CS5333 Hands-on:
Cracking SSH Passwords

Disclaimer: Try on you own risk

- This hand-on is highly risky if
  - You run SSH service on your laptop computer
  - You permit “root login”
  - You permit “password authentication”
  - Your username is known by your friends
  - Your password is weak

- To mitigate the risk caused by this hands-on
  - Review your /etc/ssh/sshd_config
  - Use Public Key Authentication and don’t use Password Authentication
  - Make your password stronger
  - Don’t inform your IP address to anybody
  - Use VM instead of your native OS

Objective of this hands-on

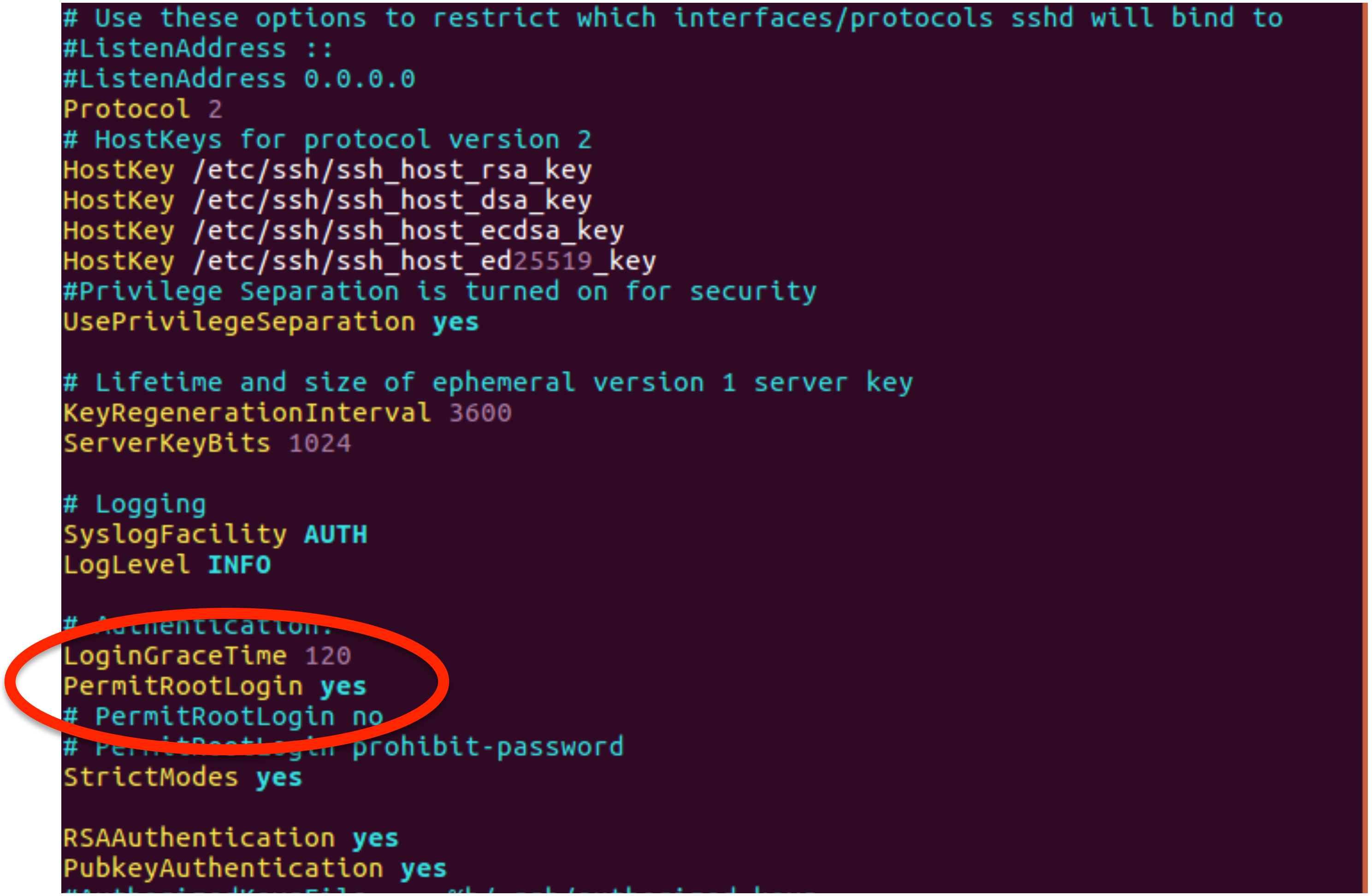
- For the users of securely configured SSH server
  - Check if your SSH server is really secure
- For the users who haven’t secure your SSH server
  - Realize what may happen to you
- For all
  - Awareness of importance of also securing other services

Recipe

- Unsecure SSH server configuration
- A password cracking tool
- A password dictionary

Unsecure SSH Server Configuration (1/3)

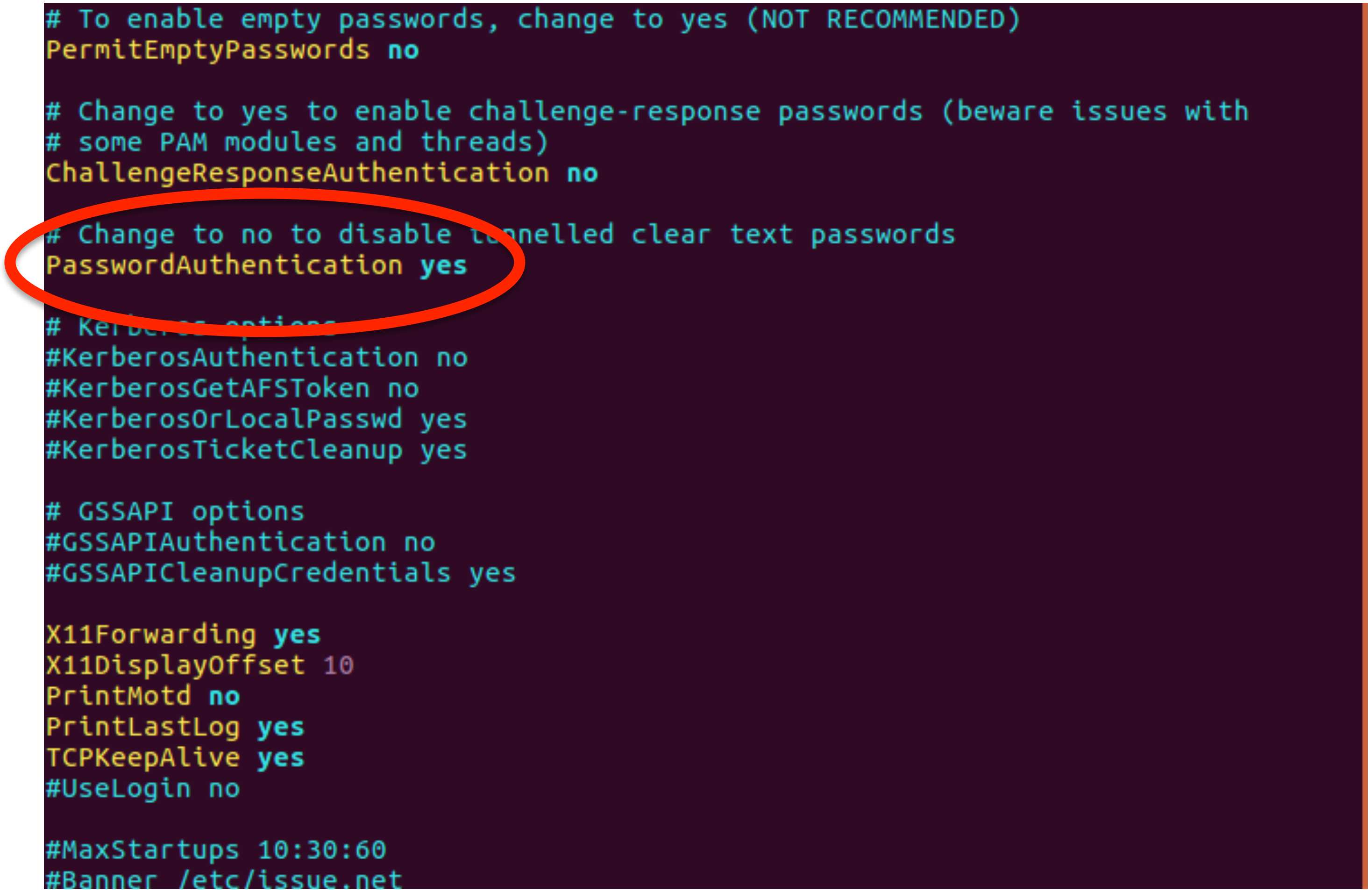
- /etc/ssh/sshd_config



```
# Use these options to restrict which interfaces/protocols sshd will bind to
#ListenAddress ::
#ListenAddress 0.0.0.0
Protocol 2
# HostKeys for protocol version 2
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key
HostKey /etc/ssh/ssh_host_ecdsa_key
HostKey /etc/ssh/ssh_host_ed25519_key
#Privilege Separation is turned on for security
UsePrivilegeSeparation yes

# Lifetime and size of ephemeral version 1 server key
KeyRegenerationInterval 3600
ServerKeyBits 1024

# Logging
SyslogFacility AUTH
LogLevel INFO

# Authentication:
LoginGraceTime 120
PermitRootLogin yes
# PermitRootLogin no
# PermitRootLogin prohibit-password
StrictModes yes

RSAAuthentication yes
PubkeyAuthentication yes
#AuthorizedKeysFile     %h/.ssh/authorized_keys
```

Unsecure SSH Server Configuration (2/3)

- /etc/ssh/sshd_config



```
# To enable empty passwords, change to yes (NOT RECOMMENDED)
PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
ChallengeResponseAuthentication no

# Change to no to disable tunnelled clear text passwords
PasswordAuthentication yes

# Kerberos options
#KerberosAuthentication no
#KerberosGetAFSToken no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes

# GSSAPI options
#GSSAPIAuthentication no
#GSSAPICleanupCredentials yes

X11Forwarding yes
X11DisplayOffset 10
PrintMotd no
PrintLastLog yes
TCPKeepAlive yes
#UseLogin no

#MaxStartups 10:30:60
#Banner /etc/issue.net
```

Unsecure SSH Server Configuration (3/3)

```
$ sudo service ssh restart
```

Password Cracking Tool

- Hydra, NCrack, Medusa

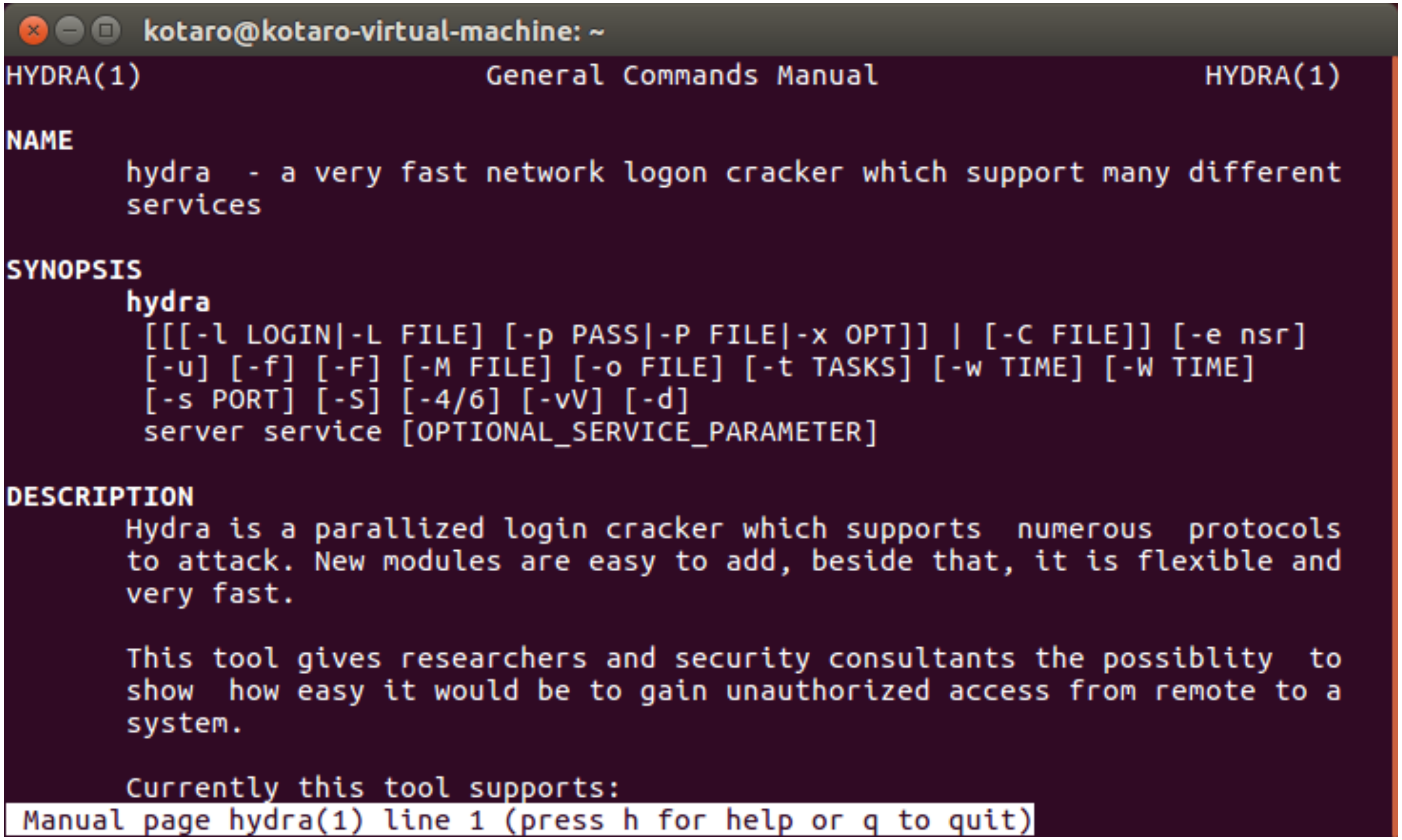
```
$ sudo apt-get update
$ sudo apt-get install hydra
```

List of bad passwords

- The Top 500 Worst Passwords of All Time https://gist.github.com/djaiss/4033452
- Can be found as Course Material of CS5333 Classroom

Hydra

- A password cracking tools (brute force attack)



kotaro@kotaro-virtual-machine: ~

```
HYDRA(1)                    General Commands Manual                   HYDRA(1)

NAME
       hydra  - a very fast network logon cracker which support many different
       services

SYNOPSIS
       hydra
        [[[-l LOGIN|-L FILE] [-p PASS|-P FILE|-x OPT]] | [-C FILE]] [-e nsr]
        [-u] [-f] [-F] [-M FILE] [-o FILE] [-t TASKS] [-w TIME] [-W TIME]
        [-s PORT] [-S] [-4/6] [-vV] [-d]
        server service [OPTIONAL_SERVICE_PARAMETER]

DESCRIPTION
       Hydra is a parallized login cracker which supports  numerous  protocols
       to attack. New modules are easy to add, beside that, it is flexible and
       very fast.

       This tool gives researchers and security consultants the possiblity  to
       show  how easy it would be to gain unauthorized access from remote to a
       system.

       Currently this tool supports:
 Manual page hydra(1) line 1 (press h for help or q to quit)
```

