

Why Information security is Important



Week 2 – Session 1



Objectives

Following this session, you should be able to:

- **Describe** and **Define** some common **Information Security Terminology**.

Malware

- What is Malware? Name some types of Malware.
 - A term used to describe malicious software.
 - Virus , Worm ,Trojan ,Ransomware
- How can an attacker get Malware on a device?
 - Email, Internet Download, Paying someone to install it for them, Remote access to a device
- How do we defend against malware?
 - Anti-Virus
 - Firewall/Internet Proxy
 - User education and awareness
 - Secure configuration of devices



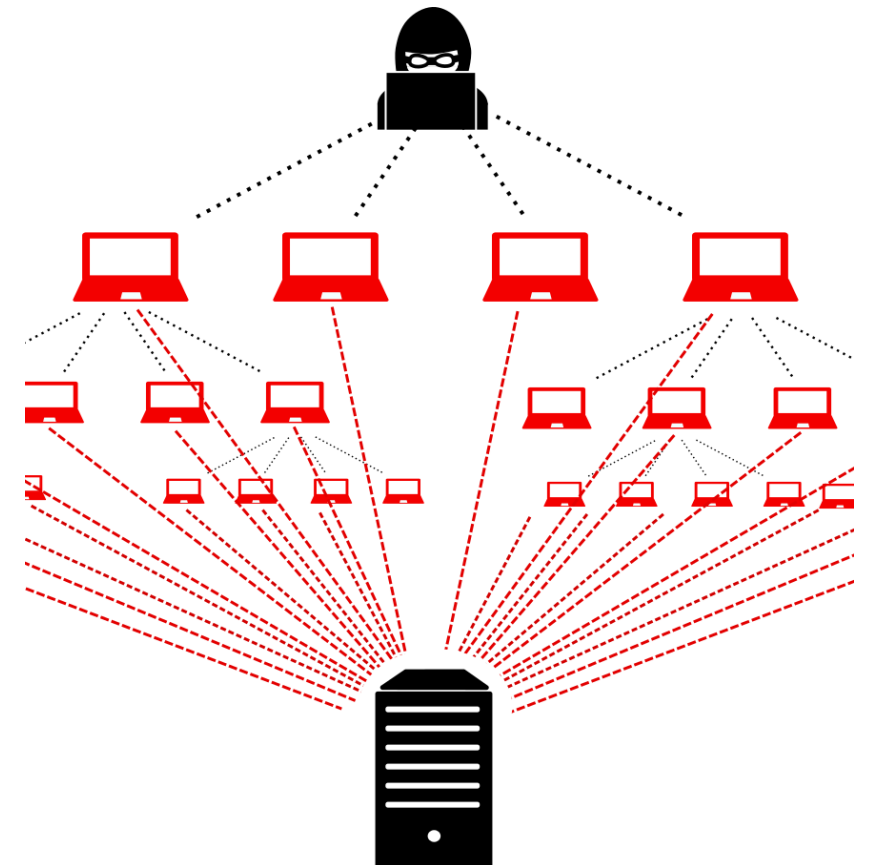
Phishing

- A technique used to **trick people** into giving away access to sensitive information, or obtaining money.
- Attackers “phish” for information by **delivering emails, pop-up adverts, text messages, phone calls and social media** to their victims.
- These will contain **misleading information**, or messaging designed to scare the reader into thinking they have to take action and will then lead the reader into clicking a fake link, downloading a malicious payload or entering sensitive information such as their username and password.
- **Defend against Phishing**
 - Security Awareness Training
 - Web Filtering/Proxy
 - Email Filtering
 - Valid HTTPS Certificates
 - DMARC on Emails



