OWASP Top 10 Security Vulnerabilities

(2017 and 2021)

The Open Web Application Security Project is an online community that produces freely-available articles,
methodologies, documentation, tools, and technologies in the field of web application security

OWASP Top 10 Security Vulnerabilities – 2017 and 2021

2017 Vulnerabilities

- **Injection**
- **Broken Authentication**
- **Sensitive Data Exposure**
- **XML External Entities**
- **Broken Access Control**
- **Security Misconfiguration**
- **Cross-Site Scripting**
- **Insecure Deserialization**
- **Using Components with Known Vulnerabilities**
- **Insufficient Logging and Monitoring**

2021 Vulnerabilities

- **Broken Access Control**
- **Cryptographic Failures**
- **Injection**
- **Insecure Design**
- **Security Misconfiguration**
- **Vulnerable and Outdated Components**
- **Identification and Authentication Failures**
- **Software and Data Integrity Failures**
- **Security Logging and Monitoring Failures**
- **Server-Side Request Forgery**

OWASP Top 10 Security Vulnerabilities 2021 (1)

1. Broken Access Control. Restrictions on what authenticated users are allowed to do are often not properly enforced. Attackers can exploit these flaws to access unauthorized functionality and/or data, such as access other users' accounts, view sensitive files, modify other users' data, change access rights, etc.
2. Cryptographic Failures It is known as Sensitive Data Exposure. Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and PII. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data may be compromised without extra protection, such as encryption at rest or in transit, and requires special precautions when exchanged with the browser.
3. Injection. Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.
4. Insecure Design: It is a new category for 2021, with a focus on risks related to design flaws. If we genuinely want to "move left" as an industry, it calls for more use of threat modeling, secure design patterns and principles, and reference architectures.

Ref: https://owasp.org/www-project-top-ten/

OWASP Top 10 Security Vulnerabilities 2021 (2)

5. Security Misconfiguration. Security misconfiguration is the most commonly seen issue. This is commonly a result of insecure default configurations, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages containing sensitive information. Not only must all operating systems, frameworks, libraries, and applications be securely configured, but they must be patched/upgraded in a timely fashion.
6. Vulnerable and Outdated Components: Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts.
7. Identification and Authentication Failures: It was previously Broken Authentication. Incorrectly implemented authentication and session management calls can be a huge security risk. If attackers notice these vulnerabilities, they may be able to easily assume legitimate users' identities. Multifactor authentication is one way to mitigate broken authentication. Implement DAST and SCA scans to detect and remove issues with implementation errors before code is deployed.

Ref: https://owasp.org/www-project-top-ten/

OWASP Top 10 Security Vulnerabilities 2021 (3)

8. Software and Data Integrity Failures: It is a new category for 2021, focusing on making assumptions related to software updates, critical data, and CI/CD pipelines without verifying integrity. One of the highest weighted impacts from Common Vulnerability and Exposures/Common Vulnerability Scoring System (CVE/CVSS) data mapped to the 10 CWEs in this category. Insecure Deserialization from 2017 is now a part of this larger category. Insecure deserialization often leads to remote code execution. Even if deserialization flaws do not result in remote code execution, they can be used to perform attacks, including replay attacks, injection attacks, and privilege escalation attacks.
9. Insufficient Logging & Monitoring. Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data. Most breach studies show time to detect a breach is over 200 days, typically detected by external parties rather than internal processes or monitoring.
10. Server-Side Request Forgery It is added from the Top 10 community survey. The data shows a relatively low incidence rate with above average testing coverage, along with above-average ratings for Exploit and Impact potential. A Server-Side Request Forgery (SSRF) attack involves an attacker abusing server functionality to access or modify resources. The attacker targets an application that supports data imports from URLs or allows them to read data from URLs.

Ref: https://owasp.org/www-project-top-ten/

Vulnerabilities to be studied in this course (1)

- Denial-of-Service (DoS) is an attack is an attack meant to shut down a machine or network, making it inaccessible to its intended users. DoS attacks accomplish this by flooding the target with traffic, or sending it information that triggers a crash.
- Cross-Site Scripting (CSS): With cross-site scripting, attackers take advantage of APIs and DOM manipulation to retrieve data from or send commands to your application. Cross-site scripting widens the attack surface for threat actors, enabling them to hijack user accounts, access browser histories, spread Trojans and worms, control browsers remotely, and more.
- Cross-Site Request Forgery (CSRF) is an attack that forces an end user to execute unwanted actions on a web application in which they're currently authenticated. With a little help of social engineering (such as sending a link via email or chat), an attacker may trick the users of a web application into executing actions of the attacker's choosing. If the victim is a normal user, a successful CSRF attack can force the user to perform state changing requests like transferring funds, changing their email address, and so forth. If the victim is an administrative account, CSRF can compromise the entire web application.

