Akwaaba Cybersecurity Milestone 3 Report

W02-12777 Project 11 Group 2

April 17th, 2022

Team Lead:	Brandon Turner
Technical Specialist:	Kowou Agbavon
Technical Specialist:	Andres Morales
Technical Specialist:	Felton Strickland
Technical Writer:	Aleksandar Punos

Project Status Update

After phase two we moved to the attack phase of the project. We performed vulnerability analysis and penetration testing on other team’s servers. We recorded our attempts and added them to a document for documentation. At the same time, we made sure our servers were not successfully attacked by another team. We continued to meet three times a week to update each other on our progress and work together to think of our next steps in the attack of other servers and defense of our own. The results of our labors are documented in further detail in another section of this report.

Scope of Project

This phase of the project included research into vulnerability analysis and penetration testing. This research was used to find methods to perform attacks on the opposing team to gain access to their server. We performed a vulnerability analysis on their server. We found some vulnerabilities and attempted to exploit them. For penetration testing, we attempted to use social engineering methods. We used phishing to acquire credentials of the opposing server. In this phase we were also charged with defense of our own server. We kept our server up during this duration.

Executive Summary

In this project IT-cybersecurity team had assignment to defend the Akwaaba (Apache) server and to make the entire system (Maria Database and VM Virtual Machine) more secure. In the first part of the project our team created an internet security policy for Akwaaba. This was crucial because this represents the foundation of a good security system. All other parts of the project were implemented successfully because of our well-defined security policy. For every part of our project, we created a Milestone report, Power Point slides and Weekly Log report and in them explained the project progress in detail.

Second part of this project was about research and finding the most effective way to protect entire Akwaaba system. As a result of this we did an excellent job with securing the server and this will be tested in the next part of the project by the blue team. The main part of the project segment was implementation of security programs on our Apache server. We have implemented some changes to Maria Database and VM to make them more secure as a system. In this segment we created a project website and updated the website with accomplishments from project’s Milestone 1.

The next segment of the project includes research of vulnerability tools that our team could use to get into another team server and putting vulnerability testing into action. The other part of this section was documenting the findings from another team attack on our system. At the same time, we kept our project website up to date with project progress.

The last part of the report will be the final report (comprehensive recorder presentation) that will have all parts summarized and presented to an IT Department team for the final evaluation of our project. Here we will introduce our team and the project- objectives, backgrounds and relative technologies and concept that we were using. One big part of this project segment would be our team experience and project management experience as well.

Research

During this project we had to do a lot of research to fully understand and properly implement every phase. We received information from various sources including our project sponsor who provided the teams with documentation on how to set up the VM that we needed to secure, and the steps to take to properly control the flow of the project. Throughout the project we were also provided with documents to help us create the information security policy, risk assessment, vulnerability analysis, and penetration testing from the sponsor. We also had to do a lot of research on our own to learn how to properly secure our server and how to take advantage of the opposing server's vulnerabilities. We did this with the work of many internet articles which can be found in our references section at the end of this paper. With the use of sites like the MariaDB document site, Kali website to help us learn and use Kali linux and more, we were able to secure our server and form a good base of attack for our red-team planning.

Methodology

The project was broken up into various phases in order, so we decided to take a qualitive approach and take each phase of the project in chronological order. The team decided to take this approach mainly because the project and the required deliverables are built out this way, but also because each stage builds on the last and helps us gain greater understanding that closely follows the waterfall model. The project is built around creating a security plan and implementing security measurements for a small restaurant named Akwaaba with the knowledge that another team will try to break-into our server, while we also try to break into their server.

Phase 1 of the project consisted of creating an information security policy commonly known as an ISP. To do this the team relied upon the documents that were provided to us by our project sponsor Dr. Li, consisting of a word document and PowerPoint that explained what an ISP is, why it is needed, and examples of how to create one. The ISP is a policy created for employees, including ownership and management to follow to produce safe security practices. We also made use of the documents provided by the project sponsor to create a risk assessment so we could plan and prepare for how we wanted to implement our security. The best way to secure something is to understand where it is the most vulnerable. After determining what assets were the most vulnerable the team brainstormed and created a technical plan to secure three vital areas: the VM (Virtual Machine) that was provided to us, an Apache Server, and a MariaDB.

