Week 1: Introduction to Enterprise Security Threats

1. Common Security Threats

In this section, we will research and document common security threats that can affect enterprises. Here's a brief overview of each:

Malware

Malware (short for "malicious software") refers to any software intentionally designed to cause damage to a computer, server, client, or computer network. There are several types of malware:

- **Viruses**: Programs that replicate themselves and spread to other files.
- **Trojans**: Software that pretends to be legitimate but has malicious intent.
- **Worms**: Similar to viruses, but they can spread independently across networks.
- **Spyware**: Software that collects information about a user without their knowledge.
- **Adware**: Software that automatically displays or downloads unwanted ads.

Phishing

Phishing is a type of cyberattack that uses fake emails, websites, or messages to deceive users into revealing sensitive information, such as usernames, passwords, or credit card numbers. Phishing can come in several forms:

- **Spear phishing**: A targeted attack, often aimed at individuals or companies.
- **Whaling**: A form of phishing targeting high-profile individuals like executives or leaders.
- **Vishing**: Voice phishing, typically done over the phone.

Ransomware

Ransomware is a type of malware that locks or encrypts a user's data, and the attacker demands payment (usually in cryptocurrency) in exchange for restoring access. Common examples include **WannaCry** and **Petya**.

DDoS (Distributed Denial of Service)

DDoS attacks involve overwhelming a server, service, or network with a flood of internet traffic to render the target unusable. These attacks are carried out using a network of infected devices, called a botnet, which causes a massive surge in traffic.

Insider Threats

Insider threats occur when individuals within an organization (employees, contractors, etc.) intentionally or unintentionally cause harm to the organization. This harm could be through data theft, sabotage, or accidental leaks.

---

2. Basic Security Frameworks

In this section, we will explore and understand some basic security frameworks that guide organizations in protecting their information assets.

NIST (National Institute of Standards and Technology)

NIST provides a comprehensive framework for improving cybersecurity across critical infrastructure. Its **Cybersecurity Framework (CSF)** helps organizations identify, assess, and manage cybersecurity risks. The CSF is made up of five core functions:

- **Identify**: Develop an understanding of the organization's assets, risks, and resources.
- **Protect**: Implement safeguards to prevent cyber threats.
- **Detect**: Identify cybersecurity events in real-time.
- **Respond**: Develop strategies to mitigate and respond to incidents.
- **Recover**: Plan for recovery and restoration of any lost or damaged assets.

NIST also provides several guidelines, including those on risk management (NIST SP 800-53) and secure software development.

ISO 27001

ISO 27001 is an international standard for managing information security. The focus of the ISO 27001 standard is on establishing, implementing, operating, monitoring, reviewing, and maintaining an Information Security Management System (ISMS). Key components include:

- **Information Security Policy**: A strategic document to set the direction for security within an organization.
- **Risk Assessment and Treatment**: Identifying risks to information security and implementing measures to mitigate them.
- **Control Objectives**: Specific security measures and practices that need to be in place, such as access controls and physical security.

CIS Controls

The **Center for Internet Security (CIS)** provides a set of **CIS Controls** that organizations can implement to protect against the most common and damaging cybersecurity threats. These controls are divided into three categories:

1. **Basic Controls**: Such as inventory and control of hardware/software, continuous vulnerability management, and controlled use of administrative privileges.
2. **Foundational Controls**: These include email and web browser protections, malware defenses, and data recovery capabilities.
3. **Organizational Controls**: These focus on incident response and security awareness training.

These frameworks and controls serve as foundational tools in designing and implementing an effective cybersecurity strategy.

3. Set Up a Virtual Lab (Kali Linux, Windows Server, Ubuntu, Metasploitable)

Setting up a virtual lab with **Parrot OS**, **Windows Server**, **Ubuntu**, and **Metasploitable** is a great idea for cybersecurity, penetration testing, or networking practice. Here's a step-by-step guide to get you started:

Prerequisites

1. **A powerful host machine**
   - At least 16GB RAM, 100GB+ disk space, and a modern CPU.
2. **Virtualization software**
   - VMware Workstation Player (Free for personal use) or
   - VirtualBox (Free and open source)
3. **ISO/OVA files**
   - Parrot OS: https://www.parrotsec.org/download/
   - Windows Server (eval): https://www.microsoft.com/en-us/evalcenter/
   - Ubuntu: https://ubuntu.com/download
   - Metasploitable 2: https://sourceforge.net/projects/metasploitable/

Steps to Set Up the Lab

1. Install VirtualBox or VMware

- Download and install your preferred virtualization platform.

