Assignment #2

B05902120 / Yu-Ting, TSENG

May 19, 2019

SSL/TLS

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

(i) Spoofing attacks: Pretend a connected client to fool a host into accepting bogus data.
Confidentiality. After handshaking, server and client would have shared secret, and the attacker wouldn't be able to eavesdrop the content. These condition let the attacker hard to pass the MAC check.

(ii) Man-in-the-middle: Act as the client to the server and as the server to the client during the key exchange phase.
Integrity. Authentication of server and client ensures there is no MitM attack during key exchange process.

(iii) Replay attacks: Replay a single SSL/TLS packet of application data.
Integrity. The transmit data is protected by MAC secret, sequence number, and so on. This would avoid replay attack.

(iv) Replay attacks: Replay a whole SSL/TLS connection. Start from replaying a ”`Client Hello`” message (the handshake phase).
Server responds message in random (SR), which makes the attacker hard to create same shared secret as real client's session.

(b) What is forward secrecy? Why forward secrecy is important?
The key has forward secrecy if it generates one random secret key per session, which means that it does not exploit deterministic algorithm. This property indicates that every sessions work independent, and would not have influences on others, and there is no value lead to compromise of considerable sessions.

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

Rollback attack is a kind of downgrade attack. Since the initial handshake messages are not protected, the attacker works as MitM while handshaking, downgrading the message send by client server to SSL 2.0. Moreover, the messages aren't protected by the authentication during transfer; this cause severe security issue.

There are some action needed to prevent this kind of attack. Firstly, generate a warning message for this kind of error caused by an attacker. Secondly, adopt authentication calculation to avoid MitM.

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

SSL Stripping attack is a kind of downgrade attack. The goal is to downgrade HTTPS to HTTP. The user is made to believe that the connection is secure, and the messages sent are encrypted. However, MitM attack might happen, the attacker send the request of the user and turn the respond into http type, which is insecure.

What we can do to prevent this situation is to adopt HSTS policy – a strict policy under which browser won't open a page unless the site has HTTPS. This promise the user browse the website in secure way.

BGP

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.

(a) 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?

AS1000 advertises its path to AS1 only rather than both AS1 and AS4, so that others send the packet whose target is AS1000 would not pass through the link between AS1000 and AS4.

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

Due to the routing policy, one would choose the path with larger prefix, therefore, AS999 might advertise a longer prefix (eg. 10.10.12.0/23), so that others would tend to go there through AS999.

(c) 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.

(i) 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.

{10.10.12.0/23, AS999 → AS2 → AS1 → AS1000}

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

One would check the AS path, if it contains oneself, there must be a loop. Attacker might prepend a AS path and advertise with longer prefix. In this case, others tend to believe the path with longer prefix, they would move to router AS999 at first, then go to AS1000 through prepending path.

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

Advantage: It is hard to observe the prepending path. In other words, it is hard to set some specific rules to avoid attacks.

Disadvantage: The attacker's router would need to tolerate plenty of flow.

SYN Cookies

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

Sever does not use the its space to implement handshaking, instead, it calculates a cookie value based on client and sends it back.

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

Server doesn't record any information, so it is needed to check if the connection is out-of-date by current time and timestamp.

(c) Explain why the cookie needs to contain the client IP address.
Server doesn't record any information, it is needed to prevent the people sending SYN and ACK packets are different. If addresses of two are different, it might indicates one of it is fake.

(d) If an attacker could forge the Message Authentication Codes (MAC) used in a SYN cookie, what could he do?
Attacker might forge a fake MAC and return ACK before client. This might convince the server their request of connection, and furthermore making the server out of resources.

NS Protocol Revenge

Platform: Unix (Macbook)
Language: Python 3.6
Flag1: BALSN{M1dT3rM_i5_S0_h4rD_QAQ}
Flag2: BALSN{R3fl3Ct1oN_4774cK_S0_p0w3RfuL}

We implement "replay attack".

(i) $X \to B : A, N_X$
(ii) $B \to S : B, N_B, \{A, N_X, T_B\}_{K_{BS}}$
(iii) $S \to X : N_B, \{A, K_{AB}, T_B\}_{K_{BS}}, \{B, N_X, K_{AB}, T_B\}_{K_{AS}}$
(iv) $X \to B : N_{B N_X}, \{A, K_{AB}, T_B\}_{K_{BS}}$

Based on the above steps, we can get the flag of initial authentication.

(i) $X \to B : N'_X, \{A, K_{AB}, T_B\}_{K_{BS}}$
(ii) $B \to X : N'_B, \{N'_X\}_{K_{AB}}$
(iii) $*X \to B : N'_B, \{A, K_{AB}, T_B\}_{K_{BS}}$
(iv) $*B \to X : N''_B, \{N'_X\}_{K_{AB}}$
(v) $X \to B : \{N'_B\}_{K_{AB}}$

Following the steps, we might be able to implement reflection attack, and get the flag.

TLS

Reference: B05902013, B05902109

Flag: BALSN{CHOOSE_CIPHER_SUIT_CAREFULLY}

The first is to find modulus and public component in the packet, which are N and e of RSA respectively. In addition, due to the p and q are too close to each other, we find two factors from $\sqrt{N}$ by brute force and get d. At last, construct "pem" decrypt all application transition, and eventually get the flag.

Eve's Revenge

Platform: Unix (Macbook)

Language: Python 2.7

Reference: https://github.com/ictar/python-doc/blob/master/Python

Flag: BALSN{Py7h0n_4lg@r!thmic_Comp13Xity_Att4ck}

We firstly break sha256 via brutely force. Observing `dictobject.c`, we find that the order detecting the key in the dictionary is fixed. Therefore, we create a collision path with $2^1 5$ elements as inputs, then choose an input which costs great time due to collision. At last the server would output flag. The most difficult part in this problem is that the server execute time wouldn't be the same, the inputs we insert could not always get the flag.