Orchestrating Resilient Red Team Operations

Protecting yourselves and your clients

Yiannis Ioannides

About me

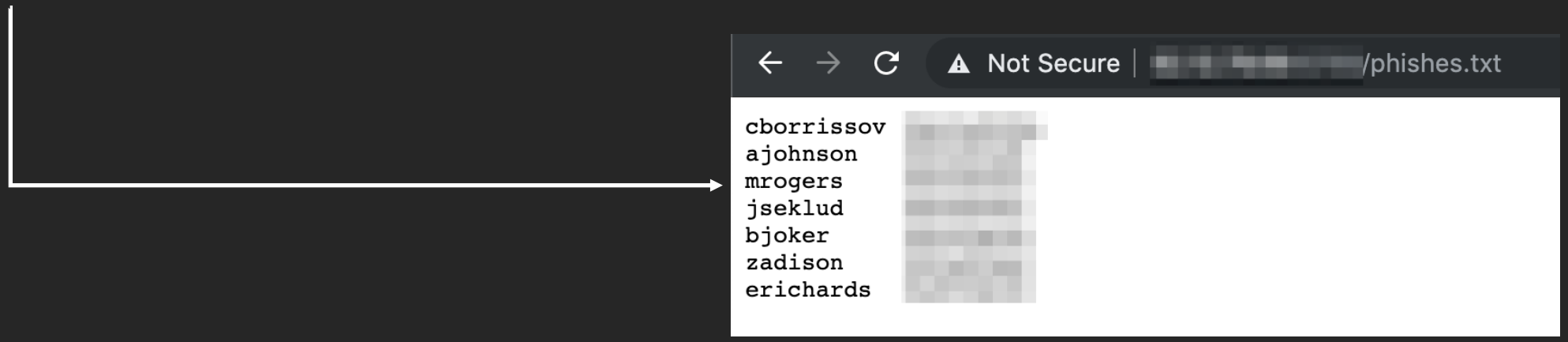
- Cyber Risk Services Director @ Deloitte
- Anything from:
  - Social Engineering
  - Physical Assessments
  - Penetration Testing
  - Incident Response / Forensics
  - ….
- @sec_groundzero
- Blog at http://offensiveops.io
- Previous talks @ Blackhat USA / EU, SECURE, ISACA…

Disclaimer

- This talk is not the “best practices”, manual or even the best way to operate
- Not a comprehensive list of techniques - Not going to cover:
  - Recon
  - Malleable C2 profiles
  - Cobalt Strike’s External C2 functionality – C3
  - Domain fronting
  - Imagination!
- For our engagements we augment these techniques to fit the operation and meet the operational goals
- Standing on the shoulders of giants
  - Tim MalcomVetter @malcomvetter
  - Jeff Dimmock - @bluscreenofjeff
  - Raphael Mudge - @armitagehacker
  - …

Why this talk?

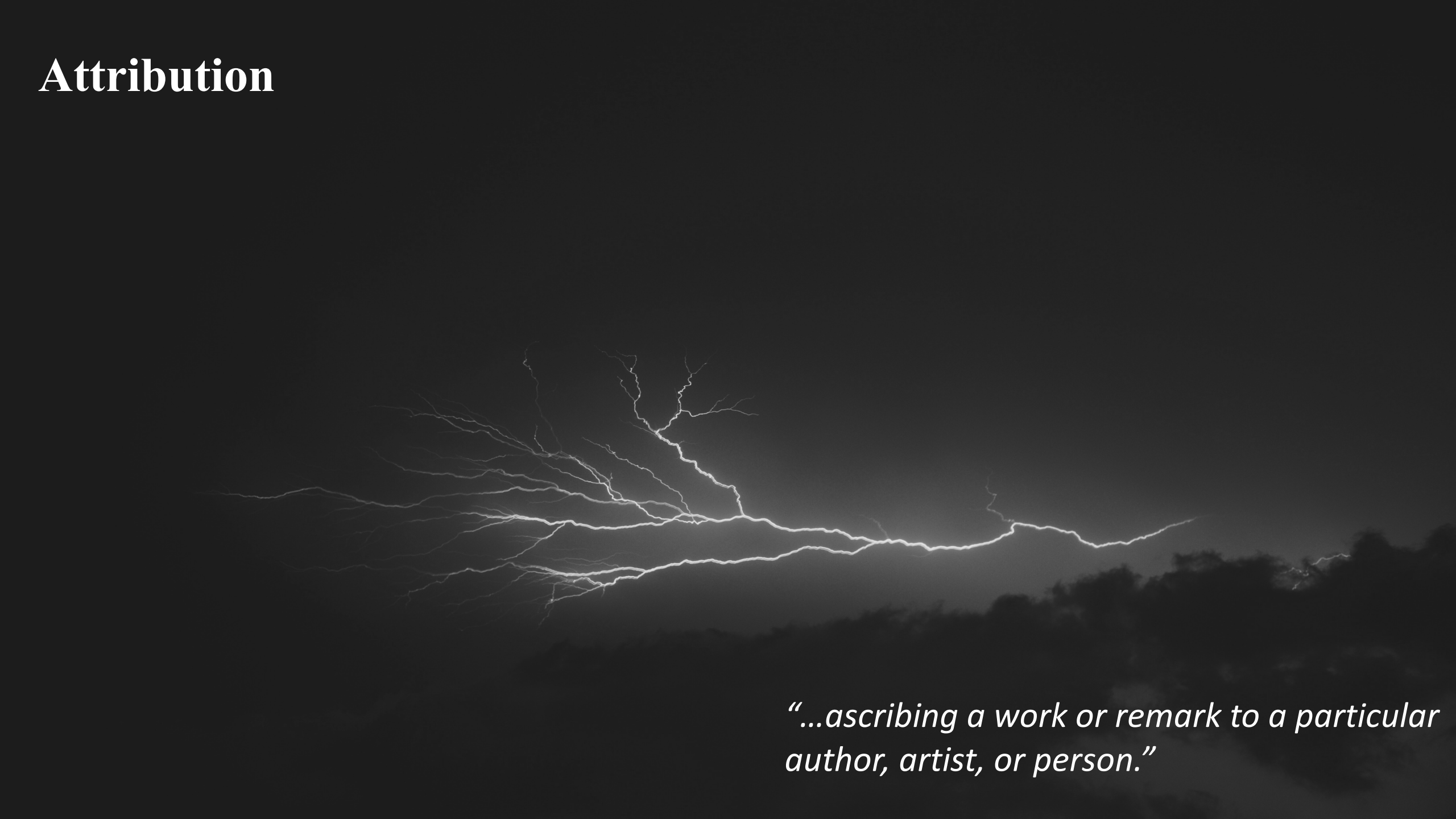
- OFFSEC is evolving daily - power to the masses
  - *git clone <insert tool name>*
  - *python <insert tool name> -t <insert target IP>*
- OPSEC not discussed as a prerequisite to using the TTPs
  - What are we exposing from our side
- OPSEC not discussed as a prerequisite to protect our clients
  - What are we exposing from the client side?
  - How are we protecting our clients?
- Clients engage us to better their defenses not expose them – measure and improve detection coverage
- We are not threat actors - we borrow and mimic some of the TTPs used by threat actors
- Sh*t can hit the fan quite easily



Red Team Infrastructure

- Designing the infrastructure can be challenging
- Varies from client to client but the foundation is similar
- Lots of moving parts and lots of building blocks that need to play together well
- Blocks will fail as defenders make moves against us
- When a block fails another one exists to support the structure (resilience)
- Ability to react to frictions – unknown / unknowns
- Aims to keeps the operation ongoing as much as possible
- Complex to the point that fits the client needs and purpose of the exercise
- A well thought infrastructure will protect both the offensive team and the client

Attribution



“…ascribing a work or remark to a particular author, artist, or person.”


