Vulnerability analysis

- **Vulnerability research** helps identify vulnerabilities which could compromise the system
- Scanning types
  - **Active scanning**: interacting directly with the target network to discover vulnerabilities
  - **Passive scanning**: discovering vulnerabilities without a direct interaction with the target network

Vulnerability categories

- Misconfiguration
- Default installations
- Buffer overflows
- Unpatched servers
- Design flaws
- Operating system flaws
- Application flaws
- Open services
- Default passwords

Vulnerability assessment types

- **Active assessment**: through network scanners
- **Passive assessment**: by sniffing the traffic
- **External assessment**: vulnerabilities & threats that are accessible outside of the organization
- **Internal assessment**: vulnerabilities & threats that are present internally
- **Host-Based assessment**: vulnerabilities & threats on a specific server by examining the configuration
- **Network assessment**: identifies potential attacks on the network
- **Application assessment**: examines the configuration of the web infrastructure
- **Wireless network assessment**: vulnerabilities & threats in the organization's wireless network

Vulnerability management

- Evaluation and control of the risks and vulnerabilities in the system
- Phases:
  - **Pre-assessment phase**
    - Creating baseline: Identifying critical assets and prioritizing them
  - **Assessment phase**
    - Vulnerability assessment: identifying known vulnerabilities
  - **Post-assessment phase**
    - Risk assessment: assessing the vulnerability and risk levels for the identified assets
    - **Remediation**: mitigating and reducing the severity of the identified vulnerabilities
    - **Verification**: ensuring that all phases have been successfully completed
    - **Monitoring**: identifying new threats and vulnerabilities

Vulnerability assessment solution types

- Product-based solutions: installed in the internal network
- Service-based solutions: offered by third parties
- Tree-based assessment: different strategies are selected for each machine
- Inference-based assessment
  i. Find the protocols to scan
  ii. Scan and find the found protocols and their services,
  iii. Select the vulnerabilities and begins with executing relevant tests.

Vulnerability scoring systems

- Vulnerabilities that are identified are stored into databases
- Certain scores based on their severity and risk

CVSS - Common Vulnerability Scoring System

- A free and open industry standard for assessing the severity of computer system security vulnerabilities
- Helps to assess and prioritize vulnerability management processes.
- Assigns severity scores to vulnerabilities
- Score calculator depends on metrics that include ease and impact of exploit.

CVE - Common Vulnerabilities and Exposures

- Mitre.org
- List of common identifiers for publicly known cybersecurity vulnerabilities

- E.g. `CVE-2020-0023` : disclosure of user contacts over bluetooth due to a missing permission check on Android.

NVD - National Vulnerability Database

- U.S. government repository of standards based vulnerability management data
- nvd.nist.gov
- Includes databases of security checklist references, security-related software flaws, misconfigurations, product names, and impact metrics
    - E.g. CVE

Vulnerability assessment report

- Written after an assessment is performed
- Classified into **security vulnerability report** and **security vulnerability summary**.
- Details of what has been done and what has been discovered during the assessment
- Created to help organizations resolve security issues if they exist
- Typically contain information about the scan, target, and results.

Vulnerability assessment tools

- Also known as **vulnerability scanner**s
- Scanning solutions perform vulnerability penetration tests in three steps
    i. locate the live hosts in the network
    ii. enumerate open ports and services
    iii. test the found services for known vulnerabilities by analyzing responses.
- Tool types
    - Host-based vulnerability assessment tools
    - Depth assessment tools
    - Application-layer vulnerability assessment tools
    - Scope assessment tools
    - Active/Passive tools
    - Location/Data examined tools
- **OpenVAS**
    - Open-source software framework of several services and tools offering vulnerability scanning and vulnerability management.

Nmap

- Nmap has scripting functionality, written in LUA
- You can scan multiple servers for multiple ports for multiple vulnerabilities.

- `-A` : Enables OS detection, version detection, **script scanning** and traceroute.
- Read more about Nmap in Nmap | Scanning Tools
- See Detecting Shellshock using Nmap | Common vulnerabilities

Nessus

- Website
- 📝 Proprietary port and vulnerability scanner
- Scans include • misconfigurations • default passwords (has Hydra built-in) • DoS vulnerabilities
- Can be used to perform compliance auditing, like internal and external PCI DSS audit scans.

Burp Suite

- 📝 Proxy tool to scan web vulnerabilities
- Allows manual testers to intercept all requests and responses between the browser and the target application
- Allows to view, edit or drop individual messages to manipulate the server-side or client-side components of the application.

