* **Malware:**
  + Describes a wide range of software that is intentionally designed to cause harm to systems and devices, networks, or users.
  + **Ransomware:**
    - Malware that takes over a computer and then demands a ransom.
      * Includes **crypto malware**, which encrypts files then holds them hostage until a ransom is paid.
      * Backup systems are the best defense against ransomware.
    - **Trojan Horses:**
      * A malware disguised as legitimate software, requiring a user to accidentally install it.
        + **Remote Access Trojans (RATs):**

Provide attackers with remote access to systems.

Popularly called **Stalkerware** because they are used by people spying on their partners.

* + - * Typically combatted with security awareness; don’t download random shit without checking.
  + **Worms:**
    - Unlike Trojans, worms can spread without user interaction, self-installing.
      * Spread through email attachments, network file shares, or other methods as well.
      * **Stuxnet:**
        + The 2010 Stuxnet attack is generally recognized as the first implementation of a worm as a cyber weapon, which copied itself to thumb drives to bypass air-gapped (physically separated systems without a network connection) computers.
  + **Rootkits:**
    - Malware specifically designed to allow attackers to access a system through a backdoor.
      * Many modern rootkits also include capabilities that work to conceal the rootkit from detection through any of a variety of techniques, ranging from leveraging filesystem drivers to ensure that users cannot see the rootkit files, to infecting startup code in the **master boot record (MBR)** of a disk, thus allowing attacks against full-disk encryption systems.
      * Difficult to detect and even remove. Thus, best course of action after detection is to rebuild the system from a “good” backup.
      * Some are intentionally installed, either as part of digital rights management (DRM) systems or as part of anti-cheating toolkits for games.
    - Like any malware type, the best prevention is normal security practices, patching, secure configurations, and ensuring that privilege management is used.
  + **Backdoors:**
    - Methods or tools that provide access that bypass normal authentication and authorization procedures, allowing attackers access to systems, devices, or applications.
      * Can be **hardware or software based**, but in most scenarios are only software based.
        + Sometimes computer manufactures pre-install backdoors but those can be controversial and unsafe.
      * Backdoors may use an unexpected open port and services or be more sophisticated and leverage existing services including web-based backdoors that require a different URL under the existing web service. There are also backdoors that conceal their traffic by tunneling out to a remote-control host using encrypted or obfuscated channels.
  + **Bots:**
    - Remotely controlled systems or devices that have a malware infection.
      * **Botnet:**
        + A group of bots, typically used by attackers who control them to perform various actions, ranging from additional compromises and infection to denial-of-service attacks or acting as spam relays.
        + Large botnets may have hundreds of thousands of bots involved in them, and some have had millions of bots in total.
        + Many botnet **Command and Control (C&C)** systems operate in a client-server mode, in which each bot will contact a central control system, which provides commands and updates, and tracks how many systems are in the botnet.
        + **Internet Relay Chat (IRC)** was frequently used to manage client-server botnets in the past, but many modern botnets rely on **secure HTTP (HTTPS)** traffic to help hide C&C traffic and to prevent it from easily being monitored and analyzed by defenders.

**C&C servers are the core of a botnet,** allowing attackers to manage the botnet, and utilize tools to help steal data, conduct distributed denial-of-service attacks on a massive scale, deploy and update additional malware capabilities, and respond to attempts by defenders to protect their network.

