ISAE Embedded systems master

Evaluation – Exercises and questions – WITH CORRECTION

25 february 2020

Embedded systems and computer security

Advice to students and supervisors : course documents (either furnished by the school or hand-written during oral courses by the student himself) are allowed during the examination, a standard calculator too (for calculation purposes only) and blank paper sheets for draft.

All other documents, communication devices or media access (like Internet) are not allowed, unless direct explicit authorization from the session supervisor.

Please, write your answers on the document itself in the place reserved.

Student name and surname :

Part I (10 pts)

This first part consists of ten questions (1 pt per question) with multiple answers proposed among which you must select the appropriate one. Unless explicitly indicated, only one answer is the correct one.

Attention, the following notation system will be used :

Correct answer : 1 point added False/no answer : 0 point

Q1 Why is it annoying to initialize the private key of a hardware security module at the factory ?

- ☐ Donald gets mad at the idea that Jinping may have tampered the motherboard.
- ☐ It is not. In fact, due to latest attack techniques based on silicium doping alterations, one needs to have its own semiconductor factory anyway for decent security.
- ☐ It raises doubts on the fact that the key might have been recorded.
- ☐ It costs a lot of energy.
- ☐ It prevents later module reuse.

Q2 What is the advantage of knowing the source code with respect to security ?

- ☐ Developers cannot hide backdoors easily.
- ☐ Managers cannot enforce dumb source obfuscation rules for intellectual property reasons.
- ☐ Everyone knows where to find the maintenance password easily: in the code. And they proofread the C code at the same time. (BTW, security paranoiacs can even change it.)
- ☐ The final user can double check alleged properties himself (and even find new ones).
- ☐ You can recompile the program with different security or optimization options.

Q3 Which one is not a security requirement ?

- ☐ User authentication.
- ☐ Production cost evaluation.
- ☐ Network compartmentalization.
- ☐ Delayed precise location disclosure.
- ☐ Contract digital signature.

Q4 What is the best way to check user input?

- ☐ Sanitize it by removing jargon words.
- ☐ Remove special characters.
- ☐ Ensure that numeric variables are within a given interval.
- ☐ Parse the input completely.
- ☐ Only offer 6-buttons user interfaces like in the NASA designs.

Q5 What is one of the main problem point between CI/CD (continuous integration / continuous development) and classical security deployment procedures ?

- ☐ Only chaotic white panthers can manage the resulting chaos.
- ☐ Security updates occur all the time, nearly continuously.
- ☐ The security level changes all the time.
- ☐ Classical security validation procedures at delivery cannot be used effectively.
- ☐ The security kernel responsible for security updates needs to be developed first.

Q6 What CPU/hardware feature does *not* help with respect to security ?

- ☐ Memory protection.
- ☐ Branch prediction.
- ☐ NX (no-execute) access right.
- ☐ Memory barriers.

Q7 Which of these aspects of system hardware is associated to a *covert* channel (NB: different from an auxiliary channel)?

- ☐ The noise of CPU fan.
- ☐ The energy consumption.
- ☐ Branch prediction logic.
- ☐ A shared communication bus with shared storage.

Q8 Which physical random number generator sounds secure to you?

- ☐ A 5 years old child smashing the keyboard.
- ☐ Low level bits of audio recording samples.
- ☐ Low level bits of a thermal sensor measurements.
- ☐ Low level bits of a Geiger sensor monitoring artificial radioactivity.
- ☐ A whole class of students trying to answer to a negatively formulated questionnaire.

Q9 Which one is an authentication protocol ?

- ☐ Diffie-Hellman
- ☐ RSA
- ☐ SHA3
- ☐ TCP
- ☐ SMTP

Q10 How can you protect a component from EM interference (in the real world) ?

- ☐ By introducing thermal noise.
- ☐ By shielding with lots of lead (Pb).
- ☐ Using a 1Tesla protection field.
- ☐ With a Faraday cage.

Part II (10 pts)

This part is composed of five open questions (2 pts each). Please write down your answer *on this document* in the appropriate space.

(example answers)

Question 1

Give an example of a security objective (i.e. a requirement) corresponding to several types of embedded systems:

- a military plane flight control computer

1) Communication of navigation and targeting command orders should always be encrypted.

1.2) All communications should always be encrypted

1.3) All communications should always be encrypted with an algorithm and key length which security is at least the one of an AES with 512 bits key.

2) The C&C DoD Unit Z51 should be able to command any foreignly purchased F35 to self destruct at any time.

2.2) The above requirement should be erased from the specification as soon as implemented and then denied obstinately under all circumstances.

3) All firmware updates of the control computer should be only performed after cryptographic signatures verification conformant to … Draft NIST/FIPS 186-5 sect.7 (aka EdDSA) (e.g.)

4) No software ROM modification of the control computer can be realized in operation. (Whole system physical replacement should be the only method for software upgrade.)

Note 4 simplicity and wonder if it is readily applicable to healthcare (3rd bullet below) ?