…Protect our clients

Red Team Infrastructure – Operating Models

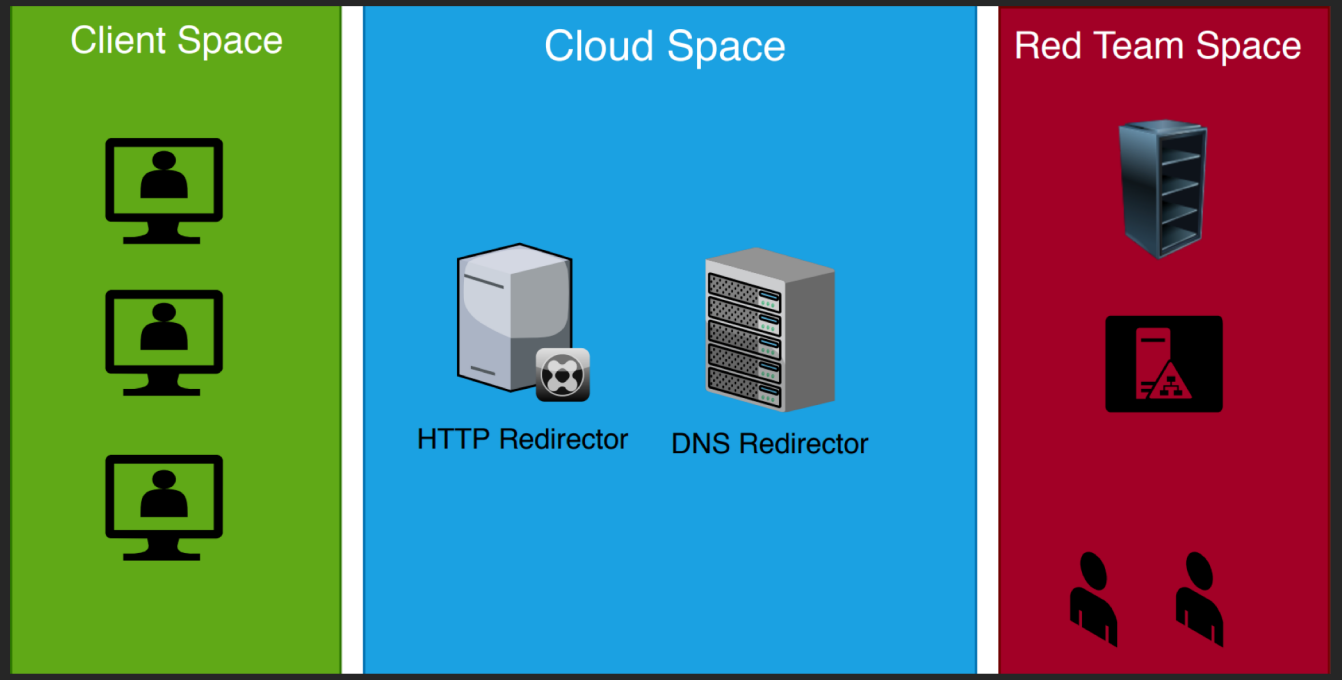
Cloud

Self-hosted

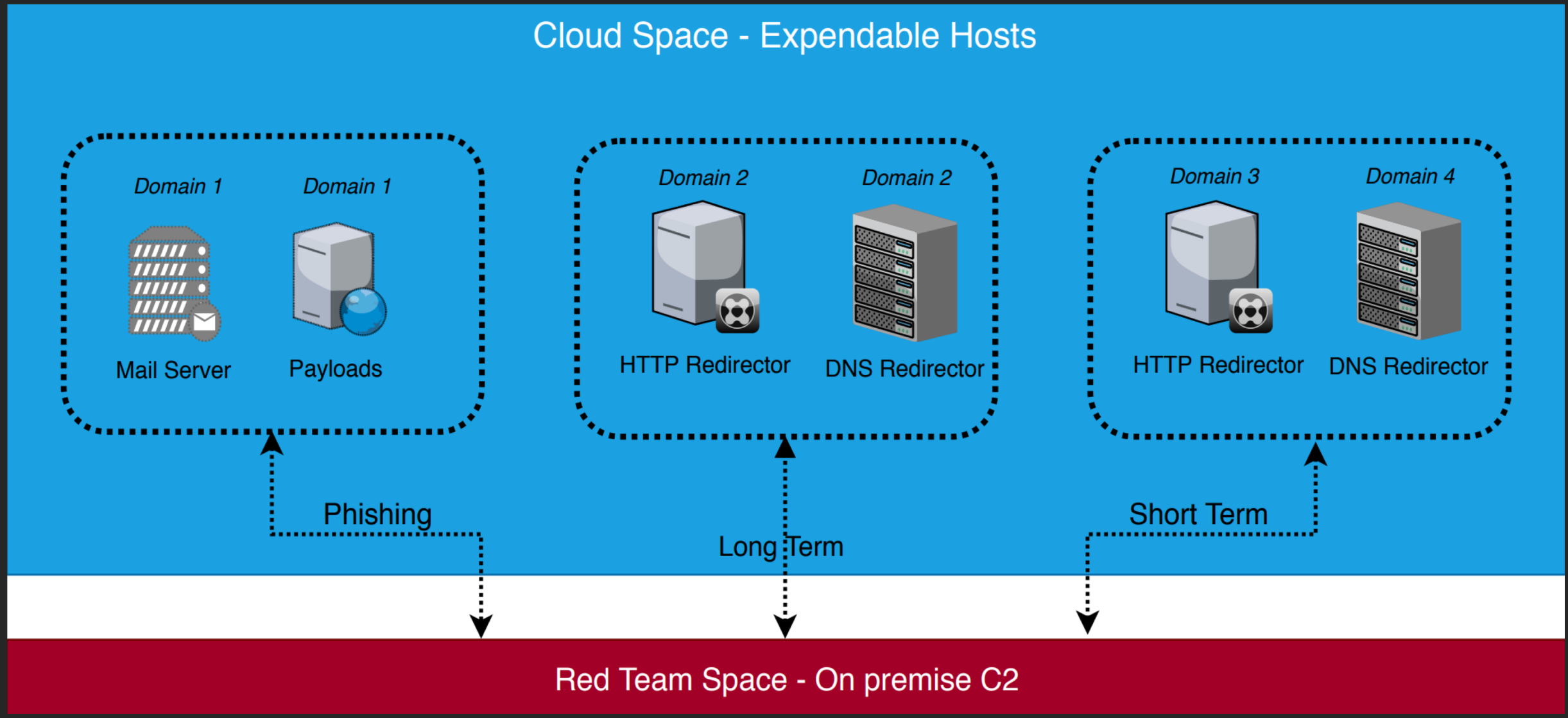
Hybrid

Redirectors

- Filtering / traffic redirection
- Cloud servers running nothing but traffic redirecting and traffic filtering commands
- Located in-between the attacking team and the client or in-between other redirectors
- Expendable, in case they are burned - Blue team blocks the IP
- Distributed across multiple cloud providers - Blue team blocks the IP space
- Can be chained as needed to include more volatile redirectors to the front
- Protect from exposing the C2 which will end the exercise or potentially expose the client

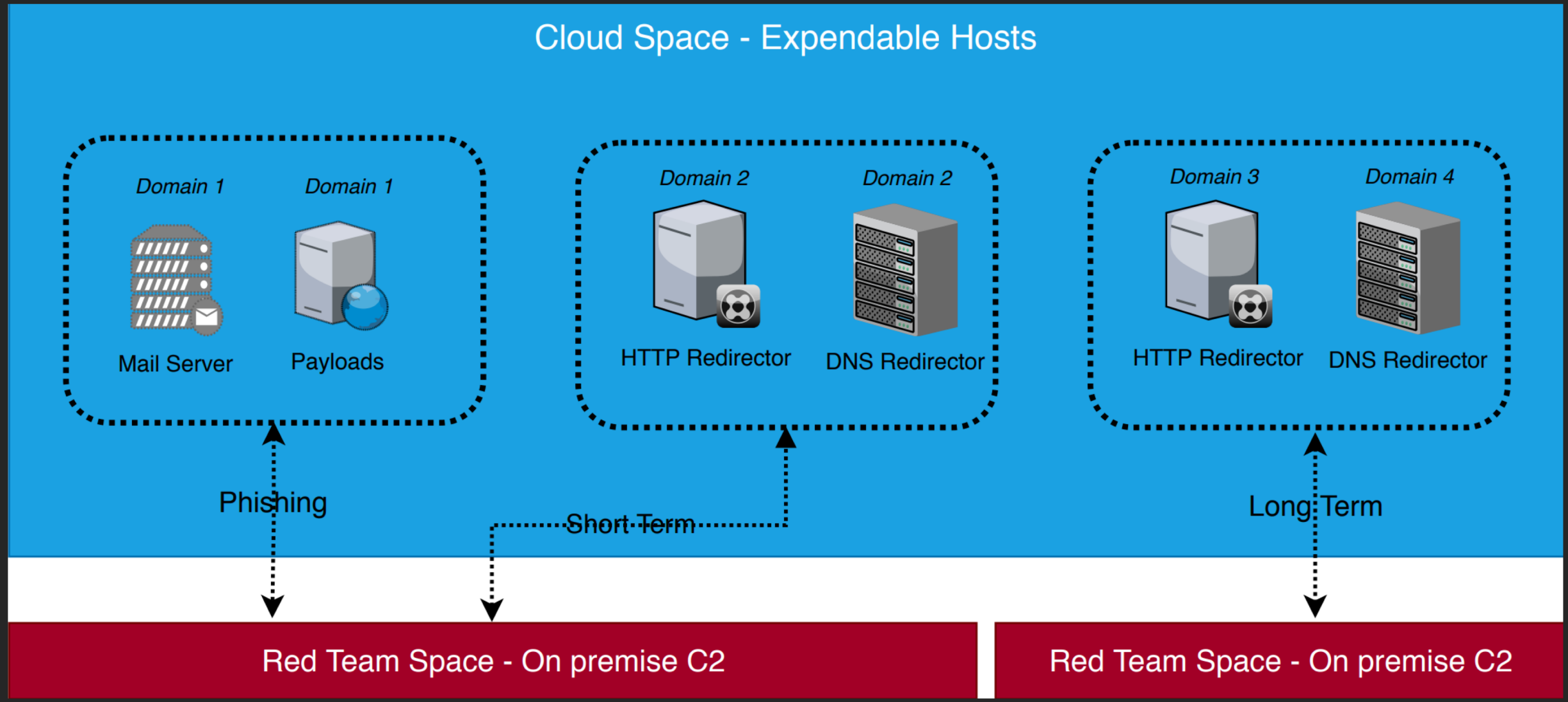


Red Team Infrastructure Distribution



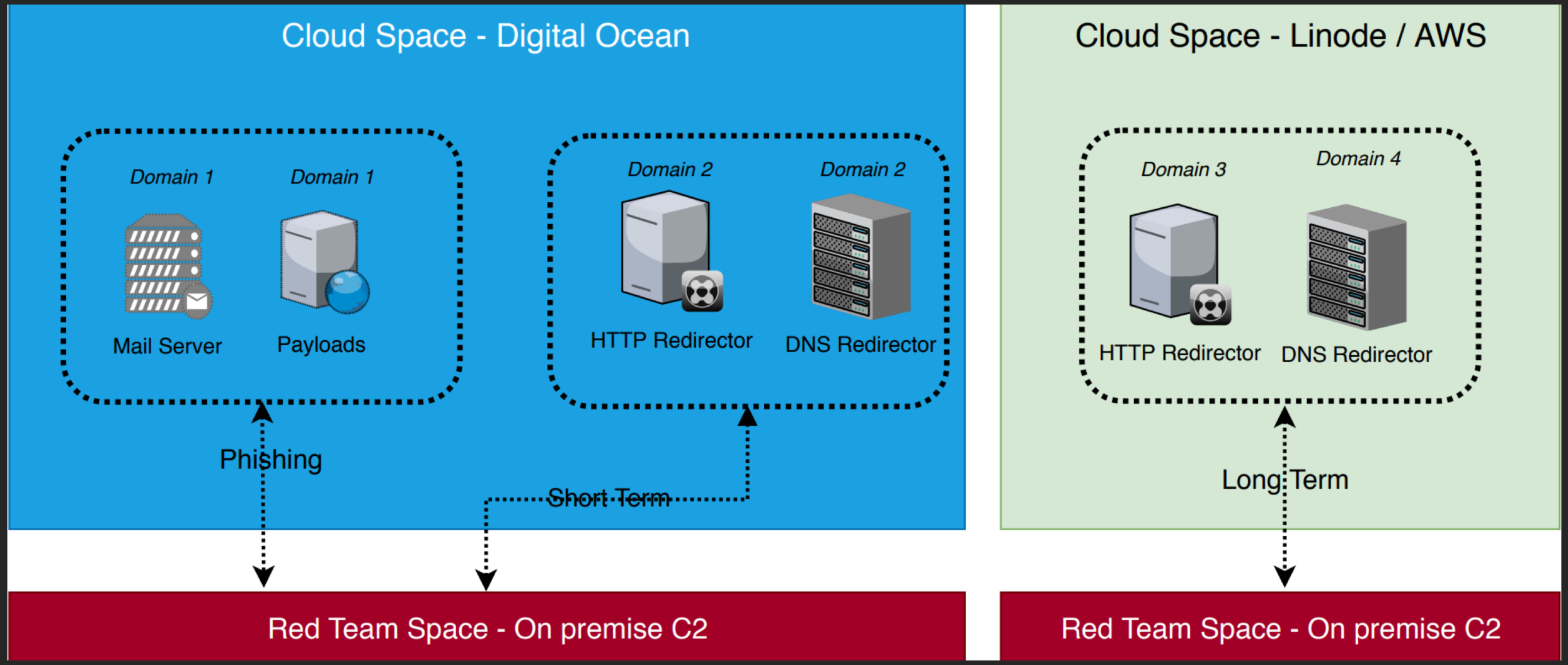
Single cloud provider – Single C2 instance

