Updated: Sep 13, 2022

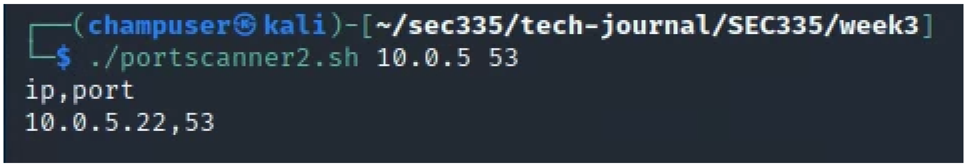
Class Activity 3.1 DNS Enumeration

DNS Enumeration using bash

> 💡DNS can be a treasure trove of information for penetration testers. Hostnames, naming conventions, hierarchical namespaces and of course IP resolution can focus your attack efforts. A misconfigured DNS server can also provide a great deal of information.

In lab 2.1 we used a technique to port scan using bash. With a partner, revise your code to repeat this scan against 10.0.5.0/24, targeting the TCP port associated with DNS. Pass a network prefix like (10.0.5) and a port (53) and scan from .1 through .254 on the network for DNS.

`Deliverable 1. Provide a screenshot of your /24 port scan against 10.0.5.0/24 similar to the one below.`



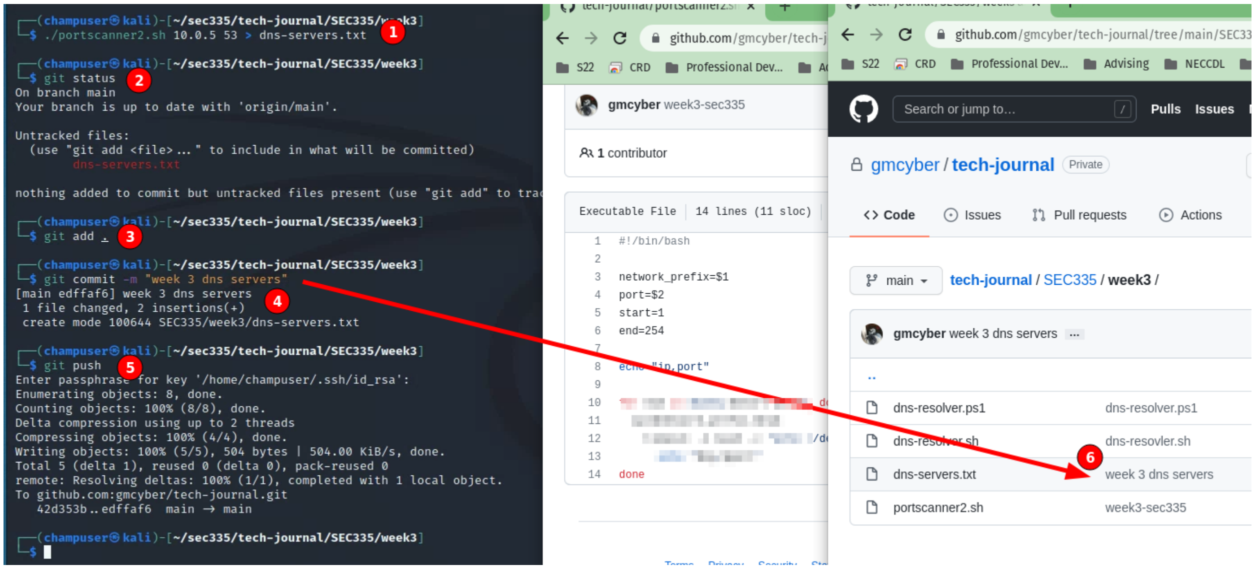
```
┌──(champuser㉿kali)-[~/sec335/tech-journal/SEC335/week3]
└─$ ./portscanner2.sh 10.0.5 53
ip,port
10.0.5.22,53
```

Organizing our Source Code

In addition to writing a lot of wiki content, we will be generating code. Organize your source code repo by:

- Cloning your repository on kali. This is done via git pull. See [Protip 2.1 - Using Git on Kali]()
- Creating an organized directory structure for code as opposed to wiki content (1)
- Saving any code written in weeks 1 and 2 and 3 to this directory structure and authoring new content within your local repository.
- Add new content to your local repo (2,3)
- commit changes locally (4)
- push changes to your github based repo(5). Here is a sample transaction.
- See your latest git submission on github (6)

