Zachary Golla

Security Homework 4

4/9/2019

**1. (15%) A computer system provides protection using the Bell-LaPadula model. How would a virus spread in the following two scenarios?**

**(a) The virus were placed at system low (the compartment being dominated by all other compartments).**

If the virus were placed at system low then “write up” would be allowed. This would give the virus the ability spread by replicating itself wherever it wants in the system.

**(b) The virus were placed at system high (the compartment dominating all other compartments).**

It would spread the same way, by replicating itself, but it would only be able to do it in the high compartment. Everything below the high compartment would be safe.

**2. (15%) A computer system provides protection using the Biba model. How would a virus spread in the following two scenarios?**

**(a) The virus were placed at system low (the compartment being dominated by all other compartments).**

In a Biba model, its essentially opposite that of the Bell-LaPadula. If a virus were placed in system low, it would spread by replicating itself, but it would only be able to replicate itself to everything in the low compartment. Anything higher than the low compartment would not be infected.

**(b) The virus were placed at system high (the compartment dominating all other compartments).**

Like the Bell-LaPadula has “write-up”, the Biba model has “write down”. That is, if the virus is placed in system high, it will spread by replicating itself to any compartment in the system.

**3. (10%) Convert the following script to a normal html without the javascript script. Copy and paste the converted html in your submission.**

**<script>**

**document.write(unescape('%3C%68%74%6D%6C%3E%0A%3C%74%69%74%6C%65%3E%45%6E%63%72%79%70%74%65%64%20%48%54%4D%4C%3C%2F%74%69%74%6C%65%3E%0A%3C%62%6F%64%79%3E%48%65%6C%6C%6F%20%77%6F%72%6C%64%21%3C%2F%62%6F%64%79%3E%0A%3C%2F%68%74%6D%6C%3E'));**

**</script>**

<html>

<title>Encrypted HTML</title>

<body>Hello world!</body>

</html>

**4. (20%) The following line of java code have the SQL injection vulnerability, where id is an input.**

**String query = "select name from students where id = " + id + ";";**

1. **Show an exploitation that always makes a true condition of the query.**

An exploitation that would always make the condition true would be to make id = “1 or 1=1”.

1. **If id is of type integer, discuss two security methods to prevent the injection.**

One security method would be to sanitize the inputs. Essentially limit the type of input and run checks to make sure that the input asked for was the type of input given.

Second method would be to parameterize SQL statement. This would always for test cases on the parameters to ensure proper input.

1. **If id is of type char, the query string will be the following. Show an exploitation that always makes a true condition of the query.**

If you make the id variable “a or a=a” this would make the condition always true.

**String query = "select name from students where id = '" + id + "';";**

**5. (10%) DoS attacks could target computing capacity by exhausting processes and CPU time. In Linux, we can use the command "ulimit" to stop this kind of DoS attacks. Show and discuss the two options of "ulimit" that are needed to stop the attack by limiting processes and CPU time?**

Ulimit -u (a number)

This will limit processes to a single user. This will not allow an attack to exhaust the system by giving it too many processes.

Ulimit -t (A number)

This command stop any single process from using too much CPU time. If an attack was design to exhaust your system by taking up all of your CPU time, you could give it a maximum amount of time any process was allowed to run.

**6. (10%) File checksum is often used to check if a file was maliciously modified.**

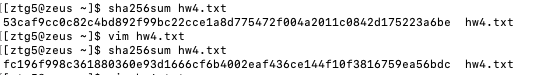
**(a) In Linux, what is the command to get a sha256 checksum of a file?**

Sha256sum /directory/file

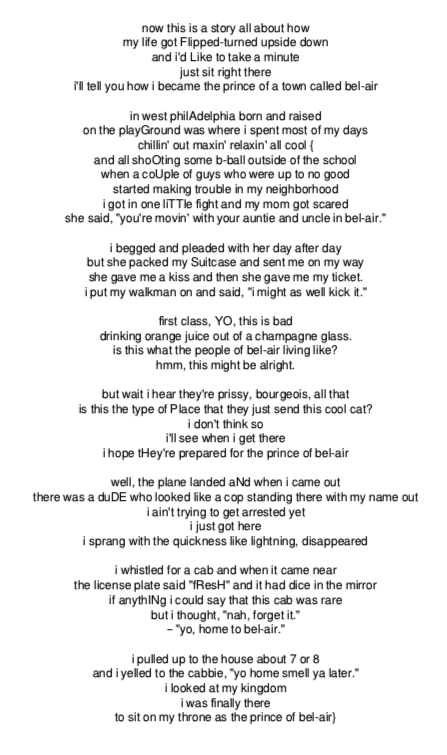
**(b) What is the sha256 checksum of "hw4.txt"?**

****

**(c) Add a space to any where in "hw4.txt", and recompute and show the new sha256 checksum of "hw4.txt"**

****

**7. (10%) Run the CTF virtual box, read the partial solution of "Is it really just a picture". Show the flag.**

****

**8. (10%) Run the CTF virtual box, read the partial solution of "PHP overwrite". Show the flag.**

FLAG{maybe\_i\_shouldn't\_have\_extracted\_everything\_huh}

****