Aadi Akyianu, Essy Ingram, Lazuli Kleinhans

Hacking Tools Comps 2024

linPEVES Project Explanation

What is linPEVES?

Linux Privilege Escalation and Vulnerability Exploit Script (also known as linPEVES) is a tool that scans for and exploits privilege escalation vulnerabilities. Inspired by the open-source script linPEAS (Linux Privilege Escalation Awesome Script), a privilege escalation vulnerability scanning tool, our script not only identifies vulnerabilities within a target system but allows the user to perform exploits on each vulnerability found.

This tool was created as part of our final Computer Science Comprehensive Exercise (Comps) project. We wanted to learn more about privilege escalation by researching vulnerabilities and figuring out how to exploit them. The goal was to understand what security measures need to be in place (or bypassed) to protect a system from privilege escalation attacks.

What is privilege escalation?

Like any human design, computers have weaknesses which are known as vulnerabilities. Our project identifies and exploits these vulnerabilities by escalating privileges within a Linux system. One example of this might be a user whose account has root access and stores their password in a file that is accessible to unprivileged users. These unprivileged users can exploit this vulnerability and escalate their privileges by signing into the root user's account.

Project Details

Set Up

To run our project we set up a Linux Virtual Machine (VM) on Amazon Web Services (AWS). We chose a cloud-based virtual machine to give us experience dealing with cloud architecture and for easy access from anywhere. The VM ran Ubuntu 24.04 with a

In order to test our scans, we added several vulnerabilities to our machine. For example: we made the /etc/passwd file globally writable; added a writable cron job that was run by root; downgraded the bash version to make it susceptible to shellshock; and many more.

We also created a test-user to determine whether our scans and exploits were able to grant root privileges to an unprivileged user.

Architecture

The project consists of:

- A `main.sh` file
- The `scans/` directory contains all available scans
- The `exploits/` directory contains all available exploits
- The `lib/` directory contains additional files that the exploits might use
- The `README.md` file which describes how to install and run the code

```
main.sh

scans/
     —> cron-scan.sh
     —> path-scan.sh
     —> ...

exploits/
     —> cron-exploit.sh
     —> path-exploit.sh
     —> ...

lib/
     —> exploit-bin
     —> exploit.sh

README.md
```

How do I run linPEVES?

Once you have installed linPEVES, make sure you are in the **`linPEVES/`** directory where the **`main.sh`** file is located.

From there, you can decide which scans and exploits you want to run. To see the table of available scans and exploits, run **`./main.sh --list`**. The list is reproduced below:

#	SCANS	EXPLOITS
0	cron-scan	cron-exploit
1	env-var-scan	env-var-exploit
2	path-scan	path-exploit
3	pkexec-scan	pkexec-exploit
4	readable-passwd-scan	readable-passwd-exploit
5	readable-shadow-scan	readable-shadow-exploit
6	shellshock-scan	shellshock-exploit
7	sudo-scan	sudo-exploit
8	sudoers-scan	sudoers-exploit
9	systemctl-bin-scan	systemctl-bin-exploit
10	writable-passwd-scan	writable-passwd-exploit
11	writable-shadow-scan	writable-shadow-exploit

You can run a scan/exploit by referencing their indices in the table. For example, if you wanted to run all scans but only exploit vulnerabilities 1 and 7, you would run the following command:

```
./main.sh -s {0..11} -e 1 7
```

Vulnerabilities Explained

V0: Writable crontab jobs run as root

- This vulnerability consists of a cronjob that is run as root but is editable by an unprivileged user. This can be exploited by writing malicious code into the cronjob that is then executed by root.

V1: Environmental variable scan

- If users choose to store an account password in an environmental variable on their system (a vulnerability), this scan will identify them and list them in the exploitable_passwords variable. The exploit of this vulnerability returns the list of passwords to the linPEVES user to gain access to any greater-privileged user that they choose.

V2: Writable binaries on PATH

- Any PATH binaries that are writable by an unprivileged user. Exploiting this vulnerability is as easy as overwriting the binary with malicious code that, if run by a root user, will grant them admin privileges.

V3: The pkexec Executable

- pkexec is a command that allows users to run commands as root. The way this scan works is by first checking for the pkexec executable then it checks if the 50-localauthority.conf. file is writable. If so and the user wishes to exploit this vulnerability, the user's group is then added to the list of allowed users. This is an alternative way to gain sudo access.

V4: Passwords in /etc/passwd

- While having the permissions to read the /etc/passwd file is not in itself a vulnerability to a system, there is still a possibility for the password of a user to be included in this file. The scan of this vulnerability iterates through each user in the /etc/passwd file checking if the password has been hidden. If the password is readable, the password and its corresponding username are printed out in the exploit.

V5: A Readable Shadow File

- This scan checks if /etc/shadow is readable by the current user. If so, the exploit returns the hashes that are read from the file and advises the user on which tools they should consult to crack the hashes so that they can gain access to the privileges of any user on the system.

V6: Shellshock: A Bash Vulnerability

- Before bash version 4.3, earlier versions of bash were susceptible to a vulnerability known as shellshock. The way this works is that by defining a variable in just the right way, you can cause bash to execute code. This is especially helpful for remote exploitations as you can create a reverse shell that grants you access to a machine without having to use ssh.
- The shellshock-scan works under the assumption that you are attacking a remote machine with a webserver and a cgi file. Using curl it sends a variable with code injected into it and depending on the response, that will determine if the machine is susceptible to shellshock or not. If it is and the user wishes to exploit that, the shellshock-exploit then sets up a reverse shell on a listener that the user provides.