Updated: Sep 13, 2022



```
┌──(champuser㉿kali)-[~/sec335/tech-journal/SEC335/week3]
└─$ ./portscanner2.sh 10.0.5 53 > dns-servers.txt

┌──(champuser㉿kali)-[~/sec335/tech-journal/SEC335/week3]
└─$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        dns-servers.txt

nothing added to commit but untracked files present (use "git add" to trac

┌──(champuser㉿kali)-[~/sec335/tech-journal/SEC335/week3]
└─$ git add .

┌──(champuser㉿kali)-[~/sec335/tech-journal/SEC335/week3]
└─$ git commit -m "week 3 dns servers"
[main edffaf6] week 3 dns servers
 1 file changed, 2 insertions(+)
 create mode 100644 SEC335/week3/dns-servers.txt

┌──(champuser㉿kali)-[~/sec335/tech-journal/SEC335/week3]
└─$ git push
Enter passphrase for key '/home/champuser/.ssh/id_rsa':
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 2 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 504 bytes | 504.00 KiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:gmcyber/tech-journal.git
   42d353b..edffaf6  main → main

┌──(champuser㉿kali)-[~/sec335/tech-journal/SEC335/week3]
└─$ 
```

gmcyber week3-sec335

1 contributor

Executable File | 14 lines (11 sloc)

```
#!/bin/bash

network_prefix=$1
port=$2
start=1
end=254

echo "ip,port"

[redacted]
done
```

