. Cyber Kill Chain (by Lockheed Martin)

It's a model that shows the **steps an attacker takes** to **breach a network or system.**

kill chain is a **military concept** related to the **structure of an attack.** It consists of target **identification, decision** and **order to attack the targe**t, and finally the target **destruction**.

The kill chain is a military idea about how an **attack is planned and carried** out: find a **target, decide to attack, then destroy it**.

Lockheed Martin, a **global security and aerospace company,** that established the **Cyber Kill Chain® framework for the cybersecurity industry in 2011**

The framework defines the steps used by **adversaries or malicious actors in cyberspace.**

To succeed, an adversary needs to go through all phases of the Kill Chain

You can use the Cyber Kill Chain to **assess your network and system security** by **identifying missing security controls** and closing certain security gaps based on your company's infrastructure.

Why learn it?
It helps **defenders (SOC analysts, threat hunters, IR teams)** spot **where attacks are happening, find missing defenses, and stop threats like ransomware or APTs earlier.**

Phases (quick):

1. **Reconnaissance** — attacker **gathers info** (OSINT, scans).
2. **Weaponization** — attacker **builds the exploit/malware.**
3. **Delivery —** mail, link, USB, supply-chain **to reach target.**
4. **Exploitation** — **delivered code runs** (vulnerability, macro).

5. **Installation** — malware **persists (services, autoruns).**
6. **Command & Control** — compromised host t**alks back to attacker.**
7. **Actions on Objectives** — **data theft, ransomware, destruction, lateral moves.**

Reconnaissance:

research & planning phase where the **attacker collects info about a target** to plan the attack.

Common intel gathered:

infrastructure, employee names/emails, tech stack, public documents, WHOIS, leaked data.

Two types

- **Passive recon:**

  no direct contact (safe for attacker) — e.g., Google, LinkedIn, WHOIS, breach DBs.

- **Active recon: direct probing** that can be detected — e.g., port scans, banner grabbing, social engineering.

OSINT (Open-Source Intelligence)

- **Sources:** search engines, social media, forums/blogs, news, public records, WHOIS, GitHub, leak sites**.**
- Use: build profiles (employees, services, versions), find weak points and phishing targets.

Email harvesting (short)

What:
Collecting **email addresses to prepare phishing** or spear-phishing attacks.

How:

- **Scraping** company websites
- **LinkedIn** and social media
- **Public lists** and directories
- WHOIS records
- **Breach databases**

Why:

- Helps craft **believable phishing emails**
- Enables targeted attacks on specific employees (finance, HR, developers)

Connection to OSINT:
Attackers use **OSINT to map the organization, find employees, and identify who to target.**

Process (simple):

1. Gather **company information** using OSINT.
2. Harvest **employee emails from public sources**.
3. Create a **convincing phishing/spear-phishing email.**
4. Send payload → leads to next steps in the kill chain.

Key Idea:
The **more information** the **attacker collects**, **the higher the success rate.** The less they know, the more likely the attack will fail.

Weaponization

Hacker would work on turning the **raw information into actionable attack tools** through crafting **malware and exploits into a payload**. usually use automated tools to generate the malware or refer to the **DarkWeb** **to purchase the malware**. **sophisticated actors or nation-sponsored APT (Advanced Persistent Threat Groups)** would write their **custom malware to make the malware sample unique** and evade detection on the target.

Term	Role	Example
Exploit	**Opens the door**	Uses a **software bug** to get in
Payload	**Does the damage**	Installs a keylogger
Malware	**The whole harmful software**	The full infected program

---

🧠 Memory trick:

Exploit → Entry
Payload → Attack
Malware → Package containing both

Malware

- A malicious **program itself.**
- Example: Virus, Trojan, Worm, Ransomware.
  👉 It is the actual harmful software that attacks your system.

💣 Exploit

- A **technique or code that uses a weakness (vulnerability)** in a system or app.
- It helps the attacker break in.
  👉 Think of it as the key that opens a locked door.

