TTM4175 Introduction to Communication Technology and data security

Network mapping

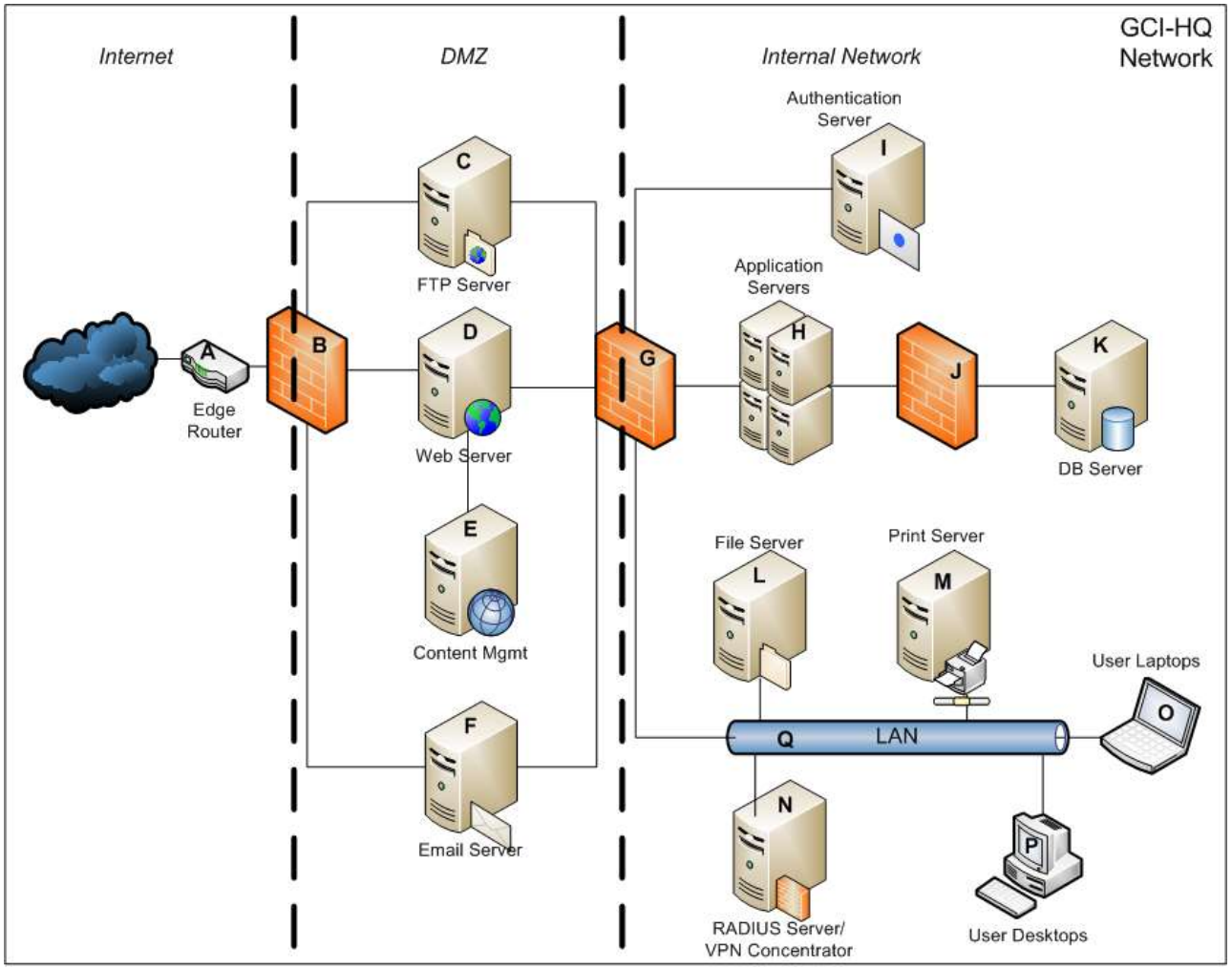
Laszlo Erdödi

laszlo.erdodi@ntnu.no

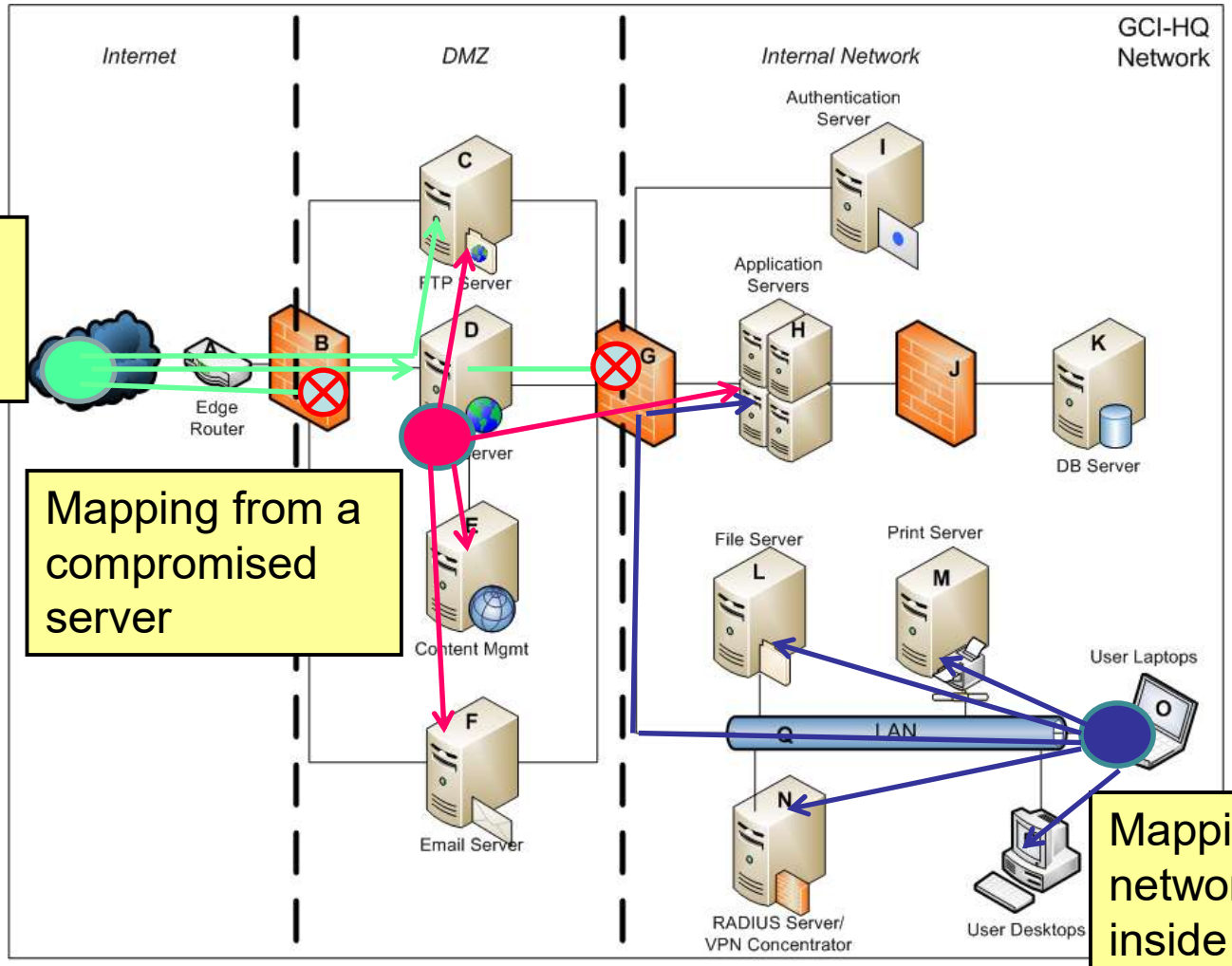
Lecture Overview

- Identifying hosts in a network
- Identifying services on a host
- What are the typical services
- Get in touch with services

Network layout example

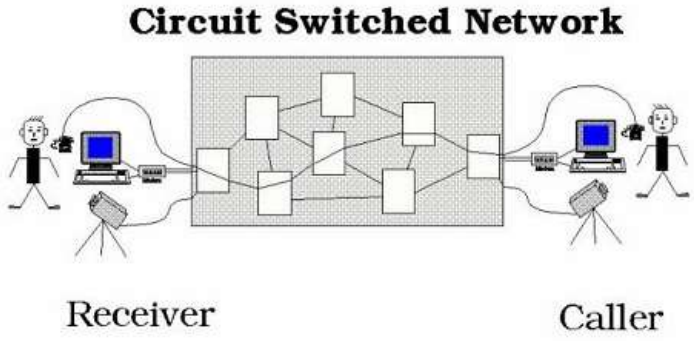


Network scanning positions

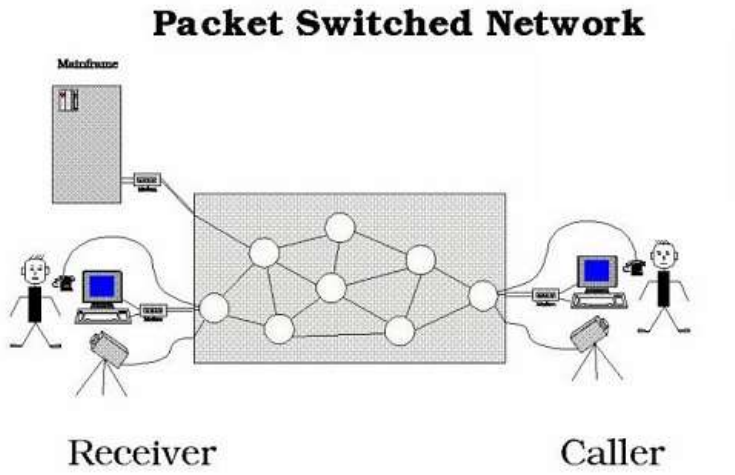


Circuit switched vs Packet switched networks

In circuit switched networks a virtual line is allocated between the communicating parties. The line is busy until the communication ends.

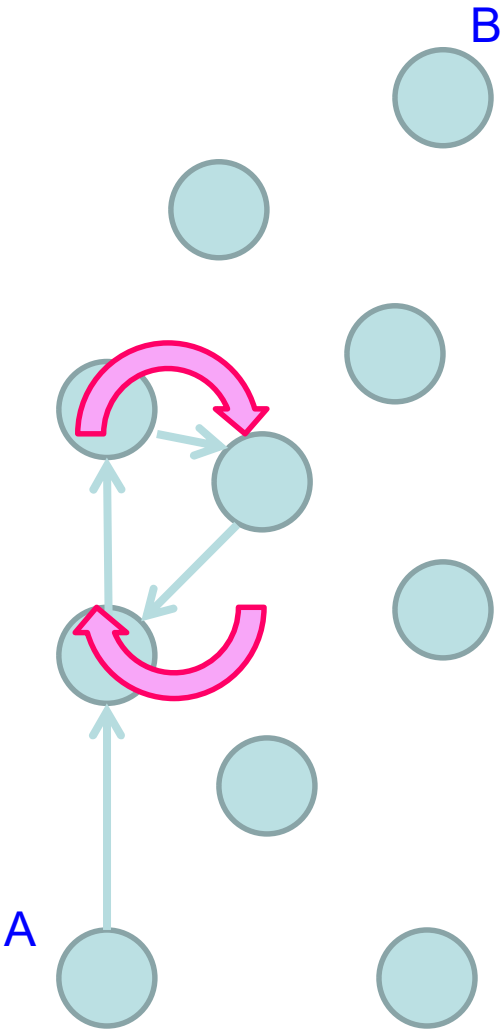


In packet switched networks the caller sends packets to the direction of the receiver. There's no planned route, each network device chooses the most appropriate device as next considering routing tables and traffic.