gmcyber / tech-journal

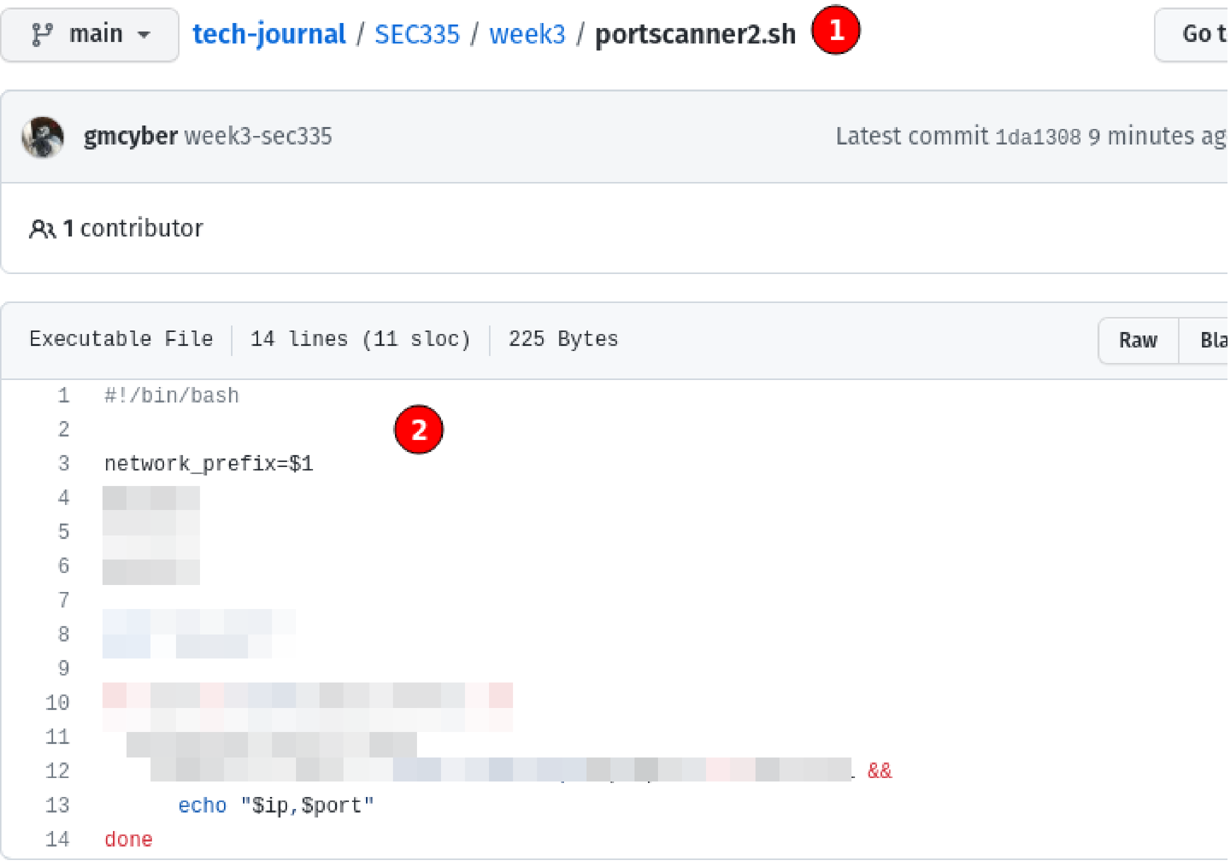
tech-journal / SEC335 / **week3** /

gmcyber week 3 dns servers

..	
dns-resolver.ps1	dns-resolver.ps1
dns-resolver.sh	dns-resovler.sh
dns-servers.txt	week 3 dns servers
portscanner2.sh	week3-sec335

Deliverable 2. Provide a screenshot similar to the one below that shows your directory structure and the source code of your /24 port scanner. Note, this code can be 1 liner, but I want you to go through the process of submitting source code to github.

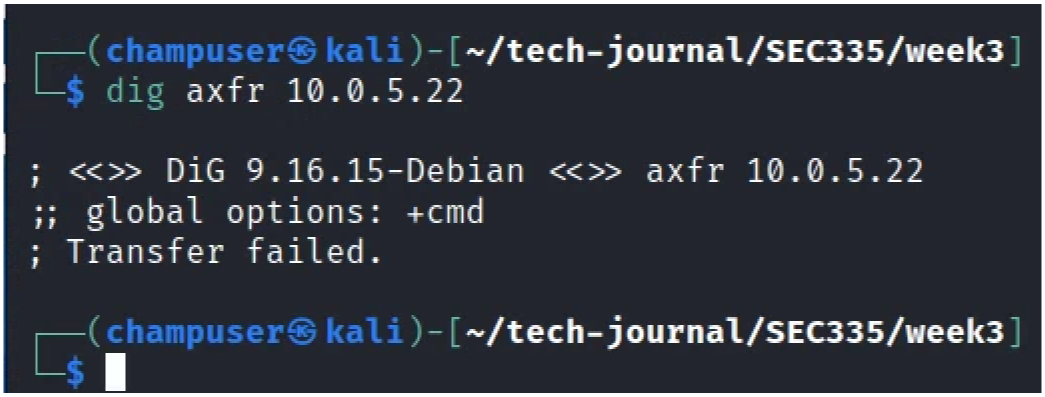
Updated: Sep 13, 2022


main
tech-journal / SEC335 / week3 / portscanner2.sh
1
Go t
gmcyber week3-sec335
Latest commit 1da1308 9 minutes ag
1 contributor
Executable File | 14 lines (11 sloc) | 225 Bytes
Raw
Bla
2

```
#!/bin/bash

network_prefix=$1
[redacted] &&
        echo "$ip,$port"
done
```

DNS Reverse Lookup

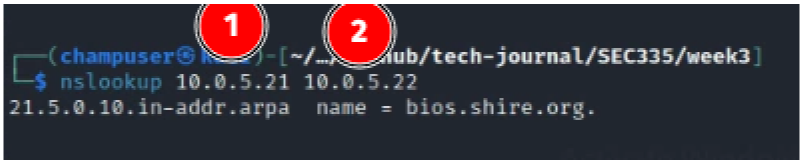
Go ahead and ignore 10.0.5.2 (the firewall if it shows up in your scan) and settle in on the one other DNS server (10.0.5.22) that you've found. We are going to attempt a zone transfer. This won't work because it is a secured DNS server.



```
┌──(champuser㉿kali)-[~/tech-journal/SEC335/week3]
└─$ dig axfr 10.0.5.22

; <<>> DiG 9.16.15-Debian <<>> axfr 10.0.5.22
;; global options: +cmd
; Transfer failed.

┌──(champuser㉿kali)-[~/tech-journal/SEC335/week3]
└─$ 
```

