UC-318

Cybersecurity Website Hardening Project

Blair Greco, Sean Lewis, Rontavious Heard, AJ Moser
Advisor: Professor Donald Privitera

KENNESAW STATE UNIVERSITY
COLLEGE OF COMPUTING AND SOFTWARE ENGINEERING

Abstract

The project we were given by Professor Privitera is to secure a web server that is a simulation of a genuine business with our case being a restaurant known as Akwaaba. The business website is hosted on Apache, MariaDB, Red Hat Linux, and PHP. We will first need to explore the network we were given to determine the system's weaknesses to evaluate the risks and create a proper security policies plan with the help of the National Institute of Standards and Technology (NIST) standards. Following the security plan we created, our group will implement these changes into the network, hardening them to these standards. Finally, we will participate in a red/blue team cybersecurity ethical hacking procedure with our network and the two other teams. We will use white-hat-hacker skills to gain access to other groups' networks while protecting our own network by patching up weaknesses if they have been breached.

Project Goals

- A Well-structured Security Policy Plan
- A Secure Business Network
- Successful Red vs Blue Procedure

Honors Project Goals

- Firewall Defense Implementation
- Password Cracking Execution

The Three Main Phases of Website Protection

Phase 1: Planning

This phase of the project was laying down the ground for what we planned to implement on the network and website we were presented. We created policies on how we would like to deal with passwords and usernames as well as access management hierarchy. A Risk Assessment was also done on the main 4 parts of our system Apache, Red Hat, WordPress, and MariaDB. An investigation into some tools that we used in the capstone were also done in this phase with key tools being Word Fence & Jetpack.

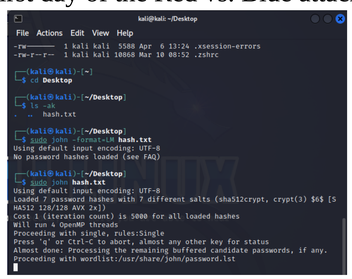
Vulnerability	Asset	Threat	Likelihood	Impact	Risk rating factor
a bug in mod_ssl when using per-location client certificate verification with TLSv1.3	Apache	allowed a client to bypass configured access control restrictions.	3	3	9
A carefully crafted SessionHeader sent by HTTP/2y	Apache	Bypass validation and be forwaded by mod_proxy	4	3	12
WP does not escape error message in RSS widget, could lead to xss	WordPress	Cross-site scripting(XSS)	1	3	3
Unauthenticated Blind SSRF via DNS rebinding	WordPress	SSRF Injection	2	3	6
The vulnerability exists due to a segmentation via the component Exec_time_tracker::get_loops/filesort_tracker::report_use/filersort	MariaDB	SQL Injection Leading to DOS	3	2	6
Use-after-free errir int the convert_const_to_int() function whe processing BIGNIT data type	MariaDB	DOS	3	2	6

Phase 2: Implantation

During this phase, we focused on making the website the most secure we could. We started by trying to implement everything we stated we would do in phase one and fixing some of our vulnerabilities from our risk rating table. To begin with we changed all passwords, then tried updating all our servers and Red Hat. We successfully updated Red Hat and WordPress to their newest versions but failed at updating MariaDB and Apache. We couldn't figure out how to fix the errors we were getting in the updating process, so we decided to disable the modules and unnecessary add-ons that proved to be not secure. Then installed FirewallD and closed all unnecessary ports.

Phase 3: Attack & Defense

The final phase of the capstone with us focusing on protecting our network while hacking into the other team's network. On the defense side, we were able to defend effectively from the other team with no evidence of tampering on our network or website. We found major success when we were on the offense and able to access our target network and website on the first day of the Red vs. Blue attack. We gain access to a lot of information, especially the password hash file.



Conclusions

Our team faced various challenges while updating and securing our website, but we were able to overcome them through perseverance and problem-solving. We updated Red Hat Linux by creating a new account and connecting it to our webserver and used WP-CLI to update our WordPress and plugins. We also added new plugins like Jetpack and Word Fence and implemented FirewallD to enhance our website's security.

Despite facing difficulties with updating Apache and MariaDB to their newest versions, we were able to harden our security by removing unnecessary and unsecure functions and modules.

Our efforts paid off as our website remained unbreeched throughout the red vs blue event, while we were able to successfully breach the opposing team's website on multiple occasions and decrypt some of their passwords using the tool Jack The Ripper.

Acknowledgments

- Professor Samuel Wolde
- Professor William Forsyth
- Professor Donald Privitera
- Professor William Haggerty
- Professor Darrin Morrow

Team Website