Spring 2018 Cryptography and Network Security

Homework 2

Release Date: 4/20/2018

Due Date: 5/17/2018, 23:59

Instruction

- **Submission Guide:** Please submit all your codes and report to CEIBA. You need to put all of them in a folder named by your student id, compress it to hw2_{student_id}.zip. For example, hw2_r04922456.zip. The report must be in **PDF** format, and named report.pdf.
- You may encounter new concepts that haven't been taught in class, and thus you're encouraged to discuss with your classmates, search online, ask TAs, etc. However, you must write your own answer and code. Violation of this policy leads to serious consequence.
- You may need to write programs in the Capture The Flag (CTF) problems. Since you can use any programming language you like, we will use a pseudo extension code**.ext** (e.g., code.py, code.c) when referring to the file name in the problem descriptions.
- You are recommended to provide a brief usage of your code in **readme.txt** (e.g., how to compile, if needed, and execute). You may lose points if TAs can't run your code.
- In each of the Capture The Flag (CTF) problems, you need to find out a flag, which is in `BALSN{...}` format, to prove that you have succeeded in solving the problem.
- Besides the flag, you also need to submit the code you used and a short write-up in the report to get full points. The code should be named **code{problem_number}.ext**. For example, code3.py.
- In some CTF problems, you need to connect to a given service to get the flag. These services only allow connections from 140.112.0.0/16, 140.118.0.0/16 and 140.122.0.0/16.

Handwriting

1. Super Cookie (15%)

1. What is HTTP Strict Transport Security (HSTS)? How does it work?
2. It seems like HSTS can store one bit of information (whether to use HTTPS or not) in client's browser. Can you leverage this to create a `super cookie` to track user even when the user is browsing your website privately?
   Note 1: `Super Cookie` means you can store a piece of data on the client side even in

private web browsing.
Note 2: `Track User` means you are able to link user across multiple browsing sessions on the same website even in private web browsing.

3. How to prevent this kind of attack?

2. BGP (15%)

In this problem, we would like you to explain and analyze two attacks against the BGP routing protocol. In both attacks, the attacker, who has control over AS999, targets AS1000. Figure 1 shows the routing paths in the normal state after AS1000 has announced 10.10.220.0/22. Each circle represents an Autonomous System (AS). A solid line indicates a link over which two neighboring ASes can exchange control messages such as BGP update messages. A dashed line indicates an established AS path to 10.10.220.0/22.

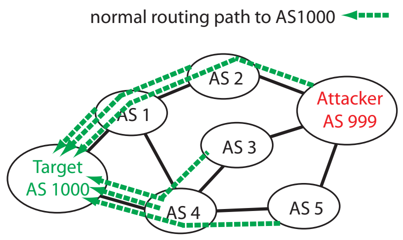

Figure 1: Normal scenario.

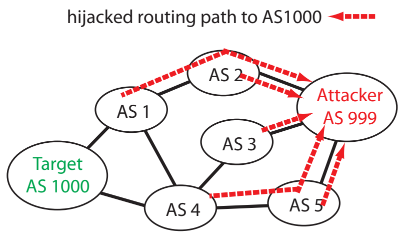

Figure 2: BGP hijacking.

1) (5 points) Describe the most likely scenario that could explain the result of Figure 2. Specifically, what did AS999 announce?

2) In the second type of attack illustrated in Figure 3, the attacker can silently redirect the hijacked traffic back to the victim along the path indicated by green lines. This attack exploits **AS Path Prepending**, where an AS inserts AS numbers at the beginning of an AS path to make this path less preferable for traffic engineering purpose, and **Loop prevention**, where AS x drops any BGP update with itself (i.e., AS x) in the AS-Path attribute to prevent routing loops.

   a) (4 points) Instead of announcing the ownership of an address block, AS999 announces a spurious BGP update: {IP prefix, {AS x, AS y, $\cdots$}}. Specify a BGP update message that could cause the result of Figure 3.

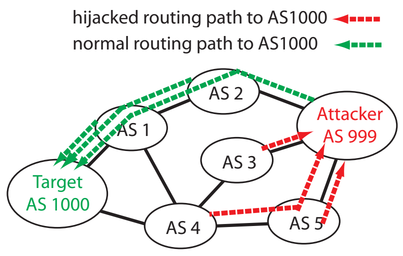
hijacked routing path to AS1000
normal routing path to AS1000
AS 2
AS 1
Attacker
AS 999
AS 3
Target
AS 1000
AS 4
AS 5

Figure 3: BGP man-in-the-middle.

b) (3 points) Briefly explain how the attacker misuses path prepending and loop prevention for malicious purpose.

c) (3 points) List one advantage and one disadvantage of this attack from the attacker's point of view.

3. PIN Authentication (15%)

Consider the following PIN Authentication Protocol between the client C and the server S:

$$S \rightarrow C : HMAC_{K1}(RS_1 \| PIN_1) \tag{1}$$
$$S \rightarrow C : HMAC_{K1}(RS_2 \| PIN_2) \tag{2}$$
$$C \rightarrow S : HMAC_{K1}(RC_1 \| PIN_1) \tag{3}$$
$$C \rightarrow S : HMAC_{K1}(RC_2 \| PIN_2) \tag{4}$$
$$C \rightarrow S : E_{K2}(RC_1) \tag{5}$$
$$S \rightarrow C : E_{K2}(RS_1) \tag{6}$$
$$C \rightarrow S : E_{K2}(RC_2) \tag{7}$$
$$S \rightarrow C : E_{K2}(RS_2) \tag{8}$$

$K1$ and $K2$ are two secret keys shared between C and S.

