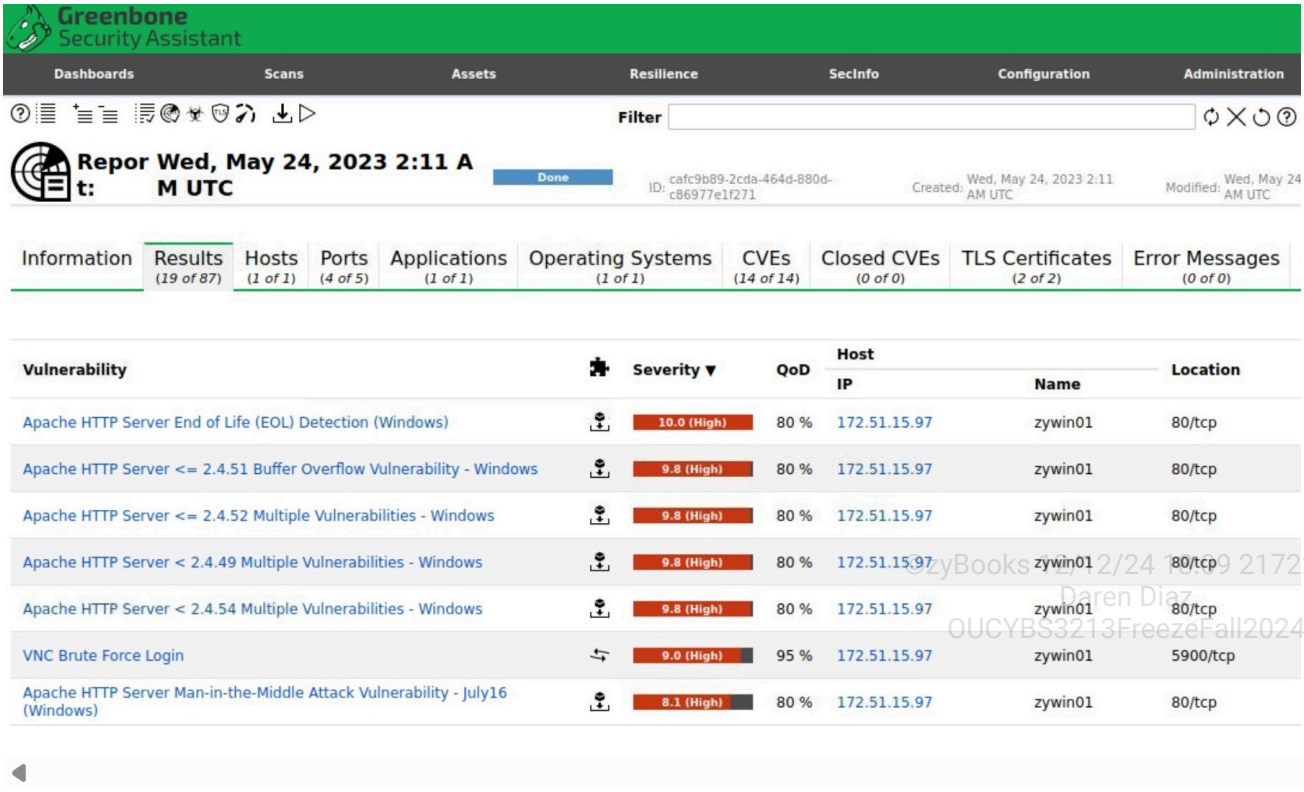
12.1 Vulnerability scans

Vulnerability scanner

Security assessment is the testing and evaluation of a system's security controls. A system could be a computer, network, native application, or web app. A security assessment determines if a system's security controls are implemented correctly and operating as intended.

A security assessment is typically automated and involves a vulnerability scanner. A ***vulnerability scanner*** is a program which scans a system for known vulnerabilities. Ex: Nessus and OpenVAS are vulnerability scanners. A vulnerability scanner identifies and prioritizes a system's vulnerabilities and provides recommendations for remediating those vulnerabilities. Ex: OpenVAS can identify an unpatched vulnerability in a Windows 11 device, rank the vulnerability based on the vulnerability's severity, and provide guidelines for patching the vulnerability.

Example 12.1.1: The results of an OpenVAS vulnerability scan of a computer running Windows Server 2022. The vulnerabilities are ranked based on the vulnerabilities' severity.



PARTICIPATION ACTIVITY

12.1.1: Security assessment.

1) The goal of a security assessment is to _____.

- automatically fix a system's vulnerabilities
- determine whether a system's security controls are implemented correctly and operating as intended
- ensure security controls were implemented by authorized personnel

2) In the vulnerability scan results shown in the above image, the severity of the VNC Brute Force Login is _____.

- 10
- 9.0
- 8.1

3) A vulnerability scanner ranks a Linux server's open TCP port 23 as a highly critical vulnerability because _____.

- the scanner cannot close the port without shutting down the server
- the port is used by telnet
- the port should only be open on Windows servers

4) The results of a vulnerability scan indicate that the default administrator password is used on a database server. The vulnerability is categorized as critical because _____.

- encryption cannot protect credentials for the administrator account

- an attacker may gain administrative privileges to the database server
- an attacker may be able to eavesdrop on network communications

Vulnerability databases

A vulnerability scanner uses databases of known vulnerabilities to identify and prioritize a system's vulnerabilities. ***Common Vulnerabilities and Exposures (CVE)*** is a list of publicly disclosed vulnerabilities and exposures. CVE is maintained by the MITRE Corporation with funding from the US Division of Homeland Security. Each CVE record is assigned a unique CVE ID. A CVE ID includes the year, followed by a four or five digit number. Ex: CVE-2022-0001 is the 1st vulnerability made public in the year 2022 and CVE-2023-33974 is the 33,974th vulnerability made public in the year 2023.

The ***National Vulnerability Database (NVD)*** is the U.S. Government repository of standards based vulnerability management data. The NVD includes databases of security checklist references, security-related software flaws, misconfigurations, product names, and impact metrics. NVD is maintained by the National Institute of Standards and Technology (NIST) and uses the Common Vulnerability Scoring System (CVSS) to evaluate the threat level of each CVE vulnerability.

Common Vulnerability Scoring System (CVSS) is an open industry standard for assessing the severity of computer system security vulnerabilities. A CVSS score is calculated using metrics that indicate ease and impact of a vulnerability exploitation. A CVSS score ranges from 0 to 10, with 10 being the most severe.

Example 12.1.2: The CVE record for the CVE-2023-28598 vulnerability. A CVE record has various information on each CVE, including a description and the date the vulnerability was created in CVE.

CVE-2023-28598 PUBLISHED

View JSON

Important CVE JSON 5 Information +

Assigner: Zoom Video Communications, Inc.
Published: 2023-06-13 **Updated:** 2023-06-13

Zoom for Linux clients prior to 5.13.10 contain an HTML injection vulnerability. If a victim starts a chat with a malicious user it could result in a Zoom application crash.

Product Status

Learn About the Versions Section +

Vendor	Versions
Zoom Video Communications, Inc.	***Default Status:*** *unaffected* • affected at **before 5.13.10**
Product Zoom for Linux clients	

References

- **https://explore.zoom.us/en/trust/security/security-bulletin/**

View additional information about **CVE-2023-28598** on NVD.
(Note: The NVD is not operated by the CVE Program)

Example 12.1.3: The NVD record for the CVE-2023-28598 vulnerability. An NVD record includes the vulnerability's assigned CVSS score.

CVE-2023-28598 Detail

Description

Zoom for Linux clients prior to 5.13.10 contain an HTML injection vulnerability. If a victim starts a chat with a malicious user it could result in a Zoom application crash.

QUICK INFO

CVE Dictionary Entry:
CVE-2023-28598
NVD Published Date:
06/13/2023
NVD Last Modified:
06/21/2023
Source:
Zoom Video Communications, Inc.

Severity

CVSS Version 3.x | CVSS Version 2.0

CVSS 3.x Severity and Metrics:

NIST: NVD — **Base Score:** 6.5 MEDIUM — **Vector:** CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:N/I:N/A:H

CNA: Zoom Video Communications, Inc. — **Base Score:** 7.5 HIGH — **Vector:** CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H

NVD Analysts use publicly available information to associate vector strings and CVSS scores. We also display any CVSS information provided within the CVE List from the CNA.

Note: It is possible that the NVD CVSS may not match that of the CNA. The most common reason for this is that publicly available information does not provide sufficient detail or that information simply was not available at the time the CVSS vector string was assigned.

References to Advisories, Solutions, and Tools

By selecting these links, you will be leaving NIST webspace. We have provided these links to other web sites because they may have information that would be of interest to you. No inferences should be drawn on account of other sites being referenced, or not, from this page. There may be other web sites that are more appropriate for your purpose. NIST does not necessarily endorse the views expressed, or concur with the facts presented on these sites. Further, NIST does not endorse any commercial products that may be mentioned on these sites. Please address comments about this page to nvd@nist.gov.

Hyperlink	Resource
https://explore.zoom.us/en/trust/security/security-bulletin/	Vendor Advisory

Weakness Enumeration

CWE-ID	CWE Name	Source
CWE-74	Improper Neutralization of Special Elements in Output Used by a Downstream Component ('Injection')	NIST

PARTICIPATION ACTIVITY 12.1.2: Vulnerability databases.

1) Why can't a vulnerability scanner use CVE to identify zero-day vulnerabilities?

- Because vulnerability scanners cannot access CVE
- Because CVE only contains a list of known vulnerabilities
- Because zero-day vulnerabilities are removed from CVE

2) Which CVE vulnerability was first made public in the year 2022?

- CVE-2023-8135

- CVE-2023-2022
- CVE-2022-1894

3) Which of the following CVE vulnerabilities is the latest vulnerability made public?

- CVE-2021-22415
- CVE-2020-19203
- CVE-2021-22512

4) What is the CVSS score assigned by NVD to the CVE-2023-28598 vulnerability?

- 3.x
- 6.5
- 7.5

5) How does CVE assign CVSS scores to each vulnerability?