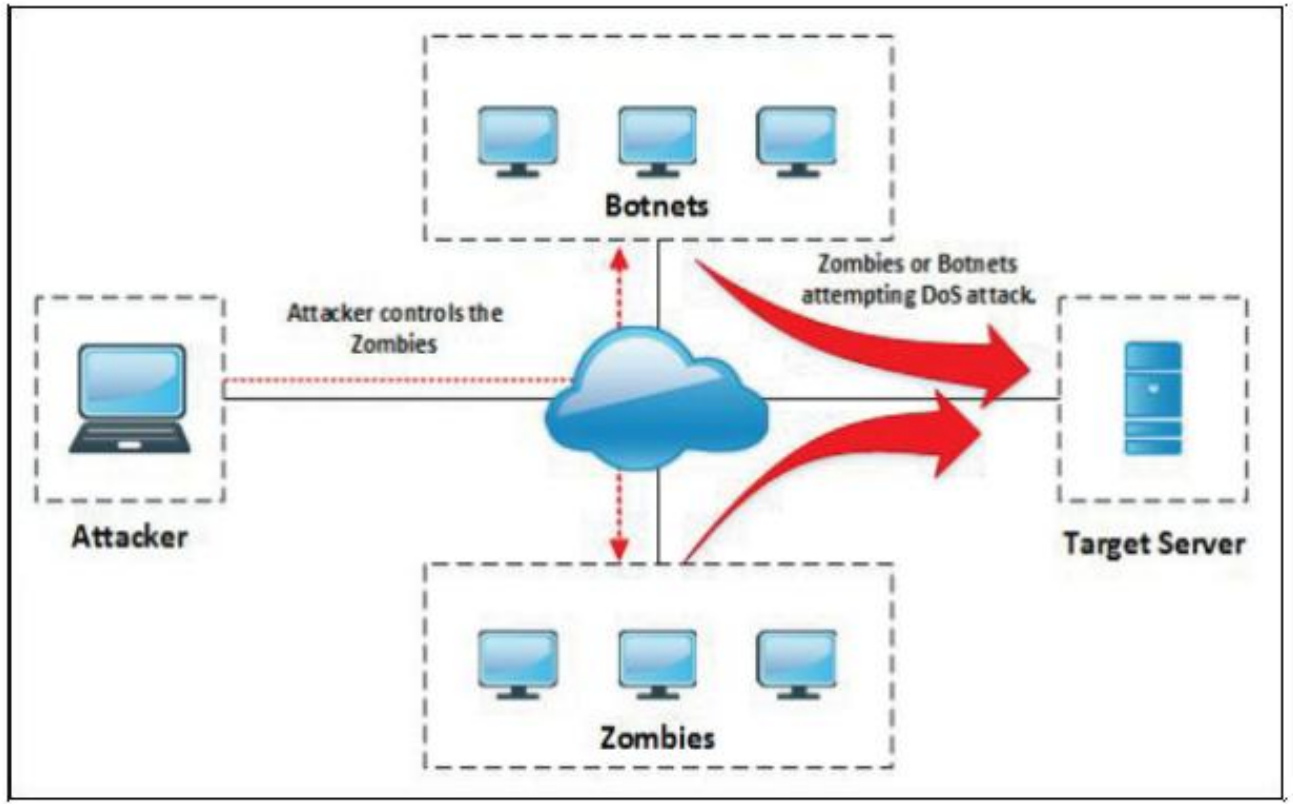
Vulnerabilities to be studied in this course (2)

- XML External Entities (XEE): This risk occurs when attackers are able to upload or include hostile XML content due to insecure code, integrations, or dependencies. An SCA scan can find risks in third-party components with known vulnerabilities and will warn you about them. Disabling XML external entity processing also reduces the likelihood of an XML entity attack.
- Injection: Injection occurs when an attacker exploits insecure code to insert (or inject) their own code into a program. Because the program is unable to determine code inserted in this way from its own code, attackers are able to use injection attacks to access secure areas and confidential information as though they are trusted users. Examples of injection include SQL injections, command injections, CRLF injections, and LDAP injections. Application security testing can reveal injection flaws and suggest remediation techniques such as stripping special characters from user input or writing parameterized SQL queries.

