NTNU
Institutt for datateknikk
og informasjonsvitenskap

Exam

TDT4237 – Software Security

Thursday December 18th 2014, 09:00 - 13:00

This exam has been created by Lillian Røstad. Quality assurance by Guttorm Sindre.
Contact person during the exam is Lillian Røstad (contact: 994 00 628).

Language: English

Category D:
No printed or hand-written materials allowed. Approved calculator allowed.

Results available: Monday January 19th 2015

Read each task carefully. Identify what the task asks for.
If you find that information is missing in a task you are free to make the assumptions you consider necessary, but remember to explain and clearly mark your assumptions.

Task 1 (30%)

a) What information are you looking for in the information gathering phase of a security test?

The OWASP testing guide (v.3.0 – v.4.0 was not published yet at the start of the semester) gives an in-depth description of this in chapter 4.1 – information gathering.
An answer is therefore expeted to include:

- *information about the application structure (e.g. Enough to create a pagemap), including different zones (e.g. open, authenticated)*
- *information about data flow within the application – e.g. Parameters and values, get and post*
- *information about the infratsucture/platform – webserver, database, programming language*

b) What are the main properties of a cryptographic hash function?

One-wayness, truly random, collision free, and possibly: random length input, fixed length output. P140-141 in the Security Engineering book.

c)OpenSAMM defines four Business Functions and for each of these three Security Practices. What security practices are included in the Verification Business Function? Briefly explain how each works.

The three security practices are:
Design review *- The Design Review (DR) Practice is focused on assessment of software design and architecture for securityrelated problems. This allows an organization to detect architecture-level issues early in software development and thereby avoid potentially large costs from refactoring later due to security concerns. Beginning with lightweight activities to build understanding of the security-relevant details about an architecture, an organization evolves toward more formal inspection methods that verify completeness in provision of security mechanisms. At the organization level, design review services are built and offered to stakeholders. In a sophisticated form, provision of this Practice involves detailed, data-level inspection of designs and enforcement of baseline expectations for conducting design assessments and reviewing findings before releases are accepted.*
Code review *- The Code Review (CR) Practice is focused on inspection of software at the source code level in order to find security vulnerabilities. Code-level vulnerabilities are generally simple to understand conceptually, but even informed developers can easily make mistakes that leave software open to potential compromise. To begin, an organization uses lightweight checklists and for efficiency, only inspects the most critical software modules. However, as an organization evolves it uses automation technology to dramatically improve coverage and efficacy of code review activities. Sophisticated provision of this Practice involves deeper integration of code review into the development process to enable project teams to find problems earlier. This also enables organizations to better audit and set expectations for code review findings before releases can be made.*
SecurityTesting *- The Security Testing (ST) Practice is focused on inspection of software in the runtime environment in order to find security problems. These testing activities bolster the assurance case for software by checking it in the same context in which it is expected to run, thus making visible operational misconfigurations or errors in business logic that are difficult to otherwise find. Starting with penetration testing and high-level test cases based on the functionality of software, an organization evolves toward usage of security testing automation to cover the wide variety of test cases that might demonstrate a vulnerability in the system. In an advanced form, provision of this Practice involves customization of testing automation to build a battery of security tests covering application-specific concerns in detail. With additional visibility at the organization level, security*

testing enables organizations to set minimum expectations for security testing results before a project release is accepted

d) The first two steps of the RMF has a focus on understanding the business context and linking technical risks to business risks. Why is this important?

The purpose is to link the risk analysis with business priorities. Most importantly it allows you to judge consequences for the business, not just for the program itself. It is also important to understand that the system and security of the system is there to support the business goals.

e) Explain the core properties of the Biba and Bell LaPadula security models. How are they different?

A good answer should include the simple security property and the star property. Biba: no write up – no read down (integrity). Bell LaPadula: no read up – no write down (confidentiality).

f) What is a buffer overflow vulnerability? How do you test for buffer overflows?

