

# Malware overview

---

- Malicious program designed
  - to cause damage to systems
  - give system access to its creators
- Includes viruses, worms, trojans, ransomware, rootkits, spyware, adware, scareware, crapware, roughware, crypters, keyloggers, botnets etc.

## Malware sources

---

- **Instant messenger applications**
  - E.g. WhatsApp, LinkedIn, Google Hangout etc.
- **Portable hardware media / removable devices**
  - E.g. flash drives, CDs/DVDs etc.
  - AutoRun (Autostart)
    - Windows Windows to run executable when a device is plugged in
    - Exploited by malware to run malicious code
    - 💡 Best practice to disable
- **Browser and email software bugs**
  - Older software has known vulnerabilities, always use latest versions.
- **Insecure patch management**
  - Unpatched software are risky and has vulnerabilities e.g. [MS Word](#), [Excel](#), [Adobe Acrobat Reader](#)
- **Rogue / decoy applications**
  - By luring victim into downloading free software
  - 💡 Webmaster should do antivirus / anti-trojan scans of distributed files
- **Untrusted sites and freeware web applications/software**
  - Many hack tools may include trojans
  - 💡 Users should scan the files before executing
- **Downloading files from Internet**
  - Trojans can be distributed through e.g. music players, games, screensavers, Word/Excel macros, audio/video files, and video subtitles.
- **Email attachments**
  - Most common way to transmit malware
  - E.g. invoice, job letter, loan approval letter etc.
  - 💡 Always confirm sender's email address
- **Network propagation**
  - E.g. mistakenly allowing Internet traffic into private networks when replacing firewalls.
  - [Blaster worm](#) infects sequential IP addresses.
- **File sharing services**
  - Open ports for file sharing or remote execution can be used by others to access systems
  - E.g. NetBIOS on port 139, FTP on port 21 and SMB on port 445
  - Turn off file and printer sharing
- **Installation by other malware**



- **Bluetooth and wireless networks**
  - Attackers set-up open Bluetooth and Wi-Fi networks to attract users
  - Allows attackers to inspect network traffic and find e.g. username and passwords

## Malware distribution techniques

---

- **Blackhat SEO**
  - Also known as *spamdexing*, *search engine spam*, *search engine poisoning*, *black-hat search engine optimization*, *search spam* or *web spam*.
  - Methods to make malware websites rank higher in search engine results
- **[Clickjacking](#)**
  - Tricking users into downloading malware with seemingly innocuous objects.
- **Spear phishing**
  - Spear phishing is phishing directed at specific individuals or organizations.
  - E.g. can mimic government institutions
- **Malvertising**
  - Injecting malicious advertisements into legitimate online advertising networks
- **Compromised websites**
  - Distributing malware through a compromised website
- **Drive-by downloads**
  - Downloads that happens without users knowledge or understanding of consequences
  - Can be done e.g. by exploiting vulnerabilities in browsers, email clients.

## Spam emails

-  **Relaying**
  - When email is accepted and then delivered to a non-local email address
-  **Open relay**
  - Allows anyone to send an e-mail without authentication
  - Allows e-mail spoofing (email messages with a forged sender address)
  - Was the default configuration in old internet but got abused by spammers/worms
  - Usually blacklisted

## Malware components

---

- **Payload**
  - Core component of malware, designed to execute its actual motive
- **Command and control (C&C)**
  - Remote control center for the malware
- **Crypter**
  - Software that makes malware harder to detect by security programs
  - It encrypts, obfuscates, and manipulates the malware
  - E.g. [BitCrypter](#)
- **Downloader**
  - Requires network resource to get malware from internet
- **Dropper**
  - Has malware embedded and drops it to the system
- **Exploit**

- Takes advantage of a software vulnerability
- May be used to deliver malware
- **Injector**
  - Malware that injects itself (or other malware) into other processes or files
- **Malicious code**
  - Code that gives malicious functionality to the malware
- **Protectors**
  - Prevents tampering and reverse engineering of programs.
  - Usually includes packing and encrypting