Red Team Infrastructure Distribution



Single cloud provider – Segregated C2 instances

Red Team Infrastructure Distribution

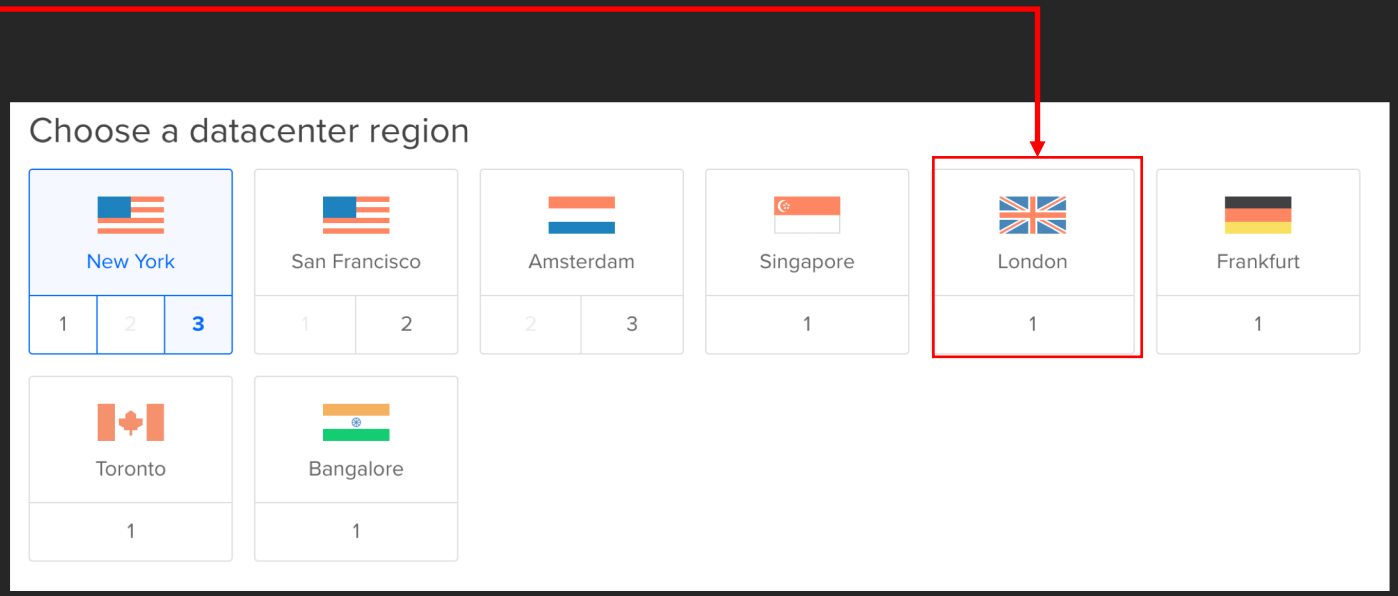


Segregated cloud providers – Segregated C2 instances

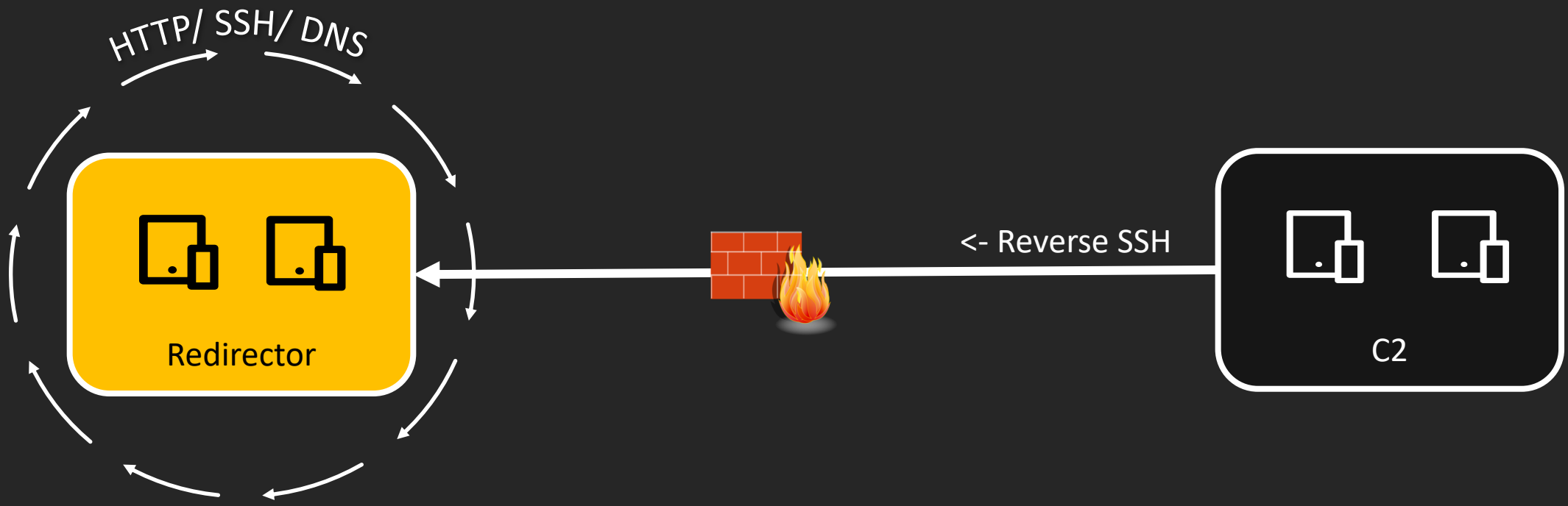
Geography

- Minimal recon can help bypass first inspections
- Identify the region the client is working in
- Incoming and outgoing traffic will be at geographic locations as the ones used by the target
- This will only assist in bypassing manual controls by people
- Blend in with similar traffic

```
┌─Yiannis at Yiannis-Mac in /Users/Yiannis
└─λ whois offensiveops.io
% IANA WHOIS server
% for more information on IANA, visit http://www.iana.org
% This query returned 1 object

refer:        whois.nic.io

contact:      technical
name:         Administrator
organisation: Internet Computer Bureau Ltd
address:      Greytown House, 221-227 High Street
address:      Orpington
address:      Kent
address:      BR6 0NZ
address:      United Kingdom
phone:        +44 (0)1689 827505
fax-no:       +44 (0)1689 831478
```



Backend Communications



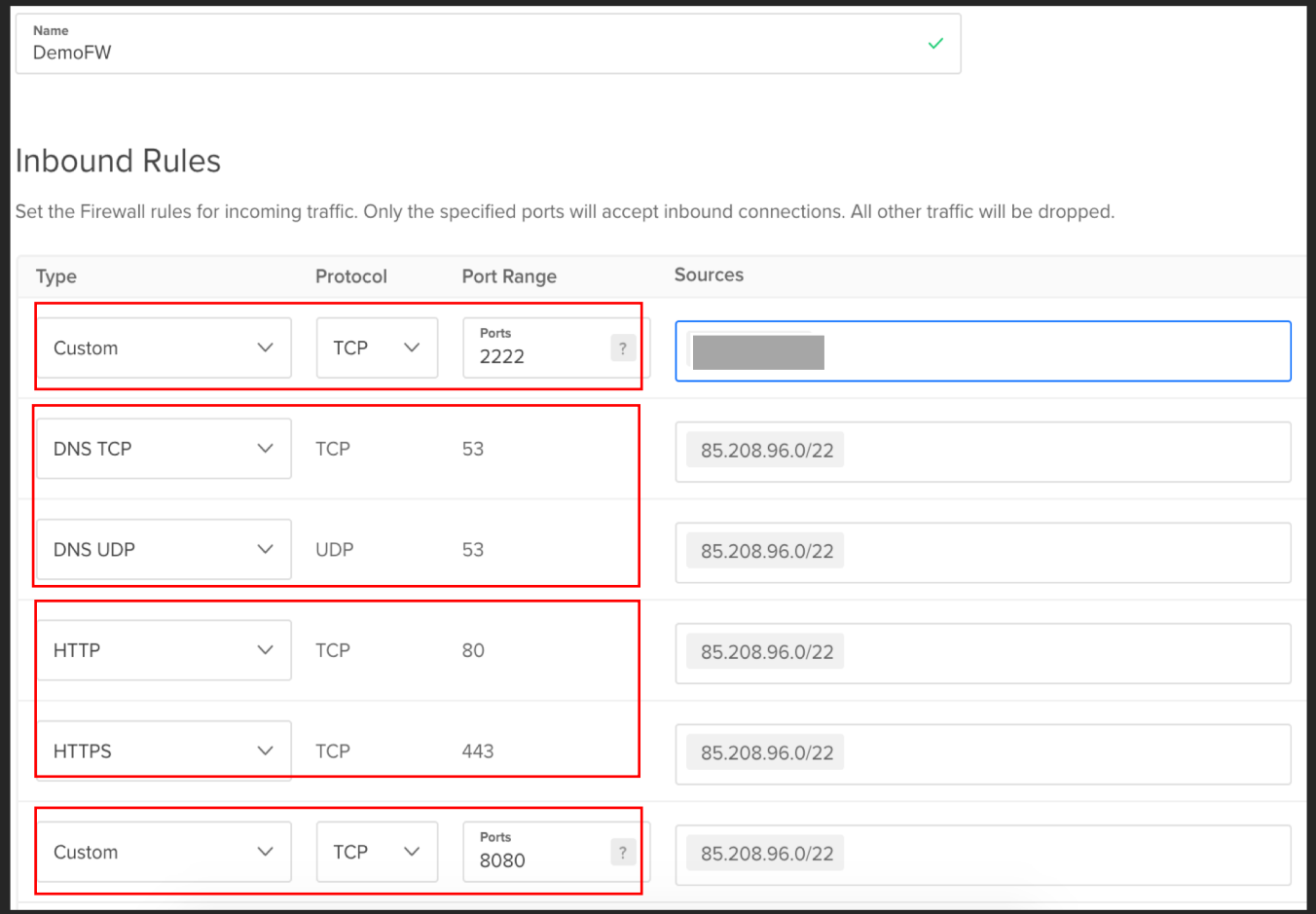
socat TCP4-LISTEN:443,fork TCP4:localhost:2222

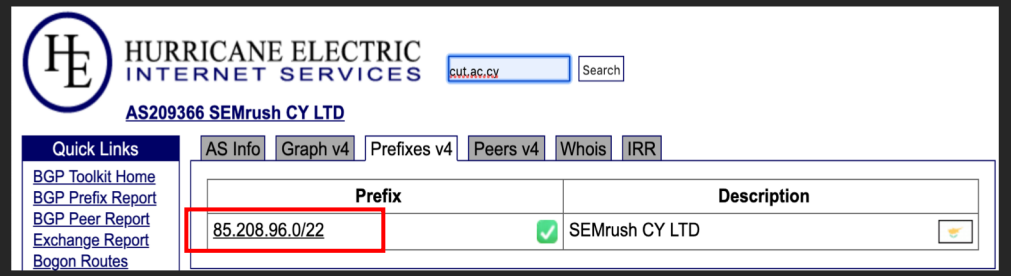
autossh -M 0 user@<CLOUD IP> -p 22 -N -R 2222:127.0.0.1:443

https://medium.com/@malcomvetter/safe-red-team-infrastructure-c5d6a0f13fac

Cloud Servers Hardening

- i.e who and how can reach our infrastructure
- What kind of traffic do we expect





If recon led us to the IP space of the client there is no need to allow any other traffic to reach our infrastructure.

Custom rule for changing the default SSH port. Only the backend server IP can access the server via SSH

Default rules to allow for our DNS and HTTP/s payloads and traffic. Only traffic originating from the target IP space can reach these ports.

Custom rule for payload delivery on a non-standard port. Only traffic originating from the target IP space can reach this port.

SSH Hardening

- Protecting against bruteforcing and password guessing attacks

```
Port 2222

# Authentication:

#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10


# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication no
#PermitEmptyPasswords no
```

- Change the default SSH port to match the firewall rule.
- Disable the root user access. Instead log in as regular user and use *sudo* to run commands
- Disable log in via a password. Instead a certificate key pair to be used.

Redirector Servers Hardening - iptables

The same can be achieved with iptables

- Allow only a specific IP to SSH to the redirector
- Allow only the IP space of the target to communicate with the redirector ports

Rule 1

```
iptables –A INPUT -p tcp -m tcp --dport 22 -s 65.7x.xx.xx -j ACCEPT
iptables –A INPUT -p tcp -m multiport --dports 53,80,443,8080 -s 85.208.96.0/22  -j ACCEPT
iptables –A INPUT -p udp -m udp --dport 53 -s 85.208.96.0/22  -j ACCEPT
```

* Assuming the C2 is directly accessible from the redirector
*** Assuming the default INPUT policy is set to DROP*

Rule 1 Breakdown

a) Accept incoming traffic to TCP port 22 from source IP 65.7x.xx.xx
b) Accept incoming traffic from the specified TCP ports list and source IP range 65.7x.xx.xx/24
c) Accept incoming traffic to UDP port 53 and source IP range 85.208.96.0/22

C2 Hardening - iptables

- Since we want only the cloud servers to talk to the C2 we can limit access to:
  - Specific IPs
  - Specific Ports
- All other traffic will be dropped

Rule 1

```
iptables –A INPUT -p tcp -m multiport --dports 80,443,8080 –s <Cloud Server IP>  -j ACCEPT
iptables –A INPUT -p udp -m udp --dport 53 –s <Cloud Server IP>  -j ACCEPT
```

* Assuming the C2 is directly accessible from the redirector
*** Assuming the default INPUT policy is set to DROP*

