CNS Homework 3

B10202012 劉仲楷

Fun Hands-on Projects

1 Welcome To Fuzzing

2 Needham-Schroeder Protocol

a) The diagram is shown in table 1. In the beginning, Alice and Bob have K_A, K_B shared with KDC, respectively. N_A, N_B are nonces chosen by Alice and Bob, respectively. Flag: `CNS{N33DH4M_5CHR03D3R_PR070C0L_15_4W350M3_8e7c1985126d2142b87cdbc8ccca86aa}`

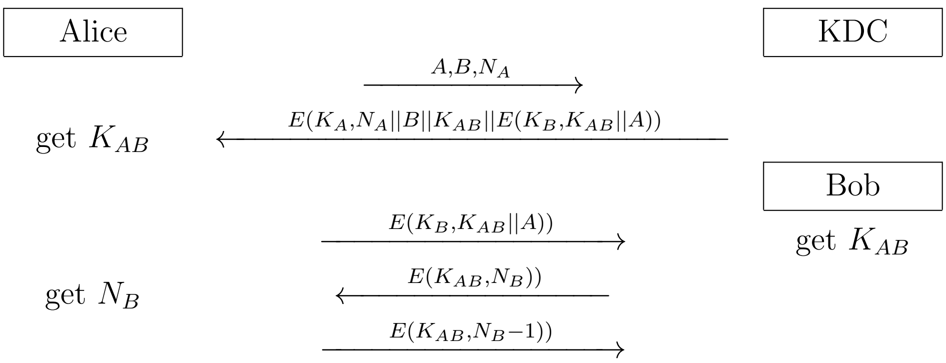


Table 1: Needham–Schroeder Protocol Scheme

b) Offline password cracking involves an attacker using previously obtained hashed passwords to guess the original passwords without interacting with the target system. The attacker uses techniques like brute force or dictionary attacks to match the guessed password hashes with the stolen hashes. Flag:
`CNS{D0N_7RY_7H15_47_H0M3_e340c50d4e72213257bab1a0deb4a2bd}`

c) In Bob's view, he cannot check the freshness of the key given by Alice because the KDC gives all the information to Alice but nothing to Bob. The key given by the KDC should include proof of freshness such as a timestamp. This could avoid a replay attack. Flag:
`CNS{R3PL4Y_4774CK_15_3V3RYWH3R3?!_d469fa0bd241f15122bc3aa38e8ed7ee}`

3 Accumulator

a) We need to construct a fake π, namely π' to create a fake membership. However, we cannot directly take the exponentiation by $1/s$. Hence, we need to know the $\phi(n)$ (or p, q) and exploit Euler's theorem

$$d^{\phi(n)} = 1 \mod n$$

when d and n is co-primes (which is always true).

$$\pi' = d^{1/s} = d^{s^{-1}} = d^{s^{-1} \mod \phi(n)} \mod n$$

Let $s^{-1} \mod \phi(n) = s'^{-1}$, then we can write $s'^{-1} = s^{-1} + k\phi$ for some $k \in \mathbb{Z}$. Thus the last equality can be verified by

$$(\pi')^s = \left(d^{s'^{-1}}\right)^s = \left(d^{s^{-1}+k\phi}\right)^s = \left(d^{s-1}\right)^s \left(d^{\phi}\right)^{ks} = d \cdot 1 = d \mod n$$

b) As the previous question, we need to construct $\pi' = (g^{a'}, b')$ with p, q satisfying

$$\begin{cases} a'u + b' \prod_{s \in S} s & = 1 \\ (g^{a'})^u \cdot d^{b'} & = g \end{cases}$$

for some $u \in S$ and a', $b' \in \mathbb{Z}$. We simply construct $(a', b') = (u^{-1}, 0)$. The verification result is

$$\begin{cases} a'u + b' \prod_{s \in S} s & = u^{-1}u + 0 \prod_{s \in S} s & = 1 + 0 & = 1 \quad \checkmark \\ (g^{a'})^u \cdot d^{b'} & = \left(g^{u^{-1}}\right)^u \cdot d^0 & = g^1 \cdot 1 & = g \quad \checkmark \end{cases}$$