Denial of Service

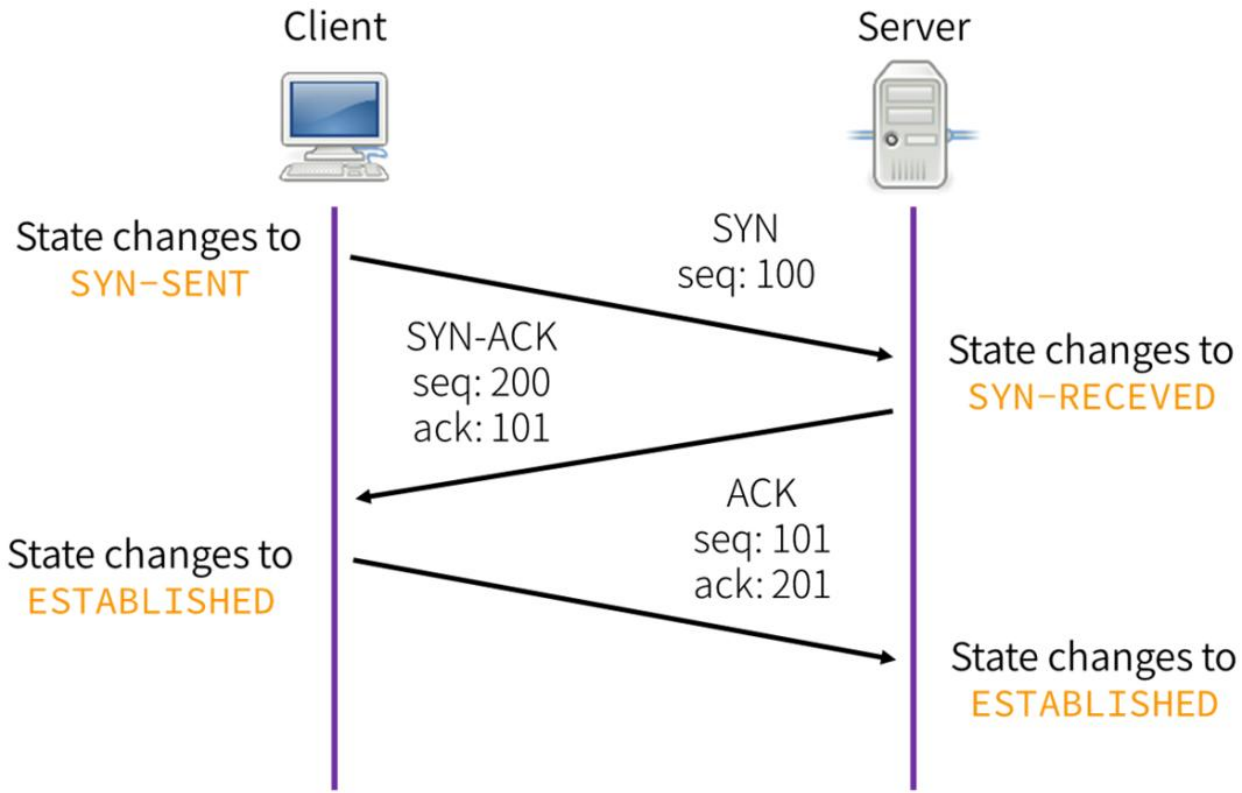
Denial of Service (DoS)

- DoS is the way of making a server/network/system to deny services.
- Symptoms of DoS
  - Slow performance
  - Increase in spam emails
  - Unavailability of a resource
  - Loss of access to a website
  - Disconnection of a wireless / wired internet connection
  - Denial of access to internet services.
- Distributed DoS is the way of making a server to deny services via multiple compromised systems (or Bonnets).



Distributed DoS

- Recall TCP 3-way Handshake



DoS Targets

- Webserver – Attacks performed targeting the web servers. Mainly to increase down time.
- Back-end resources – Attacks performed targeting database server.
- Network or computer specific – Attacks performed targeting network devices in a LAN.