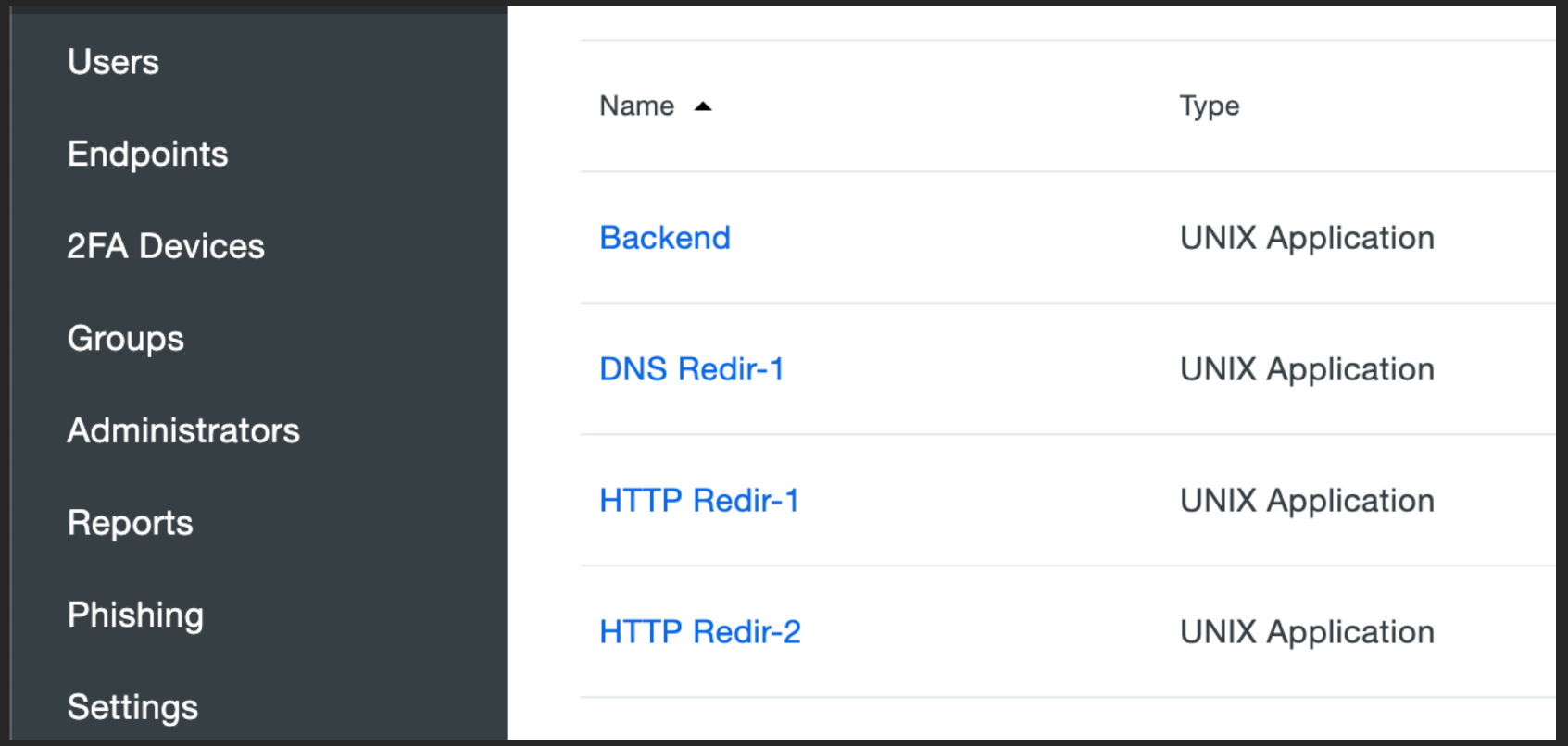
Rule 1 Breakdown

a) Accept incoming traffic from the specified TCP ports list and source the IP of the cloud redirector
b) Accept incoming traffic to UDP port 53 and source the IP of the cloud redirector

Server Access

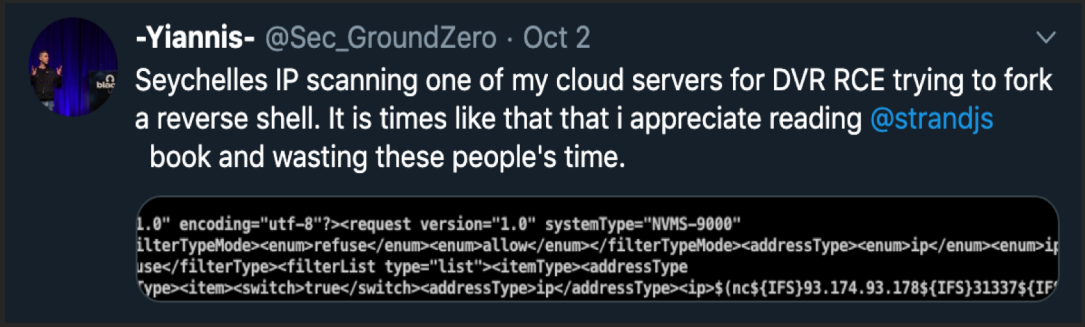
Primary method of access is SSH

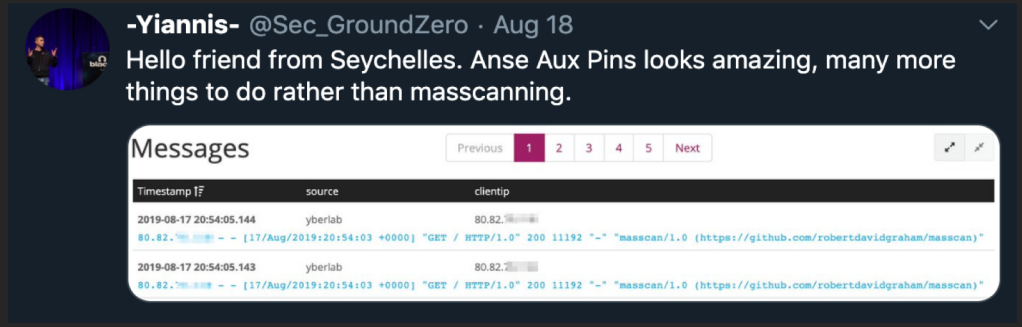
- Running on non-default ports
- Disabled root login
- Certificate based authentication
- MFA



Infrastructure Monitoring

- Monitor who is attempting to access the infrastructure.
  - Adjust firewall rules, migrate or decommission a server depending on the situation
- Multiple ways to monitor SSH logs for login failures or worst unwanted log ins.
  - /var/log/auth.log
    - Log in failures and successes
  - /var/log/fail2ban.log
    - Banned IP addresses
- Apache Web Hits
- C2 hits – Payload delivery etc
  - https://github.com/bluscreenofjeff/AggressorScripts/blob/master/apache-style-weblog-output.cna





Visualizing Logs

HTTP REDIR Web Requests / Source IP

Value	%	Count
Top 10 values		
/index.html — 104.248.82.83	1.22%	40
/js/favicon/favicon.ico — 78.158.140.211	0.61%	20
[redacted]	0.58%	19
/js/saved_resource.html — 194.154.145.65	0.49%	16
/js/activityi.html — 194.154.145.65	0.49%	16
/js/cisco-sans.min.css — 194.154.145.65	0.43%	14
/js/login-federation.css — 194.154.145.65	0.40%	13
/robots.txt — 77.247.110.69	0.40%	13
http://110.249.212.46/testget?q=23333&port=80 - 110.249.212.46	0.40%	13

a few seconds ago

Hits by location

a few seconds ago

HTTP REDIR - Web Hits

3,278

a few seconds ago

DNS REDIR Hits

66

a few seconds ago

Top 5 Incoming Countries - C2

Value	%	Count
Top 5 values		
CN	54.87%	1,770
HK	25.82%	833
CY	6.51%	210
NL	3.87%	125
US	2.23%	72
Others		
IN	1.55%	50
RU	0.90%	29
BR	0.62%	20
TR	0.25%	8
DE	0.25%	8
ID	0.22%	7
SE	0.22%	7
ES	0.22%	7
FR	0.22%	7
IR	0.19%	6
TW	0.19%	6

a few seconds ago

DNS Hits / Source IP

Value	%	Count
Top 5 values		
/js/activityi.html — 194.154.145.65	6.25%	4
/js/saved_resource.html — 194.154.145.65	6.25%	4
/ — 34.203.30.83	6.25%	4
/ — 109.248.216.5	3.13%	2
/ — 190.0.19.26	3.13%	2
Others		
/ — 89.17.44.173	3.13%	2
/abcd — 194.154.145.65	3.13%	2
/ — 107.170.237.222	3.13%	2
/ — 94.28.93.6	3.13%	2
/js/xdcLsFrame.html — 194.154.145.65	3.13%	2
/js/login-federation.css — 194.154.145.65	3.13%	2
/ — 106.75.85.117	3.13%	2
/ — 92.118.37.91	3.13%	2

a few seconds ago

SSH Failure count

0

a few seconds ago

SSH Login Failed Source IP

Value	%	Count

a few seconds ago

SSH login

Value	%	Count
Top 5 values		
yberlab	100.00%	4

a few seconds ago

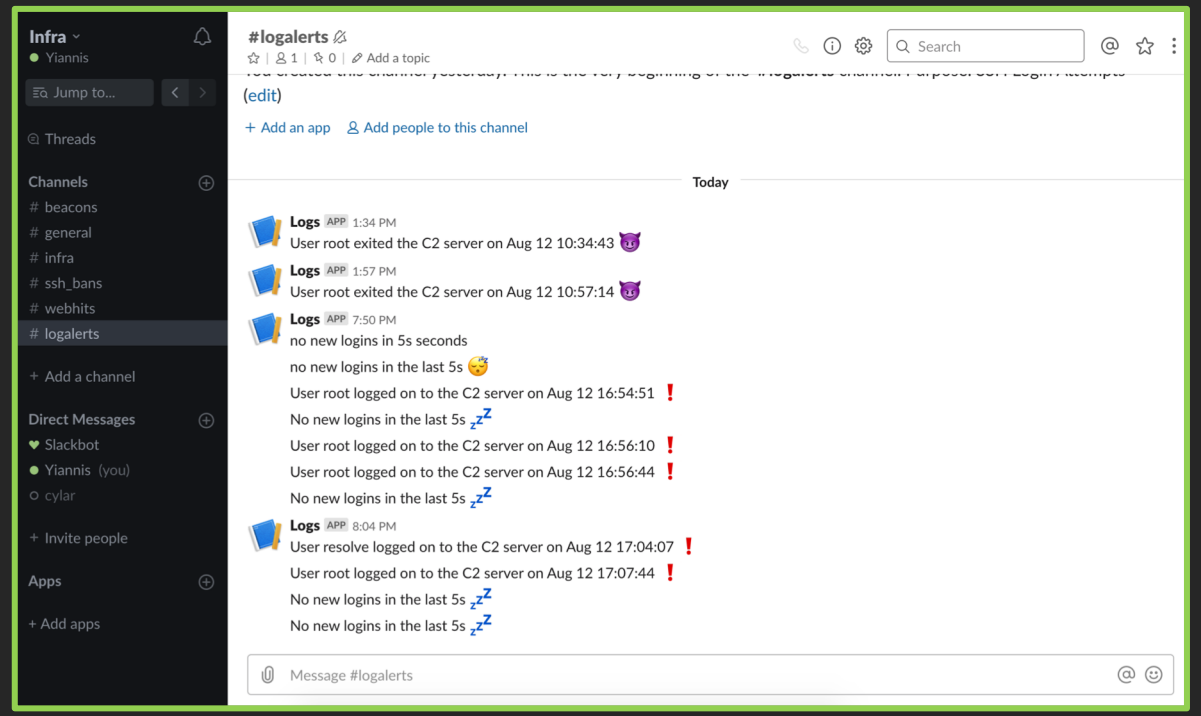
Messages

Previous 1 2 3 4 5 Next

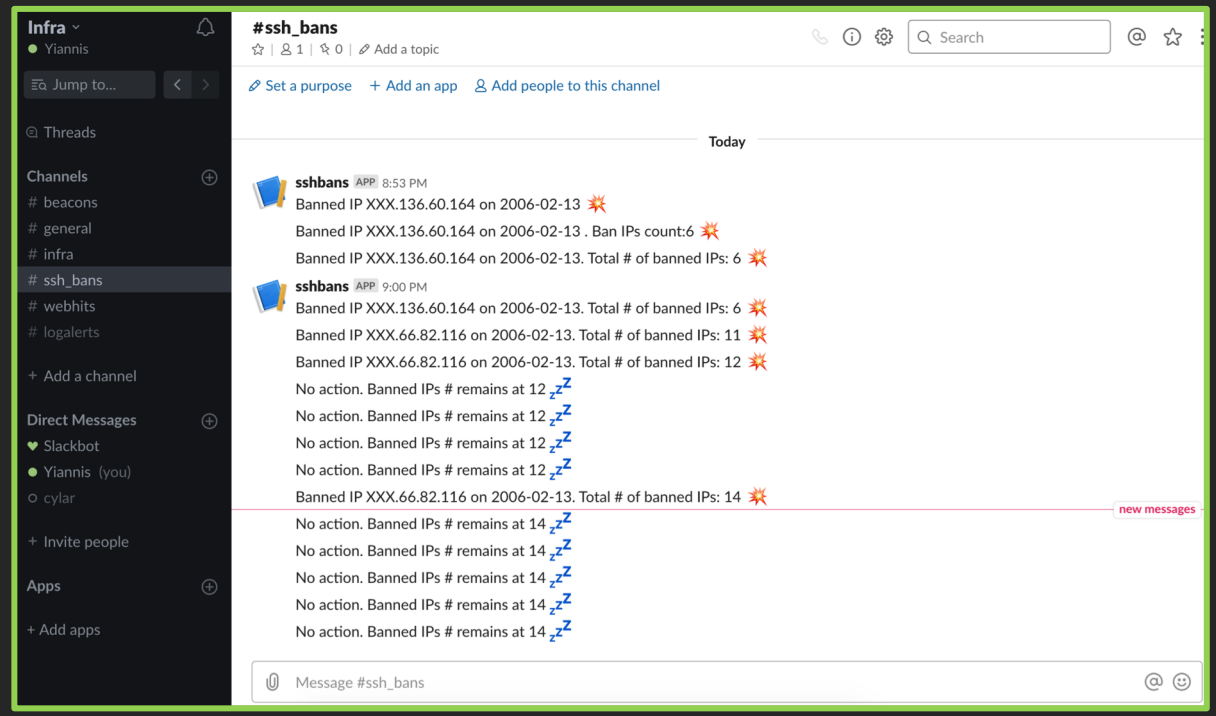
Timestamp	source	clientip
2019-08-17 20:54:05.144	yberlab	80.82.[redacted]
80.82.[redacted] - - [17/Aug/2019:20:54:03 +0000] "GET / HTTP/1.0" 200 11192 "-" "masscan/1.0 (https://github.com/robertdavidgraham/masscan)"		
2019-08-17 20:54:05.143	yberlab	80.82.[redacted]
80.82.[redacted] - - [17/Aug/2019:20:54:03 +0000] "GET / HTTP/1.0" 200 11192 "-" "masscan/1.0 (https://github.com/robertdavidgraham/masscan)"		

