ISAE Embedded systems master

Evaluation – Exercises and questions – WITH CORRECTION

30 march 2023

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 What is the moment in a software development project to consider intrusion testing?

- ☐ Before emergence of the project idea.
- ☐ After each project sprint.
- ☐ After the first payment.
- ☐ At any time during operational life.
- ☐ After your country governmental agency informs you that they have done one.

Q2 CRC codes are sometimes associated to transmitted data in order to allow detection of accidental alteration of binary data. Why cannot they be used for securing the integrity of the same data?

- ☐ Because they are too short and cannot be extended easily in size.
- ☐ Because their security has not been validated.
- ☐ Because they are linear codes and hence amenable to inversion.
- ☐ Because you would have to transmit the CRC separately from data to prevent the attacker from also tampering the checksum.

Q3 Which case is an example of a logical backdoor?

- ☐ The rear garden door.
- ☐ A malicious program sent by mail mimicking a small fun tic tac toe game
- ☐ A magical test keys sequence that lets an ATM expel one bank note.
- ☐ An SQL script destroying the company databases if a specific employee contract is terminated (UNLESS current_year > BIRTH_DATE+42).

Q4 Which attribute (of dependability) can be associated to security main attributes?

- ☐ Maintainability.
- ☐ Availability.
- ☐ Anonymity.
- ☐ Price.
- ☐ Security updates.

Q8 Here is a small snippet of C code, where variables have their natural meaning (`buf` is some buffer space, `src` some internal safe data and `val` some external not-trusted value).

```
if (val * 4 > sizeof(buf) ||
    val > UINT_MAX / 4)
        return -EINVAL;
memcpy(buf, src, val * 4);
```

The above piece of C code is :

- ☐ Incorrect.
- ☐ Inefficient because untrusted values checking should be done at the call site.
- ☐ Correct.
- ☐ Correct but written in the wrong order.

The overflow prevention check should be done first, before the actual space check. It is not harmful to do it in this order, but checkers will warn about that.

Q5 What is the main limitation of using a symmetric encryption algorithm in a digital signature scheme ?

- ☐ It is mandatory to encrypt the document in order to get a signature.
- ☐ Signatures are not useful in front of a third party as both parties can sign whatever version of the document they claim to be original.
- ☐ You cannot detach the signature from the document.
- ☐ It could be much slower than a public/private key signing system.

Q6 Why is it impossible to trust a single private key to provide adequate integrity protection?

- ☐ Because prime numbers factorization is gonna improve a lot with quantum computers.
- ☐ Because perfect security does not exist and you can always guess the key.
- ☐ Because you cannot be sure that the public key associated to the private key is actually associated to the good recipient (unless you trust the public key directory).
- ☐ Because you only have the public key to check the integrity verification and can only encrypt with it.

Q7 Which of these parts of system hardware is most vulnerable to blunt magnetic reads?

- ☐ The CPU fan.
- ☐ The motherboard .
- ☐ The hard disk.
- ☐ The FDDI channel.

Q9 Which one is a secure hash algorithm ?

- ☐ Diffie-Hellman
- ☐ Rijndael
- ☐ SHA-3
- ☐ UDP
- ☐ HTTP

Q10 How can you realistically protect a component from electromagnetic monitoring?

- ☐ By eliminating covert channels.
- ☐ By heavy shielding (of memory and logic) with 20mm strainless steel.
- ☐ With a Faraday cage.
- ☐ By a strong electromagnetic field.

Part II (10 pts)

This part is composed of four open questions (2.5 pts each). Please write down your answer *on this document* in the appropriate space.

Question 1

Consider an embedded web server inside the computer of an automotive. We assume this computer is not associated directly to any critical function of the vehicle (like speed or attitude control); but otherwise participates to any other function used by the vehicles users (navigation, entertainment, route planning, maintenance, networking, etc.) Wireless communication systems allow this computer to be connected to the Internet but with typical perturbations of a moving car (temporary loss of connectivity, limited bandwidth, etc.).

Propose 2 security objectives for this system (among all the imaginable ones), from the point of view of the car manufacturer.

