CyberSecurity – HW#2

Sahand Khoshdel - 810196607

Analytical Assignment #2

Table of Contents

Q1) Diffie-Hellman Key Exchange

a)

$$\left.\begin{array}{r} a = 7,\ \ q = 11 \\ Y_A = 3 \\ Y_A = \alpha^{X_A}\ mod\ q \end{array}\right] \Rightarrow \left[\begin{array}{l} 7^1\ mod\ 11 = 7\ \neq 3 \\ 7^2\ mod\ 11 = 5\ \neq 3 \\ \Rightarrow 7^4\ mod\ 11 = 5^2\ mod\ 11 = 3\ \Rightarrow\ \boxed{X_A = 4} \end{array}\right.$$

No, solving this problem in general involves calculating discrete logarithm that's a computationally hard problem

b)

$$K_{AB} = {Y_B}^{X_A}\ mod\ q = 5^4\ mod\ 11 = 9$$

c)

$$\left.\begin{array}{r} a = 3,\ \ q = 11 \\ Y_A = 3 \\ Y_A = \alpha^{X_A}\ mod\ q \end{array}\right] \left[\begin{array}{l} 3^1\ mod\ 11 = 3 = \ \boxed{X_A = 1}\ \ 3 \Rightarrow \\ 3^3\ mod\ 11 = 5\ \neq 3 \\ \Rightarrow 3^6\ mod\ 11 = 5^2\ mod\ 11 = 3\ \Rightarrow\ \boxed{X_A = 6} \end{array}\right.$$

As we observe, X_A (Alice's private key) can't be unique as it generates same Y_A in a 5-integer period cycle.

This is because α isn't a primitive root for q, as it violates the second condition.

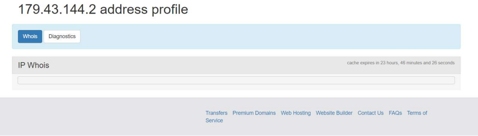
Two conditions should apply to let an integer such as α, be a primitive root of q:

1. Relative Primitivity to q ($\gcd(\alpha, q) = 1$)
2. Powers of α, from 1 to $q - 1$, should produce a complete residual class mod q

Q2) Attack Investigation

a) It seems like the attacker is trying to access different accounts by trying a password for different users. This is concluded from the multiple failures for authentication and different user names we see in the picture.

- **Unfortunately, "Who-is" IP finding server, didn't really appreciate me after finding out "Who I am", or maybe "Where I'm from" and preferred not to introduce others to me as well.**



179.43.144.2 address profile

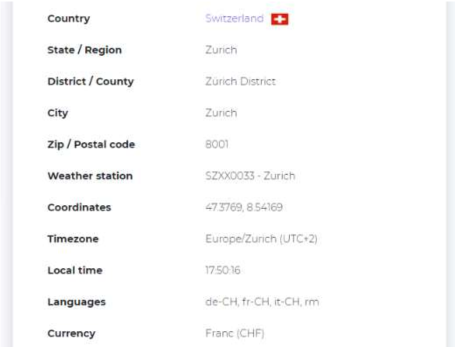
IP Whois

cache expires in 23 hours, 46 minutes and 26 seconds

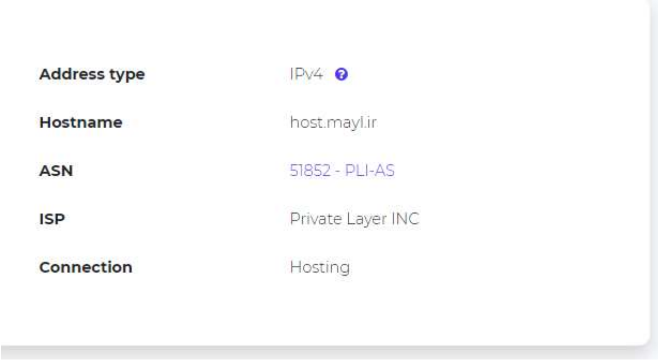
Whois
Diagnostics
Transfers Premium Domains Web Hosting Website Builder Contact Us FAQs Terms of Service

- **So, I used the service from "db-ip.com" instead:**

As we can see the attacker is from Zurich. Switzerland.