- By requesting each vendor to assign a CVSS score to each vulnerability found in the vendor's products
- By comparing the severity of each new vulnerability to the CVSS score stored in the CVE database
- CVE does not assign CVSS score

Vulnerability scan types

Two vulnerability scan types exist:

- A ***non-credentialed vulnerability scan***, or ***unauthenticated vulnerability scan***, is a vulnerability scan in which a scanner identifies a system's vulnerabilities by probing the network services exposed by the system. Ex: A non-credentialed vulnerability scan can identify a system's open ports and the vulnerable services running on those ports.

- A ***credentialed vulnerability scan***, or ***authenticated vulnerability scan***, is a vulnerability scan in which a scanner has system privileges. A credentialed vulnerability scan can find more vulnerabilities than a non-credentialed vulnerability scan because the scanner logs into a system and has direct access to the system's components and configurations. Ex: A credentialed vulnerability scan can identify a system's software misconfigurations and missing operating system patches.

A non-credentialed or credentialed vulnerability scan can either be intrusive or non-intrusive. An ***intrusive vulnerability scan*** is a vulnerability scan which attempts to exploit a system's vulnerabilities found during a scan. Ex: Install malware on a device by exploiting a vulnerability found in Server Message Block (SMB) protocol using TCP port 445. A ***non-intrusive vulnerability scan*** is a vulnerability scan which only identifies and reports a system's vulnerabilities. Ex: Report the use of a weak hashing algorithm in signing a web server's SSL certificate.

Table 12.1.1: Comparison of credentialed and non-credentialed vulnerability scans.

	Credentialed	Non-credentialed
Authentication	Required	Not required
System impact	Low	High
Scan speed	Slow	Fast
Accuracy of results	High	Low
Primary user	System owner	Attacker

PARTICIPATION ACTIVITY 12.1.3: Vulnerability scan types.

Select the vulnerability scan types in each scenario.

1) A scan performed by an attacker to find a server's open ports

   ○ Credentialed

- Non-credentialed

2) Scans performed by a company's employees who are tasked with finding vulnerabilities in the company's web servers
   - Credentialed
   - Non-credentialed

3) A scan which can identify specific versions of a database server's applied operating system patches
   - Credentialed
   - Non-credentialed

4) A scan which may miss scanning transient devices that are not always connected to a network
   - Credentialed
   - Non-credentialed

5) A scan which may be disruptive and have a negative effect on the scanned system
   - Credentialed
   - Non-credentialed

Vulnerability scan errors

Two types of errors may occur in a vulnerability scan:

- A ***false positive***, or ***Type I error***, is a system state or configuration which is mistakenly identified as a vulnerability by a scanner. Ex: Identifying lack of authentication in a database server even if the database server authenticates all users and processes. A false positive error commonly occurs in non-credentialed vulnerability scans because a scanner can access only a subset of the information the scanner needs to determine whether a vulnerability exists.
- A ***false negative***, or ***Type II error***, is a potential vulnerability which is not detected by a vulnerability scanner. Ex: Not identifying a vulnerability caused by a misconfigured web server that may allow SQL injections attacks. False negatives occur with all zero-day vulnerabilities because vulnerability scanners can only identify known vulnerabilities.

Although both types of errors may occur in a vulnerability scan, false negatives have the potential to be more damaging than false positives. False negatives allow for a system's vulnerabilities to remain undetected, and thus exploitable.

Table 12.1.2: Comparison of false positive and false negative vulnerability scans errors.

	False positive	False negative
Error	Type I	Type II
Vulnerability	Does not exist	Exists
Root cause	Insufficient information during a scan	Outdated vulnerability databases Zero-day vulnerabilities

PARTICIPATION ACTIVITY 12.1.4: Vulnerability scan errors.

Select the vulnerability scan error type in each scenario.

1) May create a false sense of security.
   - False positive
   - False negative

2) May lead to investments in time and resources without improving a system's security.
   - False positive
   - False negative

3) Failing to detect a virus in an infected file.
   - False positive
   - False negative

4) A vulnerability scanner detects that SSH-2 is installed on a server, but since

the scanner cannot determine the server's operating system, the scanner identifies a vulnerability with SSH-2.

- False positive
- False negative

5) A non-credentialed vulnerability scanner identifies a vulnerability in the Apache web server after determining the Apache software version using the Apache banner.

- False positive
- False negative

SCAP

Security content automation protocol (SCAP) *is a collection of open standards used to automate the process of securing IT systems. Developed by the National Institute of Standards and Technology (NIST), SCAP provides a standardized framework for implementing network security across various platforms and environments. SCAP's functions include automating vulnerability management, measurement, and policy compliance evaluation.*

SCAP integrates a number of individual standards, such as common vulnerabilities and exposures (CVE) and common configuration enumeration (CCE). Such an integration facilitates the efficient and consistent execution of tasks like vulnerability scanning and the assessment of system configurations against recommended benchmarks, which are standardized sets of best practices and configuration settings. By using SCAP, organizations can enhance security protocols through automated processes, ensuring compliance with standards and reducing exposure to security risks.

CHALLENGE ACTIVITY 12.1.1: Vulnerability scans.

581480.4344582.qx3zqy7

Start

Select the vulnerability database(s) with the given attributes.

CVE	NVD	
☐	☐	Uses the Common Vulnerability Scoring System (CVSS) to evaluate the t CVE vulnerability
☐	☐	Is maintained by the MITRE Corporation
☐	☐	Is a list or directory of publicly disclosed vulnerabilities without CVSS sc

1 2 3

Check Next

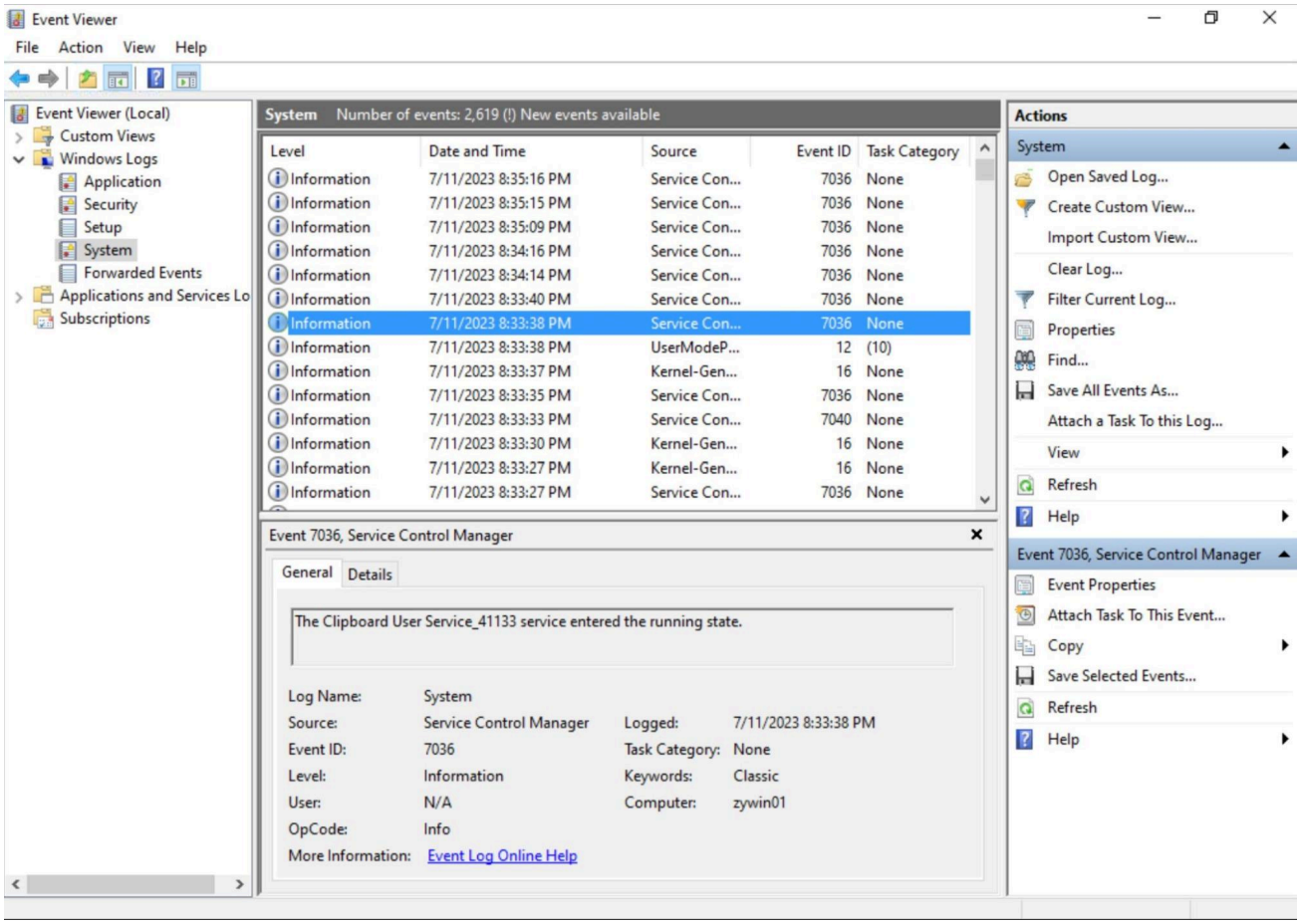
12.2 Event management

SIM and SEM

A security assessment can be conducted by analyzing log data and correlating the events occurring on various systems. ***Security information management (SIM)*** is the practice of collecting, storing, and managing log data from network devices and software applications. ***Log data*** is a timestamped record of an event. Ex: a VPN server records a VPN user's username and timestamps a login attempt as log data. A SIM system may have search, aggregation, and visualization capabilities.

Security event management (SEM) is the practice of monitoring, identifying, and reporting security-related events. A SEM system detects threats and vulnerabilities by identifying relationships and patterns in log data using statistical analysis. Ex: a SEM system can detect an IoT device's malware infection by correlating firewall, router, and IPS log data.

Example 12.2.1: Windows Event Viewer displays log of application and system messages, including errors and warnings.



PARTICIPATION ACTIVITY 12.2.1: SIM and SEM.

Select the information or event management practice in each statement.

1) Identifies relationships and patterns in log data and can detect threats
   - ○ SIM
   - ○ SEM

2) The practice of collecting, storing, and managing log data from network devices and software applications

- SIM
- SEM

3) Detects network intrusions

- SIM
- SEM

4) The practice of monitoring, identifying, and reporting security-related events

- SIM
- SEM

5) Correlates different security events

- SIM
- SEM

Security information and event management (SIEM)

Security information and event management (SIEM) is an approach to security management that combines SIM and SEM functionality into a single system. A SIEM system collects and analyzes log data, correlates events in real-time, detects threats, and generates alerts. A SIEM system can collect syslog data (a standard for message logging) and use network packet captures to gain additional insight into security events.

