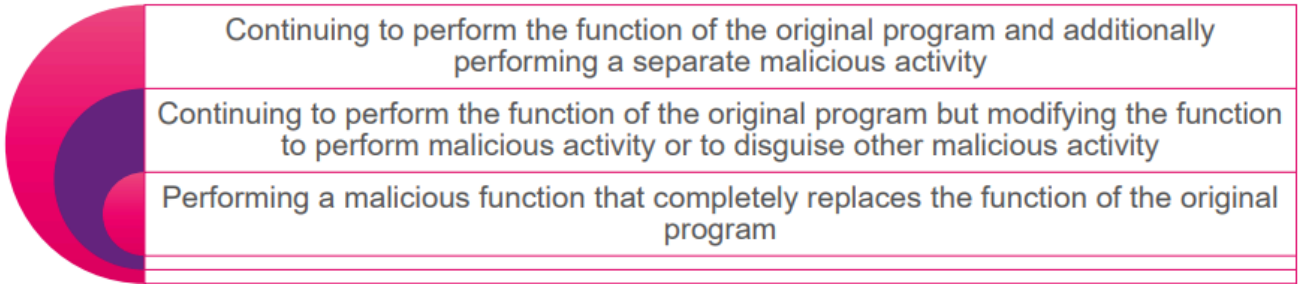
- Also known as a *user-interface (UI) redress attack*
  - Is a vulnerability used by an attacker to collect an infected user's clicks
  - The attacker can force the user to do a variety of things from adjusting the user's computer settings to unwittingly sending the user to Web sites that might have malicious code
  - Also, by taking advantage of Adobe Flash or JavaScript, an attacker could even place a button under or over a legitimate button, making it difficult for users to detect
  - A typical attack uses multiple transparent or opaque layers to trick a user into clicking on a button or link on another page when they were intending to click on the top level page
- Using a similar technique, keystrokes can also be hijacked
  - With a carefully crafted combination of stylesheets, iframes, and text boxes, a user can be led to believe they are typing in the password to their e-mail or bank account but are instead typing into an invisible frame controlled by the attacker

Spam

- Unsolicited bulk e-mail
- Imposes significant costs on both the network infrastructure needed to relay this traffic and on users who need to filter their legitimate e-mails
- Most recent spam is sent by botnets using compromised user systems
- Is a significant carrier of malware
- May be used in a phishing attack
- Although a significant security concern, in many cases it requires the user's active choice to view the e-mail and any attached document or to permit the installation of some program, in order for the compromise to occur

Trojan horses

- Is a useful, or apparently useful, program or utility containing hidden code that, when invoked, performs some unwanted or harmful function
- Can be used to accomplish functions indirectly that the attacker could not accomplish directly
- Fit into one of three models:

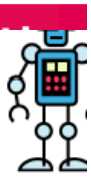


Payload – system corruption

- Once malware is active on the target system, the next concern is what actions it will take on this system

- Examples:
  - Data destruction on the infected system when certain trigger conditions were met
  - Display unwanted messages or content on the user's system when triggered
  - Encrypt the user's data and demand payment in order to access the key needed to recover this information (ransomware)
  - Inflict real-world damage on the system
    - Attempt to rewrite the BIOS code used to initially boot the computer
    - Target specific industrial control system software
  - Logic bomb
    - Code embedded in the malware that is set to "explode" when certain conditions are met

Payload –attack agent



- Malware subverts the computational and network resources of the infected system for use by the attacker
  - Bot (robot), zombie, drone
  - Secretly takes over another Internet-attached computer and then uses that computer to launch or manage attacks that are difficult to trace to the bot's creator

- A *botnet* is a collection of bots often capable of acting in a coordinated manner

Uses of bots

- Distributed denial-of-service (DDoS) attacks
- Spamming
- Sniffing traffic
- Keylogging
- Spreading new malware
- Installing advertisement add-ons and browser helper objects (BHOs)
- Attacking Internet Relay Chat (IRC) networks
- Manipulating online polls/games

Payload –information theft

Keylogger

- Captures keystrokes on the infected machine to allow an attacker to monitor user login and password credentials

Spyware

- Developed in response to efforts to try and stop keylogging
- Subvert the compromised machine to allow monitoring of a wide range of activity on the system which can result in significantly compromising the user's personal information

Phishing

- Exploits social engineering to leverage the user's trust by masquerading as communication from a trusted source

Spear-phishing

- An e-mail claiming to be from a trusted source, however, the recipients are carefully researched by the attacker, and each e-mail is carefully crafted to suit its recipient specifically

Payload –stealthing

- **Backdoor**
  - Also know as a *trapdoor*
  - Is a secret entry point into a program that allows someone who is aware of the backdoor to gain access without going through the usual security access procedures
  - Code that recognises some special sequence of input or is triggered by being run from a certain user ID or by an unlikely sequence of events
  - Usually implemented as a network service listening on some nonstandard port that the attacker can connect to and issue commands through to be run on the compromised system

Payload –stealthing

Rootkit

- A set of programs installed on a system to maintain covert access to that system with administrator (or root) privileges, while hiding evidence of its presence to the greatest extent possible
- Alters the host's standard functionality in a malicious and stealthy way
- An attacker has complete control of the system and can add or change programs and files, monitor processes, send and receive network traffic, and get backdoor access on demand
- Hides by subverting the mechanisms that monitor and report on the processes, files, and registries on a computer

Rootkits

- Can be classified using the following characteristics:

