**Vulnerability Management**

**Vulnerability identification**

* Asset criticality
* Active vs. passive scanning

Active:

* Nmap
* Gobuster
* BurpSuite

Passive:

* Packet monitoring (Wireshark)
* DNS Recon
* OSINT

Mapping/Enumeration

* SSL & TLS:
  + HTTP over TLS (1.3)
  + Use TLS 1.2 or newer.
  + Vulnerable protocols:
    - SSL 2.0, SSL 2.1, TLS 1.0
  + Must use current, secure ciphers.
  + Certificates must remain valid and uncompromised.
* Certificate Management (PKI):
  + Use of an untrusted CA
  + Expiration of a certificate
  + Mismatch in the certificate name
* Domain Name Server:
  + Harvesting data with whois and nslookup
    - BGP Looking Glasses
    - nslookup microsoft.com 8.8.8.8
    - nslookup -query=mx microsoft.com
  + MX Lookup Tool
    - Used to specify the email server(s) responsible for a domain name.
  + DNS Zone transfer:
    - The easiest way to gather complete DNS information:
      * Domain servers
      * Hostnames
      * MX and CNAME records
      * TTL records
    - host -t axfr domain.name dns-server
    - dig axfr example.com @ns1.example.com
  + DNS Brute Forcing:
    - The scripted query for each IP address that an organization uses.
    - Bypass IDS and IPS systems
  + DNS Anti-harvesting Techniques:
    - Blacklisting systems or networks that abuse the service
    - Use CAPTCHAs to prevent bots
    - Privacy services that use third-party DNS registration information
    - Implement rate limiting to ensure that lookups are not done at high speeds
    - Not publishing zone files
* Internal IP Disclosure:
  + Bad packet headers revealing information that NAT should hide
  + HTTP version 1.0 request to the server without the Host header set, the server will refer to itself by its internal IP address
  + HTTP 1.0 protocol doesn't require the Host header to be set by the client as part of a request
  + ncat -ssl owa.mymailserver.net 443 GET / HTTP/1.0
* Microsoft SQL Server:
  + TCP port 1433
  + HTTPS connection through a URL: TCP port 443
* MySQL:
  + TCP port 3306
* VPN:
  + Protocols and encryption tunnels can be vulnerable (PPTP)
* Virtualization:
  + VM Escape
  + Management Interface Access
  + Virtual Host Patching
  + Virtual Guest issues
  + Virtual Network Issues
* Use of vulnerable, down-level of protocols:
  + NTLM v1
  + LANMAN
  + NetBIOS
    - nbtstat displays NetBIOS over TCP/IP protocol statistics
    - nbtstat -c shows the contents of the NetBIOS name cache and a list of name-to-IP address mapping
  + Unsigned LDAP
  + SMB
    - Accessing SMB Shares using smbclient: smbclient \\\\{target\_IP}\\{SHARE\_NAME}
* Printers:
  + Line Printer Requester, LPR/LPD TCP port 515
  + IPP TCP port 631
* Oracle Database Net Listener:
  + TCP 1521

**Validation**

* True positive
  + A legitimate attack that triggers to produce an alarm
* False positive
  + An event signaling to produce an alarm when no attack has taken place
* True negative
  + When no alarm is raised when an attack has taken place
* False-negative
  + An event when no attack has occurred and no detection is made

**Scanning parameters and criteria**

* Risks associated with scanning activities
* Vulnerability feed
* Scope
* Credentialed vs. non-credentialed
* Server-based vs. agent-based
* Internal vs. external
* Special considerations
* Types of data
* Technical constraints
* Workflow
* Sensitivity levels
* Regulatory requirements
* Segmentation
* Intrusion prevention system (IPS), intrusion detection system (IDS), and firewall settings

**Inhibitors to remediation**

* Memorandum of understanding (MOU)
* Service-level agreement (SLA)
* Organizational governance
* Business process interruption
* Degrading functionality
* Legacy systems
* Proprietary systems