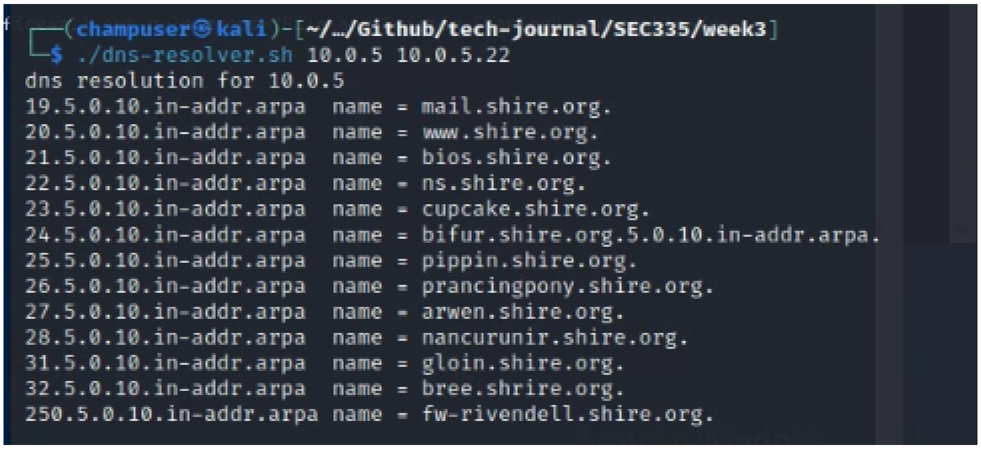
Now, let's try a reverse lookup. Take a look at the following nslookup command. We can force nslookup to use a specific dns server (2) to lookup a host (1)

Updated: Sep 13, 2022



```
┌──(champuser㉿ )-[~/…/ hub/tech-journal/SEC335/week3]
└─$ nslookup 10.0.5.21 10.0.5.22
21.5.0.10.in-addr.arpa  name = bios.shire.org.
```

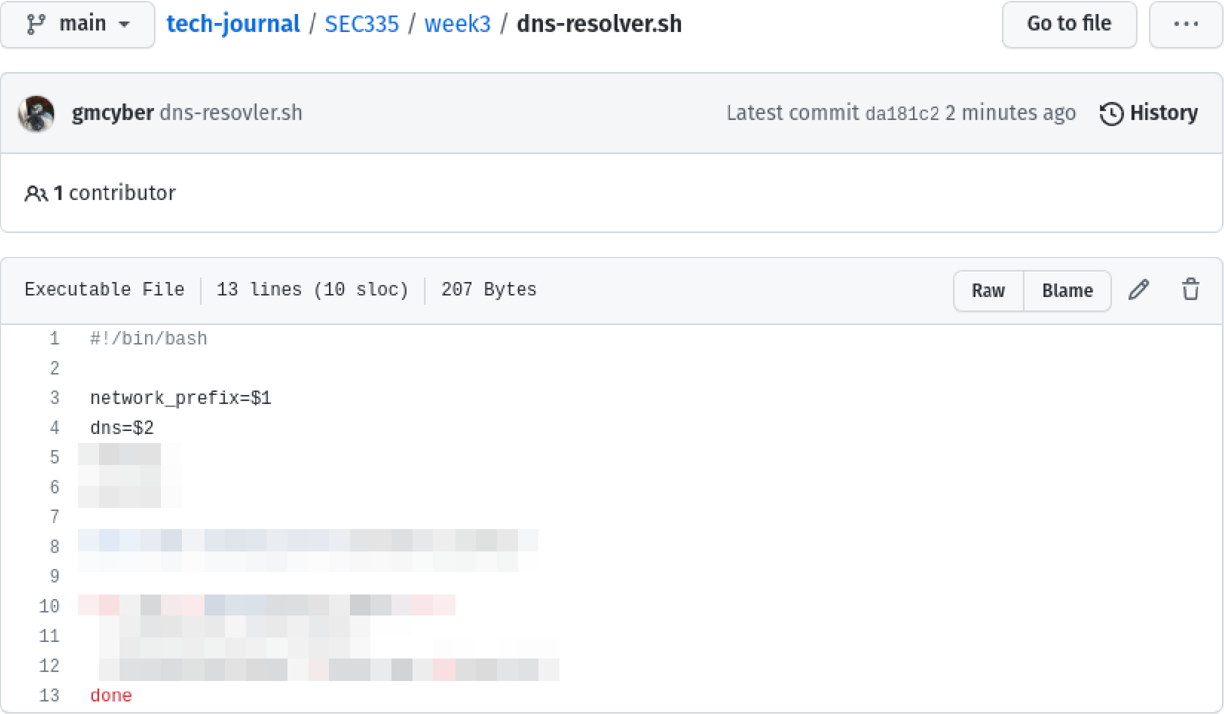
Deliverable 3. Write a script that takes a network prefix and a specific dns server in which to perform a lookup. Assume a /24 network. Provide a screenshot similar to the one below showing the program run.



```
┌──(champuser㉿kali)-[~/…/Github/tech-journal/SEC335/week3]
└─$ ./dns-resolver.sh 10.0.5 10.0.5.22
dns resolution for 10.0.5
19.5.0.10.in-addr.arpa  name = mail.shire.org.
20.5.0.10.in-addr.arpa  name = www.shire.org.
21.5.0.10.in-addr.arpa  name = bios.shire.org.
22.5.0.10.in-addr.arpa  name = ns.shire.org.
23.5.0.10.in-addr.arpa  name = cupcake.shire.org.
24.5.0.10.in-addr.arpa  name = bifur.shire.org.5.0.10.in-addr.arpa.
25.5.0.10.in-addr.arpa  name = pippin.shire.org.
26.5.0.10.in-addr.arpa  name = prancingpony.shire.org.
27.5.0.10.in-addr.arpa  name = arwen.shire.org.
28.5.0.10.in-addr.arpa  name = nancurunir.shire.org.
31.5.0.10.in-addr.arpa  name = gloin.shire.org.
32.5.0.10.in-addr.arpa  name = bree.shrire.org.
250.5.0.10.in-addr.arpa name = fw-rivendell.shire.org.
```