DDoS and DoS Attack Categories

- Volumetric attacks: These attacks focuses on consuming the bandwidth. To do this, attacker sends high volume of traffic to the server. (Repeated TCP connections, SYN Attack/Flood, ICMP Flood, Ping of death,)
- Fragmentation attacks: In this, an IP datagram is divided in to small fragments and sent through routers. To reassemble these fragments router requires more resource. Teardrop attack is a good example of it. Fragmentation attacks are
  - TCP Fragmentation attack
  - UDP and ICMP fragmentation attack (Fraggle – via UDP Echos and Sumrf – via ICMP Echos attacks)
- TCP State Exhaustion attack: This attack focuses on damaging the state information maintained by servers there by server forgets the concurrent connections.
- Application Layer Attacks: In this, application layer programs are targeted to degrade the their service ability.

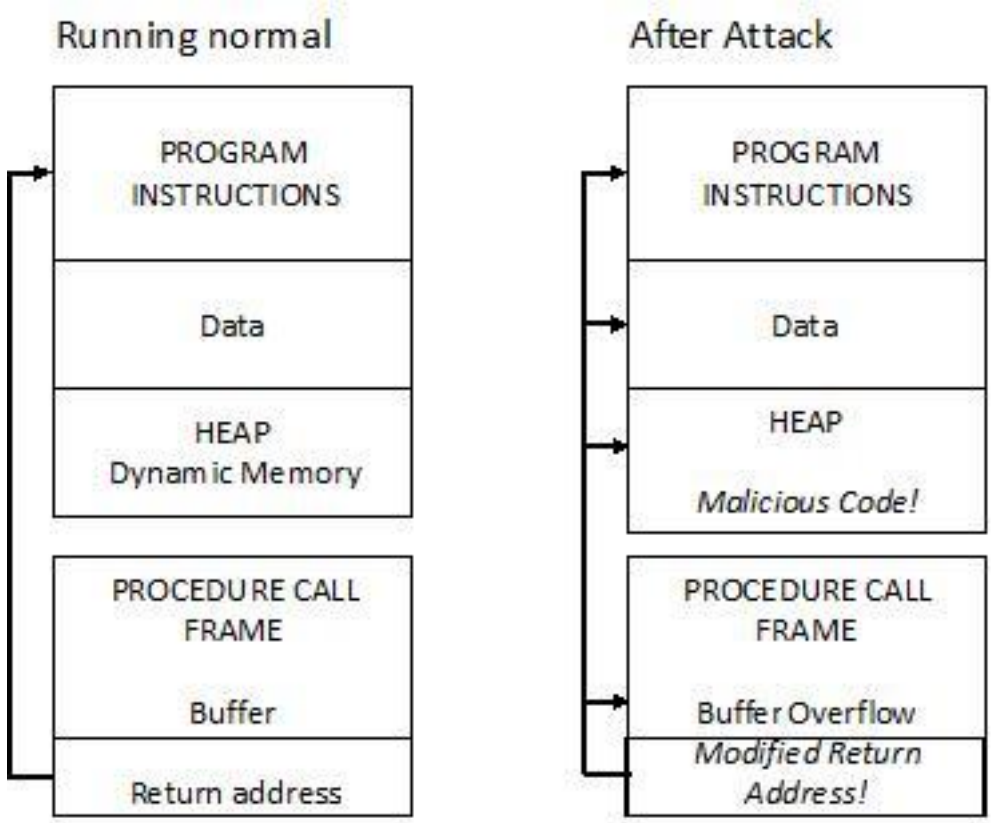
DoS / DDoS Attack Techniques

- Service Request Floods : Requesting for service to a web application and not acknowledging the response.
- SYN flooding attack: Also called as flooding attack. Attacker attempts to send SYN request with bogus IP addresses.
- ICMP Flooding attack: ICMP is a helper protocol to know the status of the route, errors, and device operations from routing devices. Hence flooding ICMP requests will not give enough time to respond by the devices.
- Peer-to-Peer Attack: Attacker aims to target Direct Connection (DC++) protocol in peer-to-peer environment. DC++ lists the peer clients in the network

DoS / DDoS Attack Techniques

- Land attack: Traffic will be sent to target machine with the source spoofed as the target machine itself. (looking it as self reply)
- Permanent DoS (PDoS): Attacker aims to target the hardware by Phalshing or Bricking the system. These two causes unrecoverable damage to the system hardware.
  - Plashing is the way of pushing bogus or incorrect updates to a victims' systems firmware.
  - This attack will make hardware unusable which is also called **bricked**.
- Application Level Flooding attacks: Attacker aims to target the vulnerabilities in the application servers.
  - Three categories are 1) Flood, 2) Disrupt – no user allowed to login, 3) Jamming – SQL Injection.
- Distributed Reflection DoS: Attacker uses intermediately victims (including primary, secondary, and all other victims in the chain) to launch DoS attacks. The attack aims to flood the traffic between the victims.

