**Title: Comprehensive Security Audit: Penetration Testing and Privilege Escalation on Hack the Box**

**Machine Level**: Medium

**Prepared By**: Anuj Kulat, Syed Hussain  
**Affiliation**: CIS , University of Wisconsin-Parkside

**Abstract**

This report presents an in-depth security audit conducted on a medium-level machine from the Hack the Box platform, employing advanced penetration testing techniques and privilege escalation methodologies. The purpose of this audit is to systematically assess the security vulnerabilities of the target machine, exploit identified weaknesses, and escalate user privileges to acquire full system control. This research further provides a holistic understanding of system vulnerabilities and presents remedial strategies to mitigate such risks in real-world scenarios.

**Introduction**

Cybersecurity is a rapidly evolving field that demands continuous assessment of systems to identify and address vulnerabilities before they can be exploited by malicious actors. Penetration testing, often referred to as ethical hacking, is a proactive approach to assessing security risks by simulating the tactics, techniques, and procedures used by attackers.

This report explores a penetration test conducted on a medium-level machine from Hack the Box, an online platform designed for cybersecurity professionals and enthusiasts to test their skills. The audit followed a structured approach from reconnaissance to post-exploitation, with an emphasis on privilege escalation. The audit findings provide valuable insights into the security posture of the target machine, helping to enhance understanding of vulnerabilities and the methods to address them effectively.

**Objectives**

The primary objectives of this audit are:

1. To perform a comprehensive penetration test on a medium-level Hack the Box machine.
2. To systematically identify and exploit vulnerabilities in the system.
3. To successfully escalate privileges from a low-privileged user to a superuser or root level.
4. To provide a detailed technical report on the findings, methodology, and remediation measures.
5. To contribute to the broader understanding of modern system vulnerabilities and security testing techniques.

**Scope of Work**

The scope of this security audit includes:

* **Reconnaissance and Enumeration**: Gathering information on open ports, running services, and possible entry points.
* **Vulnerability Assessment**: Identifying potential vulnerabilities through automated scanning and manual assessment.
* **Exploitation**: Gaining initial access by exploiting identified weaknesses.
* **Privilege Escalation**: Elevating privileges using system vulnerabilities, weak configurations, or misconfigurations.
* **Post-Exploitation**: Documenting the full system compromise and recommending strategies for mitigation.
* **Ethical Reporting**: Ensuring responsible disclosure of vulnerabilities and adherence to ethical hacking principles.

**Methodology**

The penetration testing process adheres to the industry-standard methodologies outlined by frameworks such as the Open Web Application Security Project (OWASP) and the Penetration Testing Execution Standard (PTES). The approach is divided into the following phases:

**Phase 1: Reconnaissance**

Reconnaissance involves both passive and active information gathering to understand the system’s architecture and potential vulnerabilities. Tools such as **Nmap** and **Dirb** were employed for scanning open ports, services, and directories. This phase is critical as it provides a comprehensive map of the attack surface.

* **Objective**: To gather as much relevant information as possible without alerting the target.
* **Tools**: Nmap, Nikto, Dirb, and custom scripts.

**Phase 2: Vulnerability Identification**

In this phase, automated tools such as **OpenVAS** and **Metasploit** were used to identify known vulnerabilities, followed by manual analysis to pinpoint potential attack vectors. The focus was on Common Vulnerabilities and Exposures (CVE) related to outdated software, misconfigurations, and weak encryption protocols.

* **Objective**: To identify exploitable vulnerabilities using both automated and manual techniques.
* **Tools**: OpenVAS, Metasploit, CVE databases.

**Phase 3: Exploitation**

The exploitation phase involved using vulnerabilities discovered during the assessment to gain unauthorized access to the system. By exploiting a service running on an open port, the initial foothold was established. A reverse shell was employed to maintain persistent access to the compromised system.

* **Objective**: To gain unauthorized access and maintain a stable session for further exploration.
* **Techniques**: Reverse shell, CVE exploits, custom payloads.

**Phase 4: Privilege Escalation**

Once a foothold was established with low-level access, the focus shifted to privilege escalation. Using kernel exploits, vulnerable setuid binaries, and misconfigured services, we were able to escalate privileges from a standard user to root.

* **Objective**: To elevate user privileges to root/admin and gain full system control.
* **Techniques**: Kernel exploits, misconfigured service exploitation, environment variable manipulation.

**Phase 5: Post-Exploitation**

With root-level access achieved, post-exploitation activities were carried out to assess the full impact of the compromise. This included identifying sensitive files, reviewing system logs, and mapping internal networks. The focus was on ensuring that all traces of the attack could be cleaned up, and an incident response plan was simulated.

* **Objective**: To understand the broader impact of the attack and provide strategies for mitigation.

**Key Findings**

1. **Vulnerability 1: Outdated Software**  
   A vulnerable version of **[specific service]** was running on the target machine, allowing remote code execution through a well-known CVE (CVE-XXXX-XXXX). The exploit provided an initial foothold in the system.
2. **Vulnerability 2: Weak File Permissions**  
   Critical system files were improperly configured, with weak permissions that allowed non-privileged users to modify sensitive files. This weakness was exploited to escalate privileges to root.
3. **Privilege Escalation**  
   A known kernel exploit (CVE-XXXX-XXXX) was used to escalate privileges from the initial foothold to root-level access, providing full control over the machine.
4. **Post-Exploitation Impact**  
   With root privileges, the attacker could access all sensitive information stored on the machine, including user credentials, system logs, and internal network configurations.

**Discussion**

This audit underscores the importance of maintaining up-to-date software and strict access control measures. The vulnerabilities identified during the penetration test highlight systemic issues such as poor patch management, improper file permissions, and insecure service configurations. Furthermore, the privilege escalation techniques demonstrated that even minor misconfigurations can lead to full system compromise. As a result, this audit reveals significant areas for improvement in system administration and security practices.

The ability to escalate privileges also points to a broader problem within many networked environments: insufficient isolation of user roles. This highlights the need for organizations to implement the principle of least privilege more rigorously and employ more robust authentication mechanisms.

**Recommendations**

1. **Patch Management**  
   Implement a robust patch management process to ensure that all software is updated regularly, particularly to address known CVEs. Tools like **WSUS** or **Ansible** can automate this process.
2. **Access Control**  
   Strengthen file permission policies and ensure that sensitive files are accessible only to authorized users. Employ more granular role-based access control (RBAC) mechanisms.
3. **Privilege Management**  
   Enforce the principle of least privilege, ensuring that users only have the access necessary for their role. Implement multi-factor authentication (MFA) across all access points to limit the risk of unauthorized access.
4. **Regular Security Audits**  
   Schedule regular penetration tests and security audits to identify and fix vulnerabilities before they are exploited by attackers.
5. **Incident Response**  
   Develop and maintain a comprehensive incident response plan, including regular simulations, to minimize the impact of security breaches.

**Conclusion**

This penetration testing audit successfully identified critical vulnerabilities in the target system, leading to full system compromise through privilege escalation. The findings illustrate the need for stringent security practices, particularly in the areas of software patching, access control, and user privilege management. Addressing these vulnerabilities will significantly reduce the likelihood of future attacks and enhance the overall security posture of the system.