## Obfuscator

- Usually a packer or protector for encrypting or compressing the malware
- Goal is
  - to make reverse engineering difficult
  - to make malware undetectable from antivirus scans

## Packer

- Short for **runtime packers** which are also known as **self-extracting archives**.
- Software that unpacks itself in memory when the "packed file" is executed
- Smaller footprint on infected machine
- Make reverse engineering more difficult

## Exploit kit

- Collection of pre-written exploits in a simple one-in-all tool for managing exploits together.
- Automates 5 steps of hacking
  1. **Reconnaissance:** Gathers information on the victim machine
  2. **Scanning:** Find vulnerabilities and determines the appropriate exploit
  3. **Gaining access:** Executes malware typically through silent drive-by download
  4. **Maintaining Access:** Run post-exploitation scripts to maintain further access
  5. **Covering Tracks** by e.g. erasing logs
- E.g. RIG Exploit Kit
  - Has been used to deliver many types of malware
  - Monthly subscription fee, sold in cybercriminal circles
  - spread via suspicious advertisements that have been inserted into legitimate websites

## Malware types

---

### Virus

- Designed to replicate itself to other programs and documents on the infected machine.
- Spread to other computers with the transfer of the infected files or programs.
- Transmitted through file transfers, infected flash drives, and email attachments.
- See also [viruses](#)

## Worm

- Replicates itself across network connections, e.g. bluetooth, wireless.
- Exploits vulnerabilities on the victim machines
- E.g. [Broadpwn](#) where the worm could run code on Android iOS that has WiFi turned on.

## Ransomware

- Hackers restrict access to files and folders on the target system until a payment is made.
- Victims are usually required to pay money to access their files.
- Often encrypts own files and sells decryption key.
- An indicator is that your CPU runs on higher frequencies.
- 💡 Best practices
  - Do not pay as there's no guarantee that you'll get the key
  - Keep back-ups somewhere offsite e.g. in cloud
- E.g. • Cryptobit • Cryptolocker • Cryptodefense • Cryptowall • police-themed

## Backdoor

---

- Also known as ***trapdoor, trap door, back door, back-door, trap-door.***
- 📄 Provides access to a computer program that bypasses security mechanisms
- Sometimes installed by developers for e.g. troubleshooting purposes or just by mistake.
- Often created by e.g. trojans and worms as means of delivery

# Trojan

---

- Malware contained inside seemingly harmless programs.
  - activated when such programs are executed.
- Used to gain access and/or cause damage to victims systems.
- Run with same privileges of the victim but can exploit vulnerabilities to gain more privileges.
- Symptoms include
  - change of system settings such as disabling updates, antivirus, task manager
  - more usage of system resources such as network bandwidth and CPU.
- Broad use-cases including
  - Install other malicious code
  - Use victims computer for other attacks including DDoS, or spam e-mails
  - Steal information through keyloggers
  - Running a ransomware
  - Infect victim as proxy-server to do replay attacks

## Trojan communication

---

- Different trojans use different ports for communication
- 💡 Check active connections on different ports to detect presence of trojans.

## Communication paths

### Overt channels

- Legitimate and transparent paths to send information
- E.g. HTTP and TCP/IP
- Can be exploited to create a [covert channel](#)

### Covert channels

- 📄 Sending information by an unknown, unmonitored way
- Outside of the security policy
- Useful to bypass multi-level security solutions in order to leak data out of a protected network
- May use [steganography](#).
  - E.g. storage channel
    - Reading tweets from Twitter to get commands from C&C servers
    - Leaves evidence behind
  - E.g. timing channel
    - Small pauses when watching a video sends encoded commands
    - Leaves almost no trace of its existence
    - Requires receiver to be actively listening
  - E.g. use of reserved fields in various packet headers/footers to conceal data
- Utilizes tunneling protocol (allows moving data between different networks)

## Trojan tools

---

- **Wrapper**
  - An application that can concatenate two executable files and produce an application containing both.
  - Used to embed trojans in legitimate files
  - Can utilize e.g. [petite.exe](#), [IExpress](#), [elitewrap](#)
- **Trojan Construction Kits**
  - Allows you to create a trojan in an easy way
  - E.g. DarkHorse trojan virus maker



## Steps of infecting with a trojan

---

1. Create a new trojan
2. Create a dropper to install the trojan
3. Create a wrapper to bind trojan into legitimate files
4. Propagate the trojan

## Techniques for evading antivirus

---

- Do not use a known trojan, it'll be known by antivirus
  - Write your own trojan instead
-  Distribute trojan as e.g. `.doc.exe` or `.pdf.exe`
  - Because Windows hides "known extensions" by default so they appear as `.doc` or `.pdf`
-  Perform code obfuscation or morphing to confuse anti-viruses
  - E.g. `alert( 'Hello, world!' );` becomes `var _0xc890=[\"\\x68\\x65\\x6C\\x6C\\x6F\\x20\\x77\\x6F\\x72\\x6C\\x64\"];alert(_0xc890[0])`
- Change the content / checksum or morph it to generate different signatures

## Trojan types

---

### Remote access trojans (RATs)

- Also known as **remote administration trojans**.
- Malware that includes a back door for administrative control over the target computer
- Includes an user interface to issue commands
- Usually has functionalities like keylogger, camera access, taking screenshots etc.
- E.g. • [Saefko](#) • [njRAT](#) • [turkojan](#) • [Biodox](#)

### Covert Channel Tunneling Trojan (CCTT)


- A form of RAT
- Enables attackers to gain shell interfaces into and out of a network using authorized channels covertly

### Backdoor trojans

- Trojans that installs backdoors to give uninterrupted access to attackers
- The difference from [RAT](#) is that RATs have user interface
- Can usually bypass programs e.g. by injecting connections into browser processes
- Often used to create a botnet or zombie network to execute malicious activities
- E.g. • [Qadars](#) • [z3r0 Remvio](#) • [SubRoot](#)

- [QAZ Trojan \(TROJ\\_QAZ\)](#)
  - Also known as notepad trojan
  - Replaces notepad.exe on the system in an effort to hide
- See also [backdoor](#)

## Botnet trojans

- **Bot herders** are attackers who install bot programs on victims.
- Infected machines become one of their **bots** or **zombies** in their **bot herd**.
- Bots are controlled through Command and Control (C&C) center.
-  Bots allow attackers to
  - do DDoS attacks
  - steal data
  - send spam and access the device
- Examples
  - [Conficker](#)
    - Has also worm features to infect other systems in the network.
  - [Mirai](#)
    - Infects weak IoT devices.
    - Probes IoT devices in network and brute forces login on Telnet (port 23 and 2323)
    - [Open-sourced](#)
- See also [Botnet](#) and [Botnets | Denial of Service](#)

## Rootkit trojans

- Enable access to unauthorized areas in a software
- Root (privilege account in Unix) + kit (software components that implement it)
- Type of backdoors but hard to detect as it often masks its existence
  - E.g. by subverting software that's intended to find it such as hiding its name from service lists, task lists or registry viewers.
- Does not propagate by themselves as opposed to [worms](#)
- Often used in blended threat
  - **Blended threat** is an exploit that combines elements of multiple types of malware
  - E.g. a malware consisting of
    - dropper (to install)
    - loader (causes e.g. buffer overflow and load rootkit into memory)
    - rootkit.
- Commonly hidden in the [boot sector](#) of a hard disk to evade antivirus detection.
- E.g.
  - [FinFisher](#) - government grade spyware
  - [EquationDrug](#) - by NSA sponsored [Equation Group](#)
  - [Boot.Phishar](#) - affects MBR (master boot record), starting before OS
- See also [Rootkits | Hiding Files](#)

## E-banking Trojans

- Intercepts account information before encryption and sends to attacker.
- Can steal e.g. credit card numbers, billing details
- Can also show false bank account information
- E.g. Zeus (ZBot)
  - Uses man-in-the-browser keylogging and form grabbing
  - One of the most successful banking trojans
  - Used **fast flax** to evade detection
    - Uses compromised hosts as proxies for commands
    - Idea is to change DNS record of domain very quickly using hundreds of IPs


