NetSec - ECHO

Security Portfolio Report

FIT5120

Yash Mestry : 30123305

Akash Veerabomma : 29915880

Ruilin Tian: 27606813

Table Of Contents

Introduction

We are a team of three from the MNS background here to work on a security Portfolio for diagnosing, evaluating, assessing, and conducting a penetration test against the three victim systems assigned to us: http://mobihelper.ga, https://ayemate.gobest.site, https://greenskeepers.me. Our main aim in this portfolio will be:

1. Demonstrate the ability to react and adhere to industry standard policies and restrictions.
2. Investigate possible vulnerabilities and threats for a given application system.
3. Design, implement and produce test procedures and perform evaluation of system components.
4. Provide practical security policies and strategies to reinforce vulnerable systems.
5. Design, conduct and manage penetration testing to evaluate the security of live systems.
6. Record, investigate and reflect on findings to further develop vulnerability testing approach

Part 1 : System Diagnostics, Planning & Policies

A) Diagnostic Tools:

Platform to conduct testing:

- VirtualBox: vBox is a platform that can run multiple virtual machines on a host machine while providing establishment over the hypervisor

OS flavour:

- Kali Linux: Kali Linux is the most used OS for vulnerability analysis and pen testing and attacks by Security analysts and attackers all over the world. Kali Linux comes pre-installed with various vulnerability analysis' tools which are really good for vulnerability analysis.

Vulnerability Analysis tools:

- Nmap: Nmap is used by security analysts to find open ports on a deployed server
- Nikto: Nikto is vulnerability scanner that finds possible vulnerabilities of websites that can be used by the attacker for pen testing
- WPscan: WPscan is a tool that is used for scanning vulnerabilities of websites made from WordPress
- Legion: Legion is a open source, easy-to-use, super-extensible and semi-automated network penetration testing framework that aids in discovery, reconnaissance and exploitation of information systems and comes pre-installed in Kali Linux
- Wireshark: Wireshark is an opensource tool that can be used for capturing traffic sent between a server and a client
- ZAP: ZAP is an open source vulnerability scanner pre-installed on Kali Linux

Applications for Penetration Testing:

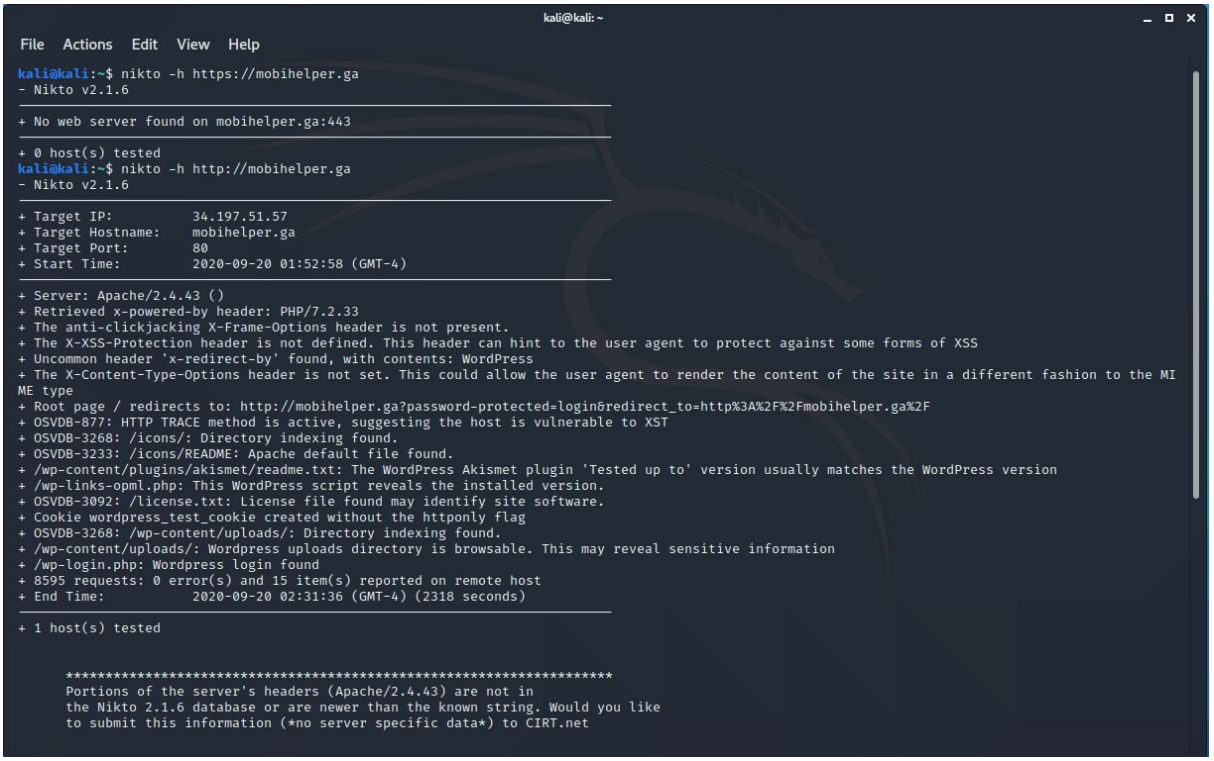
- JohnTheRipper: This tool can be used for mounting a dictionary attack on server for access
- Metasploit: Used for exploiting vulnerabilities with specific scripts after conducting analysis
- Burp Suite: Used for mounting web based attacks

B) System Component Identification & Vulnerabilities

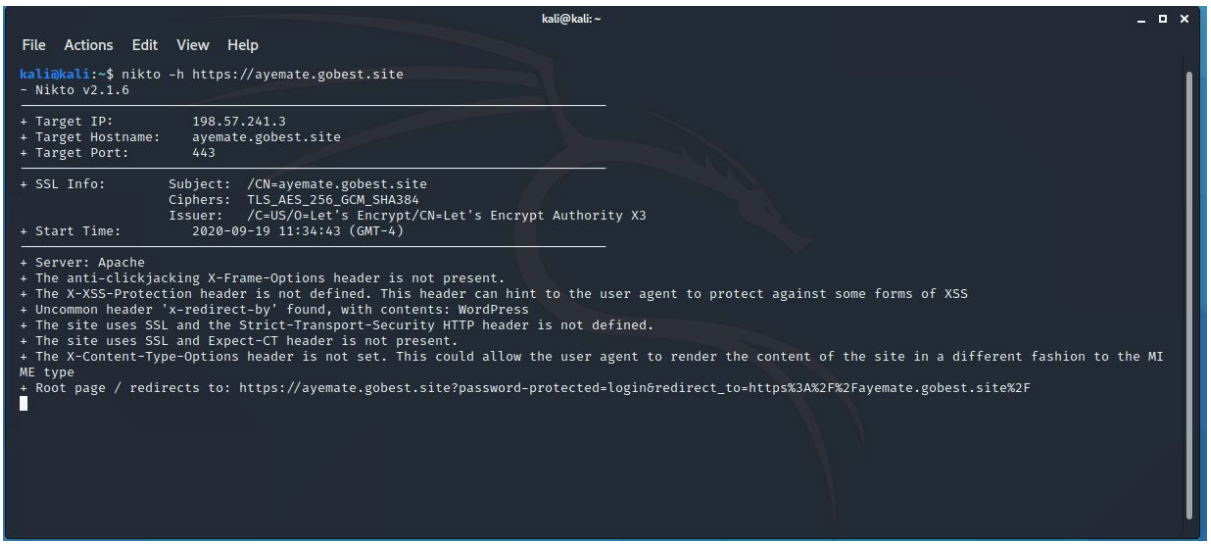
Victim Domain Name	Server OS	IP ADDRESS	Hosting Service	Analysis findings
http://mobihelper.ga	Apache	34.197.51.57	Amazon (AWS)	No SSL No HTTPS Uses WordPress Maybe vulnerable to XST Revels link that can given access to contents of the website
https://ayemate.gobest.site	Apache	198.57.241.3	Unified Layer	Uses SSL Uses HTTPS Uses WordPress Vulnerable against Clickjacking Revels link that can given access to contents of the website
https://greenskeepers.me	Nginx (Ubuntu)	35.172.115.110	Amazon (AWS)	Uses SSL Uses HTTPS Maybe vulnerable to BREACH attack Allowed HTTP methods: GET, HEAD, OPTIONS.

Nikto Vulnerability Results:

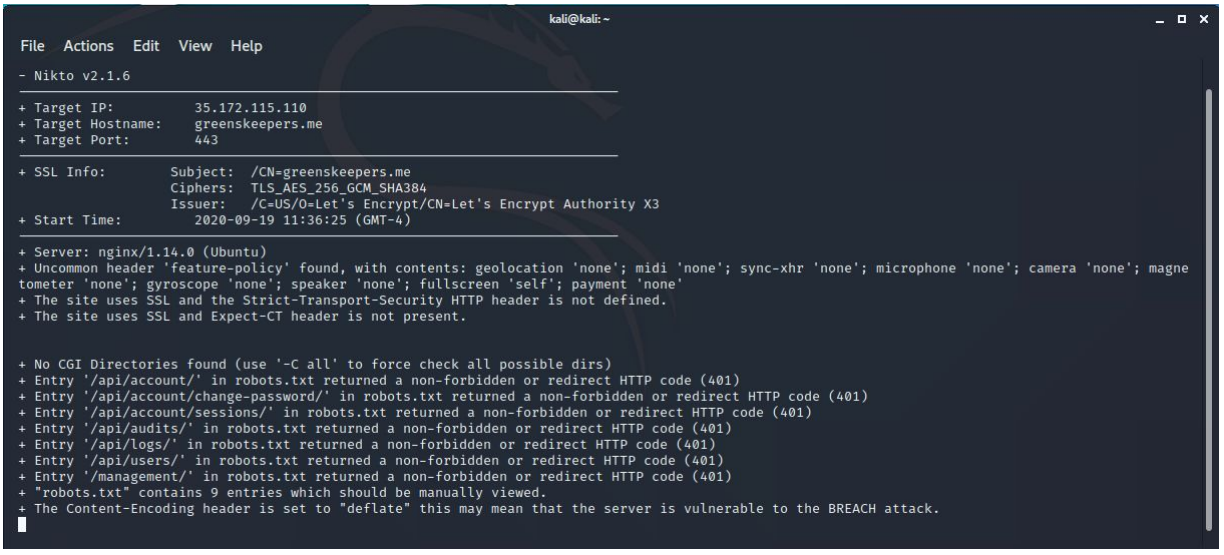
Victim 1: https://mobihelper.ga



Victim 2: https://ayemate.gobest.site



Victim 3: https://greenskeepers.me



Directory Structure Search Results:

Victim 1: https://mobihelper.ga

```
Target: http://mobihelper.ga

Output File: /Users/akash/Downloads/dirsearch/reports/mobihelper.ga/_20-09-20_20-39-26.txt

[20:39:27] Starting:
[20:39:38] 403 -  199B  - /.htaccess.bak1
[20:39:38] 403 -  199B  - /.htaccess.orig
[20:39:38] 403 -  199B  - /.htaccess.sample
[20:39:38] 403 -  199B  - /.htaccess.save
[20:39:38] 403 -  199B  - /.htaccessBAK
[20:39:38] 403 -  199B  - /.htaccessOLD
[20:39:38] 403 -  199B  - /.htaccessOLD2
[20:39:38] 403 -  199B  - /.httr-oauth
[20:40:11] 403 -  199B  - /cgi-bin/
[20:40:26] 302 -    0B  - /index.php/login/  ->  http://mobihelper.ga?password-protected=login&redirect_to=http%3A%2F%2Fmobihelper.ga%2Findex.php%2Flogin%2F
[20:40:26] 302 -    0B  - /index.php  ->  http://mobihelper.ga?password-protected=login&redirect_to=http%3A%2F%2Fmobihelper.ga%2Findex.php
[20:40:29] 200 -   19KB - /license.txt
[20:40:42] 200 -    7KB - /readme.html
[20:40:55] 301 -  238B  - /wp-admin  ->  http://mobihelper.ga/wp-admin/
[20:40:55] 200 -    0B  - /wp-content/
[20:40:55] 301 -  254B  - /wp-content/ai1wm-backups  ->  http://mobihelper.ga/wp-content/ai1wm-backups/
[20:40:55] 200 -   26B  - /wp-content/ai1wm-backups/
[20:40:55] 200 -   69B  - /wp-content/plugins/akismet/akismet.php
[20:40:55] 301 -  280B  - /wp-content/plugins/all-in-one-wp-migration/storage  ->  http://mobihelper.ga/wp-content/plugins/all-in-one-wp-migration/storage/
[20:40:55] 409 -    3KB - /wp-admin/setup-config.php
[20:40:55] 200 -    1KB - /wp-admin/install.php
[20:40:55] 200 -    2KB - /wp-content/uploads/
[20:40:55] 301 -  240B  - /wp-content  ->  http://mobihelper.ga/wp-content/
[20:40:55] 302 -    0B  - /wp-admin/  ->  http://mobihelper.ga/wp-login.php?redirect_to=http%3A%2F%2Fmobihelper.ga%2Fwp-admin%2F&reauth=1
[20:40:56] 301 -  241B  - /wp-includes  ->  http://mobihelper.ga/wp-includes/
[20:40:56] 500 -    0B  - /wp-includes/rss-functions.php
[20:40:56] 200 -    7KB - /wp-login.php
[20:40:56] 302 -    0B  - /wp-signup.php  ->  http://mobihelper.ga/wp-login.php?action=register
[20:40:56] 200 -   48KB - /wp-includes/
[20:40:57] 405 -   42B  - /xmlrpc.php

Task Completed
```