Persistent	• Activates each time the system boots
Memory based	• Has no persistent code and therefore cannot survive a reboot
User mode	• Intercepts calls to application program interfaces (APIs) and modifies returned results
Kernel mode	• Can intercept calls to native APIs in kernel mode
Virtual machine based	• Installs a lightweight virtual machine monitor and then runs the operating system in a virtual machine above it
External mode	• Malware is located outside the normal operation mode of the targeted system, in BIOS or system management mode, where it can directly access hardware

Countermeasures

- Elements of prevention:

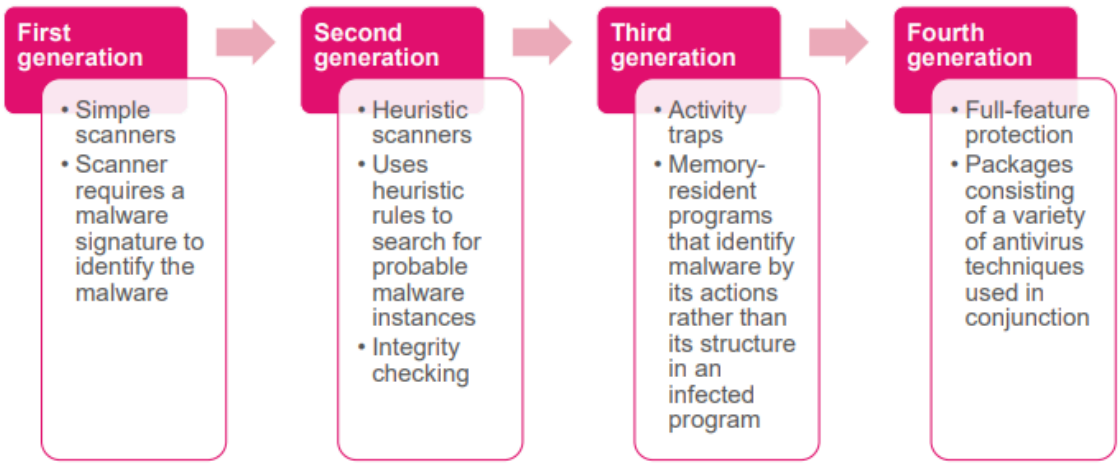
- One of the first countermeasures that should be employed is to ensure all systems are as current as possible, with all patches applied, in order to reduce the number of vulnerabilities that might be exploited on the system
- The next is to set appropriate access controls on the applications and data stored on the system, to reduce the number of files that any user can access, and hence potentially infect or corrupt, as a result of them executing some malware code
- The third common propagation mechanism, which targets users in a social engineering attack, can be countered using appropriate user awareness and training

Malware countermeasure approaches

- If prevention fails, then technical mechanisms can be used to support the following threat mitigation options:
  - Detection
  - Identification
  - Removal
- Requirements for effective malware countermeasures:
  - Generality
  - Timeliness
  - Resiliency
  - Minimal denial-of-service costs
  - Transparency
  - Global and local coverage

Host-based scanners

- Four generations of antivirus software:



Host-based behaviour-blocking software

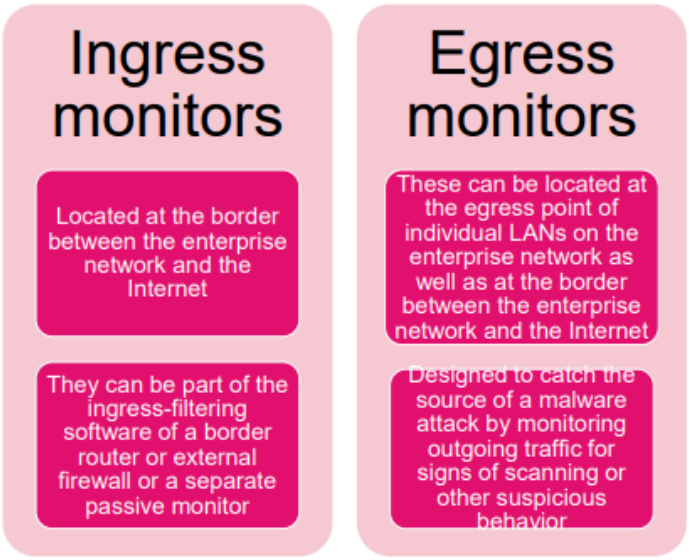
- Integrates with the operating system of a host computer and monitors program behaviour in real time for malicious actions
- The software then blocks potentially malicious actions before they have a chance to affect the system
- Can block suspicious software in real time so it has an advantage over antivirus detection techniques such as fingerprinting or heuristics
- Limitations:

  Because the malicious code must run on the target machine before all its behaviours can be identified, it can cause harm before it has been detected and blocked

Perimeter scanning approaches

- Antivirus software is used on an organisation's firewall and IDS
  - Typically included in e-mail and Web proxy services running on these systems
  - May also be included in the traffic analysis component of an IDS

Two types of monitoring software may be used:



Perimeter worm countermeasures

- Classes of worm defence:

(Class A) Signature-based worm scan filtering

- This type of approach generates a worm signature, which is then used to prevent worm scans from entering/leaving a network/host

(Class B) Filter-based worm containment

- This approach is similar to class A but focuses on worm content rather than a scan signature

(Class C) Payload-classification-based worm containment

- These network-based techniques examine packets to see if they contain a worm

Perimeter worm countermeasures

(Class D) Threshold random walk (TRW) scan detection

- Exploits randomness in picking designations to connect to as a way of detecting if a scanner is in operation

(Class E) Rate limiting

- This class limits the rate of scanlike traffic from an infected host

(Class F) Rate halting

- This approach immediately blocks outgoing traffic when a threshold is exceeded either in outgoing connection rate or in diversity of connection attempts

Worn Monitors