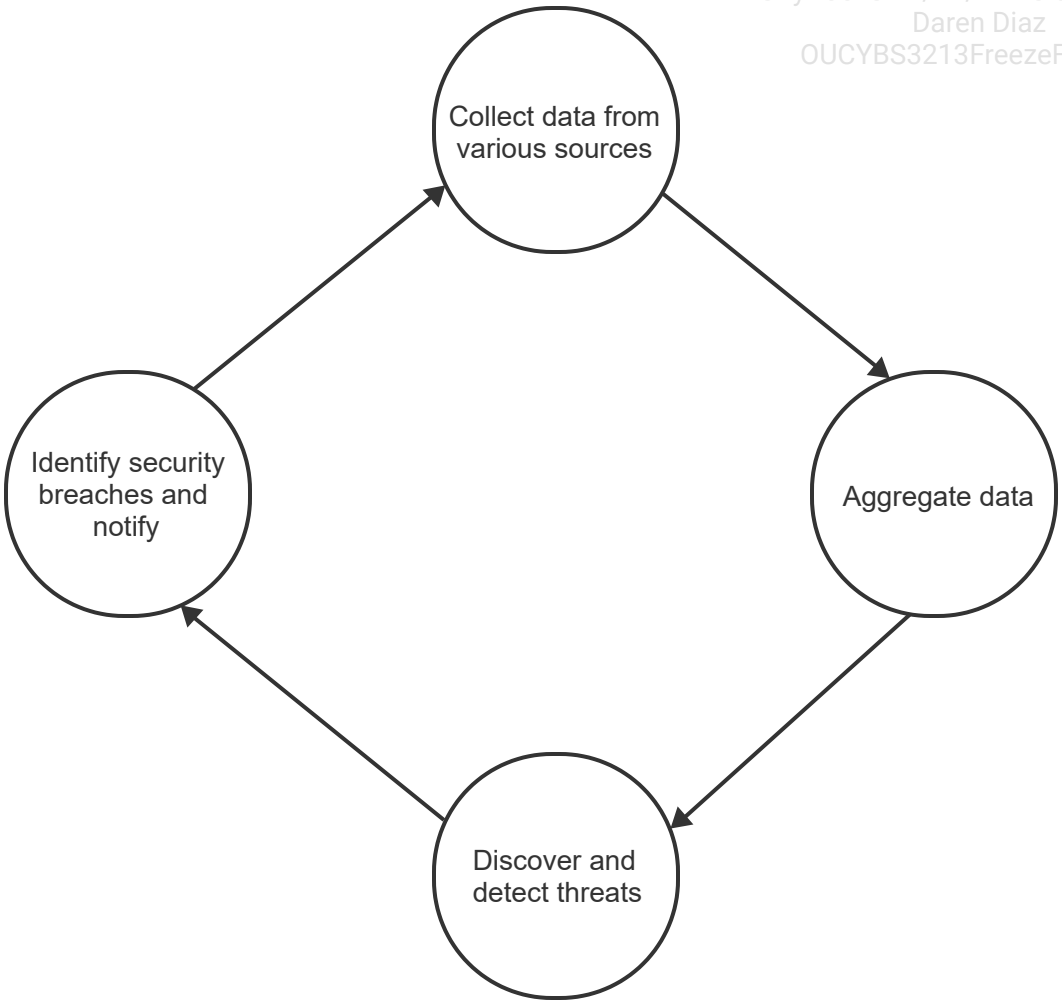
A SIEM system uses log collectors to gather log data. A ***log collector*** is a process or agent responsible for collecting log data from various systems. ***Log aggregation*** is the process by which a SIEM system combines similar events to reduce event volume. Log aggregation can be performed based on various parameters, including source IP address, destination IP address, or event ID. Ex: aggregating log data containing destination IP address 74.158.57.62. SIEM systems utilize either agent-based or agentless monitoring. Agent-based methods install software on targets for detailed data collection, while agentless methods gather data via existing network protocols.

Correlation rules define how a SIEM system generates alerts when a specific event occurs. A simple SIEM rule defines an event type and the response. Ex: a ZIP file is attached to an email. A composite rule combines two or more simple rules. Ex: six failed authentication attempts to a VPN server from the same IP address within five minutes.

A SIEM system may also have the capability to detect insider threats by analyzing user behavior. ***User behavior analytics (UBA)*** is the use of behavioral analytics and machine learning algorithms to identify

abnormal user behavior. UBA can identify insider threats or attacks that use compromised insider credentials by detecting deviations from normal user behavior.

PARTICIPATION ACTIVITY 12.2.2: SIEM process flow.



Animation content:

Static figure: Four circles positioned around an imaginary circle. The text inside the top circle states "Collect data from various sources". An arrow extends from the circle on the top to the circle on the right. The text inside the circle on the right states "Aggregate data". An arrow extends from the circle on the right to the circle at the bottom. The text inside the circle at the bottom states "Discover and detect". An arrow extends from the circle at the bottom to the circle on the left. The text inside the circle on the left states "Identify security breaches and notify".
Step 1: The circle on the top is revealed. A SIEM system collects log data and events from various sources in an IT infrastructure.
Step 2: The arrow from the circle on the top to the circle on the right and the circle on the right is revealed. Aggregate data.

Step 3:The arrow from the circle on the right to the circle on the bottom and the circle on the bottom is revealed. Discover and detect threats.
Step 4: The arrow from the circle on the bottom to the circle on the left and the circle on the left is revealed. Identify security breaches and notify.

Animation captions:

1. A SIEM system collects log data and events from various sources in an IT infrastructure.
2. Data is stored, indexed, and categorized to reduce event volume.
3. Data is correlated and examined to find threats.
4. Security breaches are identified and alerts are generated.

PARTICIPATION ACTIVITY 12.2.3: SIEM.

1) SIEM combines the functionality of which of the following systems?
   - SEM and SSL
   - SIM and SSH
   - SIM and SEM

2) What is the purpose of log aggregation?
   - To collect log data
   - To reduce event volume
   - To increase storage space

3) Which one of the following is a composite SIEM rule?
   - Assigning IP address 54.2.68.154 to a router and replacing the router's CPU.
   - A file upload from IP address 54.2.68.154
   - 120 login attempts from IP address 54.2.68.154 within 5 seconds.

Security orchestration, automation, and response (SOAR)

Automating the assessment and prioritization of security events decreases analysis errors and improves threat detection and response accuracy. Two processes automate security assessment and prioritization:

- ***Security orchestration*** is the integration and analysis of data from disparate security systems. Ex: integrating and analyzing log data from a network intrusion detection system (NID), a host-based firewall, and a wireless intrusion prevention system (WIPS).
- ***Security automation*** is the process of executing security operations-related tasks without human intervention. Ex: performing automated vulnerability scans and log data analysis.

Security orchestration, automation, and response (SOAR) is the combination of security orchestration and security automation tools to respond to security events with limited or no human intervention. SOAR systems enhance the accuracy of threat detection and response by automating tasks such as vulnerability scanning, log analysis, auditing, and incident handling.

Automated reports and dashboards offer real-time visualization of data and analytics. By aggregating information from diverse sources like network intrusion detection systems, host-based firewalls, and wireless intrusion prevention systems, dashboards present complex data in a comprehensible format. Such visualization aids quick decision-making and monitors the effectiveness of response strategies. Additionally, incorporating threat intelligence feeds into dashboards improves situational awareness and threat identification.

Beyond automated dashboards, attestation improves trust in security measures through third-party verification of compliance and control effectiveness. An audit committee oversees such processes to ensure compliance to internal and external audit standards. Internal compliance initiatives along with proactive self-assessments enable organizations to identify and address vulnerabilities promptly, ensuring readiness for external audits.

Table 12.2.1: Comparison between SIEM and SOAR systems.

	SIEM	SOAR
Data sources	Log and event data	Security alerts and threat intelligence
Notifications	Generates alerts	Ingests alerts from SIEM and other sources
Primary tasks	Analyzes and correlates data to identify potential threats	Enriches and correlates alerts to determine risk
Threat response	Notifies security analysts	Automatically orchestrates actions across integrated tools

PARTICIPATION ACTIVITY 12.2.4: Security orchestration, automation, and response (SOAR).

How to use this tool

SOAR | **Security incident response** | **Security orchestration** | **Security automation**

	The integration and analysis of data from disparate security systems
	The process of executing security operations-related tasks with limited human intervention.
	The processes used to prepare, detect, contain, and recover from security events.
	The technologies that automate incident response.

Reset

Threat hunting

Threat hunting is the practice of proactively searching for undetected threats on an organization's network and finding security incidents that detection systems such as SIEM have not detected. Threat hunting assumes that the organization has already been breached and aims to identify and mitigate the risk of attacks before the attacks cause harm to the organization.

Threat hunting utilizes various threat information sources to gain deeper insight into the threat landscape and to better understand new vulnerabilities and attack vectors. Threat information sources may include security advisories and bulletins, threat feeds, user behavior analytics, and vulnerability databases.

CHALLENGE ACTIVITY | 12.2.1: Security information and event management (SIEM).

581480.4344582.qx3zqy7

Start

Select the information or event management practice in each statement.

(1) Reports a failed attempt by anti-virus software to remove a virus from a web server

Pick

(2) Monitors all external connection attempts from a malicious IP address

Pick

(3) Identifies an administrator/root login outside working hours

Pick

(4) Provides functionality to reduce storage requirements for log data

Pick

(5) Manages log data from hubs

Pick

12.3 Penetration testing

Penetration testing

Penetration testing (pentesting), or ***ethical hacking***, is an authorized attack on a system to evaluate the system's security. A penetration tester, or pentester, attempts to breach some or all of a system's security controls by using tools and techniques an attacker uses. Penetration testing types:

- ***Physical penetration testing*** evaluates physical security measures like locks and access controls.
- ***Offensive penetration testing*** targets breaching security measures from an attacker's perspective.
- ***Defensive penetration testing*** assesses the effectiveness of defensive mechanisms against unauthorized access.
- ***Integrated penetration testing*** combines offensive and defensive testing to provide a comprehensive security assessment.

