ECM2426: Network and Computer Security

Continuous Assessment

UNIVERSITY OF EXETER

Academic year: Term 1, 2022/23
Hand-in date: 2022-12-08 (Week 11)

Abstract

This continuous assessment covers topic taught in the module ECM2426 "Computer and Network Security". The continuous assessment focuses on your understanding of the practical aspects of security that you acquired during the labs, lecture, and practical exercises in the lecture notes.

This continuous assessment requires you work a virtual machine that is assigned to you individually, and to submit your solution in a very specific format. Read the instructions carefully, it is your responsibility to ensure that your submission adheres to the specified format. Please ensure you read the entire document before you begin the assessment

General Instructions

System Access

You can access all necessary tools on VMs provided on Azure Labs, i.e., the same system that we are also using for the workshop. The VM for the CA is named "ecm2426-ca-2022" and it uses the default credentials:

- User: `lh`
- Password: `Initial1`

Note that sometimes the Azure dashboard seems to hang in a "starting" state. This seems to be a UI bug, i.e., you can already connect to the virtual machine. You can access the RDP connection details in the dashboard, even if the VM is not running.

System Resources

Note that access to the cloud system is expensive and billed "by the minute". Thus, please always switch VMs off, after you finished your work. To mitigate the risk of "cost explosion", the VM has a maximum amount of hours assigned, which will be increased over time.

Until the end of week 10, you can ask for more time by sending am email (body can be empty) with the subject "`ECM2426 CA: Please check available VM time''` to the module lead. The deadline for such requests is the last Monday before the submission deadline, precisely:

2022-12-05T17:59:59 GMT

Referencing and Academic Conduct

You are required to cite the work of others used in your solution and include a list of references, and must avoid plagiarism, collusion and any academic misconduct behaviours. For further information and guidance, please take the self-learning "Academic Honesty and Plagiarism".

Furthermore, you will need to work on the VM assigned to you. Not following this rule, will also consider as academic misconduct.

Marking Scheme

The marks for the individual questions are denoted for each (sub)-question. In addition, please note:

- You need to work on the VM assigned to you and only to you. Some tasks result in solutions that are unqique to the VM. Hence, **solutions obtained on a different VM will result in zero marks.**
- Your submission needs to follow a specific format, detailed in the next section. Not following this format might result in zero marks. As part of the VM, a script is supplied that does provide a basic check of the required format. You are encouraged to make use of this script. If this script cannot recognise your solution/submission, it will likely result in zero marks.

Submission

The submission will be done via "eBart (ZIP)". The zip file should contain exactly the following files (and file hierachy):

```
solution
├── ecm2426.ods
├── exercise02
│   ├── data.tgz
│   └── data.sig
├── exercise04
│   └── ecm2426.AnB
└── exercise05
    └── main.py
```

You should complete your coursework in the folder `/home/lh/solution`, which also contains a template of the file `ecm2426.ods`. After completing the coursework, you should create a zip-archive only containing the `solution` folder. You can create such a zip archive on the Linux command line as follows:

Bash

```
lh@ML-RefVm-326614:~$ zip -r solution.zip solution
```

The VM contains a tool `check-submission` that you can use to check the validity of your created zip file. **It is strongly recommended** that you use this tool to check the

compliance of your zip file to the required submission format, before submitting your solutions via eBart:

```Bash
lh@ML-RefVm-326614:~$ ls
solution.zip
lh@ML-RefVm-326614:~$ check-submission
```

This script will the syntactic compliance of your submission, i.e., if the expected file names exist and if the spreadsheet has been completed. It does not check for the correctness of your submission. Note that a successful run of the `check-submission` script does neither guarantee that your submission is complete nor that it follows the guidelines. *Checking that the submission is complete and adheres to the documented file hierarchy is your responsibility!*

You can, e.g., copy the created archive to your local machine using `scp` or `sftp`. Note that also for scp (and sftp), you need to configure the correct port. Assuming that you can connect to the VM using the following command:

```Bash
achim@logicalhacking:~$ ssh -p 65042 lh@host.azure.com
```

Then you can use

```Bash
achim@logicalhacking:~$ scp -P 65042 lh@host.azure.com:solution.zip .
```

to copy the archive `solution.zip` to your local machine. Of course, you can also use a graphical front-end, as, e.g., provided by putty or winscp.