2. Create Individual Virtual Machines

◆ **Parrot OS**

- Use Parrot Security Edition (ideal for pentesting).
- Assign ~2 CPUs, 2-4GB RAM, 20GB+ disk.
- Install from ISO like a regular Linux distro.

◆ **Ubuntu**

- Use the Desktop or Server version depending on your goals.
- Assign ~1-2 CPUs, 2GB RAM, 20GB+ disk.

◆ **Windows Server**

- Download the evaluation ISO (e.g., Server 2019 or 2022).
- Assign ~2 CPUs, 4GB+ RAM, 40GB+ disk.
- Activate or use trial license for 180 days.

◆ **Metasploitable 2**

- Download the pre-configured VM.
- Import directly into VirtualBox/VMware (no install needed).
- Keep it isolated or on a private network—**it's intentionally vulnerable.**

Networking Setup

To allow interaction between machines:

Option 1: Host-only Adapter (Recommended for safety)

- All VMs can talk to each other but not the internet.
- Great for internal testing.

Option 2: Internal Network

- Like Host-only but restricted to VMs (no access to host machine).

Option 3: NAT or Bridged (Use with caution)

- VMs can access internet or local network (can be dangerous with Metasploitable).

Optional Tools and Configuration

- **Install guest additions** on Linux VMs for better performance.
- **Snapshots** before you test tools or exploits.
- Set **static IPs** or configure DHCP server on Windows Server.
- Install tools like Wireshark, Burp Suite, or Nessus on Parrot/Ubuntu.

Test Your Lab

- Ping between machines.
- Run nmap from Parrot to Metasploitable.
- Test shared folders or RDP to Windows Server.
- Launch Metasploit against Metasploitable.

Week 2: Logging & Monitoring (Detailed Plan)

Objective

- Install and configure a SIEM tool (Splunk, Wazuh, or ELK Stack).
- Enable logging on Windows (Sysmon, Event Viewer) and Linux (Syslog).
- Collect and analyze system logs to identify security events.

---

1. Install and Configure SIEM Tools

Choose one SIEM tool to install in your virtual lab (Kali Linux, Ubuntu, or Windows Server). Below are setup steps for each.

- **Splunk** (Recommended for Beginners)
  - **Overview**: Centralized log management with powerful search and visualization.
  - **Installation**:
    1. Download Splunk Enterprise (free tier, 500MB/day) from https://www.splunk.com/en_us/download.html.
    2. On Ubuntu:
    3. On Windows: Run the .msi installer, follow the wizard, and start Splunk.
    4. Access Splunk at http://localhost:8000 (default credentials: admin/changeme).
  - **Configuration**:
    - Add a data input (e.g., monitor a log file like /var/log/syslog).
    - Create a basic dashboard (e.g., display login attempts).
  - **Task**: Ingest a sample log file (e.g., /var/log/auth.log) and verify it appears in Splunk.
- **Wazuh** (Open-Source, Host-Focused)
  - **Overview**: Combines SIEM with host intrusion detection and compliance monitoring.
  - **Installation** (Ubuntu recommended):
    0. Install Wazuh all-in-one deployment:
    1. Access Wazuh dashboard at https://<server-ip>:5601 (use generated credentials).
  - **Configuration**:
    - Deploy Wazuh agents on Windows/Linux VMs (download from Wazuh manager).
    - Enable log collection for /var/log/syslog (Linux) or Event Viewer (Windows).
  - **Task**: Add one agent (e.g., Ubuntu VM) and confirm logs are received.
- **ELK Stack** (Advanced, Open-Source)
  - **Overview**: Elasticsearch (storage), Logstash (processing), Kibana (visualization).
  - **Installation** (Ubuntu):
    0. Install Elasticsearch:
    1. Install Logstash and Kibana similarly.
    2. Start services:
    3. Access Kibana at http://localhost:5601.
  - **Configuration**:
    - Configure Logstash to ingest logs (e.g., /var/log/syslog).
    - Create a Kibana index pattern for log data.
  - **Task**: Visualize log counts in Kibana (e.g., graph of log entries over time).
- **Recommendation**: Start with Splunk for ease of use, or Wazuh for open-source flexibility. ELK is complex but powerful for advanced users.

