System Security 2024

Mock exam

Instructions

- The exam contains both open and multiple choice questions.
- The exam is individual. You are not allowed to communicate with other students, and you will be expelled from the exam if you do.
- Your final grade is the average of the assignment points and the exam points, as long as you scored at least 5.0/10 in the assignments and 5.5/10 in the exam. To pass the course, the final average must be at least 5.5, rounded to a 6.
- The exam lasts 2 hours.

READ CAREFULLY <u>ALL THE POINTS</u> OF EACH QUESTION <u>BEFORE</u> WRITING YOUR <u>ANSWERS</u>

Question 1

Consider the ESP32 board that you used during the assignments. This board includes several security features.

a. [3 points]. Your ESP32 chip supports a feature called “Secure Boot”. Describe briefly how it works and what kind of attacks it prevents.

Secure Boot guarantees code integrity, i.e., it allows for verification of the firmware loaded during the booting process. This mechanism prevents attacks that tamper with the firmware, as it guarantees that only firmware that is cryptographically signed by the vendor is executed.

b. [3 points]. Describe an example of how and where encryption keys relevant to the Secure Boot could be stored, briefly discussing the security of the described mechanism.

The vendor can flash their public key on a one-time writable storage (fuse). The vendor signs the firmware images with their private key, which is never stored on the device. During boot a cryptographic co-process verifies the integrity of the firmware by using the public key. The one-time writable storage guarantees that attackers cannot replace the public with a custom one.

c. [3 points]. Another feature that the ESP32 provides is the possibility to install MicroPython to implement the firmware in Python. What kind of implication would this have and what kind of security vulnerabilities would you avoid this way (e.g., in comparison to using C)? What kind of vulnerabilities are instead not mitigated by using Python?

Python is a memory-safe language. Thus, developing firmware in Python prevents memory corruption vulnerabilities such as buffer overflow, use-after-free, and memory leaks. Using Python does not prevent vulnerabilities such as command injection, authentication bypass, and missing security checks (e.g., TLS certificate validation).

d. [3 points]. Despite the aforementioned features, the ESP32, as most IoT devices, does not provide many security features that are present in modern Linux environments (e.g., Ubuntu on a modern laptop). What features that make successful exploitation harder are missing?

Control flow integrity, ASLR, seccomp...

Question 2

Consider a newly designed IoT device: a smart pillow. The device has Bluetooth connectivity and can connect to a mobile companion app to send data about the sleep quality. The mobile app also interacts with a cloud backend to store data and fetch updates.

a. [2 points]. The device runs a monolithic firmware image and you want to reverse engineer it. Discuss one of the challenges that this particular type of firmware imposes for static analysis and reversing, with respect to a traditional Linux program.

Monolithic firmware images do not have a standard binary format (e.g., ELF). Thus, identifying their data and code segments requires manually studying the physical memory layout of the hardware they run on. For instance, this is required to load an image in Ghidra.

Assume to be the vendor of such a smart device. You find a security vulnerability in your firmware and need to release a firmware update. Consider and evaluate the security of the following scenarios, pointing out their issues and strengths.

b. [2 points]. The mobile application downloads the updates to be installed from the cloud backend through an HTTP connection. The update is then sent to the device through an encrypted and authenticated Bluetooth connection.

No authenticity and integrity check on the firmware download. A remote attacker can perform a Man In The Middle attack and install malicious firmware on the device.

c. [2 points]. The mobile application first downloads a certificate through an HTTP connection to the backend, and then downloads, again through an HTTP connection, the updates to be installed. The certificate is then used to cryptographically verify the integrity and the authenticity of the updates. As in question b, the app sends the update to the device through an encrypted and authenticated Bluetooth connection.

The integrity and authenticity check can be bypassed by performing a MITM attack, passing an arbitrary certificate. An attacker can alter both the updates and the certificate.

d. [2 points]. Same as question c, but now the certificate is downloaded through an HTTPS connection, whereas the updates are still downloaded through an HTTP connection.

The certificate is downloaded through HTTPS so we can assume it can not be modified by an attacker. The certificate is then used to verify the updates in a secure way. However, the update verification process is performed on the mobile app only. This means that if the mobile app/smartphone is compromised, an attacker can install malicious firmware on the device.

e. [2 points]. Same as question d, but the application now downloads both the certificate and the updates through an HTTPS connection.

This scheme is as secure as the previous one, no improvements given. Minor difference: this scheme improves on confidentiality of the updates.

Question 3

Consider the following scenario. A smart device constantly sends an update regarding its status to a cloud backend. The device can be in three different states: “ON”, “OFF”, “UPDATING”. A status update is sent every 30 seconds by performing the following HTTP request:

```
POST /status HTTP/1.1
Accept: text/html
Accept-Encoding: gzip, deflate
Accept-Language: en-US
```

```
Cache-Control: max-age=0
Connection: keep-alive
Host: www.syssec-exam.nl
User-Agent: Super Secure IoT Device v.0.3

state=[ON | OFF | UPDATING]
```

A security monitor inspects the requests sent to the cloud backend and uses an anomaly-based detection approach to identify potential threats. In practice, any change to the above HTTP request is identified as an anomaly - also including changes in the timing between requests.

a. [2 points]. Describe a covert channel that would allow an attacker that compromised both the device and the cloud backend to exfiltrate information from the IoT device without being detected by the security monitor. Assume that the attacker can only operate at the application layer (i.e., HTTP layer).

