Lecture 1: Course Introduction & Overview

- **Concepts & Details:**
  - **Course Purpose & Structure:**
    The professor introduced computer security as a broad field covering cryptography, network security, operating system hardening, and secure coding. He explained that the course is hands-on and project-oriented. Emphasis was placed on learning both how to defend systems (by hardening software, enforcing least privilege, and using proper security protocols) and how to think like an attacker.
  - **Expectations & Integrity:**
    Academic integrity is critical; you must produce your own work (no AI-generated answers) and participate actively. Attendance and active engagement are stressed since many practical exercises (like working on a Linux VM) require you to be present.
  - **Exam & Assignment Format:**
    The course will feature weekly quizzes (multiple choice/true-false) and two major exams (one covering midterm topics and a comprehensive final). The professor mentioned that the exam questions will focus on practical aspects and key security principles.

Lecture 2: Linux Command Line & File Management

- **Concepts & Details:**
  - **Navigating Linux:**
    The lecture covered the basics of the Linux CLI. The professor explained that every file system is organized as directories (not “folders” as in Windows) starting at the root `/`. Commands such as `pwd` (to show your current directory), `ls` (to list directory contents), `cd` (to change directories), and `cat` (to display file contents) were demonstrated.
  - **Understanding Paths:**
    Special symbols were explained:
    - `.` means “current directory”
    - `..` means “parent directory”
    - `~` stands for the current user’s home directory
    - `/` is the root of the file system
  - **Piping & Redirection:**
    The professor showed how you can send the output of one command as input to another (using the pipe `|`) and redirect output to a file (using `>` or `>>`). For example, capturing process lists with `ps -ef > process_log` was discussed.
  - **File Permissions:**
    A detailed explanation was given about Unix file permissions (read, write, execute for user, group, and others). Numeric modes were described (e.g., 700 gives full permission to the owner and none to others). The command `chmod` was introduced to modify these permissions.

- **Key Commands:**

```
pwd
ls -l
```

```
cat filename
cd /path/to/directory
grep "search-term" filename
chmod 700 filename
```

- **Exam–Relevant Explanation:**
  You should be able to describe how each command works, why redirection and piping are useful, and how file permissions enforce security by restricting who can read or execute a file.

Lecture 3: Cryptography – Introduction & PBKDF2 Key Derivation

- **Concepts & Details:**
  - **Encryption vs. Encoding vs. Hashing:**
    The professor clarified that encryption is a process to transform plaintext into an unreadable format using a key, while encoding (such as Base64) is merely a way to represent binary data in text form. Hashing is a one-way function used for verifying integrity (it cannot be “decrypted”).
  - **Symmetric vs. Asymmetric Encryption:**
    - *Symmetric encryption* uses a single shared key for both encryption and decryption; it is fast and uses small key sizes (commonly 128 or 256 bits).
    - *Asymmetric encryption* uses a key pair—a public key for encryption and a private key for decryption. This method is slower and requires larger key sizes (usually at least 2048 bits).
  - **Password-Based Key Derivation & Salting:**
    To generate a secure key from a password, the professor introduced PBKDF2. By using a secure random salt (a unique random value) and multiple iterations (e.g., 1024 iterations), PBKDF2 makes brute-force attacks and rainbow table attacks much more difficult.

- **Key Code Example (Java PBKDF2):**

```
// Generate a random salt
SecureRandom random = new SecureRandom();
byte[] salt = new byte[16];
random.nextBytes(salt);
String saltString = Base64.getEncoder().encodeToString(salt);

// Obtain the user’s password and create a key specification
Scanner scanner = new Scanner(System.in);
System.out.print("Enter password: ");
String keyString = scanner.nextLine();
KeySpec spec = new PBEKeySpec(keyString.toCharArray(), salt, 1024,
128);

// Generate the secret key using PBKDF2 with HMAC-SHA256
SecretKeyFactory factory =
SecretKeyFactory.getInstance("PBKDF2WithHmacSHA256");
SecretKey privateKey = factory.generateSecret(spec);
```

- **Exam–Relevant Explanation:**
  Be ready to explain why a unique salt is necessary per user, how PBKDF2 strengthens key derivation, and the practical differences between symmetric and asymmetric encryption.