Deliverable 4. Provide a screenshot similar to the one below that shows your directory structure and the source code of your dns resolver.

Updated: Sep 13, 2022



```
#!/bin/bash

network_prefix=$1
dns=$2
[redacted]

[redacted]
done
```

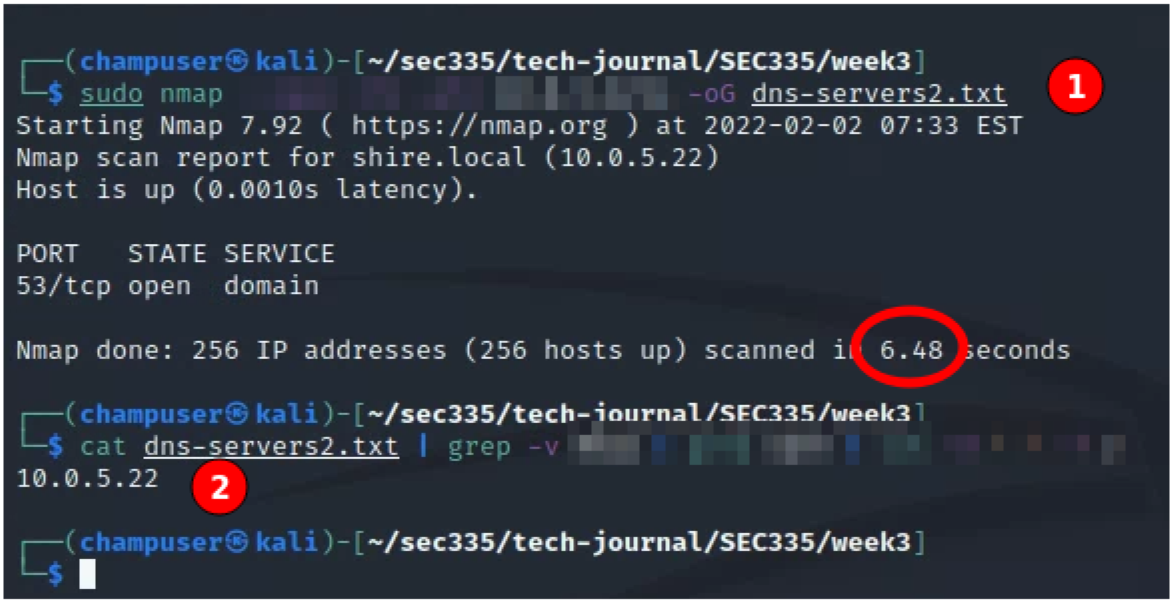
Updated: Sep 13, 2022

using nmap

Deliverable 5. Use nmap to find your DNS servers. Figure out how to:

- skip host discovery
- use a grepable output to send results to dns-servers2.txt
- only scan for a single tcp port across 10.0.5.0/24
- only report "open" ports
- see if you can use a bash 1 or 2 liner to list the unique IP addresses that respond to DNS lookups.