**Analyze the output from standard vulnerability assessment tools**

**OWASP Zed Attack Proxy (ZAP)**

* OWASP ZAP is an open-source web application security scanner that detects common threats to web applications
  + SQL injection
  + XSS (Cross-site scripting)
  + CSRF (Cross-site request forgery)
  + Misconfigurations
  + Data leakage
* BurpSuite
  + A set of tools used for the penetration testing of *web applications*
    - It contains an intercepting proxy to see and modify the content of requests and responses in transit
    - Intruder fuzzer for brute-force enumeration and dictionary attacks on password forms
    - Repeater to change requests and responses
* Nikto
  + An open-source web server and web application scanner
    - Find SQL injection and XXS vulnerabilities
    - Report on unusual headers
    - Identify installed software
* Arachni
  + Open-source penetration testing framework for web applications
* HTTP Methods
  + GET
  + POST
  + PUT
  + HEAD
  + DELETE
  + PATCH
  + OPTIONS
  + CONNECT
  + TRACE

**Infrastructure vulnerability scanner**

* Common Vulnerabilities and Exposures (CVE)
* Nessus
* OpenVAS
* Qualys

**Software assessment tools and techniques**

* Static analysis
  + Source code analysis is done by reviewing the code for an application
* Dynamic analysis
  + Relies on the execution of the code while providing it input to test the software
* Reverse engineering
  + Use a compiler to convert the source code into binary (machine) code that the computer can read.
  + Use a decompiler to convert the binary code back into source code
* Fuzzing
  + Sending invalid or random data to an application to test its ability to handle unexpected data

**Enumeration**

* Nmap
  + Common Platform Enumeration (CPE) is a scheme for identifying hardware devices, operating systems, and applications developed by MITRE.
  + By default, Nmap uses a TCP SYN scan.
  + Nmap supports HTTP and SOCK 4 proxies, allowing the attacker to configure a remote host as a reverse HTTP proxy and bounce their scans through it, or you can use *proxy chains*:
    - proxychains nmap -sT -Pn -n -p445,139,88,80 10.10.226.53
  + Target Specification
    - nmap 10.129.86.241/24
  + Service Version Detection
    - nmap -sV 10.129.86.241
  + Operating System Detection
    - nmap -O 10.129.86.241
  + Host Discovery
    - nmap -sn 10.129.86.241
    - *Disable host discovery, port scan only*:
      * nmap -Pn 10.10.226.53
  + Timing & Performance
    - *Paranoid (0) Intrusion Detection System evasion*:
      * nmap 10.10.226.53 -T0
  + Firewall / IDS Evasion and Spoofing
    - *Use tiny fragmented IP packets*:
      * nmap 10.10.226.53 -f
    - *Relay connections through HTTP/SOCKS4 proxies*:
      * nmap –proxies http://192.168.1.1:8080, http://192.168.1.2:8080 192.168.1.1
    - *Set offset size*:
      * nmap 10.10.226.53 –mtu 32
    - *Appends random data to sent packets*:
      * nmap –data-length 200 10.10.226.53
  + Port Scans
    - -sS TCP SYN packet
    - -sU UDP Scan
    - -sT TCP connect; conducts a three-way handshake scan
    - -N Null scan sends a packet with the header bit set to 0
    - -sF send an unexpected FIN packet
    - -sX sends a packet with the FIN, PSH, and URG flags set to 1
    - -p Port Range
* hping / hping3
  + TCP/IP packet assembler and analyzer
    - sudo hping3 -S --flood -p 80 www.wisetut.com
    - sudo hping3 -S 192.168.200.15, send SYN packets to the target IP address
    - sudo hping3 -0 192.168.200.15, send a raw IP through the network
* Active vs. passive
  + Active:
    - Pinging hosts
    - Angry IP Scanner
    - Port scanning and service discovery (Nmap)
    - DNS forward or reverse lookup
    - DNS Zone Transfer
    - Banner Grabbing
      * Telnet
      * Wget
      * echo " " | nc -v 10.10.226.5 80
  + Passive Footprinting:
    - DNS Registrar checks
    - WHOIS lookups
    - BGP looking glass
    - Log data and configuration analysis
    - Capturing network traffic by using a sniffing tool
      * Wireshark
      * Zeek or Bro
      * pof
      * Netflow is a Cisco network protocol that collects IP traffic information
    - netstat shows active TCP and UDP connections filtered by protocol
      * Windows, Linux & macOS
    - *Disable promiscuous mode for all NICs* to mitigate attackers from analyzing system traffic

**Cloud infrastructure assessment tools**

* Scout Suite
  + A multi-cloud auditing tool into user's accounts of cloud service providers and retrieves configuration information using their service API
* Prowler / Inspector
  + An AWS-specific security configuration testing tool
* Pacu
  + An AWS-specific exploitation framework used for cloud service penetration tests

**Threats and vulnerabilities associated with specialized technology**

**Mobile**

* Vulnerable Legitimate Apps
* Jailbreaking
* Outdated OS
* Malware and Malicious Apps
* Compromised Wi-Fi Hotspots
* Phishing

**Internet of Things (IoT)**

* Weak default passwords
* Lack of security updates
* Lack of encryption *(in transit and at rest)*

**Embedded**

* Computer integrated into the operations of another device, such as a vehicle.