---

2. Enable Logging on Windows/Linux Systems

Set up logging to capture system and security events for SIEM analysis.

- **Windows Logging**
  - **Sysmon** (System Monitor):
    1. Download Sysmon from Microsoft Sysinternals: https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon.
    2. Install with a configuration file (e.g., SwiftOnSecurity's config for detailed logging): powershell
    3. Verify logs in **Event Viewer** (Applications and Services Logs > Microsoft > Windows > Sysmon).
  - **Event Viewer**:
    - Enable security logging (e.g., audit logon events): powershell
    - Focus on Event IDs: 4624 (successful login), 4625 (failed login), 4688 (process creation).
  - **Task**: Export 10 Sysmon logs and 5 security logs to a .evtx file.
- **Linux Logging**
  - **Syslog**:
    0. Verify rsyslog is installed:
    1. Configure /etc/rsyslog.conf to log to /var/log/syslog or /var/log/auth.log.
    2. Test logging:
  - **Task**: Collect 10 logs from /var/log/auth.log (e.g., SSH login attempts).

---

3. Collect System Logs and Analyze Security Events

- **Collect Logs**:
  - **Windows**: Forward Event Viewer logs to SIEM:
    - Splunk: Use Splunk Universal Forwarder to send logs.
    - Wazuh: Wazuh agent automatically collects Windows events.
  - **Linux**: Forward Syslog to SIEM:
    - Splunk: Monitor /var/log/syslog via Splunk input.
    - Wazuh: Configure agent to send Syslog.
  - **Task**: Ensure at least 100 log entries are ingested into your SIEM from both Windows and Linux.
- **Analyze Security Events**:
  - **Search for Events**:
    - Splunk: Use SPL (Search Processing Language), e.g., index=main sourcetype=syslog failed to find failed logins.
    - Wazuh: Use dashboard filters (e.g., rule ID 533 for brute-force attempts).
    - ELK: Query in Kibana, e.g., event.code:4625 for Windows failed logins.
  - **Common Events to Identify**:
    - Failed login attempts (Event ID 4625 on Windows, sshd: Failed in Linux).

- Suspicious process creation (Event ID 4688, e.g., cmd.exe spawning unusual processes).
- Unauthorized access attempts (e.g., Syslog entries for sudo misuse).

- **Task**: Identify and document 3 suspicious events (e.g., multiple failed logins from one IP). Include:
  - Log details (timestamp, source, event description).
  - Potential threat (e.g., brute-force attack).
  - Mitigation (e.g., block IP in firewall).

Deliverables

1. **SIEM Setup**: Screenshot of SIEM dashboard showing ingested logs.
2. **Logging Configuration**:
   - Windows: 10 Sysmon logs and 5 Event Viewer logs.
   - Linux: 10 Syslog entries from /var/log/auth.log.
3. **Analysis Report**: Document 3 suspicious events with details and mitigations (1-2 pages).

Tips for Success

- **Test Connectivity**: Ensure VMs can communicate with the SIEM server (e.g., ping <siem-ip>).
- **Sample Logs**: Generate test events (e.g., failed SSH logins: ssh invalid@<ip>).
- **Documentation**: Record all commands and configurations in a lab journal.
- **Resources**:
  - Splunk Tutorials: https://www.splunk.com/en_us/training.html
  - Wazuh Docs: https://documentation.wazuh.com
  - Sysmon Config: https://github.com/SwiftOnSecurity/sysmon-config
  - TryHackMe SIEM Labs: For hands-on practice.

