IT-Sikkerhed, Assignment 3

ITS - 2021

Department of Computer Science
University of Copenhagen

M. Ali
M.Chleih

April 2, 2022

Review Questions

6.1

What are three broad mechanisms that malware can use to propagate?
infection of existing executable or interpreted content by viruses that is subsequently spread to other systems by modifying them. The modification includes injecting the original code with a routine to make copies of the virus code, which can then go on to infect other content.

Exploit of software vulnerabilities either locally or over a network by worms or drive-by-downloads to allow the malware to replicate.

Social engineering attacks that convince users to bypass security mechanisms to install Trojans, or to respond to phishing attacks.

6.2

What are four broad categories of payloads that malware may carry?
System Corruption
These payloads takes place when certain conditions are met. Usually with the intent to simply spread or do some real damage like, data destruction, ransomware (by encrypting a users file and requiring payment), real-world damage (damage to physical hardware) and logic bombs, which is code embedded in the malware that is set to "explode" when certain conditions are met, resulting in altering or deletion of data or entire files, cause a machine to halt, or do some other damage.

Attack Agent - Zombie, Bots
In this case the malware subverts the computational and network resources of the infected system for use by the attacker and secretly takes over another Internet-attached computer then uses that computer to launch or manage attacks that are difficult to trace to the bot's creator.

Information Theft - Keyloggers, Phising, Spyware
Here malware gathers data stored on the infected system for use by the attacker.

Common targets include the user's login and password credentials to banking, gaming, which the attacker then uses to impersonate the user to access these sites for gain.

Stealthing - Backdoors, Rootkits
This technique hide its presence in the infected systems, and then tries to provide access to to that system. It also attacks the integrity of the system. It includes backdoors (trapdoor) which is a secret entry point into a program without going through security clearance. Rootkit is also includes here, which is set of programs installed on a system to maintain covert access to that system with administrator privileges, while hiding evidence of its presence to the greatest extent possible.

6.8

What is a "drive-by-download" and how does it differ from a worm?
A Drive-by-Download is an attack using code on a compromised website that exploits a browser vulnerability to attack a client system when the site is viewed whereas a worm is a computer program that can run independently and can propagate a complete working version of itself onto other hosts on a network, by exploiting software vulnerabilities in the target system, or using captured authorization credentials.

13.2

List and briefly define three cloud service models.
Software as a Service (SaaS)
SaaS provides service to customers in the form of software, specifically application software, running on and accessible in the cloud. SaaS enables the customer to use the cloud provider's applications running on the provider's cloud infrastructure. The use of SaaS avoids the complexity of software installation, maintenance, upgrades, and patches.

Platform as a Service (PaaS)
PaaS provides service to customers in the form of a platform on which the customer's applications can run. PaaS enables the customer to deploy onto the

cloud infrastructure customer-created, or acquired, applications. A PaaS cloud provides useful software building blocks, plus a number of development tools, such as programming language tools, run-time environments, and other tools that assist in deploying new applications. In effect, PaaS can be seen as an OS but in the cloud.

Infrastructure as a Service (IaaS)
The customer has access to the resources of the underlying cloud infrastructure. The cloud service user does not manage or control the resources of the underlying cloud infrastructure but has control over operating systems, deployed applications, and limited control of selected networking components. IaaS provides VMs and other virtualised hardware and operating systems. IaaS offers the customer processing, storage, networks, so that the customer is able to deploy and run arbitrary software.

13.4

Describe some of the main cloud-specific security threats.
Abuse and nefarious use of cloud computing
For many Cloud Service Providers (CSP), it is easy to register and begin using cloud services. This enables attackers to get inside the cloud to conduct various attacks, such as spamming, malicious code attacks, or denial of service.

Insecure interfaces and API
Cloud service providers expose a set of software interfaces or APIs that customers use to manage and interact with cloud services. The security and availability of general cloud services is dependent upon the security of these basic APIs. Includes authentication and encryption.

Malicious insiders
Under the cloud computing paradigm, an organisation relinquishes direct control over many aspects of security. This confers a level of trust onto the cloud service provider. This give risks to malicious insider activity. Cloud architectures necessitate certain roles that are extremely high risk. These insiders could be system administrators for cloud service providers.