**Real-time operating system (RTOS)**

* Slimmed-down operating systems packed onto a single chip

**System-on-Chip (SoC)**

* An integrated circuit that combines many elements of a computer system into a single chip

**Field programmable gate array (FPGA)**

* Computer chips that allow the end user to reprogram their function
* FPFAs are easily cloned, increasing the risk of intellectual property theft

**Building automation systems**

* BAS is an intelligent system of both hardware and software, connecting heating, venting, and air conditioning systems (HVAC), lighting, security, and other systems to communicate on a single platform
  + HVAC systems
  + Electrical systems, including lighting
  + Security systems, including surveillance cameras and alarms
  + Plumbing systems
  + Fire alarms and other emergency systems
  + Elevators

**Threats and vulnerabilities operating in the cloud**

**Cloud service models**

* Software as a Service (SaaS)
* Platform as a Service (PaaS)
* Infrastructure as a Service (IaaS)

**Cloud deployment models**

* Public
* Private
* Community
* Hybrid

**Function as a Service (FaaS)/ serverless architecture**

* AWS Lambda
* Google Cloud Functions
* Microsoft Azure Functions

**Infrastructure as code (IaC)**

* DevOps
* Azure Resource Manager (ARM) templates

**Insecure application programming interface (API)**

* Limit exposure of the API keys to the smallest set of individuals possible
* Use different API keys for other users, applications, and services
* Restrict the rights associated with each API key
* Never transmit API keys over unencrypted channels
* Never store an API key in an unencrypted form
* Ensure API keys are removed from any public code repository like GitHub

**Improper key management**

* Unauthorized disclosure of a key
* Non-rotation of keys
* Inappropriate key storage

**Unprotected storage**

**Logging and monitoring**

* Insufficient logging and monitoring
* Inability to access

**Attack types**

* Extensible markup language (XML) attack
* Structured query language (SQL) injection
* Overflow attack
  + Buffer
    - Overflowing a memory location by placing a string longer than a program expects into a variable
    - Enable ASLR (Address Space Location Randomisation) and DEP (Data Execution Prevention) in Windows
  + Integer
  + Heap
* Remote code execution
* Directory traversal
  + Insert file system path values into a query string
  + http://10.10.226.146/policy?document='aup.pdf'
  + Enumerate web server on port 80 with gobuster  
    gobuster dir -u http://10.10.226.146/ -w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt -o gobuster\_scan.txt
* Privilege escalation
* Password Reuse:
  + Password spraying
    - Brute-force attacks resulting in thousands or millions of username attempts
  + Credential stuffing
    - Stolen username and password wordlists from another site
* Impersonation
  + OAuth open redirects
* On-path attack (previously known as man-in-the-middle attack)
  + Interfere in the communication flow between two systems
* Session hijacking
  + Session key or cookie exploitation causes the session to pass through a system under the attacker's control.
* Rootkit
  + Combined multiple malicious software tools to provide continued access to a system while hiding their existence
* Cross-site scripting (XSS)
  + An attacker embeds scripting commands on the website, which an unsuspecting user can execute

// Inside of input field, the following command will help find XSS by creating a simple alert

* <script>alert(1)</script>
  + Add the payload to the URL

# Exploiting a vulnerable URL parameter and alerting the user's cookie

* http://10.10.226.56/vulnerabilities/xss\_r/?name=<script>alert(document.cookie)</script>
  + Enable Output Encoding to prevent this type of attack
    - Translates special characters so that it won't be interpreted as part of the script
* Reflected
  + The attacker tricks the user into sending the attack to the server disguised as a legitimate query string or other content.
  + The server sends (reflects) the attack back to the user, causing it to execute
* Persistent
  + The attacker can store the attack code on a server, waiting for the user to request the infected content
* Document object model (DOM)
  + see *JavaScript*
  + Attacks occur within a database maintained by the user's web browser

**Vulnerabilities**

* Improper error handling
* Dereferencing
* Insecure object reference
* Race condition
* Broken authentication
* Sensitive data exposure
* Insecure components
* Insufficient logging and monitoring
* Weak or default configurations