diego-treitos / **linux-smart-enumeration** Public

Notifications Fork 573 Star 3.4k

Code Issues 3 Pull requests Actions Projects Security Insights

master Go to file Code

diego-treitos Merge pull request #83 from Bornunique9... eb80976 · last year 299 Commits

cve	added check for cve-2023-22809 (Sud...	last year
doc	Improved setuid binaries	3 years ago
screenshots	Added screenshots	5 years ago
tools	Added scripts and code for the CVE fea...	2 years ago
LICENSE	Update LICENSE	4 years ago
README.md	Update README.md	last year
lse.sh	Version 4.14nw	last year

README GPL-3.0 license

First, a couple of useful oneliners ;)

```
wget "https://github.com/diego-treitos/linux-smart-enumeration/relea
```

```
curl "https://github.com/diego-treitos/linux-smart-enumeration/relea
```

Note that since version `2.10` you can *serve the script* to other hosts with the `-S` flag!

linux-smart-enumeration

Linux enumeration tools for pentesting and CTFs

This project was inspired by https://github.com/rebootuser/LinEnum and uses many of its tests.

Unlike LinEnum, `lse` tries to gradualy expose the information depending on its importance from a privesc point of view.

What is it?

This shell script will show relevant information about the security of the local Linux system, helping to escalate privileges.

From version **2.0** it is *mostly* **POSIX** compliant and tested with `shellcheck` and `posh`.

It can also **monitor processes to discover recurrent program executions**. It monitors while it is executing all the other tests so you save some time. By default it monitors during 1 minute but you can choose the watch time with the `-p` parameter.

It has 3 levels of verbosity so you can control how much information you see.

About

Linux enumeration tool for pentesting and CTFs with verbosity levels

hacking pentesting privilege-escalation oscp ctfs privesc hackthebox linux-enumeration

Readme
GPL-3.0 license
Activity
3.4k stars
57 watching
573 forks
Report repository

Releases 15

Release 4.14nw Latest
on Dec 23, 2023

+ 14 releases

Packages

No packages published

Contributors 7

Languages

Shell 100.0%

In the default level you should see the highly important security flaws in the system. The level `1` (`./lse.sh -l1`) shows interesting information that should help you to privesc. The level `2` (`./lse.sh -l2`) will just dump all the information it gathers about the system.

By default it will ask you some questions: mainly the current user password (if you know it ;) so it can do some additional tests.

How to use it?

The idea is to get the information gradually.

First you should execute it just like `./lse.sh`. If you see some green `yes!`, you probably have already some good stuff to work with.

If not, you should try the `level 1` verbosity with `./lse.sh -l1` and you will see some more information that can be interesting.

If that does not help, `level 2` will just dump everything you can gather about the service using `./lse.sh -l2`. In this case you might find useful to use `./lse.sh -l2 | less -r`.

You can also select what tests to execute by passing the `-s` parameter. With it you can select specific tests or sections to be executed. For example `./lse.sh -l2 -s usr010,net,pro` will execute the test `usr010` and all the tests in the sections `net` and `pro`.