Victim 2: https://ayemate.gobest.site

```
Target: https://ayemate.gobest.site

Output File: /Users/akash/Downloads/dirsearch/reports/ayemate.gobest.site/_20-09-20_20-44-11.txt

[20:44:11] Starting:
[20:44:21] 406 -  226B  - /CHANGELOG.md.old
[20:44:21] 406 -  226B  - /CHANGELOG.md.bak
[20:44:25] 302 -    0B  - /CHANGELOG.md  ->  https://ayemate.gobest.site?password-protected=login&redirect_to=https%3A%2F%2Fayemate.gobest.site%2FCHANGELOG.md
[20:44:32] 406 -  226B  - /.bash_history
[20:44:37] 406 -  226B  - /.c9/metadata/environment/.env
[20:44:44] 406 -  226B  - /.config.inc.php.swp
[20:44:55] 406 -  226B  - /.docker/.env
[20:44:55] 406 -  226B  - /.docker/laravel/app/.env
[20:45:02] 406 -  226B  - /.env.backup
[20:45:08] 406 -  226B  - /.env~
[20:45:29] 406 -  226B  - /.gitignore.orig
[20:45:29] 406 -  226B  - /.gitignore.swp
[20:45:29] 406 -  226B  - /.gitlab-ci/.env
[20:45:40] 406 -  226B  - /.htaccess.bak
[20:45:41] 406 -  226B  - /.htaccess.old
[20:45:41] 406 -  226B  - /.htaccess.orig
[20:45:43] 406 -  226B  - /.htpasswd.bak
[20:45:55] 406 -  226B  - /.index.php.swp
[20:46:03] 406 -  226B  - /.keys.yml.swp
[20:46:15] 406 -  226B  - /.mdb
[20:47:11] 406 -  226B  - /.ssh/id_dsa
[20:47:11] 406 -  226B  - /.ssh/id_rsa
[20:47:36] 406 -  226B  - /.travis.yml.swp
[20:47:36] 406 -  226B  - /.travis.yml~
[20:47:44] 406 -  226B  - /.vscode/.env
[20:47:49] 301 -  263B  - /.well-known/acme-challenge  ->  https://ayemate.gobest.site/.well-known/acme-challenge/
[20:48:06] 406 -  226B  - /.wp-config.php.swo
[20:48:06] 406 -  226B  - /.wp-config.swp
[20:48:16] 406 -  226B  - /0.htpasswd
[20:48:23] 406 -  226B  - /1.htaccess
[20:48:23] 406 -  226B  - /1.htpasswd
[20:48:23] 406 -  226B  - /1.sql
[20:48:29] 406 -  226B  - /2.sql
[20:48:34] 406 -  226B  - /2012.sql
[20:48:35] 406 -  226B  - /2013.sql
[20:48:36] 406 -  226B  - /2014.sql
[20:48:39] 406 -  226B  - /2015.sql
[20:48:41] 406 -  226B  - /2016.sql
[20:48:42] 406 -  226B  - /2017.sql
[20:48:44] 406 -  226B  - /2018.sql
[20:48:45] 406 -  226B  - /2019.sql
[20:48:47] 406 -  226B  - /2020.sql
[20:51:57] 406 -  226B  - /_.htpasswd
[20:54:14] 302 -    0B  - /adminCHANGELOG.md  ->  https://ayemate.gobest.site?password-protected=login&redirect_to=https%3A%2F%2Fayemate.gobest.site%2FadminCHANGELOG.md
[20:54:30] 406 -  226B  - /admin.mdb
[20:54:30] 406 -  226B  - /admin.old
[20:55:02] 406 -  226B  - /admin/includes/configure.php~
[20:58:07] 406 -  226B  - /affiliates.sql
```

Victim 3: https://greenskeepers.me

```
Target: https://greenskeepers.me

Output File: /Users/akash/Downloads/dirsearch/reports/greenskeepers.me/_20-09-20_21-01-25.txt

[21:01:25] Starting:
[21:01:28] 400 -  182B  - /%2e%2e//google.com
[21:01:28] 500 -  173B  - /..;/
[21:01:29] 200 -    5KB - /.aws/config
[21:01:29] 200 -    5KB - /.bzr/README
[21:01:29] 200 -    5KB - /.bzr/branch-format
[21:01:30] 200 -    5KB - /.aws/credentials
[21:01:30] 200 -    5KB - /.cointop/config
[21:01:30] 200 -    5KB - /.concrete/dev_mode
[21:01:30] 200 -    5KB - /.config/gcloud/configurations/config_default
[21:01:30] 200 -    5KB - /.config/gcloud/credentials
[21:01:32] 200 -    5KB - /.gem/credentials
[21:01:32] 200 -    5KB - /.git/COMMIT_EDITMSG
[21:01:32] 200 -    5KB - /.git/FETCH_HEAD
[21:01:32] 200 -    5KB - /.git/config
[21:01:32] 200 -    5KB - /.git/description
[21:01:32] 200 -    5KB - /.git/configf
[21:01:32] 200 -    5KB - /.git/head
[21:01:32] 200 -    5KB - /.git/index
[21:01:32] 200 -    5KB - /.git/info/attributes
[21:01:32] 200 -    5KB - /.git/info/exclude
[21:01:32] 200 -    5KB - /.git/logs/head
[21:01:32] 200 -    5KB - /.git/logs/refs
[21:01:32] 200 -    5KB - /.git/logs/refs/heads
[21:01:32] 200 -    5KB - /.git/logs/refs/heads/master
[21:01:32] 200 -    5KB - /.git/logs/refs/remotes
[21:01:32] 200 -    5KB - /.git/logs/refs/remotes/origin
[21:01:32] 200 -    5KB - /.git/logs/refs/remotes/origin/HEAD
[21:01:32] 200 -    5KB - /.git/logs/refs/remotes/origin/master
[21:01:32] 200 -    5KB - /.git/packed-refs
[21:01:32] 200 -    5KB - /.git/refs/heads/master
[21:01:32] 200 -    5KB - /.git/refs/heads
[21:01:32] 200 -    5KB - /.git/refs/remotes
[21:01:32] 200 -    5KB - /.git/refs/remotes/origin
[21:01:32] 200 -    5KB - /.git/refs/remotes/origin/HEAD
[21:01:32] 200 -    5KB - /.git/refs/tags
[21:01:32] 200 -    5KB - /.git/refs/remotes/origin/master
[21:01:33] 200 -    5KB - /.hg/branch
[21:01:33] 200 -    5KB - /.hg/dirstate
[21:01:33] 200 -    5KB - /.hg/hgrc
[21:01:33] 200 -    5KB - /.hg/requires
[21:01:33] 200 -    5KB - /.hg/store/undo
[21:01:33] 200 -    5KB - /.idea/dictionaries
[21:01:33] 200 -    5KB - /.idea/httprequests
[21:01:33] 200 -    5KB - /.idea/libraries
[21:01:33] 200 -    5KB - /.idea/modules
[21:01:34] 200 -    5KB - /.kube/config
[21:01:35] 200 -    5KB - /.magnolia/installer/start
[21:01:35] 200 -    5KB - /.op/config
[21:01:37] 200 -    5KB - /.ssh/authorized_keys
[21:01:37] 200 -    5KB - /.ssh/config
[21:01:37] 200 -    5KB - /.ssh/ansible_rsa
[21:01:37] 200 -    5KB - /.ssh/google_compute_engine
[21:01:37] 200 -    5KB - /.ssh/id_dsa
[21:01:37] 200 -    5KB - /.ssh/id_rsa
[21:01:37] 200 -    5KB - /.ssh/identity
```

C) Hosting provider security policies and restrictions

AWS CLOUD SECURITY POLICY:

1. Penetration testing can only be performed on the following AWS services:
    a. Amazon EC2 instances, NAT Gateways, and Elastic Load Balancers
    b. Amazon RDS
    c. Amazon CloudFront
    d. Amazon Aurora
    e. Amazon API Gateway
    f. AWS Lambda and Edge Functions
    g. Amazon Lightsail Resources
    h. Amazon Elastic Beanstalk resources
2. Penetration Testing Prohibited Activities are:
    a. DNS Zone walking
    b. Port Flooding
    c. Protocol Flooding
    d. Request Flooding
    e. Denial Of Service(DOS), DIstributed Denial Of Service(DDoS), Simulated DoS
3. If somebody discovers a security issue within any AWS services in the course of security assessment, it has to be reported to AWS
4. Security groups
    a. Amazon offers a virtual firewall facility for filtering the traffic that crosses your cloud network segment; but the way that AWS firewalls are managed differs slightly from the approach used by traditional firewalls. The central component of AWS firewalls is the “security group”
    b. AWS rules let you specify the traffic source, or the traffic destination — but not both on the same rule. For Inbound rules, there is a source that states where the traffic comes from, but no destination telling it where to go. For Outbound rules, it the other way around: you can specify the destination but not the source. The reason for this is that the AWS security group always sets the unspecified side (source or destination, as the case may be) as the instance to which the security group is applied.
5. AWS VPC Flow Logs
    a. Enable AWS VPC Flow Logs for your VPC or Subnet or ENI level. AWS VPC flow logs can be configured to capture both accept and reject entries flowing through the ENI and Security groups of the EC2, ELB, and some additional services. These VPC Flow log entries can be scanned to detect attack patterns, alert abnormal activities and information flow inside the VPC, and provide valuable insights to the SOC/MS team operations
6. Managed policies
    a. Managed policies came later. So not all old AWS cloud best practices that existed during the inline policies era might hold good in the present day. So, it is

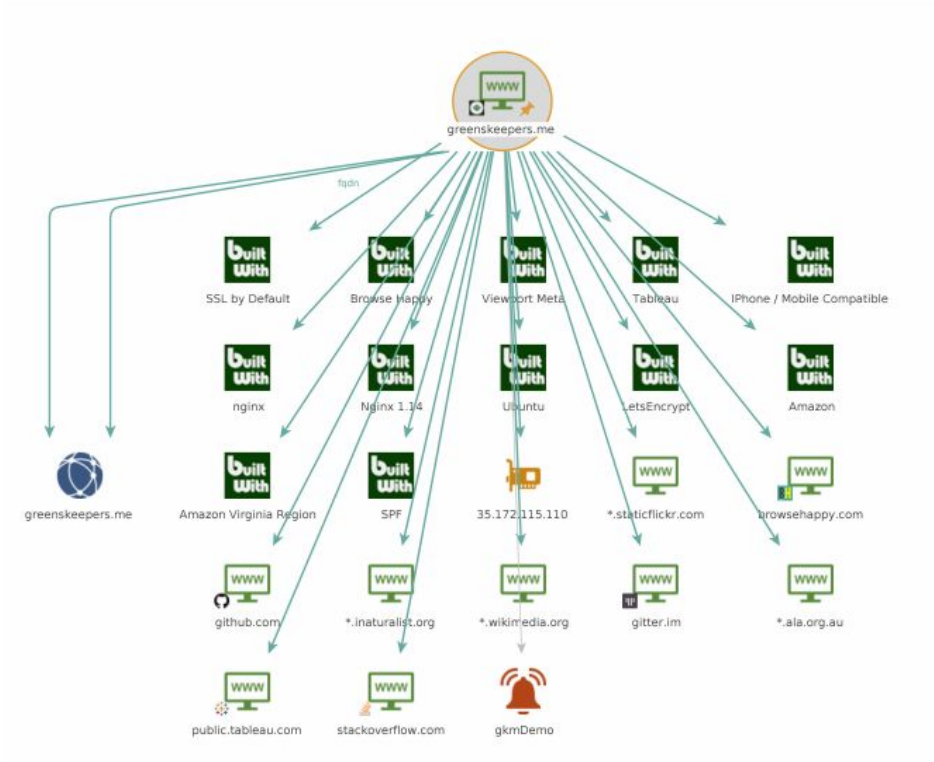
recommended that you use managed policies that are available now. With managed policies you can manage permissions from a central place rather than having it attached directly to users. It also enables you to properly categorize policies and reuse them. Updating permissions also becomes easier when a single managed policy is attached to multiple users. Plus, in managed policies you can add up to 10 managed policies to a user, role, or group. The size of each managed policy, however, cannot exceed 5,120 characters.