- a bike for rent

1) The unscheduled removal of the battery component should trigger an immediate alarm to the management station and recording of location.

1.2) The unscheduled removal of the battery component should self-lock the battery.

1.3) The remote management station should be able to unlock a locked battery remotely.

2) Customers should authenticate with a smartphone in order to rent a bike.

3) The bike location should be known from the management station at all times

3.2) Customer privacy should not be compromised by the recording of bike location

~~*3.3) Contradictive high level requirements should be solved by underpaid subcontractors in the detailed specifications (or even cheaper, directly in the code)… Oops. No.*~~

- a pace maker (heart stimulator)

1) Only the surgeon or the pace maker manufacturer chief engineer are allowed to kill the patient.

2) Firmware updates of the pace maker should involve an electronic signature check including the verification of approbation by a (healthcare-domain) certification authority.

3) The pace maker should demonstrate permanent adequate resistance to any intentional EM-based perturbations of low to moderate power (< 1 W at 10 cm without obstacle).

4) The pace maker should not stop heart stimulation during a software update (even if security-oriented…).

Note: common safety+security concern

Question 2

Consider a live videoconferencing application allowing the passengers of cars *of a given manufacturer only* to communicate freely with each others (via a manufacturer controlled infrastructure) while traveling on roads.

Consider the manufacturer is based in Borduria, which country seems dominated by a semi totalitarian and belligerent government.

Propose 6 security objectives for this system (among all the imaginable ones).

2 of them corresponding to the needs of the passengers of the cars:

(examples)

- *The manufacturer should not be able to eavesdrop on the communication content.*
- *The manufacturer should not be allowed to know the list of participants of a given video conference.*
- *The manufacturer should not known the precise duration of a specific videoconference (NB: pretty difficult to address depending on service contract billing conditions).*

2 of them corresponding to the needs of the manufacturer of the cars:

(examples)

- *The videoconference service customers should be billed appropriately given their service contract.*
- *Competing manufacturers should not be able to use the videoconferencing infrastructure to mount a similar offer (without similar investment)*
- *The company should comply with investigation laws of all the countries where it expects to provide the service.*

And the 2 others to the needs of the government of Syldavia (Borduria historical rival) :

(examples)

- *Microphone and camera equipment setup inside Bordurian cars should be security-validated by an independent third party to demonstrate that they are not backdoored to allow spying on the car passengers.*
- *The manufacturer-based videoconferencing infrastructure should be able to operate without interaction with Bordurian-based infrastructure. (In practice, it may mean that the manufacturer may be obliged to deploy at least some of this infrastructure on Syldavia land.)*
- *Police investigation should be feasible for Syldavian law officers following regular sovereign law procedures.*

Question 3

Propose a comparison of CBC and ECB modes of operation for an encrypted communication channel (at least 2 advantages and 2 drawbacks of each option).

Check the document for modes of operation description.

ECB mode offers simpler and more efficient implementation opportunities: e.g. it is possible to encrypt several parts of the message in parallel on (multiple) hardware encryption cores.

ECB does not need any initialization vector negociation, which is an interesting advantage, especially for encryption of small messages (those which length is less than the blocksize).

However ECB mode does not prevent an attacker to build a correctly-decrypting (but probably incorrect) message from several intercepted encrypted messages. Hence, internal checks of integrity should be added to message for cleartext semantic verification.

CBC mode necessitates the integration of the negotiation of an initialization vector in the authentication/establishment phase.

CBC mode necessitates pipelining for parallel hardware usage effectiveness.

CBC mode ensures intra message encryption protection so an attacker capturing ciphertexts cannot forge a new valid ciphertext by mixing them.

CBC mode implies that decryption cannot start until all the message is received.

Question 4

Propose 3 basic programming rules in C with the objective of protecting primarily system *integrity,* against malicious faults of course.

Examples taken from CERT C Secure Coding Recommendations:

- *Do not use floating-point variables as loop counters*
- *Prevent or detect domain and range errors in math functions (like sqrt(-1.0) or pow(10., 1e6))*
- *Allocate sufficient memory for an objective*
- *Do not call system() (use excve(), which does not use a full shell interpreter)*
- *Sanitize the environment when invoking external programs*
- *Do not allow attackers to influence environment variables that control concurrency parameters (debate open with e.g. OpenMP())*
- *Do not form or use out-of-bounds pointers or array subscripts*

Question 5

Explain the advantages and drawbacks of enforcing a company acquisition policy where only ISO15408 (common criteria) certified IT systems can be bought.

Advantages (examples):

- *A uniform security and strong security level is achieved through the whole internal IT system.*
- *Customers get easy access to security guarantees for services or even products integrated by the company.*
- *Suppliers can be selected with respect to security based on a recognized standard evaluation scale for their products.*

Drawbacks (sic):

- *Price may be much less negotiable.*
- *Some commonly expected products may not be available.*
- *Some protection profiles may not be defined for needed systems and the overhead of defining them could be pretty high.*