Packet switched networks – avoiding infinite loops

- As there's no planned route between the sender and the receiver it can happen that a packet gets stuck in the network following an infinite loop
- Messages are placed in network packets according to the *OSI* model
- Every packet should contain a *ttl* value (*Time to Live*) that is decreasing when arriving to the next network device (network hop)
- When *ttl* is 1 the packet has to be dropped



Layer 3 – Internet Control Message Protocol (ICMP)

IP Datagram

	Bits 0–7	Bits 8–15	Bits 16–23	Bits 24–31
IP Header (20 bytes)	Version/IHL	Type of service	Length	
	Identification		*flags* and *offset*	
	Time To Live (TTL)	Protocol	Checksum	
	Source IP address			
	Destination IP address			
ICMP Header (8 bytes)	Type of message	Code	Checksum	
	Header Data			
ICMP Payload (*optional*)	Payload Data			

- To check if a host is responding
- *Echo request* – *Echo reply* to make sure a host is turned on

Network mapping - answer options

- **Positive answer**

  In case of *icmp* we get an echo reply for our echo request

- **Negative answer**

  In case of *icmp* we get destination unreachable / host unreachable message

- **No answer**

  In case of *icmp*, we have no response from the host that was addressed by the echo request

Internet Control Message Protocol (ICMP) examples - ping

```
root@kali:~# ping www.uio.no
PING www.uio.no (129.240.171.52) 56(84) bytes of data.
64 bytes from www.uio.no (129.240.171.52): icmp_seq=1 ttl=128 time=14.6 ms
64 bytes from www.uio.no (129.240.171.52): icmp_seq=2 ttl=128 time=48.2 ms
64 bytes from www.uio.no (129.240.171.52): icmp_seq=3 ttl=128 time=11.0 ms
^C
--- www.uio.no ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 11.082/24.657/48.205/16.716 ms
```

Type	Message
0	Echo reply
3	Destination unreachable
4	Source quench
5	Redirect
8	Echo request
11	Time exceeded
12	Parameter unintelligible
13	Time-stamp request
14	Time-stamp reply
15	Information request
16	Information reply
17	Address mask request
18	Address mask reply

https://www.slideshare.net/asimnawaz54/internet-control-message-protocol

Layer 3 – Internet Control Message Protocol (ICMP)

Since ICMP contains the *ttl* value, it is possible to guess the receiver host's operating system by its *ttl*.

Initial *ttl* values:

- Windows: 128 since Windows2000
- Linux: 64 for 2.0.x kernel
- Solaris: 255

Detailed list at *Subin's Blog*: https://subinsb.com/default-device-ttl-values/

ICMP practice examples:

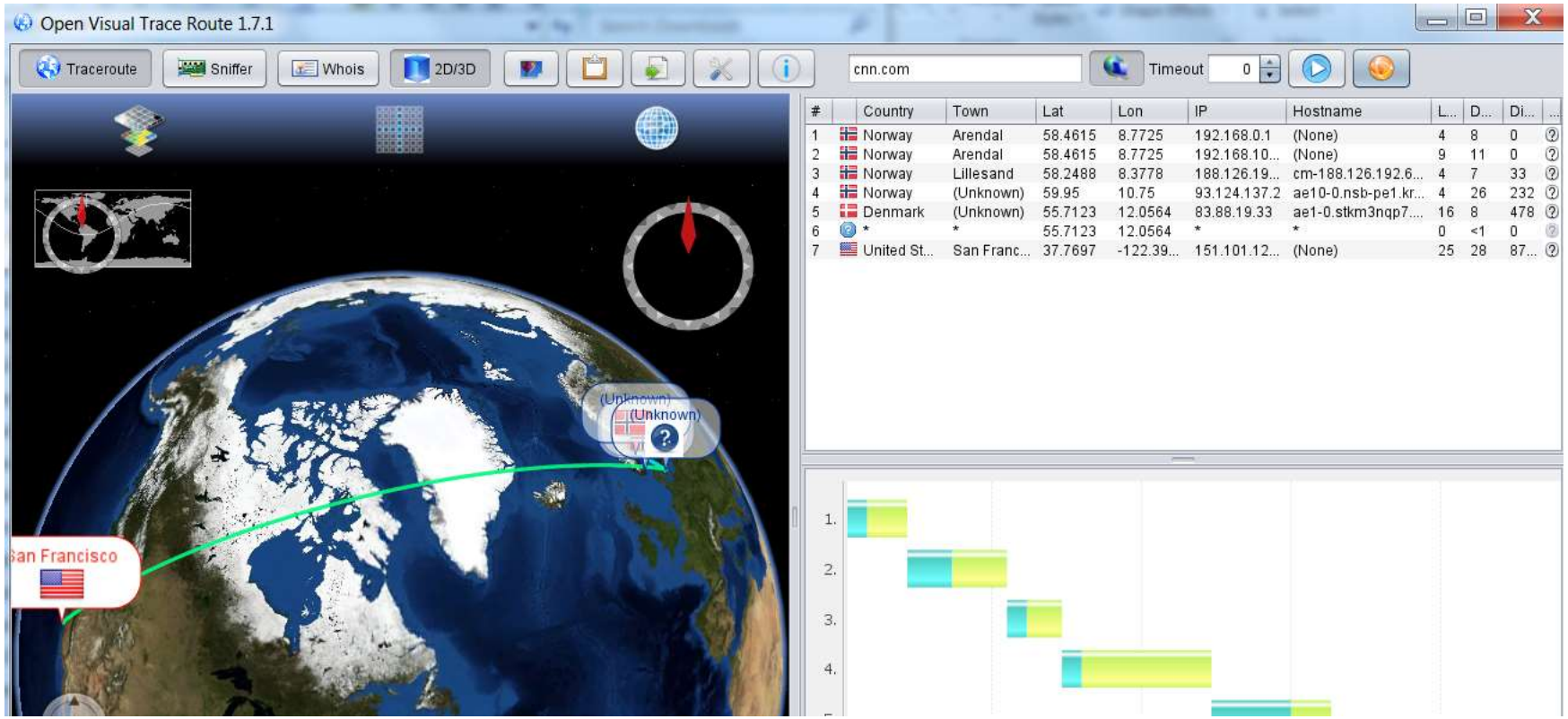
- Find a host with 64 as initial *ttl*
- Find a host with 128 as initial *ttl*