Part 2: Vulnerability Analysis & Attack Plan

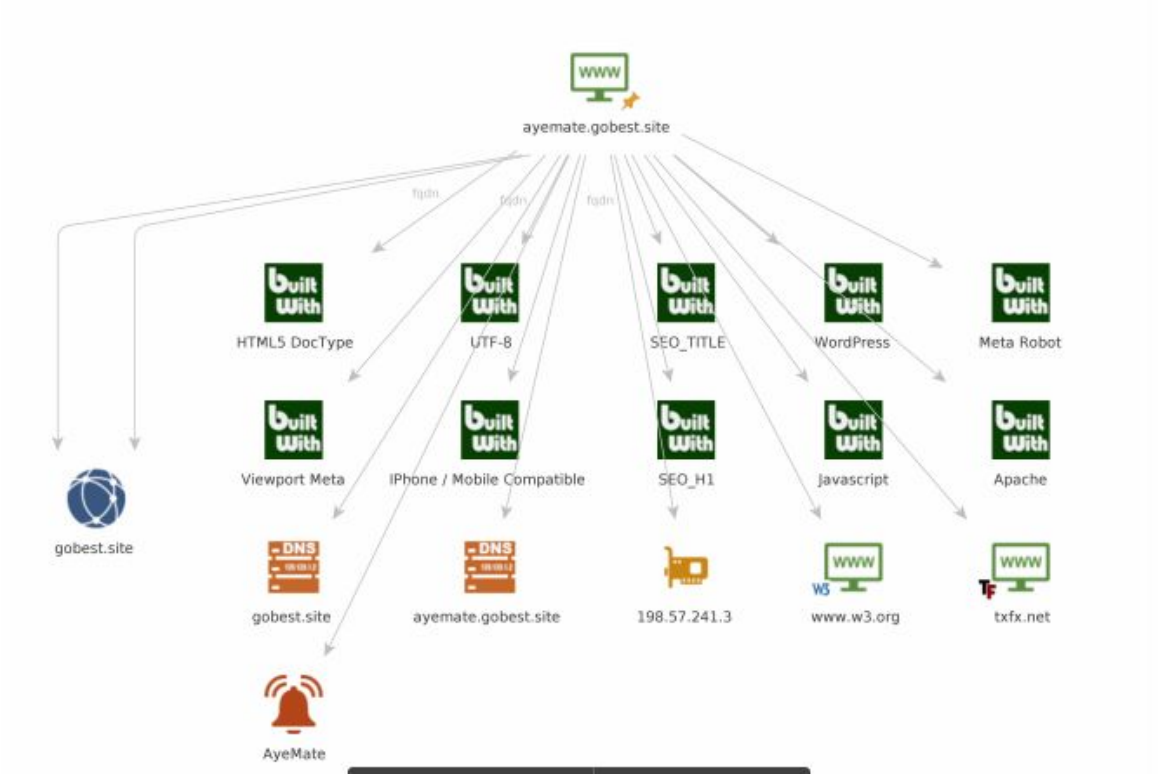
a) Vulnerability Analysis

From Part 1 we used Maltego on all websites to get more information about the websites. The complete report of Maltego's Analysis on the website is given in the summary graphs of all websites are given below.

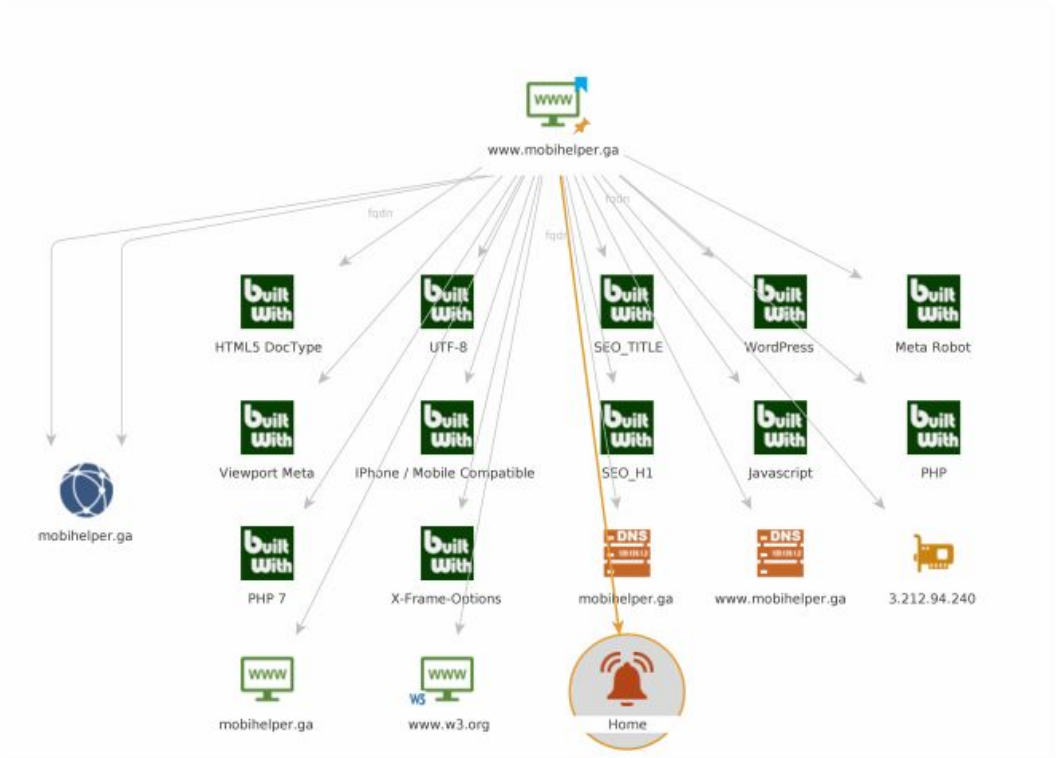
1) Greenskeepers.me:



2) Ayemate.gobest.site:



3) mobihelper.ga



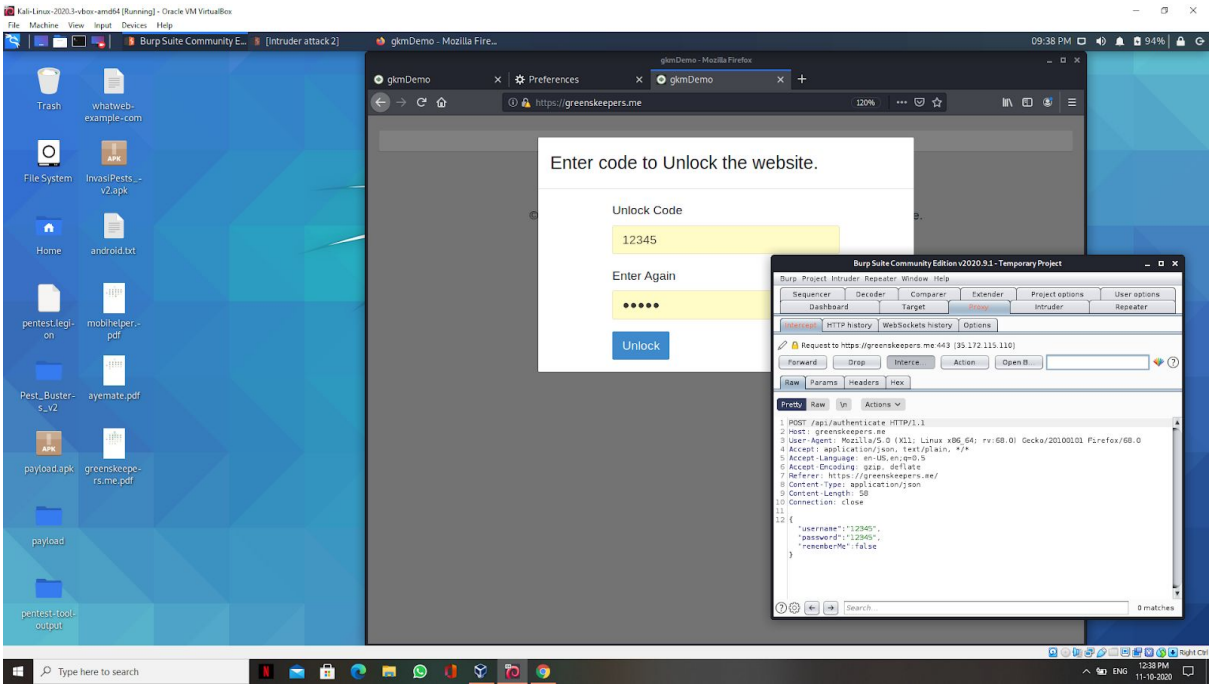
Upon researching the documents the IP addresses and the build was found. This information is similar to that obtained by nitko which confirms our findings carried out in part 1.

After that we started fiddling around with the websites using burpsuite proxy to find any vulnerabilities we can spot that we can exploit during pentesting of the websites. The following analysis was found.

1) Greenskeepers.me

During analysis on greenskeepers.me login page we were able to find the api gateway that was used to authenticate the user on the website. This also disclosed the json format of the body that is sent in the API request. This is showcased in the image below.

‘



As the image above shows greenskeepers.me uses “POST” method and the api gateway is “api/authenticate” using HTTP/1.1 and the body of the API request can also be seen in JSON format. the JSON body is as follows:

{

“username”:”12345”

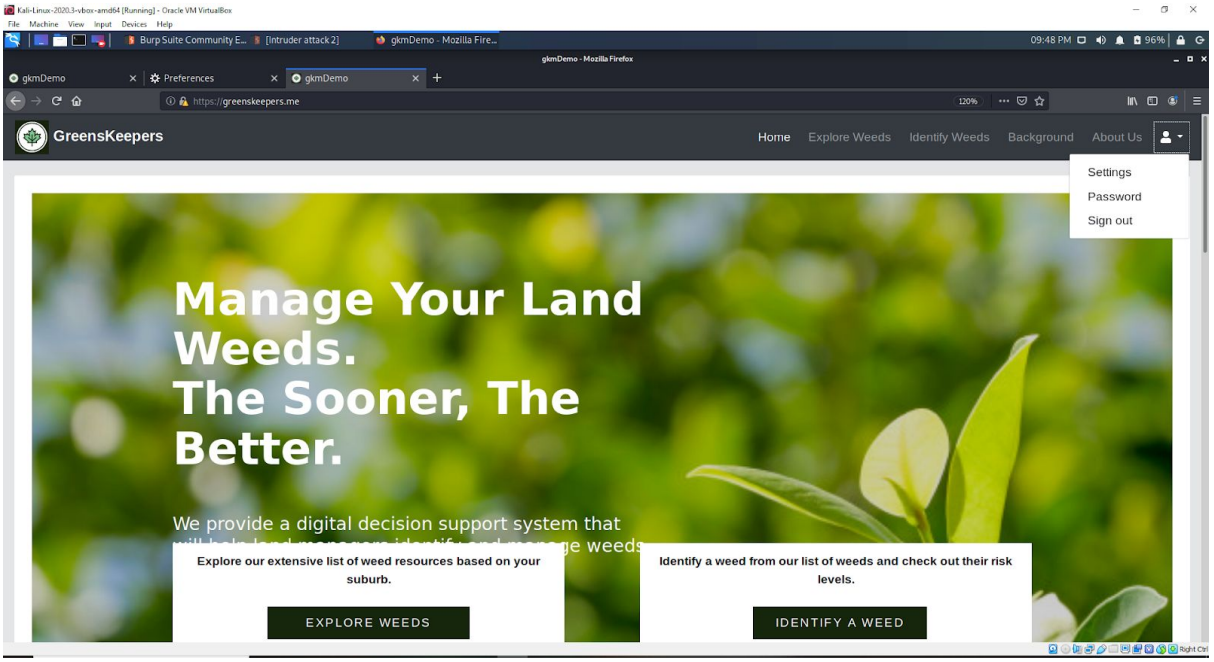
“password”:”12345”

“rememberMe”:”false”

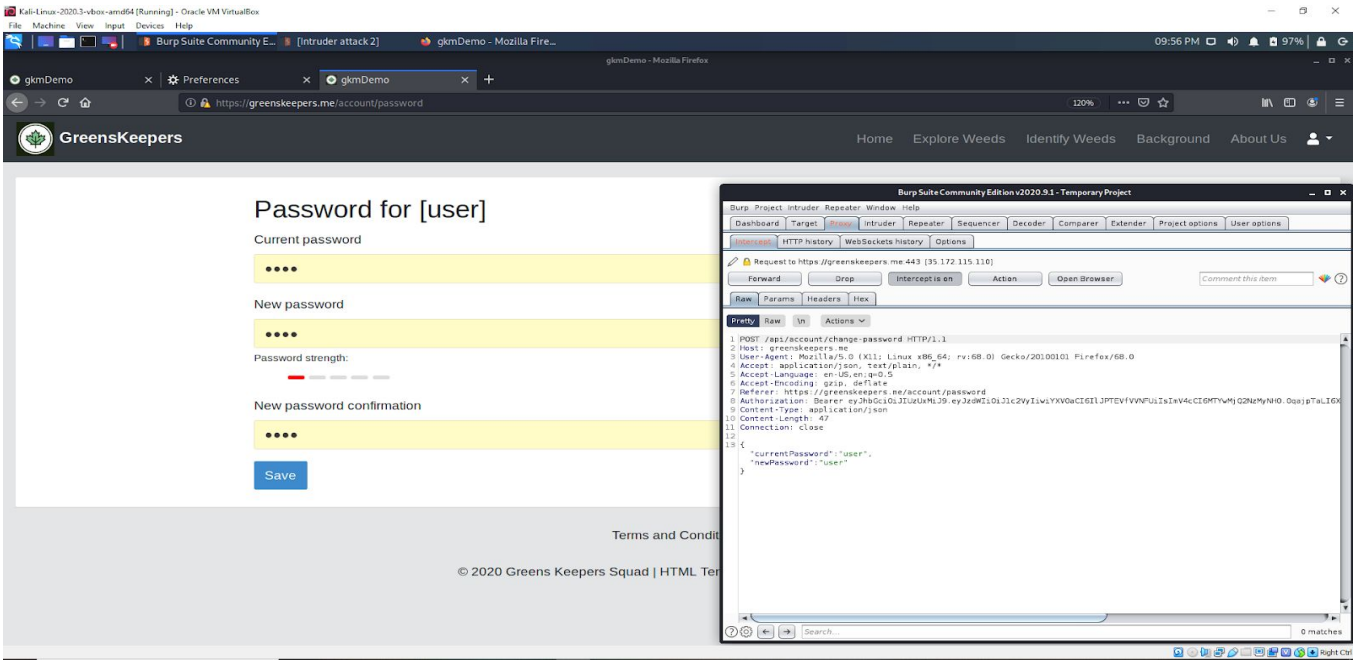
}

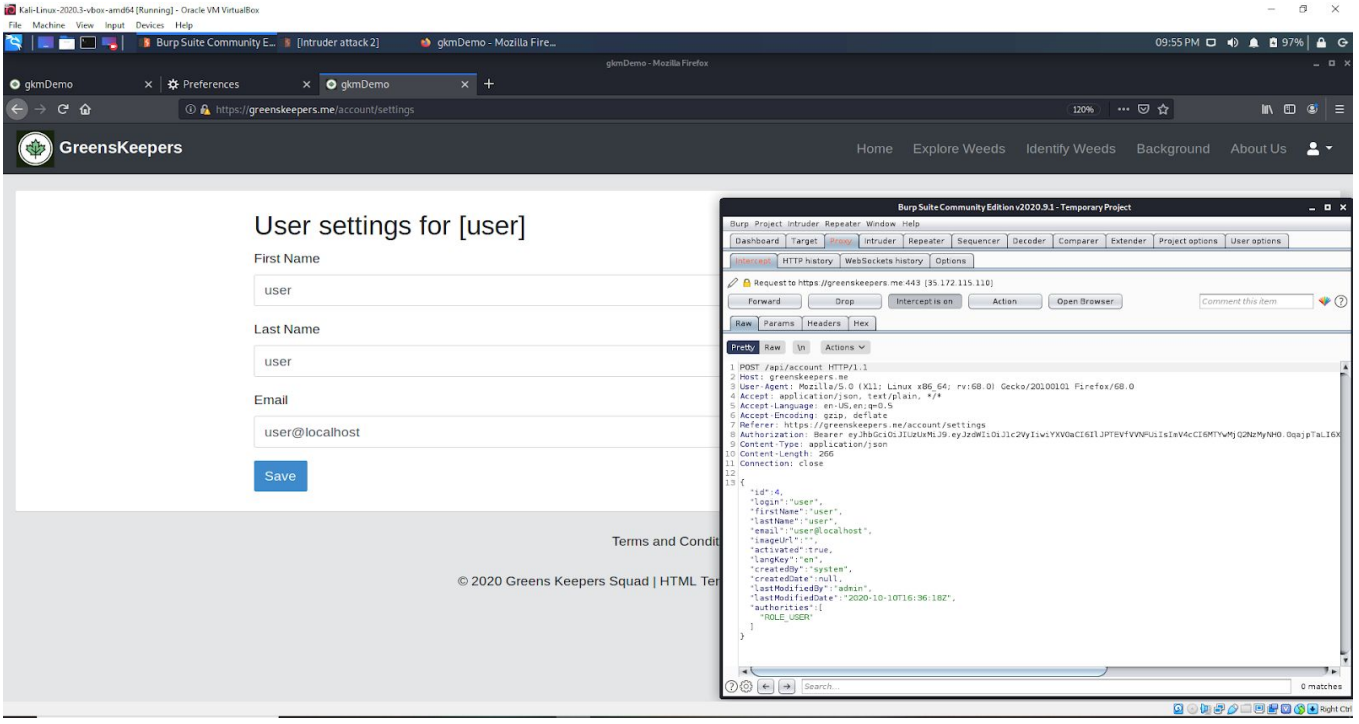
This means that the website is vulnerable to brute force attacks.

Since we know the user login for the website we can enter the website using that to see any other vulnerabilities we can exploit. The homepage of the website looks as follows



As you can see on the top right hand side once we login we can get the settings, password and sign out option for the user. The pages returned by Settings and password are shown below along with the API gateways used for these and the JSON bodys were available for these as well.

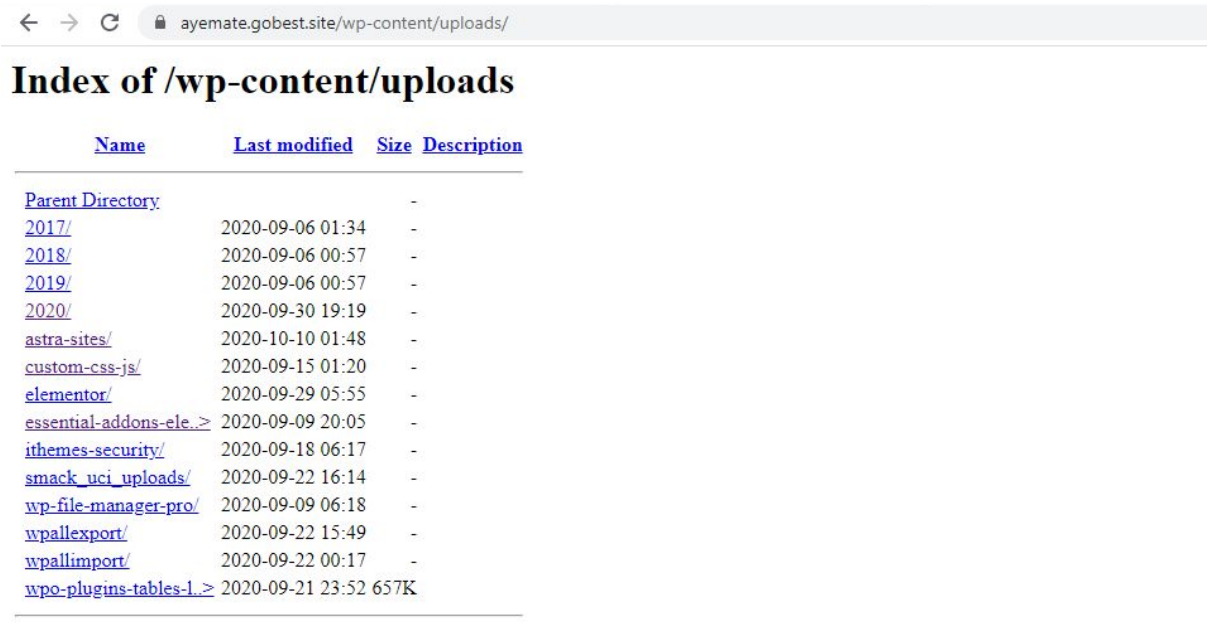




Since these vulnerabilities were found the attack plan can be formed for getting ADMIN access to the website.

2) ayemate.gobest.site

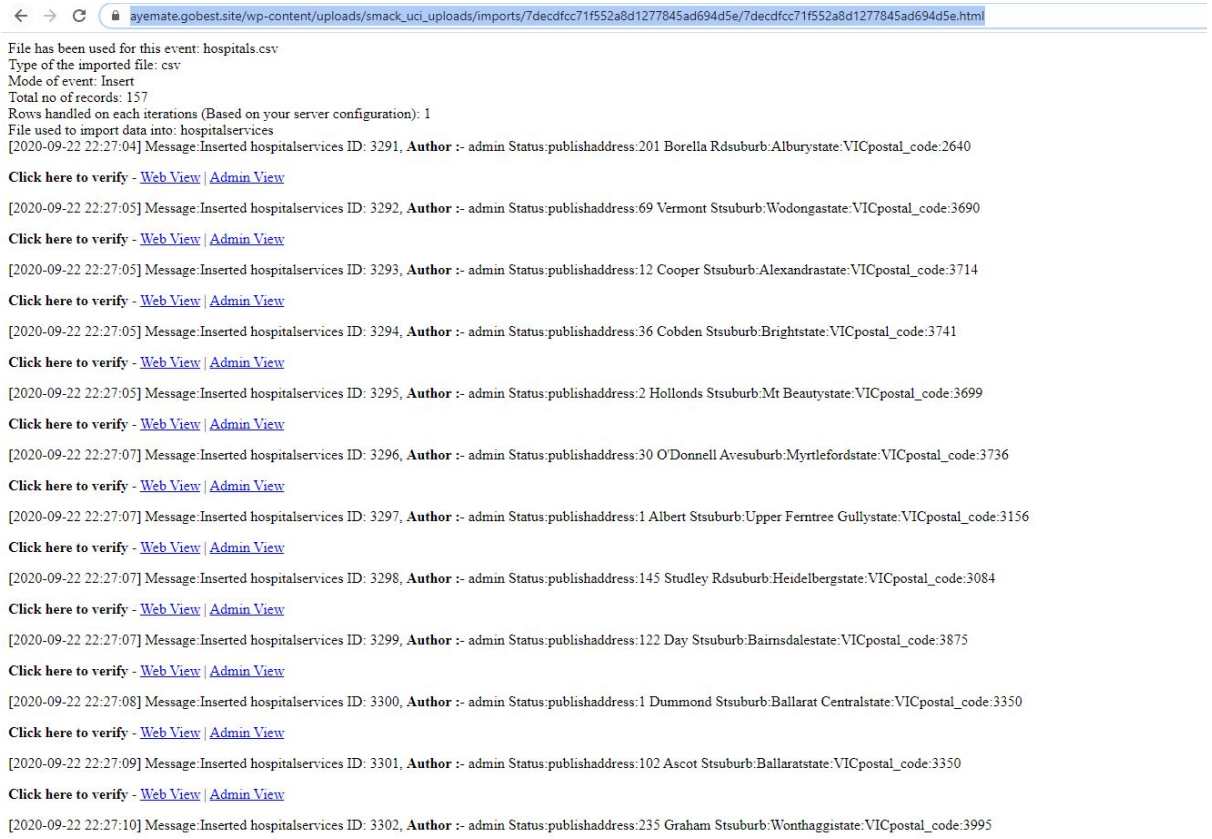
From the research carried out we know that the website is made using wordpress and is vulnerable to many attacks.Although the latest version of wordpress is 5.5.1 which was released in september 2020 being fairly new it means that it is protected from most common known vulnerabilities. Upon analysis we found a directory that could be accessed which was “/wp-content/uploads”. The directory can be seen in the image below



Upon carrying out reconnaissance on this directory we found out the following link:

https://ayemate.gobest.site/wp-content/uploads/smack_uci_uploads/imports/7decdfcc71f552a8d1277845ad694d5e/7decdfcc71f552a8d1277845ad694d5e.html

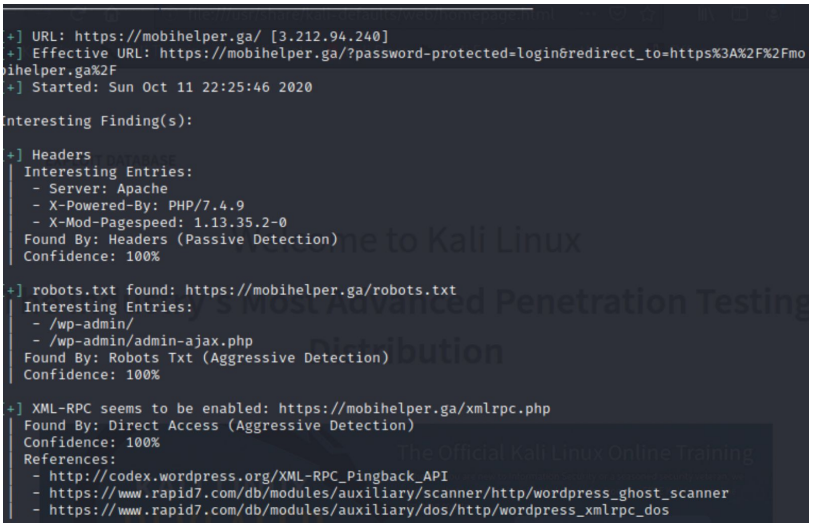
The following logs were obtained from the link above and can be seen in the image below



We can see in the image above the Author is admin. This reveals information that can mean the possible username is admin which is the default wordpress login used and hasn't been changed. This directory also revealed other information. We tried using wpscan to get the usernames so we can try and bruteforce in but it did not reveal any usernames but the logs above still gave us an idea of possible usernames.