Account or service hijacking
Account and service hijacking, think stolen credentials, remains a top threat. With stolen credentials, attackers can often access critical areas of deployed cloud computing services, allowing them to compromise the confidentiality, integrity, and availability of those services.

Unknown risk profile
In using cloud infrastructures, the client necessarily cedes control to the cloud provider on a number of issues that may affect security.

Problems

6.10

Should you be suspicious that a game wants these types of permissions? What threat might the app pose to your smartphone, should you grant these permissions and proceed to install it? What types of malware might it be?
A game asking for these permissions is in fact suspicious. Coupled with the fact that the game was downloaded from a "free marketplace" which has little to no quality control for the software it makes available, makes this game seem like a Trojan or spyware.
If given those permissions, the so-called game could collect details of the user's contacts or message history where confidential information might have been sent, and sends the collected information to the attacker through messages or other means.

6.11

What type of threat might this pose to your computer system should you indeed select the "Open" button? How could you check your suspicions without threatening your system? What type of attack is this type of message associated with? How many people are likely to have received this particular e-mail?
The prompt that the problem text is describing, is a security measure implemented by Adobe to warn the user of potential malware. The blank lines followed by the message "Click the 'Open' button to view this document.", is

something that has to be manually set by the attacker, hence is should raise a red flag. The PDF could contain a worm that could propagate throughout the network and cause damage. It could also contain a macro virus that contains a malicious payload.
If we want to make sure that the PDF is safe, we could use ant-virus software to scan the PDF, open the PDF in a virtual environment or even ask colleagues if they had opened the file since they have most likely received it as well considering the level of planning that went into the attack and the attackers knowledge about the workplace.

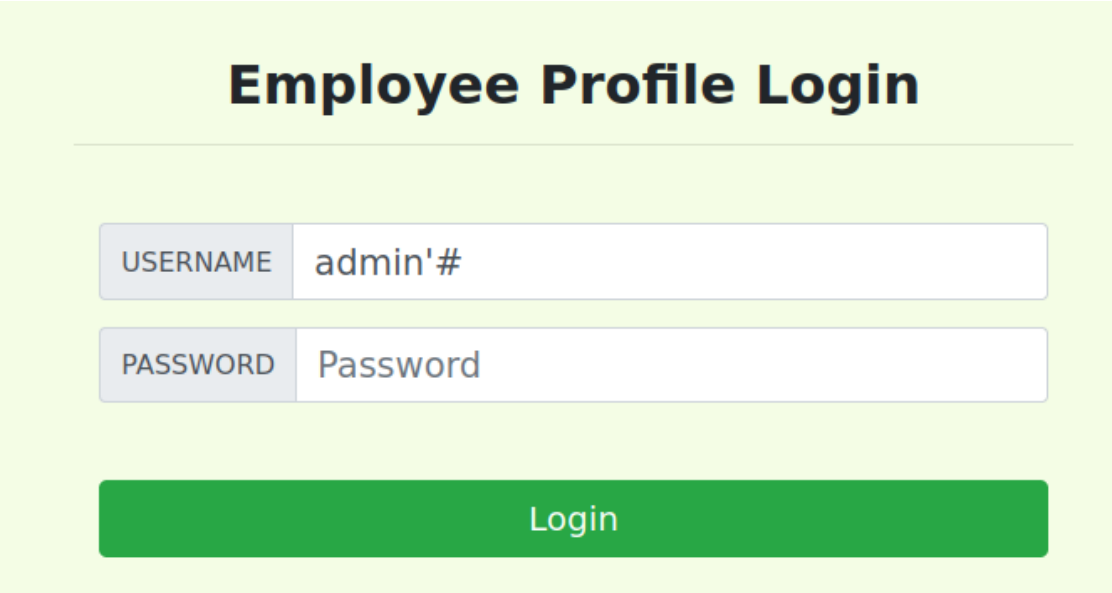
Seed Labs

After setting up the environment, we displayed all the profile information of the employee Alice as was asked, using the following SQL command:

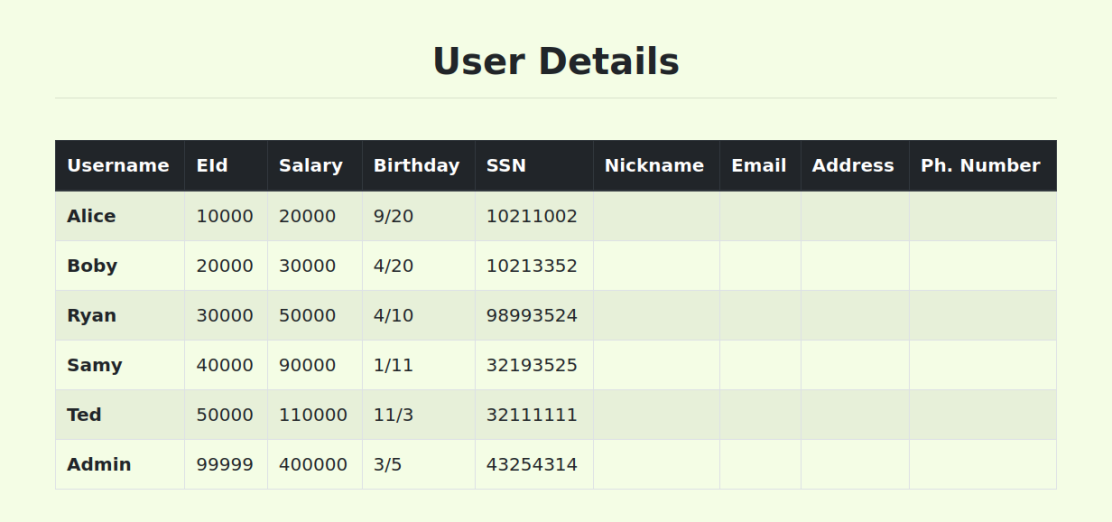
```
mysql> SELECT * FROM credential WHERE name="Alice";
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
| ID | Name  | EID   | Salary | birth | SSN      | PhoneNumber | Address | Email | NickName | Password                                 |
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
|  1 | Alice | 10000 |  20000 | 9/20  | 10211002 |             |         |       |          | fdbe918bdae83000aa54747fc95fe0470fff4976 |
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
1 row in set (0.01 sec)

mysql>
```

Task 2.1

After taking a look at the implementation in `unsafe home.php`, we realized that we could directly alter the SQL query since the data we give to `$input_uname` is directly compiled with the SQL query.
Thus in the `username` field we entered `admin'#`. This will effectively change the query so that it ends at `WHERE name= '$input_uname'` and the `#` will comment out the remaining commands.



After pressing `Login`:



User Details

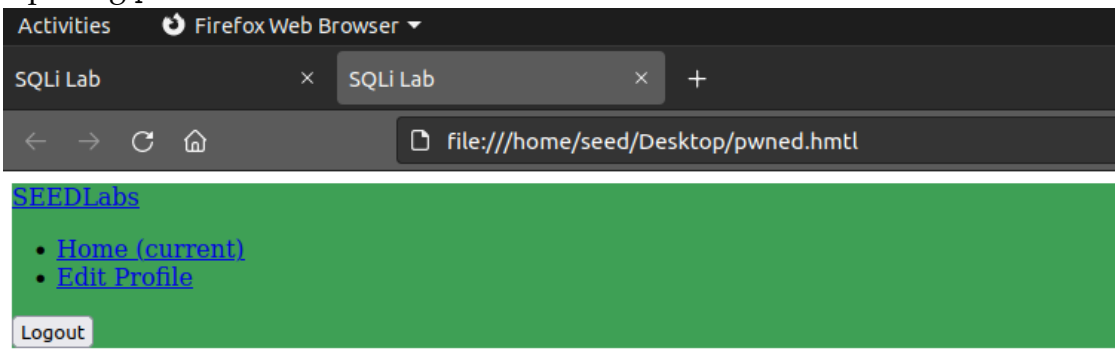
Username	EId	Salary	Birthday	SSN	Nickname	Email	Address	Ph. Number
Alice	10000	20000	9/20	10211002				
Boby	20000	30000	4/20	10213352				
Ryan	30000	50000	4/10	98993524				
Samy	40000	90000	1/11	32193525				
Ted	50000	110000	11/3	32111111				
Admin	99999	400000	3/5	43254314				