Internet Control Message Protocol (ICMP) examples - traceroute

Since all devices have to drop the packets with *ttl*=1, it is possible to map the route of a packet by repeating the ping with increasing *ttl* values. First, the initial *ttl* is 2, so after the first hop the device sends a time exceeded message. With *ttl*=3 the time exceed message is coming from the device at the second hop, etc.

```
C:\Users\laszloe>tracert htgth.com

Tracing route to htgth.com [69.16.220.113]
over a maximum of 30 hops:

  1     2 ms     1 ms     1 ms  192.168.0.1
  2     1 ms     1 ms     1 ms  192.168.100.1
  3     7 ms     4 ms     5 ms  cm-188.126.192.69.getinternet.no [188.126.192.69]
  4     5 ms     3 ms     4 ms  ae10-0.nsb-pe1.krs.no.ip.tdc.net [93.124.137.2]
  5    18 ms    16 ms    17 ms  ae1-0.stkm3nqp7.se.ip.tdc.net [83.88.19.33]
  6    16 ms    16 ms    16 ms  ae-10.bar1.Stokholm1.Level3.net [4.68.73.101]
  7     *        *        *     Request timed out.
  8   141 ms   136 ms   136 ms  4-15-84-142.liquidweb.com [4.15.84.142]
  9   144 ms   141 ms   141 ms  lw-dc2-core1-nexus-eth3-20.rtr.liquidweb.com [209.59.157.81]
 10   141 ms   141 ms   142 ms  lw-dc2-dist1-nexus-eth4-1.rtr.liquidweb.com [209.59.157.201]
 11   136 ms   137 ms   136 ms  host1.heretodaygonetohell.com [69.16.220.113]

Trace complete.
```

Internet Control Message Protocol (ICMP) examples – visual traceroute



Nmap basic usage

Nmap is an universal port scanner

It is able to carry out ordinary and specific host and service discoveries

Nmap has a scripting engine which makes it capable of carrying out complex scanning as well as vulnerability discovery, fuzzing, etc. tasks

For one simple ping the following command has to be used:

Nmap basic usage

Host(s) to be scanned can be set in multiple ways:

With domain: www.uio.no

With *ip*: 129.240.171.52

With *ip* range (CIDR): 129.240.171.0/24

With *ip* range (from-to) 129.240.171.2-6, 129.240.170-175.1

With list: 129.240.171.1,129.240.171.2

The main parameter is the scanning type that can be set with the –*s* switch, e.g. -*sP*: ping scan

Example task: How many hosts are alive in our current local network range? E.g. *nmap –sP 192.168.0.0/24*

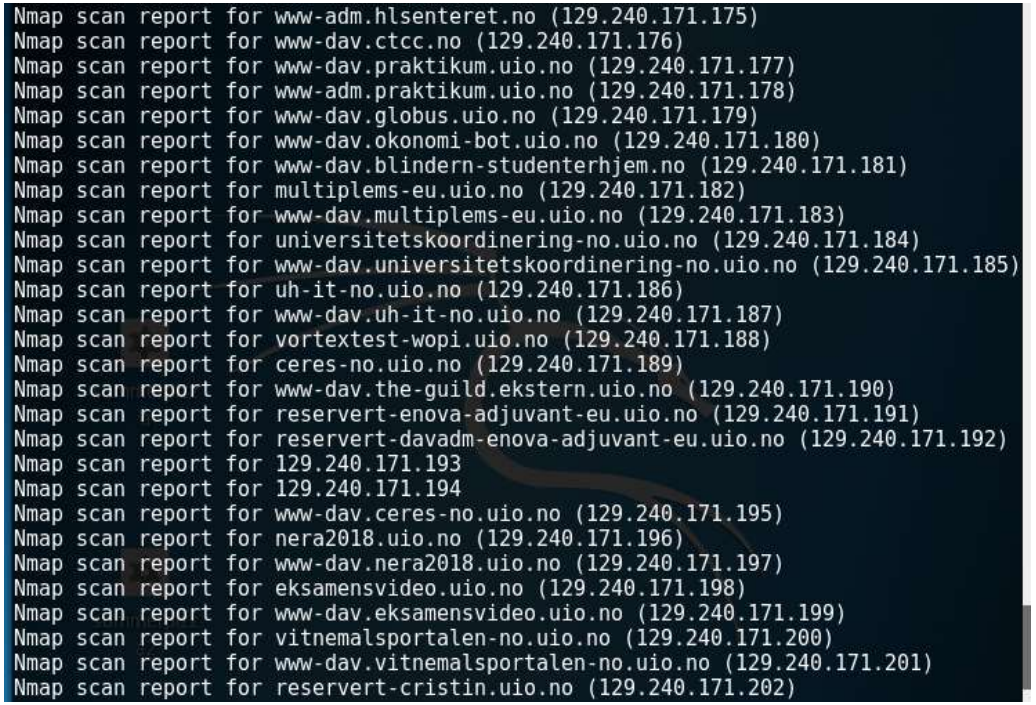
Nmap basic usage

With *nmap* it can be set:

- Type of scan (see detailed list later)
- Additional tests (e.g. version detection)
- Timing option (how many tries, how many parallel requests, max retries, scan delay, etc.)
- Hosts / host input
- Output result format (flat file, *xml*, etc.)
- Filtering (e.g. show only open ports)
- Scripts to run

Nmap - List scan

- With the *–sL* switch
- Has no connection with the hosts
- The *DNS* server is asked if a specific domain is registered in its database



Nmap - ping scan

- With the *–sP* switch
- *Nmap* pings all the specified hosts
- The available hosts are listed with their *MAC* address
- *ICMP* messages are not always allowed in a network

```
root@kali:~# nmap -sP 192.168.0.0/24

Starting Nmap 7.40 ( https://nmap.org ) at 2018-09-01 10:23 EDT
Nmap scan report for 192.168.0.1
Host is up (0.00090s latency).
MAC Address: F8:1A:67:BD:C1:BE (Tp-link Technologies)
Nmap scan report for 192.168.0.100
Host is up (0.0027s latency).
MAC Address: 00:1A:79:1C:5F:7F (Telecomunication Technologies)
Nmap scan report for 192.168.0.102
Host is up (0.013s latency).
MAC Address: F8:3F:51:2D:63:4B (Samsung Electronics)
Nmap scan report for 192.168.0.105
Host is up (0.039s latency).
MAC Address: F0:D5:BF:D2:D4:7B (Intel Corporate)
Nmap scan report for 192.168.0.106
Host is up (0.0014s latency).
MAC Address: C8:D3:FF:73:3D:F6 (Hewlett Packard)
Nmap scan report for 192.168.0.107
Host is up (0.017s latency).
MAC Address: 04:E5:36:DC:66:17 (Apple)
Nmap scan report for 192.168.0.101
Host is up.
Nmap done: 256 IP addresses (7 hosts up) scanned in 2.21 seconds
```