However, u^{-1} cannot be known unless we know $\phi(n)$. Once we know p, q, we can calculate $\phi(n) = (p-1)(q-1)$. The rest is constructed like the previous question. Thus, to create a fake non-membership proof on $u \in S$, let

$$\pi' = \left(g^{u^{-1} \mod \phi(n)}, 0\right).$$

c) We can verify by checking

$$f(d, g_2) = f(\pi, g_2^c g_2^{-s}).$$

We can use the definition of the bilinear function to show the correctness.

$$f(d, g_2) = f(g_1^{\prod_{t \in S}(c-t)}, g_2) = f(g_1, g_2)^{\prod_{t \in S}(c-t)} = f(g_1^{\prod_{t \in S \setminus \{s\}}(c-t)}, g_2^{c-s}) = f(\pi, g_2^c g_2^{-s})$$

d) For $s \notin S$, to fake a membership proof by c, we can simply send

$$\pi' = d^{(c-s)^{-1}}$$

because

$$f(\pi', g_2^c g_2^{-s}) = f(g_1^{(c-s)^{-1} \prod_{t \in S}(c-t)}, g_2^{c-s}) = f(g_1^{\prod_{t \in S}(c-t)}, g_2)^{(c-s)^{-1}(c-s)} = f(d, g_2).$$

e) We can verify by checking

$$f(dg_1^{-b}, g_2) = f(g_1^{q(c)}, g_2^c g_2^{-u}).$$

The correctness can be shown similarly.

$$\begin{aligned} f(g_1^{q(c)}, g_2^c g_2^{-u}) &= f(g_1^{q(c)}, g_2^{c-u}) \quad = f(g_1, g_2)^{q(c)(c-u)} \\ &= f(g_1, g_2)^{p(c)-b} \quad = f(g_1^{p(c)-b}, g_2) \quad = f(g_1^{p(c)} g_1^{-b}, g_2) = f(dg_1^{-b}, g_2) \end{aligned}$$

f) Similarly, to fake non-membership proof for $u \in S$, we can send

$$\pi' = (dg_1^{-b} g_1^{(c-u)^{-1}}, b)$$

for any $b \neq 0$ (We can simply let it be 1.) because

$$\begin{aligned} f(dg_1^{-b} g_1^{(c-u)^{-1}}, g_2^c g_2^{-u}) &= f(g_1^{-b(c-u)^{-1} \prod_{t\in S}(c-t)}, g_2^{c-u}) \\ &= f(g_1^{-b \prod_{t\in S}(c-t)}, g_2)^{(c-u)^{-1}(c-u)} = f(dg_1^{-b}, g_2) \end{aligned}$$

g) Just implement the above protocol.

1. `CNS{bI1iN34r_4cCumU14tOr5_4rE_FUn}`
2. `CNS{311ipTIc_CUrV3S_Ar3_41SO_FuN}`
3.
4.

h) [1] In this situation, one can exploit the MOV attack. Suppose we want to solve the discrete log problem given the points P and rP, i.e. find r. We can solve discrete log problems in $\mathbb{F}_{p^k}^{\times}$ instead of in the original elliptic curve. In other words, we can compute $u = f(P, Q)$ and $u^r = f(P, Q)^r = f(rP, Q)$ for any point Q and solve the r. The group $\mathbb{F}_{p^k}^{\times}$ has the order $p^m - 1$. Thus, it is easier to solve the discrete log problem for a small embedding degree (No matter whether by brute force or by other attacks such as Pollard ρ algorithm).

i)
- (secp256k1) 19298681539552699237261830834781317975472927379845817397100860523586360249056
- (Curve25519) 1206167596222043702328864427173832373476186059896651267666991823047575708498

They are safe against those attacks since the embedding degree is large. Typically, only the embedding degree ≤ 6 can be solved efficiently. [1, 2]

4 DDoS

1. Figure 1 shows that the DDOS attack starts at about 24.8 s to 24.9 s. We then observe the traffic between the attack time interval and we can know that the victim is 192.168.232.95. With this information, we can search for the first packet sent to the victim within that time interval. The result is 24.945277 s, as shown in the figure.