Executing Hydra for Cracking SSH

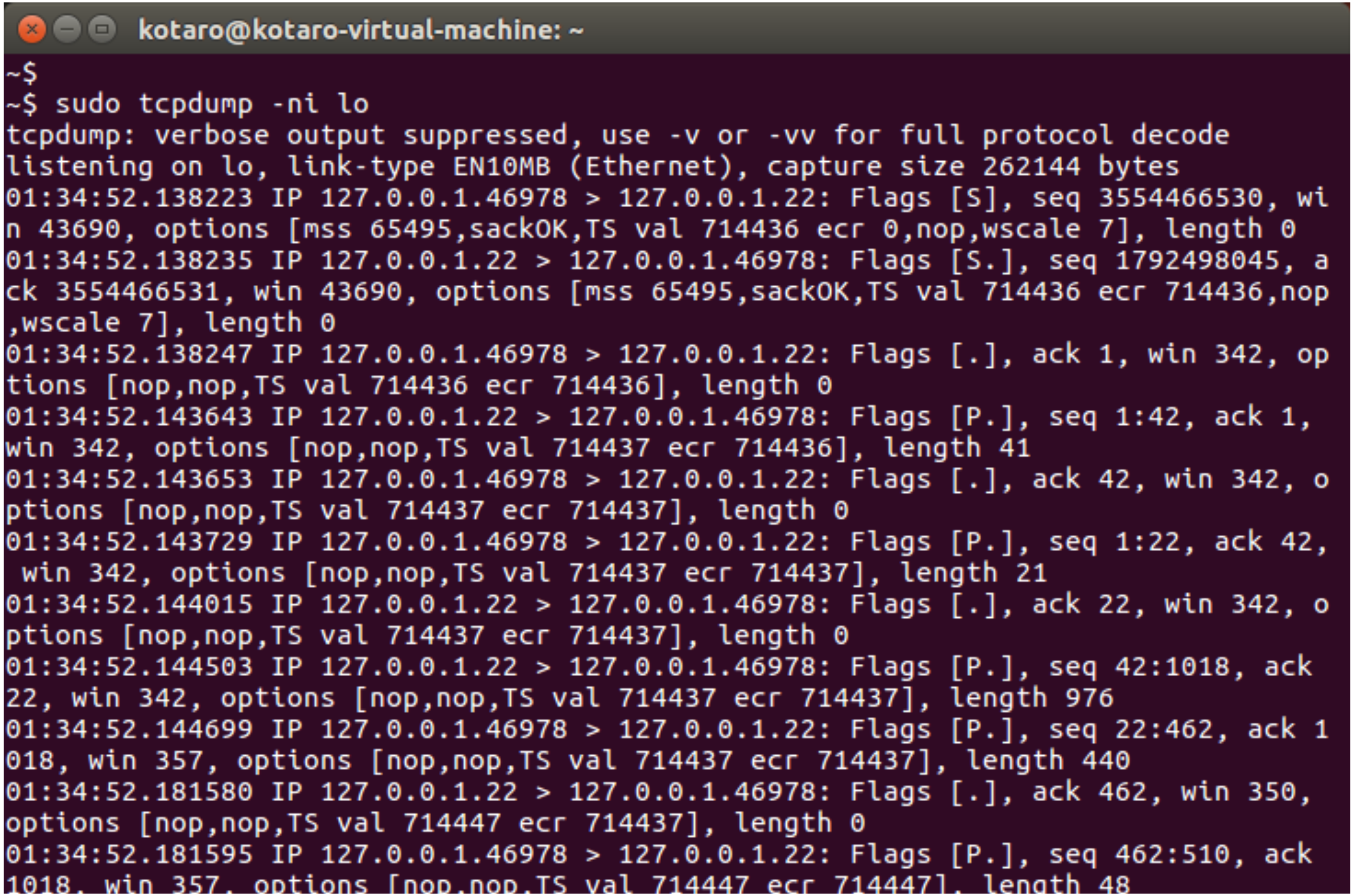
- Attacking one-by-one

```
$ hydra -l [username] -p [password]
               [IP address / hostname] ssh
```