Lecture 4: Cryptography – Encryption/Decryption Process & Base64 Encoding

- **Concepts & Details:**
  - **Encryption Cycle:**
    The lecture continued with the encryption/decryption process. After generating a key with PBKDF2, the professor demonstrated how to encrypt a plaintext message using the AES algorithm and then how to convert the binary output to a text format using Base64 encoding.
  - **Decryption Process:**
    The reverse process was explained: first Base64-decoding the encrypted message back into bytes, then initializing the cipher in decryption mode with the same key, and finally converting the decrypted bytes back into a readable string.

- **Key Code Snippet (Decryption Portion):**

```
// Initialize cipher for decryption
Cipher cipher = Cipher.getInstance("AES");
SecretKeySpec keySpec = new SecretKeySpec(privateKey.getEncoded(),
"AES");
cipher.init(Cipher.DECRYPT_MODE, keySpec);

// Decode Base64 and decrypt
byte[] encryptedData = Base64.getDecoder().decode(message);
byte[] decryptedData = cipher.doFinal(encryptedData);
String decryptedMessage = new String(decryptedData);
System.out.println("Message is " + decryptedMessage);
```

- **Exam–Relevant Explanation:**
  You must know the steps involved in the encryption–decryption cycle and why Base64 is used for converting binary data into a transportable string format.

Lecture 5: Operating System Security & SSH Hardening

- **Concepts & Details:**
  - **System Updates & Software Management:**
    The professor explained that keeping the system updated is vital. Commands like `sudo apt update` (to refresh package lists) and `sudo apt upgrade` (to apply updates) are essential practices to reduce vulnerabilities.
  - **Service Management:**
    Using `systemctl` to check the status of services, restart them, or disable them at boot was

demonstrated. This ensures that unnecessary services are not left running, reducing the attack surface.

- **SSH Security:**
  Detailed steps were provided on hardening SSH access:
  - Generate an SSH key pair using `ssh-keygen` instead of relying on password-based logins.
  - Transfer the public key to the server with `ssh-copy-id`.
  - Modify the SSH configuration file (`/etc/ssh/sshd_config`) to disable password authentication (set `PasswordAuthentication no`) and disable remote root login (`PermitRootLogin no`).
  - Optionally, use PAM (Pluggable Authentication Modules) and restrict access using `host.allow` files.
- **Principle of Least Privilege:**
  The professor demonstrated creating a non-privileged user (e.g., “Sophie”) and explained the importance of giving users only the minimum permissions needed. Exercises with changing file permissions (`chmod`) illustrated this concept.

- **Key Commands:**

```
sudo apt update && sudo apt upgrade -y
sudo apt install openssh-server apache2
ssh-keygen -t rsa -b 2048 -f ~/.ssh/my_key
ssh-copy-id -i ~/.ssh/my_key.pub user@server
sudo nano /etc/ssh/sshd_config  # set: PermitRootLogin no,
PasswordAuthentication no
sudo systemctl restart sshd
sudo adduser Sophie
groups Sophie
sudo chmod 540 somefile
```

- **Exam–Relevant Explanation:**
  Be prepared to describe the steps for SSH hardening, explain why key-based authentication is superior, and discuss how enforcing least privilege (e.g., using non-privileged user accounts) helps prevent unauthorized access.

Lecture 6: Network Security Fundamentals & Traffic Analysis

- **Concepts & Details:**

  - **OSI Model & IP Addressing:**
    The professor broke down the layered model—from the physical and data link layers (which deal with MAC addresses and error checking) to the network layer (where IP addresses and routing decisions are made) and the transport layer (which handles TCP/UDP details).
  - **NAT & Private IPs:**
    Due to IPv4 limitations, private IP ranges (e.g., 10.x.x.x, 192.168.x.x) are used internally and then translated to a public IP via NAT. This translation not only conserves IP addresses but also provides a basic level of protection from external attacks.

- **Traffic Capture Tools:**
  Tools such as Wireshark and tcpdump were introduced to capture live network traffic. The professor demonstrated how packets contain headers added at each OSI layer (Ethernet, IP, TCP, HTTP) and explained how filtering (e.g., by protocol “HTTP” or by IP) helps analyze specific traffic.
- **DNS & WHOIS:**
  The use of `dig` and `host` to resolve domain names and verify DNS records was shown, along with how WHOIS can reveal domain registration details.
- **Firewalls:**
  The lecture compared a simple packet filter firewall (which uses only IP and port information) with a stateful firewall (which also monitors connection states such as whether a TCP connection is complete).

