COM-301 Computer Security
Exercise 6: Attacks

1. From the attack engineering process, what is the adversary exploiting in each of these cases: errors in the model (principals, assets, threat model,...), errors in the design (weaknesses in the design), or errors in the implementation (bugs, operational misuse). Justify:

   (a) A government uses the newest deep learning techniques to infer content from the length of encrypted traffic.

   (b) A hacker loads custom code to a device normally only accessible in wireless mode going through the device's USB port.[1]

   (c) An eavesdropper observes traffic encrypted with DES with a 56-bit key, and can decrypt it.

   (d) An attacker can read more data than allowed because the program does not check the number of characters requested by a read instruction.[2]

   (e) A student creates a copy of the exam that passes as the true one because the signature was made over an MD5 hash that is not collision resistant.

   (f) The polish decrypt german messages encrypted using Enigma because the keys were used more than once.[3]

2. We have learned in the class (Week 8) that the WEP protocol use of RC4 with a very short 40-bit IV lead to a vulnerability. Would this vulnerability be solved if AES-CTR was used instead of RC4 (assume that AES could work with a short IV)?

3. According to the STRIDE methodology, what threats are these?

   (a) Cersei learns that Stannis plans to attack King's Landing

   (b) Cersei denies knowing how Bran fell from the window

[1] True story: https://www.forbes.com/sites/aarontilley/2015/03/06/nest-thermostat-hack-home-network/6ca3d8543986

[2] True story: https://en.wikipedia.org/wiki/Heartbleed (short version: https://xkcd.com/1354/)

[3] True story (a bit tweaked :)): http://www.math.ucsd.edu/ crypto/students/enigma.html

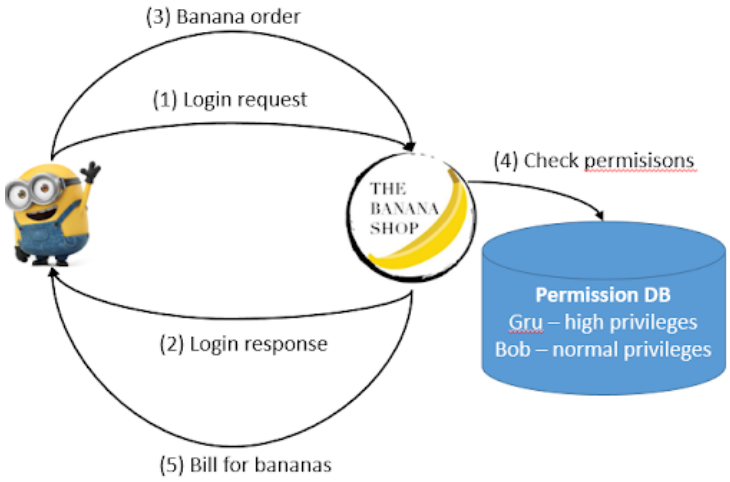

Figure 1: Scenario for Question 4.

(c) Tyrion intercepts a message from Jaime to Cersei and signs it as Tywin

(d) Cersei uses Tommen's "credentials" to rule the Small Council

Write your own example of Denial of Service.

4. Make a STRIDE analysis of the scenario in Figure 1. Write three possible threats, describe what flow they affect, and outline a possible countermeasure. [Note that this question is very open. Try to make a good security argument.]

5. Are the following statements True or False. Justify your answer:

   (a) Cross-site Reference Forgery requires tricking a user to introduce his password on a website controlled by the adversary.

   (b) Sanitization is the key to avoid injection attacks.

   (c) Complete mediation can help avoiding attacks based on misidentification of assets.

   (d) HTTP Sessions assume ambient authority and thus are susceptible to confused deputy problems.

6. Suppose a web page /site.com/index.php contains the following PHP script:

```
<?php echo "Hello". $_GET["username"];>
```

LIKE Operator	Description
WHERE CustomerName LIKE 'a%'	Finds any values that start with "a"
WHERE CustomerName LIKE '%a'	Finds any values that end with "a"
WHERE CustomerName LIKE '%or%'	Finds any values that have "or" in any position
WHERE CustomerName LIKE '_r%'	Finds any values that have "r" in the second position
WHERE CustomerName LIKE 'a_%_%'	Finds any values that start with "a" and are at least 3 characters in length
WHERE ContactName LIKE 'a%o'	Finds any values that start with "a" and ends with "o"

Figure 2: Use of the LIKE operator.

What vulnerability does this cause? Write a url that exploits this vulnerability to inform a third party stealingparty.com of the browser version that is being used by the user visiting the page. [4]

7. Consider the following server-side PHP code fragment in `http://site.com/index.php`

```
$sql = "SELECT username FROM MyUsers WHERE firstname LIKE '" .
    \$_GET["firstname"] . "'";
$result = $conn -> query(\$sql); // issue SQL query

if (\$result -> num_rows > 0) {
    print("Welcome back" . \$result) // if matching record
        found
} else {
    print("User not found") // otherwise
}
```

Here `$_GET["firstname"]` is a first name provided by the browser in the HTTP request.

`$sql` is a MySQL query that SELECTS the usernames of the table MyUsers that match with a specified pattern. For the purpose of this exercise the relevant syntax of the operator LIKE is shown in Figure 2.

The function print writes its argument to the Web page sent back to the browser.

How can the adversary learn all the usernames of users whose first name start by A? Can the adversary learn whether a target user is in the database or not? How? How can you avoid this vulnerability?

8. FooCorp has an internal web application that its employees can use to fill out travel vouchers. Unfortunately, FooCorp's system administrators have recently discovered that the voucher web application has cross-site request forgery (CSRF) vulnerabilities. FooCorp has a firewall that inspects all connections to the travel vouchers and checks that the cookies contain

[4] You can try out this vulnerability and many more on WebGoat → `https://www.owasp.org/index.php/Category:OWASP_WebGoat_Project`

correct authentication credentials. Does FooCorp's firewall prevent exploitation of the CSRF vulnerabilities in its travel voucher application? Justify