Provide a screenshot similar to the one below that shows the nmap run and output as well as the parsing of dns-servers2.txt

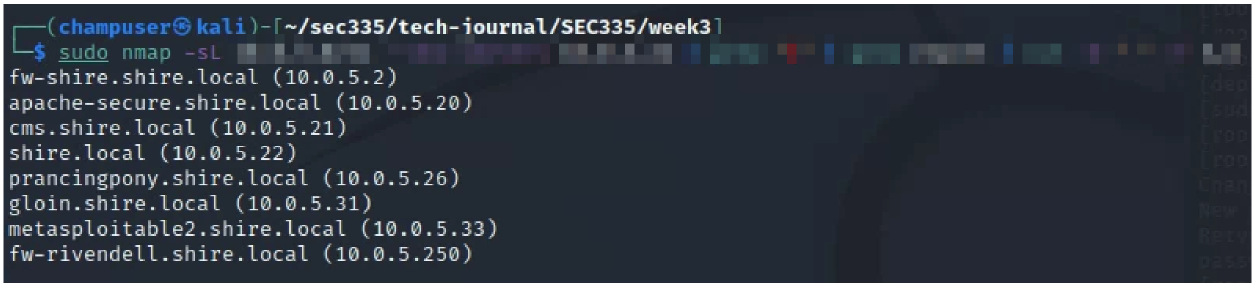


```
┌──(champuser㉿kali)-[~/sec335/tech-journal/SEC335/week3]
└─$ sudo nmap [redacted] -oG dns-servers2.txt
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-02 07:33 EST
Nmap scan report for shire.local (10.0.5.22)
Host is up (0.0010s latency).

PORT   STATE SERVICE
53/tcp open  domain

Nmap done: 256 IP addresses (256 hosts up) scanned in 6.48 seconds

┌──(champuser㉿kali)-[~/sec335/tech-journal/SEC335/week3]
└─$ cat dns-servers2.txt | grep -v [redacted]
10.0.5.22

┌──(champuser㉿kali)-[~/sec335/tech-journal/SEC335/week3]
└─$ 
```

Updated: Sep 13, 2022

reverse lookup with nmap

Deliverable 6. The following nmap command will use -sL (list targets) while specifying a dns server. See if you can do some magic with grep and cut or awk to produce output similar to the one below. Provide a screenshot showing your modified nmap run. Note, you may have different hosts listed as our target environment changes and grows over time.



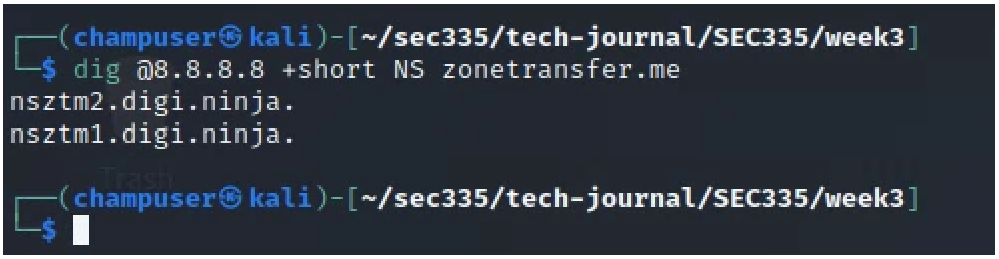
```
┌──(champuser㉿kali)-[~/sec335/tech-journal/SEC335/week3]
└─$ sudo nmap -sL [redacted]
fw-shire.shire.local (10.0.5.2)
apache-secure.shire.local (10.0.5.20)
cms.shire.local (10.0.5.21)
shire.local (10.0.5.22)
prancingpony.shire.local (10.0.5.26)
gloin.shire.local (10.0.5.31)
metasploitable2.shire.local (10.0.5.33)
fw-rivendell.shire.local (10.0.5.250)
```

zone transfer

Refer to the following site: https://digi.ninja/projects/zonetransferme.php. The documentation is simply outstanding.

This security researcher has kindly set up a weak dns server that allows zone transfer. We will use the following commands to see what a successful zone transfer looks like.

find the name servers.