Attackers can encode information in the device state. For instance:
UPDATING -> start/stop transmission
ON -> 1
OFF -> 0

b. [4 points]. How can the security monitor be improved to detect this covert channel?

The security monitor can learn a profile of the normal behavior of the device (e.g., statistical profile: how often the device switches from ON to OFF in 10 minutes). Abnormal behaviors (e.g. ON->OFF->ON->OFF) are flagged as potential exfiltration.

c. [4 points]. How can an attacker bypass the previously described improved security monitor?

The attacker can perform a victim-aware adaptive covert channel. They first learn the same behavioral profile and then produce very marginal differences, e.g., encoding one bit (ON/OFF) in one request every hour.

Question 4

Consider the following scenario. You want to develop a web-based platform for games that run in a browser. Games are composed of a module implemented in JavaScript and of another module on the server side that saves the game state and allows interaction with other players. The platform allows developers to deploy their games (both modules), exchange ideas and patches with other developers, and also let other users play their games instantly.

a. [2 points]. Discuss, on a high-level, the potential threats and isolation issues in this platform.

Code deployed by the developers has to be isolated to avoid (1) that they can modify/access code from other developers and (2) that they can take control of the platform/server.

b. [2 points]. Assume that the user base is composed of 5-10 developers. Discuss a mechanism to provide isolation among developers.

The app of each developer is deployed on an isolated virtual machine on top of the Xen hypervisor.

c. [3 points]. Your portal becomes quite popular and you suddenly have 200+ developers. Do you need any adjustments to the previously described isolation mechanism? Explain.

One VM per developer becomes too heavy. Basic solution: one docker container per developer. Better solution: one docker container per developer + containers are allocated/deployed on top of a fixed pool (e.g., 10 - depending on available resources) of VMs.

d. [3 points]. Discuss your threat model (for the scenario with 200+ developers). What are the assumptions your isolation mechanism relies on? What are the potential risks?

We assume that the kernel is not compromised. If the developers exploit a vulnerability in the underlying kernel, then they can escape the docker container.

Question 5

Consider the following snippet of code extracted from a custom TCP server:

```
 1    int main(int argc, char** argv){
 2          // more stuff
 3          while (1) {
 4                child_fd = accept(fd, &client_addr, &client_len);
 5                pid = fork();
 6                if (pid == 0) {
 7                      process(child_fd, &client_addr);
 8                      return 0;
 9          }
10          // …
11    }
12    void process(int fd, struct sockaddr_in *client_addr){
13          // init local vars
14          log_file = fopen(“mylog.txt”, “a”);
15          fprintf(log_file, “Processing request”);
16          dosomething(fd);
17          write(fd, “something done”, 14);
18          system(“sleep 1”);
19          // more stuff
20    }
21    void dosomething(int fd){
22          // init local vars
23          readLine(fd, &buf); // writes the read line in buff
24          // more stuff
25          data_file = fopen(“mydata.txt”, “a”);
26          fprintf(data_file, buf);
27    }
```

a. [4 points]. We want to insert a Seccomp filter to restrict the privileges of this program and mitigate damage in the case a remote attacker compromises the application. Where do you insert the Seccomp rules/filter (refer to line numbers)? Why?

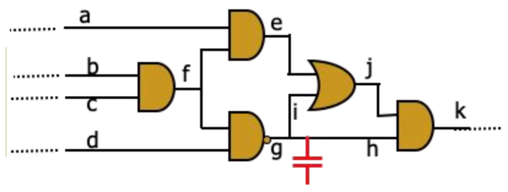
We insert the filter at line 22, before the first byte of untrusted input is read and processed.

b. [4 points]. Can you modify the program, keeping the same functionality, in a way that you can better restrict the required privileges (i.e., syscalls)? What would you change and why?

We move the file open at line 25 to line 22 (before installing our filter) so that we do not need to whitelist the open/openat syscall. Also, we replace `system(“sleep 1”)` with `sleep(1)` so that we do not whitelist extremely powerful/dangerous syscalls such as execve.

Question 6

The Hardware Trojan in figure (A) is:



a☐ Always active

b☐ Reduces propagation delay on d

c☐ Increases delay on e

d☐ Changes the implemented logical function

Question 7

In a side-channel attack based on power analysis of a cryptographic circuit, the attack exploits:

a☐ The circuit's static power consumption

b☐ The circuit's dynamic power consumption

c☐ The timing information derived from power consumption

d☐ The statistical correlation between delay and cryptographic operations

Question 8

Which statement is true regarding a Physical Unclonable Function (PUF)?

a☐ Stores a public key internally

b☐ Increases the reliability of a digital system

c☐ Uniquely identifies each device.

d☐ Are equivalent to FIR filters

Question 9

Which of the following is a detection methodology used to identify hardware Trojans?

a☐ Software vulnerability scanning

b☐ File integrity checking

c☐ Side-channel analysis .

d☐ Network traffic analysis

Question 10

Which of the following is a common method used by hardware Trojans to leak secret information?

a☐ Increasing power consumption

b☐ Modifying the clock frequency

c☐ Altering the data bus values.

d☐ Transmitting information via covert channels

Question 11

What is a primary advantage of SPA over CPA in side-channel attacks?

a☐ SPA can analyze devices with any level of signal noise

b☐ SPA requires fewer data samples to conduct an analysis

c☐ SPA is unaffected by countermeasures like randomization.

d☐ SPA can be conducted without knowledge of the device’s operation