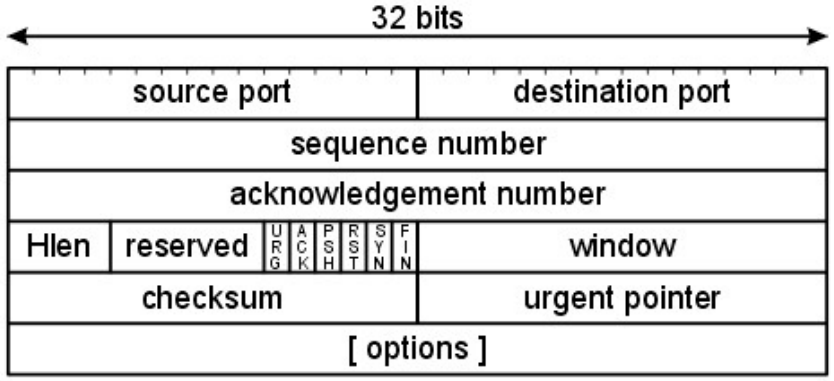
Layer 4 – Data transmission

Apart from sending short simple messages, bigger data blocks can be transmitted between the hosts. The data transfer is carried out in the 4th layer by using 2 different approaches:

- *UDP*: streaming the data (no guarantee that all data will arrive, but fast)
- *TCP*: the arrival of all data is guaranteed in the right order (trustworthy transmission, slower than *UDP*)

In addition, the data transmission is carried out using port numbers. One host can send and receive data in multiple channels using different port numbers for different services.

Layer 4 – TCP protocol



In order to ensure that the packages arrived in the right order the sequence number and the acknowledgement number are used.

TCP flags are for maintaining the connection status (*urg, ack, psh, rst, syn, fin*).

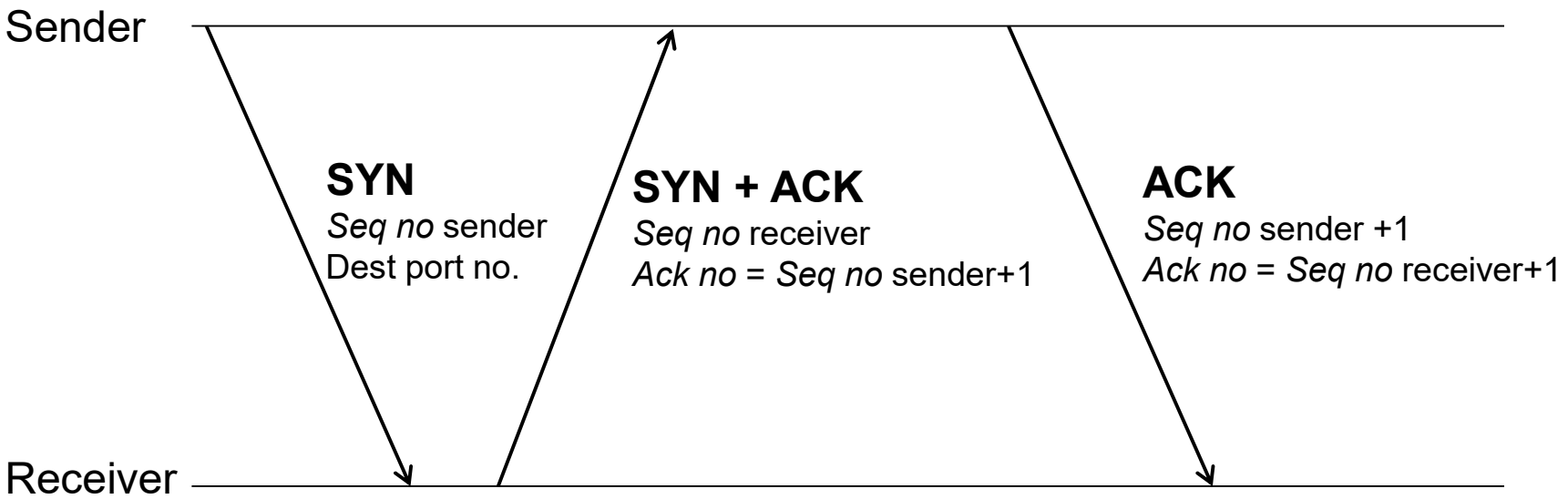
Layer 4 – TCP typical services

- *TCP* 80: *web http*
- *TCP* 443: *web https*
- *TCP* 20,21: *ftp*
- *TCP* 22: *ssh*
- *TCP* 25: *smtp*
- *TCP* 137,139,445: *netbios*
- *TCP* 3306: *mysql*
- *TCP* 3389: *remote desktop*
- *TCP* 5900: *VNC*