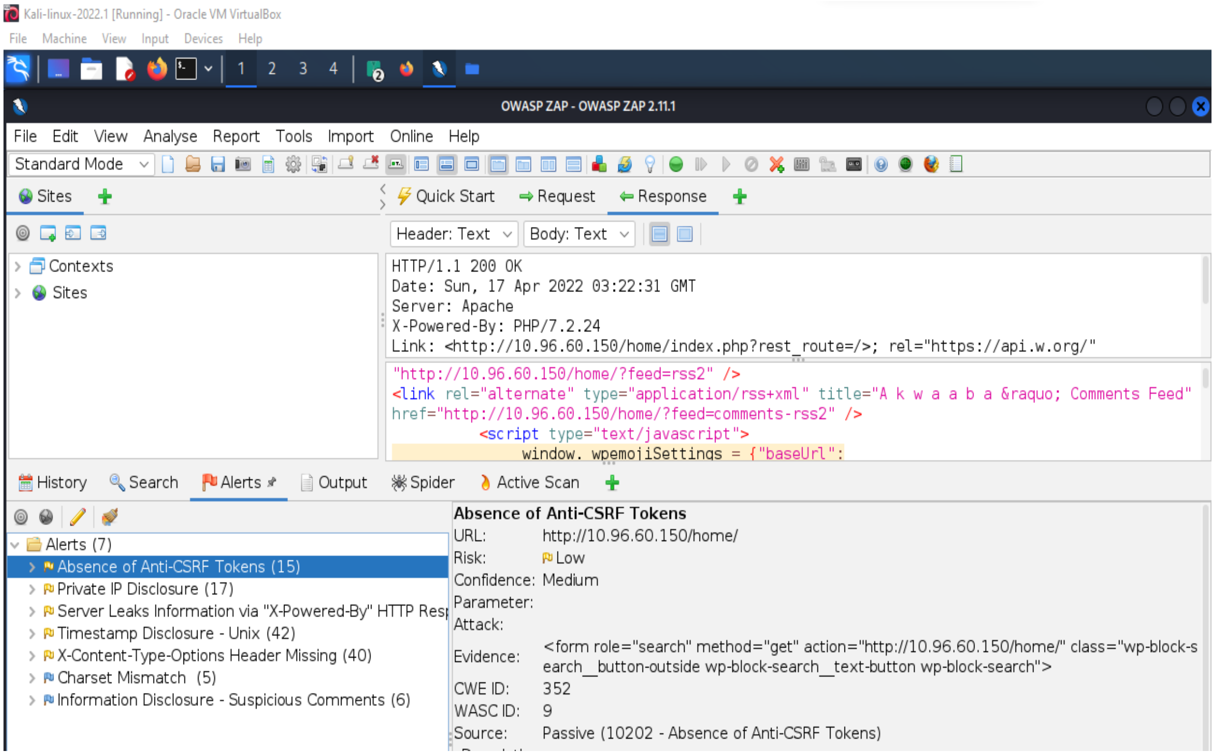
Phase 2 of the project focused on further research and implementation of a security plan. We decided the best way to do this would be to split up the work by having separate team members focus on different assets. However, at this stage is where obstacles begin to arise. A few things that the team wanted to accomplish such as encrypting the MariaDB and the data in the DB as well as updating the VM to implement a proper firewall caused the VM to crash due to instability. With these limitations in mind, we adjusted and still implement good security measurements. The sites that were used to research what security measurements to implement and how to implement them are listed in the references section. In this phase we also planned on which tools could be used for vulnerability analysis and penetration testing for the next stage and tested these tools on our own server to ensure that the server is no longer as vulnerable.

The final phase of the project was running a red-team/blue-team exercise where our group attempted to obtain access to an opposing group's server while also trying to protect and watch our server. We decided that a great tool to use would be Kali Linux as it provided a whole suite of tools that can be used to try to gain access to the server, it also had tools that could be used for social engineering. We will go more into detail on the plan and the results in a subsequent portion of this paper. With the knowledge and results that we have obtained from this project it made each member of the team more adjusted in the world of cybersecurity which we can carry into future careers and everyday life. The team has also submitted the project to C-Day in hopes to present and spread what we have learned to others as well.

Analysis Results - Kowou

The testing of the web application using the open-source tool OWASP-ZAP reveals information from unauthorized access of the hostname Amazon EC2 has been in the HTTP response body. In the same way, the IP address 10.96.60.123 has been disclosed. The information may be helpful to the attacker to exploit the system. The web application server is leaking information via the HTTP response header field, which exposes the frameworks component of the web application and can facilitate the attacker's access to credential information. Moreover, in the absence of application functionality, anti-CSRF Tokens, uses to control URL forms from repeating, and will allow an unauthorized individual to send an HTTP request to the victim without their knowledge. Finally, the X-Content-Options header missing will allow the internet browser to perform

sniffing pages from their actual content affected by injection.


