Web Security and Malware Analysis

Assignment 1 – 21/11/2019

Goal: Solve each task by providing a brief explanation of the solution that you adopted. You must use BURP to solve the tasks (when required). Explanations should mainly include screenshots and some brief comments (except for task 3, where you have to provide a more detailed discussion).

Deadline: Dec. 2nd,2019

What to do: Send the report in a PDF format to davide.maiorca@unica.it with the subject "[WEBSEC] - Report For Assignment 1 – NAME SURNAME" and name the file "websec_report_1_name_surname.pdf". Remember to include your name, surname and matriculation number in your report!

Task 1 - Spidering

Perform a web spidering of this web application:

http://demo.testfire.net/

You can also use these credentials to log in as user:

User: jsmith

Password: Demo1234

Report all the useful information that you can find, such as:

- Pages found automatically (you must use BURP)
- Pages not found automatically (if any)

Hints: Try to see what happens if you use BURP 1.7 and if you use BURP 2.1. Some functionalities are different…

Task 2 – Basic Web Hacking

Complete the levels of the Natas wargame until Natas 8 (included) by using BURP to manipulate requests (when needed). By solving every level, you obtain a password to proceed to the next level.

The wargame starts here:
http://natas0.natas.labs.overthewire.org/

User: natas0

Pass: natas0

Every level gives you the password for the next level.

To access the next level, just go to the corresponding URL. For example, for natas1:

http://natas1.natas.labs.overthewire.org/

User: natas1

Pass: PASSWORD_FOUND_IN_NATAS0

You have to briefly describe the solution for each level (some screenshots + brief description of the solution is enough)

Hints that may be useful:

Passwords are hidden in the webserver under the folder /etc/natas_webpass/natas_level (e.g. password for natas1 is stored in /etc/natas_webpass/natas1 - this may be useful for some levels)

Task 3 – PHP Analysis

Consider the following PHP code:

```
if ($_COOKIE[MyPhPAdmin] == '') {
if (!$_POST[login] == 'login') {
die("Please Login:<BR><form method=post><input   type=password
name=password><input type=hidden value=login name=login><input
type=submit></form>");
} elseif($_POST[password] == $bcadminpass) {
setcookie("MyPhPAdmin","LOGGEDIN", time() + 60 * 60);
header("Location: admin.php"); } else { die("Incorrect"); }
}
```

Briefly provide a brief answer to the following questions:

1) What does the code do? When is the user authenticated and what happens after authentication?
2) Imagine a user contacts a server containing this code from a page. What kind of request should he do and what kind of client technology should he use?
3) What kind of security issue does this cookie have? Is it possible to bypass this by manipulating a request? And how?