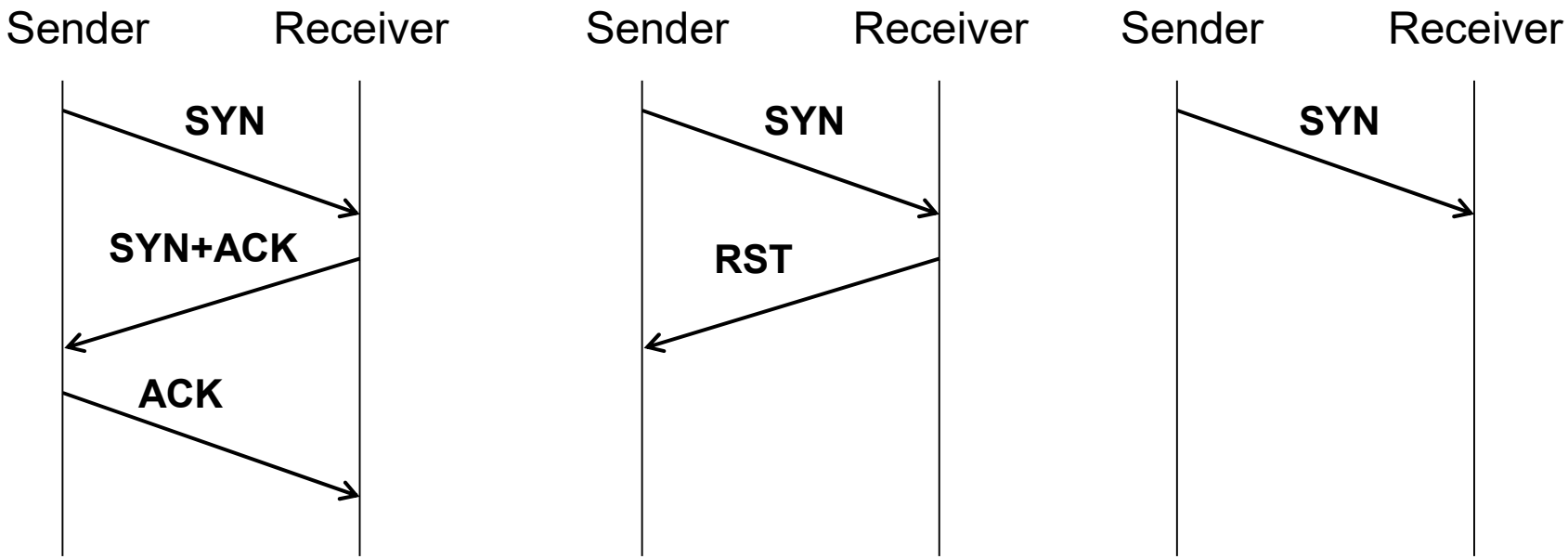
Remember that any service can be used in any port, these are only recommendations

Layer 4 – TCP 3-way handshake

TCP handshake is the process when a connection is about to be established in a specific port.



Tcp scan (full tcp scan)



Port is open **Port is closed** **Port is filtered**

Nmap carries out *tcp* scan with the *–sT* switch
Port numbers can be specified optionally
Example: *nmap –sT –p80,43 host*

Tcp scan (full tcp scan)

The number of possible ports is 65535, scanning all ports requires too much time (and too noisy).
We can reduce the port numbers by specifying them with the *–p* switch.
Without *–p nmap* will scan the 1024 most popular ports.

```
root@kali:~# nmap -sT 192.168.0.101-109

Starting Nmap 7.40 ( https://nmap.org ) at 2018-09-01
Nmap scan report for 192.168.0.101
Host is up (0.00016s latency).
All 1000 scanned ports on 192.168.0.101 are closed

Nmap scan report for 192.168.0.102
Host is up (0.0087s latency).
Not shown: 991 closed ports
PORT      STATE SERVICE
7676/tcp  open  imqbrokerd
8001/tcp  open  vcom-tunnel
8002/tcp  open  teradataordbms
8080/tcp  open  http-proxy
9999/tcp  open  abyss
32768/tcp open  filenet-tms
32769/tcp open  filenet-rpc
32770/tcp open  sometimes-rpc3
32771/tcp open  sometimes-rpc5
MAC Address: F8:3F:51:2D:63:4B (Samsung Electronics)

Nmap scan report for 192.168.0.103
Host is up (0.050s latency).
All 1000 scanned ports on 192.168.0.103 are filtered
MAC Address: F0:CB:A1:08:A6:E4 (Apple)

Nmap scan report for 192.168.0.105
Host is up (0.012s latency).
Not shown: 995 filtered ports
PORT     STATE SERVICE
902/tcp  open  iss-realsecure
912/tcp  open  apex-mesh
2701/tcp open  sms-rcinfo
2869/tcp open  icslap
5357/tcp open  wsdapi
MAC Address: F0:D5:BF:D2:D4:7B (Intel Corporate)
```

Operating System detection

Nmap's remote *OS* detection uses *TCP/IP* stack fingerprinting. Nmap sends a series of *TCP* and *UDP* packets to the remote host and examines practically every bit in the responses.

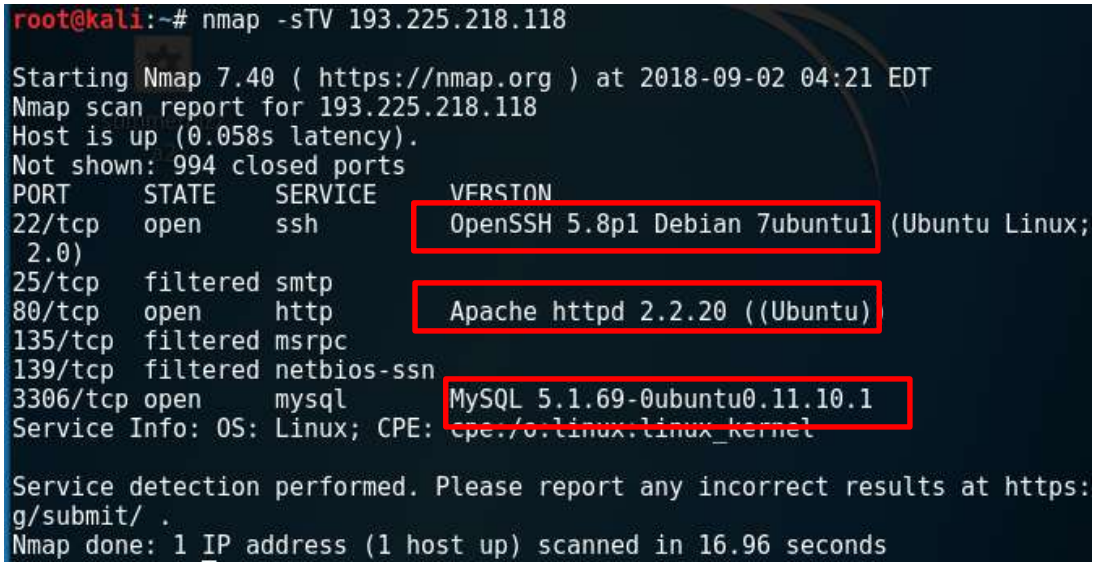
After performing dozens of tests such as *TCP ISN* sampling, *TCP* options support and ordering, *IP ID* sampling, and the initial window size check, *Nmap* compares the results to its *nmap-os-db* database of more than 2,600 known *OS* fingerprints and prints out the *OS* details if there is a match.

```
root@kali:~# nmap -O 193.225.218.118

Starting Nmap 7.40 ( https://nmap.org ) at 2018-09-02 04:16 EDT
Nmap scan report for 193.225.218.118
Host is up (0.059s latency).
Not shown: 994 closed ports
PORT     STATE    SERVICE
22/tcp   open     ssh
25/tcp   filtered smtp
80/tcp   open     http
135/tcp  filtered msrpc
139/tcp  filtered netbios-ssn
3306/tcp open     mysql
Device type: general purpose|broadband router|storage-misc|router|firewall|media de
vice|WAP
Running (JUST GUESSING): Linux 2.6.X|3.X|4.X (94%), HP embedded (91%), MikroTik Rou
terOS 6.X (90%), WatchGuard embedded (90%), AVM FritzOS 6.X (88%)
OS CPE: cpe:/o:linux:linux_kernel:2.6 cpe:/o:linux:linux_kernel:3 cpe:/h:hp:p2000_g
3 cpe:/o:mikrotik:routeros:6.32.1 cpe:/h:watchguard:xtm_525 cpe:/o:linux:linux_kern
el:4 cpe:/o:linux:linux_kernel:3.x cpe:/o:avm:fritzos:6.51
Aggressive OS guesses: Linux 2.6.32 - 3.1 (94%), OpenWrt 12.09-rc1 Attitude Adjustm
ent (Linux 3.3 - 3.7) (94%), Linux 3.2 (94%), Linux 2.6.32 - 3.13 (94%), Linux 2.6.
32 - 2.6.39 (92%), Linux 3.2 - 3.8 (92%), HP P2000 G3 NAS device (91%), Linux 3.5 (
90%), Linux 2.6.32 - 3.10 (90%), Linux 2.6.32 - 3.9 (90%)
No exact OS matches for host (test conditions non-ideal).

OS detection performed. Please report any incorrect results at https://nmap.org/sub
mit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.74 seconds
```