Task 2.2

Using the following command:
`curl 'http://www.seed-server.com/unsafe_home.php?username=admin%27%23Password='`
it downloads and prints out the `HTML` code, which we then can store into an `HTML` file and opening it thereafter:

```
[10/02/21]seed@VM:~/Desktop$ curl 'http://www.seed-server.com/unsafe_home.php?us
ername=admin%27%23&Password=' > pwned.hmtl
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  3365  100  3365    0     0  1643k      0 --:--:-- --:--:-- --:--:-- 1643k
```

Opening `pwned.html`:



User Details

Username	EId	Salary	Birthday	SSN	Nickname	Email	Address	Ph. Number
Alice	10000	20000	9/20	10211002				
Boby	20000	30000	4/20	10213352				
Ryan	30000	50000	4/10	98993524				
Samy	40000	90000	1/11	32193525				
Ted	50000	110000	11/3	32111111				
Admin	99999	400000	3/5	43254314				

Copyright © SEED LABs

Task 2.3

In the username input field, we tried to give the input "`admin' ; DELETE FROM credential WHERE name="Alice"`". This obviously did not work due to a countermeasure preventing the use of multiple SQL statements.
Since we knew that implementation in `unsafe_home.php` is using the PHP MYSQL API, we took a look at the documentation. Under the `Multiple Statements` sec-

tion, we read that in order to allow for the use of multiple statements, the function `multi_query()` had to be used. The implementation in `unsafe_home.php` is using the `query()` which does not allow for multiple statements.

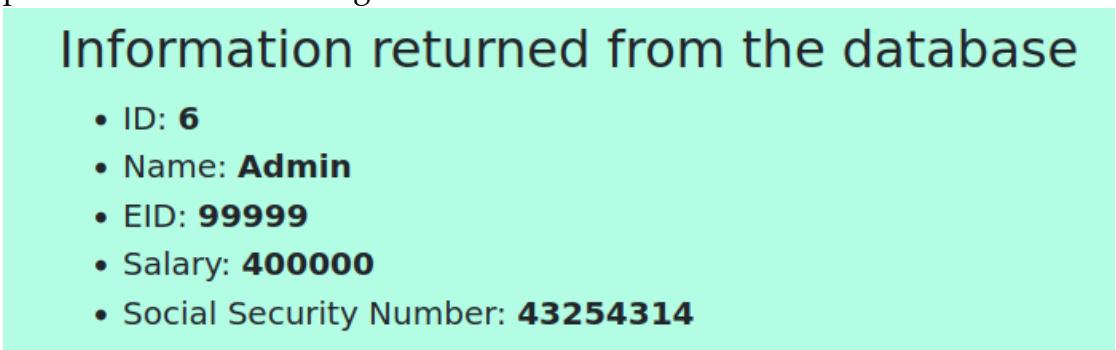
Task 4

We changed the source code in `unsafe.php` to include prepare statements. The modified code is the following:

```
$result = $conn->prepare("SELECT id, name, eid, salary, ssn
                    FROM credential
                    WHERE name= ? and Password= ?");
$result->bind_param("ss", $name, $pwd);
$result->execute();
$result->bind_result($bind_id, $bind_name, $bind_eid, $bind_salary,
                                                    $bind_ssn);
$result->fetch();
```

- We use the '?' symbol as a placeholder that temporarily takes the place of the data. Thus, when the query is compiled, it is compiled with the placeholders.
- "ss" tells mysql that the parameters for name and Password are strings and binds those parameters to name and Password.
- bind_result() binds the resulting columns to the variables.

Before we have made any changes to the implementation, when giving the input `admin'#` the following would show:



After using the same input but with prepare statements implemented: