

실습 개요 :

이번 실습은 ARP 스푸핑, DNS 스푸핑, SSL 스니핑과 같은 중간자 공격(MITM) 기법을 직접 수행해 봄으로써, 네트워크 환경에서 발생할 수 있는 보안 위협을 체험하고 원리와 대응을 이해함

공격자는 피해자(Client)와 인터넷 사이의 중간에 위치한 게이트웨이를 공격 대상으로 하여, ARP 리다이렉션을 통해 네트워크 흐름을 가로채고, DNS 응답을 위조함으로써 피해자의 웹 요청을 공격자 자신에게 유도함

실습 환경

- 공격자 : 칼리(192.168.174.131)
- 피해자(클라) : Windows7 (192.168.174.130)
- 서버(게이트웨이) : 192.168.174.2

1. SSL 통신 확인

- 전북대학교 웹사이트 접속 테스트
- HTTPS 인증서 정보 확인

2. dnsspoof.hosts 파일 설정

- 위조 DNS 응답 설정
- 공격자 IP로 리다이렉트 구성

3. SSL 위조 인증서 생성 및 webmitm 실행

- webmitm을 통한 인증서 생성
- 공격자 SSL 서버 구동

4. ARP 리다이렉션 공격 및 패킷 릴레이

- arpspoof를 통한 MITM 구성
- fragrouter를 통한 패킷 전달

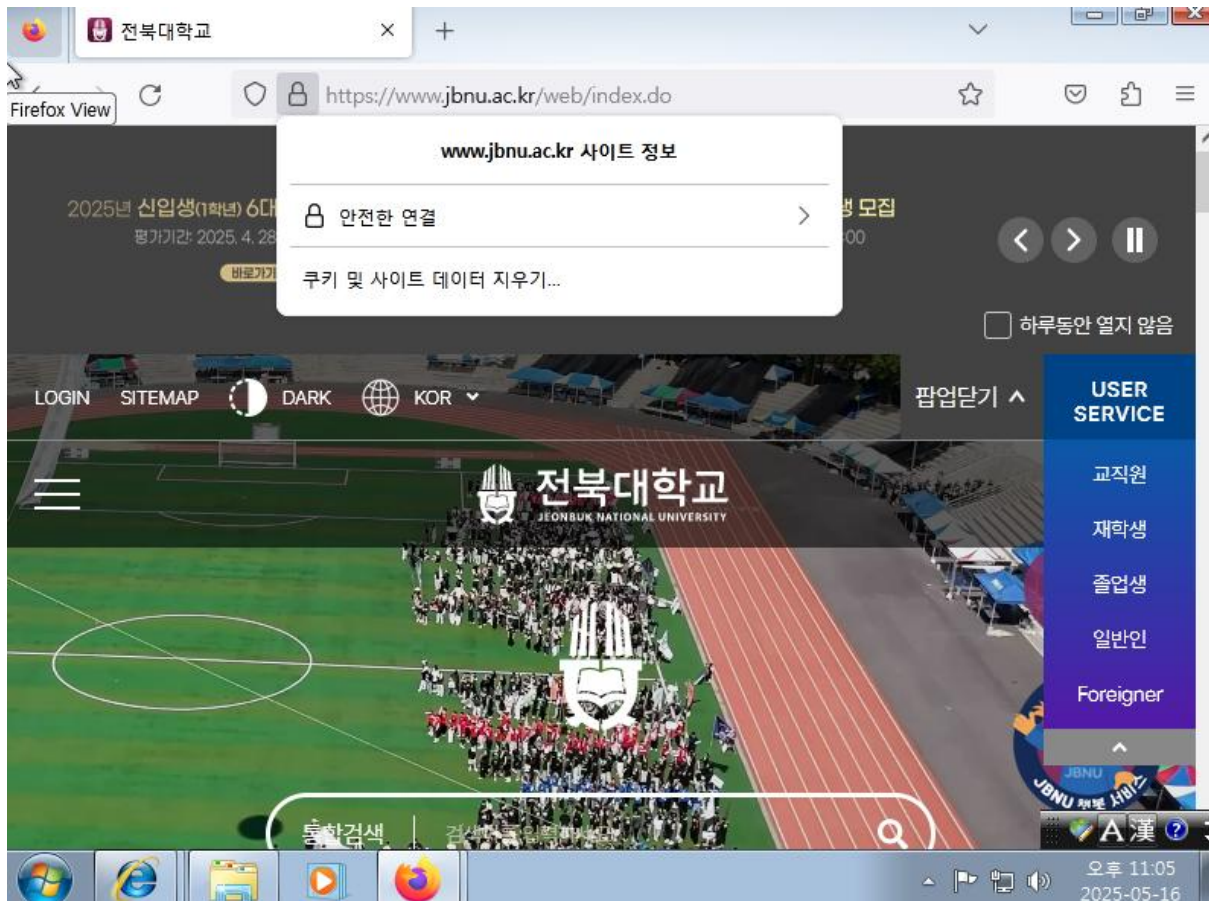
5. DNS 스푸핑 공격

- dnsspoof 실행
- 피해자 요청에 대해 위조된 DNS 응답 제공

6. 클라이언트에서 접속 시도

- 피해자가 공격자 SSL 서버로 접속
- 브라우저에서 인증서 경고 발생 확인

실습 과정



윈도우 7 환경에서 크롬이 다운로드 되지 않아 파이어 폭스 esr 115 버전으로 다운로드 하여 SSL 확인



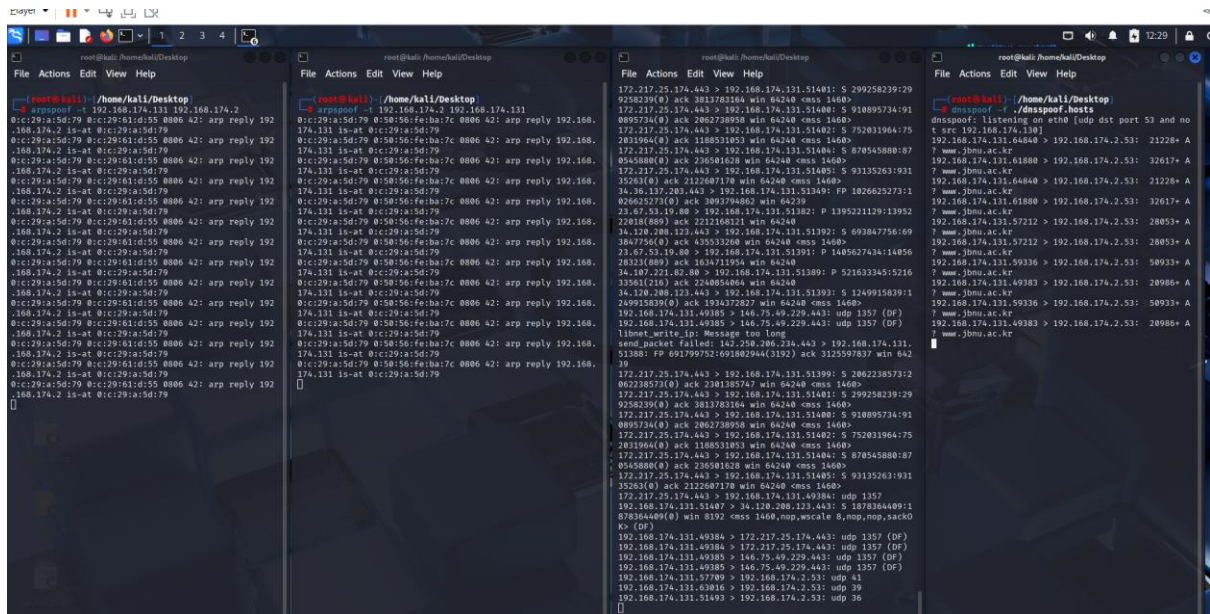
DNS 스푸핑을 위해 파일을 설정 dnsspoof.hosts 파일은 피해자가 *.jbnu.ac.kr 에 접속하려고 하면 192.168.174.130 인 공격자(kali)의 IP 로 리다이렉트 되도록 설정하였음

```
(root@kali)-[/home/kali/Desktop]
# webmitm
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:KR
State or Province Name (full name) [Some-State]:JEONJU
Locality Name (eg, city) []:JEONJU
Organization Name (eg, company) [Internet Widgits Pty Ltd]:JBNU
Organizational Unit Name (eg, section) []:CSAI
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Certificate request self-signature ok
subject=C=KR, ST=JEONJU, L=JEONJU, O=JBNU, OU=CSAI
webmitm: certificate generated
webmitm: relaying transparently
^C
```

```
(root@kali)-[/home/kali/Desktop]
# cat webmitm.crt
-----BEGIN PRIVATE KEY-----
MIICdwIBADANBgkqhkiG9w0BAQEFAASCAmEwggJdAgEAAoGBAMtgVWBSmfZ71LIC
sI43sa4V36MnPx+RdJC1llrBEx6lnOyKJb0x8N39BilmpyakbMDjQsb2QvW0mpt
xtB8f3oLYfcLQvjdcdPOpxsQsRJwegl/t58Pig3Ws85tmhh3LPIM17AaZ1uxWGWd
KVKtWo2+ggTdui3H9N1KAB81K8xfAgMBAAECgYAWQaCWPfPjaYHJSUoRBj9HkdU7
gZwGVOcWxW6C2AZCNCGMMUW/C4j3QCWGMzLXo77TDQvNwj1pri4NPNA9Z7thfXx
ULq07dORA4BPx0++iHjl3HHZmeRh/VaySMrx754fvLx3kS1xrATKeIjYbbUDEd4L
/Gfv50nGvGqRWEnHAQJBAP+Zs60IYMffffP+09C1rGBkddfp04+cooJDOrXg0ViAx
+aM2YjORTHxs923yoCcR+NpzFqadH7IVa32/gy7wvyECQQDLsbrrk+ngbzPqUUnh
uEbcvHB+I0SpYW5aksJ1SD/keH6bKpZfu4s6ZmpsxnCDVwhLiDmoMizlIEcsAnA7
hJt/AkEA1gt+/e5ALeEoKlMqhEZk3dNgEwexrsjE/b82Ya1iWn+/UECgqBKLfL87
zWUbXNyeC0HJvClke5uYcqviTazqQQJAHJAxq/Mz6prtZsA7YgGbPjzWcXm0mSPm
e6D0z8lv3DPhXRTdvVLpNSLDZzl0cZZ0vVK+8w0GKqMaF9vHrTDtNQJBAObQMu51
LpQ9g5SRZecNuo7HBB8aFIAZigcp5B9v6Cxt3bI/Z0ItXx+S+d8BuG+NDVTxB+QL
YDuevxLhp6j4WfU=
-----END PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
MIICRDCCAa2gAwIBAgIUbho0c4EYCdXSxyiL1eyUslt1GvQwDQYJKoZIhvcNAQEL
BQAwTTElMAkGA1UEBhMCS1IxZDZANBgNVBAGMBkpFT05KVTEPMA0GA1UEBwwGSkVP
TkpVMQ0wCwYDVQQKDARKQk5VMQ0wCwYDVQQLDARU0FJMB4XDTE1MDUxNjE0MjQ1
NV0XDTI1MDUxNjE0MjQ1NVowTTElMAkGA1UEBhMCS1IxZDZANBgNVBAGMBkpFT05K
VTEPMA0GA1UEBwwGSkVPtKpVMQ0wCwYDVQQKDARKQk5VMQ0wCwYDVQQLDARU0FJ
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDLyFVgUpn2e9SyArCON7GuFd+j
Jz8/kXSQtZZawRMRpZzsiiWzsFDd/QSJZsKcmpGzA40LG9kL1tJqcbcbQFH96C2H3
C0L43XAZzqcbELESchoJf7efD4oN1rP0bZoYdyzyDNewGmdbsVhlnSlSk8KNvoIE
3botx/TdSgAfNSvMXwIDAQABoyEwHzAdBgNVHQ4EFgQUHW6pmcSkcV1WmV0U4KlA
VByrE9MwDQYJKoZIhvcNAQELBQADgYEAPFh4CcmiEsnBnFACwxRXQ8l3h3uD10JV
QosQ4L9anyrmAsNXsTQEz8yUryRhFWmLXS0cRwlNrfS3KfI9PKNLHqY8hyfSn4T2
Dgj6r3mGpiIDNXgFSHj0Idq3zTP8BG5X4ldnNlyMm9jl4cMP8H4tRWHI1mYS7Bt0
MDDz3d40eyc=
-----END CERTIFICATE-----
```

피해자에게 가짜 HTTPS 인증서를 제공하고, 공격자가 중간에서 TLS 연결을 중계함으로써 SSL 스니핑을 가능하게 한다. 피해자의 브라우저에서는 “안전하지 않은 연결” 경고가 나오게 됨

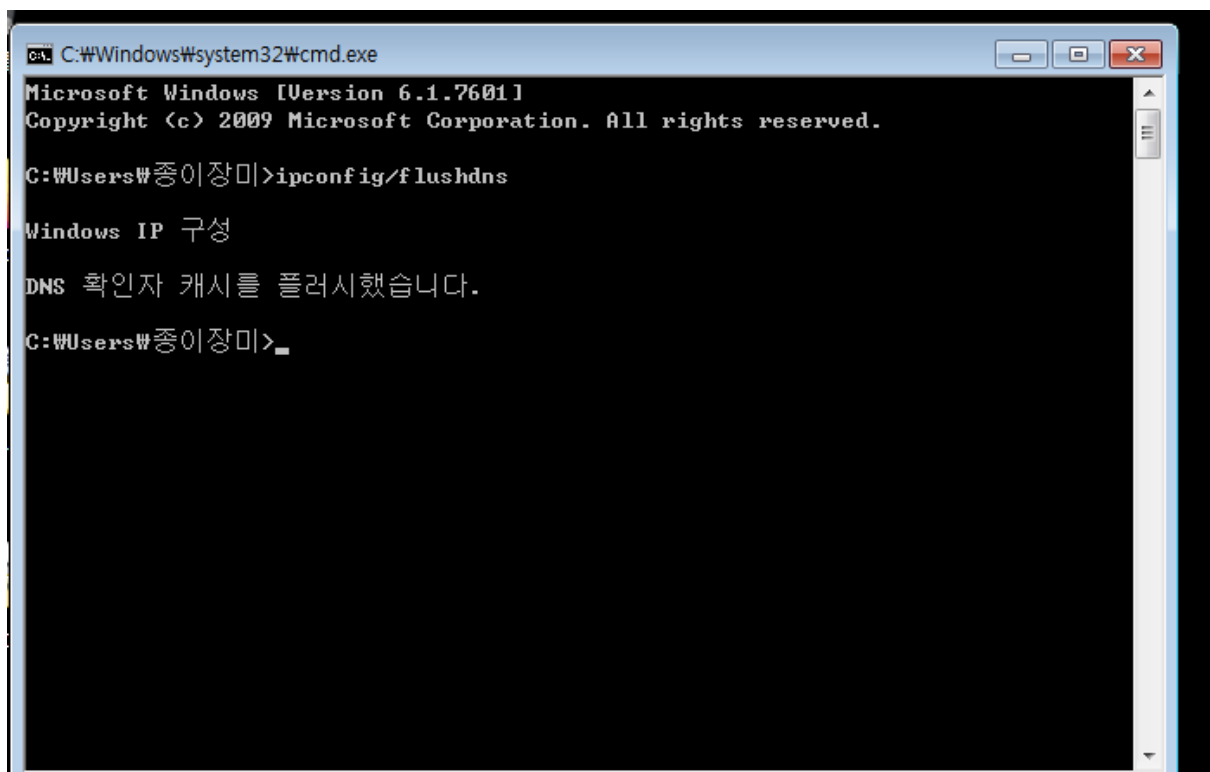


첫 번째 터미널 : 피해자에게 칼리가 게이트웨이라고 속이는 ARP 스푸핑

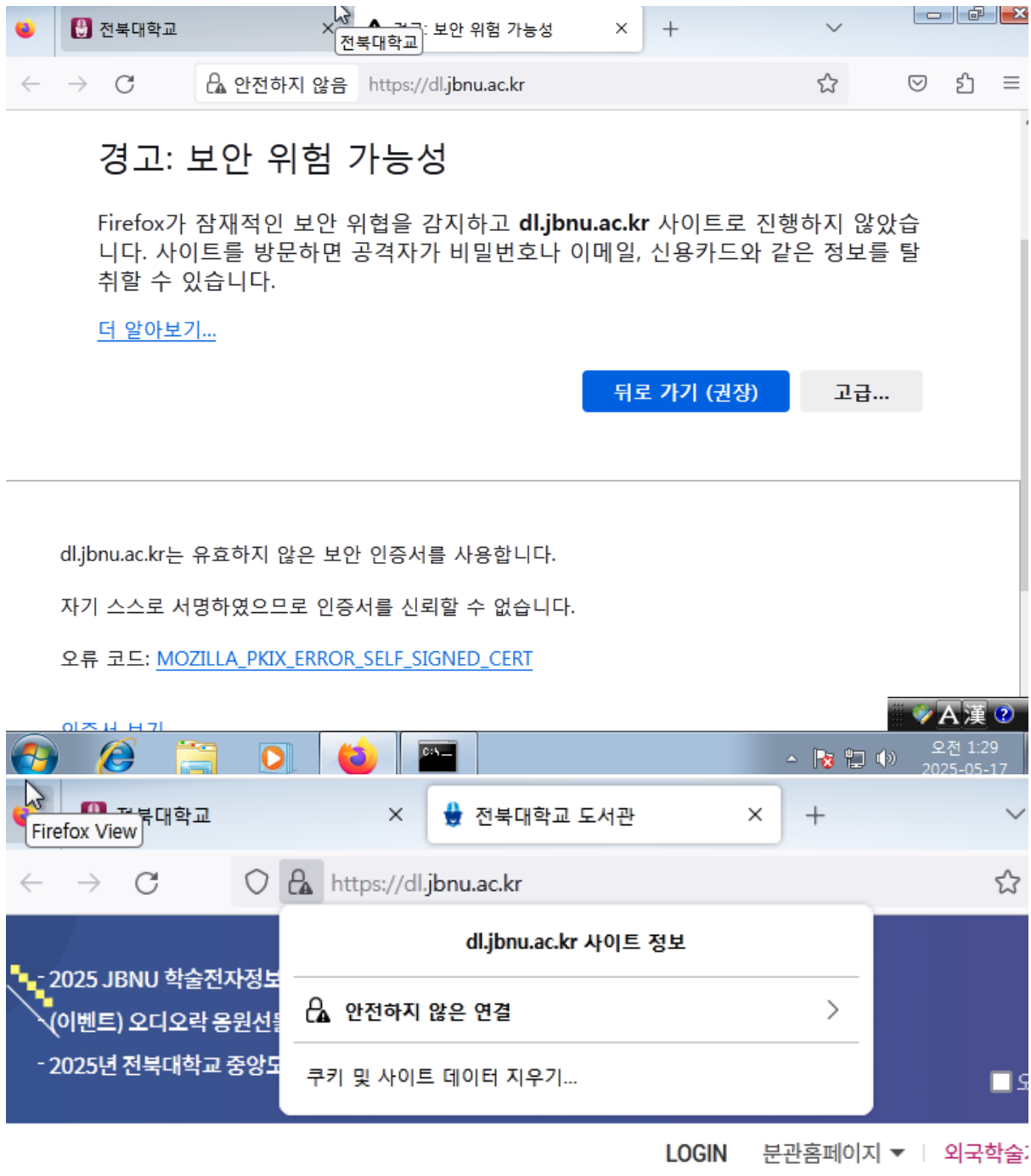
두 번째 터미널: 게이트웨이에게 칼리가 클라이언트(피해자)라고 속이는 ARP 스푸핑

세 번째 터미널 : 중간자 위치에서 릴레이

네 번째 터미널 : 피해자의 DNS 요청에 가짜 IP 응답을 반환하는 DNS 스푸핑



No.	Time	Source	Destination	Protocol	Length	Info
25207	00:59:273207	192.168.174.131	192.168.174.2	TCP	85	Standard query xccddid A push.services.mozilla.com
25208	00:59:283544	192.168.174.131	192.168.174.2	DNS	107	Destination unreachable (Port unreachable)
25209	00:59:291441	192.168.174.131	192.168.174.2	TCP	208	Standard query response 0xd45a A www.youtube.com CNAME youtube-u-l.google.com A 142.250.76.142 A 172.217.25.174 A 142.250.206.142
25210	00:59:2958704	192.168.174.2	192.168.174.131	DNS	193	Standard query response 0xc6407 A jnn-pa.googleapis.com A 142.250.207.106 A 172.217.25.170 A 172.217.161.202 A 172.217.161.234 A
25211	00:59:3026384	192.168.174.131	192.168.174.2	DNS	81	Standard query response 0xc3515 A jnn-pa.googleapis.com
25212	00:59:30976341	192.168.174.131	192.168.174.2	TCP	208	Standard query response 0xd45a A www.youtube.com CNAME youtube-u-l.google.com A 142.250.207.110 A 142.250.76.142 A 172.217.161.202
25213	00:59:315696	192.168.174.2	192.168.174.131	TCP	103	Standard query response 0x7472 AAAA push.google.com AAAA 2046:6880:408a:8041::208e
25214	00:59:3215768	192.250.160.0	192.168.174.131	TCP	1514	TCP Retransmission 443 - 35861 [ACK] Seq=1 Win=64240 Len=160
25215	00:59:403601	192.168.174.131	192.168.174.2	DNS	75	Standard query 0xd45a A www.youtube.com
25216	00:59:4034438	192.168.174.131	192.168.174.2	DNS	85	Standard query 0xcdd4d A push.services.mozilla.com
25217	00:59:4082958	192.168.174.131	192.168.174.2	DNS	81	Standard query 0xc3515 A jnn-pa.googleapis.com
25218	00:59:407769591	192.168.174.131	192.168.174.2	DNS	83	Standard query 0xcdd4d A push.services.mozilla.com
25219	00:59:40835509	192.168.174.2	192.168.174.131	TCP	221	Destination unreachable (Port unreachable)
25220	00:59:40926965	192.168.174.2	192.168.174.131	TCP	161	Standard query response 0xc3515 A jnn-pa.googleapis.com A 142.250.76.138 A 142.250.206.234 A 172.217.25.170 A 142.250.206.202 A
25221	00:59:4047851	192.168.174.131	192.168.174.2	DNS	203	Standard query response 0xd45a A www.youtube.com CNAME youtube-u-l.google.com A 142.250.207.110 A 172.217.161.206 A 142.250.206.142
25222	00:59:40905146	192.168.174.2	192.168.174.131	DNS	101	Standard query response 0xcdd4d A push.services.mozilla.com A 34.107.243.93
25223	00:59:45746791	192.168.174.2	192.168.174.131	DNS	181	Standard query response 0xcdd4d A push.services.mozilla.com A 34.107.243.93
25224	00:59:45954959	192.168.174.2	192.168.174.131	DNS	177	Standard query response 0xc3515 A jnn-pa.googleapis.com A 142.250.76.138 A 142.250.207.106 A 172.217.25.170 A 142.250.206.202 A
25225	00:59:46229841	192.168.174.131	192.168.174.2	TCP	60	TCP Keep-Alive 51470 - 443 [ACK] Seq=518 Acker=64240 Win=64240 Len=0
25226	00:59:471088959	192.168.174.2	192.168.174.131	NBNS	10	Refresh NB WIN-EPOLTRD3C055
25227	00:59:48116101	192.168.174.131	192.168.174.2	TCP	2077	TCP Retransmission 51470 - 443 [ACK] Seq=518 Acker=64240 Win=64240 Len=202
25228	00:59:48111601	192.168.174.131	146.75.50.133	TCP	60	TCP Keep-Alive 51576 - 80 [ACK] Seq=1769 Acker=2037 Win=63732 Len=1
25229	00:59:48118094	192.168.174.131	146.75.50.133	TCP	60	TCP Keep-Alive 51577 - 80 [ACK] Seq=747 Acker=1964 Win=64240 Len=1
25230	00:59:48118931	192.168.174.131	34.126.208.123	TCP	60	TCP Keep-Alive 51487 - 443 [ACK] Seq=427 Acker=4294963782 Win=64240 Len=1
25231	00:59:48209913	192.168.174.131	34.126.208.100	TCP	60	TCP Keep-Alive 51468 - 443 [ACK] Seq=427 Acker=4294961200 Win=64240 Len=1
25232	00:59:48209951	192.168.174.131	58.229.180.88	TCP	60	TCP Keep-Alive 51478 - 443 [ACK] Seq=427 Acker=4294961200 Win=64240 Len=1
25233	00:59:48217117	192.168.174.131	58.229.180.88	TCP	60	TCP Keep-Alive 51471 - 443 [ACK] Seq=427 Acker=4294961200 Win=64240 Len=1
25234	00:59:48217216	192.168.174.131	58.229.180.88	TCP	60	TCP Keep-Alive 51463 - 443 [ACK] Seq=427 Acker=4294961200 Win=64240 Len=1
25235	00:59:48212346	192.168.174.131	58.229.180.88	TCP	60	TCP Keep-Alive 51464 - 443 [ACK] Seq=427 Acker=4294961200 Win=64240 Len=1
25236	00:59:483144445	192.168.174.131	58.229.180.88	TCP	60	TCP Keep-Alive 51465 - 443 [ACK] Seq=427 Acker=4294961200 Win=64240 Len=1
25237	00:59:483140454	192.168.174.131	58.229.180.88	TCP	60	TCP Keep-Alive 514



공격자가 자체 생성한 SSL 인증서를 통해 피해자의 HTTPS 접속 요청을 가로챘고 이로 인해 브라우저에서 인증서 경고창이 띄워진걸 볼 수 있고 인증서가 신뢰되지 않음을 알 수 있음

사용자가 경고를 무시하고 접속하면 공격자는 HTTPS 내용을 가로챌 수 있음

오류 메시지인 **MOZILLA_PKIX_ERROR_SELF_SIGNED_CERT** 이 메시지는 webmitm 에서 생성된 인증서임을 알 수 있음

```
Player ▾ || 🔄 📄 🖥️ 🖨️
```

```
root@kali: /home/kali/Des
File Actions Edit View Help
File Actions Edit View Help

(root@kali)~[/home/kali/Desktop]
# # attacker (kali)
ssldump -a -d -r packet.pcap -k webmitm.crt > wireshark_dec.txt
Not enough data. Found 294 bytes (expecting 32767)
Not enough data. Found 294 bytes (expecting 32767)
Not enough data. Found 294 bytes (expecting 32767)
Not enough data. Found 294 bytes (expecting 32767)
Not enough data. Found 294 bytes (expecting 32767)
Not enough data. Found 294 bytes (expecting 32767)
Not enough data. Found 294 bytes (expecting 32767)
Not enough data. Found 294 bytes (expecting 32767)
Not enough data. Found 294 bytes (expecting 32767)
Not enough data. Found 294 bytes (expecting 32767)
Not enough data. Found 294 bytes (expecting 32767)
Not enough data. Found 294 bytes (expecting 32767)
Not enough data. Found 294 bytes (expecting 32767)
Not enough data. Found 294 bytes (expecting 32767)
Not enough data. Found 294 bytes (expecting 32767)
Error: short handshake length: expected 9277543 got 957
Not enough data. Found 109 bytes (expecting 32767)
Not enough data. Found 109 bytes (expecting 32767)
Not enough data. Found 109 bytes (expecting 32767)
Not enough data. Found 110 bytes (expecting 32767)
Not enough data. Found 109 bytes (expecting 32767)
Not enough data. Found 110 bytes (expecting 32767)
1 inactive connection(s) cleaned from connection pool
1 inactive connection(s) cleaned from connection pool
1 inactive connection(s) cleaned from connection pool
1 inactive connection(s) cleaned from connection pool
3 inactive connection(s) cleaned from connection pool
2 inactive connection(s) cleaned from connection pool
8 inactive connection(s) cleaned from connection pool
4 inactive connection(s) cleaned from connection pool
1 inactive connection(s) cleaned from connection pool
3 inactive connection(s) cleaned from connection pool
2 inactive connection(s) cleaned from connection pool
2 inactive connection(s) cleaned from connection pool
1 inactive connection(s) cleaned from connection pool
1 inactive connection(s) cleaned from connection pool
Cleaned 38 remaining connection(s) from connection pool

(root@kali)~[/home/kali/Desktop]
# ls
dns2tcpd_config dnsspoof.hosts dnsspoof.hosts.save packet.pcap pra.pcapng webmitm.crt wireshark_dec.txt
```

패킷 복호화를 위해 ssldump 를 사용하였으나 대부분 세션에서 Not enough data 가 출력되었음
여러가지 시도를 해보았지만 계속 동일하게 나와 그대로 실습을 진행하였음

```
(root@kali)-[/home/kali/Desktop]
# cat wireshark_dec.txt
New TCP connection #9: 192.168.174.131(51685) ↔ 192.168.174.130(443)
9 1 0.0036 (0.0036) C>S Handshake
  ClientHello
    Version 3.3
    resume [32]=
      2b ff 78 bd a3 2b 09 aa e0 c3 b3 57 80 88 7b 9e
      bc 07 d8 35 92 c4 ee 85 e0 f1 9a 6b 11 a7 3c ad
    cipher suites
      TLS_AES_128_GCM_SHA256
      TLS_CHACHA20_POLY1305_SHA256
      TLS_AES_256_GCM_SHA384
      TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
      TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
      TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256
      TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256
      TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
      TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
      TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
      TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
      TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
      TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
      TLS_RSA_WITH_AES_128_GCM_SHA256
      TLS_RSA_WITH_AES_256_GCM_SHA384
      TLS_RSA_WITH_AES_128_CBC_SHA
      TLS_RSA_WITH_AES_256_CBC_SHA
    compression methods
      NULL
    extensions
      server_name
        host_name: www.jbnu.ac.kr
      extended_master_secret
      renegotiation_info
      supported_groups
        supported group          x25519
        supported group          secp256r1
        supported group          secp384r1
        supported group          secp521r1
        supported group          ffdhe2048
        supported group          ffdhe3072
      ec_point_formats
        ec point format          uncompressed
      session_ticket
      application_layer_protocol_negotiation
      status_request
      delegated_credentials
      key_share
      supported_versions
```

복호화된 패킷은 클라가 공격자의 443 포트로 HTTPS 연결을 시도하며 전송한 ClientHello 메시지는 TLS 핸드셰이크 과정에서 전송하는 메시지로 클라와 서버가 암호화 통신을 시작하기 위한 메시지임

TCP 연결정보

Source IP : 192.168.174.131:51685(피해자)

Destination IP : 192.168.174.130:443(공격자)

TLS 버전 : 1.2 (version 3.3)

Resume : 클라는 이전 서버와 TLS 통신을 했던 세션 ID 를 재전송하여 해당 세션을 재사용함

Cipher Suites : 클라가 지원하는 암호화 방식을 나열한 것임

Compression Method : 사용하지 않았으므로 NULL

Extensions

SNI : www.jbnu.ac.kr 도메인의 접근할 것을 지정하였음

Extended Master Secret : 세션 키 유도 시 보안을 강화하는 확장

Supported Groups : x25519, secp256r1, secp384r1, ffdhe2048 등 클라가 지원하는 키 교환 알고리즘

실습 후기

이번 실습을 통해 ARP 및 DNS 스푸핑 기반으로한 SSL 중간자 공격의 구체적인 수행을 할 수 있었다.

칼리를 이용해 클라와 서버 사이의 트래픽을 가로채고 가짜 인증서를 생성하여 SSL 통신을 복호화 해보는 과정은 상당히 흥미로웠다. 실습중 ssldump 복호화가 제대로 되지 않아 애를 먹고 여러가지를 찾아보고 원인을 분석해봤다. 이러한 결과로 SSL 프로토콜 구조와 패킷 복호화 과정에 대한 이해도가 깊어졌다.

이번 실습을 통하여 공격 방법을 익히는 것을 넘어 보안이 왜 중요하고 왜 HTTPS 와 인증서 검증이 필수적인지 실질적으로 느낄 수 있었다. 요즘같이 해킹이 난무한 시대에 보안은 정말로 필수적으로 알고 있어야 하는 것 같다. 보안은 뚫는 법을 알아야 막을 수 있다는 말이 있는 것처럼 다음엔 방어 하는 실습도 해보고 싶다.