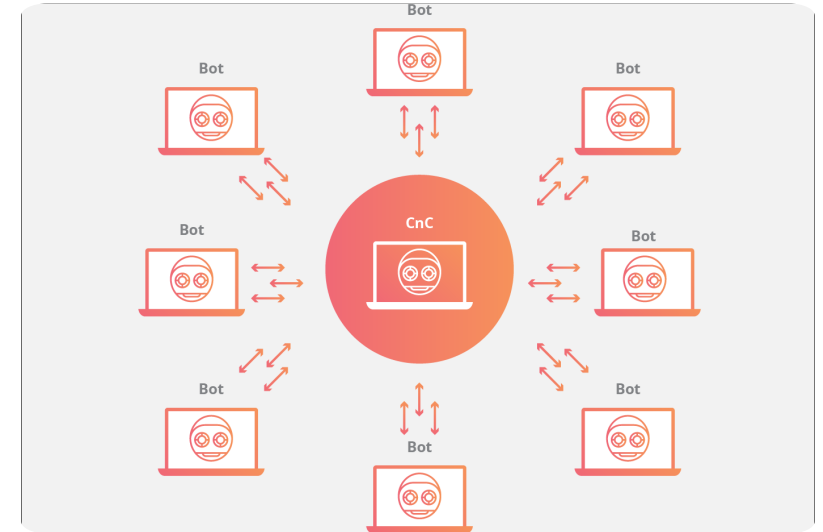
DDoS Attack

- What is a DDoS attack?
 - **Large data packets sent** over the internet to saturate the bandwidth available to the interface.
- How can an attacker conduct a DDoS?
 - Commonly delivered via a **Botnet** (see next slide).
 - Some attackers offer a **payed for service**, where you can pay for them to direct a DDoS to an endpoint of your choice.
 - **Vulnerabilities and weaknesses** in the endpoint could enable an attacker to increase their attack power.
 - **Lack of disaster recovery** could also allow an attacker to effectively deploy a DDoS attack
- How do we defend against a DDoS?
 - **Patch and upgrade systems**
 - **Configure Routers and Firewalls properly** to deal with traffic accordingly.
 - **Over-Provisioning of the required bandwidth** to “soak” up additional peaks in traffic.
 - **Correct server, load balancing and web interface configuration**
 - **Correct disaster recovery practices.**



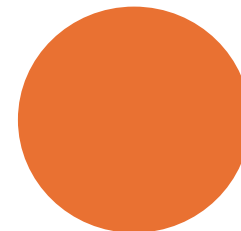
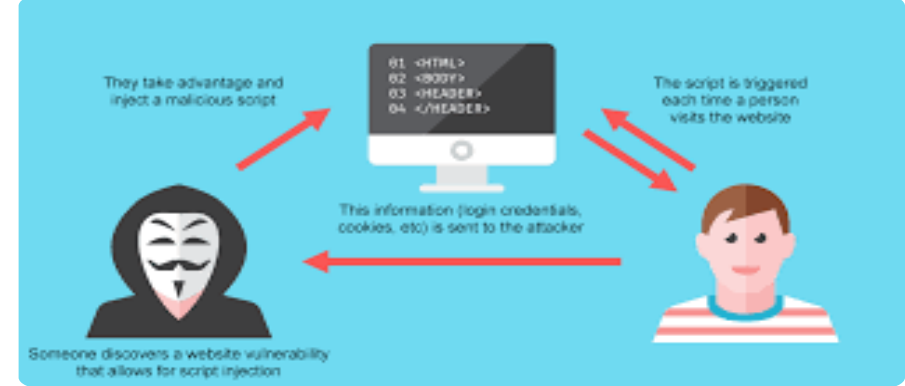
Botnets

- What is a Botnet?
 - A network of compromised devices which listen and report to an attackers command and control server.
 - They do as they're told and **run attacks and jobs on behalf of the attacker.**
- How can an attacker accumulate a Botnet?
 - Deployment of Malware via Phishing, USB drops, Malicious Web Downloads, etc..
 - Purchasing one from the dark-web.
- How do we defend against becoming part of a Botnet?
 - **Security patching of devices.**
 - **Correct device configuration.** Most botnet malware can be prevented by enabling windows firewalls.
 - **Anti-Virus and Endpoint detect and response** software
 - **Firewall, proxy and router config** to prevent outbound and inbound traffic from rouge IP addresses.
 - **Security awareness training.** Ensuring that users know how to spot a phishing email, rouge website or dodgy link.



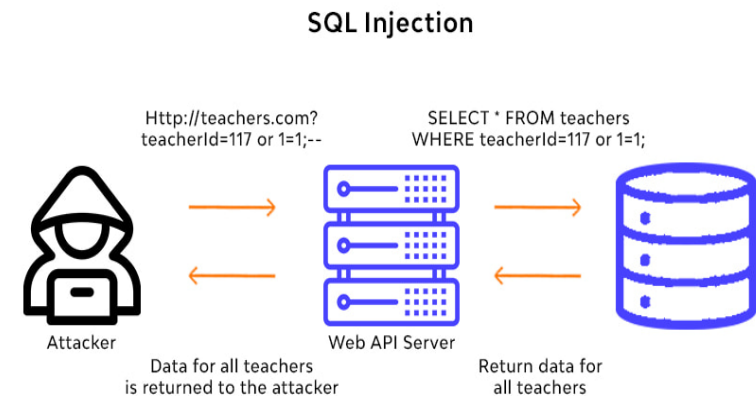
Cross-Site Scripting (XSS)

- What is Cross-Site Scripting?
- A web application **injection attack**, where malicious code is injected into the website to either re-direct users to malicious websites, send malicious code to other users, or collect users data.
- How can an attacker conduct a cross-site scripting attack?
- **Lack of security updates** in web applications and web servers.
- **Poor web development allows** attackers to trick a web application into accepting their code.
- **Uses Web Scripting languages** (PHP, Java etc..) to manipulate existing functionality.
- How do we defend against cross-site scripting attacks?
- Input validation. Make sure that the input fields on the website only allow specific data. Open Libraries exist to scan the input for common XSS techniques.
- Web Server patching and upgrades.
- End-user browser **upgrades**. Sometimes browsers have vulnerabilities which allow data to be leaked between tabs, cookies and sessions.



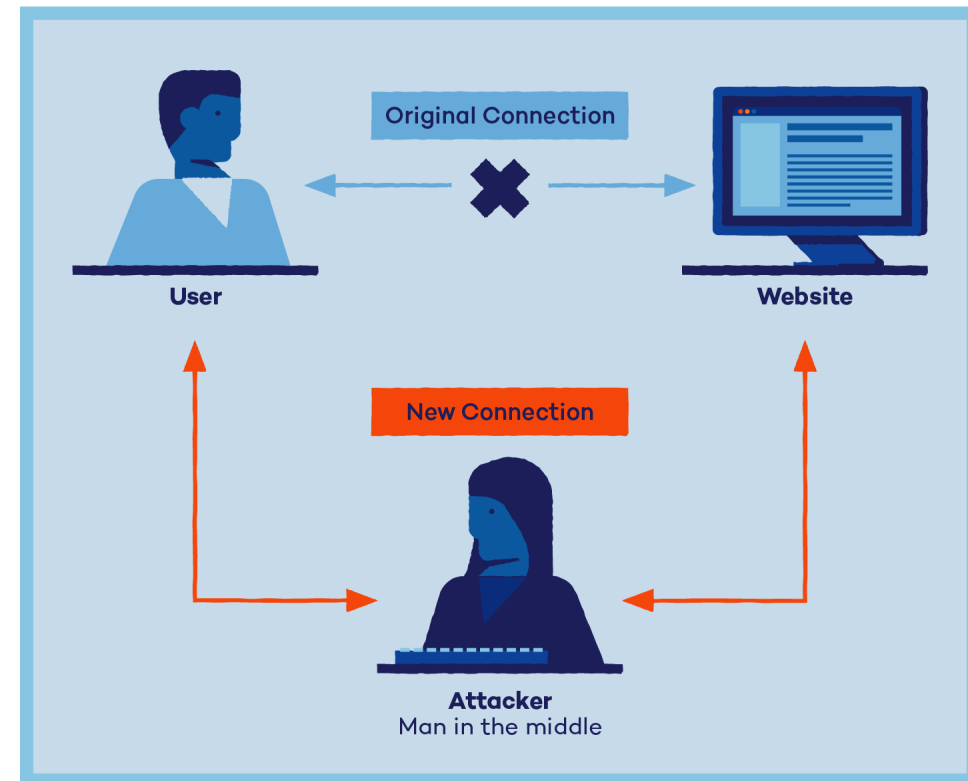
SQL Injection

- What is SQL Injection?
- Another type of application injection attack, where malicious code is injected into the application in order to try and access the database backend.
- How can an attacker conduct an SQL Injection attack?
- If a **vulnerability** is present in the application or database, an attacker can use this to trick the application into treating their input as if it were a legitimate request.
- **Poorly developed applications** can allow an attacker to convince the application to treat their code as a legitimate request.
- How do we defend against SQL Injection attacks?
- Anti-Virus and Intrusion detection and prevention systems. As well as Web Application Firewalls.
- Security awareness and code scanning for the developers.



Session Hijacking and Man-in-the-Middle Attacks

- Exploitation of a valid “session” between two devices, by using a **stolen or crafted cookie** to trick a server into handing over data from a valid user to an attacker
- A man in the middle attack is similar, where an attack tricks a server into sending information to the attacker, before they then pass it on to the user.

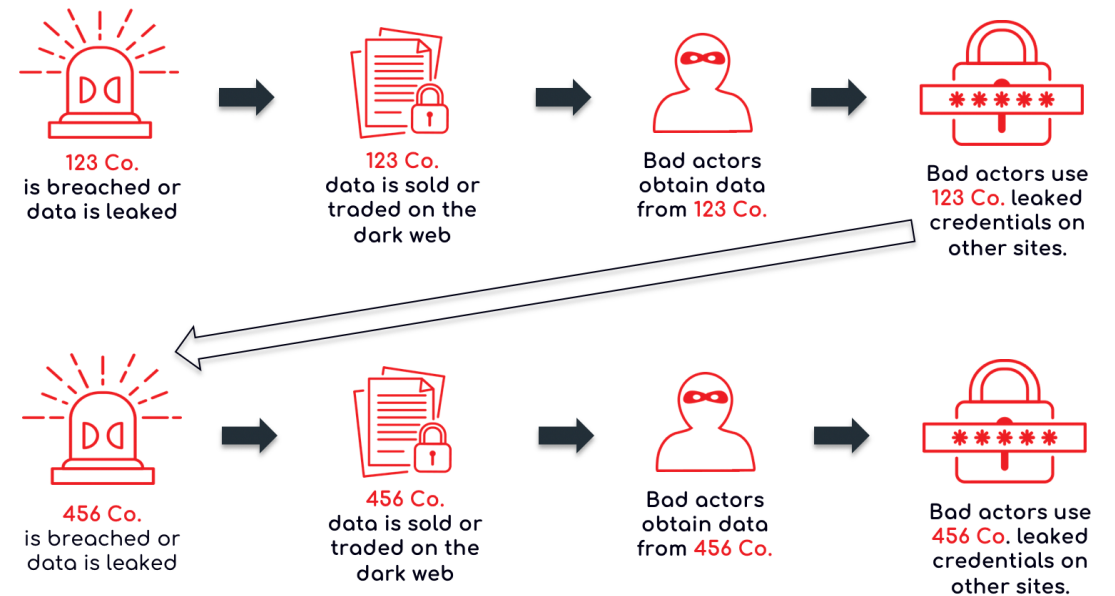


Credential Reuse

- What is Credential Reuse?
- Automated injection of known compromised user credentials.
- Attackers can leverage accounts reusing credentials to gain access to multiple systems.
- **How do we defend against a credential reuse attack?**
- Circulate passwords on a regular basis.
- Don't use the **same passwords** for different/multiple accounts.
- Use smart network monitoring to detect failed login attempts for accounts and from the same endpoint/IP address.

Credential Stuffing Domino Effect

Assumes most people reuse the same credentials on multiple sites. the attacker relies on breached credentials.



The process repeats over and over again with new organizations.

The Basics of Vulnerabilities



Vulnerability Exploitation

- **Exploitation Tools:**

- An attacker will use tools similar to those of a pen tester. The most common tool is Kali Linux.

- **Deploying Payloads:**

- An attacker will use a vulnerability to deploy a malicious payload. This could be an application designed to be clicked by the end user. Or a line of code designed to manipulate the application.

- **Elevating Privileges:**

- Once in the attacker will want to try and gain the best privileges they can. Either Super-Admin or System if possible.

- **Remote access and backdoors:**

- Finally, the attacker will want to ensure that they can continue accessing the compromised device remotely. They will also want to implement a backdoor so that if the vulnerability is patched, they can continue to access the device regardless. Some attackers may even patch the vulnerability to prevent them from being discovered, or from other attackers gaining access.