1 Foundations & Access Control

Exercise 1 **(20 marks)**

The virtual machine `2022_ECM2426_CA` has several regular user accounts configured. Amongst them the user `alice` with the password `alice2000`.

E.1.1. List all users with their login that have numerical user id larger than 1000.
Submit your answer in cell B2 of the spreadsheet `ecm2426.ods`. Separate the logins with a comma, e.g., "`joe, jane, root`".

E.1.2. Execute a guessing attack on the password for the user `bob`. What is the password of `bob`.
Submit your answer in cell B3 of the spreadsheet `ecm2426.ods`.

E.1.3. What is the content of the file `/home/alice/exercise1/flag`.
Submit your answer in cell B4 of the spreadsheet `ecm2426.ods`.

E.1.4. List *all* users with their login that can read the file `/home/alice/exercise1/flag` . Separate the logins with a comma, e.g., "`joe, jane, root''`.
Submit your answer in cell B5 of the spreadsheet `ecm2426.ods`.

2 Cryptography, Signatures & PKIs

Exercise 2 **(15 marks)**

Alice (login: `alice`, password: `alice2000`) has an X.509 certificate stored in her home directory in a file called `/home/alice/exercise02/webserver.pem`. She plans to use for her website. Moreover, in the file `/home/alice/exercise02/key.pem` she also has an asymmetric key pair stored. The key has no password set.

E.2.1. Name the the signature algorithm that has been used for signing the certificate.
Submit your answer in cell B6 of the spreadsheet `ecm2426.ods`.

E.2.2. List *all* domains for which the certificate is valid.
Submit your answer in cell B7 of the spreadsheet `ecm2426.ods`. Separate the domains with a comma, e.g., "`127.0.0.1, *.exeter.ac.uk, www.oxford.ac.uk`".

E.2.3. Sign the source code archive `/home/alice/exercise02/data.tgz` using the provided key of alice. Store the signature as `data.sig`.
Submit the files `data.tgz` and `data.sig` as part of your zip file in a folder `solution/exercise02` .

Gor this excercise, you might want to use the `openssl` command line tool, which is provided on the virtual machine.

3 Security Protocols

Exercise 3 **(15 marks)**

The provided VM allows you to access a network that uses the IP range `172.17.0.*` and your virtual machine is connected to this network via the network interface `docker0`. This network contains a number of computers that might have insecure services installed.

As user `lh`, analyse the network traffic on the `172.17.0.*` (e.g., using Wireshark) and answer the following questions related to the http session:

E.3.1. Give the full URL of the password protected web area accessed in this session.
Submit your answer in cell B9 of the spreadsheet `ecm2426.ods`.

E.3.2. Give the username of the user accessing the protected area.
Submit your answer in cell B10 of the spreadsheet `ecm2426.ods`.

E.3.3. Give the (cleartext) password of the user accessing the protected area.
Submit your answer in cell B11 of the spreadsheet `ecm2426.ods`.

4 Formal Analysis of Security Protocols

Exercise 4 **(15 marks)**

The file `/home/lh/exercise04/ecm2426.AnB` contains a security protocol specification in Alice&Bob notation.

With the help of `ofmc`,

E.4.1 Complete the initial knowledge so that the specification is executable (for this, you need to update the initial knowledge of at most one of the roles/agents in the knowledge section):

E.4.1.a. Which role/agent has an imcomplete initial knowledge?
Submit your answer in cell I2 of the spreadsheet `ecm2426.ods`.