Kali-linux-2022.1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
OWASP ZAP - OWASP ZAP 2.11.1
File Edit View Analyse Report Tools Import Online Help
Standard Mode
Sites
Quick Start
Request
Response
Header: Text
Body: Text
Contexts
Sites
History
Search
Alerts
Output
Spider
Active Scan

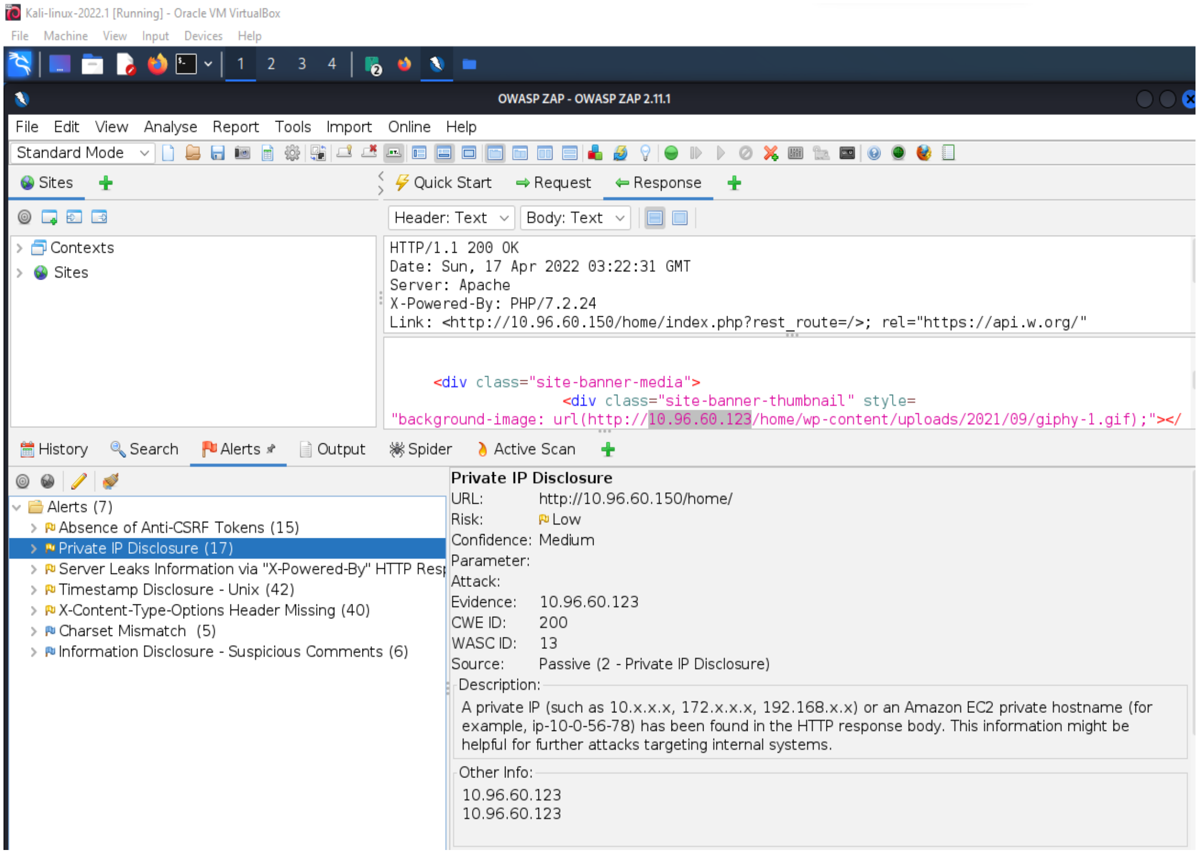
```
HTTP/1.1 200 OK
Date: Sun, 17 Apr 2022 03:22:31 GMT
Server: Apache
X-Powered-By: PHP/7.2.24
Link: <http://10.96.60.150/home/index.php?rest_route=/>; rel="https://api.w.org/"
```

```
"http://10.96.60.150/home/?feed=rss2" />
<link rel="alternate" type="application/rss+xml" title="A k w a a b a &raquo; Comments Feed"
href="http://10.96.60.150/home/?feed=comments-rss2" />
        <script type="text/javascript">
            window._wpemojiSettings = {"baseUrl":
```