A buffer overflow vulnerability is made possible by unsafe memory management. Preconditions: unsafe programming language and poor programming – using unsafe functions. A buffer overflow vulnerability is present when X bytes has been allocated for a buffer, but the program actually allows Y bytes (where Y > X) to be written to that buffer. So the result is that some part of memory is overwritten. What happens depends on where it occurs. You test for buffer overflows by provoking a program with very large inputs.

Task 2 (30%)

A university is testing a new system for managing exams digitally. The system allows the exam tasks to be distributes to a dedicated set of computers that the students can use to answer and upload their exam. The computers are set up so they can only communicate with the server of the exam management system. No other communication is allowed. All communication with the exam management server is encrypted. The system is based on web-technology and set up with single sign-on so that the students are using their ordinary username and password issued by the university to log on to the system.

Your task is to perform a security analysis of the model, by using three different types of threat modeling: data flow diagrams, attack trees and misuse cases. For each of the threat models you create, explain the pros and cons of this type of threat model.

From the questions at the day of the exam it seems some of the students were unclear about what a security analysis is – should they do more than threat modelling?
What is expected here is:

- *as always showing both method knowledge as well as providing meaningfull content*
- *a good answer have to include all three types of models and they have to be used right*
- *DFDs provide an overview of the system's attack surface – where does data flow? Who has access to what parts of the system?*
- *Misuse cases allow mapping from wanted functionality via how it can be exploited to necessary security functionality*
- *Attack trees put the attackers goal in focus and how that can be achieved. (Often) a more technical and detailed analysis than misuse cases.*

All three used together provides a very good overview. Note that the students are not expected to include every possible threat model for this system here, due to time constraints, but all three types of models have to be present, and used in a meaningful way that demonstrates how they complement each other. The students are expected to explain their models – just providing models without any explanation of their thinking is not a great answer.

Task 3 (20%)

For each of the code snippets listen below, your task is to:

- Identify the security vulnerability.
- Explain why this is a security issue.
- Fix the code. You may use pseudo-code for this. Remember to explain your solution.

Code snippet 1

```
<?php
$email = "abc123@sdsd.com";
$regex = '/^[_a-z0-9-]+(\.[_a-z0-9-]+)*@[a-z0-9-]+(\.[a-z0-9-]+)*(\.[a-z]{2,3})$/';
if (preg_match($regex, $email)) {
 echo $email . " is a valid email. We can accept it.";
} else {
 echo $email . " is an invalid email. Please try again.";
}
?>
```

Example from: http://stackoverflow.com/questions/18094623/whats-wrong-with-my-code-preg-match.

Issue: The issue her is that the developer attempts to do input validation manually – which is never a good idea. (ref OWASP: How to write insecure code. Input validation – let developers validate their way. To create truly insecure code, you should try to validate as many different ways as possible, and in as many different places as possible. Don't bother with a standard way of doing validation, you're just cramping developer style.) PHP has a built-in function to do this for you. It is actually used in code snippet 4 – supposed to be a hint. This is not a great regex for validation of email addresses. Many students will probably point that out but it is not the vulnerability we are looking for here.

Code snippet 2

```
try {
openDbConnection();
}
catch (Exception $e) {
echo 'Caught exception: ', $e->getMessage(), '\n';
echo 'Check credentials in config file at: ', $Mysql_config_location, '\n';
}
```

