MUYSHONDT, ALEXANDER D
4/9/19

HOMEWORK 4

Due on 4/9, submit in HARDCOPY

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).

Computer viruses spread by writing to other computers. The Bell-LaPadula model allows writing to any compartment that dominates the compartment containing the process. As all the other compartments dominate the infected compartment, the virus may write up to higher compartments infecting the rest of the system.

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

The Bell-LaPadula model forbids writing down from higher level dominant compartments, therefore the virus would not spread. However, it may still infect objects within the same compartment.

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).

The virus would not be able to spread further through the system because at low levels, writing is not available due to having less integrity. May still infect objects within its own container.

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

The virus may spread through the rest of the system as higher levels are regarded as having more integrity and allowed to write down to lower level containers.

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%3
C%2F%62%6F%64%79%3E%0A%3C%2F%68%74%6D%6C%3E'));
</script>
```

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 +
";";
```

(a) Show an exploitation that always makes a true condition of the query.

Exploit: id = ” or 1=1 #

(b) If id is of type integer, discuss two security methods to prevent the injection.

1. Sanitize the data entered by the user to prevent abusing a query
2. Use a parameterized query where the language handles the input to prevent abuse

(c) 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.

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

Exploit: id = ’ or TRUE #

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?

1)ulimit -u :-

By this command ; we can limit the no. of processes that an individual can run.

Syntax:- ulimit -Su/-Hu number_of_users

where Su stands for soft limit for no_of_users
Hu stands for hard limit for no_of_users

Here, soft limit sets the limit for maximum spawned processes by non-root users, and this limit can be modified by the non-privileged user.Whereas hard limits sets the maximum num of processes for programs running and only privileged user root can impose changes to that

2)ulimit -t:-

By this command ; we can limit the user processes sucht that they are unable to use more than maximum number of CPU time

Syntax:- ulimit -St/-Ht seconds

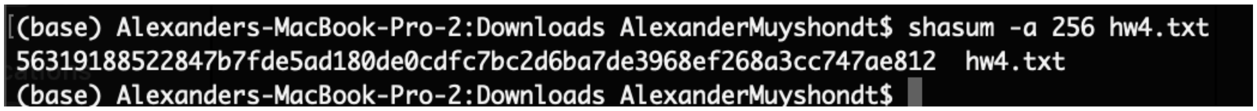
where   St stands for soft limit
   Ht stands for hard limit

This commands actually limits the CPU time a process can use. So,whenever a process exceeds the soft CPU time limit it will be sent a XCPU signal (and process is terminated if the signal isn't caught).When it exceeds the hard limit it will be send a kill signal

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?

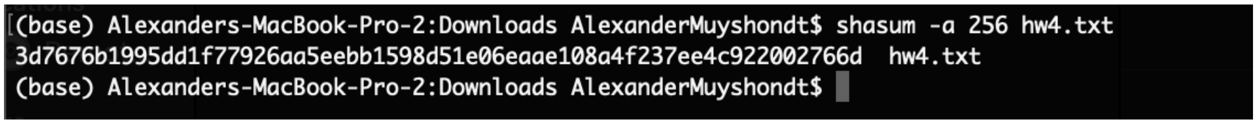
```
$ shasum –a 256 /path/to/file
```

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



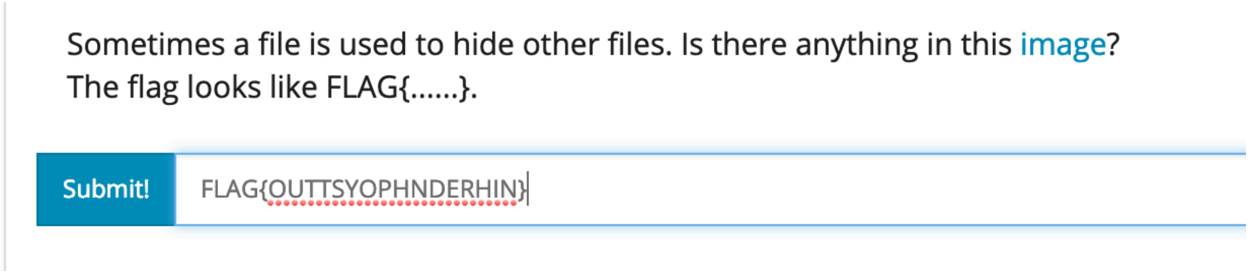
```
(base) Alexanders-MacBook-Pro-2:Downloads AlexanderMuyshondt$ shasum -a 256 hw4.txt
56319188522847b7fde5ad180de0cdfc7bc2d6ba7de3968ef268a3cc747ae812  hw4.txt
(base) Alexanders-MacBook-Pro-2:Downloads AlexanderMuyshondt$
```

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



```
(base) Alexanders-MacBook-Pro-2:Downloads AlexanderMuyshondt$ shasum -a 256 hw4.txt
3d7676b1995dd1f77926aa5eebb1598d51e06eaae108a4f237ee4c922002766d  hw4.txt
(base) Alexanders-MacBook-Pro-2:Downloads AlexanderMuyshondt$
```

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

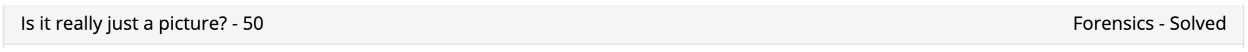


Sometimes a file is used to hide other files. Is there anything in this image?
The flag looks like FLAG{......}.

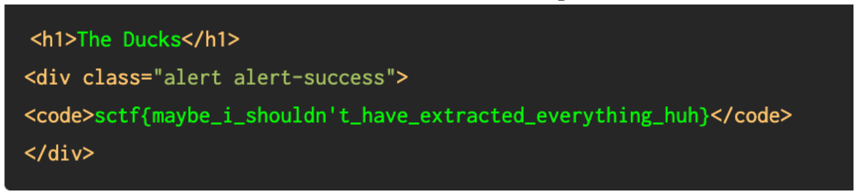
FLAG{OUTTSYOPHNDERHIN}

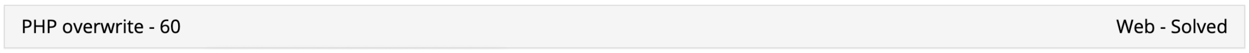
MUYSHONDT, ALEXANDER D
4/9/19

HOMEWORK 4


Is it really just a picture? - 50 Forensics - Solved

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


```
<h1>The Ducks</h1>
<div class="alert alert-success">
<code>sctf{maybe_i_shouldn't_have_extracted_everything_huh}</code>
</div>
```


PHP overwrite - 60 Web - Solved