- Alerts (7)
  - Absence of Anti-CSRF Tokens (15)
  - Private IP Disclosure (17)
  - Server Leaks Information via "X-Powered-By" HTTP Resp
  - Timestamp Disclosure - Unix (42)
  - X-Content-Type-Options Header Missing (40)
  - Charset Mismatch (5)
  - Information Disclosure - Suspicious Comments (6)

Absence of Anti-CSRF Tokens

URL:	http://10.96.60.150/home/
Risk:	Low
Confidence:	Medium
Parameter:	
Attack:	
Evidence:	<form role="search" method="get" action="http://10.96.60.150/home/" class="wp-block-search__button-outside wp-block-search__text-button wp-block-search">
CWE ID:	352
WASC ID:	9
Source:	Passive (10202 - Absence of Anti-CSRF Tokens)


Kali-linux-2022.1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
OWASP ZAP - OWASP ZAP 2.11.1
File Edit View Analyse Report Tools Import Online Help
Standard Mode
Sites
Quick Start
Request
Response
Header: Text
Body: Text
Contexts
Sites
History
Search
Alerts
Output
Spider
Active Scan

```
HTTP/1.1 200 OK
Date: Sun, 17 Apr 2022 03:22:31 GMT
Server: Apache
X-Powered-By: PHP/7.2.24
Link: <http://10.96.60.150/home/index.php?rest_route=/>; rel="https://api.w.org/"
```

```
        <div class="site-banner-media">
                <div class="site-banner-thumbnail" style=
"background-image: url(http://10.96.60.123/home/wp-content/uploads/2021/09/giphy-1.gif);"></
```

- Alerts (7)
  - Absence of Anti-CSRF Tokens (15)
  - Private IP Disclosure (17)
  - Server Leaks Information via "X-Powered-By" HTTP Resp
  - Timestamp Disclosure - Unix (42)
  - X-Content-Type-Options Header Missing (40)
  - Charset Mismatch (5)
  - Information Disclosure - Suspicious Comments (6)

Private IP Disclosure

URL:	http://10.96.60.150/home/
Risk:	Low
Confidence:	Medium
Parameter:	
Attack:	
Evidence:	10.96.60.123
CWE ID:	200
WASC ID:	13
Source:	Passive (2 - Private IP Disclosure)

Description:

A private IP (such as 10.x.x.x, 172.x.x.x, 192.168.x.x) or an Amazon EC2 private hostname (for example, ip-10-0-56-78) has been found in the HTTP response body. This information might be helpful for further attacks targeting internal systems.

Other Info:

10.96.60.123
10.96.60.123

Infrastructure Defense Plan – Kowou

Defending the business from these threats requires a significant effort of organization and increasing deployment of security measures to harden the intrusion and prevent an attack. Our strategies will extend the defense in the depth of web application.

Updating software or components that are vulnerable to attack by patching the software at risk. To prevent the cross-site script injection into the web service, it is essential to correct the content-type header that sets the X-Content –Type to a no-sniff web page to ensure every user complies not to perform sniffing. Also, the web server and the load balancer will be reconfigured to suppress the HTTP response header to leak information. A validation user input can be added to the application never to allow an untrust user to bypass the authentication request to access the confidential information, such as the private IP address disclosed: 10.96.60.123. A set up of intrusion detection can monitor the logs and security events on the web application by alerting the security administrator of any security failures.

Mitigation Plan

Our mitigation plan consisted of applying and enforcing all the security controls within our security policy. These measures ranged from administrative, technical, and physical controls. Most notably were the technical/logical controls which consisted of installing a ModSecurity web application firewall (WAF) to protect our network from cross site scripting attacks (XSS) and SQL injections attacks. After conducting a vulnerability and risk assessment we concluded that our systems will more than likely be more susceptible to XSS and SQL injection attacks considering that our web application is public. To prevent other common attacks associated with web applications we also implemented HTTPS redirect to prevent users from accessing our site using HTTP, an unsecure protocol for web communication.

Other measures were also taken to prevent brute force attacks via SSH such as changing the root password and applying the password requirement policy. The new password will make it extremely difficult for a hacker to brute force the root account. Unused and unsecure ports were closed to decrease entry points into the network. Additionally, administrative accounts were also established for each administrator and administrators were prohibited from using the root account. This helped established accounting, authorization, and auditing for our network. By doing these we are now able to see exactly who logged into the server and the activity they performed while being logged in.

Policies

The security policy consisted of several sections which included data classification, access control, security awareness and behavior, and a technical plan based on a security evaluation of the Akwaaba network. Our team of administrators realized that Akwaaba will be storing public personally identifiable information (PII) that will belong to our customers. Therefore, we established a data classification policy that enforced stricter rules and regulations regarding the systems that stored PII. We considered these systems as being “Highly Security Systems” and required these systems to be labeled as such.