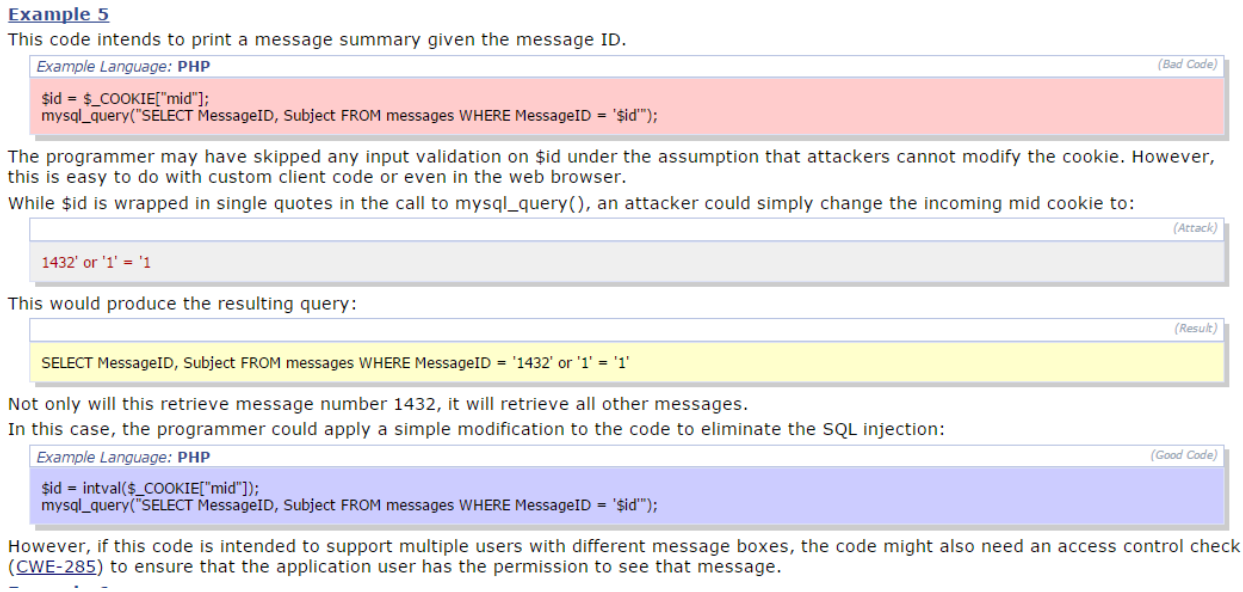
Issue: Poor error management. This code tries to open a database connection, and prints any exceptions that occur. If an exception occurs, the printed message exposes the location of the configuration file the script is using. An attacker can use this information to target the configuration file (perhaps exploiting a Path Traversal weakness). If the file can be read, the attacker could gain credentials for accessing the database. The attacker may also be able to replace the file with a malicious one, causing the application to use an arbitrary database.

http://cwe.mitre.org/data/definitions/209.html

Code snippet 3

```
$id = $_COOKIE["mid"];
mysql_query("SELECT MessageID, Subject FROM messages WHERE MessageID = '$id'");
```

Issue: SQLinjection



Example 5

This code intends to print a message summary given the message ID.

Example Language: **PHP** (Bad Code)

```
$id = $_COOKIE["mid"];
mysql_query("SELECT MessageID, Subject FROM messages WHERE MessageID = '$id'");
```

The programmer may have skipped any input validation on $id under the assumption that attackers cannot modify the cookie. However, this is easy to do with custom client code or even in the web browser.

While $id is wrapped in single quotes in the call to mysql_query(), an attacker could simply change the incoming mid cookie to:

(Attack)

```
1432' or '1' = '1
```

This would produce the resulting query:

(Result)

```
SELECT MessageID, Subject FROM messages WHERE MessageID = '1432' or '1' = '1'
```

Not only will this retrieve message number 1432, it will retrieve all other messages.

