Assignment 4
Solutions

Aftab Alam
Atul Anand Jha
Priyasha Chatterjee

1 Designing secure Software

(a) In order to keep the update process simple, MetaCortex's first idea is to not verify the image, but just establish a secure connection over HTTPS. Briefly explain why it is important to check the integrity of a download with a signature.

It is important to check the integrity of a download with a signature because HTTPS is not enough and it is not only to stay safe and avoid any of malware downloaded instead of benign file but it also make sure that the file you have downloaded is not altered and arrived unchanged because it is possible that transmission errors can occur anytime.

(b) The esp ota ops library includes tools which simplify the update process. Given the header file and an example, briefly explain how the update process works. In order to do so, give an example where you outline which parts of the flash is used for what during the update process. Furthermore state one drawback of this mechanism.

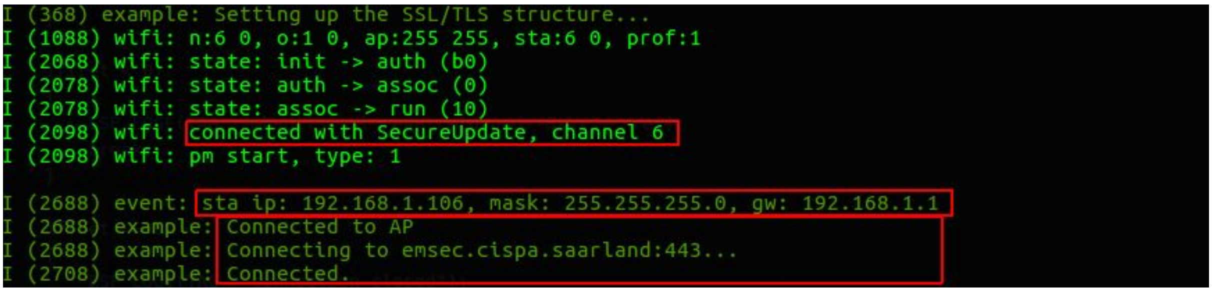
The OTA update is written to the specified partition. The Opaque handle "esp_ota_handle_t" for an application OTA update. esp_ota_begin() returns a handle which is then used for subsequent calls to esp_ota_write() and esp_ota_end(). When in the beginning the image size is not know the OTA_SIZE_UNKNOWN is passed which lead to erased the entire partition . esp_ota_begin function starts an OTA update writing to the specified partition else the specified partition is erased to the specified image size. On success, this function esp_ota_begin allocates memory that remains in use until esp_ota_end() is called with the returned handle. esp_ota_write is function that is called multiple times as data is received during the OTA operation. Data is written sequentially to the partition and It take 3 parameters handle that Handle obtained from esp_ota_begin, data that handles the Data buffer to write and parameter size that is the Size of data buffer in bytes. then esp_ota_end function finish OTA update and validate newly written app image. esp_ota_set_boot_partition is a function that Configure OTA data for a new boot partition. If this function returns ESP_OK, calling esp_restart() will boot the newly configured app partition. esp_ota_get_running_partition constant returns the partition information of currently running app. esp_ota_get_next_update_partition return the next OTA app partition which should be written with a new firmware.

One drawback is that the image is not verified and image is flashed directly and also it is carried out using plain HTTP no HTTPS. Any bad image can be delivered by the attacker and will be flashed happily.

(c) Implement your approach. Your program should…

Connect to the WiFi **SecureUpdate** with the password **EmsecExercise42018** and obtain an IP via DHCP

Done

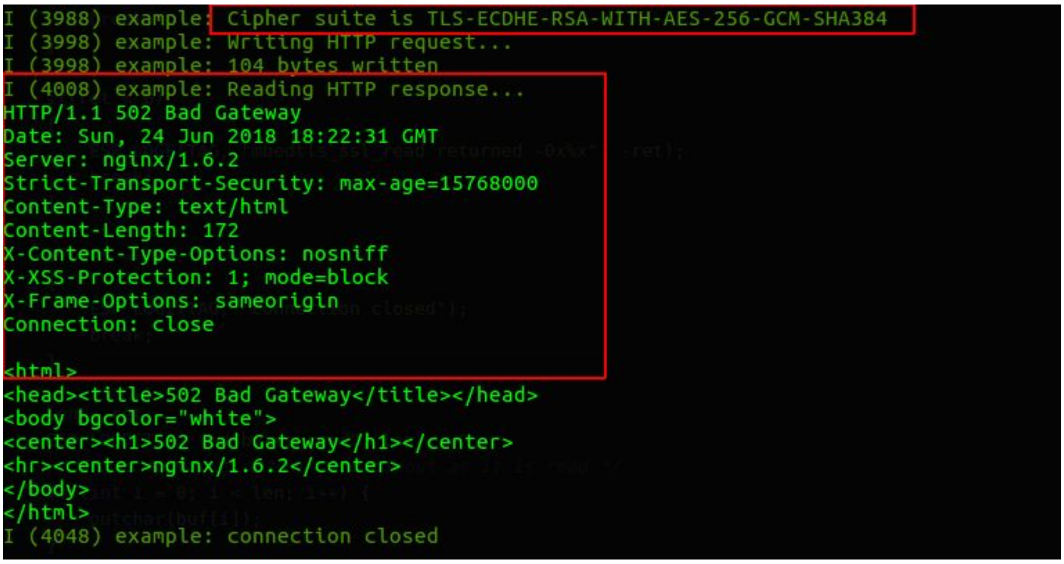


```
I (368) example: Setting up the SSL/TLS structure...
I (1088) wifi: n:6 0, o:1 0, ap:255 255, sta:6 0, prof:1
I (2068) wifi: state: init -> auth (b0)
I (2078) wifi: state: auth -> assoc (0)
I (2078) wifi: state: assoc -> run (10)
I (2098) wifi: connected with SecureUpdate, channel 6
I (2098) wifi: pm start, type: 1

I (2688) event: sta ip: 192.168.1.106, mask: 255.255.255.0, gw: 192.168.1.1
I (2688) example: Connected to AP
I (2688) example: Connecting to emsec.cispa.saarland:443...
I (2708) example: Connected.
```

