Homework #7 Solution

Contact TAs: vegetable@csie.ntu.edu.tw

Network Administration

1. Cryptolocker (10%)

- The key space is actually no 1024 bits. It is generated by user input of length 8. Also, the encryption is separated into 4 stages, each stages having a key space of 8*2 = 16 bits.
- When decrypting, we can verify if a key is right by checking the AES padding. Therefore, we can brute-force each stage separately.
- We only have to brute-force $2^{16} * 4 = 262144$ time in total.

You can find the answer here.

2. 2AES (15%)

- Having a (plain, cipher) pair allows us to perform meet in the middle attack in 2 time encryption scheme.
- By using meet in the middle attack, we can decrease the key space into $2^{23} * 2$.

You can find the answer here.

3. Man in the Middle 2 (25%)

- As usual MitM attack, we open 2 connections and see what can we do between them.
- If we simply forward the message between the two parties, they'll exchange key successfully and print the encrypted flag. However, we don't know the key exchanged, so we can't decrypt it.
- But what if we modify something in the key exchange phrase? Say we simply forward the message in the first 9 rounds. But in the 10th round, we:
  1. Guess a **g** between [1, 20].
  2. Perform a MitM attack on Diffie-Hellman key exchange like our lab did.
  3. We now have **k1** shared with the first party and **k2** shared with the second party.
  4. If we guess the **g** right, then the encrypted flag by both party **c1** and **c2** will have the following property: $c1^{k1} == c2^{k2}$

5. By checking the property above, we can brute-force the **g** of each round in 20 times. Total of $20 * 10 = 200$ times.

- After having all 10 of the **g**, simply perform a normal Diffie-Hellman key exchange with the service and we'll have the key to decrypt the flag.

You can find the answer here.

System Administration

1 Lab

1.1 概念解釋 (15%)

1. 請解釋 proxy server 與 reverse proxy，並簡單舉例它們的用途。

   - forward proxy gets the resource from the Internet and sends it back to the client, it can be used to bypass IP address blocking.
   - reverse proxy takes requests from the Internet and forwards them to servers in an internal network, can be used for load balancing or security.

2. 試比較 nginx 與 apache 兩種 web server 的優缺，並說明你認為何者更適用於 reverse proxy。

   - nginx: light weight, performance
   - apache: more features, proper organizational support
   - As for reverse proxy, any resonable consideration will be accept

1.2 Apache (20% - 35%)

1. reverse proxy 會使用到 mod_proxy 與相關 module，無論預設環境是否有做好這件事，請明確寫出以保證你能使用到這些 module。作為提示，可在 `/etc/httpd` 之下找到相關的 conf。

   - In `/etc/httpd/conf/httpd.conf`, add `Include conf.modules.d/00-proxy.conf` (anyway that can load the correct modules is fine)

2. 請描述新增/修改的部分，並解釋所使用到的 directive。

```
ProxyPass / http://[ip of web-app1]/
ProxyPassReverse / http://[ip of web-app1]/
```

   - ProxyPass: allows remote servers to be mapped into the space of the local server (reverse proxy).
   - ProxyPassReverse: adjusts url to avoid bypassing the reverse proxy when the backend server redirects the url.

3. 將修改的內容成功套用到 apache 上。

```
sudo systemctl reload httpd
```

2 Directive (15%)

Note: Double quotes make no difference.

1. 將 `/var/www/html` 對應至網址中的 `http://www.example.com/`

```
DocumentRoot /var/www/html
```

2. 將 `/var/www/html/nasa` 這個 directory 設定為僅允許 `192.168.1.105` 存取，拒絕其他 ip (請避免使用舊版 apache 2.2 的 Allow, Deny)

```
<Directory /var/www/html/nasa>
    Require all denied
    Require ip 192.168.1.105
</Directory>
```

3. 為了安全網站啓用了 https，需將 http 的使用者導向 `https://www.example.com/`

```
Redirect / https://www.example.com/
```