⚙️ Payload

- The **malicious action that runs after the exploit works.**
- It's the "damage" part — what the attacker wants to do (e.g. install backdoor, steal data).
  👉 It is delivered through the exploit.

Delivery:

Delivery is how the attacker (Megatron) **gets malware onto the target's machines** decides to choose the **method for transmitting the payload or the malware onto the target environment.**

Phishing email: a malicious actor could **craft a malicious email** that would target either **a specific person (spear phishing attack) or multiple people in the company.**

An attacker might decide to conduct a **sophisticated USB Drop Attack** by printing the company's logo on the USB drives and **mailing them to the company while pretending to be a customer sending the USB devices as a gift.**

Watering-hole attack — attacker **hacks a website** that a **target group often visits**. When **victims visit that site, they're redirected**

Exploitation:

Exploitation = the moment **the attacker's code actually runs on the victim's computer** by taking advantage of **a weakness.**

Common ways attackers exploit systems

- **Malicious macros**
  Example: an email attachment with a Word file — when opened, a hidden macro runs and installs ransomware.

- **Zero-day exploit**
  A bug nobody knows about yet — no patch exists, so detection is hard.

- **Known CVEs (unpatched vulnerabilities)**
  The attacker uses publicly known flaws that the system owner hasn't patched.

Escalate privileges (make a **normal user into an admin**)
Move laterally (jump to **other machines on the network)**

Installation (Persistence Phase)

After exploiting a system, attackers install **persistent backdoors** so they can return anytime. Persistence methods include:

1) Web Shell

What: A small **malicious script (PHP, ASPX, JSP)** uploaded **to a website/web server**
Why dangerous: Looks like **a normal file; attacker can run commands or** upload/download files.
Detect/Defend: Monitor for **new/modified web files, validate uploads, scan for suspicious code.**

2) Backdoor Payloads (e.g., Meterpreter)

What: Malware that gives the attacker a **remote interactive shell.**
Why dangerous: Provides ongoing **remote control; can auto-start.**
Detect/Defend: EDR alerts, check unknown network connections, block unsigned binaries.

3) Malicious Windows Services

What: Creating or modifying services to **auto-run malware at boot.**
Why dangerous: Runs with high privileges and can be disguised.
Detect/Defend: Audit service creation/changes, verify executable paths, restrict admin rights.

4) Registry Run Keys / Startup Folder

What: Entries that **auto-launch programs when users log in.**
Why dangerous: Very simple, **survives reboot, common for persistence.**
Detect/Defend: Monitor Run keys/Startup folders, use integrity monitoring, block unknown executables.

5) Timestomping

What: Changing file timestamps to look **old/legitimate.**
Purpose: Hide malware, confuse forensic timelines.

Command & Control:

After gaining persistence, the malware on the **victim's machine** opens a **C2 (Command & Control) channel.** This allows the attacker to remotely control the infected device. The infected host **repeatedly contacts the attacker's server** — this repeated communication is known as **C2 beaconing.**

How it works

- The compromised **endpoint connects to an external attacker-controlled server.**
- Once connected, the **attacker can issue commands, upload/download files, move laterally, or execute further actions.**
- Older attacks used **IRC (Internet Relay Chat). IRC is a *protocol* (and a chat service), not a port or a software.,** but modern security tools detect IRC easily, so attackers now use stealthier channels.

Common C2 Channels

- **HTTP / HTTPS (Ports 80 & 443)**
  Blends with normal web traffic, making detection harder. Common for modern malware.
- DNS Tunneling
  DNS Tunneling
- Malware hides **its communication inside normal-looking DNS requests.**
- These DNS queries go to a DNS server controlled by the attacker.
- This helps bypass firewalls because DNS traffic is usually allowed everywhere.

Note

The **C2 server may be controlled by:**

- **the attacker, or**

- another **compromised system** acting as **part of the attacker's infrastructure.**

Delete backups and shadow copies so the victim **cannot restore their system (Shadow Copy = Windows automatic backup snapshots).**
Overwrite or corrupt data **to cause damage or make recovery impossible.**