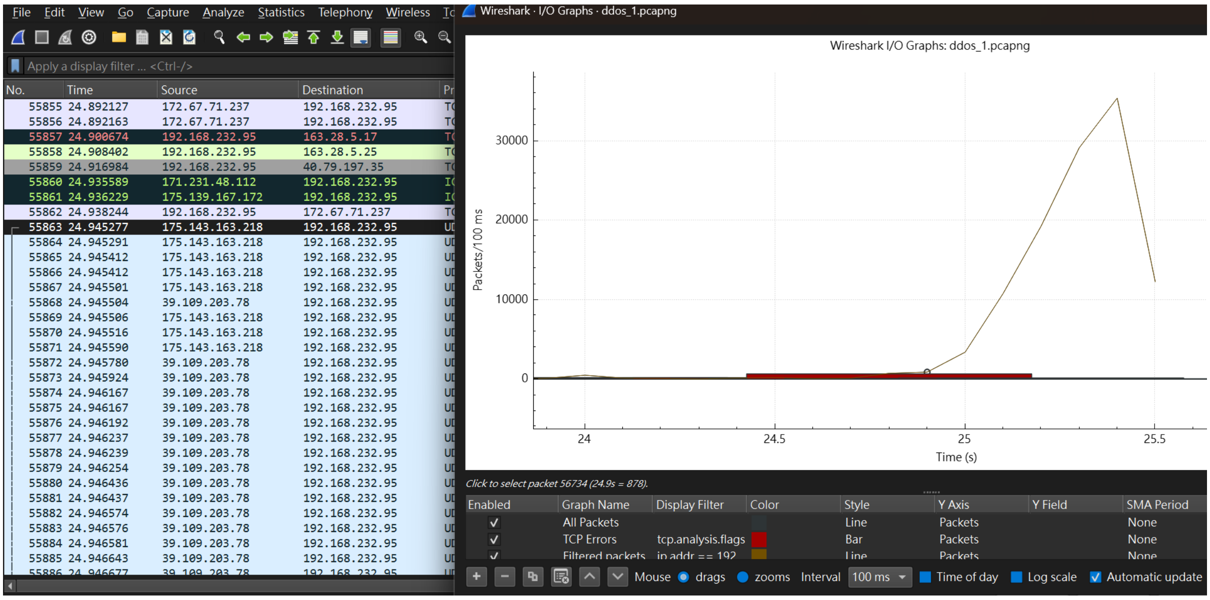
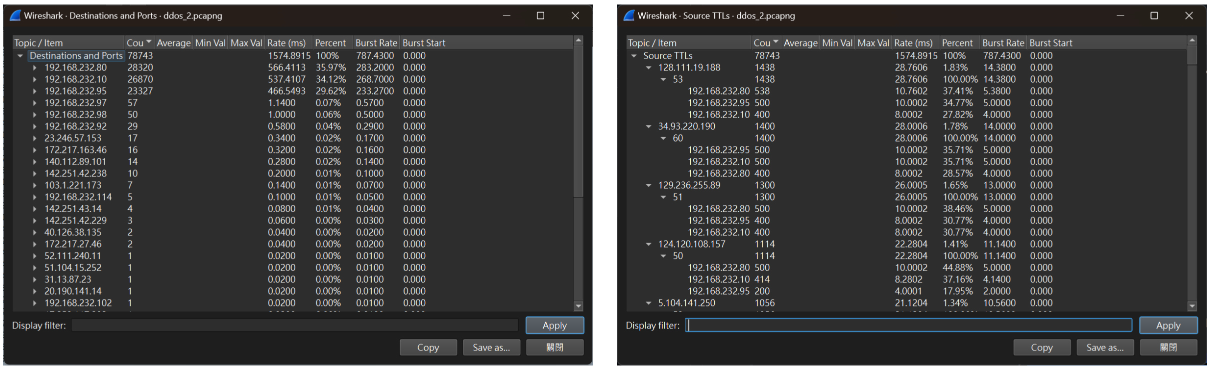


Figure 1: I/O Graphs of `ddos_1.pcapng`



(a) Destination IP

(b) Source IP

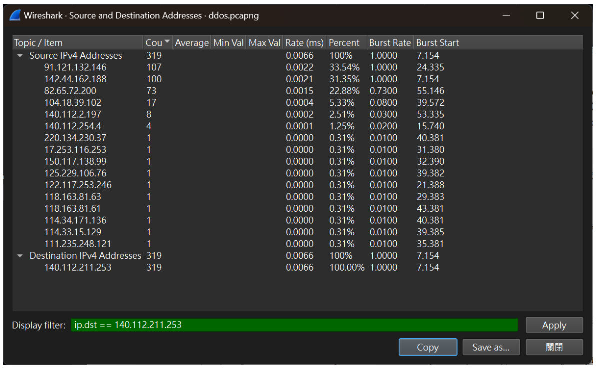
Figure 2: IPv4 Statistics of `ddos_2.pcapng`

2. With the information of the victim, we can filter with `ip.dst == 192.168.232.95`. Then we can find that most packets exploit the protocol of UDP and the length of 482 bytes.