- **Key Commands:**

```
ip addr                # display network interfaces and IPs
nmap -sV target_host   # scan for open ports and service versions
dig google.com         # query DNS records
host caslab.case.edu   # lookup domain IP
nc www.case.edu 80     # use netcat to connect to a web server on port
80
```

- **Exam–Relevant Explanation:**
  You should be able to explain what happens at each OSI layer, how NAT preserves private networks, and the operational differences between packet filter and stateful firewalls. Additionally, know how basic network tools are used for troubleshooting.

Lecture 7: Advanced OS Hardening & Least Privilege Implementation

- **Concepts & Details:**

  - **User & Group Management:**
    The lecture focused on reinforcing the principle of least privilege. The professor demonstrated adding a new user (e.g., “Sophie”) and checking group memberships. He explained that every user should have only the permissions needed for their role.
  - **Modifying File Permissions:**
    Detailed exercises were given on changing permissions for files and directories. For instance, setting one file so that a specific group (e.g., “devs”) can read it while another file is restricted, and then modifying directory permissions so that the group can read, write, and execute.
  - **Sudo & Auditing:**
    The proper use of `sudo` and the configuration of the sudoers file (or adding users to the sudo group) were discussed. Logging (using tools like `journalctl`) was mentioned as a way to audit actions for security purposes.

- **Key Commands:**

```

```

```
sudo adduser Sophie
groups Sophie
sudo usermod -aG sudo Sophie   # grant limited sudo access if required
chmod 540 file.txt              # file permission example: read &
execute for owner, read for group, none for others
ls -l                           # list permissions
journalctl                      # view system logs
```

- **Exam–Relevant Explanation:**
  Understand how improper permissions can lead to vulnerabilities. Be ready to explain how adding a user to a specific group (or to the sudoers list) impacts security and how logs can be used to detect unauthorized activities.

Lecture 8: Traffic Capture & Protocol Analysis Deep Dive

- **Concepts & Details:**
  - **Wireshark Usage:**
    The professor explained how to capture live network traffic with Wireshark. He demonstrated filtering by protocol (such as HTTP) and explained how each captured packet contains details added by each OSI layer—from MAC addresses at the data link layer to source/destination IPs and ports at the network/transport layers, and finally the application data (HTTP headers, etc.).
  - **Man-in-the-Middle (MITM) Context:**
    While Wireshark itself is passive and only captures packets on your own network interface, the professor discussed MITM attacks—where an attacker intercepts traffic between endpoints—and explained that proper certificate validation (via TLS) helps defend against such attacks.
- **Exam–Relevant Explanation:**
  You should understand the significance of each protocol layer in a packet and how tools like Wireshark or tcpdump are used to analyze traffic. Also, be prepared to discuss why encryption (e.g., TLS) is essential to protect against MITM attacks.

Lecture 9: Network Scanning & Protocol Attacks

- **Concepts & Details:**
  - **Network Scanning with nmap:**
    The session covered how to use nmap to scan a host. The professor showed that using the `-sV` flag allows you to detect not only which ports are open but also the versions of the running software.
  - **DNS Tools & WHOIS:**
    Using `dig` and `host` commands, the instructor demonstrated how to retrieve DNS records. WHOIS was also used to extract domain registration details, which might be used in social engineering or risk assessment.
  - **Netcat (nc) Usage:**
    The professor demonstrated netcat as a tool to open a TCP connection (e.g., to port 80) and

manually send HTTP requests, revealing raw responses from the web server.
    - **Protocol Attacks:**
      A discussion on TCP SYN flood attacks highlighted that attackers spoof source IP addresses so that a server's SYN-ACK responses overwhelm it. This was contrasted with simple packet filtering.

- **Key Commands:**

```
nmap -sV target_hostname   # scan open ports and determine service versions
dig example.com            # perform DNS lookup
host caslab.case.edu       # resolve domain to IP
nc www.case.edu 80         # connect to web server using netcat
```

- **Exam–Relevant Explanation:**
  Be ready to articulate how scanning tools work, describe the mechanism behind a TCP SYN flood (spoofing source IP), and explain how DNS queries function as part of network troubleshooting.