Monitoring SSH logs

- Simple bash scripts to monitor changes to the files and post to Slack
- Ideally no strange logs should appear
  - If they do - investigate, eradicate, remediate



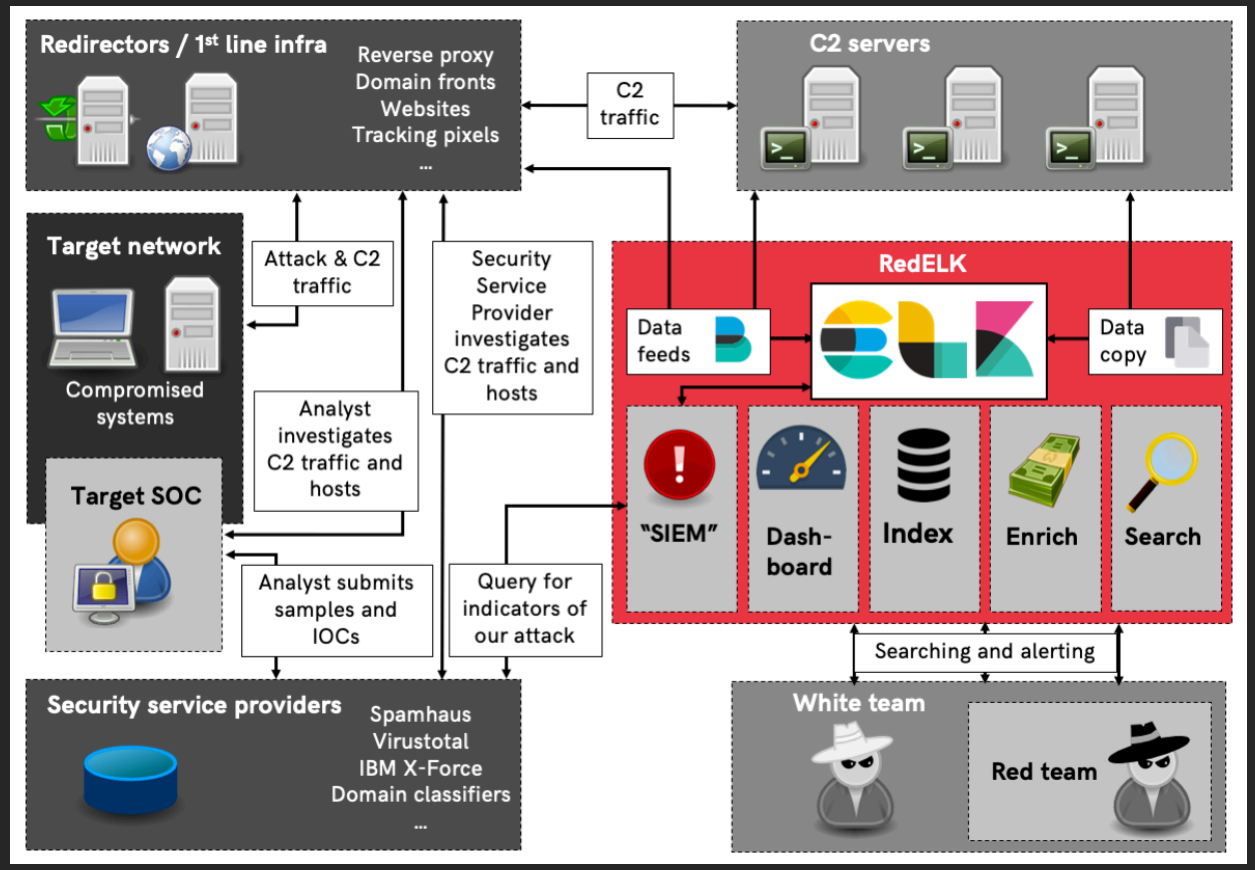
SSH Logins monitoring



SSH fail2ban monitoring

Red Team SIEM

Red Team's SIEM - tool for Red Teams used for tracking and alarming about Blue Team activities as well as better usability for the Red Team in long term operations.



https://github.com/outflanknl/RedELK

Team Activities

- .bashrc hack using the script command
- Easier documentation and evidencing
- Backup, retrace steps and output
- Debugging with client if needed
- Corelate detections with actions
- Improve client detections - IOCs

```
if [ "$color_prompt" = yes ]; then
    #PS1='${debian_chroot:+($debian_chroot)}\[\033[01;31m\]\u@\h\[\033[00m\]:\[\033[01;34m\]\w\[\033[00m\]\$ '
    #ETH0
    PS1='\e[32m[`date  +"%F"`]\e[36m `ifconfig eth0 2>/dev/null | sed -n 2,2p | cut -d" " -f 10 ` \e[39m\w\ > '
    #ETH1
    #PS1='\e[32m[`date  +"%F"`]\e[36m `ifconfig eth1 2>/dev/null | sed -n 2,2p | cut -d" " -f 10 ` \e[39m\w\ > '
```

```
[2019-08-14] 192.168.2.27 ~\ > ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.2.27  netmask 255.255.255.0  broadcast 192.168.2.255
        inet6 fe80::20c:29ff:fed2:bcfb  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:d2:bc:fb  txqueuelen 1000  (Ethernet)
        RX packets 33619  bytes 50066746 (47.7 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 10031  bytes 823224 (803.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[2019-08-14] 192.168.2.27 ~\ >
```

```
x> ~/.bash_logs ls -la
total 8
drwxr-xr-x  2 root root 4096 Aug 20 04:57 ./
drwxr-xr-x 21 root root 4096 Aug 20 04:56 ../
-rw-r--r--  1 root root    0 Aug 20 04:57 root-bash-log-2019-09-25.log
-rw-r--r--  1 root root    0 Aug 20 04:52 root-bash-log-2019-10-25.log
x> ~/.bash_logs
```

Archiving Team Logs

- Project Manager sets up a local git for archiving
- Each team member has their own folder for the logs
- Committing each change with a timestamp

Manager

mkdir teamlogs.git
cd teamlogs.git/
git init --bare
git daemon --reuseaddr --base-path=. --export-all --verbose --enable=receive-pack

Team Members

git clone git://<Manager IP>/teamlogs.git
cp /root/.bash/logs/ Yiannis/ && git add . && git commit –m "`date`" && git push origin master*

```
root@kali:~/Desktop/TeamLogs# cp /root/.bash_logs/* Yiannis/ && git add . && git commit -m "`date`" && git push origin master
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
root@kali:~/Desktop/TeamLogs#
```

```
Yiannis [master] » pwd
/root/Desktop/TeamLogs/Yiannis
Yiannis [master] » ls -la
total 8
drwxr-xr-x 2 root root 4096 Aug 20 12:45 ./
drwxr-xr-x 8 root root 4096 Aug 20 12:41 ../
-rw-r--r-- 1 root root    0 Aug 20 12:41 README
-rw-r--r-- 1 root root    0 Aug 20 12:45 root-bash-log-2019-09-25.log
-rw-r--r-- 1 root root    0 Aug 20 12:45 root-bash-log-2019-10-25.log
Yiannis [master] »
```

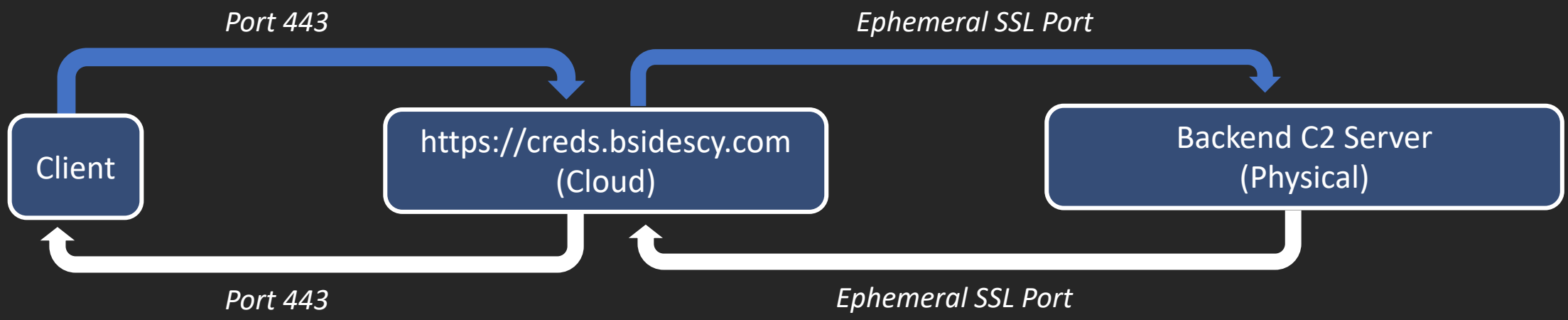
```
commit 14cadf718af19c8f0a293288aa0ff8b69ec8df6f
Author: Yiannis <yiannis@localhost.localdomain>
Date:   Tue Aug 20 04:56:27 2019 -0400

    Tue Aug 20 04:56:27 EDT 2019

commit 6e8c2d6302712ac2dc27f212d6d8ed1e60625246
Author: Yiannis <yiannis@localhost.localdomain>
Date:   Tue Aug 20 04:55:47 2019 -0400

    Tue Aug 20 04:54:23 EDT 2019

commit c2d60002e9fda537be9c741c29a34b81de97db21
Author: root <root@localhost.localdomain>
Date:   Tue Aug 20 04:54:23 2019 -0400

    Tue Aug 20 04:54:23 EDT 2019

commit 14e342fe999e425bba878a821f09a1378535df67
Author: root <root@localhost.localdomain>
Date:   Tue Aug 20 04:49:53 2019 -0400

    Tue Aug 20 04:49:53 EDT 2019 Yiannis
```

Honeypots / System monitoring

- Honeypots can be set up to confuse attackers/defender
- Buy you some time while making infrastructure changes
- Examples:
  - Artillery - https://github.com/BinaryDefense/artillery
    - Sets up services on various ports and monitors for action
    - Automatic Blacklisting
    - Monitor user specified folders for changes
    - Email alerting
  - Cowrie – https://github.com/cowrie/cowrie
    - Sets up services on various ports and monitors for action
    - Fake filesystem to observe attacker moves
  - Denyhosts - https://github.com/denyhosts/denyhosts
    - Automatic host blocking
    - Email about intrusion attempts

Phishing setup

- Scenario:
  - User clicks on the malicious link
  - Redirected to https://creds.bsidescy.com
  - Content served from the backend C2 server
  - Captured credentials stored on the backend server
  - If the cloud server is compromised it does not hold any client information
- Adverse Scenario:
  - Unwanted traffic hitting our server exposing our phishing campaign
  - Domain is listed by search engines
  - Blacklisting of the domain by web content filters
  - Attack out of scope legitimate business users or clients of our client



Traffic Filtering – Apache mod_rewrite

- Apache module which does URL manipulations on the fly
- Somewhat complex syntax but it can be very powerful