Rules of Engagement (ROE) is a document describing the conditions and limitations under which a pentester conducts pentesting. An ROE is established before the start of pentesting to provide pentesters authority to conduct defined activities without the need for additional permissions. Ex: the type and scope of testing and procedures for handling sensitive data.

Pentesting environments:

- In an unknown environment, or close-box pentesting, a pentester is not provided with any information on the target systems.
- In a partially known environment, a pentester is provided with limited information on some of the target systems.
- In a known environment, or open-box pentesting, a pentester is provided with full privileges and information on all the target systems.

Table 12.3.1: Pentesting environments.

	Unknown environment	Partially known environment	Known environment
Objective	Mimic a true attack	Evaluate an organization's vulnerability to insider threats	Simulate an attack where an attacker gains access to a privileged account
Access level	No access	Limited access	Full access
Advantages	Most realistic (testing is performed from attackers' point of view	Most efficient (saves time and money)	Most comprehensive (less likely to miss a vulnerability)
Disadvantages	Time consuming and more likely to miss a vulnerability	May miss vulnerabilities on systems with no information	More information is released to pentesters

PARTICIPATION ACTIVITY 12.3.1: Penetration testing.

1) A pentester performs what type of attack against target systems?
   - Unauthorized
   - Authorized
   - Illegal

2) Why would pentesters not find a vulnerability in a web server?
   - Because web server vulnerabilities are patched before pentesting begins
   - Because pentesters may not have been authorized to find web server vulnerabilities
   - Because pentesters cannot use the tools and techniques that

attackers use to find web server vulnerabilities

3) Which pentesting environment presents a similar situation to a pentester as to an attacker?

- Partially known environment
- Known environment
- Unknown environment

4) Why would a web app's user credentials be provided to a pentester tasked with pentesting the web app?

- Because the main objective of pentesting a web app is to find vulnerabilities within the web app
- Because a pentester cannot obtain the user credentials to a web app in any other way
- Because a pentester does not need user credentials to a web app to evaluate the security of the web app

Reconnaissance

Pentesting begins with gathering information on the target environment. ***Reconnaissance***, or ***footprinting***, is the act of discovering a target system's potential vulnerabilities. Reconnaissance types:

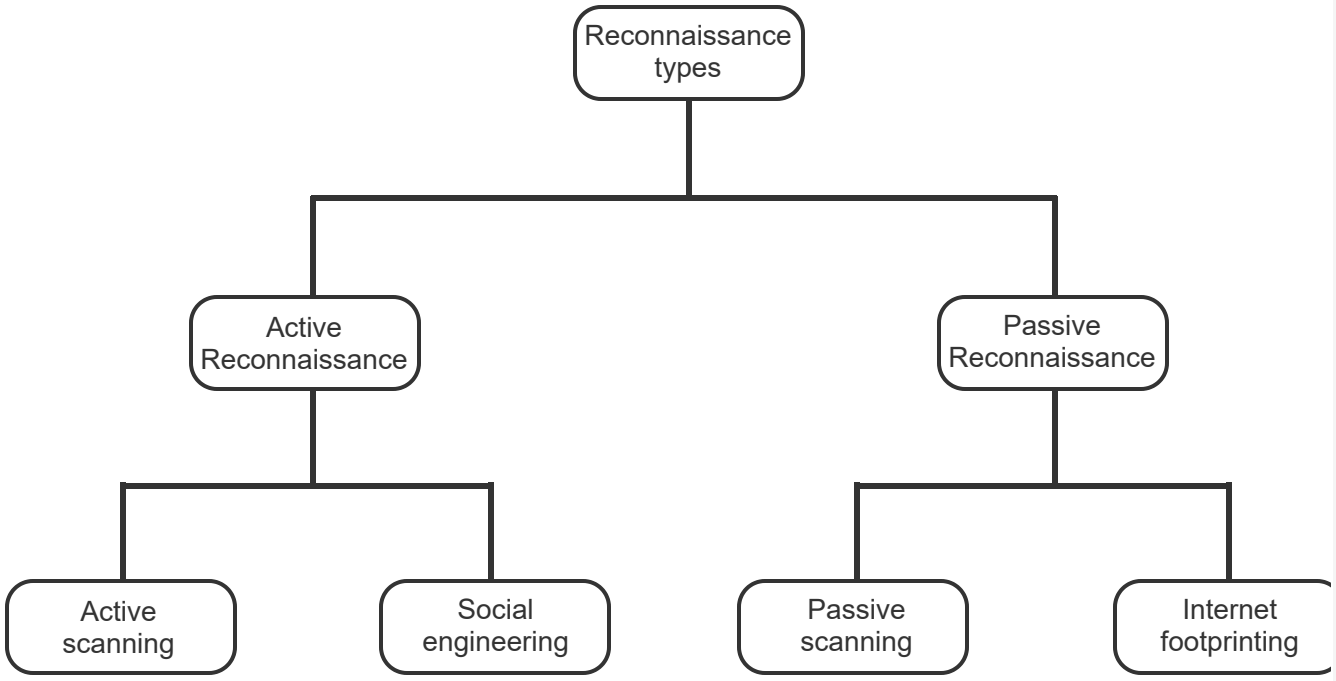
- ***Active reconnaissance*** collects information through direct interaction. Ex: conducting a port scan on a web server. Active reconnaissance produces actionable results quickly, but active reconnaissance is noisy and detectable.
- ***Passive reconnaissance*** collects information without direct interaction. Ex: using open-source intelligence (OSINT) to collect information on a company's employees. Passive reconnaissance collects information slower, but has a lower risk of detection.

Passive reconnaissance techniques:

- ***War driving*** is the act of searching for wireless networks from a moving vehicle.
- ***War flying*** is the act of searching for wireless networks from a flying object such as a drone, helicopter, or airplane.

Active and passive reconnaissance may be used together in pentesting. Ex: war driving to locate a wireless network followed by pinging a wireless access point on the network.

PARTICIPATION ACTIVITY 12.3.2: Reconnaissance.



Animation content:

Static figure: Seven text boxes with the top box stating reconnaissance types. Underneath the reconnaissance text box are two other text boxes. The text box on the left states active reconnaissance and the text box on the right states passive reconnaissance. Underneath the active reconnaissance text box are the active scanning and social engineering text boxes. Underneath the passive reconnaissance text box are the passive scanning and internet footprinting text boxes.
Step 1 : The two types of reconnaissance are active and passive.
Step 2: Active reconnaissance includes active scanning of an organization's computing resources and may include, port scanning, ping sweeps, and OS fingerprinting.
Step 3: Social engineering techniques target an organization's employees to obtain sensitive information or to install malware on employees' devices.
Step 4: Passive reconnaissance includes passive scanning using tools such as Wireshark for packet capture, or search engines such as Shodan to find an organization's Internet-connected servers and IoT devices.

Step 5: Internet footprinting is used to obtain information on an organization and the organization's employees using OSINT, social media platforms, and websites such as Google Earth and Google Maps.

Animation captions:

1. The two types of reconnaissance are active and passive.
2. Active reconnaissance includes active scanning of an organization's computing resources and may include, port scanning, ping sweeps, and OS fingerprinting.
3. Social engineering techniques target an organization's employees to obtain sensitive information or to install malware on employees' devices.
4. Passive reconnaissance includes the use of search engines such as Shodan to find an organization's Internet-connected servers and IoT devices.
5. Internet footprinting is used to obtain information on an organization and the organization's employees using OSINT, social media platforms, and websites such as Google Earth and Google Maps.

PARTICIPATION ACTIVITY 12.3.3: Reconnaissance.

Select the reconnaissance type in each scenario.

1) Using Wireshark to capture wireless network packets
   - ○ Active
   - ○ Passive

2) Using Google to search for a firewall's default password
   - ○ Active
   - ○ Passive

3) Pinging an application server
   - ○ Active
   - ○ Passive

4) Collecting information on a company's employees on LinkedIn
   - ○ Active
   - ○ Passive

5) Using Nessus to scan for a database server's vulnerabilities

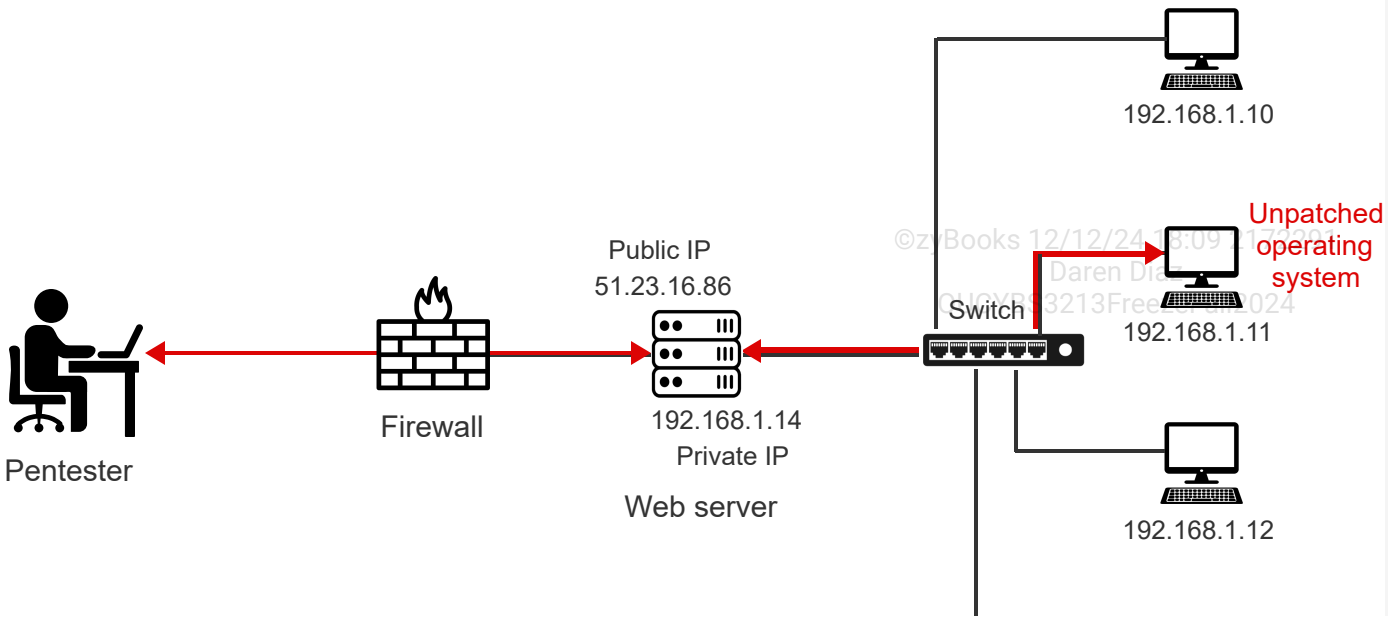
- Active
- Passive

Penetration testing techniques

A pentester uses various techniques to find and exploit vulnerabilities on as many systems as possible. A ***lateral movement*** is the act of moving from a compromised host to another host on the same network. Lateral movements are used for finding and exploiting vulnerabilities in other systems located on the same network. Ex: moving laterally from a compromised application server to a database server to exploit the database server's vulnerabilities. A lateral movement may also be used for privilege escalation on the same host. Ex: gaining administrator/root account access on a compromised database server. ***Pivoting*** is the act of using a compromised system as a platform to launch attacks against other systems. Pivoting bypasses security controls aimed at preventing unauthorized access to internal systems such as those implemented by firewalls.