V7: Outdated sudo version

- Any version of sudo 1.8.0 or earlier contains a bug that allows the current user to edit the sudoers file. The scan of this vulnerability verifies the paths that allow the linPEVES user to edit the sudoers. If the version is compatible with this scan/exploit and there are any paths found to edit sudo, then the exploit will add the linPEVES user to the list of sudoers.

V8: Writable sudoers file

- The /etc/sudoers file is used by the system to specify added privileges of users and groups in the system. The sudoers-scan checks to see if the /etc/sudoers file is writable by the user. If it is and the user wishes to exploit it, the sudoers-exploit adds the user to the sudoers file with access to everything and NOPASSWD enabled.

V9: Writable systemctl binaries

- This vulnerability is present on a system if any of the binaries executed by systemctl (the Linux service manager) are writable by an unprivileged user. Much like the writable PATH binary scan, exploiting consists of overwriting the binary with malicious code that can give admin privileges.

V10: Writable passwd file

- The /etc/passwd file contains information about the users of the system. Typically the /etc/passwd file does not contain the password hashes for the users however, it is possible to store them there. If there is a hash present in the /etc/passwd file, the system uses that hash to authenticate the user rather than the hash stored in the /etc/shadow file.
- The writable-passwd-scan checks to see if the /etc/passwd file is writable by the user. If it is and the user wishes to exploit it, the writable-passwd-exploit changes the hash of the root user to a known hash and notifies the user of the new password.

V11: Writable shadow file

- Similar to /etc/passwd, the /etc/shadow file also contains some information about the system users. However, in this case, the shadow file contains information about the users' passwords as well as the hashes of the passwords. The writable-shadow-scan checks to see if the /etc/shadow file is writable by the user. If it is and the user wishes to exploit it, the writable-shadow-exploit changes the hash of the root user to a known hash and notifies the user of the new password.

Example Scan Functionality

Each scan looks through the system for indications that a chosen vulnerability is present on the system. If a file or program exists that matches the criteria, the relevant information is written to the exploit file. For example, for the cron-scan, if it finds a cronjob that is running with sudo permissions and is editable by an unprivileged user, it will write the path to the cronjob script file to the exploit file.

Example Exploit Functionality

The vulnerability's respective exploit file will then read the data that was written to it by the scan file and then act upon that information. For the cron-exploit, it will overwrite the cronjob script with malicious code that gives the unprivileged user root access.

Bibliography

Archlinux.org Contributors. “[Solved] Pkexec Fails, Can’t Run Vmware Player. / AUR Issues, Discussion & PKGBUILD Requests / Arch Linux Forums.” Archlinux.org, 2023. https://bbs.archlinux.org/viewtopic.php?id=285391.

Carson, Joseph. “Privilege Escalation on Linux.” delinea.com, 2024. https://delinea.com/blog/linux-privilege-escalation.

Dezso, Richard. “Linux Privilege Escalation Techniques for Hacking.” StationX, May 11, 2024. https://www.stationx.net/linux-privilege-escalation.

GeeksforGeeks. “How to Setup Cron Jobs in Ubuntu?” GeeksforGeeks, August 7, 2024. https://www.geeksforgeeks.org/how-to-setup-cron-jobs-in-ubuntu/.

INE. “Lab Walkthrough - Shockin’ Shells: ShellShock (CVE-2014-6271).” INE Expert IT Training. INE, Inc., 2014. https://ine.com/blog/shockin-shells-shellshock-cve-2014-6271.

Li, Vickie. “Vickie Li’s Security Blog.” Vickie Li’s Security Blog, September 29, 2020. https://vickieli.dev/system%20security/cron-security/.

Mehndiratta, Madhav. “Exploiting a Shellshock Vulnerability.” Infosec Articles, June 25, 2020. https://www.infosecarticles.com/exploiting-shellshock-vulnerability/.

opsxcq. “Shellshock Exploit + Vulnerable Environment.” GitHub, April 14, 2023. https://github.com/opsxcq/exploit-CVE-2014-6271.

Polop, Carlos. “Linux Privilege Escalation - HackTricks.” Hacktricks.xyz, August 2024. https://book.hacktricks.xyz/linux-hardening/privilege-escalation.

Rashid, Fahmida. “Sudo Flaw Gives Linux Users Root Access.” Decipher. Duo Security, January 26, 2021. https://duo.com/decipher/sudo-flaw-gives-linux-users-root-access.

Týč, Matěj. “Argbash Documentation — Argbash 2.10.0 Documentation.” Readthedocs.io, 2014. https://argbash.readthedocs.io/en/stable/.

———. “Argbash: Bash Argument Parsing Made Easy.” Argbash.dev, 2016. https://argbash.dev/.

Zeuthen, David. “Manual - Section 8: Pklocalauthority.” Fm4dd.com, 2024. https://linux.fm4dd.com/en/man8/pklocalauthority.shtm.

———. “Pkexec(1): Execute Command as Another User - Linux Man Page.” linux.die.net, n.d. https://linux.die.net/man/1/pkexec.