## Banking information analysis

- Keylogging
- Form data capture
- Inserting fraudulent form fields
- Screen captures and video recording
- Mimicking financial websites
- Redirecting to banking websites
- Man-in-the-middle attack

## Tan Gabber

- **Transaction Authentication Number (TAN)**
  - Single use [one-time passwords \(OTPs\)](#) to authorize financial transactions
  - E.g. ChipTAN
    - A card needs to be inserted to a device to get the code
    - Used by many German and Austrian banks
- Trojan intercepts the number and replaces it
  - User gets rejected
  - Attacker logs in using target's login details.

## HTML injection

- Also known as **Webinjects**
- Injects HTML or JavaScript code into e-banking content before it's rendered on a web browser
-  Manipulates original forms in bank webpages with additional fields
  - E.g. login credentials, credit card numbers, CVVs, PINs, tokens, etc.
- Goal is to prompt user to give out more information that'll be collected

## Form Grabber

- Retrieves authorization and log-in credentials from a web forms before they're sent
- More effective than keyloggers as it acquire credentials even if they use virtual keyboard, autofill etc.



### **Covert credential grabber**

- Hides itself on a machine
- Searches through session cookies for financial transaction info
- Sends the information the attacker

### **Proxy-server trojans**

- Allows attacker to use victims computers as proxy to connect to the Internet.
- Starts a hidden proxy server on victim machine
- Used for attackers for illegal activities such as purchasing goods with illegal cards
- E.g. Linux.Proxy.10, [Pinkslipbot](#)

### **Defacement trojan**

- Resource editors allow to view, edit, extract, and replace strings, bitmaps, logos and icons from any Windows program.
- E.g. changes title of Word documents to "You've been hacked"
- See also [Website defacement](#) | [Web threats and attacks](#)
- E.g. using [Restorator](#) to modify files' icons.


# Viruses

---

## Virus type

---

### Stealth virus

- Virus takes active steps to conceal infection from antivirus
-  Characteristic behaviors
  - Restores original file timestamp
  - Intercepts system calls to play back original information of file to e.g.
    - change system libraries to hide its existence from antiviruses
    - run the rootkit


### Tunneling virus

- Backtracks interrupt chain to go directly to DOS and BIOS interrupt handlers
- Avoids monitoring
- Kernel software protected in other OS
- Legacy, was only possible in MS-DOS

### Logic Bomb virus

- Not self-replicating, zero population growth, possibly parasitic
- Consists of
  - Payload
  - An action to be performed
  - Trigger
    - Boolean condition to be executed
- E.g. if Bob is not getting paid then delete the [cloudarchitecture.io](https://cloudarchitecture.io) website

### Polymorphic virus


-  Modifies their payload to avoid signature detection
- Mutates its payload and usually encrypts it.
- Can hide file changes against simple checksums

### Metamorphic virus

- Viruses that can reprogram/rewrite itself.
- In polymorphic virus, the mutation engine is always the same while payload is mutated, metamorphic virus can also mutate its own mutation engine.
- Usually
  - Inserts dead code
  - Reshapes the expressions
  - Reorders instructions
  - Encrypts program code
  - Modifies the program control structure

- E.g. [Win32/Simile](#) and [Zmist](#)

## Macro virus

- Changes or creates new macro for MS Office products
-  **Macros**
  - Code that is part of documents.
  - Used extensively in MS Office Tools
    - Written in or translated to Visual Basic for Applications (VBA) code
  - **Macro language**: a programming language which is embedded inside a software application
- **Protective strategies**
  - Later versions of MS Office have security levels for execution of macros
    - Level high only executes signed macros
  - MS Office provides warnings when files contain macros
- E.g. [Concept](#), first macro virus for Microsoft Word (1995-1997)
  - Infects Word's global document-template `NORMAL.DOT`
  - Creates `PayLoad` and `FileSaveAs` macros
  - Infects all documents saved with the Save As command
- E.g. [Laroux](#), first macro virus for Microsoft Excel (1996)
  - Consists of `auto_open` and `check_files`
    - `auto_open` executes whenever an infected spreadsheet is opened, followed by `check_files`
    - Virus looks for `PERSONAL.XLS`
  - Virus contains no malicious payload