Persistence is the set of techniques used for maintaining access to compromised systems. Ex: adding startup scripts, creating privileged accounts, and scheduling system tasks. Persistence enables a pentester to maintain system foothold across changed credentials, restarts, and other events causing access disruptions. ***Cleanup*** is the act of removing all the environment changes performed during an attack. Ex: deleting all accounts created, undoing all system configuration changes, and removing log files.

PARTICIPATION ACTIVITY 12.3.4: Pivoting.



Animation content:

Static figure: A local area network with four hosts and a web server connected to a switch. The host with IP address 192.168.1.11 has a label which states unpatched operating system. The web server has a private IP address of 192.168.1.14 and a public IP address of 51.23.16.86 which is connected to a firewall. On the left, a pentester is sitting at a desk working on a computer.
Step 1: A LAN consisting of four hosts and a web server. The network hosts are protected by a firewall which only allows inbound connections to the web server.
Step 2: A pentester exploits a web server vulnerability which enables the pentester to open a root terminal on the web server. The pentester can now use pivoting to launch attacks against the hosts on the LAN.
Step 3: The pentester uses the compromised web server to scan for vulnerabilities on the LAN and finds a host (IP address 192.168.1.11) with an unpatched operating system.
Step 4: The firewall prevents access to the vulnerable host from outside the LAN. However, the pentester uses the compromised web server to exploit the host's vulnerability, bypassing the firewall by pivoting.

Animation captions:

1. A LAN consists of four hosts and a web server protected by a firewall, which only allows inbound connections to the web server.
2. A pentester exploits a web server vulnerability enabling the pentester to open a root terminal on the web server. The pentester can now use pivoting to launch attacks against the hosts on the LAN.
3. The pentester uses the compromised web server to scan for vulnerabilities on the LAN and finds a host (IP address 192.168.1.11) with an unpatched operating system.
4. The firewall is unable to prevent access to the vulnerable host from outside the LAN because the pentester pivoted from the compromised web server to the vulnerable host.

PARTICIPATION ACTIVITY | 12.3.5: Penetration testing activities.

1) Which of the following requires a lateral movement?

   - ○ Escalating access privileges on a compromised system

- Removing changes made to a compromised system
- Exploiting vulnerabilities in other systems on a network

2) Which activity involves removing traces of pentesting from the compromised systems?

- Pivoting
- Cleanup
- Persistence

3) How is a bug bounty program different from pentesting?

- Pentesting does not reward pentesters
- Pentesting is not an ongoing activity and can only be done in coordination with the system owner
- Pentesting does not attempt to find software vulnerabilities

Responsible disclosure programs

*Software vulnerabilities can be discovered and disclosed through a combination of responsible disclosure and bug bounty programs. A **responsible disclosure program** provides a framework for reporting and addressing security vulnerabilities in a controlled and ethical manner. A **bug bounty program** is a responsible disclosure program component where a vendor encourages individuals to identify and report vulnerabilities in software or a website in exchange for recognition or compensation. By engaging a diverse community of security researchers, a bug bounty program fosters a collaborative approach to security and facilitates early detection and resolution of vulnerabilities.*

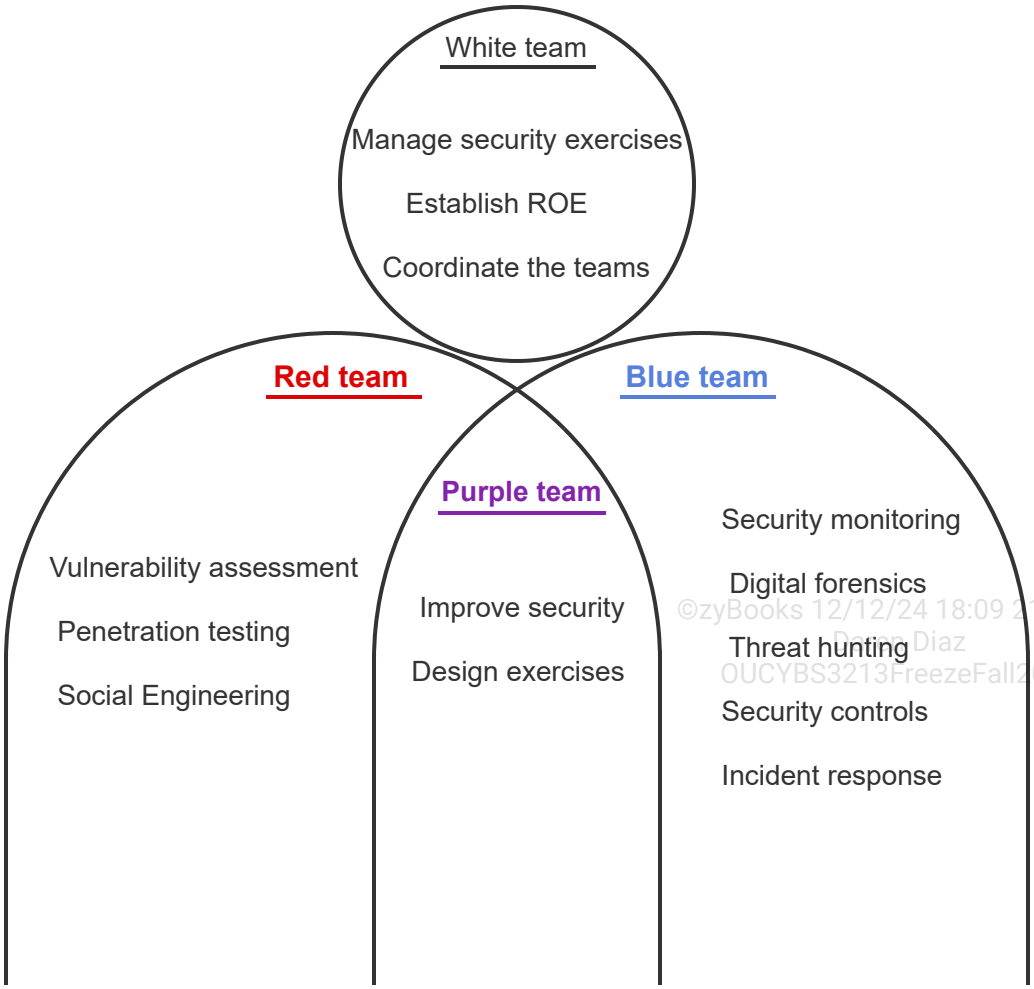
12.4 Security teams and TTP analysis

Security teams

An organization's security can be assessed by a security team conducting a security exercise. A security team is tasked with simulating an attack or utilizing incident response to test an organization's overall security posture. Four security teams participate in security exercises:

- A ***red team*** is a security team tasked with attacking an organization. Red team members can be internal or external to an organization.
- A ***blue team*** is a security team tasked with defending an organization. Blue teams identify and remove security vulnerabilities and verify security control effectiveness.
- A ***purple team*** is the temporary combination of red and blue teams. Purple teams help the blue teams improve security defenses by collaborating with the red teams.
- A ***white team*** is a security team tasked with managing red and blue team operations. White teams also establish the rules of engagement (ROE) for the red teams.

PARTICIPATION ACTIVITY 12.4.1: Red, blue, purple, and white teams.



Animation content:

Static figure: Venn diagram on the left and a text box on the right. The left circle is labeled red team and includes "vulnerability assessment", "penetration testing", "social engineering", and "replicate TTPs". The right circle is labeled blue team and includes "security monitoring", "digital forensics", "threat hunting", "security controls", and "incident response". The overlapping section is labeled purple team and includes "improve security", "create new TTPs", and "design exercises". The text box on the right includes "manage security exercises", "establish ROE", and "coordinate the teams".
Step 1: A red team is tasked with attacking an organization by finding and exploiting vulnerabilities using various techniques. A red team may replicate TTPs to emulate real-world threats.
Step 2: A blue team is tasked with defending an organization using various techniques. A blue team conducts threat hunting to find undetected security events and performs incident response.
Step 3: A purple team consists of red and blue team members and is tasked with improving the organization's security. A purple team designs security exercises to measure the organization's security readiness.
Step 4: A white team coordinates the teams, manages the security exercises, and creates ROEs for the red team.

Animation captions:

1. A red team is tasked with attacking an organization by finding and exploiting vulnerabilities using various techniques.
2. A blue team is tasked with defending an organization using various techniques. A blue team conducts threat hunting to find undetected security events and performs incident response.
3. A purple team consists of red and blue team members and is tasked with improving the organization's security. A purple team designs security exercises to measure the organization's security readiness.
4. A white team coordinates the teams, manages the security exercises, and creates ROEs for the red team.

PARTICIPATION ACTIVITY 12.4.2: Training for IT personnel.