1) Enable the secure distribution of content to the cars in operation (maybe only data content, but probably also software updates).

2) Ideally, allow the distribution of paying content (even on-demand?), in this case, protect content from cars passengers.

3) Prevent owners to change the operational software of the computer system.

4) Prevent any user to have access to the critical embedded systems of the car.

Identify two additional security objectives for this system (among all the imaginable ones), from the point of view of the car *owner*.

1) Allow owner or law enforcement to locate, safely immobilize and gain access to stolen cars.

2) Certify that battery usage has followed a certain use profile (and that battery is still worth a given fraction of its initial price).

3) Link car authentication to parking access control systems automatically (ie. open my automatic garage doors without compromising house security).

4) Open doors and switch off engine in case of accident (yep, that's also a safety objective).

5) Close doors and safely immobilizes vehicle in case of car robbery attempt (yep, that's interestingly different from the previous one).

Try to imagine a situation where the car owner and the car manufacturer would have contradictory security expectations.

The manufacturers may want maintenance operators to be able to access the car internals even in the absence of the owner, and hence have a privileged pass key.

The owner may not want maintenance operators to be able to open his/her car without obtaining his/her explicit authorization first, whatever the opionion of the manufacturer.

More generally, maintenance and/or software hacking will certainly give rise to conflictual situations.

Question 2

Propose 4 security requirements associated to the development environment (editor, compiler, testing framework, version management system, software distribution) of a software company.

Strong developer authentication, which imply some developer/user authentication infrastructure setup and management.

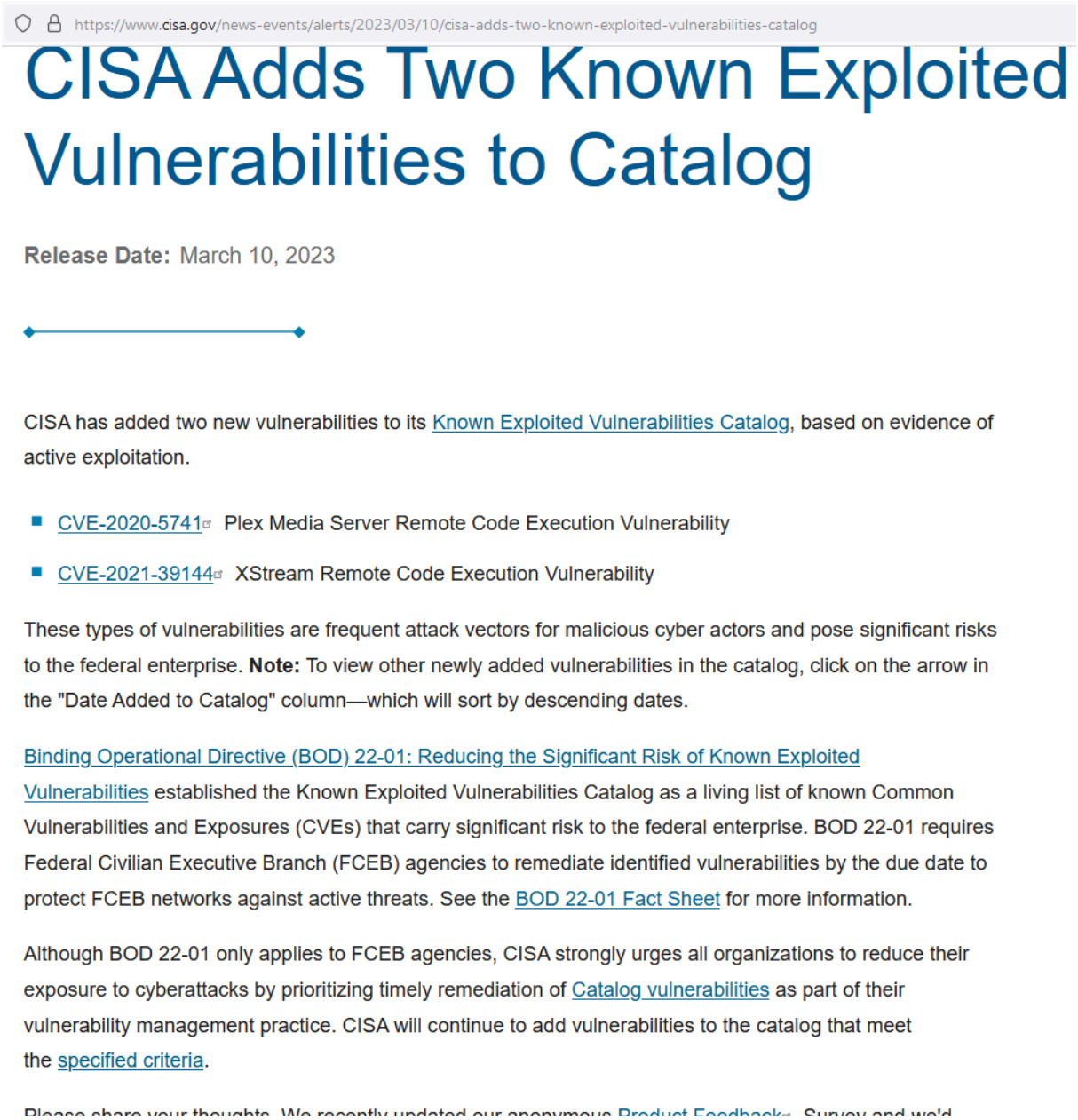
Mandatory software checkin signatures: each new software function or version should be signed by their developers..