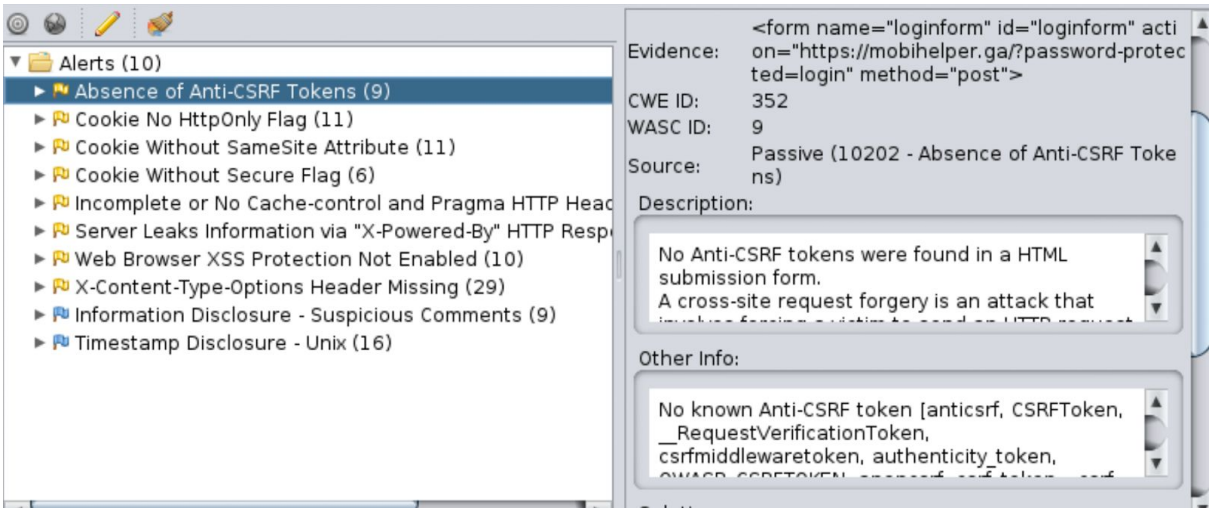
3) mobihelper.ga

From the analysis we found out that they have secured the website with SSL which was not there previously. As it is a wordpress website from the previous findings, wpscan is vulnerability scanner tool used to scan wordpress installations to find security issues.



mobihelper.ga wpscan result

Also some vulnerabilities found from ZAP tool, which is an penetration testing tool for finding vulnerabilities some of the vulnerabilities are as follows



b) Attack Plan

Upon analysis of the vulnerabilities the attack plan for the websites is as follows:

1) Greenskeepers.me

Using the vulnerabilities of the API gateways used we will carry out a brute force attack using burpsuite's Intruder tool to get admin access. We will generate a word list of commonly used usernames and passwords that can be added into the intruder tool and generate a PAYLOAD using the world list. We will then brute force the login to get Admin access.

2) Ayemate.gobest.site

Using the vulnerabilities found in "part a" above, we know the admin login web page can be accessed using the link "ayemate.gobest.site/wp-admin/". Using this and the logs we found we can try various word lists to use a simple brute force attack on the system.

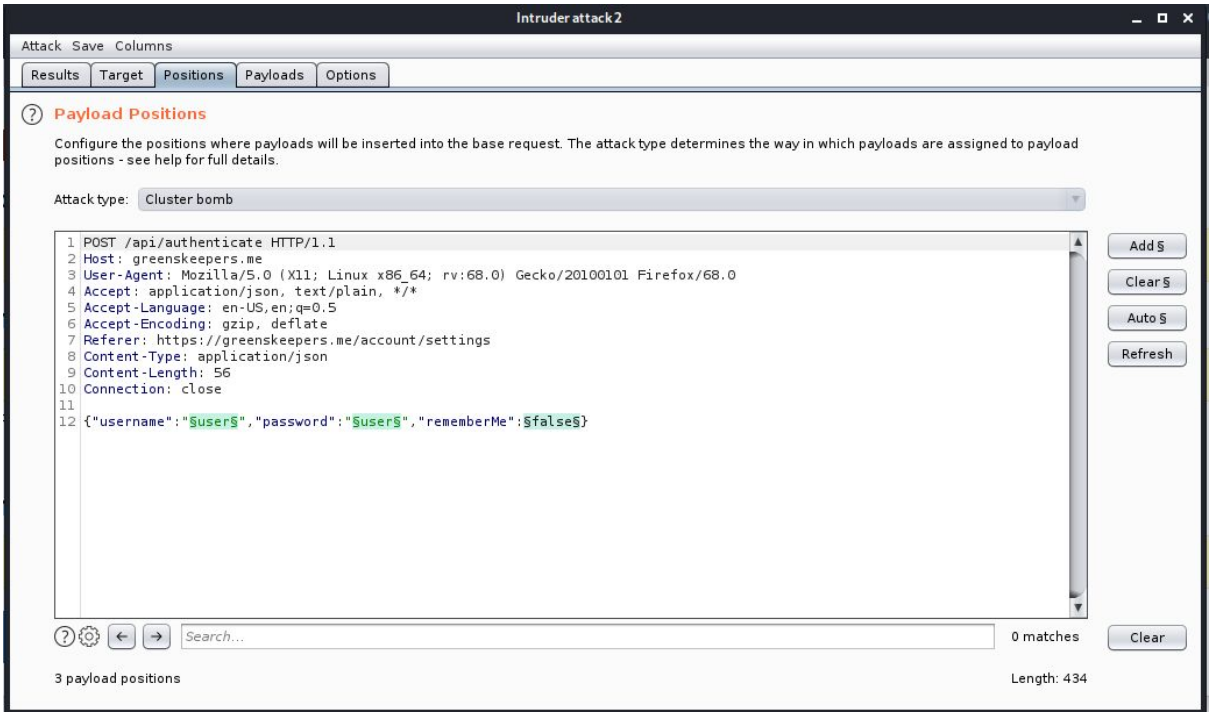
3) mobihelper.ga

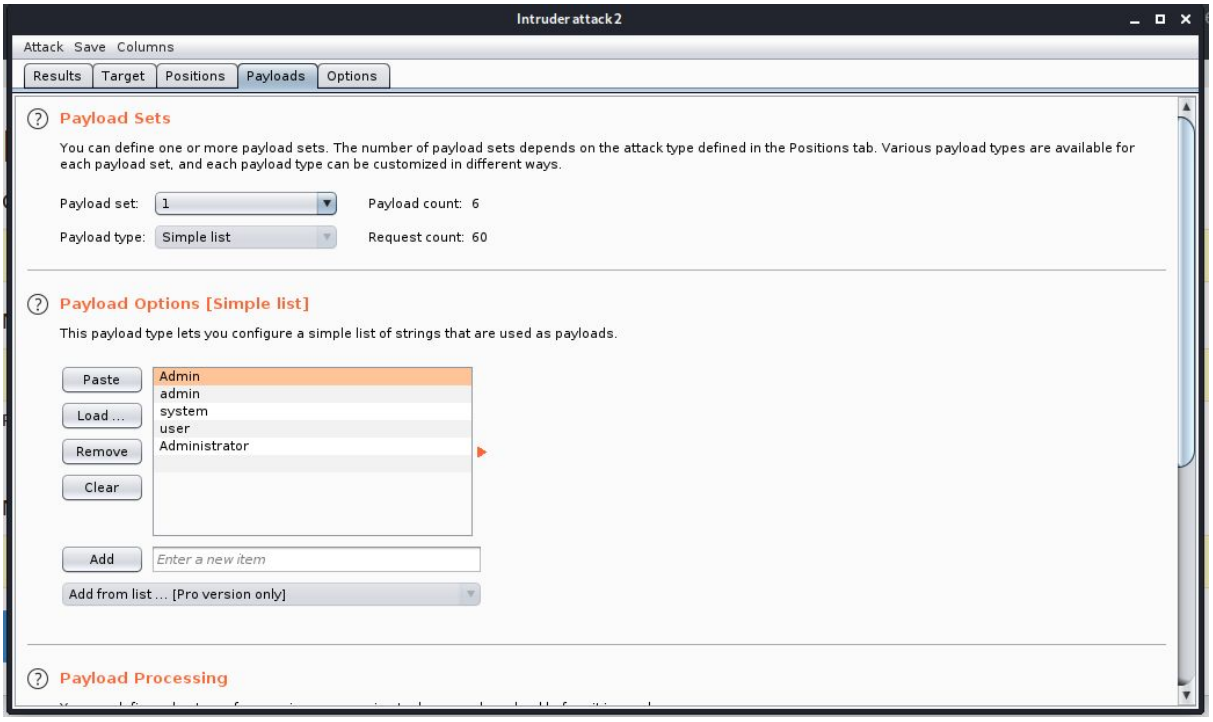
Using the vulnerabilities found in part a above, using burpsuite we can grab the code used for entering username and password. Then we will find the correct username using CeWL to create a username list then using Hydra to brute force usernames. Then we can use either hydra or wpscan to bruteforce the password. This way we can get both the username and password of admin.

Part 3: Penetration Testing & Findings

1) greenskeepers.me

As mentioned in Part 2, we will use burpsuite tool to try and get Admin access into greenskeepers.me. The settings and payload for the attack can be seen in the images below.





There are 3 payload sets we need to add according to the JSON body of the API post request,

The JSON body looks like follows:

{

“username”:”12345”

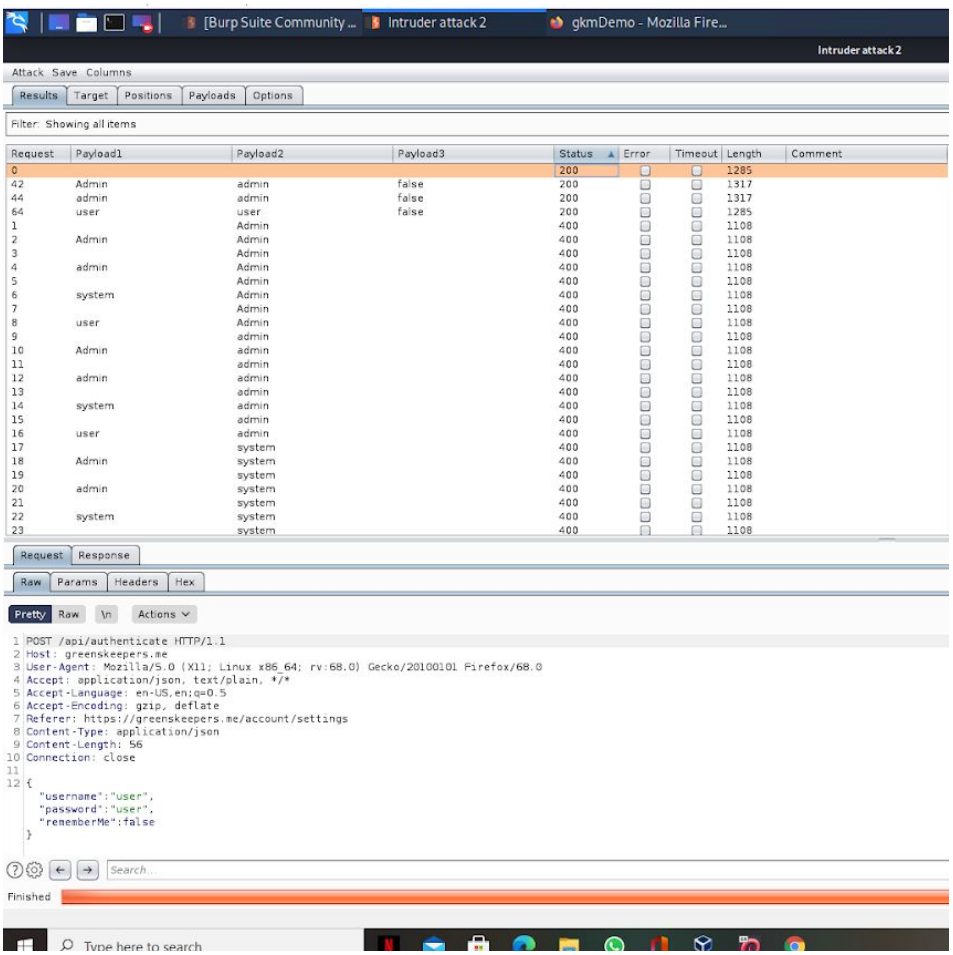
“password”:”12345”

“rememberMe”:”false”

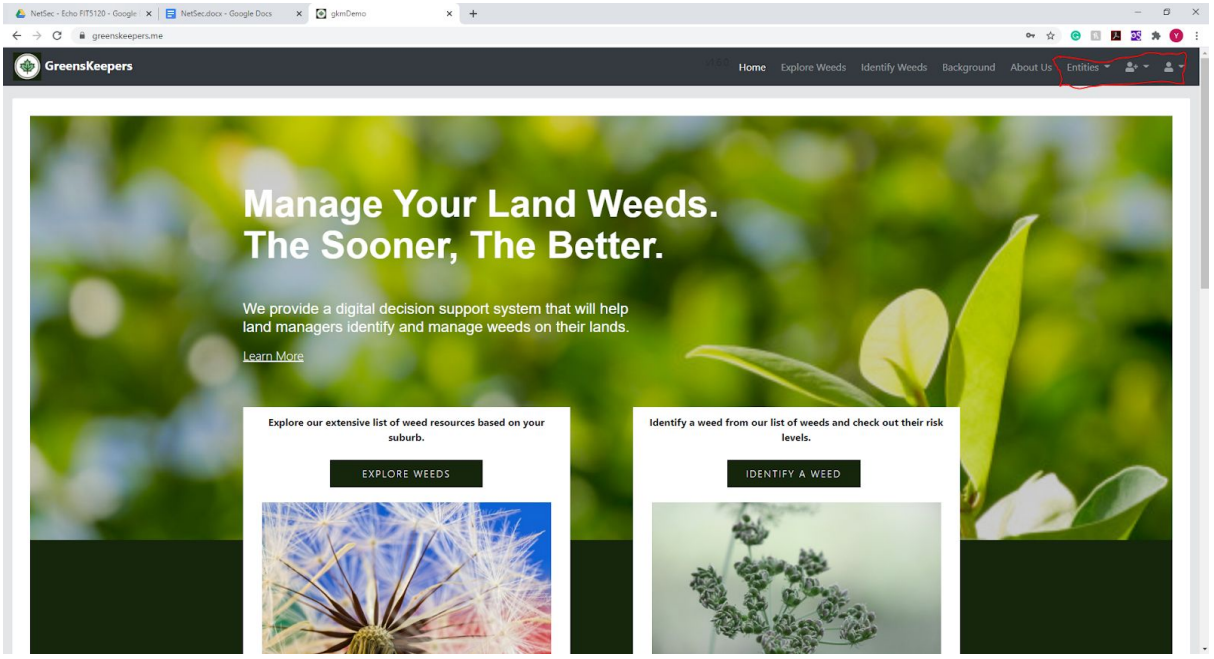
}

The 3 payload sets will be for username, password and rememberMe respectively, we make payload 1 and 2 with a wordlist as shown in the image above and keep rememberME as false in payload 3.

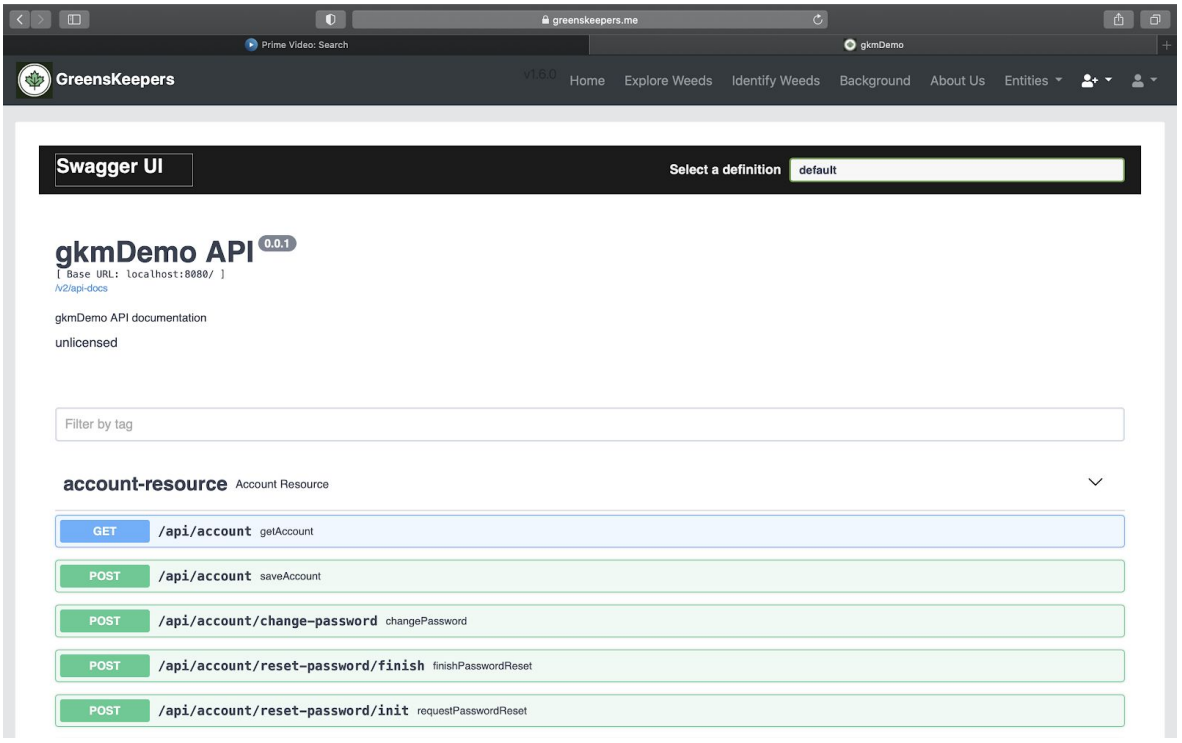
Once this is done we can start the attack. Once the attack was carried out the following was reported as seen in the image below.

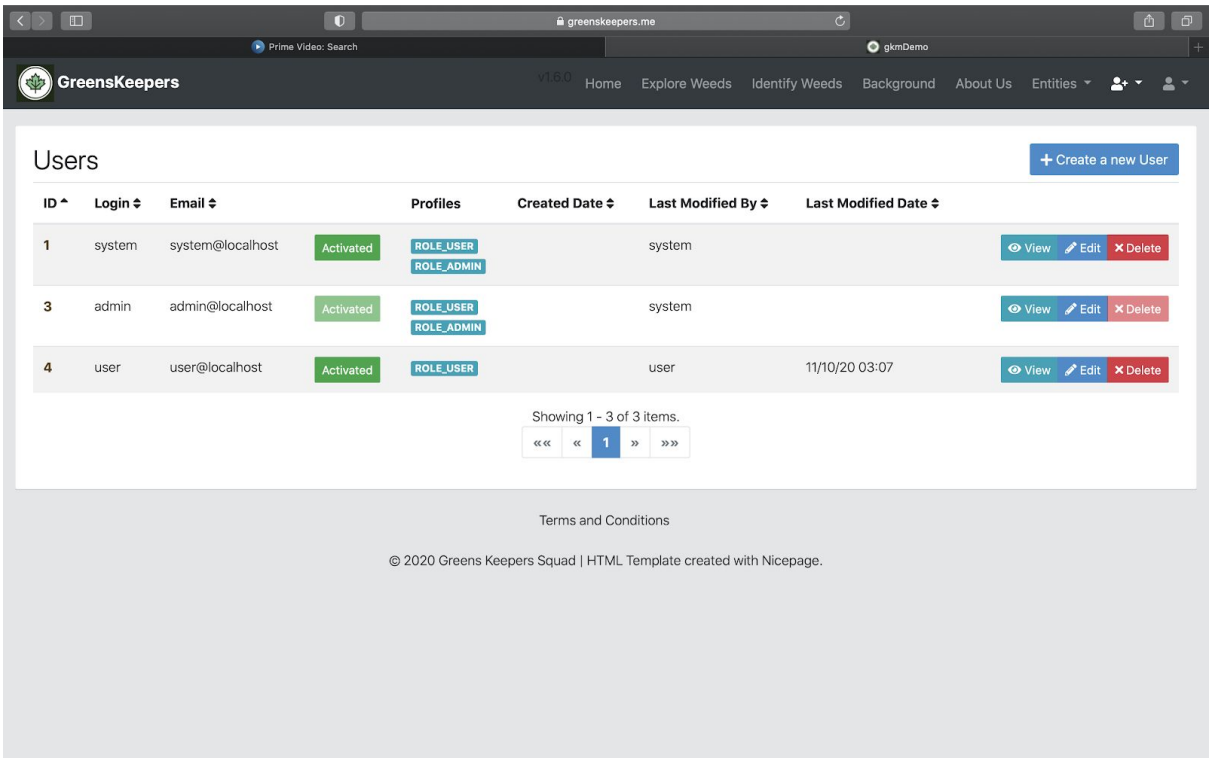


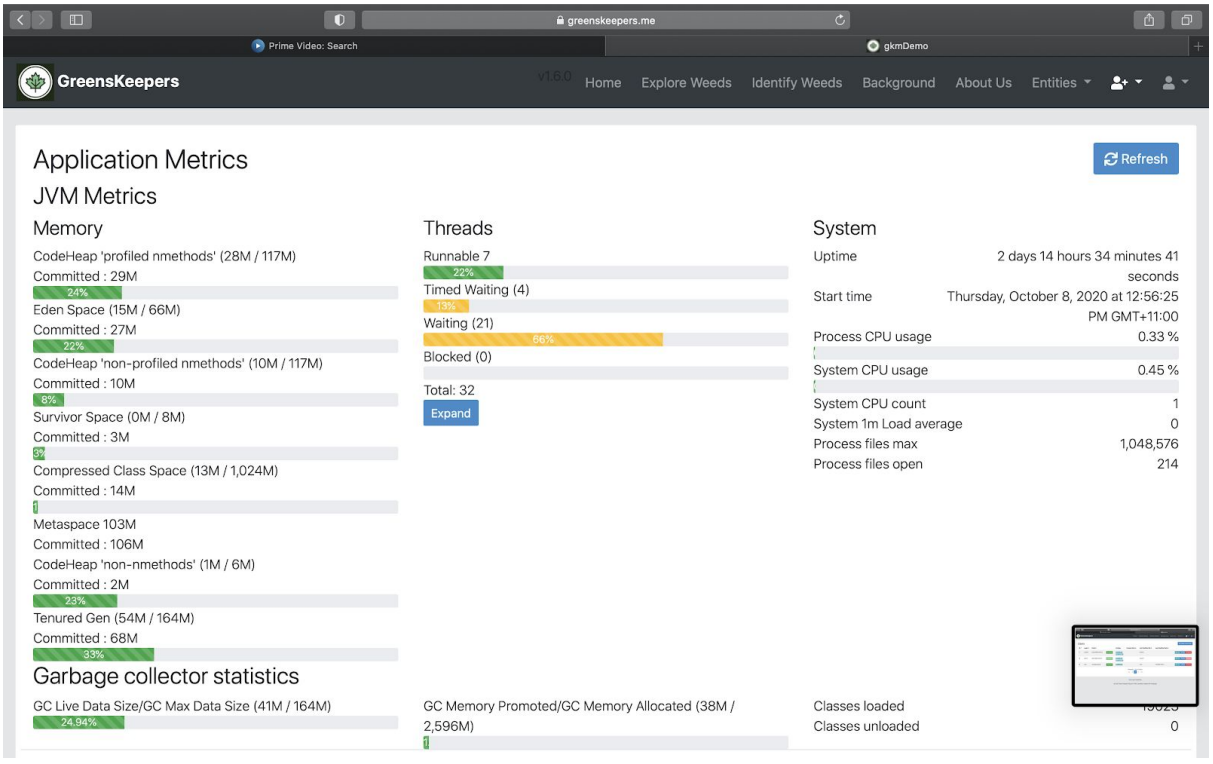
As seen in the image above we can get the passwords of the admin access on the status code the websites server responded with. We know that **200 means OK, 400 means error and 401 means not authorized**. We can also see the attacked returned 200 on Admin and admin as the username and password. After we tried that in the login we managed to get admin access to the website as shown in the image below.