We also implemented a role-based access control policy in which administrators were the only personnel allowed to access the servers and the server rooms. Additionally, administrators are to access the server remotely using their respective account. All administrators are prohibited from accessing the root account. Additionally physical controls were applied to secure the server rooms such as security cameras, access log records, and RFID badges for access.

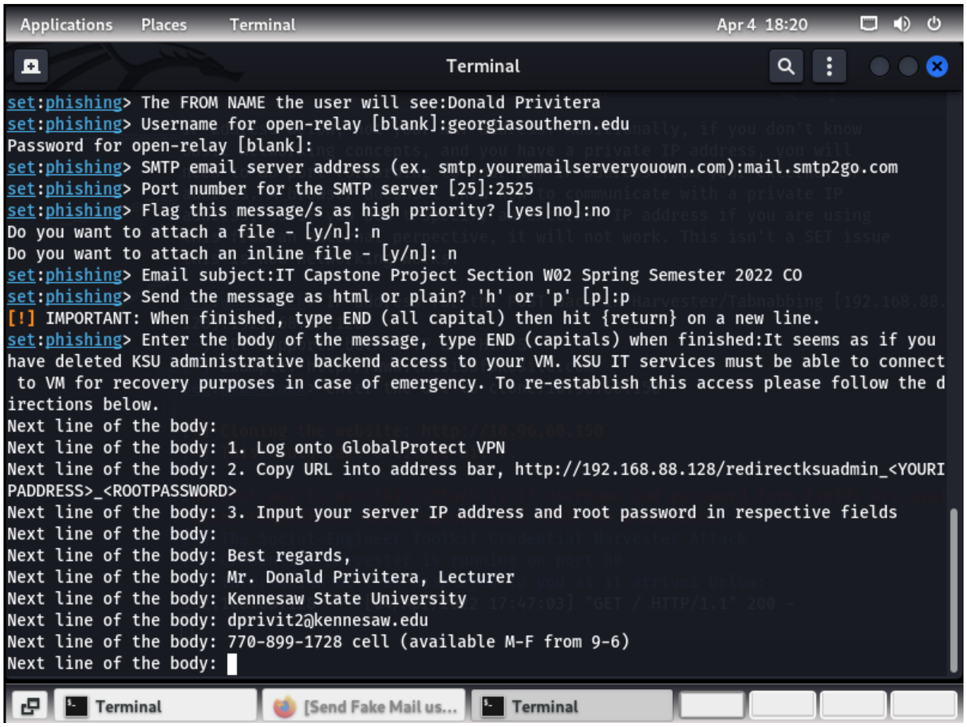
Because the security effort is one that should include everyone within the business, we decided to add a security awareness section to the security policy as well. This policy outlines the expectations of the company regarding the manner in which employees use information technology resources. The policy also requires all employees to complete a security awareness training course once every six months. The Akwaaba security policy should also be available on the company website so that employees can access and review it at any given moment.

Red-team Plan/Results

Our red team plan consisted of two phases vulnerability analysis and penetration testing. Our initial efforts to conduct vulnerability analysis consisted of using a series of passive/active foot printing techniques to acquire as much information as possible. These techniques included finding the operating system and kernel information of the opposing team’s server. We also attempted to enumerate the network by finding open ports using Netstat, both attempts were successful. Vulnerabilities were also found on the website regarding the site’s means of communicating web traffic. We found that HTTPS was not being used after identifying the open ports via Netstat, the only open ports were associated with HTTP and SSH.

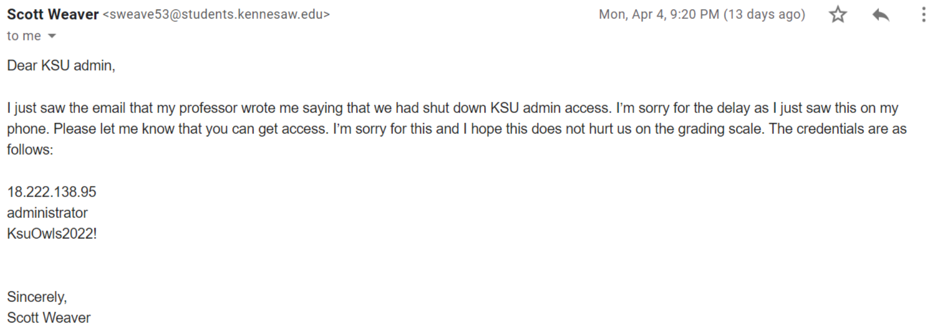
After finding the vulnerabilities mentioned earlier, we decided to move forward with exploiting these vulnerabilities, mainly the ones associated with the web application. A series of XXS and SQL injection attacks were performed in which they were all unsuccessful. From this we concluded that a WAF is probably being used to reject this suspicious traffic.