Service version detection

Version detection interrogates the ports to determine more about what is actually running. The *nmap-service-probes* database contains probes for querying various services and match expressions to recognize and parse responses.

Nmap tries to determine the service protocol, the version number, hostname, device, the *OS* family. With *banner grabbing* completely exact version numbers can be retrieved (*Banner* info can be modified).



Nmap scripting engine

Example: *nmap –sT –p21 –script==ftp-vuln-cve2010-4221 target*

Script output:

```
PORT   STATE SERVICE
21/tcp open  ftp
| ftp-vuln-cve2010-4221:
|   VULNERABLE:
|   ProFTPD server TELNET IAC stack overflow
|     State: VULNERABLE
|     IDs:  CVE:CVE-2010-4221  BID:44562  OSVDB:68985
|     Risk factor: High  CVSSv2: 10.0 (HIGH) (AV:N/AC:L/Au:N/C:C/I:C/A:C)
|     Description:
|       ProFTPD server (version 1.3.2rc3 through 1.3.3b) is vulnerable to
|       stack-based buffer overflow. By sending a large number of TELNET_IAC
|       escape sequence, a remote attacker will be able to corrupt the stack and
|       execute arbitrary code.
|     Disclosure date: 2010-11-02
|     References:
|       http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-4221
|       http://osvdb.org/68985
|       http://www.metasploit.com/modules/exploit/freebsd/ftp/proftp_telnet_iac
|       http://bugs.proftpd.org/show_bug.cgi?id=3521
|_      http://www.securityfocus.com/bid/44562
```

Other examples:

All scripts from a category: *nmap –sT –p21 –script==vuln target*

All scripts (carpet bombing!): *nmap –sT –p21 –script==all target*

Where are we in the process of ethical hacking?

- We have several general information about the target
- We have the technical details (domains, ip ranges)
- We mapped the target network and have an inventory (live hosts, responding services)

- What's next?
- We try to compromise services
  - Find a vulnerability
  - Exploit the vulnerability

How to start compromising a service?

What kind of services do we have to face from outside?

Web, Ftp, ssh, dns, mail (SMTP, POP3, IMAP, Exchange), VPN and many others

Typical services inside:

Netbios, SMB, Printer, RDP, DB services, LDAP, etc.

How to start compromising a service?

What kind of errors (vulnerabilities) can we expect?

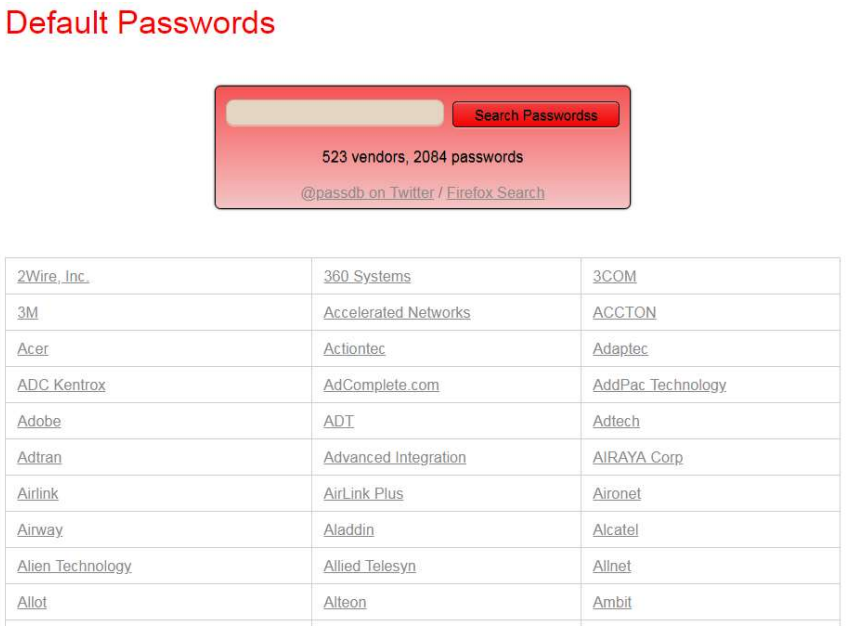
- Configuration related errors
  - Default credentials
  - Easy to guess credentials (we had information gathering before)
  - No or inappropriate protection against guessing (brute-force)
  - Unnecessary function
  - Privilege misconfigurations
  - Other configuration errors
- Software vulnerability related error
  - No input validation
  - Memory handling errors
  - Several others (see later)

How to start compromising a service?