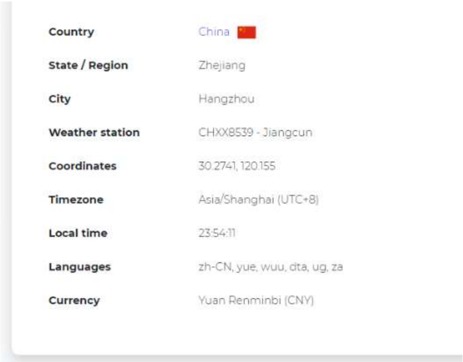
Country	Switzerland 🇨🇭
State / Region	Zurich
District / County	Zürich District
City	Zurich
Zip / Postal code	8001
Weather station	SZXX0033 - Zurich
Coordinates	47.3769, 8.54169
Timezone	Europe/Zurich (UTC+2)
Local time	17:50:16
Languages	de-CH, fr-CH, it-CH, rm
Currency	Franc (CHF)



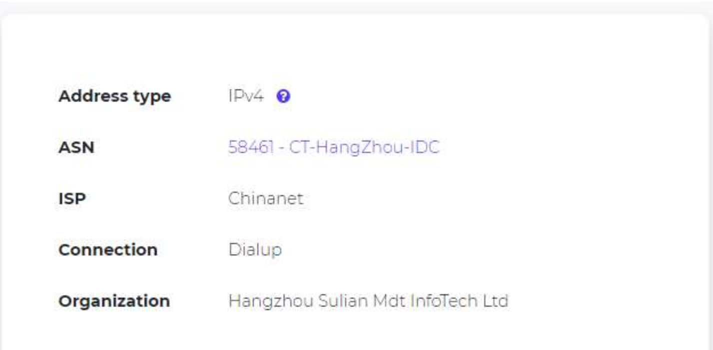
Address type	IPv4
Hostname	host.mayl.ir
ASN	51852 - PLI-AS
ISP	Private Layer INC
Connection	Hosting

b) By setting a limitation for the amount of login's that can be done in 24 hours for instance.

c) This is like a brute-force attack where the attacker is trying different passwords for the root, each time on a different port. The Mechanism is a 5-time limit for entering passwords and we can see that the attacker's IP is blocked after this limit. The figure below shows that the attack is from China.



Country	China 🇨🇳
State / Region	Zhejiang
City	Hangzhou
Weather station	CHXX8539 - Jiangcun
Coordinates	30.2741, 120.155
Timezone	Asia/Shanghai (UTC+8)
Local time	23:54:11
Languages	zh-CN, yue, wuu, dta, ug, za
Currency	Yuan Renminbi (CNY)



Address type	IPv4
ASN	58461 - CT-HangZhou-IDC
ISP	Chinanet
Connection	Dialup
Organization	Hangzhou Sulian Mdt InfoTech Ltd

Q3) Security Evaluation

a) **First Line:** Alice encrypts her recommended shared key with Bob (R) with the key she shares with the Server (K_a). Next, she attaches sender & destination's address to the beginning and sends it to the server.

Second Line: Server looks at the destination, picks the key that's shared with Bob (K_b) and uses it to encrypt Alice's recommended key (R), which has been decrypted using K_a and sends it to Alice as a reply.

Third Line: Alice who's been authenticated by the server using (K_a) now sends both the encrypted recommended key with (K_b) which the server has provided, and the encrypted message with her recommended key (R).

b) Yes, the destination address is known, as it's been sent to the server in plain text. Hacker Z can intercept the first packet and change the destination to his own address.

As the reply from the server to A, can't be decrypted by her, she can never now an interception is done. So, she will send her encrypted message and the encrypted recommended key (R) to B. (which Z can simply intercept again and decrypt it with (K_z), achieve the recommended session key (R), read the message and even start a session with A!

Attack Steps:

$A \rightarrow S\colon\ A, S, E(K_a, R)$ ✗ (Z intercepts): $Z \rightarrow S\colon\ A, Z, E(K_a, R)$ ✓ (Server authorizes Z as A!)