* + - * **Peer-to-Peer Botnet:**
        + Unlike a client-server botnet, this botnet has some bots connect to the server, but also connect to other bots as well. This means that even if a C&C serve is taken down, the bots can still operate. Additionally, this makes tracking an attack so much more difficult as well.
      * Many botnets use **fast flux DNS**, which uses many IP addresses that are used to answer queries for one or more fully qualified DNS names. **Frequent updates (fast flux)** mean that the many systems in the network of control hosts register and de-register their addresses, often every few minutes on an ongoing basis.
        + More advanced techniques also perform similar rapid changes to the DNS server for the DNS zone, making it harder to take the network down. Techniques like that can be defeated in controlled networks by forcing DNS requests to organizationally controlled DNS servers rather than allowing outbound DNS requests.
        + Logging all DNS requests can also provide useful information when malware hunting, because machine-generated DNS entries can frequently be easily spotted in logs.
        + Although IRC was commonly used for botnet control, new botnets often use fast flux DNS and encrypted C&C channels disguised as otherwise innocuous-seeming web, DNS, or other traffic.
      * Taking down the domain name is the best way to defeat a fast flux DNS-based botnet or malware, but not every DNS registrar is helpful when a complaint is made.
      * Detecting botnets is often accomplished by analysis of bot traffic using network monitoring tools like IPSs and IDSs and other network traffic analysis systems.
        + Additional data is gathered through reverse engineering and analysis of malware infections associated with the bot.
        + Malware can be detected with antivirus, antimalware, and endpoint detection tools.
    - **Botnets and Distributed Denial-of-Service (DDoS) Attacks:**
      * Botnets can be used to attack services and applications, and distributed denial-of-service (DDoS) attacks against applications are one common application of botnets.
      * Botnets rely on a combination of their size, which can overwhelm applications and services, and the number of systems that are in them, which makes it nearly impossible to identify which hosts are maliciously consuming resources or sending legitimate-appearing traffic with a malicious intent.
      * Identifying a botnet-driven DDoS attack requires monitoring network traffic, trends, and sometimes upstream visibility from an Internet Service Provider.
      * **Security Information and Event Management (SIEM) Systems:**
        + Can help correlate data from multiple sources, to help identify a botnet.
      * Behavior analysis tools can also help differentiate a DDoS from more typical traffic patterns.
* **Keyloggers:**
  + Programs (or hardware) that capture keystrokes from keyboards, although keylogger applications may also capture other input like mouse movement, touchscreen inputs, or credit card swipes from attack devices.
    - Work in many ways, from capturing data from the kernel to APIs or scripts, or even directly from memory.
    - Preventing keyloggers is to just follow normal security practices to prevent it from being installed in the first place.
  + **Multifactor Authentication (MFA)** can also help prevent problems as even if the attacker has your password, they are still a step away from unlocking your account.
* **Logic Bombs:**
  + Are not independent malicious programs. They are functions or code that placed inside other programs that will activate when set conditions are met.
    - Sometimes they will activate when a specific date or set of conditions are met.
* **Viruses:**
  + Malicious programs that self-copy and self-replicate.
    - They require one or more infection mechanisms that they use to spread themselves, typically paired with some form of search capability to find new places to spread to.
  + **Trigger:**
    - The conditions for when the virus will execute.
  + **Payload:**
    - What the virus does, delivers, or the actions it performs.
  + Come in many varieties, including:
    - **Memory-Resident Viruses:**
      * Remain in memory while the system of device is running.
    - **Non-Memory-Resident Viruses:**
      * Execute, spread, and then shutdown.
    - **Boot Sector Viruses:**
      * Reside inside the boot sector of a drive or storage media.
    - **Macro Viruses:**
      * Use macros or code inside word processing software or other tools to spread.
    - **Email Viruses:**
      * Spread via email either as attachments or as part of the email itself using flaws within email clients.
* **Fileless Viruses:**
  + Attacks are like traditional viruses in a number of critical ways. They spread via methods like spam email and malicious websites, and they exploit flaws in browser plug-ins and web browsers themselves. Once they successfully find way into a system, they inject themselves into memory and conduct further malicious activity, including adding the ability to reinfect the system by the same process at reboot through a registry entry or other technique. At no point do they require local file storage, because they remain memory resident throughout their entire active life. In fact, the only stored artifact of many fileless attacks would be the artifacts of their persistence techniques, like the registry entry shown.
    - **Email link to malicious website.**
      * **Website exploits browser plug-in vulnerability.**
        + **Shell code runs command-line script to download and execute payload.**