## File infectors

- Virus infects executables

## Appending virus

- At the end
- To get control
  1. Save original instruction in code
  2. Replace by jump to viral code
  3. Execute virus
  4. Restore original instruction and jump to them
    - or run original instruction at saved location followed by jump to the rest of the code

## Overwriting file virus

-  Also known as **cavity virus** or **spacefiller virus**
-  Houses itself in target files without altering their size.
- Virus gets control in normal execution of file
- Placement Strategies

- Place virus in superfluous data
- Place virus in file slack or unused allocated file space
- Stash overwritten contents in a companion file
- Compress (parts of) the original file, decompress
- E.g. [Lehigh](#) (an early DOS virus)


## Inserting virus

- Move target code out of way
- Intersperse small pieces of virus with infected file

## Companion virus

- Virus gets executed before infected file
- Infected file barely changed
- Examples
  - Change name of target file
    - Copy `notepad.exe` to `notepad.exp`
    - Virus is in new `notepad.exe`, which calls `notepad.exp`
  - Virus placed earlier in search path
    - `notepad.exe` in a different directory than real `notepad.exe`
    - `notepad.com` is executed before `notepad.exe`
  - Use Windows registry to change association for `.exe` files
  - Change interpreter in ELF files
    - Typically the run-time linker, but now virus
  - Associate icon of target with virus

## Boot sector infectors

- Contains code that runs when a system starts up.
- Also known as **boot sector virus**
-  Copies itself into the MBR or VBR on hard disk
  - Typically after making copy of MBR in a "safe location"
- Extinct in the wild
  - Floppies are rarely used to boot, disabling the propagation mechanism
  - OS prevent writing to a disk's boot sector without proper authorization
  - BIOS can enable boot block protection
- E.g. [Michelangelo](#) (1991)
  - Moves original boot sector to safe location
  - Infects all floppy disks inserted into computer
  - Payload: overwrites file system with zeroes
- E.g. [Stoned Virus](#) (1988)
  - Infects 360KB diskettes and MBR
  - Many variants
  - Payload: Shows "Your PC is now stoned!"


## Boot record types

- Volume Boot Record
  - First sector of an unpartitioned storage device
  - First sector of an individual partition
- Master Boot Record
  - First sector of data storage device that has been partitioned

## Booting

- Bootstrap loader
  - Loads software to start OS
- Multi-stage bootstrap loader
- Boot sequence on IBM-PC
  - Runs instruction at memory location F000:FFF0 of BIOS
  - Jumps to execution of BIOS startup program
  - Executes Power-On Self-Test (POST)
    - Checks, initializes devices
  - Goes through preconfigured list of devices
  - If it finds bootable device, loads, and executes boot sector
    - Assume MBR on hard drive
    - MBR contains address of bootable partition
    - Load boot sector of bootable partition
    - Boot sector moves OS kernel into memory and starts it

## Multipartite viruses

- Also known as **hybrid virus**
-  Combines [file infector](#)s and [boot record infectors](#)
- Re-infects a system repeatedly
- In order for it to be eradicated, the whole virus has to be removed from the system
- E.g. [Ghostball](#), first multipartite virus (1989)
  - Infects both executable .COM-files and boot sectors.

## Other virus types

- **Camouflage** virus: Disguise as legit files.
- **Network**: Spreads via network shares.
- **Shell virus**
  - Like [boot sector](#) but wrapped around application code, and run on application start.
- **Sparse infector**
  - Only fire when a specific condition is met
  - E.g. a virus which infects only the 20th time a file is executed.



# Malware analysis

---

- Reverse engineering of a malware program
- Purpose is to
  - determine how the malware works
  - assess the potential damage it could cause
- Helps find and remove the infections that exist in a system through using designed tools and techniques.

## Malware analysis types

---

### Static malware analysis


- Analyzing the malware without running or installing it
- Malware's binary code is examined
- Checks for any data structures or function calls that have malicious behavior.