In this case, the programmer could apply a simple modification to the code to eliminate the SQL injection:

Example Language: **PHP** (Good Code)

```
$id = intval($_COOKIE["mid"]);
mysql_query("SELECT MessageID, Subject FROM messages WHERE MessageID = '$id'");
```

However, if this code is intended to support multiple users with different message boxes, the code might also need an access control check (CWE-285) to ensure that the application user has the permission to see that message.

http://cwe.mitre.org/data/definitions/89.html

Code snippet 4

Issue: CSRF

2

CSRF for the win.

```
<?php
$newEmail = filter_input(INPUT_POST, 'email', FILTER_SANITIZE_EMAIL);
$pdoStatement = $pdoDb->prepare('UPDATE user SET email=:email WHERE ID=:id');
$pdoStatement->execute(array(':email'=>$newEmail, ':id'=>$_SESSION['userId']));
```

You feel safe with this kind of code. All is good your users can change their emails without injecting SQL because of your code. But, imagine you have this on your site http://siteA/, one of your users is connected. With the same browser, he goes on http://siteB/ where some AJAX does the equivalent of this code :

```
<form method="post" action="http://site/updateMyAccount.php">
  <p>
    <input name="email" value="badguy@siteB"/>
    <input type="submit"/>
  </p>
</form>
```

Your user just got his email changed without him knowing it. If you don't think this kind of attack is dangerous, ask google about it

To help against this kind of attacks, you can either :

- Check your user REFERER (far from perfect)
- Implement some tokens you had to your forms and check their presence when getting your data back.

Another one is session hijacking. One of the methods to do it is piggybacking. If your server accepts non cookie sessions, you can have URLs like http://siteA/?PHPSESSID=blabla which means your session ID is blabla.

An attacker can start a session and note his session ID, then give the link http://siteA/?PHPSESSID=attackerSessionId to other users of your website. When these users follow this link, they share the same session as your attacker : a not logged session. So they login. If the website does not do anything, your attacker and your user are still sharing the same session with the same rights. Bad thing if the user is an admin.

To mitigate this, you have to use session_regenerate_id when your users credentials change (log in and out, goes in administration section etc.).

share | improve this answer

edited Nov 23 '09 at 14:36

answered Nov 23 '09 at 14:18

Arkh

5,990 ● 21 ● 30

add a comment

http://stackoverflow.com/questions/1783137/examples-of-vulnerable-php-code

Task 4 (20%)

The Risk Management Framework enables you to manage risks in the Software Development Lifecycle (SDL). The output of the RMF depends on where you are in the SDL.

You have been given the task of performing a security review of an existing system for ordering movie tickets. The system allows the users to see information about upcoming movies, see the schedule for the week, book and pay for tickets. Issued tickets are sent to the customer's e-mail address.

The RMF has been used by the team developing this system, so you are lucky. You have a lot of information to base your review on. Below is an excerpt of a spreadsheet that summarizes identified business risks, corresponding technical risks, and implemented mitigation techniques.

	Probability	Consequence	Risk	Mitigation (requirements)
BR1: System unavailable				
TR1: DDoS				
TR1.1: Botnet attack	M	H	H	System shall be able to handle 100 000 reuests per second.
TR2: System crash				
TR2.1: Server hacked	L	H	M	All servers included in the system shall always be up-to-date
BR2: Payment card data stolen				
TR1:System hacked				
TR1.1: SQL-injection to dump data	L	M	M	Input validation shall be performed all on information that is used to communicate with database.
TR1.2: Utilize vulnerability in database server	L	H	M	All servers included in the system shall always be up-to-date
TR1.3: Utilize default account on database server	L	H	M	All servers included in the system shall be properly hardened.
TR2: Insider access to database				
TR2.1: Operations engineer misuse	H	H	H	Only allow personal users for traceability. Review logs
TR2.2: Application admin misuse	M	M	M	Only allow personal users for traceability. Review logs periodically. Extracting data shall be a two-step process

To perform the security review, your first action is to test if the mitigation techniques are in place and works as intended. Using the RMF-excerpt above, your task is to create a risk-based test plan to use in your review.

–

From the questions on the day of the exam it seems some students were confused here if they were supposed to do an entire security review and what that meant. What we are looking for is just the risk-based test plan. And of course an explanation of it. How they assess risks. And details on each test including necessary inputs and expected result. They all did this in exercise 3 so this should not be hard. But we are looking for sound arguments in their explanations. A table with no other explanamtion is definitely not a good answer.