LAB15B: PENTESTING

[Simon X. Camilo. Cybersecurity Student]

[Redacted]

Introduction

This document tries to find and detail every vulnerability found in the computers in the network, and offers solutions to these problems. An administrator computer is going to be used to find and fix these vulnerabilities, using only the tools that are available in this computer.

# Analysis

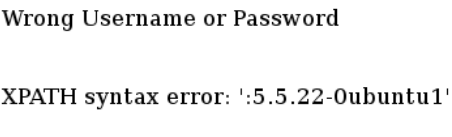
## Vulnerabilities

* The website is vulnerable to xss and sqli attacks
* The network is vulnerable to Ping of Death DoS attacks
* The network is allowing users outside of the lan to access the secure shell.
* The network is not collecting logs on suspicious activity

### Vulnerable to XSS & SQLI attacks

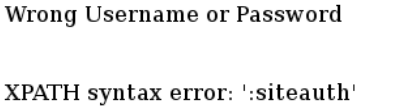
The website “urbank” doesn’t have any protections against XSS and SQLI attacks. This allow attacks to do all sorts of things, such as:

Image 1: Find system information

  
**Image 1**

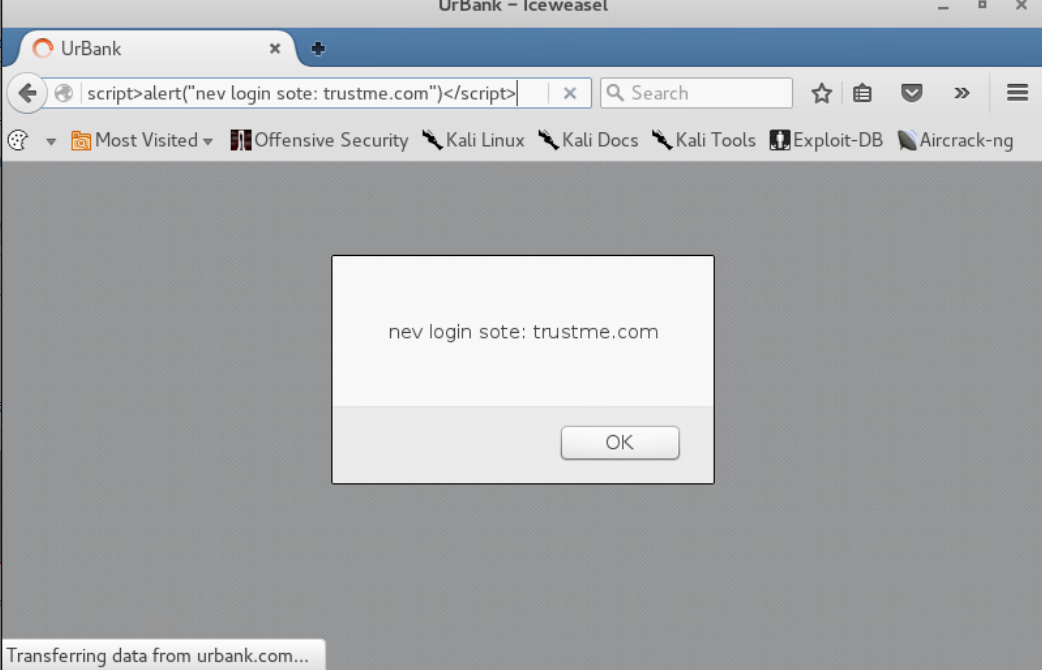
*Exposed system info*

Image 2: Find information about the database

  
**Image 2**

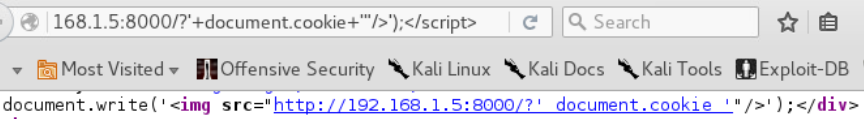
*Exposed database name*

Image 3: Cause an alert and make changes on a website

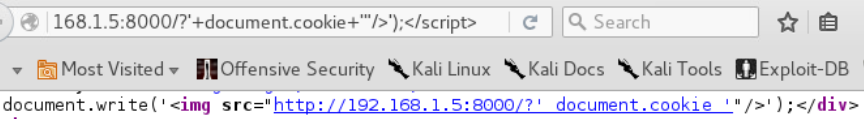
  
**Image 3**

*Social engineering exploit*

Image 4-5: Track session cookies and send them to a remote server

  
**Image 4**

*Malicious code inserted into the site’s html*

  
**Image 5**Extracted s*ession cookie*

The solution to these attacks is similar, . Below is the code that was written to prevent every attack of this kind.

preg\_replace('/|<|>|\^|\{|\}|\(|\)|\[|\]|\,|\\*|=|window.close[\(\)]|%/’, '', html\_entity\_decode($s, ENT\_QUOTES));

This code is very strict and will be used in url and login fields. It removes every character that is not supposed to be there from these fields, which is an essential procedure to prevent these kind of attacks. This code needs to be copied in the following files.

Every page and file on the website can be seen with the command dir /var/www/WebServer/. Image 6 shows every page located on the website



**Image 6**

*The contents /of the var/www/WebServer directory*

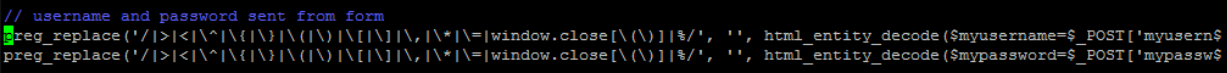
/var/www/WebServer/index.php line 15 as shown in image 7



**Image 7**

*Code that prevents XSS and SQLI attacks*

/var/www/WebServer/checklogin.html line 16 and 17 as shown in image 8



**Image 8**

*Code preventing attackers from executing attacks by using the username and password fields*

A different function needs to be used for information that is posted on the server, such as posts and messages. This is to provide a user friendly experience, allowing them to use special characters or be able to talk about code, without running it nor affecting the site. The following code is going to be used

htmlspecialchars($s, ENT\_QUOTES | ENT\_SUBSTITUTE);

htmlspecialchars will translate every special character into a html entity, which will allow users to input special characters without executing scripts. This code needs to be copied in the following files

/var/www/WebServer/login\_success.php line 20 and 21 as shown in image 9

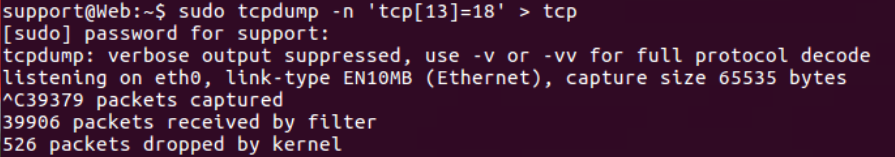


**Image 9**

*Code used to protect name and comment fields.*

### The network is vulnerable to Ping of Death DoS attacks

Attackers are also able perform DoS attacks that can reduce the availability of the network as shown on image 10. These attacks could make the website run slower for both users and employees, which can hinder performance, can lead to a worse user experience, and it also reduces profits since the server is down. These attacks are very lethal

  
**Image 10**

*Device receiving thousands of icmp DoS packets*

To prevent DoS, a tool called “Snort” is needed. Snort is a powerful network intrusion detection and prevention system that can be used to block these DoS attacks. A icmp filter can be created with snort, so attackers won’t be successful when performing a Ping of Death attack but will still allow users to ping the server.

“block icmp any any → any any \ (msg:”Potential Ping of Death attack”; sid: 10000054; dsize: >80;)”

### dsize option will make it so only icmp packets that are larger than 80 bytes are going to be filtered. ICMP packets are usually 74 bytes more or less, making 80 a safe limit for ICMP packets.

### The network is not collecting logs on suspicious activity

The ssh is accessible to anyone in the network, which is dangerous because no one outside of the network needs access to the server, and this allows unauthorized users to easily access the server if they are already aware of the credentials for the server.

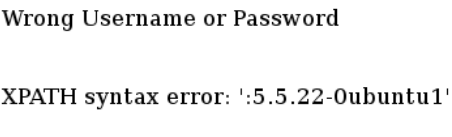
The following rule can be written to prevent unauthorized users from accessing the ssh

drop tcp any any → any any \

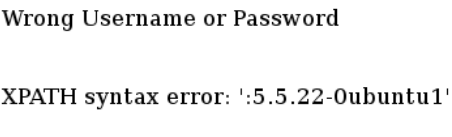
(content:"|70 70 6f 72 74 40 53 54|"; \

msg:"Remote Shell"; sid:1000111;)

After that, it must be confirmed that the rule works, by trying to connect to the server shell in the same way an attacker would. For this to happen, a listening port should be opened on port 8, and access must be given through this port, as shown on image 11. if the attack fails, it means that the rule was successful, and snort would had reported this on their logs. The logs can be accessed with the command “cat tail /ids\_dmz.log” and it would look like image 12.

  
**Image 11**

*Connected to port 8?*

  
**Image 12**

*Snort logs found in ids\_dmz.log*

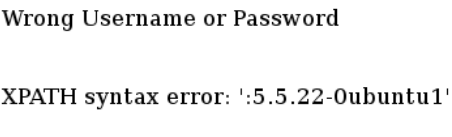
### The ids (snort) is not logging any information about the server

The server is also not using any kind of network protection or logging. System administrators are not alerted of suspicious activity and attackers are able to infiltrate into other’s computers by having them execute an infected program.

The following filter will make sure suspicious activity is filtered in Snort, the administrators will be reported every time an attempt like this is made

alert tcp any any → any any (msg:”Web injection”; flow:to\_server,established; pcre:”/(%3C)|(<)|(%3E)|(>)|(%27)|(‘)/i”; sid:1000042;)

Every time a web injection attack is detected, the web administrator will be alerted, the alert will look like image 13

  
**Image 1**

*Reported web injection attempts from snort*

# Conclusion

It’s heavily recommended for the system administrator to review these vulnerabilities and fix them as soon as possible, because these are vulnerabilities that an attacker with little knowledge could easily exploit. Fixing these vulnerabilities is critical to ensure the security of the employees and the users who are involved with this service. Further pentesting will be required when further development is made in the website.

References

PHP preg\_replace() Function. (n.d.). Retrieved February 26, 2021, from <https://www.w3schools.com/php/func_regex_preg_replace.asp>

Campbell, H. (2011, March 28). How to remove single and double quotes from a string. Retrieved March 06, 2021, from <https://stackoverflow.com/questions/5454273/how-to-remove-single-and-double-quotes-from-a-string>

Htmlspecialchars. (2001, March 6). Retrieved March 06, 2021, from <https://www.php.net/htmlspecialchars>

https://owasp.org/www-community/attacks/xss/

[https://cheatsheetseries.owasp.org/cheatsheets/Cross\_Site\_Scripting\_Prevention\_Cheat\_Sheet.html#Why\_Can.27t\_I\_Just\_HTML\_Entity\_Encode\_Untrusted\_Data.3F](https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html" \l "Why_Can.27t_I_Just_HTML_Entity_Encode_Untrusted_Data.3F)

https://stackoverflow.com/questions/2159724/how-can-escaping-be-used-to-prevent-xss-attacks

https://www.wordfence.com/learn/how-to-prevent-cross-site-scripting-attacks/