### Dynamic malware analysis

- Requires the malware program to be running in a monitored environment such as sandbox or a virtual machine.
- Helps in understanding how the malware works by monitoring its activities on the system.

## Windows integrity monitoring

### Port monitoring

- Involves monitoring services running on different ports.
- Features can include
  - analytics for packet rates, CPU, power, and bandwidth of ports
  - mirroring the traffic from one port to another
-  Tools include
  - `netstat` (terminal)
    - Displays network connections, available on many OSes
    - E.g. `netstat -an` to display all connections and listening ports (`-a`) in a numerical format `-n`
  - [TCPView](#) (GUI)
    - Windows tool to enumerate network connections and owner processes
    - Refreshes automatically
  - [CurrPorts](#) (GUI)
    - View open ports and connections per process on Windows
- See also • [Common ports to scan](#) | [Scanning networks](#) • [Common ports and services to enumerate](#)

## Process monitoring

- Use e.g. [Process Monitor](#) to see what processes malware starts
- Built-in `sc` command provides all sorts of information about running services on a Windows machine.
  - E.g. `sc query` to lists the running services

## Registry monitoring

- Registry contains information, settings, options, and other values for programs and hardware installed on all versions of Microsoft Windows operating systems.
- Malware modifies registry including keys such as `Run`, `RunServices`, `RunOnce`, `RunServicesOnce`, `HKEY_CLASSES_ROOT\exefile\shell\open\command "%1" %*`.
- Use native `regedit` or e.g. [RegScanner](#), [Registry Viewer](#), [Active Registry Monitor](#) to monitor registry changes.

## Windows services monitoring

- Malware usually install and run themselves as services.
- Use e.g. [Windows Service Manager \(SrvMan\)](#), [Process Hacker](#), [AnVir Task manager](#) to monitor services

## Startup programs monitoring

- Malware modify startup settings to execute themselves when system starts
- Check:
  - Startup registry keys
  - Automatically loaded drivers
  - `boot.ini` or `bcd (bootmgr)` entries
  - Services that starts automatically in `services.msc`
  - Startup folder
- Tools include [Autoruns for Windows](#), [Autorun Organizer](#), [WinTools.net: Startup Manager](#)

## Event logs monitoring/analysis

- Analyze logs on IDS/IPS, web servers, authentication servers etc.
- In Windows you can use Event Viewer to see system, application and security logs
- Tools include [Loggly](#), [SolarWinds Security Event Manager \(SIEM\)](#), [Splunk](#)

## Installation monitoring

- See what has been modified during installation process
- Tools include [SysAnalyzer](#), [Mireksoft Install Monitor](#), [Revo Uninstaller Pro](#)

## Files and folder monitoring

- Scan system files for suspicious files and folders
- Tools include:
  - `Sigverif`
    - Built-in Windows tool
    - Identifies unsigned drivers
  - [Tripwire File Integrity Manager](#)
  - [CSP File Integrity Checker](#).



## Device drivers monitoring

- Malware installs with some infected drivers
- Drivers can be seen by: Run -> `msinfo32` -> Software Environment -> System Drivers
- Tools include [DriverView](#), [Driver Booster](#)

## Network traffic monitoring/analysis

- Includes capturing traffic to look for malware activity
- Tools for capturing and monitoring include: [Wireshark](#), [Capsa Network Analyzer](#)

## DNS monitoring/resolution

- DNSChanger is a DNS hijacking Trojan that can point DNS entries toward malicious name servers.
- Use e.g. [DNSQuerySniffer](#), [DNSstuff](#).

## API calls monitoring

- Malware use Windows APIs to perform malicious task
- API call monitoring tools include [API Monitor](#), [Runscope](#)

## System baselining

- Allows monitoring security configuration changes over time
- Flow
  1. Take snapshots before and then after malware execution.
  2. Compare the snapshots to understand changes made by the malware.