Binary software distribution signatures (including reference of the builder, build system and its configuration).

Build system integrity protection: check that software is compiled on a trusted system (via regular files fingerprinting of that system for example).

Interdiction of real data usage for testing systems (personal data for example for legal reasons, or actual security secret keys for security reasons).

Question 3



https://www.cisa.gov/news-events/alerts/2023/03/10/cisa-adds-two-known-exploited-vulnerabilities-catalog

CISA Adds Two Known Exploited Vulnerabilities to Catalog

Release Date: March 10, 2023

CISA has added two new vulnerabilities to its Known Exploited Vulnerabilities Catalog, based on evidence of active exploitation.

- CVE-2020-5741 Plex Media Server Remote Code Execution Vulnerability
- CVE-2021-39144 XStream Remote Code Execution Vulnerability

These types of vulnerabilities are frequent attack vectors for malicious cyber actors and pose significant risks to the federal enterprise. **Note:** To view other newly added vulnerabilities in the catalog, click on the arrow in the "Date Added to Catalog" column—which will sort by descending dates.

Binding Operational Directive (BOD) 22-01: Reducing the Significant Risk of Known Exploited Vulnerabilities established the Known Exploited Vulnerabilities Catalog as a living list of known Common Vulnerabilities and Exposures (CVEs) that carry significant risk to the federal enterprise. BOD 22-01 requires Federal Civilian Executive Branch (FCEB) agencies to remediate identified vulnerabilities by the due date to protect FCEB networks against active threats. See the BOD 22-01 Fact Sheet for more information.

Although BOD 22-01 only applies to FCEB agencies, CISA strongly urges all organizations to reduce their exposure to cyberattacks by prioritizing timely remediation of Catalog vulnerabilities as part of their vulnerability management practice. CISA will continue to add vulnerabilities to the catalog that meet the specified criteria.

Please share your thoughts. We recently updated our anonymous Product Feedback Survey and we'd

Study the above news alert published on an official web site of the US government, the one of the Cybersecurity & Infrastructure Security Agency (CISA, "America's Cyber Defense Agency").

1) How do you know (simply) this is an official website of the US government ?

Site URL indicates a protocol with security features (HTTPS) and a governmental site (.gov site name).

2) Which procedure and computer security management approach can you infer for the organization of US governmental agencies from the analysis of this alert ?

Federal civilian agencies should integrate in their IT management procedures prompt remediation of some software vulnerabilities.

The list of the vulnerabilities that they have to manage in priority is maintained and published by CISA. These vulnerabilites are selected by the US government cybersecurity agency due to evidence of active exploitation.