Lastly, after unsuccessful attempts to exploit the vulnerabilities associated with the web application, we attempted to use social engineering techniques were used to acquire the server credentials. Using Kali Linux’s social engineering toolkit (SET), we were able to acquire the credential to server associated with IP address 10.96.60.64 via phishing. The email was sent to all team leaders of the IT Capstone class. The idea behind the phishing attempt was to impersonate a trusted KSU administrator and trick the team members into giving us their credentials so that we may re-establish backend access for KSU IT administrators. Three emails were sent out to the respective team members. Two of the three team members responded. One team member responded with valid credentials for their server, the other team member responded with invalid credentials possibly signifying that they were aware of our plan. Screenshots of the attacks and results are listed below.



```
set:phishing> The FROM NAME the user will see:Donald Privitera
set:phishing> Username for open-relay [blank]:georgiasouthern.edu
Password for open-relay [blank]:
set:phishing> SMTP email server address (ex. smtp.youremailserveryouown.com):mail.smtp2go.com
set:phishing> Port number for the SMTP server [25]:2525
set:phishing> Flag this message/s as high priority? [yes|no]:no
Do you want to attach a file - [y/n]: n
Do you want to attach an inline file - [y/n]: n
set:phishing> Email subject:IT Capstone Project Section W02 Spring Semester 2022 CO
set:phishing> Send the message as html or plain? 'h' or 'p' [p]:p
[!] IMPORTANT: When finished, type END (all capital) then hit {return} on a new line.
set:phishing> Enter the body of the message, type END (capitals) when finished:It seems as if you
have deleted KSU administrative backend access to your VM. KSU IT services must be able to connect
 to VM for recovery purposes in case of emergency. To re-establish this access please follow the d
irections below.
Next line of the body:
Next line of the body: 1. Log onto GlobalProtect VPN
Next line of the body: 2. Copy URL into address bar, http://192.168.88.128/redirectksuadmin_<YOURI
PADDRESS>_<ROOTPASSWORD>
Next line of the body: 3. Input your server IP address and root password in respective fields
Next line of the body:
Next line of the body: Best regards,
Next line of the body: Mr. Donald Privitera, Lecturer
Next line of the body: Kennesaw State University
Next line of the body: dprivit2@kennesaw.edu
Next line of the body: 770-899-1728 cell (available M-F from 9-6)
Next line of the body: 
```

Explanation: This photo shows the use of Kali Linux social engineering toolkit (SET) to conduct the phishing attempts on the team leaders. Team leaders were later given the option to send credentials directly to KSU IT services via email if the proposed method listed above failed to work.



Scott Weaver <sweave53@students.kennesaw.edu>
to me

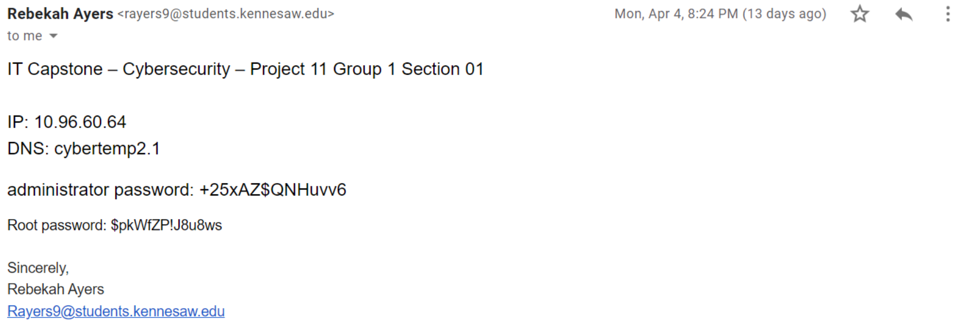
Mon, Apr 4, 9:20 PM (13 days ago)

Dear KSU admin,

I just saw the email that my professor wrote me saying that we had shut down KSU admin access. I'm sorry for the delay as I just saw this on my phone. Please let me know that you can get access. I'm sorry for this and I hope this does not hurt us on the grading scale. The credentials are as follows:

18.222.138.95
administrator
KsuOwls2022!