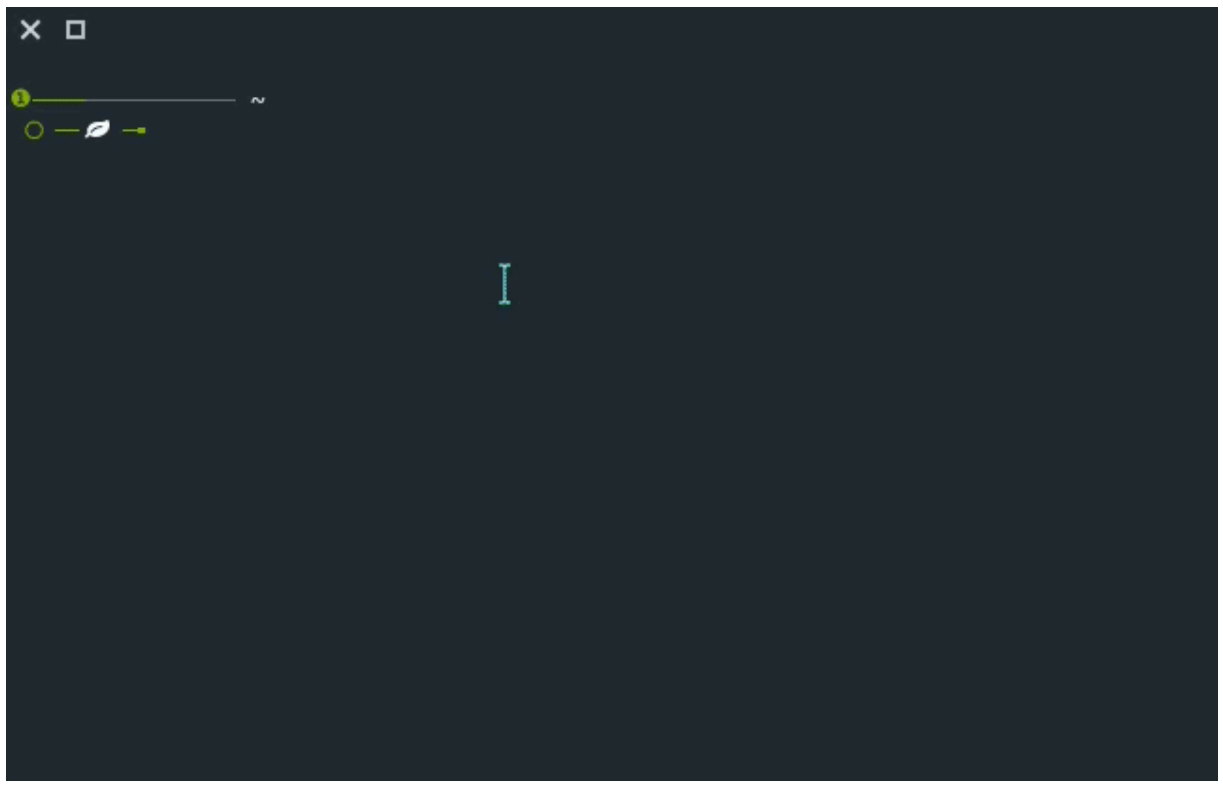
```
Use: ./lse.sh [options]

 OPTIONS
  -c           Disable color
  -i           Non interactive mode
  -h           This help
  -l LEVEL     Output verbosity level
                 0: Show highly important results. (default)
                 1: Show interesting results.
                 2: Show all gathered information.
  -s SELECTION Comma separated list of sections or tests to run. Ava
               sections:
                 usr: User related tests.
                 sud: Sudo related tests.
                 fst: File system related tests.
                 sys: System related tests.
                 sec: Security measures related tests.
                 ret: Recurren tasks (cron, timers) related tests.
                 net: Network related tests.
                 srv: Services related tests.
                 pro: Processes related tests.
                 sof: Software related tests.
                 ctn: Container (docker, lxc) related tests.
                 cve: CVE related tests.
               Specific tests can be used with their IDs (i.e.: usr0
  -e PATHS     Comma separated list of paths to exclude. This allows
               to do faster scans at the cost of completeness
  -p SECONDS   Time that the process monitor will spend watching for
               processes. A value of 0 will disable any watch (defau
  -S           Serve the lse.sh script in this host so it can be ret
               from a remote host.
```

Is it pretty?

Usage demo

Also available in webm video



Level 0 (default) output sample

```
    Hostname: dvwa
       Linux: 3.16.0-4-amd64
Distribution: Debian GNU/Linux 8.6 (jessie)
Architecture: x86_64

================================================================( users )=====
[i] usr000 Current user groups.................................................. yes!
[*] usr010 Is current user in an administrative group?.......................... nope
[*] usr020 Are there other users in an administrative groups?................... yes!
[*] usr030 Other users with shell............................................... yes!
[i] usr040 Environment information.............................................. skip
[i] usr050 Groups for other users............................................... skip
[i] usr060 Other users.......................................................... skip
================================================================( sudo )=====
[!] sud000 Can we sudo without a password?...................................... nope
[!] sud010 Can we list sudo commands without a password?........................ nope
[*] sud040 Can we read /etc/sudoers?............................................ nope
[*] sud050 Do we know if any other users used sudo?............................. nope
============================================================( file system )=====
[*] fst000 Writable files outside user's home................................... yes!
[*] fst010 Binaries with setuid bit............................................. yes!
[!] fst020 Uncommon setuid binaries............................................. nope
[!] fst030 Can we write to any setuid binary?.................................... nope
[*] fst040 Binaries with setgid bit............................................. skip
[!] fst050 Uncommon setgid binaries............................................. skip
[!] fst060 Can we write to any setgid binary?.................................... skip
[*] fst070 Can we read /root?................................................... nope
[*] fst080 Can we read subdirectories under /home?.............................. nope
[*] fst090 SSH files in home directories........................................ nope
[*] fst100 Useful binaries...................................................... yes!
[*] fst110 Other interesting files in home directories.......................... nope
[!] fst120 Are there any credentials in fstab/mtab?............................. nope
[*] fst130 Does 'treitos' have mail?............................................ nope
[!] fst140 Can we access other users mail?....................................... nope
[*] fst150 Looking for GIT/SVN repositories...................................... nope
[i] fst500 Files owned by user 'treitos'........................................ skip
[i] fst510 SSH files anywhere................................................... skip
[i] fst520 Check hosts.equiv file and its contents.............................. skip
[i] fst530 List NFS server shares............................................... skip
[i] fst540 Dump fstab file...................................................... skip
================================================================( system )=====
[i] sys000 Who is logged in..................................................... skip
[i] sys010 Last logged in users................................................. skip
[!] sys020 Does the /etc/passwd have hashes?.................................... nope
[!] sys030 Can we read /etc/shadow file?........................................ nope
[!] sys030 Can we read /etc/shadow- file?....................................... nope
[!] sys030 Can we read /etc/shadow~ file?....................................... nope
[!] sys030 Can we read /etc/master.passwd file?................................. nope
[*] sys040 Check for other superuser accounts................................... nope
[*] sys050 Can root user log in via SSH?........................................ yes!
[i] sys060 List available shells................................................ skip
[i] sys070 System umask in /etc/login.defs...................................... skip
[i] sys080 System password policies in /etc/login.defs.......................... skip
================================================================( security )=====
[*] sec000 Is SELinux present?.................................................. nope
[*] sec010 List files with capabilities......................................... yes!
[!] sec020 Can we write to a binary with caps?.................................. nope
[!] sec030 Do we have all caps in any binary?................................... yes!
---
/home/treitos/openssl =ep
---
[*] sec040 Users with associated capabilities.................................. nope
[!] sec050 Does current user have capabilities?................................. skip
============================================================( recurrent tasks )=====
[*] ret000 User crontab........................................................ nope
[!] ret010 Cron tasks writable by user......................................... nope
[*] ret020 Cron jobs                                                           yes!
```