- First use in the normal way
  - Is there any information disclosure?
  - Error messages, etc.
  - Restrictions
- Force it to error and obtain information
  - Provide invalid data
  - Use it in an invalid way
- Try factory defaults
- Brute-forcing
- Search for known exploits
- Service specific exploitations
- Unique ways

Factory defaults

- Default credentials
  - http://cirt.net
  - http://phenoelit.org/dpl/dpl.html
  - http://www.defaultpassword.com/’



- Default functions

Brute-forcing

- Trying out multiple combinations
- How to generate the options?
  - Random
  - Trying out all combinations
  - Using a list or dictionary

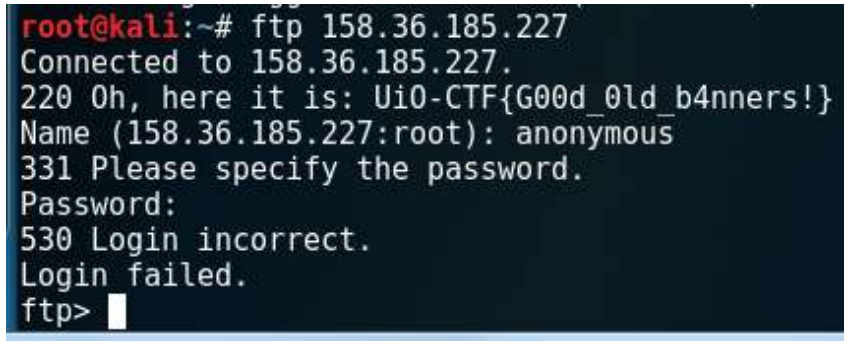
- Brute forcing tools
  - THC Hydra (ssh, ftp, http)

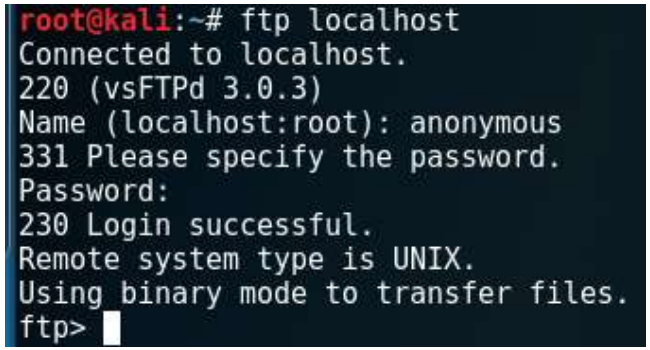
  Hydra was created by a hacker group The Hacker's choice. It is an universal brute-force tool that can be used for several protocols.
  - Ncrack
  - Medusa

What is an exploit?

An **exploit** (from the English verb *to exploit*, meaning "to use something to one's own advantage") is a piece of software, a chunk of data, or a sequence of commands that takes advantage of a bug or vulnerability to cause unintended or unanticipated behavior to occur on computer software, hardware, or something electronic (usually computerized). Such behavior frequently includes things like gaining control of a computer system, allowing privilege escalation, or a denial-of-service (DoS or related DDoS) attack.

Attacking ftp service: anonymous login





If anonymous login is enabled, anyone can log in (username: anonymous, password: arbitrary email)

anon_upload_enable, anon_other_write_enable settings are also important: e.g. if upload is enabled and the webroot is accessible attacking scripts can be uploaded.

Attacking ftp service: brute-forcing with Hydra

```
root@kali:~# hydra -t 2 -l admin -P pass.lst -vV localhost ftp
Hydra v8.3 (c) 2016 by van Hauser/THC - Please do not use in military or secret servic
 organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2018-09-07 07:16:46
[DATA] max 2 tasks per 1 server, overall 64 tasks, 5 login tries (l:1/p:5), ~0 tries p
r task
[DATA] attacking service ftp on port 21
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[ATTEMPT] target localhost - login "admin" - pass "1234" - 1 of 5 [child 0] (0/0)
[ATTEMPT] target localhost - login "admin" - pass "123456" - 2 of 5 [child 1] (0/0)
[ATTEMPT] target localhost - login "admin" - pass "iloveyou" - 3 of 5 [child 0] (0/0)
[ATTEMPT] target localhost - login "admin" - pass "qwerty" - 4 of 5 [child 1] (0/0)
[ATTEMPT] target localhost - login "admin" - pass "suzie" - 5 of 5 [child 0] (0/0)
[STATUS] attack finished for localhost (waiting for children to complete tests)
1 of 1 target completed, 0 valid passwords found
Hydra (http://www.thc.org/thc-hydra) finished at 2018-09-07 07:16:57
```

-l for single user –L user list (the list has to be named after)

-p for single password –P password list (the list file has to be named after)

-t parallel tries (default 16)

Attacking ssh service – brute force

Without the valid password:

```
root@kali:~# hydra -l uioctf -P pass.lst 193.225.218.118 -t 1 ssh
Hydra v8.3 (c) 2016 by van Hauser/THC - Please do not use in military or secret service
 organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2018-09-08 15:39:26
[WARNING] Restorefile (./hydra.restore) from a previous session found, to prevent overw
riting, you have 10 seconds to abort...
[DATA] max 1 task per 1 server, overall 64 tasks, 5 login tries (l:1/p:5), ~0 tries per
 task
[DATA] attacking service ssh on port 22
1 of 1 target completed, 0 valid passwords found
Hydra (http://www.thc.org/thc-hydra) finished at 2018-09-08 15:39:47
root@kali:~#
```

With the valid password:

```
root@kali:~# hydra -l uioctf -P pass.lst 193.225.218.118 -t 1 ssh
Hydra v8.3 (c) 2016 by van Hauser/THC - Please do not use in military or secret service
 organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2018-09-08 15:41:23
[DATA] max 1 task per 1 server, overall 64 tasks, 6 login tries (l:1/p:6), ~0 tries per
 task
[DATA] attacking service ssh on port 22
[22][ssh] host: 193.225.218.118   login: uioctf   password: ethicalhacking999
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2018-09-08 15:41:37
root@kali:~#
```

End of lecture