- Automating attacks

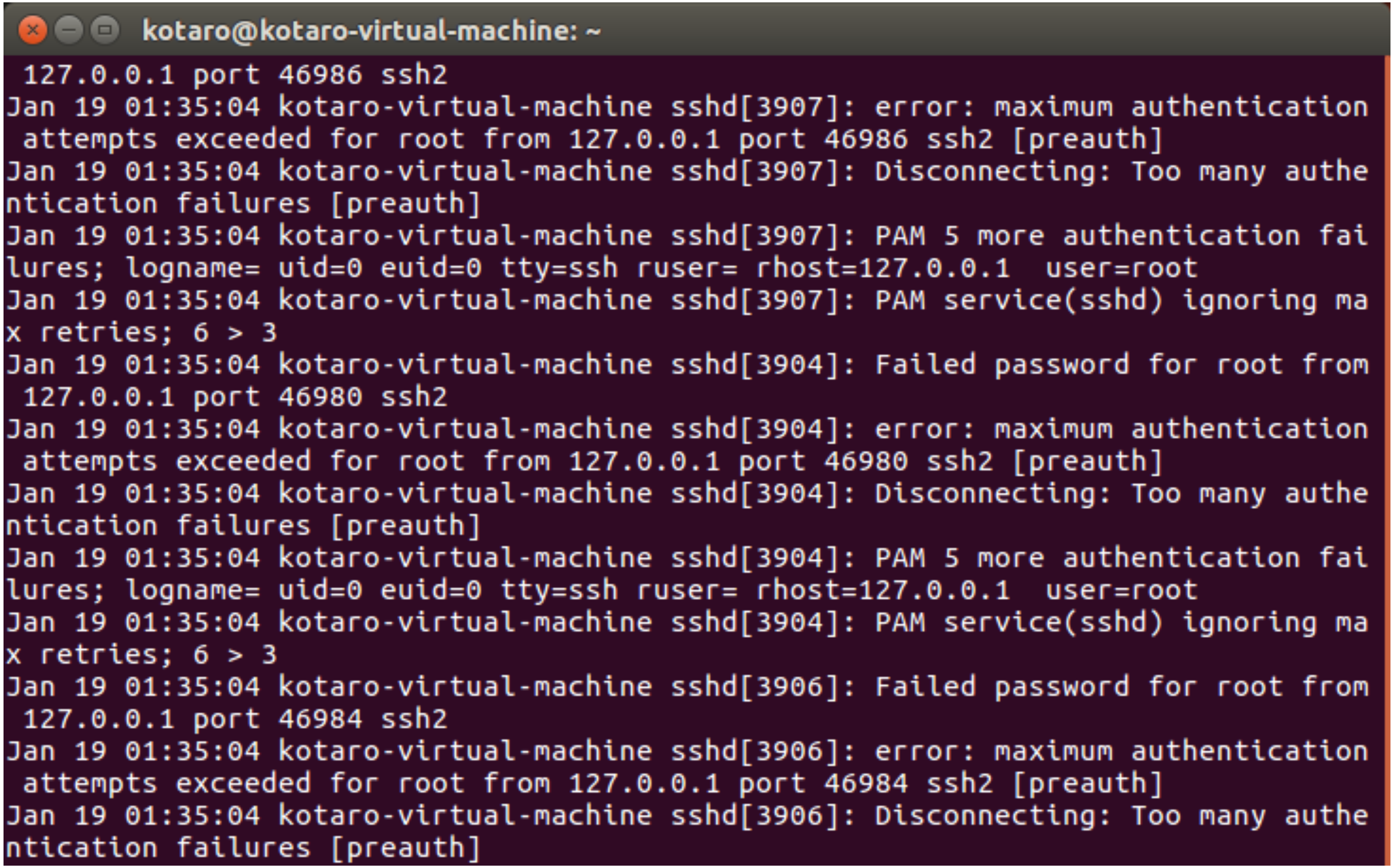
```
$ hydra -L [username File]
               -P [password file]
               [IP address / hostname] ssh
```