Vulnerability Detection

- **Scanning Tools:**
 - Tools like Nessus and Rapid 7 also allow the good guys to **detect vulnerabilities** on their systems.
- **Vulnerability Scores:**
 - Prioritising vulnerabilities by the severity score, allows you to **prioritise which systems need** patching first.
- **Likelihood and Impact:**
 - Based on the **findings and scores**, you may wish to produce a likelihood and impact rating to understand how much of a threat the vulnerability is. Some tools will do this for you as part of the CVSS score.
- **Open Source Intelligence:**
 - Finally, open source intel feeds will help to identify newly found vulnerabilities and how to detect them.

Vulnerability Patching

- **Vendor Releases:**
 - Vendors should provide **fixes for vulnerabilities** on their applications and devices. Grabbing the latest version of software or firmware is the quickest way to get rid of the most vulnerabilities.
- **Operating System Patching:**
 - Each operating system has different ways of patching. It is important that you get familiar with how each OS in your environment deals with patching and plan accordingly.
- **Prioritising Patches:**
 - Again it is important to **prioritise** which systems receive **patches** based on the severity of the findings. Some systems may also require down time for the patch to be applied, so this needs to be scheduled.

Penetration Testing



Penetration Testing

External Pen-Test

- A third party provider is **hired to conduct testing** on an organisation, or specified systems.
- **Due-diligence** must be carried out on the Third Party to ensure they themselves are secure.
- Using an external pen testing company **ensures a true representation** of what an attacker would likely see (plus a bit more).
- Why use an External Pen-Test:
 - **Provides an understanding** of the security of your organisation from an outsiders perspective.
 - Means that skilled individuals can really test the systems (but only those in scope).
 - Legislative and regulatory purposes – Such as PCI-DSS or ISO 27001
 - Risk assessments
- **Types of Exercises:**
 - Password hygiene scanning
 - Vulnerability scanning
 - Secure configuration
 - Application security and configuration
 - Elevation or Privileges



Red Team

- A **hired team, or internal team**, of skilled information security technicians who try to break into your systems as if they were an attacker

Why use a Red Team:

- Perfect for seeing **how secure your systems** really are.
- Can provide **a good indication of risk ranking scores**. If someone you employ can hack your systems, then the likelihood of a genuine attacker is high.
- Enables you to **test** Disaster Recovery and Business Continuity strategies.

Types of Exercises:

- Password hash cracking
- Brute force attacks
- DDoS attacks (within reason)
- Vulnerability Exploitation
- Phishing attacks

Blue Team

- **A team of security analysts** setup to see what can and can't be seen when the organisation is under attack.
- Usually goes hand in hand with a Red Team event.
- Why use a Blue Team:
 - Enables testing **of security monitoring** controls and Disaster Recovery.
 - Education for the **internal** teams.
 - Gaps in security **monitoring**.
- Types of Exercises:
 - Security control implementation
 - Invoking DR controls
 - Stopping attackers from spreading
 - System isolation
 - Incident triage and response



Pen-Testing Tools

- **Nessus Scan:** looks for **software and configuration vulnerabilities** and **produces easy to read reports**.
- **Metasploit:** Scans devices for vulnerabilities and then offers exploits for them
- **Kali Linux:** Pre-built into the Operating System. Perfect for learning and testing security. Can also be used by attackers
- **Wireshark:** Packet Sniffing software
- **Nmap:** Shows IP addresses, services running, open ports and device ID if the device is broadcasting it.

Understanding Pen-Test Results

Scope of a PenTest: The pentest results should say **what was in scope**. Check this over and make sure nothing has been **missed, likewise**, nothing has been **scanned** which shouldn't have been.

Criticality Rating: Each finding should have **a criticality rating from Critical to Informational**. Informational means that it isn't a priority right now but could cause issues further down the line. Critical findings will usually mean you should consider shutting down the device/interface it was found on and fix it immediately.

CVSS Scores: combine various factors such as exploitability, likelihood, impact etc.. To produce an overall score. The higher the score, the worse the finding.

Remediation Recommendations: Each finding should have a **recommended** remediation which indicates **how best to fix** the problem.

False Positives: Some findings may be **considered false positives based** on how they are presented. It is up to you as an Information Security professional to understand if this is the case.

Activity

Case Study:

- Research the Fortnite XSS Attack.
- Research the 3ve Botnet

Self Learning

- Kali Linux Website - <https://www.kali.org/>
- Red Team vs. Blue Team - <https://www.youtube.com/watch?v=NoY-3bO0HH4>



-
- Questions?