$HMAC_K(message)$ is Hash-based Message Authentication Code using K as the secret key.

$E_K(message)$ is AES-CBC Encryption Algorithm using K as the secret key.

RS_1 and RS_2 are two different 128 random bits generated by server S, and RC_1 and RC_2 are two different 128 random bits generated by client C.

PIN is a 8 digits password, PIN_1 is the first half of PIN and PIN_2 is the second half of PIN. For instance, if PIN = '12345678', then PIN_1 = '1234' and PIN_2 = '5678'.

In this protocol, client C and server S can authenticate the PIN even without sending the real PIN to each other. They need to compute the hashes ($HMAC$ in this protocol) of the combination of the random bits ($R\{S,C\}_1$ or $R\{S,C\}_2$) and the PIN. If the computed hashes are the same as the received ones, then they are authenticated.

For instance, after client C send $HMAC_{K1}(RC_1 \| PIN_1)$ and RC_1 to server S, server S can compute the hash $HMAC_{K1}(RC_1 \| PIN_1)$ and verify the value with the one received from client C. Server S will abort if the computed hash and the received one are different.

Can you break this protocol? Please explain the vulnerability in detail and explain how the attack works. Note that the protocol will abort immediately when verification fails.

Capture The Flag

4. Can't Beat CBC (15%)

(1) (4%) Hi, I am Hao. You learnt about the weaknesses of ECB Encryption Mode in class. Let me tell you a secret. CBC is unbreakable if you don't expose your key! You won't be able to find my $flag$ because I am using CBC Mode! Wait what? You don't believe? Try to find the $flag$ in `cbc1`. I bet you can't!

(The challenge is in `cbc1`, access the challenge by `nc 140.112.31.96 10124`)

(2) (5%) My name is Tu. I am Hao's friend. Hao is a lazy person, but he is right. CBC Mode is very nice and it is the most commonly used mode of operation. To prove it, I will give you my encrypted $flag$, but you will never know how to decrypt it without the key!

(The challenge is in `cbc2`, access the challenge by `nc 140.112.31.96 10125`)

(3) (6%) I am Hect. I love CBC and MAC. The former keep the confidentiality and the latter keep the integrity. If we use both together, we can keep everything safe and secure! If you can find Hao and Tu's $flag$, I am pretty sure you won't be able to find my $flag$!

(The challenge is in `cbc3`, access the challenge by `nc 140.112.31.96 10126`)

Note: In these challenges, you can submit your $flag$ to the server to check if you get the correct $flag$.

5. Man-in-the-middle Attack(15%)

`Man-in-the-middle attack` is a common attack technique where an attacker can secretly relay and possibly alter the messages between victims while making them beleive they are talking to each other. In the class, we talk about this attack when introducing authentication mechanisms. In this homework, you are asked to perform such an attack to a communication service `mitm/mitm.py`. In the service, the server uses a `Diffie-Hellman protocol` to exchange key information with another end-point. The key is later used to encrypt the flag. Can you recover the flag? You can access the service by `nc 140.112.31.96 10127`. Explain how you perform the attack in your write-up and provide your code.

Hint1: The value space of each password is small.
Hint2: Have you heard of the reflection attack?

6. Cloudburst (15%)

DDoS attack incidents are increasing recently. Many enterprises turn to Cloud-based Security Providers (CBSPs) to protect their websites. Basically, they update the DNS record of the origin website to a third-party cloud. The cloud not only proxies the user's request, but also has more network bandwidth to resist DDoS attack. However, this solution is based on hiding the origin IP address. Once the origin IP is exposed, an adversary can bypass CBSP easily.

The owner of `the-real-reason-to-not-use-sigkill.csie.org`, adopts CBSP solution as well. He uses a high-bandwidth cloud server `140.112.31.96` as the proxy. He also updated its DNS record. You can visit his website via `https://the-real-reason-to-not-use-sigkill.csie.org:10130/`.

Please find the origin IP of the website, and you will get the flag. You must describe how you find the origin IP. The only information is that the origin server is listening on https standard port (443), and the geographical location is in CSIE building, NTU.

Note: Please be careful when finding the origin IP address. In the NTU campus and the dormitory, suspicious network activity may result in being banned by NTUCC. (Is this a violation of privacy?)

7. One-time Wallet (15%)

One-time Bitcoin wallet generator is a service that can generate temporary Bitcoin wallets. It can randomly generate a new Bitcoin address and a password to protect the private key. You can access Bitcoin stored in the wallet only if you know the password. Most people use this temporary wallet only once, so they do not care if the old wallet password was leaked.

One day, one hundred of old wallet addresses with passwords are leaked. Given the generator code in `one-time-wallet.py`, prove that you can attack the generator by guessing the password of the next wallet and thus steal the Bitcoins. You can access the leaked data and the challenge by `nc 140.112.31.96 10128`. Explain how you perform the attack in your write-up and provide your code.

8. TLS Certificate (15%)

Do you want to join balsn? Please use balsn's root certificate `rootCA.crt` and root key `rootCA.key` to sign a certificate. The certificate information should be identical to the one of `https://www.csie.ntu.edu.tw/`. That is, only the issuer of the two certificates is different. The server `check.py` will check your certificate carefully.

Connect to the verification server via `nc 140.112.31.96 10129` and prove you are one of balsn.