Seminar on Reference Monitor

Hendon Tutor: Dr Xiaochun Cheng

M00783479

1. Microprocessors on smart cards used to have their entire card operating system in ROM. Currently, there are moves towards microprocessors where part of the operating system can be downloaded into EEPROM. What are the advantages and disadvantages of keeping the operating system in ROM? What are the security implications of moving parts of the operating system into EEPROM

Answer.

The operating system of microprocessors in ROM cannot be modified and this feature ensures the integrity of the system. A disadvantage if this feature is that the microprocessor is tied to its original executable code as no changes can be made, including updating the system.

However, moving parts of the operating system into EEPROM provides a possibility for system modification which may be exploited by an attacker.

2. Can you have security without security kernels? Discuss the advantages and disadvantages of having a security kernel as the TCB.

Answer:

NO, there may not be security without security kernels. This is because Security kernels are an essential mechanism for protection over code or data or any other type of insecure machine state. In all modern Operating Systems, there exist different and separated security levels inside the kernels, that deal with intrusion or the execution of malicious code.

A disadvantage of having a security kernel as TCB is that security is centralized, where compromise to the kernel means compromise to the entire system.

3. Look for examples that show how the following three principles are applied in building secure systems: separation of duties, abstract data types, and atomic operations. (An atomic operation has to be executed in its entirety to preserve security. If it is interrupted, the system may end up in an insecure state.)

4. So-called parasitic viruses infect executable programs. How can the ability to distinguish between programs and data help to construct a defense against such viruses?

Parasitic viruses spread by attaching themselves to executable programs. When a program infected with a parasitic virus is opened, the virus code runs. To hide, the virus passes control back to the original program. And the computer sees the virus as part of the program trying to run and gives it the same rights. These rights allow the virus to copy and install itself in memory, or make changes to the computer. Therefore, the ability to distinguish between programs and data provides the protection of programs against modification by a parasitic virus

5. Some buffer overrun attacks put the code they want to be executed on the call stack. How can the ability to distinguish between programs and data help to construct a defense against this particular type of buffer overrun attacks?

Answer:

Buffer overrun attacks work by overwriting the memory of an application by intentionally flooding the buffer with data beyond its limit. This changes the execution path of the program, triggering a response that damages files or makes the system vulnerable.

To defend against this type of buffer overrun attack, the security mechanism would have 2 states, switching between Data mode and Program mode. Where No code would be allowed to be executed when the assembly directions were manipulating data, and secondly, no such code would execute when on Data mode with specific instructions on what can be handled by the CPU at that point.

6. Anti-virus software scans files for attack signatures. How could a virus intercept the read requests to memory and hide its existence?

A virus can hide its existence by intercepting the read request of the infected file, handling the request itself, and returning an uninfected version of the file to the antivirus software. The interception can occur by code injection of the actual operating system files that would handle the read request. Thus, an antivirus software attempting to detect the virus will either not be permitted to read the infected file, or, the read request will be served with the uninfected version of the same file.

7. Consider a system that writes event numbers to its audit log and uses a table to translate these numbers into messages. What is the potential advantage of using this level of indirection in log file entries? What are the potential dangers?