**Spring 2019 Cryptography and Network Security**

# Homework 2

*Release Date: 4/29/2019*

*Due Date: 5/20/2019, 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 hw1_{student_id}.zip. For example, hw1_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.
- This homework set are worthy of 100 points.
- 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. SSL/TLS (21%)

(a) (12%) Please explain what features of SSL/TLS is used to defend the following attacks:

(i) (3%) Spoofing attacks: Pretend a connected client to fool a host into accepting bogus data.

(ii) (3%) Man-in-the-middle: Act as the client to the server and as the server to the client during the key exchange phase.

(iii) (3%) Replay attacks: Replay a single SSL/TLS packet of application data.

(iv) (3%) Replay attacks: Replay a whole SSL/TLS connection. Start from replaying a `"Client Hello"` message (the handshake phase).

(b) (3%) What is forward secrecy? Why forward secrecy is important?

(c) (3%) What is Rollback attack? How to prevent it?

(d) (3%) What is SSL Stripping attack? Explain how HTTP Strict Transport Security (HSTS) defends against SSL Stripping attack.

## 2. BGP (17%)

In this problem, we would like you to explain and analyze some attacks and defenses 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 10.10.12.0/22. 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 10.10.12.0/22.

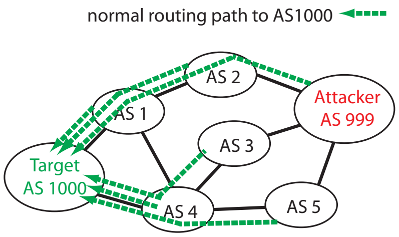

Figure 1: Normal scenario.

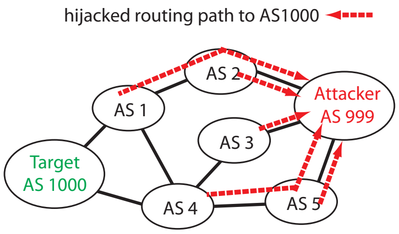

Figure 2: BGP hijacking.

1. (4 points) Please refer to figure 1. Assume AS999 is not an attacker in this subproblem, and the target AS999 somewhat notices that the link between AS1000 and AS4 are slow and congested. However, your dad, located in AS3, asks you to fix the network. Describe a solution for the AS1000 to reroute the traffic around congestion. That is, what BGP update messages should AS1000 announce?

2. (3 points) Describe the most likely scenario that could explain the result of Figure 2. Specifically, what did AS999 announce?

3. 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.

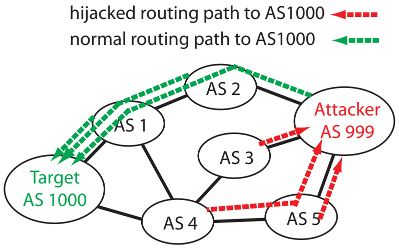
hijacked routing path to AS1000
normal routing path to AS1000
AS 2
AS 1
Attacker
AS 999
AS 3
Target
AS 1000
AS 4
AS 5

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

a) (3 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.

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

c) (4 points) List one advantage and one disadvantage of this attack from the attacker's point of view.

## 3. SYN Cookies (12%)

(a) (3%) Explain why SYN cookies can mitigate SYN flooding attacks.

(b) (3%) Explain why the cookie needs to contain a timestamp.

(c) (3%) Explain why the cookie needs to contain the client IP address.

(d) (3%) If an attacker could forge the Message Authentication Codes (MAC) used in a SYN cookie, what could he do?

# Capture The Flag

## 4. NS Protocol Revenge (20%)

> Hey Alice, I am Bob. I got some secret to tell you. Please come find me if you are free.

You are a secret agent and you are asked to inpersonate as Alice in order to get the secret. Can you accomplish your mission?

(1) (10%) Initial Authentication

(2) (10%) Subsequent Authentication

You can access this challenge by `nc 140.112.31.97 10158`. The challenge is also included in `hw2/ns`.

*Note: The keys in the server will update every minute.*

## 5. TLS (15%)

> Hi, I am Tom. I eavesdropped the communication between Alice and Bob. Unfortunately it seems they're using TLS to protect their message. Can you crack the protocol and find the flag?

You are provided with a packet capture file containing a TLS handshake and later encrypted packet. You can load it up with Wireshark. The challenge is included in `hw2/tls`.

Hint: What happen if the two prime factors $p$ and $q$ of an RSA modulus $n$ is too close to each other?

## 6. Eve's Revenge (15%)

> Hi, I am Eve. As an experienced attacker, I am always eager to paralyze those servers with vulnerabilities. Just now, I found a service provided by Alice, who just spoiled *Avengers: Endgame* to me last night. Therefore, I would like to exploit her server as a revenge. The server looks quite vulnerable to denial-of-service attack. Can you exhaust the server for me? I've prepared some benefits for you!
>
> Oh, I forgot to remind you that there are some proof-of-work puzzles to solve before accessing Alice's service. Good luck!

Alice's service is written in Python 2.7.15. You can access her service by `nc 140.112.31.97 10159`, and find the source code of her server in `dos.py`. Try to slow down the service by providing a set of malicious input. Please save your code as `code8.ext`, and your malicious inputs to the server as `input8.txt`.

Hint: How about reading the implemention details in `dictobject.c`?
Not Hint: #DontSpoilTheEndgame