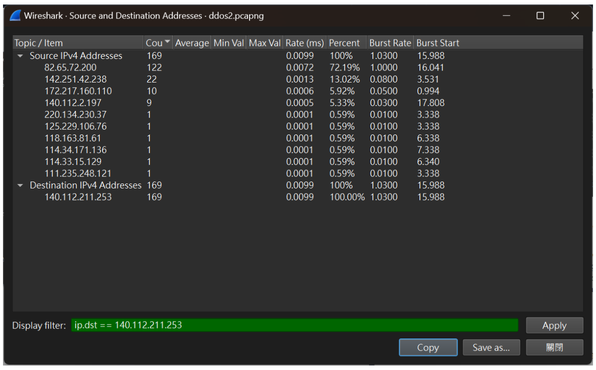
3. The victim of this DDoS attack is likely a server. The attack's intention seems to be exhausting the link bandwidth and paralyzing web services by targeting port 443 with a flood of UDP packets and causing the server to respond with ICMP "Destination unreachable" messages.

4. From fig. 2a, we found that the victims 192.168.232.80, 192.168.232.10, 192.168.232.95 received 28320, 26870, 23327 packets, respectively. From fig. 2b, we can see that 128.111.19.188, 34.93.220.190 and 129.236.255.89 are the three major amplifiers that send the most packets to the victim.

5. - **nmap:** We can use the `Source and Destination Addresses` in `IPv4 Statistics` and filter with `ip.dst == 192.168.232.95 && udp`. Then, we can get a list of IP addresses. Scan through the IP addresses with `nmap -sU {IP} -p`



(a) The First File



(b) The Second File

Figure 3: The Experiment on Amplification Factor

`123`. We ignored the result filtered or closed ntp. Besides the three given, I found another opening address 124.195.180.2.

- **ntpdc:** Next, we used `ntpdc -n -c monlist {IP}` and captured the traffic with Wireshark using the filter `utp`. The captured file is in `ddos.pcapng` and `ddos2.pcapng`. The third IP would not always work and the fourth was totally fail. Thus, I put the success result of the third address independently in the second file.
- **Analysis:** Finally, we analyzed the results as above with the filter `ip.dst == 140.112.211.253`. In fig. 3, we can see that 100[1] 482-byte packets were received from each of the IP addresses. On the other hand, the forwarding packet is 234 bytes. Thus, the amplification factor is

$$\frac{100 \times 482}{234} \approx 206.$$

6. [3] The operator of the amplifier (server) could disable the monlist functionality. The network administrator of the victim's network could use techniques such as BGP blackholing and IP filtering to protect the users from this attack.

5 Private Information Retrieval

6 Randomness Casino

1. `CNS{f1N@L_con7RIbU7iON_47T@cK}`
   Since all the number x_i is shown before "guessing" the number, we can "guess" the number y by

$$y = 800 - 1 - \left(\sum_i x_i \mod 800\right)$$

[1] 107 and 122 packets can be seen in the fig. 3 because I sent the request multiple times. However, for the successful session, only 100 packets were received.

2. `CNS{MT19937PredictorAlsoKnowsThePast!?}`
   Use the package Extend MT19937 Predictor [4], we can predict MT19937 after giving 32×624 bits generated numbers. Besides, backtracking is feasible, so we can easily predict the VIP and the number after us. The rest is as the previous expect the -1 (799) need to change as the number of the player.

3.

References

[1] Lawrence C. Washington. *Elliptic Curves: Number Theory and Cryptography*. 2nd ed., pp. 59–64.

[2] Daniel J. Bernstein and Tanja Lange. *SafeCurves: Transfers*. https://safecurves.cr.yp.to/transfer.html. Version 2013.10.13. 2013.

[3] Cloudflare. *What is an NTP amplification DDoS attack?* Accessed: 2024-05-31. 2024. URL: https://www.cloudflare.com/zh-tw/learning/ddos/ntp-amplification-ddos-attack/.

[4] NonupleBroken. *ExtendMT19937Predictor*. https://github.com/NonupleBroken/ExtendMT19937Predictor. 2024.