Level 1 verbosity output sample

```
    Hostname: dvwa
       Linux: 3.16.0-4-amd64
Distribution: Debian GNU/Linux 8.6 (jessie)
Architecture: x86_64

================================================================( users )=====
[i] usr000 Current user groups.............................................. yes!
[*] usr010 Is current user in an administrative group?...................... nope
[*] usr020 Are there other users in an administrative groups?............... yes!
---
adm:x:4:zabbix
---
[*] usr030 Other users with shell........................................... yes!
---
root:x:0:0:root:/root:/bin/bash
treitos:x:1000:1000:treitos,,,:/home/treitos:/bin/bash
web-admin:x:1001:33::/var/www/sites:/usr/bin/rssh
---
[i] usr040 Environment information.......................................... skip
[i] usr050 Groups for other users........................................... skip
[i] usr060 Other users...................................................... skip
================================================================( sudo )=====
[!] sud000 Can we sudo without a password?.................................. nope
[!] sud010 Can we list sudo commands without a password?.................... nope
[*] sud040 Can we read /etc/sudoers?........................................ nope
[*] sud050 Do we know if any other users used sudo?......................... nope
============================================================( file system )=====
[*] fst000 Writable files outside user's home............................... yes!
---
/home/treitos
/var/lib/php5/sessions
/var/spool/postfix/dev/log
/var/spool/postfix/dev/random
/var/spool/postfix/dev/urandom
/var/tmp
/tmp
/tmp/.XIM-unix
/tmp/.Test-unix
/tmp/.ICE-unix
/tmp/.X11-unix
/tmp/.font-unix
---
[*] fst010 Binaries with setuid bit......................................... yes!
---
/usr/lib/eject/dmcrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/bin/gpasswd
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/passwd
/usr/bin/newgrp
/bin/su
/bin/umount
/bin/mount
---
[!] fst020 Uncommon setuid binaries......................................... nope
[!] fst030 Can we write to any setuid binary?............................... nope
[*] fst040 Binaries with setgid bit......................................... skip
[!] fst050 Uncommon setgid binaries......................................... skip
[!] fst060 Can we write to any setgid binary?............................... skip
[*] fst070 Can we read /root?............................................... nope
[*] fst080 Can we read subdirectories under /home?.......................... nope
[*] fst090 SSH files in home directories.................................... nope
[*] fst100 Useful binaries.................................................. yes!
---
/bin/nc
/bin/netcat
```

Level 2 verbosity output sample

```
    Hostname: dvwa
       Linux: 3.16.0-4-amd64
Distribution: Debian GNU/Linux 8.6 (jessie)
Architecture: x86_64

================================================================( users )=====
[i] usr000 Current user groups.......................................... yes!
---
treitos cdrom floppy audio dip video plugdev netdev
---
[*] usr010 Is current user in an administrative group?..................... nope
[*] usr020 Are there other users in an administrative groups?.............. yes!
---
adm:x:4:zabbix
---
[*] usr030 Other users with shell....................................... yes!
---
root:x:0:0:root:/root:/bin/bash
treitos:x:1000:1000:treitos,,,:/home/treitos:/bin/bash
web-admin:x:1001:33::/var/www/sites:/usr/bin/rssh
---
[i] usr040 Environment information...................................... yes!
---
SHELL=/bin/sh
PATH=/usr/bin:/bin:/sbin:/usr/sbin
PWD=/tmp
LANG=gl_ES.UTF-8
SHLVL=3
HOME=/home/treitos
LOGNAME=treitos
_=/usr/bin/env
---
[i] usr050 Groups for other users....................................... yes!
---
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:zabbix
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:
fax:x:21:
voice:x:22:
cdrom:x:24:treitos
floppy:x:25:treitos
tape:x:26:
sudo:x:27:
audio:x:29:treitos
dip:x:30:treitos
www-data:x:33:
backup:x:34:
operator:x:37:
list:x:38:
```