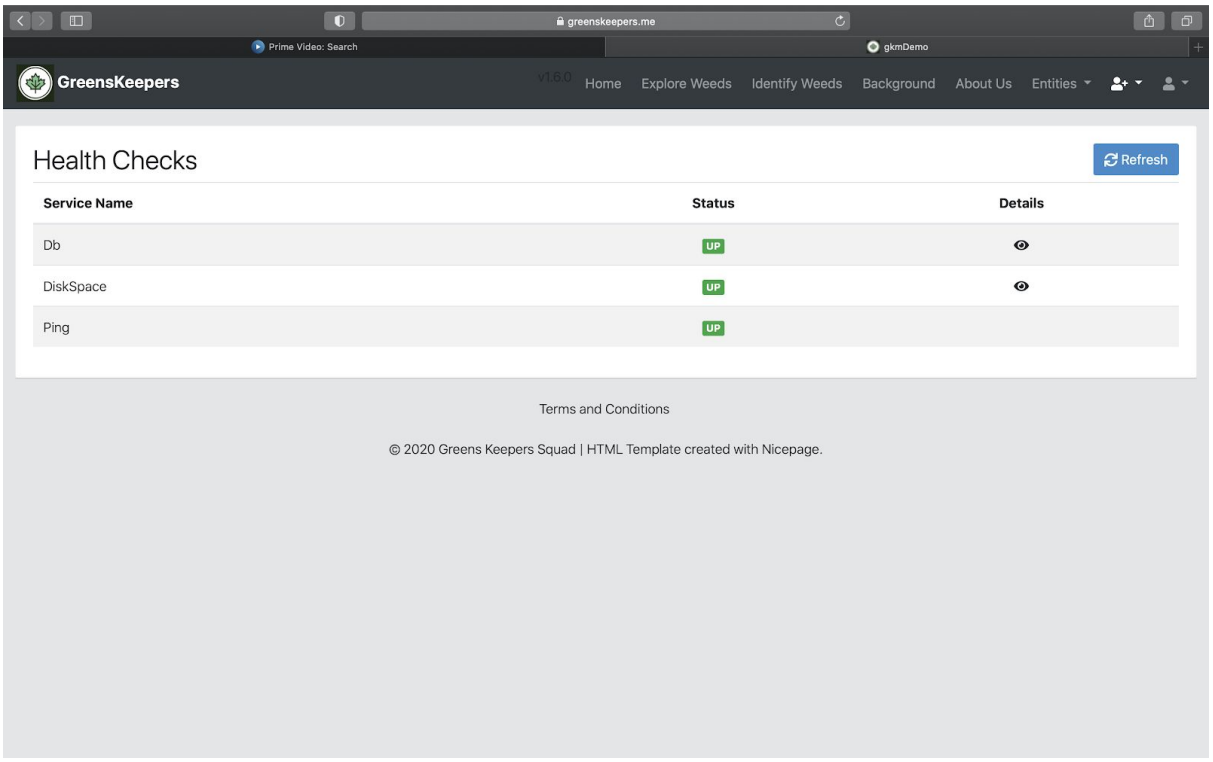


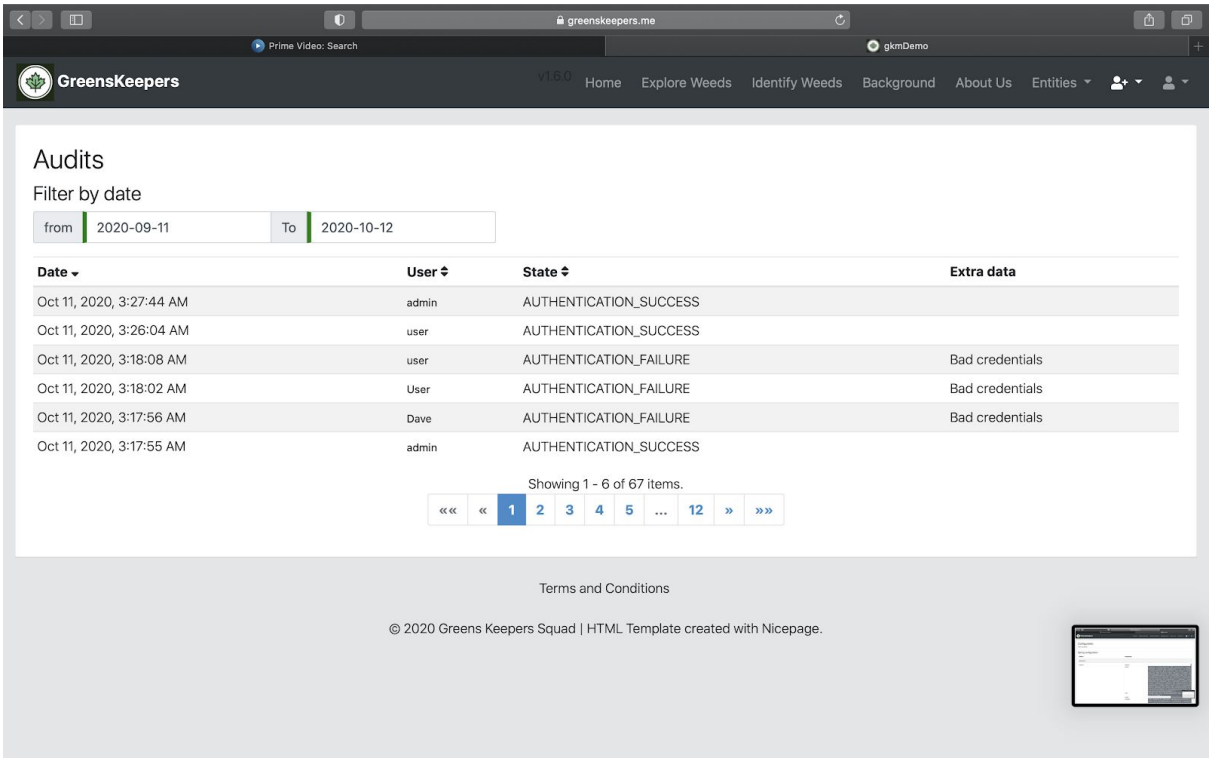
As the image shows above there are 3 extra tabs when we log in as Admin. Upon further research on these tabs images were taken to show the Admin rules , the user list along with the ability to edit the databases and audits and all logs of the website. These images can be seen below.

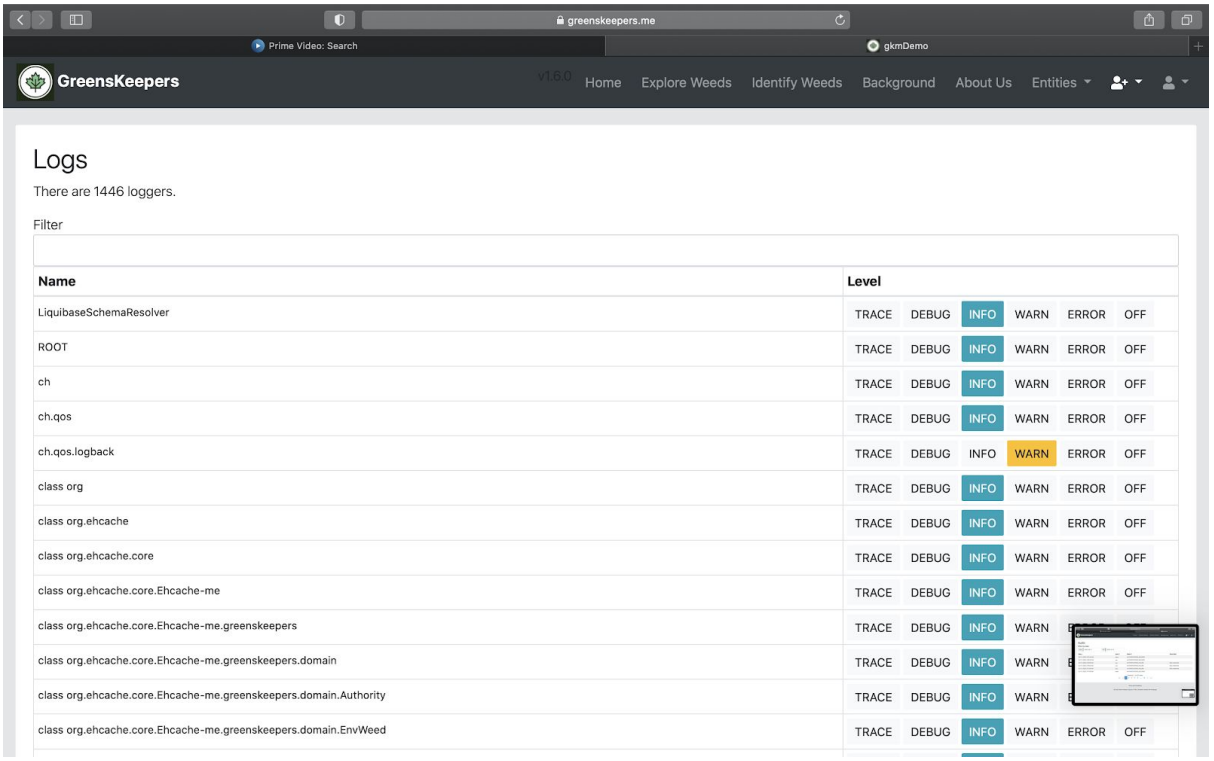



greenskeepers.me
Prime Video: Search
gkmDemo
GreensKeepers
v1.6.0
Home
Explore Weeds
Identify Weeds
Background
About Us
Entities
Users
+ Create a new User
ID
Login
Email
Profiles
Created Date
Last Modified By
Last Modified Date
1 system system@localhost Activated ROLE_USER ROLE_ADMIN system View Edit Delete
3 admin admin@localhost Activated ROLE_USER ROLE_ADMIN system View Edit Delete
4 user user@localhost Activated ROLE_USER user 11/10/20 03:07 View Edit Delete
Showing 1 - 3 of 3 items.
«« « 1 » »»
Terms and Conditions
© 2020 Greens Keepers Squad | HTML Template created with Nicepage.


greenskeepers.me
Prime Video: Search
gkmDemo
GreensKeepers
v1.6.0
Home
Explore Weeds
Identify Weeds
Background
About Us
Entities
Application Metrics
Refresh
JVM Metrics
Memory
CodeHeap 'profiled nmethods' (28M / 117M)
Committed : 29M
24%
Eden Space (15M / 66M)
Committed : 27M
22%
CodeHeap 'non-profiled nmethods' (10M / 117M)
Committed : 10M
8%
Survivor Space (0M / 8M)
Committed : 3M
3%
Compressed Class Space (13M / 1,024M)
Committed : 14M
1%
Metaspace 103M
Committed : 106M
CodeHeap 'non-nmethods' (1M / 6M)
Committed : 2M
23%
Tenured Gen (54M / 164M)
Committed : 68M
33%
Threads
Runnable 7
22%
Timed Waiting (4)
13%
Waiting (21)
66%
Blocked (0)
Total: 32
Expand
System
Uptime 2 days 14 hours 34 minutes 41 seconds
Start time Thursday, October 8, 2020 at 12:56:25 PM GMT+11:00
Process CPU usage 0.33 %
System CPU usage 0.45 %
System CPU count 1
System 1m Load average 0
Process files max 1,048,576
Process files open 214
Garbage collector statistics
GC Live Data Size/GC Max Data Size (41M / 164M)
24.94%
GC Memory Promoted/GC Memory Allocated (38M / 2,596M)
1%
Classes loaded
Classes unloaded 0


greenskeepers.me
Prime Video: Search
gkmDemo
GreensKeepers
v1.6.0
Home
Explore Weeds
Identify Weeds
Background
About Us
Entities
Health Checks
Refresh
Service Name
Status
Details
Db
UP
DiskSpace
UP
Ping
UP
Terms and Conditions
© 2020 Greens Keepers Squad | HTML Template created with Nicepage.


greenskeepers.me
Prime Video: Search
gkmDemo
GreensKeepers
v1.6.0
Home
Explore Weeds
Identify Weeds
Background
About Us
Entities
Audits
Filter by date
from
2020-09-11
To
2020-10-12
Date
User
State
Extra data
Oct 11, 2020, 3:27:44 AM
admin
AUTHENTICATION_SUCCESS
Oct 11, 2020, 3:26:04 AM
user
AUTHENTICATION_SUCCESS
Oct 11, 2020, 3:18:08 AM
user
AUTHENTICATION_FAILURE
Bad credentials
Oct 11, 2020, 3:18:02 AM
User
AUTHENTICATION_FAILURE
Bad credentials
Oct 11, 2020, 3:17:56 AM
Dave
AUTHENTICATION_FAILURE
Bad credentials
Oct 11, 2020, 3:17:55 AM
admin
AUTHENTICATION_SUCCESS
Showing 1 - 6 of 67 items.
«« « 1 2 3 4 5 ... 12 » »»
Terms and Conditions
© 2020 Greens Keepers Squad | HTML Template created with Nicepage.