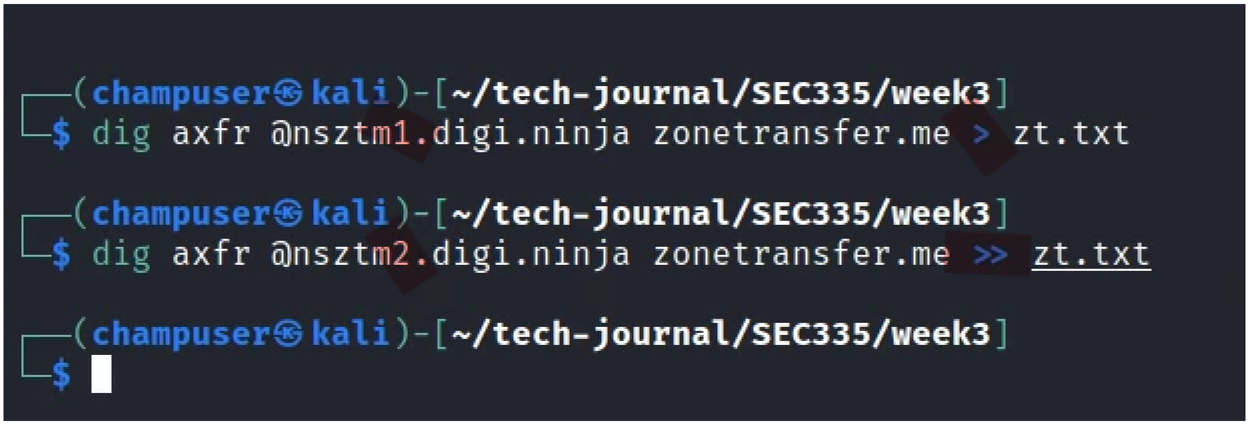


```
┌──(champuser㉿kali)-[~/sec335/tech-journal/SEC335/week3]
└─$ dig @8.8.8.8 +short NS zonetransfer.me
nsztm2.digi.ninja.
nsztm1.digi.ninja.

┌──(champuser㉿kali)-[~/sec335/tech-journal/SEC335/week3]
└─$ 
```

💡Note, in some of the later versions of Kali, you need to hunt down the name servers within very verbose output.