1) A _____ team member utilizes active reconnaissance during a security exercise.
   - red
   - blue

- white

2) The _____ team consists of defenders in a security training exercise.
   - red
   - blue
   - purple

3) The _____ team may include members of a security operations center (SOC).
   - Blue
   - Red
   - White

4) A _____ team consists of both red and blue team members.
   - white
   - blue
   - Purple

5) A _____ team is responsible for overseeing a security exercise.
   - red
   - blue
   - white

6) A _____ team may include pentesters.
   - red
   - white
   - blue

TTP analysis

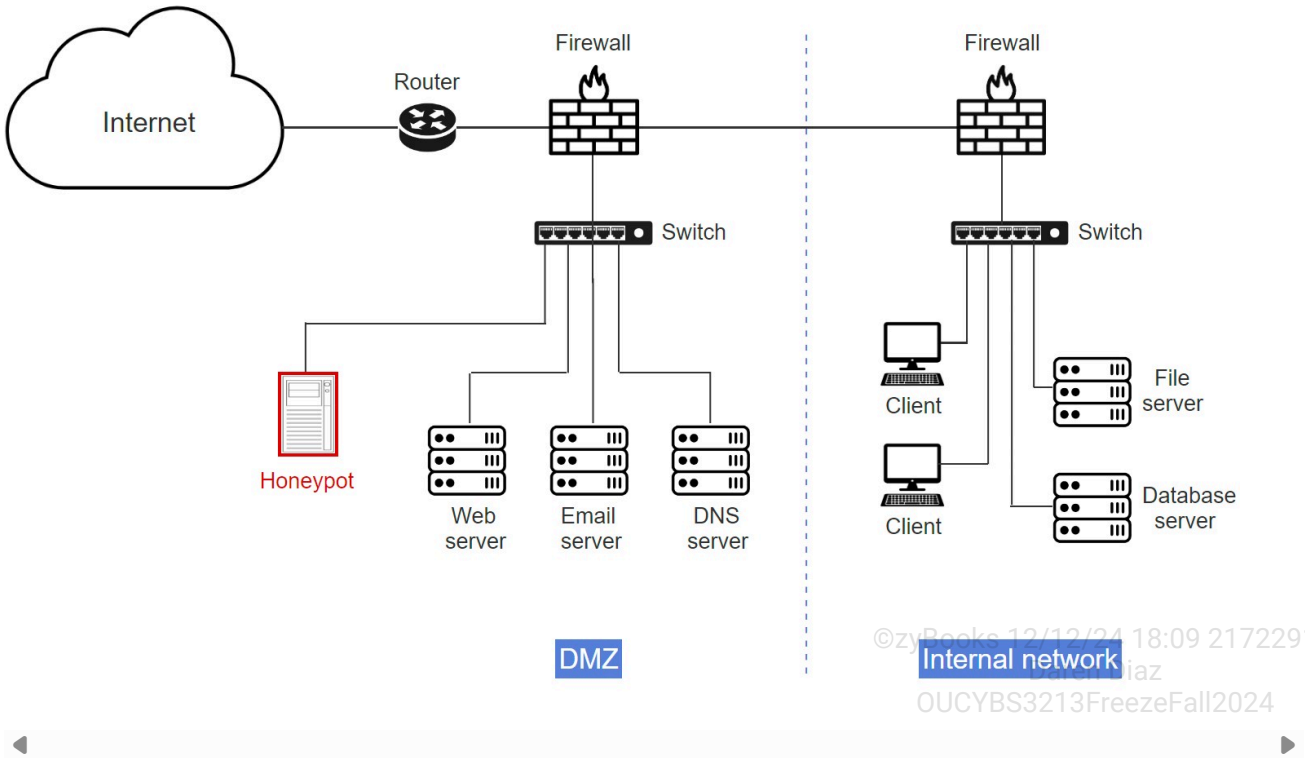
An organization's security can be improved by understanding an attacker's tactics, techniques, and procedures (TTPs). ***Tactics, techniques, and procedures (TTPs)*** are patterns of activities or methods associated with a specific attacker or group of attackers. Ex: Receiving phishing emails from a specific domain. TTP analysis can help security teams detect and mitigate attacks by understanding the way attackers operate.

A technique for understanding of TTPs is to set up decoys to lure attackers into a controlled environment for monitoring and behavioral analysis. Decoys include:

- A ***honeyfile*** is a fake file intentionally placed in a location to detect an attacker. Honeyfile names are chosen to attract more attackers. Ex: password.txt.
- A ***honeytoken*** is a decoy data element or resource placed within a system to detect an attacker. Ex: A fake database record designed to serve as an alert trigger when accessed.
- A ***honeypot*** is a security mechanism used for learning about attacks by luring attackers to an isolated and monitored environment. A honeypot is a single service or networked computer and is typically set up in a DMZ.
- A ***honeynet*** is a network of honeypots. A honeynet is set up with intentional vulnerabilities to ease access by potential attackers. Fake telemetry from network devices may also be generated on the network, providing additional targets for attackers.

Honeypots, honeynets, and honeyfiles may also be used to divert malicious traffic away from real systems, networks, and files.

Example 12.4.1: A honeypot is located in a DMZ (screened subnet) to protect the internal network.



PARTICIPATION ACTIVITY 12.4.3: TTP analysis.

1) Which of the following may be considered TTP?

- One failed login attempt from an internal IP
- Spam email send from an unknown domain
- A connection from a malicious IP

2) Why is a honeypot set up in a DMZ?

- Because a DMZ isolates the honeypot from the main network, minimizing the risk of a network breach
- Because a honeypot cannot be set up on an internal network
- Because attackers cannot access any devices on an internal network

3) Why would a large organization set up a honeynet that is not an exact replica of the organization's network?

- Because the complexity and expense of setting up an exact replica of a large network is prohibitive
- Because honeynets are not meant to replicate a real network
- Because setting up a honeypot is sufficient for understanding attackers' TTP profiles

4) Which option may be considered after analyzing a honeynet's traffic and discovering numerous outbound connections to IP 21.32.89.15?

- Add a perimeter firewall rule to block inbound connections from IP 21.32.89.15

- Add a perimeter firewall rule to allow outbound connection to IP 21.32.89.15
- Add a perimeter firewall rule to block outbound connections to IP 21.32.89.15

Honeypot

Honeypot comes from the world of espionage where a spy who uses a fake romantic relationship as a way to steal secrets is referred to as a honeypot. An enemy is compromised by a honeypot and convinced to hand over secrets the enemy knows. In computer security, a honeypot works in a similar way, baiting a trap for attackers to learn how the attackers operate.

12.5 Indicators of attack and compromise

Indicators of attack and compromise

An ***indicator of attack (IoA)*** is an abnormal or suspicious sign indicating potential security incidents or malicious activities. Detected through network traffic, system logs, or security data analysis, IoAs play an important role in threat detection and incident response. Tools like intrusion detection systems (IDS) and security information and event management (SIEM) systems utilize IoAs to identify and address security issues. Ex: Frequent failed login attempts over a short timeframe is an IoA indicating a brute-force attack.

While IoAs focus on identifying potential threats, indicators of compromise provide evidence of actual security incidents. An ***indicator of compromise (IoC)*** is a specific data point or artifact indicating a past security breach. An IoC facilitates threat investigation and response. Ex: Existence of a system file with a cryptographic hash matching a known malware's hash is an IoC indicating a system compromise.

Table 12.5.1: Comparison of IoA and IoC.

Aspect	IoA	IoC
Characteristics	Patterns indicating suspicious activity	Clear signs of security breaches
Purpose	Detect and respond to emerging or active threats	Investigate and respond to confirmed security breaches
Detection	Monitor network traffic and analyze system logs	Identify specific artifacts such as malware signatures
Focus	Identify potential attack methods and strategies	Determine the extent and impact of security breaches
Outcomes	Prevent potential incidents, minimize current threats	Understand and remediate security breaches, implement policy refinements
Examples	Unusual network traffic, repeated failed logins, firewall activity spikes	Unauthorized registry changes, unexpected user account creation, evidence of file tampering

PARTICIPATION ACTIVITY 12.5.1: IoAs and IoCs.

1) What is an IoA?

- A confirmed security breach report
- A sign of security incident or malicious activity
- A normal system log entry

2) Why are IoAs important for maintaining security of systems and data?

- IoAs prevent security incidents from occurring
- IoAs help identify potential security incidents and threats
- IoAs are used to create security policies

3) In what way are IoCs different from IoAs?

- IoCs are used to detect ongoing attacks, whereas IoAs identify past breaches
- IoCs are used to enforce policies, whereas IoAs are used to create policies
- IoCs are used to identify past security breaches, whereas IoAs focus on detecting ongoing or potential attacks

4) In what way are IoCs similar to IoAs?

- Both are mainly used for legal and compliance purposes
- Both are reactive measures taken after a security breach
- Both provide evidence about a security threat or incident

5) Which of the following scenarios is more indicative of an IoC rather than an IoA?

- Unusual outbound traffic patterns detected in network logs
- Discovery of unauthorized encryption of files indicating ransomware
- Alerts from a network intrusion prevention system about suspicious activity

6) What makes differentiating between normal activity, IoAs, and IoCs in cloud computing environments difficult?

- In cloud environments, IoAs and IoCs are typically identical
- Cloud environments only exhibit IoCs due to the cloud's inherent

security measures

- Rapid elasticity and resource scaling can mask IoAs, complicating the differentiation from normal activity and IoCs

Common IoAs

IoAs enable the identification of potential incidents and attackers' TTPs by monitoring behavior patterns and system anomalies, and preventing escalation. Ex: Frequent account lockouts due to repeated failed login attempts and concurrent sessions from multiple locations suggest brute force attacks and potential account compromise, and impossible travel alerts, triggered by logins from geographically distant locations in a short timeframe, indicate credential misuse.

IoAs include network defense alerts, resource usage anomalies, and unusual access patterns. Frequent content blocks by network defenses such as firewalls, along with spikes in CPU or memory usage, indicate attempts to access malicious sites, malware infections, or unauthorized processes. Signs like the inability to access data, unusual login times, or missing logs not only suggest ransomware or network problems but also covert exploration by attackers exploiting known vulnerabilities and attempting to hide malicious activities.

Table 12.5.2: Common IoAs.

IoA	Likely attack type	Impacted security principle
Account lockout	Brute force	Availability
Resource consumption	DDoS, malware	
Resource inaccessibility	Ransomware, network	
Published vulnerability exploitation	Targeted exploits	Availability, Integrity
Impossible travel	Credential fraud	Confidentiality
Concurrent session	Account hijacking	Confidentiality, Integrity
Blocked content	Malware, phishing	
Out-of-cycle logging	Unauthorized access	

Missing logs	Tampering	Integrity

PARTICIPATION ACTIVITY 12.5.2: Common IoAs.