$S \rightarrow A\colon\ E(K_z, R)$ ✓

$A \rightarrow B\colon\ E(K_z, R), E(R, M)$ ✗ (Z intercepts) Attack is actually finished here! (Z knows M)

Optional:

$Z \rightarrow A\colon\ Confirmation\ and\ further\ communicaiton$ ☠

Q4) Security Evaluation + Improvement

a) Descriptive statements:

- **A believes that she's shared $K'_{AB}$with B**, because she has received a reply message from B in step #2 containing her encrypted nonce, with a previously shared key K_{AB}.

- **B believes that he's shared $K'_{AB}$with A**, because A has sent her nonce encrypted with this key meaning, authenticating herself by having access to K'_{AB}, via the previous key (K_{AB}).

- **A believes the K'_{AB} is a fresh key**, as B has sent it to her with her nonce, so the message that A has received can't be a reply message

- **B believes that K'_{AB} is a fresh key** because he, himself has made it!

b) The Attacker (C) can perform a reflection attack on A via the following steps:

$A \rightarrow B$: A, N_A × (C intercepts): $C \rightarrow A$: B, N_A ✓ (C performs a reflection attack spoofing B's address)

$A \rightarrow B$: $E(K_{AB}, [N_A, K'_{AB}])$ × (C intercepts, can't decrypt) $C \rightarrow A$: $E(K_{AB}, [N_A, K'_{AB}])$ ✓ (complete reflection)

$A \rightarrow B$: $E(K'_{AB}, N_A)$ × (C intercepts) From now on C has established a new connection with A, spoofing B, without even knowing their previous session key.

- We are assuming that A is unfortunately dumb enough to not even detect complete reflection in step 2
- Nonce is only acting as a time stamp so messages from the previous conversation can't be used as replay attack. (it doesn't help this attack scenario)

c) The problem with the reflection attack was answering a question with a question. By involving asymmetric information within messages, that can't be spoofed as well, reflection attacks can be prevented. For example, we can use hash functions and MAC's that authorize the creator of a content. Or we can easily compare the encrypted message received in step 2 and if its equal to our previous message the reflection attack can be detected because the attacker doesn't have access to the content in step 2.

Q5) Security Evaluation + Authentication

a) A sends her ID (ID_A) to B in the first message, B replies with a message R_1, A sends R_1, which is been encrypted with A's private key PR_A, Assuming she has shared her public key to b for authentication, B can decrypt A's message with the pair public key he has from A, a proof to the fact that no one except A, herself has sent B her ID. As a result, address spoofing is prevented and Asymmetric encryption, provides authentication service via private and public pairings.

b) A spoofing attack.
The bug is with the other side (B's) authenticity! A proves her authenticity by encrypting a message with a private key of herself but B doesn't need to prove his authenticity requesting A to do so, unless he wants to do future communication.

Therefore, a spoofer called C can intercept messages from A, and fake himself being B, requesting an encryption for any message he wants with A's private pair from her. After receiving this encrypted message, by another interception, C can send it as A's message to B, as he doesn't need to authenticate himself.

Q6) PGP (Pretty Good Privacy)

a) PGP is simply an encrypted hash with a session key, putting the first 16 bits of the hash function besides the encrypted message is actually done for evaluating the correctness of the encryption function. Authentication is done when someone uses our public key to decrypt and obtain the hash and comparing it with another hash constructed in the receiver side.

So having one's public key is enough to see the hash, and the point isn't in seeing the hash or the message, it's in the fact that fabricating and altering a message, via a spoofing attack or etc. needs exponentially high computational power to make it nearly impossible.

So, sending 16 plaintext bits of the hash won't cause any damage.

b) As we said sending plaintext doesn't cause security damage, yet has the weakness of not noticing whether an error occurred or not. This doesn't happen for the rest of the message because in case of an error in the encrypted bits the receiver can't decrypt it with the session key and will inform us.

So, we have to calculate the probability of an incorrect encryption algorithm being evaluated as a correct encryption. That's the probability of mapping from a wrong key to the right one in a 16-bit key space.

$$\Pr\{\text{error}\} = \frac{1}{2^{16}} \rightarrow \Pr\{correct\ evaluation\} = 1 - \frac{1}{2^{16}} = 99.998\ \%$$