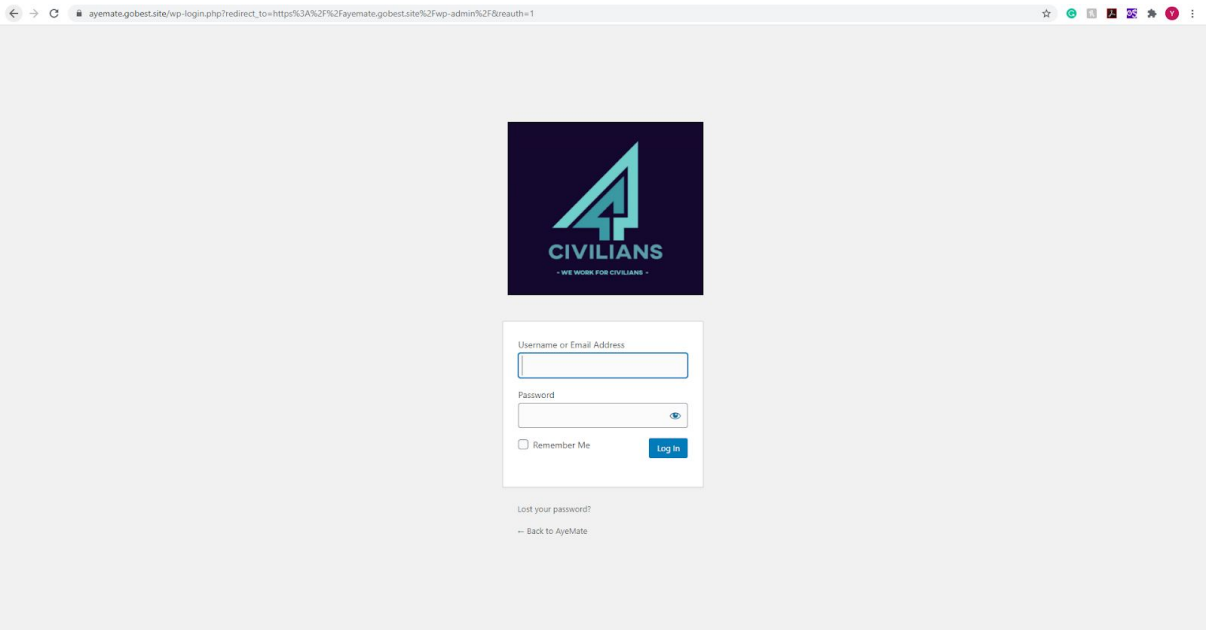


As it can be seen in the images above we can see the user list, add or delete users along with changing passwords of the admin. We can also view all the API's used and all the logs and audits carried out in the website.

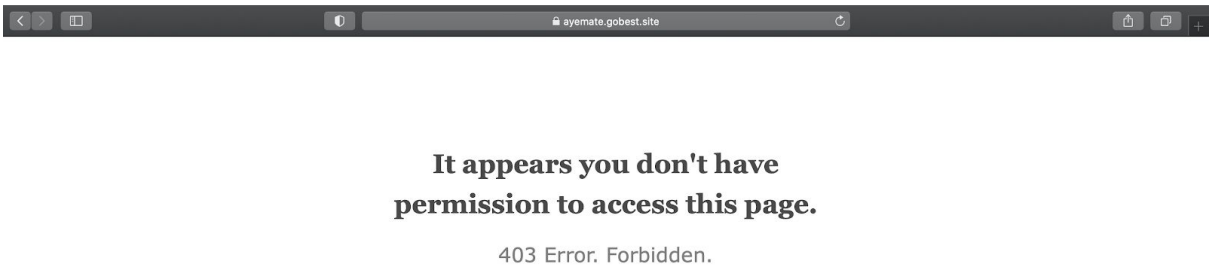
This shows how such a small vulnerability was exploited to access the website's admin login. The findings also show the human nature of trusting the systems and using easy usernames and passwords such as admin and the audits show that an IDS(Intrusion Detection System) is in place for detecting attacks, but wasn't able to prevent the attack from happening.

2) Ayemate.gobest.site

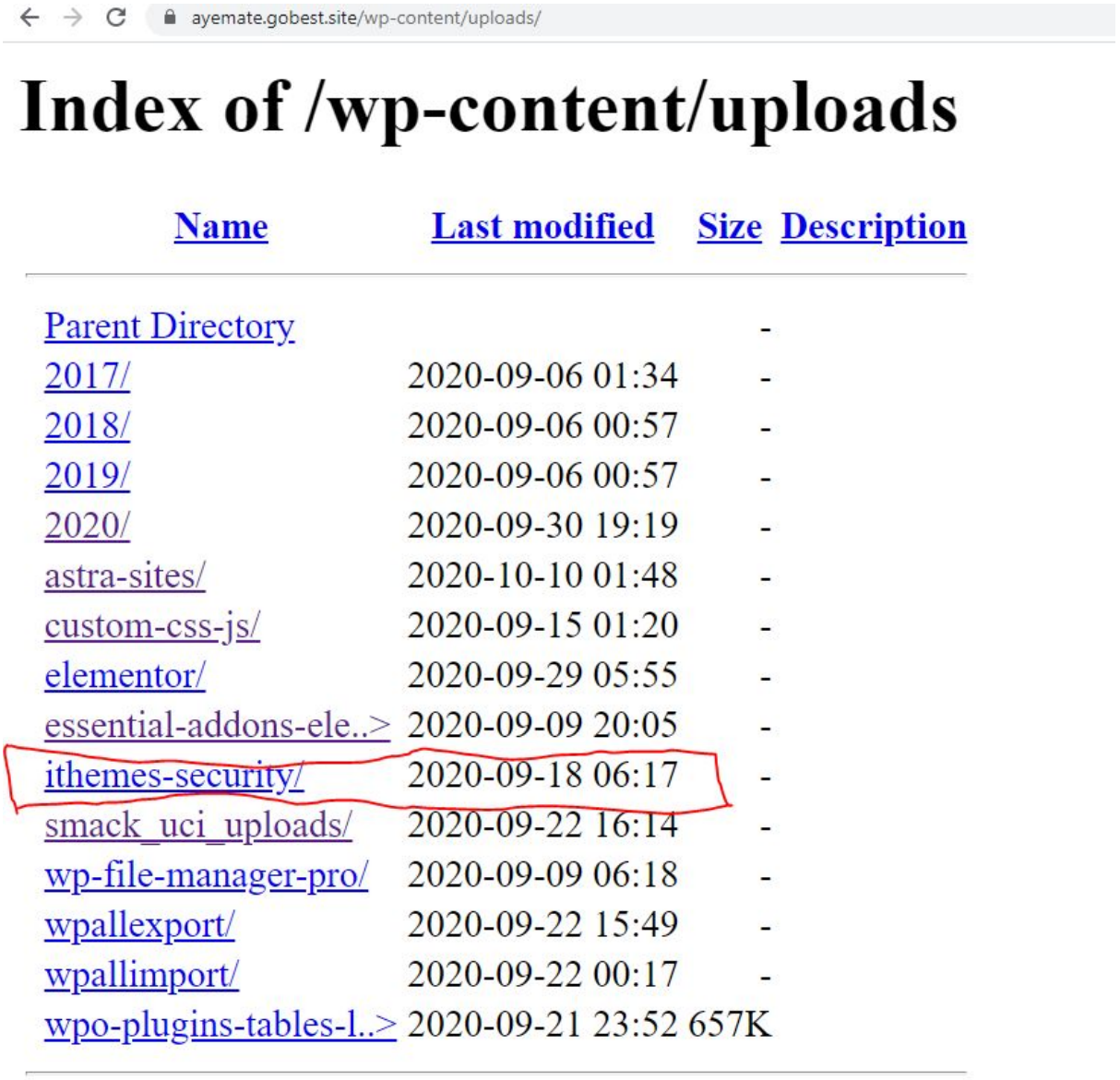
Using the vulnerability analysis we found in Part 2, we were able to see the admin login web page. This can be seen in the image below.



We tried the **username as "Admin" and password as "Admin"** and we got in the first try and were able to access the admin dashboard of the website. However, before we could take screenshots of the dashboard we were locked out of the website. Unfortunately, after multiple attempts we couldn't get access to the admin dashboard again and after a while we got the following error as seen in the image below.



We wondered why this happened and started searching for the cause to find out what we did wrong. We found out that a plugin was enabled in which was called “ithemes-security”. This can be seen in the screenshot below.



Upon research we found out that Ithemes-security is a plugin used in wordpress websites that can prevent attackers from gaining access to websites. Further research showed that the plugins also block IP addresses of systems of the attackers and notifies the owner of any attack attempts carried out. This was probably notified to the user prompting them to change the login credentials making us unable to access the dashboard again.

3) mobihelper.ga

From the attack plan in part 2B we found the username of website as wpadmin using CeWL and hydra. But we weren't successful in finding the password using wpscan from password list.

Part 4: Result Evaluation, Discussion & Reinforcement

We sent emails to greenskeepers.me and ayemate.gobest.site regarding the attacks we carried out on their website. We didn't sent an email to mobihelper.ga as our attack on their website was unsuccessful. The transcripts of the emails sent can be found below.

1)Greenskeepers.me

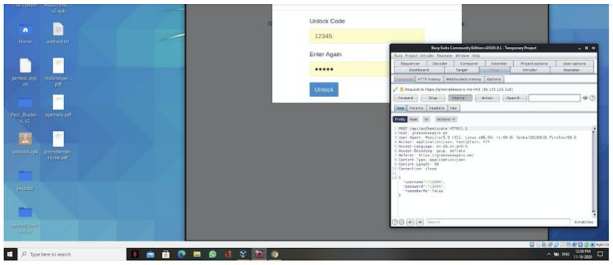
Hello Development Team of Greenkeepers,

We are NetSec Echo and we were the attack team assigned to carry out an attack on your IE project website https://greenskeepers.me. We gained administrative access to your website on the night of 11/10/2020. We exploited the vulnerabilities of the api gateways of your login page and carried out a brute force attack to gain access. On gaining access we could see your databases, your list of users, delete or add users along with giving unnecessary privilege to any user desired. We do think that the attack was caught on your system as your website did reveal some audit logs that could show the login requests and number of calls done which prompted you to change your Admin login credentials.

The vulnerabilities found and possible solutions on your website are as follows:
Your API reveals information as what data is sent to the server in JSON format. Please refer to screenshots below. Since your api gateway is open and not obfuscated it is easy to do a brute force attack on the website to get access. The api gateway reveals the following information:
"username, password, and rememberME" in json format. We were able to modify the API request to see which username and passwords would return a 200 OK status. To our surprise the login username and password was "Admin" and "admin". As a protective measure please don't use "admin" or "Admin" as your login as these are very easy for the attacker to predict and use more difficult to guess login credentials.

Since you had audits and logs generated for your website you were possibly able to detect the attack which is a good sign of an IDS(Intrusion detection system) in place which allowed you to change the passwords before we could do any changes to the website.

Your website is pretty secure otherwise and we can suggest you to add some obfuscation to your API requests preventing attacks such as the one we carried out.



We will follow up with you next week to see if you have implemented any more security measures in place. This email is to inform you that your system has some vulnerabilities and can be exploited by an adversary for gaining unwanted access to your system. We didn't change anything in your website even though we could have as we have a good 24 hours before your login credentials were changed, we didn't change anything because we know about the amount of work you have put in your website and system.

Please do not hesitate to contact us if you have any doubts about this email and need help with the security of your website.

Regards,
Yash Mestry
Akash Veerabomma

P.S: This attack carried out isn't personal and we don't have any personal vengeance with you as a team and your project. We carried out the attack as part of our assignment and in the rules and regulations given to use by Monash University

2) Ayemate.gobest.site

We are NetSec Echo and we were the attack team assigned to carry out an attack on your IE project website https://ayemate.gobest.site . We gained administrative access to your website in these past weeks when we tried to exploit some vulnerabilities in your system. We exploited the vulnerabilities of the api gateways of your admin login page and carried out a brute force attack to gain access. On gaining access we could see your databases, your list of users, delete or add users along with giving unnecessary privilege to any user desired and make any changes to the website as required. We do think that the attack was caught on your system as your website did reveal some audit logs that could show the login requests and number of calls done which prompted you to change your Admin login credentials. We were able to modify the API request to see which username and passwords would return a 200 OK status. To our surprise the login username and password was "admin" and "admin". As a protective measure please don't use "admin" or "Admin" as your login as these are very easy for the attacker to predict and use more difficult to guess login credentials. We also know that you have used word press to make a website which means it has a number of vulnerabilities. You can google about word press vulnerabilities and secure your website more.

You also had plugins installed such as **ithemes-security** which prevented us from accessing your websites again. This is a very good security method. When we did try brute forcing in your system again our IP address was blocked. This made us very happy to see your website was protected from such an attack and is quite secure. Although an attacker uses ip spoofing to stay anonymous on the internet and can mask his IP address to try and get access to your website again bypassing this security.

Since you had audits and logs generated for your website you were possibly able to detect the attack which is a good sign of an IDS(Intrusion detection system) in place which allowed you to change the passwords before we could do any changes to the website.

Your website is pretty secure otherwise and we can suggest you to add some obfuscation to your API requests preventing attacks such as the one we carried out.

We will follow up with you next week to see if you have implemented any more security measures in place. This email is to inform you that your system has some vulnerabilities and can be exploited by an adversary for gaining unwanted access to your system. We didn't change anything in your website even though we could have as we have a good 3 to 4 hours before your login credentials were changed, we didn't change anything because we know about the amount of work you have put in your website and system.

Please do not hesitate to contact us if you have any doubts about this email and need help with the security of your website.

Regards,
Yash Mestry
Akash Veerabomma

P.S: This attack carried out isn't personal and we don't have any personal vengeance with you as a team and your project. We carried out the attack as part of our assignment and in the rules and regulations given to use by Monash University

Upon sending the emails we have received responses from both the teams and we will do a followup during PART 4 of this report.

Part 4: Result Evaluation, Discussion & Reinforcement

We sent emails in the following week to follow up on any new security measures imposed by the victim teams. The responses of the teams can be seen in the screenshots below.

1) Greenskeepers.me