connect to the Server given above through HTTPS

Done



```
I (3988) example: Cipher suite is TLS-ECDHE-RSA-WITH-AES-256-GCM-SHA384
I (3998) example: Writing HTTP request...
I (3998) example: 104 bytes written
I (4008) example: Reading HTTP response...
HTTP/1.1 502 Bad Gateway
Date: Sun, 24 Jun 2018 18:22:31 GMT
Server: nginx/1.6.2
Strict-Transport-Security: max-age=15768000
Content-Type: text/html
Content-Length: 172
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
X-Frame-Options: sameorigin
Connection: close

<html>
<head><title>502 Bad Gateway</title></head>
<body bgcolor="white">
<center><h1>502 Bad Gateway</h1></center>
<hr><center>nginx/1.6.2</center>
</body>
</html>
I (4048) example: connection closed
```

Download the flash file and its certificate, verify the image and boot from it if the signature is valid. It should fail downloading the image

(d) MetaCortex now created the software for a media control system. Unfortunately, the image size is fairly huge (about 3MB) in comparison to the 4MB flash size. They propose to create a small custom second stage bootloader (a few hundred KB) that can automatically connect to a network, download the image from the server in small chunks, feed these chunks to a hash function (like SHA256) and if the resulting hash is successfully checked against the certificate, the server is trusted and the image file seen as not compromised. Then the image file is downloaded again by the bootloader, overwriting the old firmware. What could possibly go wrong?

An attacker which has broken into the untrusted server can detect the first and second instance of image download by the client over network. Once the client can completed its verification of the file, using SHA256 hash, the attacker can replace the file with a malicious file which will be downloaded by the client in the second connection.
This situation arises because the checksum verification is not done on the actual image which is written on the flash.
Once the image file is downloaded and replaces the original image on the device, the device is compromised and in control of the attacker. After a reboot, everything will go wrong.

<u>2 Wireless problems</u>

(a) Briefly explain why bluetooth is not susceptible to narrowband jamming. Explain how this technique works.

In narrowband jamming, the jamming frequency is spread over a narrow spectrum, which is easy for Bluetooth to avoid, since it uses a Frequency Hopping Spread Spectrum,(FHSS) technology, which allows the the signal to “hop” from one channel to another, using a pseudorandom sequence. This sequence must be known to both transmitter and receiver. The seed to pseudorandom sequence generator is shared between the two devices during the ‘pairing’ process.

Bluetooth also makes use of Adaptive Frequency-Hopping (AFH) which means it will only use “good” frequencies, and avoid frequencies that interfere with it. So for Bluetooth to be susceptible to narrowband jamming, its pseudorandom sequence
would have to be known, which is unlikely. And then it would simply skip the interfering (jamming) frequency.

(b) Briefly explain why the Blueborne attack collection is especially critical for IoTdevices. Hint: Think about which Android version runs today on a Samsung Galaxy S4 device and what that means for software security.

BlueBorne allows attackers to leverage Bluetooth connections to penetrate and take complete control over targeted devices and can attack Bluetooth-connected devices over the air, without the device even being paired to the attacker's device. Even with this vulnerability it is possible of creating large botnets out of critical IoT devices like the Mirai Botnet. As attackers can force vulnerable devices to open Bluetooth connections and this vulnerability affect nearly 8.2 bi billion devices world wide, it is believed majority of the impacted devices will never be patched in 2018 is because these devices are old and won’t receive firmware updates at all or because updating them is too complicated and users won’t bother. Similar to android bug for samsung galaxy s4 hard to properly patch, because fixing it relies on wireless carriers all over the world getting their respective fingers out.

(c) Describe how you can get access to the encrypted communication of a Zigbee Light Link device. Consider how the gateway and the Zigbee device agree on a common key.

When any device joins the Zigbee network, it receives the encryption key from the Coordinator or the ‘gateway’. During classical ZigBee commissioning where a non-ZLL device is being joined to a ZLL network without a trust center, a pre-installed link key is used to secure the transfer of the network key when authenticating. This pre-installed link key is known to all the certified ZLL devices. When a new device requests to be added to the network, the trust center (TC) key is in the startup set, it could lead to a vulnerable initial exchange of keys, so it would be possible for an external attacker to disturb the network join using selective jamming and then wait for an insecure join to get access to the exchanged key material. As every ZLL device joining a network shall use the ZLL master key to derive the active network key, knowledge of the ZLL master key allows an attacker to intercept the key-exchange and acquire the current active network key. This would then allow the attacker to control all devices in the ZigBee network. Further, ZigBee is targeted for low-cost applications, and the nodes are not tamper-proof. If an intruder acquires a node from an operating network, he could obtain the actual key from the device.

(d) Briefly explain a way for the gateway and the Zigbee device to securely agree on a common key.

A common key is programmed into the devices in the factory. When connecting to a network, the device communicated with the gateway with a secure communication encrypted using the factory (default) key. The device receives the new key from the gateway over the secure channel and overwrites the existing key with the new key, which will be used henceforth for communication.