## Unix integrity monitoring

- Display processes: `ps -ef`
  - `-e`: selects all processes
  - `-f`: switch provides a full listing

# Sandboxing

---

- Technique in which you create an isolated test environment
  - Allows secure experimentation
  - Nothing (no harm) can be spilled out of the environment.
    - If something happens, the damage is confined to that sandbox
- Examples
  - **Chrome web-browser**
    - Sandboxing through multi-process architecture.
    - One or more processes are assigned to run scripts of each site.
    - Each Chrome extension and app runs in its own process
  - **Virtual machines**
    - Good for testing / reverse engineering malware
    - E.g. YouTubers messing with scammers utilizes virtual machines, [video](#), [video](#)
    - 💡 Good hypervisor is important to ensure nothing goes out of the environment.
      - E.g. KVM (used by AWS) is good on AWS, and Hyper-V in Windows

- KVM installation in Fedora: `dnf install @virtualization` and then `virt-manager` to start a GUI.
- VirtualBox is not as feature rich.
- 💡 Make sure host environment is safe in first place
  - E.g. in Linux you can enable [Security-Enhanced Linux](#) (SELinux).
    - Supported by Fedora, Debian, Ubuntu, used by default by Android.
    - `setenforce 1` to enable, `getenforce` to query status

## Anti-malware software

---

- Includes e.g. antivirus, anti-spyware, anti-trojans, anti-spamware, anti-phishing, and email scanners.
- Helps detecting, mitigating, preventing and repairing any damage by malware.
- Looks for behavior typical to viruses and give warnings.
- Looks for already known virus signatures and warns the user if a threat is found.
- E.g. Kaspersky, McAfee, AVG, Norton, Avira, Bitdefender

## Detection types

- **Signature-based**
  - Compare file hash and malware hash
  - ⚠ Anything new or custom written will not be detected
- **Rule-based (behavior-based)**
  - 📝 Relies on differentiating expected vs anomalous behavior
  - Analyzes certain characteristics of a program.
    - E.g. application accessing user login file. Why?
  - Can utilize AI & ML to decide whether something is a malware.
- **Sandboxing**
  - Creates environment, lets program run and examines its behavior.
  - Good to find out behavior of e.g. self-modifying code, encrypted code.

## Virus detection methods

---

- **Scanning**
  - Scans malware for known signatures (characteristics)
  - ⚠ Only known and predefined viruses can be detected
- **Integrity checking**
  - Verifies files against their recorded integrated data
- **Interception**
  - Intercepts the virus if it detect suspicious behavior (e.g. network access) and asks user if the user wants to continue.
  - Useful for logic bombs (only executed if certain conditions are met) or trojans
- **Code emulation**
  - Executes a virtual machine mimicking CPU and memory
  - Useful against encrypted, polymorphic or metamorphic viruses
- **Heuristic analysis**
  - Helps in detecting new or unknown viruses
  - **Static:** anti-virus decompiles and analyzes the binary


- **Dynamic:** anti-virus runs code emulation to determine if the code is viral
- Prone to many false positives

## Malware countermeasures


---

- Use up-to-date anti-virus, firewall and intrusion detection software with regular scans
- Block all unnecessary ports at the host and firewall.
- On Windows
  - Enable Windows Defender
  - Enable [Data Execution Prevention \(DEP\)](#)
  - Run registry monitoring tools to find malicious registry entries added by the backdoor
- Enable [Address space layout randomization \(ASLR\)](#)
- Do not open files with more than one file type extension
- Use [anti-malware software](#)
- Avoid accepting executables sent as messages or downloaded from untrusted sources.
- Inspect network packets using protocol monitoring tools

## Data Execution Prevention (DEP)

-  Marks memory regions as non-executable, such that an attempt to execute machine code in these regions will cause an exception
- Executable space protection in Windows
- Read more on [Data Execution Prevention | Microsoft Docs](#)

## Address space layout randomization (ASLR)

-  Prevents exploitation of memory corruption vulnerabilities.
- Involves randomly positioning the base address of an executable and the position of libraries, heap, and stack, in a process's address space
- Breaks assumptions that attackers could make about where programs and libraries would lie in memory at runtime