The above alert note indicate two such known vulnerabilities. IT managers of all federal civilian agency should asses the relevance of such alerts to their system (by verifying if they have a vulnerable system or not). If that is the case, they are ordered to take appropriate action to eliminate that risk (either by applying corrective actions or by neutralizing these systems).

It is certainly also needed to investigate for potential active exploitation of the vulnerabilities and recent intrusion through these breaches if existing.

Reporting in due time to CISA or via federal security dashboards management procedures is also certainly demanded to pursue the application of the directive.

3) As private company *contractor* of the US government, how would you organize security management in your organization with respect to this policy (both from the cost-efficiency and customer satisfaction point of views) ?

Evidently, your US govt customer would appreciate if you implement the same management procedure as they do internally.

However, CISA focuses on US govt IT systems, so some adaptation may be in order and/or accepted by such a customer. Such an adaptation should been two fold:

- on one side, if you use systems or software scrutinized by CISA, the agency is actually delivering you an interesting service for adequate protection and if you know such systems do not exist in your own IT, you can indicate that to your customer and easily mark alerts irrelevant ;

- on the other side, if you use equivalent systems not especially taken into account but CISA, you may be on your own to reach the same level of trust.

Efforts to follow CISA directives may or may not be the subject of a paying service by your company. That is not the topic of this exam, but note that such billing may not be actually considered legitimate. (The protection resulting from such activity does not only benefit to the US govt… it also protect the private company own assets.) So your mileage may vary on the cost efficiency of this security task (and you should probably delegate the topic to the commercial department or the appropriate executive to see if they want to cash or expend on it).

As a final note, one can underline however that, overall, fully ignoring such alerts sounds extremely unwise.

Question 4

You are tasked with proposing a secure software distribution mechanism for an experimental Unix-like operating system by a research & education academic project. Propose an overall system for such a task by outlining the design principles and the target properties.

Each participant to the project generates a personal private/public key pairs (à la GnuPG). These public keys are certified by mutual signing at the next research project seminar. (Alternatively, you can use individual authentication certificates delivered by each academic institutions if you are more into centralized PKIs than good old academic freedom[1]…)

One of the academic partners is tasked for software building and distribution setup. He/She manages a "project key" (changed for each major version or on a time-based period) and signs the distribution using this system.

The project public key is available (along with signatures and distribution files). Any user can verify the software distribution files integrity using the signatures and the project key.

The correct attribution of the project key can be certified by all project participants through the peer-to-peer trust relationship built by the mutual signing of personal keys during signing sessions. (An external partner would have to meet one of the project participants – once – in order to complete this step.)

(Alternatively, if you have relied on centralized PKIs, you need primarily to trust the university PKI of the project participant in charge of software distribution.)

In a second step, you are hired by a high profile aerospace civilian manufacturer that would like to reuse your system for onboard software updates of embedded systems on operational planes.

Discuss the difficulties in reusing your system in this context ? Do you think some workarounds are feasible (or some weaknesses acceptable) ?

In the academic system: individuals are the actual foundations of the peer to peer key management solution. In a company context, users would probably find pretty counter-intuitive to check (the integrity of the public key used to) software integrity via such an apparently ad-hoc personal solution. They would resort to single keys per company (which means that the security of the public key distribution would be probably rather informal) or to companies PKI.

Therefore, for example, a weak point of the system would certainly be the installment of public keys at the verification point (e.g. on data loaders in planes, or on maintenance terminals). Probably the worst impact then (IMHO) is the difficulty to widely deploy and automate security verifications, unlike in some open source Unix-systems. (In their case, this automation relies on the strongly connected trust networks of individual public/private keypairs actively used by their holders.) Hence, security verifications made possible by cryptography will have to be double checked by the operators actually doing the software installation in the field (using their own key sources). This heavily diminishes the interest of this verification (because, if you have to trust the operators anyway, then the overall trust in the system comes from organizational measures more than technical ones).

[1]Or if you anticipated the next question.