CNS Homework 2

資工三 b05902058 陳竣宇

1. SSL/TLS

a

i

- Authentication
  - 確保client身分的正確性

ii

- Integrity
  - server與client在key exchange phase收到的資訊都需經過授權並未受篡改

iii

- Integrity
  - 不允許未經授權的通訊

iv

- Authentication
  - 若能從Client Hello開始進行replay attack，server就會誤認client的真實性

b

- 一個property，指的是短期的session key無法由長期使用的主金鑰long-term key獲得
- forward secrecy能夠確保即使密碼或金鑰在未來的時間點洩漏，先前的歷史訊息不會被攻擊者破密。即使攻擊者能夠取得某次transaction的資訊，這也只能讓他們access和這次交換相關的數據

c

- 在SSL/TLS的進行連線的過程中主要包括handshake和data transfer，Rollback attack主要就是發生在handshake的時候。類似Man-in-the-middle attack，攻擊者在client送出ClientHello時更改protocol version或cipher suite perferences成更老式、安全性更差的工作方式
- 要求client和server都要送出一個Finished message，其中包含所有先前handshake message的MAC，如此client和server就能確保他們協議的參數沒有在中間被攻擊者更改

d

- 一種使網路連線從secure HTTPS降級成insecure HTTP的技術，使自己暴露在竊聽和資料被篡改的危險中
- 強制client以HTTPS與server建立連線。當client通過HTTPS發出request後，server返回HSTS header，使得在未來的 `max-age` 時間內對於此domain的request都強制使用HTTPS

2. BGP

1

$10.10.12.0/22, AS4 \rightarrow AS1 \rightarrow AS1000$

2

$10.10.12.0/22, AS1000$

3

a

$10.10.12.0/23, AS2 \rightarrow AS1 \rightarrow AS1000$

b

path prepending原本的目的是想藉由在某個AS_path添加一串AS_path使得BGP避免去選擇這條而去選擇其他更短的路徑。攻擊者利用這個特性使得其可以將特定的route從原始最短路徑轉移成其所希望的路徑(把原本不經過attacker的route變成經過它)

c

- advantage
  - 大部分的router會使用這個hijacked route，且當attacker和victim相距不遠時latency並不高，因此不容易被偵測到
- disadvantage
  - 所有traffic都會流向攻擊者，可能造成攻擊者本身有congestion的情況

3. SYN Cookies

a

- 主要概念是希望server避免在SYN queue被填滿時自動丟棄連線
- 收到request後server傳回SYN+ACK但不保留state，當server之後收到client傳來的ACK之後才分配state。因為不正常的連線不會被server分配resource，因此可以減緩SYN flooding attack

b

可以防止偽造的sequence number。如果線路遇到帶out-of-line數值的舊sequence number，timestamp可以用來判斷它是不是舊封包

c

因為server會給每個請求連接的IP address分配一個cookie，若短時間內收到來自同個IP的多次重複request，就判斷為受到攻擊，之後會自動丟棄來自這個IP的封包

d

如果攻擊者可以偽造MAC的話，他們在不使用real IP的情況下就可以直接算出值並回傳給server使server分配資源建立連線。如此一來就不容易被server偵測出來從而導致資源耗盡

4. NS Protocol Revenge

4-1

```
BALSN{M1dT3rM_i5_S0_h4rD_QAQ}
```

- The paradox attack
  - $X \rightarrow B : A, N_x$

- $B \rightarrow S: B, \{A, N_x, T_b\}K_{bs}, N_b$
- $X \rightarrow B: \{A, N_x, T_b\}K_{bs}, \{N_b\}N_x$
- session key: N_x

4-2

`BALSN{R3fl3Ct1oN_4774cK_S0_p0w3RfuL}`

- The parallel session attack (Reflection attack)
  - $X \rightarrow B: N'_x, \{A, K_{ab}, T_b\}K_{bs}$
  - 開一個新連線，傳送原始連線B回傳的N'_b和$\{A, K_{ab}, T_b\}K_{bs}$
  - 取得新連線B回傳的$\{N'_b\}K_{ab}$
  - 在原連線送出$\{N'_b\}K_{ab}$

5. TLS

`BALSN{CHOOSE_CIPHER_SUIT_CAREFULLY}`

- 查看封包得到RSA public key n, e
- 使用Fermat's factorization method得到p, q
- 用 `Crypto.RSA` 構建private key
- 參考 https://blogs.technet.microsoft.com/nettracer/2010/10/01/how-to-decrypt-an-ssl-or-tls-session-by-using-wireshark/ (https://blogs.technet.microsoft.com/nettracer/2010/10/01/how-to-decrypt-an-ssl-or-tls-session-by-using-wireshark/) ，實際操作後取得flag

6. Eve's Revenge

`BALSN{Py7h0n_4lg@r!thmic_Comp13Xity_Att4ck}`

- reference (proof-of-work)
  - https://github.com/santisiri/proof-of-work/blob/master/pow.py (https://github.com/santisiri/proof-of-work/blob/master/pow.py)
  - 因為只有用數字去hash所以可能會找不到，再多跑幾次就可以
- 查看source code後可以知道python2是如何處理hash collision
- 使用公式產生要insert進去dict的值
- 接著再brute force找出最容易發生collision的number
- 10000個由python2處理hash collision機制產生的number，40000個brute force找出的number