Week 3: Firewall & Endpoint Security

1. Configure Firewalls

Windows Defender Firewall (Windows)

- **Access Firewall Settings**:
  - Open Control Panel > System and Security > Windows Defender Firewall > Advanced Settings.
- **Create Rules**:
  - **Inbound Rules**: Block untrusted traffic (e.g., deny all except specific ports like 80, 443).
    - Right-click Inbound Rules > New Rule > Port > TCP/UDP > Specific Ports > Block the connection.
  - **Outbound Rules**: Allow only necessary outbound connections (e.g., DNS, HTTP/HTTPS).

    - Similar process: New Rule > Port/Program > Allow connection for trusted apps.
- **Enable Logging**:
    - In Advanced Settings, click "Properties" > Customize Logging > Enable "Log dropped packets" for monitoring.
- **Test Configuration**:
    - Use ping or telnet to verify blocked/allowed connections.

- **Install iptables** (installed)
- **Basic Configuration**:
    - Flush existing rules:
    - Set default policies (block all incoming, allow outgoing):
    - Allow specific traffic (e.g., SSH, HTTP):
    - Save rules:
- **Enable Persistence**:
    - Install iptables-persistent:
    - Rules auto-load on boot.
- **Test Rules**:
    - Use nmap or curl to verify open/closed ports.

---

2. Set Up IDS/IPS (Snort or Suricata)

Snort (Open-Source IDS/IPS)

- **Install Snort** (Ubuntu/Debian):
- **Configure Snort**:
    - Edit /etc/snort/snort.conf:
        - Set network variables (e.g., HOME_NET to your LAN, like 192.168.1.0/24).
        - Enable rules by uncommenting relevant .rules files (e.g., local.rules).
    - Create custom rules in /etc/snort/rules/local.rules: plaintext
- **Run Snort**:
    - IDS mode (monitor traffic):
    - IPS mode (block traffic, requires inline setup):
- **Update Rules**:
    - Use PulledPork or manually download community rules from snort.org.
- **Test**:
    - Generate traffic (e.g., HTTP requests) and check logs (/var/log/snort/alert).

- **Install Suricata**:
- **Configure Suricata**:
    - Edit /etc/suricata/suricata.yaml:
        - Set HOME_NET and enable rules.
    - Update rules:
- **Run Suricata**:
    - IDS mode:
    - IPS mode (with nfqueue):
- **Test**:
    - Monitor alerts in /var/log/suricata/fast.log.

3. Implement Basic Endpoint Security

Antivirus

- **Windows**:
  - **Windows Defender** (built-in):
    - Ensure real-time protection is enabled: Settings > Update & Security > Windows Security > Virus & Threat Protection.
    - Schedule full scans: Windows Security > Virus & Threat Protection > Manage Settings > Automatic Sample Submission.
  - **Third-Party Option**: Install free antivirus like Avast or Malwarebytes for additional layers.
- **Linux**:
  - Install ClamAV:
  - Update database:
  - Run scan:

Endpoint Detection and Response (EDR)

- **Microsoft Defender for Endpoint**:
  - **Setup** (requires Microsoft 365 subscription):
    - Enroll devices via Microsoft Endpoint Manager.
    - Enable cloud-delivered protection and automatic sample submission.
  - **Features**:
    - Real-time threat detection, behavioral analysis, and automated remediation.
  - **Monitoring**:
    - Use the Microsoft 365 Defender portal to view alerts and incidents.
- **CrowdStrike Falcon** (Enterprise):
  - **Setup**:
    - Deploy Falcon agent via CrowdStrike console (requires licensing).
    - Configure policies for threat prevention, USB device control, and firewall management.
  - **Features**:
    - AI-driven threat hunting, ransomware protection, and incident response.
  - **Monitoring**:
    - Use Falcon dashboard for real-time alerts and forensics.
- **Open-Source Alternative**: Install Wazuh (EDR-like capabilities):
  - Configure to report to a Wazuh server for centralized monitoring.

Testing and Validation

- **Firewall**: Use nmap to scan for open ports and verify rules.
- **IDS/IPS**: Simulate attacks (e.g., hping3 for DoS) and check logs for alerts.
- **Endpoint Security**: Test with EICAR test file (www.eicar.org) to verify antivirus/EDR detection.