Nikto

- Nikto is an open source Nikto web server vulnerability scanner.
- Majorly looks for outdated software, dangerous files/CGI etc.
- E.g. `nikto -host cloudarchitecture.io`
- 🤗 Many of the modern scanners including Nessus, OpenVAS use Nikto to get information for their analysis.

Microsoft Baseline Security Analyzer (MBSA)

- Identifies missing security updates and common security misconfigurations
- Assesses Windows and its sofware e.g.• Internet Explorer • IIS web server • Microsoft SQL Server, • Office macro settings
- It's deprecated

Common vulnerabilities

Shellshock

- Also known as ***bashdoor*** or ***bash bug***
- Privilege escalation vulnerability enabling arbitrary commands execution
- 📝 Caused by family of security bugs in the Unix Bash shell
- Related CVE entries include: • *CVE-2014-6271*, *CVE-2014-6277* • *CVE-2014-6278*, *CVE-2014-7169* • *CVE-2014-7186*, *CVE-2014-7187*
- Achieved by manipulating the environment variable list and then cause Bash to run
- Upon startup Bash parser executes scripts saved as environment variables
- E.g. `$ env x='() { :;}; echo vulnerable' bash -c "echo this is a test"`
  - Prints first `vulnerable` then `this is a test`
- To exploit there needs to be away to talk to Bash.
- Often exploits websites using CGI
  - CGI stands for "Common Gateway Interface"
  - In Apache it's done using mod_cgi
    - Way to let Apache execute script files and send the output to the client
    - Apache passes information to CGI scripts using environment variables
  - E.g. if you you have a HTTP header named `Sike` in your request, you will have an environment variable named `HTTP_SIKE` available in your CGI.
- Big impact
  - Thousands of attacks were reported when the bug was revealed including botnets against United States Department of Defense.
  - > "Shellshock makes Heartbleed look insignificant" - ZDNet

Detecting Shellshock using Nmap

- Can use Shellshock script with Nmap scripting engine.
- `nmap -sV -p 80 --script http-shellshock --script-args uri=/cgi-bin/blabla.sh,cmd=ls 192.168.122.17.8`
  - `-sV` : detect services and versions
  - `-p` : port 80, you can also do -p- to scan for entire port range
  - `--script` : You can test different scripts / vulnerabilities, choose anything from scripts page

- `--script-args` : optional, 2 args, uri and cmd

SSL/TLS Vulnerabilities

Heartbleed

- 📝 Bug in OpenSSL library a widely used implementation of TLS.
- Introduced and patched in April 2014.
- Results from improper input validation (no boundary check) in TLS heartbeat extension
- Causing server to send more data in the memory than it allowed
  - Classified as **buffer over-read**
- Flow
  - TLS/DTLS Heartbeat flow:
    - Client: `Send me 4 letter word: "bird"` -> Server: `"bird"`
  - Malicious Heartbeat flow:
    - Client: `Send me 500 letter word: "bird"` -> Server: `bird. Server master key is 3131531535. User Carol wants to change password to "password 1 2 3"...`
- **Reverse Heartbleed**
  - Malicious server exploiting Heartbleed to read from client memory.
- Millions of webpages were affected, still there are IoT devices are vulnerable (see shodan search)
- Had big impact, some known ones are stealing of millions of patient records, hijacking accounts CEO impersonation ....
  - > "Heartbleed is the worst vulnerability found" - Forbes
- Can be exploited
  - Using Nmap: `nmap -p 443 --script ssl-heartbleed <target>`
    - Will return "State: NOT VULNERABLE" if not vulnerable.
  - Using Metasploit: openssl_heartbleed module

POODLE

- POODLE stands for "Padding Oracle On Downgraded Legacy Encryption"
- 📝 Forcing a degradation to a vulnerable SSL/TLS version
  - TLS handshakes are walked down the connection until a usable/vulnerable one is found
  - Exploits backwards compatibility
- Man-in-the-middle exploit
- Affects both SSL and TLS
  - Vulnerability was disclosed in October 2014 for SSL.
  - A variation used to attack TLS was disclosed in December 2014.
- **POODLE attack against SSL**
  - Takes advantage of Internet and security software clients' fallback to SSL 3.0.