Match each IoA with the corresponding behavior pattern.

How to use this tool

Concurrent session **Resource consumption** **Impossible travel**

Out-of-cycle/missing logs **Resource inaccessibility** **Account lockout**

Blocked content

	Repeated failed login attempts, possibly due to brute force attack
	Account compromise with multiple simultaneous logins
	Attempts to access malicious sites or download malware
	Compromised credentials used from different locations
	Malware infection or unauthorized resource-intensive processes
	Potential ransomware attack or network disruption
	Unauthorized access or deliberate deletion or alteration of logs to hide activities.

©zyBooks 12/12/24 18:09 2172291
Daren Diaz
OUCYBS3213FreezeFall2024

Reset

Common IoCs

IoCs serve as evidence that a security breach has occurred, indicating the presence of malicious activity within a network or system. IoCs range from malware signatures, which are detected through antivirus scans, to unusual outbound network traffic that suggests data exfiltration. Changes in file integrity can also indicate unauthorized access or tampering, while unauthorized user activities, such as access attempts at odd hours or the creation of unexplained user accounts, suggest a compromised network.

A ransomware attack is signaled by sudden encryption of files and the appearance of ransom notes, pointing to a system breach. Deleted or modified system logs indicate that malicious activities have likely occurred and efforts have been made to hide unauthorized access or other security breaches.

The early recognition of IoCs enables a rapid response with forensic analysis and containment measures, which is critical for minimizing the damage of a breach and preventing future security incidents.

Table 12.5.3: Common IoCs.

IoC	Likely attack type	Impacted security principle
Sudden file encryption	Ransomware	Availability
Unusual outbound traffic	Data exfiltration	Confidentiality
Unauthorized user activities	Unauthorized access	
Suspicious network connections	Command and control communication	
Unexplained user accounts	Privilege escalation	
Malware signatures	Malware infection	Integrity
File integrity changes	Tampering	
Deleted or modified system logs	Obfuscation, covering tracks	
Anomalous database read/write	SQL injection, database breach	

PARTICIPATION ACTIVITY 12.5.3: IoCs.

1) What is the primary purpose of IoCs?
   - To improve network speed
   - To serve as evidence of a security breach
   - To increase storage capacity

2) What is the significance of early detection of IoCs?
   - Allow for immediate system upgrades
   - Enable forensic analysis and containment efforts
   - Facilitate the deployment of new software

3) How do IoCs contribute to improving an organization's overall security posture over time?
   - By diverting resources away from proactive security measures
   - By enabling organizations to learn from incidents and enhance response strategies
   - By ensuring that security incidents no longer occur

4) What does the presence of IoCs in multiple layers of an organization's security infrastructure indicate?
   - The need for a simplified security strategy
   - Overreporting by security tools
   - A potential widespread and multi-stage attack

5) How should a sudden increase in cloud service costs be interpreted as an IoC?

- As an indication of improved service quality
- As potential unauthorized use of cloud resources
- As an expected outcome of company growth

6) In a cloud computing environment, what does the unexpected creation of multiple new administrative users signify?
   - Standard operational scaling to meet increased user demand
   - Potential unauthorized access or account compromise
   - Normal fluctuations in cloud service usage patterns

IoAs, IoCs, and OSINT

Integrating IoAs and IoCs with Open Source Intelligence (OSINT) and security sharing standards such as the Structured Threat Information eXpression (STIX) and the Trusted Automated Exchange of Indicator Information (TAXII) creates a comprehensive security strategy. Utilizing publicly available data through OSINT enhances threat detection and analysis by associating OSINT data with known IoAs/IoCs, thereby improving security teams' predictive and response capabilities.

12.6 Security automation and orchestration

Security automation and orchestration

Automating routine security operations strengthens an organization's defenses. Combining automated processes with orchestration improves efficiency and accuracy in essential security

functions including user and access management, resource management and configuration, and incident response and security operations.

User access management

Efficient security automation ensures that sensitive information remains accessible only to authorized personnel. Three components contribute to a secure access management system:

- Automated user provisioning leverages security automation to manage the lifecycle of user accounts across IT systems efficiently and encompasses the automated creation, update, and removal of user accounts, while adhering to the principle of least privilege.
- Automated security group management uses automation tools to facilitate the creation, modification, and management of security groups, thereby reducing manual overhead and the likelihood of errors.
- Automated access control for services ensures timely modification of user access rights and prevents unauthorized access, especially when a user's role changes or the access is no longer required.

Configuration enforcement is the automated application and verification of security settings across systems to ensure compliance with defined standards. Configuration enforcement in user access management ensures that all automated provisioning and access control activities are executed in compliance with an organization's security policies, safeguarding sensitive information by maintaining strict control over user roles and access privileges.

Table 12.6.1: Access management automation components.

Component	Objective	Process	Benefit
Automated user provisioning	Align access with user roles	Create, update, and remove accounts; assign specific permissions	Ensures secure, role-based access; minimizes unauthorized access risks
Automated security group management	Manage permissions efficiently	Group users by access needs; assign group-level permissions	Simplifies access control; enhances administrative efficiency
Automated access control	Ensure access matches current roles	Activate or deactivate access based on role changes	Maintains operational security; aligns access with current roles and needs

PARTICIPATION ACTIVITY 12.6.1: User and access management.

1) What is the purpose of user provisioning in security automation?

- To monitor user activities in real-time
- To encrypt user data stored on the server
- To create, update, and remove user accounts within various IT resources

2) What is the main reason for using security groups within access management frameworks?

- To ensure that individual user settings override group settings
- To facilitate easier and more efficient management of permissions for multiple users
- To prevent users from knowing who else is in the same group

3) What is the purpose of access control for services in security automation?

- To ensure that only authorized users have access to sensitive information
- To increase the number of services a user can access
- To track user activities outside of the organization

4) What are the potential outcomes associated with improperly managing the lifecycle of user access control?

- Enhanced system performance from unrestricted access to

resources and services

- Security breaches and unauthorized access due to outdated or unnecessary user access rights
- Reduced IT costs as a result of fewer access restrictions and simpler management

5) How does the principle of least privilege specifically influence the design and implementation of user provisioning systems?

- By requiring that all users be provided with administrative privileges to ensure equal access
- By designing systems to automatically provide the minimum necessary access to users based on users' role and responsibilities
- By encouraging the system to default to the highest possible security settings for all users

Resource management and configuration

Automation in resource management and configuration streamlines and secures the deployment of IT infrastructure. Organizations can automate the scaling of resources up or down based on demand, ensuring consistent security configurations across the organization's environments and reducing the risk of human error that can lead to security vulnerabilities. Two automated processes that enhance resource management and configuration exist:

- Automated resource provisioning securely allocates and manages computing resources, such as virtual machines, networks, and storage, following best practices to prevent unauthorized access and potential breaches.
- Automated guard rails are mechanisms designed to maintain compliance with established security policies, including checks that prevent misconfigurations and ensure that all configurations and deployments adhere to the organization's security standards.

Configuration enforcement within resource management maintains the integrity of security configurations across all automated processes. Configuration enforcement ensures that resource provisioning and guard rails are consistently applied in line with organizational security standards, thus reducing the risk of misconfigurations that could lead to vulnerabilities.

Table 12.6.2: Resource management and configuration automation components.

Component	Objective	Process	Benefit
Automated resource provisioning	Secure allocation and management of resources	Implements best practices to avoid unauthorized access	Enhances security and resource efficiency
Automated guard rails	Compliance with security policies	Conducts automatic checks against misconfigurations	Reduces risks and ensures policy adherence

PARTICIPATION ACTIVITY 12.6.2: Resource management and configuration.

1) What is the primary goal of automation in resource management and configuration?
   - To reduce the number of IT staff
   - To increase manual intervention
   - To streamline and secure IT infrastructure deployment

2) What does automated resource provisioning help prevent?
   - Authorized access
   - Unauthorized access and potential breaches
   - Efficient resource use

3) What is a benefit of automated guard rails?

- Automating the removal of security vulnerabilities
- Increasing the risk of misconfigurations
- Reducing risks and ensuring policy adherence

4) How do organizations benefit from automated resource scaling?

- By maintaining a fixed resource level at all times
- By ensuring consistent security configurations across environments
- By following outdated but functioning security configurations

5) What is the primary purpose of automated configuration management in cloud environments?

- To ensure consistent and error-free application of configurations across all cloud resources
- To increase the dependency on manual configuration tasks
- To decrease the speed of deployment and updates to improve security

6) How does cloud-based automated guard rail implementation benefit multi-cloud environments?

- By reducing the automation in resource provisioning
- By encouraging manual intervention in security settings
- By applying uniform security policies across different cloud services

Incident response and security operations

Integrating automation and orchestration into incident response and security operations enhances organizational security. Four automated processes provide a structured method for incident management:

- Automated ticket creation initiates incident response upon detection of potential security issues, ensuring timely documentation and reducing the chance of human error.
- Automated escalation procedures increase an incident's priority and ensure that the relevant management levels or specialized teams are informed.
- Automated security testing allows for the early detection and remediation of vulnerabilities, integrating security measures consistently throughout the development lifecycle.
- Automated system integration leverages application programming interfaces (APIs) to connect different security tools and systems, enabling seamless information sharing and unified response across the security infrastructure.

Table 12.6.3: Incident response and security operations automation components.

Component	Objective	Process	Benefit
Automated ticket creation	Incident documentation and tracking	Create tickets upon detecting incidents	Ensures organized tracking of security incidents
Automated escalation	Incident prioritization and resolution	Increase priority based on severity	Directs resources for effective incident resolution
Automated security testing	Early vulnerability detection	Integrate security tests into development	Prevents early-stage vulnerabilities
Automated system integration	Streamlined information sharing	Connect tools and systems via APIs	Improves efficiency in detecting and responding to incidents

PARTICIPATION ACTIVITY

12.6.3: Incident response and security operations.

1) What is the primary purpose of automated ticket creation in incident response?

   - To delay incident response
   - To ensure timely documentation
   - To ignore security incidents

2) How do automated escalation procedures enhance incident response?