Week 4: Threat Detection Basics

1. Learn How to Write Detection Rules

- **Purpose**: Identify malicious files based on patterns (strings, hex, regex).
- **Install YARA** (Ubuntu/Debian):
- **Write a YARA Rule**:
  - Create a file (e.g., malware.yar): yara
- **Test the Rule**:
  - Scan a file or directory:
- **Resources**: Use YARA rule repositories (e.g., GitHub's yara-rules) for examples.

- **Purpose**: Create platform-agnostic detection rules for SIEMs.
- **Setup**:
  - Clone Sigma repository:
  - Install sigmac (Sigma converter):
- **Write a Sigma Rule**:
  - Create a YAML file (e.g., brute_force.yml): yaml
- **Convert to SIEM Format**:
  - Convert to Splunk, Elastic, etc.:
- **Resources**: Explore Sigma's rule repository for templates.

- **Purpose**: Detect malicious network traffic.
- **Write a Suricata Rule**:
  - Edit /etc/suricata/rules/local.rules: plaintext
  - Explanation:
    - alert: Trigger an alert.
    - tcp any any -> $HOME_NET 22: Monitor TCP traffic to port 22.
    - threshold: Alert after 5 attempts in 60 seconds.
- **Reload Rules**: bash
- **Test**: Generate SSH login attempts and check /var/log/suricata/fast.log.

---

2. Simulate Attacks and Capture Logs

- **Setup Target**:
  - Ensure an SSH server is running (sudo apt install openssh-server).
  - Enable logging: Edit /etc/ssh/sshd_config to set LogLevel VERBOSE.
  - Restart SSH: sudo systemctl restart sshd.
- **Simulate Attack**:
  - Use hydra to simulate brute force: bash
  - Alternatively, script with Python: python
- **Capture Logs**:
  - Check /var/log/auth.log (Linux) or Event Viewer (Windows) for failed login attempts.
  - Example log: sshd: Failed password for user from <attacker_ip>.

- **Safe Malware Simulation**:
  - Use EICAR test file (not harmful): bash
  - Alternatively, create a benign script mimicking malware behavior: bash

- **Capture Logs**:
  - Use antivirus (e.g., ClamAV, Windows Defender) to scan and log detection:
  - Check SIEM or EDR logs (see below).
  - Monitor system logs: /var/log/syslog or Windows Event Viewer (Security/Application).

---

3. Generate Alerts Using SIEM Tools

Open-Source SIEM: Wazuh

- **Install Wazuh**:
  - Deploy Wazuh all-in-one (manager, agent, dashboard): bash
  - Install Wazuh dashboard (follow official docs for Elastic/Kibana integration).
- **Configure Wazuh**:
  - Enroll agents on endpoints: bash
  - Add custom rules in /var/ossec/etc/rules/local_rules.xml: xml
- **Generate Alerts**:
  - Run brute force or malware simulation.
  - View alerts in Wazuh dashboard (default: https://<wazuh_ip>:5601).
  - Filter by rule ID, source IP, or event type.

Alternative: Elastic SIEM

- **Setup**:
  - Install Elastic Stack (Elasticsearch, Kibana, Filebeat): bash
  - Configure Filebeat to collect logs (e.g., /var/log/auth.log).
- **Ingest Logs**:
  - Use Filebeat to forward logs to Elasticsearch.
  - Create a Sigma rule (from above) and convert to Elastic query.
- **Generate Alerts**:
  - In Kibana, create a detection rule:
    - Security > Rules > Create New Rule > EQL or Query-based.
    - Example query for brute force: plaintext
  - Simulate attack and monitor alerts in Kibana's Security app.

Testing and Validation

- **YARA**: Scan test files (e.g., EICAR) and verify rule triggers.
- **Sigma**: Convert rules to SIEM format and ensure alerts appear in Wazuh/Elastic.
- **Suricata**: Check fast.log for network-based alerts during simulated attacks.
- **SIEM**: Confirm brute force/malware alerts in Wazuh or Elastic dashboards.