Sincerely,
Scott Weaver



Rebekah Ayers <rayers9@students.kennesaw.edu>
to me

Mon, Apr 4, 8:24 PM (13 days ago)

IT Capstone – Cybersecurity – Project 11 Group 1 Section 01

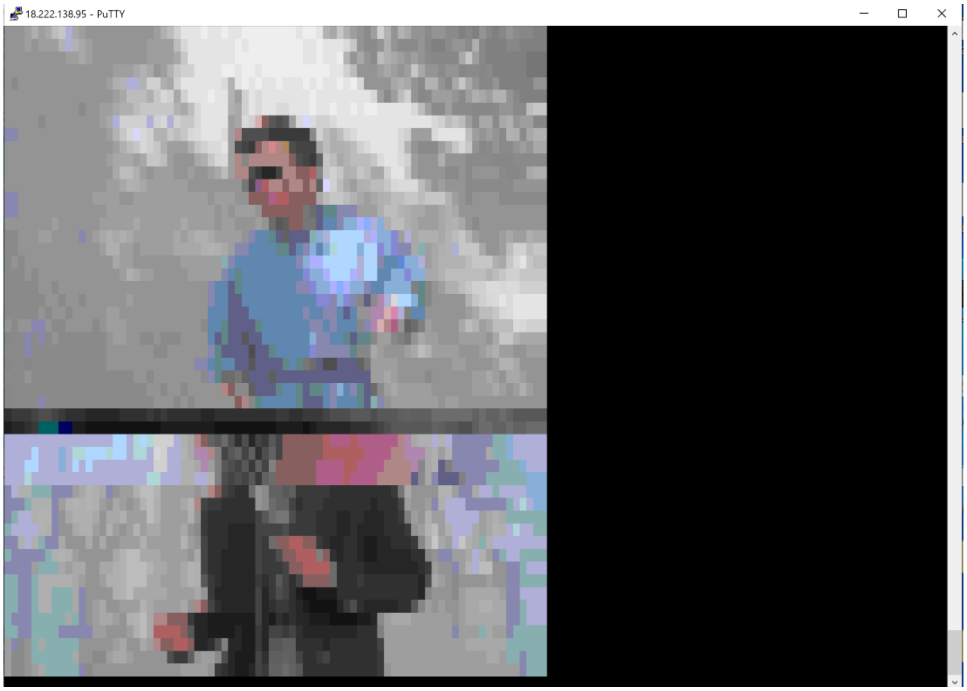
IP: 10.96.60.64
DNS: cybertemp2.1

administrator password: +25xAZ$QNHuvv6

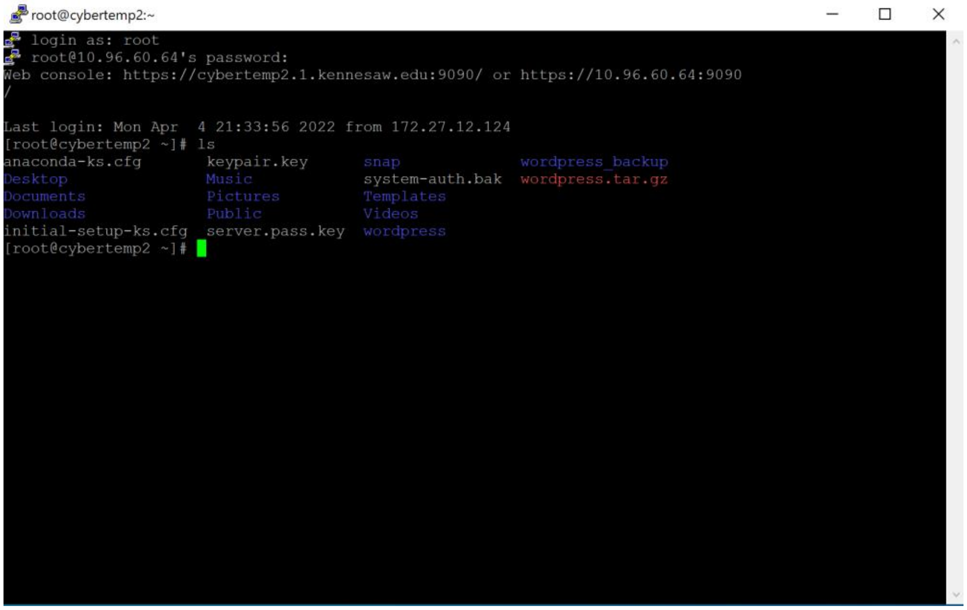
Root password: $pkWfZP!J8u8ws

Sincerely,
Rebekah Ayers
Rayers9@students.kennesaw.edu

Explanation: The two photos above show the responses from two of the three team leaders. Team leader Scott Weaver responded with fake credentials while Rebekah Ayers responded with valid credentials for her team's respective server.