E.4.1.b. Which facts do you need to add to the initial knowledge?
Submit your answer in cell J2 of the spreadsheet `ecm2426.ods`.

E.4.2. Fix the protocol so that ofmc does not report an attack. For this, you need to update one message in the actions sections.
Submit the updated message in cell K2 of the spreadsheet `ecm2426.ods`.

Include the completed AnB file in your sumission (`solution/exercise04/ecm2426.AnB`).

5 Static Security Testing

Exercise 5 ***Static Analysis*** **(15 marks)**

In the directory Analyse the application using version 1.6.2 of the static security scanner "bandit" (`https://github.com/PyCQA/bandit`).[1] The bandit tool is part of the provided VM and the version installed on the VM will be used as a reference for assessing your solution.

You can run bandit as follows, and it should report three potential security issues (the actual output will provide more details):

```
achim@logicalhacking:~/coursework$ bandit -r todo-app
Test results:
>> Issue: [B413]
>> Issue: [B304]
>> Issue: [B608]
```

Bash

The ToDo-App has several real vulnerabilities, but not all of them are necessarily found by bandit and neither are all findings of bandit real vulnerabilities. Moreover, we only focus on software vulnerabilities in the application itself, i.e., we do not consider configuration issues (for example, using http instead of https or the fact that the application is configured to run in development mode, are out of scope for this exercise).

For each of the three findings reported by bandit:

> Decide if it is a true positive or not. For this exercise, we only rate findings as true positives, if they can be exploited by an attacker.
>
> If it is a true positive, name the CWE identifier if of the issue. Write N/A (not applicable) if the finding is a false positive,

Submit your solution in the cells B15 to B20 in `ecm2426.ods` and copy the file `todo-app/main.py` into the zip archive in your zip file as `solution/exercise05/main.py`.

[1] You can check the version of bandit by invoking bandit with the argument `--version`. Please ensure that you are using version 1.7.4, as provided on the VM.

6 Dynamic Security Testing

Exercise 6 **(20 marks)**

The VM has another variant of the ToDo-App installed that you can explore at `http://127.0.0.1:5000`, using a web browser that runs on the VM.

The user `lh` can use `sudo` to start/stop the ToDo-application:

```Bash
lh@ML-RefVm-326614:~$ sudo systemctl start todo
lh@ML-RefVm-326614:~$ sudo systemctl stop todo
```

During the start of the application, the database is reset to an empty state.

Assume you are a penetration tester tasked to analyse this application. You should test the application for Cross-Site Scripting (XSS) and SQL Injection (SQLi) vulnerabilities.

In particular, you should find:

E6.XXS. One Cross-Site Scripting (XSS) vulnerability:

- **E6.XSS.1.** Name the URL that has an entry field with a XXS vulnerability. *Submit your answer in cell B21 of the spreadsheet `ecm2426.ods`.*
- **E6.XSS.2.** Name the vulnerable input field. *Submit your answer in cell B22 of the spreadsheet `ecm2426.ods`.*
- **E6.XSS.2.** Name the URL at which the XSS vulnerability can be observed (i.e., the XSS payload is triggered). *Submit your answer in cell B23 of the spreadsheet `ecm2426.ods`.*
- **E6.XXS.3.** Name the payload that triggers the XSS vulnerability. *Submit your answer in cell B24 of the spreadsheet `ecm2426.ods`.*

E6.SQLi One SQL Injection (SQLi) vulnerability:

- **E6.SQLi.1.** Name the URL that has an entry field with a SQL inection vulnerability. *Submit your answer in cell B25 of the spreadsheet `ecm2426.ods`.*
- **E6.SQLi.2.** Name the vulnerable input field. *Submit your answer in cell B26 of the spreadsheet `ecm2426.ods`.*
- **E6.XXS.3.** Name a payload that successfully deletes a database table in the ToDo-app. *Submit your answer in cell B27 of the spreadsheet `ecm2426.ods`.*