Rule 1 – Redirection based on User Agent – Avoid search engines from indexing the phishing domain / Curious users accessing the link via their mobile device

```
RewriteEngine On
RewriteCond %{HTTP_USER_AGENT} "Macintosh|android|googlebot-mobile|ipad|iphone|webos" [NC]
RewriteRule ^.*$ https://www.google.com [L,R=302]
RewriteRule ^.*$ http://BACKEND-IP%{REQUEST_URI} [P]
```

Rule 1 Breakdown

If the incoming user-agent matches the list in any case (NC):
- Redirect (R) the user to google.com – Change the URL in the URL bar and don't process any other rules (L)

else
- Fetch the requested URI from the backend server but don't change the URL in the URL bar – Act as a proxy (P)

Credit:
https://bluescreenofjeff.com/2018-04-12-https-payload-and-c2-redirectors/

Traffic Filtering – Apache mod_rewrite

- Users some times try to be “smart” or curious and request other URIs
- Automated web content discovery tools
- Malicious actors testing our infrastructure will be directed to Google

Rule 2 – Redirection based on requested URL

```
RewriteEngine On
RewriteCond %{REQUEST_URI} ^/(malware\.exe|register\.html)/?$ [NC]
RewriteRule ^.*$ http://BACKEND-IP%{REQUEST_URI} [P]
RewriteRule ^.*$ https://google.com/ [L,R=302]
```

Rule 2 Breakdown

If the incoming URI request matches one or the other URI in the list in any case (NC):
- Fetch the requested URI from the backend server but don’t change the URL in the URL bar – Act as a proxy (P)

else
- Redirect (R) the user to google.com – Change the URL in the URL bar and don’t process any other rules (L)

Credit:
https://bluescreenofjeff.com/2018-04-12-https-payload-and-c2-redirectors/

Traffic Filtering – Apache mod_rewrite

- IP filtering if we know the IP space of the client

Rule 3 – Redirection based on originating IP

```
RewriteEngine On
RewriteCond %{REMOTE_ADDR} ^63\.0\.0\. [OR]
RewriteCond %{REMOTE_ADDR} ^64\.0\.1\.
RewriteRule ^.*$ http://BACKEND-IP%{REQUEST_URI} [P]
RewriteRule ^.*$ https://google.com/ [L,R=302]
```

Rule 3 Breakdown

If the originating IP request matches one or the other (OR) rule:
- Fetch the requested URI from the backend server but don't change the URL in the URL bar – Act as a proxy (P)

else:
- Redirect (R) the user to google.com – Change the URL in the URL bar and don't process any other rules (L)

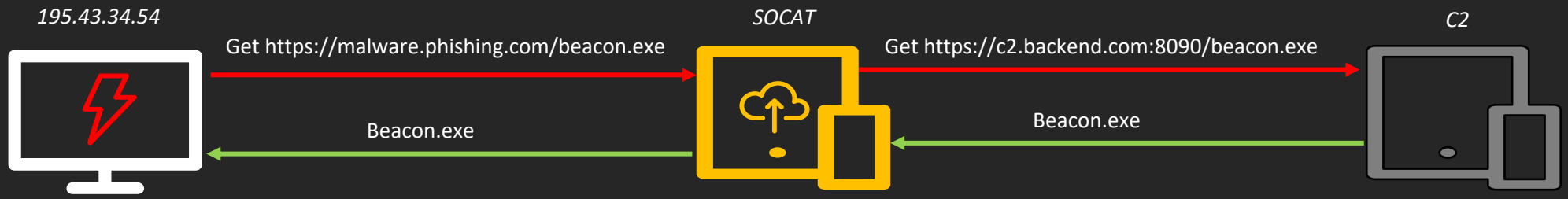
Credit:
https://bluescreenofjeff.com/2016-04-12-combatting-incident-responders-with-apache-mod_rewrite/

HTTP/S Redirectors – socat

Example 1 – Packet redirection based on port

socat TCP4-LISTEN:80,fork TCP4:<C2 IP>:8090

TCP4-LISTEN: Listen for connections on TCP port 80
Fork: Create new child process so that the parent process can go back to listening
TCP4:<IP>:8090 – forward the received packet to the TCP port 8090 of the c2 IP



Example 2 – Packet redirection – Reverse tunnel

autossh -M 0 user@<CLOUD IP> -p 22 -N -R 2222:127.0.0.1:8090
socat TCP4-LISTEN:443,fork TCP4:localhost:2222

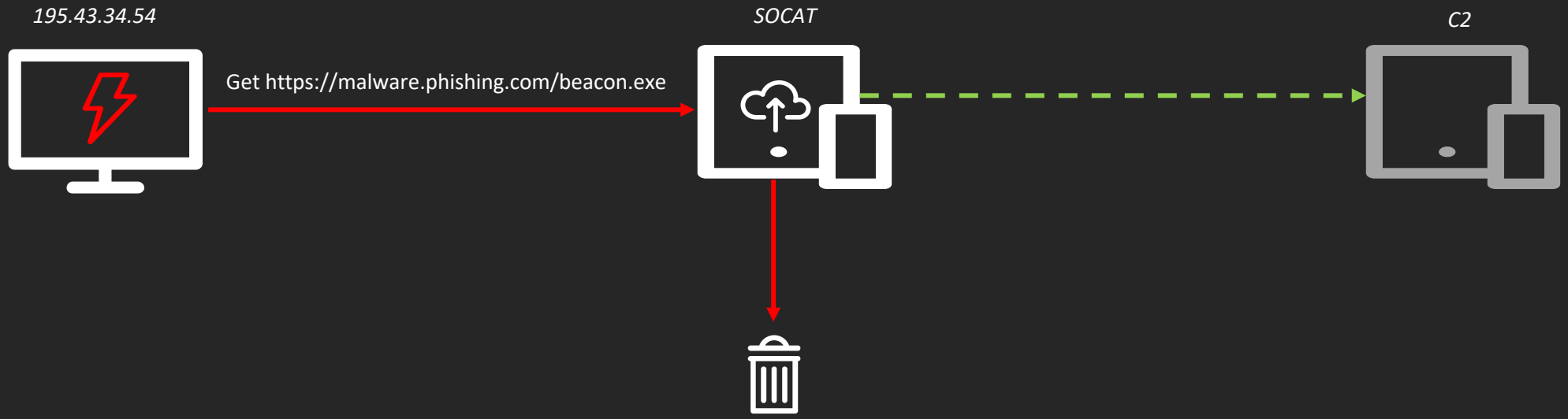
TCP4-LISTEN: Listen for packets on TCP port 443
Fork: Create new child process so that the parent process can go back to listening
TCP4:<IP>:8090 – forward the received packet to localhost TCP port 2222

HTTP/S Redirectors – socat

Example 2 - Packet redirecting with ingress filtering

socat TCP-LISTEN:80,fork,range=10.10.10.10/32 TCP:<C2 IP>:8090

TCP4-LISTEN: Listen for packets on TCP port 80
Fork: Create new child process so that the parent process can go back to listening
Range: Accepted range for incoming IPs for forwarding packets
TCP4:<IP>:8090 – forward the received packet to the TCP port 8090 of the C2 IP



HTTP/S Redirectors – HAPROXY / Layer 7

```
global
        maxconn 2000
        stats socket :9000 mode 660 level admin

defaults

        mode http
        log global
        option httplog
        timeout connect 15s
        timeout client 15s
        timeout http-request 15s
        timeout server 15s

frontend redirector
        bind :80
        http-request deny unless { src 85.208.96.0/22 }
        default_backend c2

backend c2
        server server1 XXX.XXX.XXX.XXX:8080
```

Redirection based on source IP range

Listen on port 80 and accept and redirect traffic only from a specific range.

Accepted traffic to be redirected to the C2 IP and port.

HTTP/S Redirectors – HAPROXY / Layer 7

```
global
        maxconn 2000
        stats socket :9000 mode 660 level admin

defaults

        ....

frontend redirector
        bind :80
        acl malware path_beg /beacon
        use_backend c2 if malware
        default_backend byebye

backend c2
        server server1 XXX.XXX.XXX.XXX:8080

backend byebye
        server server2 www.google.com
```

Redirection based on requested URI

Listen on port 80 and accept and redirect requests for a specific URI.

Accepted traffic to be redirected to the C2 IP and port.

HTTP/S Redirectors – iptables

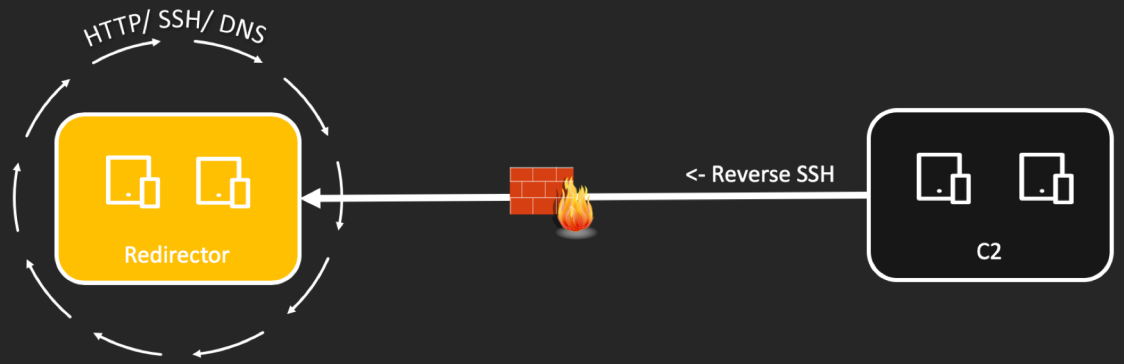
Rule – iptables NAT

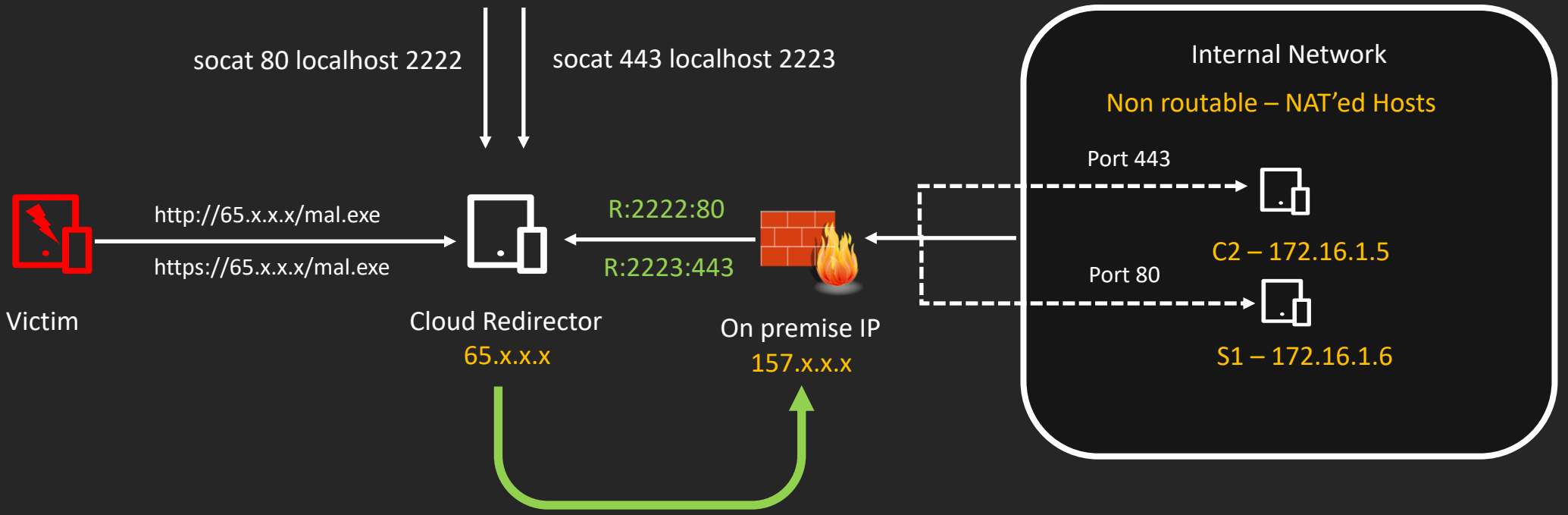
```
# Accept TCP traffic to port 80
iptables -I INPUT -p tcp -m tcp --dport 80 -j ACCEPT

# Use the PREROUTING chain to NAT incoming packets to the C2 IP address and port
iptables -t nat -A PREROUTING -p tcp --dport 80 -j DNAT --to-destination <REMOTE-HOST-IP-ADDRESS>:8090

# Mangle the packet as it leaves the interface to include the router's IP address – NAT
iptables -t nat -A POSTROUTING -j MASQUERADE

# Authorize forwarding from LAN
iptables -I FORWARD -j ACCEPT

# Set the default policy to accept to forward packets
iptables -P FORWARD ACCEPT
```