Explanation: This server corresponds to the credentials sent by Scott Weaver. After logging in the server with the provided credentials, I was welcomed by a video of a dancing man pictured above. After discovering the suspicious IP address sent by Mr. Weaver the server was accessed via a sandbox environment.



```
login as: root
root@10.96.60.64's password:
Web console: https://cybertemp2.1.kennesaw.edu:9090/ or https://10.96.60.64:9090
/

Last login: Mon Apr  4 21:33:56 2022 from 172.27.12.124
[root@cybertemp2 ~]# ls
anaconda-ks.cfg       keypair.key      snap             wordpress_backup
Desktop               Music            system-auth.bak  wordpress.tar.gz
Documents             Pictures         Templates
Downloads             Public           Videos
initial-setup-ks.cfg  server.pass.key  wordpress
[root@cybertemp2 ~]# 
```

<u>Explanation:</u> Proof of access for server with IP address 10.96.60.64 with credentials sent by Rebekah Ayers.

<u>Project Documents</u>

Project documentation is the process of recording the key project details and producing the documents that are essential to implement it successfully. The main documents in this project are Milestone reports, Milestone PowerPoints slides reports, Weekly Logs reports. The other not less important documents are documenting that show are progress with defending our Akwaaba system –Apache server, Maria Database and VM-and documents that represents our vulnerability testing and trying to take over blue team server and their system.

Valuable document that incorporates all our project stages with time progress week by week is Gaant Chart. This document consists of our project deliverables, and current weekly progress status as well. All these documents are so essential for our Akwaaba project, and all together implemented successfully are very beneficial to our project.

References

Acunetix. (2021, March 12). *10 tips for Apache Security*. Acunetix. Retrieved February 19, 2022, from https://www.acunetix.com/blog/articles/10-tips-secure-apache-installation/

Clouder , A. (2019, July 15). *How to secure connections to mariadb with SSL encryption*. Alibaba Cloud Community. Retrieved February 19, 2022, from https://www.alibabacloud.com/blog/how-to-secure-connections-to-mariadb-with-ssl-encryption_595079

Managing User Via Command-Line Tools . Red Hat Customer Portal . (n.d.). Retrieved February 19, 2022, from https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/deployment_guide/s2-users-cl-tools

Sen, K. (2021, October 17). *Tripwire Open Source vs OSSEC: Is This Tripwire Alternative Right for You?* UpGuard . Retrieved February 19, 2022, from https://www.upguard.com/blog/tripwire-open-source-vs-ossec-which-is-right-for-you

What is an RSA key used for? Namecheap. (2019, September 17). Retrieved March 16, 2022, from https://www.namecheap.com/support/knowledgebase/article.aspx/798/67/what-is-an-rsa-key-used-for/

Shrivastava, T. (2016, November 14). *13 Apache Web Server Security and Hardening Tips*. Tecmint. Retrieved March 16, 2022, from https://www.tecmint.com/apache-security-tips/

Ubiq. (2020, November 20). *How to install mod_security on centos 7*. Ubiq BI. Retrieved March 16, 2022, from https://ubiq.co/tech-blog/how-to-install-mod_security-on-centos-7/

Ellingwood, J. (2013, July 23). *How to secure mysql and mariadb databases in a linux VPS*. DigitalOcean. Retrieved March 19, 2022, from https://www.digitalocean.com/community/tutorials/how-to-secure-mysql-and-mariadb-databases-in-a-linux-vps

Ksiazek, K. (2022, February 8). *Ten tips on how to achieve mysql and mariadb security*. Severalnines. Retrieved March 19, 2022, from https://severalnines.com/database-blog/ten-tips-how-achieve-mysql-and-mariadb-security

Mariadb administration. MariaDB KnowledgeBase. (n.d.). Retrieved March 19, 2022, from https://mariadb.com/kb/en/mariadb-administration/

Widenius, M., Axmark, D., & Arno, K. (n.d.). *MySQL Reference Manual*. O'Reilly Online Learning. Retrieved March 19, 2022, from https://www.oreilly.com/library/view/mysql-reference-manual/0596002653/ch04s02.html

Penetration testing and ethical hacking linux distribution. Kali Linux. (n.d.). Retrieved April 17, 2022, from https://www.kali.org/