**Payload runs in memory in memory and executes further attack.**

**Registry entry created to repeat shell code download and execution at boot.**

* + As can be seen, fileless attacks require a vulnerability to succeed, so ensuring that browsers, plug-ins, and other software that might be exploited by attackers are up to date and protected can prevent most attacks.
    - **Antimalware and IPSs can also help stop or detect this stuff.**
* **Spyware:**
  + Malware designed to obtain information about an individual, organization, or system. Various types of spyware exist, with different types of information targeted by each.
    - Many track users’ browsing habits, installed software, or similar information and report it back to a central server.
    - Some spyware is relatively harmless, but others can access sensitive data, and even allow remote access to web cameras and other systems.
  + Spyware is associated with identity theft and fraud, advertising and redirection of traffic**, digital rights management (DRM) monitoring**, and with **stalkerware**, a type of spyware used to illicitly monitor patterners in relationships.
  + Can be countered with antimalware and user awareness.
* **Potentially Unwanted Programs (PUPs):**
  + Programs that may not be wanted by the user but are not as dangerous as other types of malwares. They are typically installed without the user’s awareness or as part of a software bundle or other installation.
    - Include adware, browser toolbars, web browser-tracking programs, etc.
    - Can typically be countered with antimalware and user awareness.
    - **It is important to note that a system with a PUP is not necessarily compromised.**
* **Malicious Code:**
  + Scripts and custom-built code that isn’t malware but can have malicious actors. 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.
  + **PowerShell**, the built-in Windows scripting language, is a popular target because of the powerful capabilities it provides. It allows remote and local execution, network access, and man other capabilities. In addition, since it is available by default on Windows systems and is often not carefully monitored, attackers can leverage in many ways, including for fileless malware attacks where PowerShell scripts are executed locally once a browser or plug-in is compromised.
    - **Constrained Language Mode**: Can limit sensitive commands in PowerShell.
    - **Windows Defender Application Control tool or AppLocker** can validate scripts and limit which modules and plug-ins can be run.
    - Logging can also be turned on for PowerShell for command-line auditing.
  + **Microsoft Office** also have macros written in **Visual Basic for Applications (VBA)** which can be leveraged by attackers but is not very common.
    - Macros are disabled by default, however.
  + **Linux systems have tools like Python, Perl, and Bash** which can be leveraged to create remote accesses 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.
  + While it may be a simple solution to just disable these services to prevent them from being leveraged by attackers, they are an important part of how users interact with their system so cannot just be fully disabled.
    - This is why preventing attackers from gaining access to the system in the first place is always the most important layer of defense.
  + Bash has a built-in security mode called the **restricted shell** that limits what users can do, including things like specifying command names containing slashes, importing function definitions from the shell environment, and others.
  + Fortunately, as well, there exists tools to search for rootkits like **chkrootkit** and **rkhunter**. Behavior-based security tools can also monitor system logs and network traffic to help defenders identify compromised systems.
* **Adversarial Artificial Intelligence:**
  + A developing field where AI is used by attackers for malicious purposes. Adversarial AI attacks currently tend to deal with **data poisoning**, providing security and analytic AI and machine learning algorithms with adversarial input that serves the attacker’s purposes, or attack against privacy. So they pretty much just fuck up the training data and even learning algorithms.
  + As artificial intelligence and machine learning continue to become more prominent in security toolsets, like network analytics, but it is important that the training data is as intended. Because if you want to train an AI using “normal” network traffic as a baseline, but that network traffic is actually compromised, then the AI is not going to perform how you like it.
    - **Tainted training data** for machine learning algorithms will be a target for attackers, and the security of machine learning algorithms themselves will be increasingly important.
  + Here are some ways to reduce problems of potential data poisoning:
    - Understand the quality and security of source data.
    - Work with AI and ML developers to ensure that they are working in secure environments and that data sources, systems, and tools are maintained in a secure manner.
    - Ensure that changes to AI and ML algorithms are reviewed, tested, and documented.
    - Encourage reviews to prevent intentional or unintentional bias in algorithms.
    - Engage domain experts whenever possible.