    - Attackers make 256 SSL 3.0 request on average to reveal a single byte.
- **POODLE attack against TLS**
    - Caused by some implementation not following the TLS specifications.
    - Exploits CBC encryption mode in the TLS 1.0 - 1.2 protocols

FREAK

- Stands for "Factoring RSA Export Keys"
- Man-in-the-middle attack forcing downgrade of RSA key to a weaker length
- Enables successful brute-force attacks.
- Exploits cryptographic weakness in the SSL/TLS protocols

SSL/TLS Renegotiation

- 📝 Leads to plaintext injection attacks against SSL 3.0 and all current versions of TLS
- **Background**
    - Marsh Ray and Steve Dispensa release a document discussing a vulnerability in the design of TLS – November 4, 2009
    - Turkish grad student, Anil Kurmus, exploits the vulnerability to steal Twitter login credentials – November 10, 2009
- **Mitigation**
    - Quick fix was the renegotiation
    - Proposed standard (RFC 5746) is to verify previous renegotiation handshakes between client and server.
- **Testing**
    - Use `open_ssl s_client -connect <website>:443`
    - Then type `R` for renegotiate and `[ENTER]`

DROWN

- Stands for "Decrypting RSA with Obsolete and Weakened eNcryption"
- Exploits modern SSL/TLS suites by exploiting their obsolete SSLv2 protocol support.
- The only viable countermeasure is to disable SSLv2 on all servers

chapters\05-vulnerabilities\automated-penetration-testing-tools.md

Automated penetration testing tools

- CANVAS (proprietary)
  - Exploit gallery and development framework
- Core Impact (proprietary)
  - All-inclusive automated testing framework
- Nmap with custom scripts
  - Can used for • footprinting • scanning • vulnerability analysis
  - Also to carry out attacks e.g. as DoS tool

Automated vs manual penetration testing

- Automated testing cannot fully replace manual testing but as it has its own advantages and disadvantages
- **Automated testing advantages**
  - Help the initial analysis to understand where potential vulnerabilities exist
  - Enable the testers to build efficient exploit strategies to confirm the security vulnerabilities and weaknesses.
  - Same pen test multiple times from different entry points
  - Reduces costs
- **Automated testing disadvantages**
  - It can miss unforeseen instances
  - Usually works from "inside" of the network
  - Fails to work in complex scenarios
  - Usually does not exploit the vulnerabilities
  - Not as creative as humans (yet 😉) in e.g. social engineering

Metasploit

- 📝 Framework for building and performing exploit attacks against targets.
- Source code | Website
- Modular architecture allowing code re-use instead of copying or re-implement on a per-exploit basis

Free version

- Developing and executing exploit code against a remote target machine.
- Database of vulnerabilities and platform to execute different exploits for them.
- Fuzzing tools to discover vulnerabilities
- Automated exploitation of known vulnerabilities such as weak passwords for e.g. Telnet, SSH, HTTP.
- Manual exploitation and manual brute forcing
- Zenmap (Nmap GUI)

Paid (Pro) version

- Web application testing (OWASP Top 10)
- Dynamic payloads for anti-virus evasion
- Has web interface
  - 💡📝 A free alternative is **Armitage** that's open-source GUI.

Metasploit interfaces

`meterpreter`

- 📝 Payload that provides control over an exploited target system
- Runs as a DLL loaded inside of any process on a target machine
- Resides entirely in memory and writes nothing to disk

`msfvenom`

- Generates stand-alone payload
- 📝 Combines
  - Payload generation (old tool: `msfpayload`)
    - `-p <payload-name>` e.g. `-p windows/meterpreter/bind_tcp`
    - `-f <format>` e.g. `-f exe` or `-f raw` (shellcode)
  - Encoding (old tool: `msfencode`)
    - Used to avoid antivirus detection
    - Done by `-b` or `-e` flags
    - `-i <number>` allows encoding multiple times for more stealth
- E.g. `msfvenom -a x86 --platform Windows -p windows/shell/bind_tcp -e x86/shikata_ga_nai -b '\x00' -f python`
- See also msfvenom | Hiding files

`msfconsole`

- All-in-one centralized console for all of the options available in the MSF

- Contains the most features and is the most stable MSF interface
- E.g. flow for using unreal exploit:
    i. Run `msfconsole`
    ii. You can search for a service e.g. `unrealirc`
        - ❗Disclosure date is not same as when vulnerability found, it can be before but not published.
    iii. Use with `use exploit/unix/irc/unreal_ircd_3281_backdoor`
        - There can be multiple payloads, check with `show payload` and then set with `set PAYLOAD <name>`
        - Set required options (`show options` to list) and `set <option-name> <option-value>` to set
    iv. Run exploit using `exploit`
        - Hopefully you'll end up in terminal session as root :)