Lecture 10: OS Service Management & System Maintenance

- **Concepts & Details:**

    - **Service Control:**
      The lecture detailed managing Linux services with `systemctl`—starting, stopping, restarting, enabling/disabling services (like Apache2). The professor emphasized that restarting a service is often needed after configuration changes.
    - **Software Updates:**
      The importance of updating the package repository (`apt update`) and then upgrading installed software (`apt upgrade`) was stressed. This practice minimizes vulnerabilities from outdated software.
    - **Backup & Logging:**
      Regular backups and proper logging (using tools like `journalctl`) were presented as critical for maintaining availability and for post-incident analysis.
    - **Configuration Hardening:**
      The professor discussed disabling directory listings in Apache to prevent attackers from viewing sensitive configuration files, and the use of custom error pages to limit information disclosure.

- **Key Commands:**

```
sudo apt update && sudo apt upgrade -y
sudo systemctl restart apache2
systemctl list-unit-files --state=disabled
sudo systemctl enable apache2
```

- **Exam–Relevant Explanation:**
  Understand why regular updates and careful service management reduce attack surfaces. Be ready to discuss how misconfigured services (like Apache showing an index listing) can expose sensitive information.

Lecture 11: Advanced OS Hardening, Malware, & Exam Review

- **Concepts & Details:**
  - **Enhanced SSH Security & PAM Configuration:**
    The professor provided a deep dive into SSH hardening. Detailed steps included generating and deploying SSH keys, editing the SSH configuration file (`/etc/ssh/sshd_config`) to disable password-based logins, and using PAM along with host-allow lists to restrict which IPs and users may access the system.
  - **Malware and Threat Awareness:**
    The discussion covered various malware types (viruses, trojans, spyware) and social engineering methods like phishing. Best practices such as not opening unsolicited email attachments and verifying sender domains were stressed.
  - **Exam Review:**
    The professor revisited core security concepts:
    - The CIA triad (Confidentiality, Integrity, Availability) as the guiding principles
    - Defense in depth (layered security measures)
    - The concept of risk management: identifying assets, threats, vulnerabilities, and acceptable risks
    - The importance of incident response planning (preparation, detection, containment, eradication, recovery)

- **Key Code/Command Example (SSH Hardening):**

```
# Generate SSH key pair
ssh-keygen -t rsa -b 2048 -f ~/.ssh/temp_key

# Copy the public key to the server
ssh-copy-id -i ~/.ssh/temp_key.pub user@server

# In /etc/ssh/sshd_config, set:
PermitRootLogin no
PasswordAuthentication no

# Restart SSH service
sudo systemctl restart sshd
```

- **Exam–Relevant Explanation:**
  Be prepared to detail the full process of securing SSH, including key generation, key deployment, and configuration changes. Also, know how layered defenses (defense in depth) and thorough incident response plans help mitigate both external attacks and insider threats.

Quiz Review Summary

- **Salting & Hashing:**
  - A unique salt is generated per user and can be stored alongside the hashed password to protect against rainbow table attacks.
  - A single salt for all users is insecure.
- **Certificates:**
  - Certificates serve to verify the identity of the certificate holder and provide a way to share public keys. They are not used for symmetric encryption or as password hashes.
- **Encryption Characteristics:**
  - Base64 is an encoding scheme (not an encryption algorithm).
  - Symmetric encryption (using one shared key) is faster and uses smaller key sizes compared to asymmetric encryption, which requires a key pair.
  - The proper method for public-key encryption is to use the recipient's public key for encryption and the recipient's private key for decryption.
- **Firewall Types:**
  - A packet filter firewall makes decisions based solely on IP addresses and port numbers, whereas a stateful firewall also tracks connection state (e.g., whether a TCP session is established) to enforce rules.

Potential Exam Questions

1. **Cryptography & Key Derivation:**
   - Explain how PBKDF2 uses a password and a unique salt to generate a cryptographic key. Why is it important to use a unique salt for each user?
2. **Encryption Differences:**
   - Compare symmetric and asymmetric encryption in terms of key management, speed, and typical use cases.
3. **SSH Hardening:**
   - Describe the process of securing SSH access on a Linux server, including key generation, deployment, and configuration changes (e.g., disabling password authentication).
4. **OSI Model & Network Tools:**
   - Explain the function of each layer in the OSI model and how tools like Wireshark can help analyze the data added at each layer.
5. **Firewalls:**
   - What is the operational difference between a packet filter firewall and a stateful firewall? Provide examples of when one might be preferred over the other.
6. **CIA Triad & Risk Management:**
   - Define confidentiality, integrity, and availability. How do these principles apply to protecting data at rest versus data in motion?
7. **Incident Response & Defense in Depth:**
   - Outline the steps of an effective incident response plan. How does the concept of defense in depth help mitigate both external and internal threats?