* Assuming the C2 is directly accessible from the redirector
*** Assuming the default INPUT policy is set to DROP*

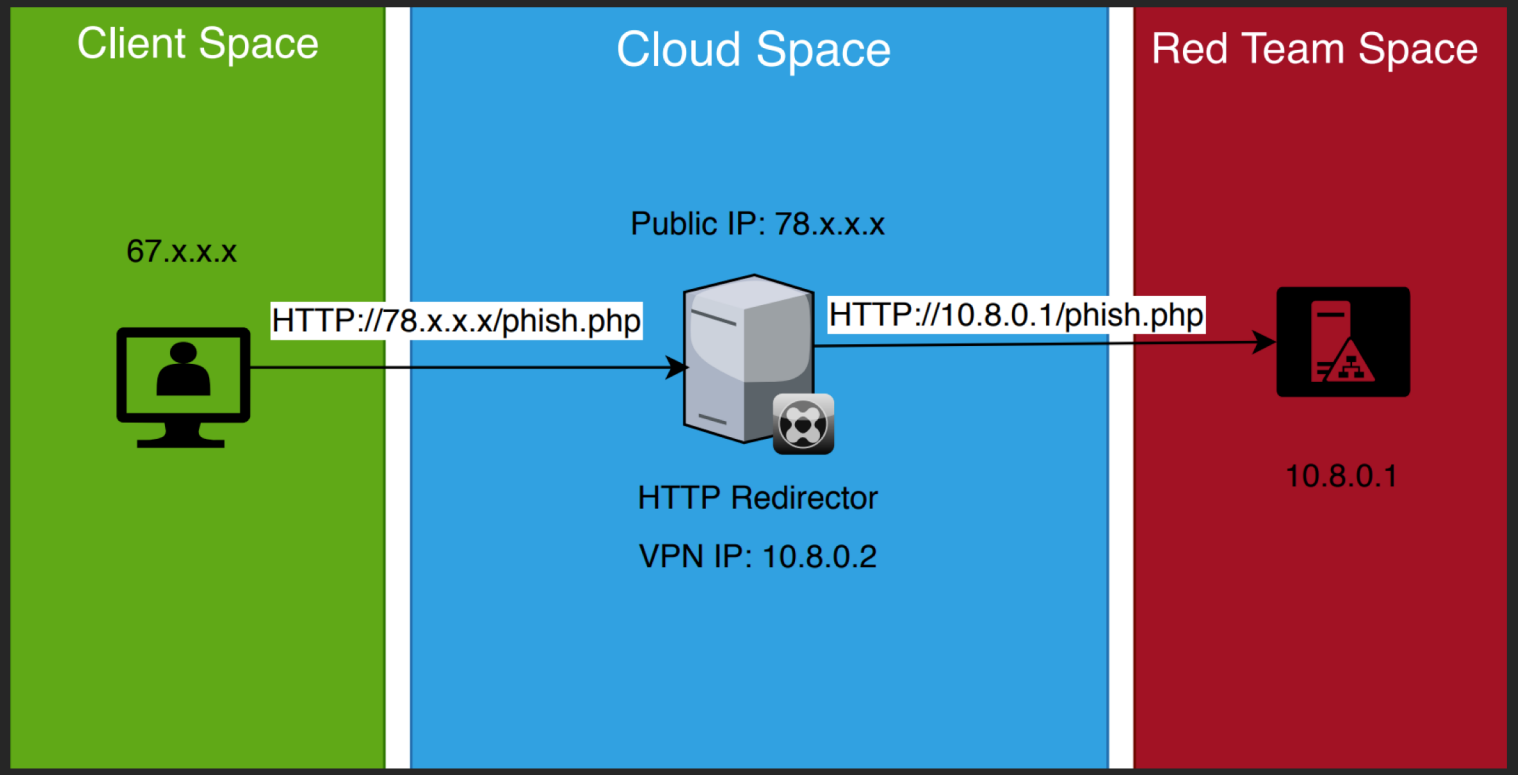
Examples and ideas for the rules:
https://bluescreenofjeff.com/2018-04-12-https-payload-and-c2-redirectors/

Communications Breakdown

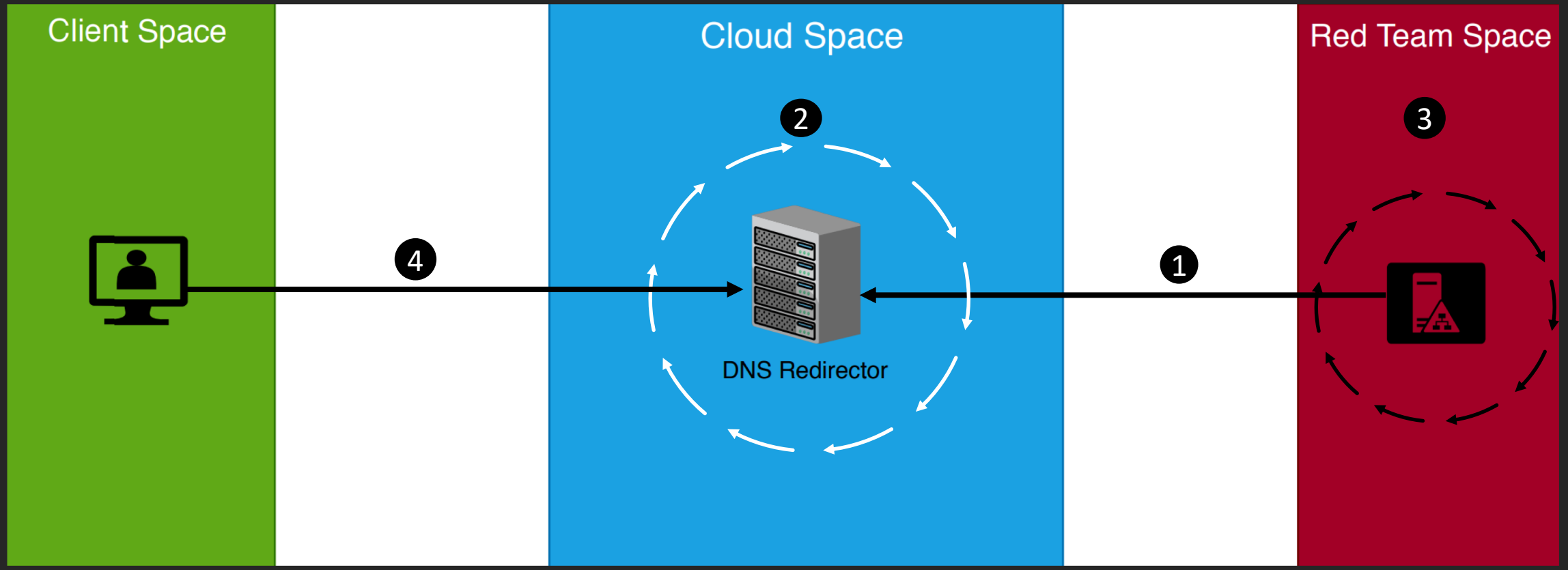




HTTP/S Redirectors – VPN



DNS Redirectors – socat



1) *autossh -M 0 root@<DNS REDIRECTOR IP> -p 22 -N -R 2222:127.0.0.1:53* — Reverse SSH on port 2222 forward to port TCP 53
2) *socat udp4-LISTEN:53,fork tcp4:localhost:2222* — Send incoming traffic on port UDP 53 to localhost port 2222
3) *socat tcp4-LISTEN:53,fork udp:localhost:53* — *Forward incoming traffic on port TCP 53 to port UDP 53 on localhost*
4) *Client makes request for malware.bsidescy.com* — *Payload is served from the C2*

Examples and ideas:
https://medium.com/rvrsh3ll/redirecting-cobalt-strike-dns-beacons-e3dcdb5a8b9b

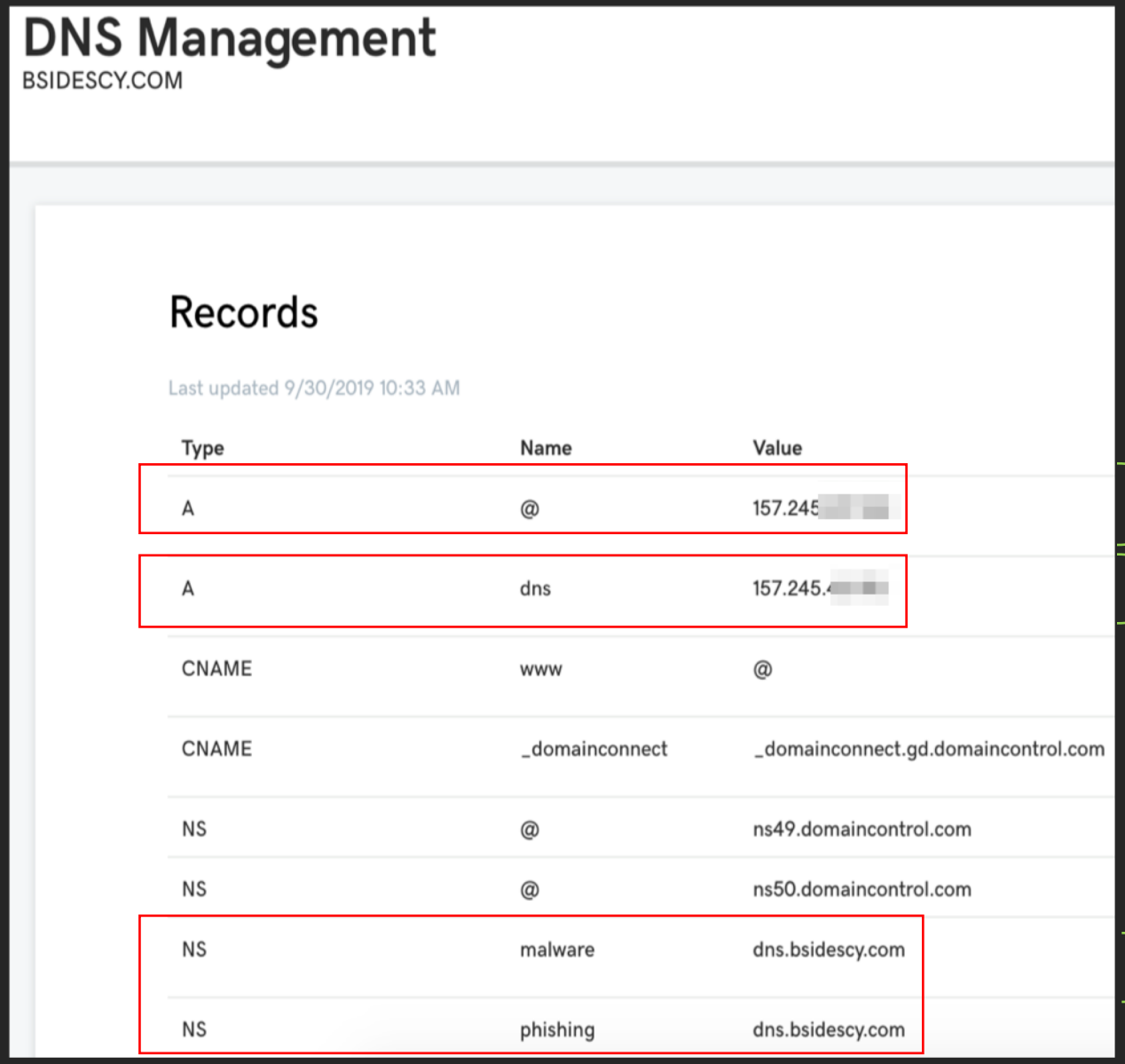
DNS Redirectors – iptables

Rule – iptables NAT

```
#Accept UDP traffic to port 53 for DNS payload
iptables -I INPUT -p udp -m udp --dport 53 -j ACCEPT

# Use the PREROUTING chain to NAT incoming packets to the C2 IP address
iptables -t nat -A PREROUTING -p udp --dport 53 -j DNAT --to-destination <C2 IP ADDRESS>:53

# Mangle the packet as it leaves the interface to include the router's IP address - NAT
iptables -t nat -A POSTROUTING -j MASQUERADE

# Authorize forwarding from LAN
iptables -I FORWARD -j ACCEPT

# Set the default policy to accept to forward packets
iptables -P FORWARD ACCEPT

* Assuming the C2 is directly accessible from the redirector
** Assuming the default INPUT policy is set to DROP
```