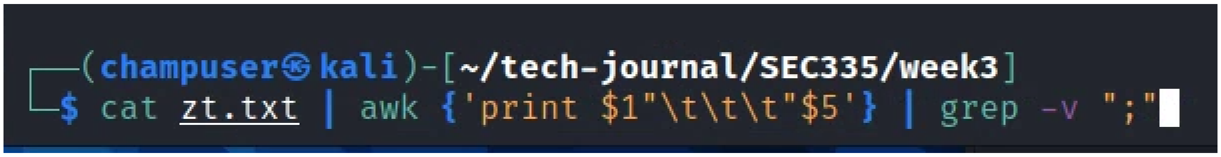
Updated: Sep 13, 2022

attempt the zone transfer



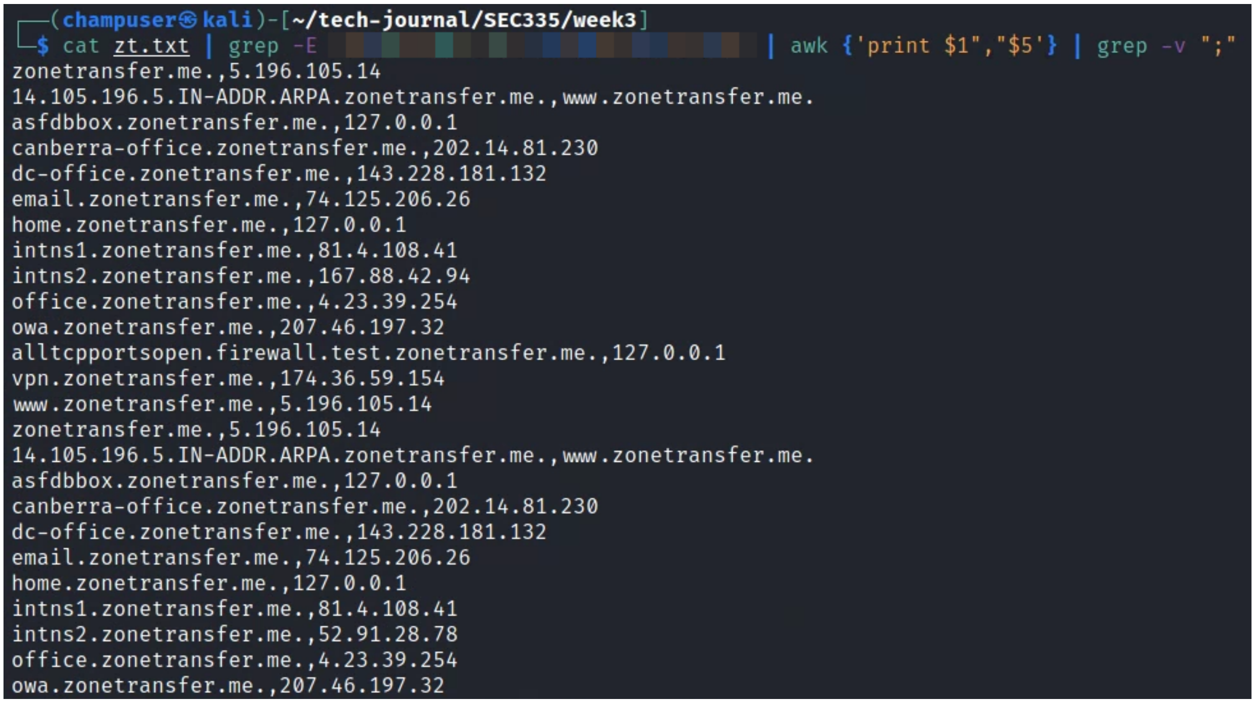
```
┌──(champuser㉿kali)-[~/tech-journal/SEC335/week3]
└─$ dig axfr @nsztm1.digi.ninja zonetransfer.me > zt.txt

┌──(champuser㉿kali)-[~/tech-journal/SEC335/week3]
└─$ dig axfr @nsztm2.digi.ninja zonetransfer.me >> zt.txt

┌──(champuser㉿kali)-[~/tech-journal/SEC335/week3]
└─$ 
```

some parsing of zt.txt



```
┌──(champuser㉿kali)-[~/tech-journal/SEC335/week3]
└─$ cat zt.txt | awk {'print $1"\t\t\t"$5'} | grep -v ";"
```

Deliverable 7. zt.txt should have some useful information, see what you can do to parse it in a manner that we have a hostname and associated ip address. Provide a screenshot similar to the one below. Note, the screenshot below is not quite perfect as not every host has an IP address.



```
┌──(champuser㉿kali)-[~/tech-journal/SEC335/week3]
└─$ cat zt.txt | grep -E [redacted] | awk {'print $1","$5'} | grep -v ";"
zonetransfer.me.,5.196.105.14
14.105.196.5.IN-ADDR.ARPA.zonetransfer.me.,www.zonetransfer.me.
asfdbbox.zonetransfer.me.,127.0.0.1
canberra-office.zonetransfer.me.,202.14.81.230
dc-office.zonetransfer.me.,143.228.181.132
email.zonetransfer.me.,74.125.206.26
home.zonetransfer.me.,127.0.0.1
intns1.zonetransfer.me.,81.4.108.41
intns2.zonetransfer.me.,167.88.42.94
office.zonetransfer.me.,4.23.39.254
owa.zonetransfer.me.,207.46.197.32
alltcpportsopen.firewall.test.zonetransfer.me.,127.0.0.1
vpn.zonetransfer.me.,174.36.59.154
www.zonetransfer.me.,5.196.105.14
zonetransfer.me.,5.196.105.14
14.105.196.5.IN-ADDR.ARPA.zonetransfer.me.,www.zonetransfer.me.
asfdbbox.zonetransfer.me.,127.0.0.1
canberra-office.zonetransfer.me.,202.14.81.230
dc-office.zonetransfer.me.,143.228.181.132
email.zonetransfer.me.,74.125.206.26
home.zonetransfer.me.,127.0.0.1
intns1.zonetransfer.me.,81.4.108.41
intns2.zonetransfer.me.,52.91.28.78
office.zonetransfer.me.,4.23.39.254
owa.zonetransfer.me.,207.46.197.32
```

Updated: Sep 13, 2022

Deliverable 8. You've already uploaded source code, make sure that you create a page on dns enumeration and link to your uploaded shell scripts and also create content on the other interactive commands you used in this lab. Provide a link to this content page.

Deliverable 9. Your chronological reflection should capture what you learned and wrote during this activity (you can link to things like your source code and your technical articles). Provide a link to your reflection entry.