What does happen to the network?



```
~$
~$ sudo tcpdump -ni lo
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on lo, link-type EN10MB (Ethernet), capture size 262144 bytes
01:34:52.138223 IP 127.0.0.1.46978 > 127.0.0.1.22: Flags [S], seq 3554466530, wi
n 43690, options [mss 65495,sackOK,TS val 714436 ecr 0,nop,wscale 7], length 0
01:34:52.138235 IP 127.0.0.1.22 > 127.0.0.1.46978: Flags [S.], seq 1792498045, a
ck 3554466531, win 43690, options [mss 65495,sackOK,TS val 714436 ecr 714436,nop
,wscale 7], length 0
01:34:52.138247 IP 127.0.0.1.46978 > 127.0.0.1.22: Flags [.], ack 1, win 342, op
tions [nop,nop,TS val 714436 ecr 714436], length 0
01:34:52.143643 IP 127.0.0.1.22 > 127.0.0.1.46978: Flags [P.], seq 1:42, ack 1,
win 342, options [nop,nop,TS val 714437 ecr 714436], length 41
01:34:52.143653 IP 127.0.0.1.46978 > 127.0.0.1.22: Flags [.], ack 42, win 342, o
ptions [nop,nop,TS val 714437 ecr 714437], length 0
01:34:52.143729 IP 127.0.0.1.46978 > 127.0.0.1.22: Flags [P.], seq 1:22, ack 42,
 win 342, options [nop,nop,TS val 714437 ecr 714437], length 21
01:34:52.144015 IP 127.0.0.1.22 > 127.0.0.1.46978: Flags [.], ack 22, win 342, o
ptions [nop,nop,TS val 714437 ecr 714437], length 0
01:34:52.144503 IP 127.0.0.1.22 > 127.0.0.1.46978: Flags [P.], seq 42:1018, ack
22, win 342, options [nop,nop,TS val 714437 ecr 714437], length 976
01:34:52.144699 IP 127.0.0.1.46978 > 127.0.0.1.22: Flags [P.], seq 22:462, ack 1
018, win 357, options [nop,nop,TS val 714437 ecr 714437], length 440
01:34:52.181580 IP 127.0.0.1.22 > 127.0.0.1.46978: Flags [.], ack 462, win 350,
options [nop,nop,TS val 714447 ecr 714437], length 0
01:34:52.181595 IP 127.0.0.1.46978 > 127.0.0.1.22: Flags [P.], seq 462:510, ack
1018, win 357, options [nop,nop,TS val 714447 ecr 714447], length 48
```

How do you know from a log file?



```
 127.0.0.1 port 46986 ssh2
Jan 19 01:35:04 kotaro-virtual-machine sshd[3907]: error: maximum authentication
 attempts exceeded for root from 127.0.0.1 port 46986 ssh2 [preauth]
Jan 19 01:35:04 kotaro-virtual-machine sshd[3907]: Disconnecting: Too many authe
ntication failures [preauth]
Jan 19 01:35:04 kotaro-virtual-machine sshd[3907]: PAM 5 more authentication fai
lures; logname= uid=0 euid=0 tty=ssh ruser= rhost=127.0.0.1  user=root
Jan 19 01:35:04 kotaro-virtual-machine sshd[3907]: PAM service(sshd) ignoring ma
x retries; 6 > 3
Jan 19 01:35:04 kotaro-virtual-machine sshd[3904]: Failed password for root from
 127.0.0.1 port 46980 ssh2
Jan 19 01:35:04 kotaro-virtual-machine sshd[3904]: error: maximum authentication
 attempts exceeded for root from 127.0.0.1 port 46980 ssh2 [preauth]
Jan 19 01:35:04 kotaro-virtual-machine sshd[3904]: Disconnecting: Too many authe
ntication failures [preauth]
Jan 19 01:35:04 kotaro-virtual-machine sshd[3904]: PAM 5 more authentication fai
lures; logname= uid=0 euid=0 tty=ssh ruser= rhost=127.0.0.1  user=root
Jan 19 01:35:04 kotaro-virtual-machine sshd[3904]: PAM service(sshd) ignoring ma
x retries; 6 > 3
Jan 19 01:35:04 kotaro-virtual-machine sshd[3906]: Failed password for root from
 127.0.0.1 port 46984 ssh2
Jan 19 01:35:04 kotaro-virtual-machine sshd[3906]: error: maximum authentication
 attempts exceeded for root from 127.0.0.1 port 46984 ssh2 [preauth]
Jan 19 01:35:04 kotaro-virtual-machine sshd[3906]: Disconnecting: Too many authe
ntication failures [preauth]
```

Points to Think

- Preparedness
  - How does your operating system react to the attacks?
  - Anyway, attacks come to your computer. What is the fundamental solution?
  - Do you assignment
- Compliance
  - Don’t try this to anybody rather than yourself.
- Imagination
  - Is it only about SSH? What to do for the other services?