   - By decreasing resource allocation
   - By reducing the incident's priority
   - By escalating incidents and alerting relevant teams

3) What is a key benefit of integrating automation and orchestration into incident response?

   - Enhances organizational security
   - Reduces the efficiency of security operations
   - Increases the chance of human error

4) How does automated integration and APIs improve security operations?

   - By disconnecting various security systems
   - By reducing the efficiency of responses
   - By improving efficiency in detecting and responding

5) How does the dynamic nature of cloud environments impact the effectiveness

of automated ticket creation for incident response?

- Leads to a decrease in ticket creation due to the static nature of cloud environments
- Causes an over-generation of tickets when improperly configured, requiring refined filtering and prioritization mechanisms
- Removes the need for ticket creation, as cloud environments resolve incidents automatically

6) Given the elasticity of cloud resources, how does automated escalation adapt to fluctuating workloads during a security incident?

- By maintaining a fixed response level, disregarding workload changes
- By automatically adjusting the prioritization and resource allocation based on the current workload and severity of the incident
- By reducing resource allocation during peak workloads to conserve resources

Playbooks

*A **playbook** is a predefined set of rules, procedures, and actions designed to automate responses to security events or implement configuration changes. Playbooks are designed based on best practices and organizational policies to ensure a consistent, rapid, and effective response to identified security events.*

By integrating with various security tools and technologies, playbooks facilitate a coordinated approach to detect, investigate, and remediate incidents across different environments and platforms. Such an orchestration of tasks not only enhances the

efficiency of security operations but also reduces the possibility of human error and the response time to threats, thereby minimizing potential damage.

12.7 Vulnerability management

Vulnerability response and remediation

Vulnerability management is the process of identifying, assessing, prioritizing, remediating, verifying, and reporting security vulnerabilities in an IT environment. Covering a wide range of activities, from scanning for unpatched software vulnerabilities to updating server firmware, the goal of vulnerability management is to ensure the confidentiality, integrity, and availability of information assets. Ex: Regularly patching an operating system reduces the risk of vulnerability exploitation, ensuring the availability of critical services provided by the operating system. Similarly, adopting cybersecurity insurance enhances financial resilience by covering post-breach recovery and damage expenses, while network segmentation limits an attacker's lateral movements by partitioning the network into isolated segments.

When direct remediation is not possible, compensating controls are implemented to mitigate vulnerability risk. Such measures ensure that, even if a system remains vulnerable, the overall exposure to potential exploits is minimized. Ex: Implementing enhanced monitoring to detect any unusual activities or potential threats early, and limiting the use of USB drives to prevent malware infections and data leakage.

Table 12.7.1: Vulnerability management phases.

Phase	Description	Example
Identification	Detecting vulnerabilities using tools and techniques	Scanning for network devices using default passwords
Assessment	Evaluating severity and impact using metrics like CVSS	Assessing the severity of an SQL injection vulnerability

Prioritization	Ranking vulnerabilities by severity, exploitability, and impact, guided by the organization's risk tolerance	Prioritizing a critical remote code execution flaw
Remediation	Fixing vulnerabilities through patches, configurations, or other measures	Patching a security flaw in an application
Verification	Ensuring vulnerabilities are resolved without introducing new issues	Confirming a patch's effectiveness by rescanning
Reporting	Documenting the process, findings, actions taken, and outcomes for accountability and future planning	Generating a report for an audit compliance

PARTICIPATION ACTIVITY 12.7.1: Vulnerability response and remediation.

1) What is the primary goal of vulnerability management in an IT environment?
   - To reduce IT operational costs
   - To increase system performance
   - To ensure the CIA of information assets

2) Why might an organization adopt cybersecurity insurance to manage certain vulnerabilities?
   - To reduce the IT staff's workload
   - To segment the network into isolated sections
   - To cover post-breach recovery and damage expenses

3) What is a compensating control in vulnerability management?
   - A software update that introduces new features

- A method for increasing network speed and efficiency
- Additional monitoring when direct remediation is not possible

4) Which activity is an example of the identification phase in vulnerability management?

- Implementing a firewall
- Detecting outdated encryption protocols
- Adopting cybersecurity insurance

5) How does vulnerability management in the cloud differ from a traditional on-premise environment?

- Is less complex and requires minimal effort from the client's side
- Relies solely on cloud service providers for all security measures
- Requires coordination between the cloud service provider and client to secure a cloud-based resource

Exemptions and exceptions

In situations where standard security protocols are inadequate, two practices ensure ongoing protection in vulnerability management:

- An ***exemption*** is a vulnerability management practice that grants formal permission to bypass specific security policies or controls when compliance is not possible or practical, without compromising overall security posture. Ex: Allowing an outdated, yet critical application to operate without the latest security patch because the patch would disable essential functionality.
- An ***exception*** is a vulnerability management practice that involves a temporary deviation from established security policies, granted under specific conditions and for a limited time, while

seeking a permanent solution. Ex: Temporarily permitting less secure access methods due to an emergency, with a deadline to revert to standard security protocols.

Exemptions and exceptions ensure operational continuity, aligning immediate requirements with the overall goal of maintaining security when standard practices are not feasible.

Table 12.7.2: Comparison of exemption and exception measures.

Aspect	Exemption	Exception
Purpose	When compliance is not possible or practical	Addressing specific conditions or emergencies
Duration	Typically long-term	Temporary, with a defined timeframe
Compliance requirement	Not feasible or practical	Actively working on a lasting compliance resolution
Potential risks	Increased vulnerability to security breaches due to lack of updates or controls	Temporary weakening of security posture, potential for exploitation during the exception period
Example	Allowing a critical medical device to operate without the latest security update due to compatibility issues	Temporarily permitting access to a critical system from untrusted locations during a natural disaster

PARTICIPATION ACTIVITY 12.7.2: Exemptions and exceptions.

Select the appropriate vulnerability management practice for each scenario.

1) During a migration process to a cloud-native application, certain security measures are relaxed to ensure a smooth transition.

   - Exemption
   - Exception

2) An organization temporarily suspends an intrusion detection system (IDS) to

facilitate system upgrades without triggering false alerts.

- ○ Exemption
- ○ Exception

3) An organization faces challenges in implementing standard security controls for a legacy system due to the system's outdated architecture.

- ○ Exemption
- ○ Exception

4) In response to a critical security incident, an organization suspends certain access controls to allow incident response teams unrestricted access to affected systems.

- ○ Exemption
- ○ Exception

5) During a security audit, the audit team discovers that a critical cloud service cannot be fully compliant with existing security policies due to inherent limitations in the service's architecture.

- ○ Exemption
- ○ Exception

Validation of vulnerability remediation

Following remediation efforts, validation ensures the effectiveness of control measures through rescanning, auditing, and verification processes. Rescanning with vulnerability scanners checks if patches have been applied correctly and vulnerabilities addressed. Ex: A post-update firewall scan, conducted after a security breach, confirms the closure of vulnerable ports, thus validating the firewall's improved security.

Audits, both internal and external, review adherence to security policies and the effectiveness of controls. Verification, often through testing or review, confirms that the remediation actions have mitigated the identified risks. Ex: Following the implementation of stricter password policies, an audit checks user accounts for compliance, verifying that all accounts meet the new security standards.

Reporting documents the discovery, remediation, and validation of vulnerabilities. Reports provide insights into the security posture of an organization, enabling stakeholders to make informed

decisions about future security investments and policies. Ex: A report that details how a specific vulnerability was patched, verified through rescanning, and how the process informs future security measures.

Table 12.7.3: Documentation role in vulnerability management phases.

Phase	Role	Example
Identification	Document initial discovery of vulnerabilities, tools used, and potential impact	Discovered high severity SQL injection vulnerability in application X with scanner Y
Assessment	Evaluate the vulnerabilities for severity, impact, and exploitability	Assessed SQL injection in application X as critical due to data breach risk
Prioritization	Rank vulnerabilities based on severity, impact, and urgency for resolution	Prioritized SQL injection in application X for immediate remediation due to high business impact
Remediation	Record actions taken for each vulnerability, including patches and configurations	Patched SQL injection in application X and adjusted configurations
Verification	Detail results from rescanning and audits to verify remediation effectiveness	Post-patch scan confirms no SQL injection vulnerability in application X
Reporting	Summarize the vulnerability management cycle, insights, trends, and recommendations	Quarterly report: 15 vulnerabilities found, 12 fixed Highlight: Effective SQL injection patch

PARTICIPATION ACTIVITY 12.7.3: Validation of vulnerability remediation.

1) What is the purpose of the reporting role in vulnerability management?

   - Documenting initial discovery of vulnerabilities

- Evaluating the vulnerabilities for severity, impact, and exploitability
- Summarizing the vulnerability management cycle and providing insights

2) What role does documentation play in the vulnerability management identification phase?
   - Tracking patch implementation
   - Evaluating vulnerability severity
   - Documenting initial discovery of vulnerabilities

3) What exemplifies the verification role in vulnerability management for microservices deployed in a cloud environment?
   - Conducting regular security assessments
   - Recording actions taken for each vulnerability
   - Confirming remediation effectiveness through rescanning

4) In vulnerability management, what approach involves proactively monitoring and analyzing network traffic to detect and respond to potential security threats in real-time?
   - Threat hunting
   - Patch management
   - Vulnerability scanning

5) In vulnerability management for cloud-based services, what practice focuses on ongoing assessment and improving security measures to align with evolving threats?

- Security auditing
- Continuous monitoring
- Threat intelligence analysis

The dark web

*The **dark web** is an encrypted part of the internet not indexed by conventional search engines and accessible only through specialized tools. The dark web enables individuals to communicate, share, and operate with a significant level of anonymity, facilitating not just illicit transactions and services, but also the exchange of cybersecurity threats such as exploitation tools and zero-day vulnerabilities.*

The monitoring of dark web activities helps an organization detect threats at an early stage, guiding prioritization and remediation of vulnerabilities. Anticipating such threats improves the organization's vulnerability management strategies, transitioning from reactive to proactive measures, thereby ensuring potential threats are addressed before any harm occurs.

12.8 LAB: Vulnerability assessment with OpenVAS (Walkthrough)

IT-Labs are not printable at this time.

12.9 LAB: Log management in Windows and Linux (Walkthrough)

IT-Labs are not printable at this time.