DoS with Buffer Overflow Attack



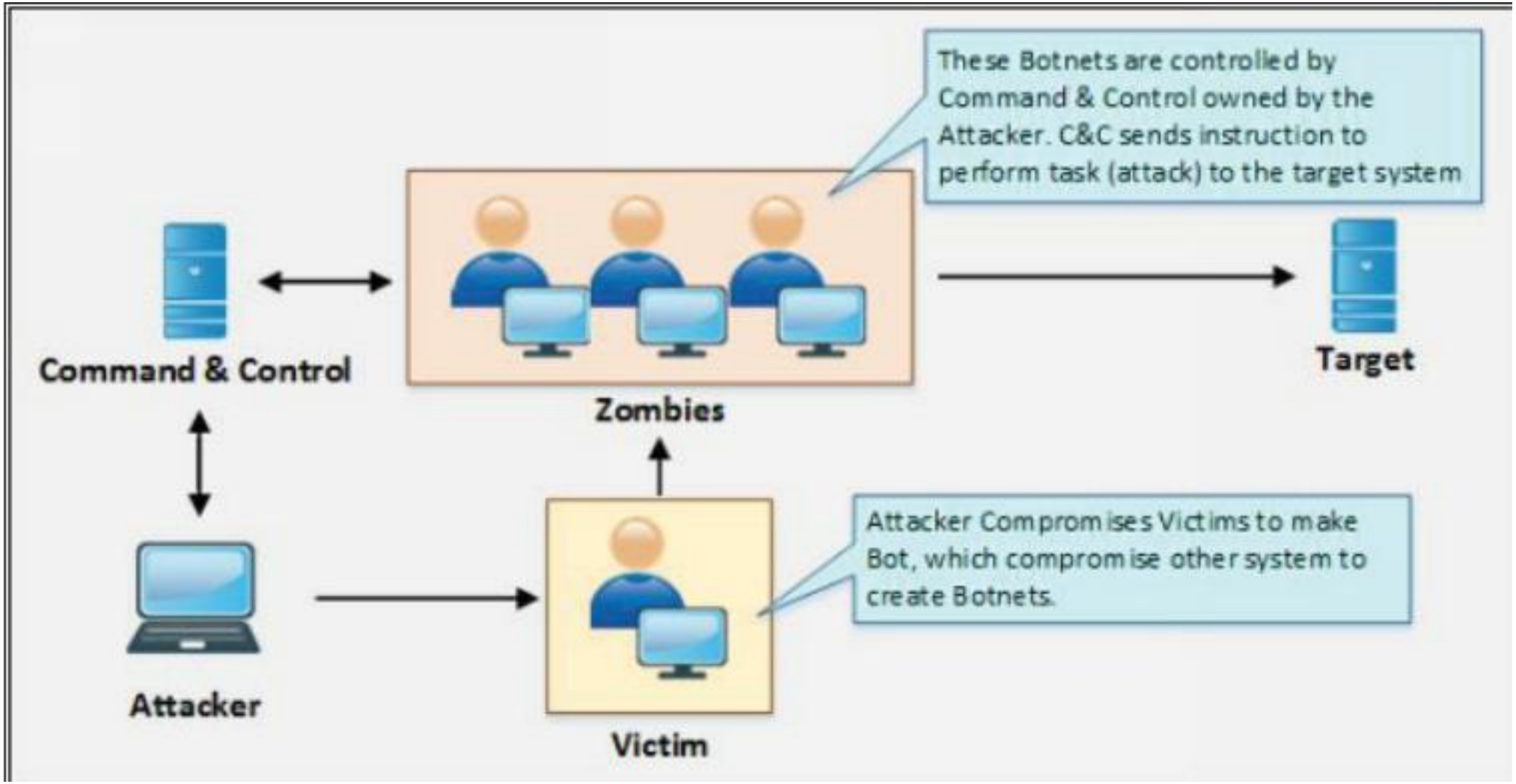
Attacker plants code that overflows buffer and corrupts the return address. Instead of returning to the appropriate calling procedure, the modified return address returns control to malicious code, located elsewhere in process memory.

Absence of Bound checking is the root cause of buffer overflow attack.
Ex: functions such as gets(), scanf(), strcpy(), strcat() in C programming.

Self study: Search the topic "smashing the stack for fun and profit" via Google search

Reference: http://cis1.towson.edu/~cssecinj/modules/cs2/buffer-overflow-cs2-c/

Distributed Denial of Service with Botnets



Scanning vulnerabilities by Botnets

- Random scanning technique: Bots randomly scan the machines in an IP range and will identify vulnerabilities.
- Hit-List scanning techniques: Attacker may feed the list of vulnerable system IP addresses.
- Topological scanning technique: Knowing another vulnerable system from a compromised system. It will be done using malicious URLs
- Subnet scanning technique: It will be done by a compromised system with in LAN (behind the firewall).
- Permutation scanning technique: In this scanning takes place via sharing of pseudorandom permutation of IP addresses.

DoS using Metasploit

- **SYN Flooding using Metasploit (In Kali Linux, open the Metasploit in root mode) (Let IP address of Windows 7 is 192.168.1.4)**

1. **Scan for TCP ports using nmap (use extensive scan to know)**
   nmap -T4 -A -v 192.168.1.4
2. **List the tcp ports that are open**
3. **Choose a port of your choice**
4. **Open metasploit on Kali Linux (msfconsole)**
   use auxiliary/dos/tcp/synflood
5. **to set options use the command - show options**
6. **set the parameters**
   set RHOST 192.168.1.4
   set RPORT 21
   set SHOST localhost (Kali Linux IP)
   set TIMEOUT 20000
7. **once the parameters are set then use the command**
   exploit
8. **Now, move to windows 7 system and check the traffic via windump and open the task manager and check for the CPU usage.**

DoS Defensive Strategies

- Disabling Unnecessary Services: it can be done via task manager or listing services that are not used anytime.
- Using Anti-malware: Always enable the real-time protection provided by the operating systems. Also update them regularly.
- Enabling Router Throttling: This will provide a time buffer for network administrators to respond if traffic saturation is observed.
- Using a Reverse Proxy: It is opposite of forward proxy. In this traffic is diverted to a middle server instead of original server. Middle server will take care of heavy traffic.
- Enabling Ingress and Egress Filtering: Ingress filtering protect from DoS via incoming traffic, where as egress filtering protects from DoS via outgoing traffic.
- Degrading Services: Enabling automatic throttling or shut down of services in the event of attack.
- Absorbing the Attacks: Adding extra power or services or bandwidth or load balancing methods.

Summary

- The aim of both DoS and DDoS is to disturb the availability of services.
- DoS can focus on flooding bogus traffic.
- DDoS can focus on launching DoS via Botnet.
- DoS attack can be performed at various levels of protocol stack.
- Permanent DoS is targeted to fail the device hardware.
- Buffer overflow based DoS may occur due to flaw in the application program.
  - EIP is the point of execution in the stack, it will be shifted when over flow occurs.
  - Be cautious while using boundless C functions.
  - NOP – No Operation (0x90) instruction equates to a full CPU cycle with no actual work being accomplished.

References

1) Sean-Philip Oriyano, "Certified Ethical Hacker Version 9 - Study Guide", EXAM 312-50, Sybex Wiely,2016.
2) Georgia Weidman, "Penetration testing A Hands-On Introduction to Hacking", No Scratch Press, 2014.
3) Raphaël Hertzog, Jim O'Gorman, and Mati AharoniKali, "Linux Revealed Mastering the Penetration Testing Distribution", OFFSEC Press, 2017
4) Corey P. Schultz, Bob Percianccante, "Kali Linux Cook Book", Second edition, Packet Publishing, 2017.
5) Lee Allen, Tedi Heriyanto, Shakeel Ali, "Kali Linux – Assuring Security by Penetration Testing, Packet Publishing, 2014.
6) James Corley, Kent Backman, and Michael T. Simpson, "Hands-On Ethical Hacking and Network Defense", 2006.
7) Willie L. Pritchett, David De Smet, "Kali Linux Cook book", Packet publishing, 2013.
8) Georgia Weidman, Penetration Testing - A Hands Introduction to hacking, No Starch press, 2014.
9) Jessey Bullock, Jeff T. Parker, Weireshark for security professionals using Wireshark and Metasploit Framework, Wiely, 2015.
10) Dejey, Murugan, "Cyber Forensics", Oxoford University Press, 2018.
11) Online material from https://www.ethicalhackx.com
12) https://www.youtube.com/watch?v=2sb8_VTd-D0
13) https://www.youtube.com/watch?v=E5U2whB37I4
14) https://www.youtube.com/watch?v=YaYoY0fXhuo