Examples and ideas for the rules:
https://bluescreenofjeff.com/2018-04-12-https-payload-and-c2-redirectors/

DNS Redirectors – socat – Cobalt Strike DNS/Reverse HTTP



DNS Management

BSIDESCY.COM

Records

Last updated 9/30/2019 10:33 AM

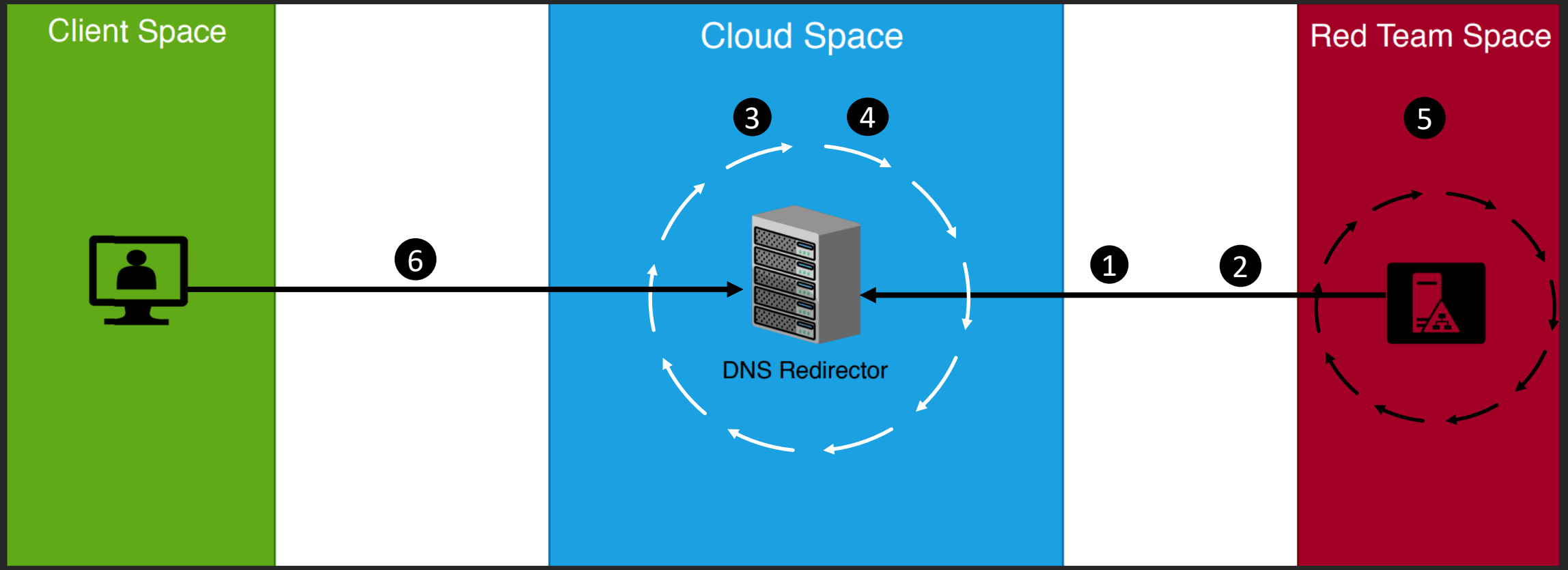
Type	Name	Value
A	@	157.245
A	dns	157.245.
CNAME	www	@
CNAME	_domainconnect	_domainconnect.gd.domaincontrol.com
NS	@	ns49.domaincontrol.com
NS	@	ns50.domaincontrol.com
NS	malware	dns.bsidescy.com
NS	phishing	dns.bsidescy.com

A records pointing to the HTTP redirector in the cloud infrastructure

A record pointing to the DNS redirector in the cloud infrastructure

NS records pointing to the DNS redirector A Record

DNS Redirectors – socat – Cobalt Strike DNS/Reverse HTTP



1) *autossh -M 0 root@<DNS REDIRECTOR IP> -p 22 -N -R 2222:127.0.0.1:53* — Reverse SSH on port 2222 forward to port TCP 53 – DNS Payload
2) *autossh -M 0 root@<DNS REDIRECTOR IP> -p 22 -N -R 2223:127.0.0.1:443* — Reverse SSH on port 2223 forward to port TCP 443 – HTTP Stager
3) *socat udp4-LISTEN:53,fork tcp4:localhost:2222* — Send incoming traffic on port UDP 53 to localhost port 2222
4) *socat tcp4-LISTEN:443,fork tcp4:localhost:2223* — Send incoming traffic on port TCP 443 to localhost port 2223
5) *socat tcp4-LISTEN:53,fork udp:localhost:53* — *Forward incoming traffic on port TCP 53 to port UDP 53 on localhost*
6) *Client makes request for malware.bsidescy.com* — *Payload is served from the C2*

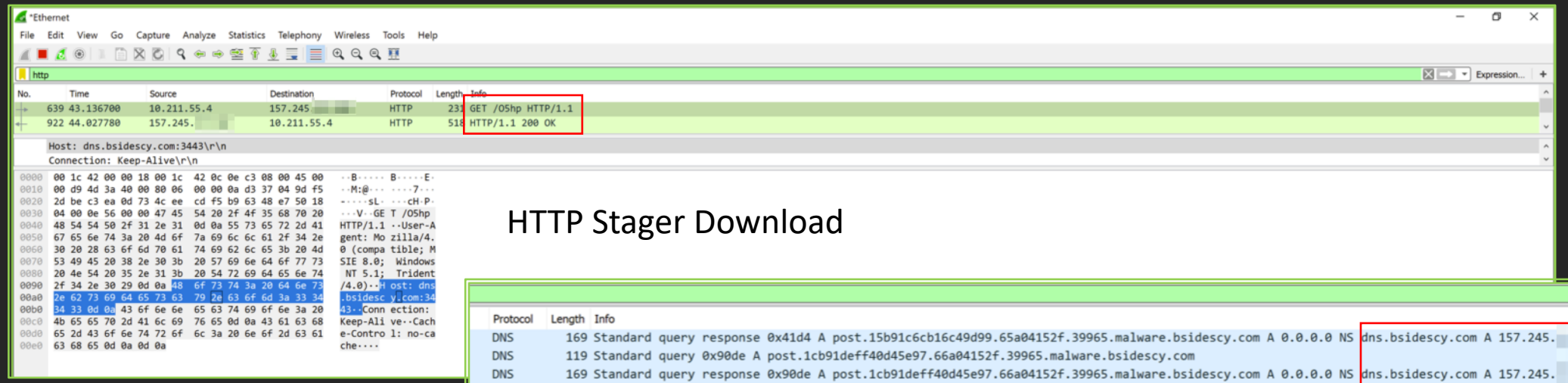
DNS Redirectors – iptables - Cobalt Strike DNS/Reverse HTTP

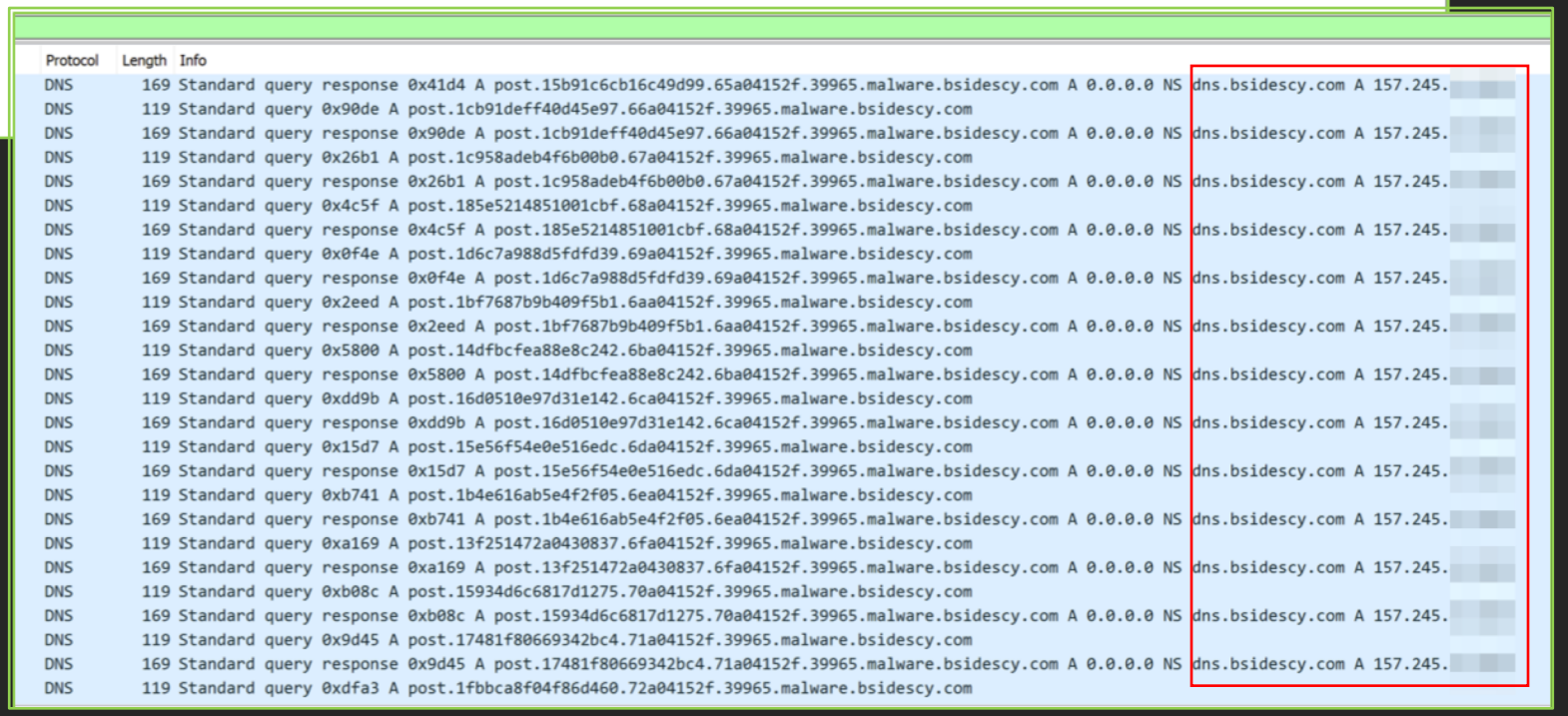
Rule – iptables NAT

```
#Accept TCP traffic to port 443 for DNS payload staging over HTTP
iptables -I INPUT -p tcp -m tcp --dport 443 -j ACCEPT

#Accept UDP traffic to port 53 for DNS payload
iptables -I INPUT -p udp -m udp --dport 53 -j ACCEPT

# Use the PREROUTING chain to NAT incoming packets to the C2 IP address - NAT
iptables -t nat -A PREROUTING -p udp --dport 53 -j DNAT --to-destination <C2 IP ADDRESS>:53
iptables -t nat -A PREROUTING -p tcp --dport 443 -j DNAT --to-destination <C2 IP ADDRESS>:443

# Mangle the packet as it leaves the interface to include the router's IP address
iptables -t nat -A POSTROUTING -j MASQUERADE

# Authorize forwarding from LAN
iptables -I FORWARD -j ACCEPT

# Set the default policy to accept to forward packets
iptables -P FORWARD ACCEPT

* Assuming the C2 is directly accessible from the redirector
** Assuming the default INPUT policy is set to DROP
```

Examples and ideas for the rules:
https://bluescreenofjeff.com/2018-04-12-https-payload-and-c2-redirectors/

DNS Redirector - Client side view





DNS tasks – Redirector IP and domain

Redirectors setup - tmux automation

C2 server tmux config file

```
#!/bin/sh

tmux new-session -d -s c2 ;
tmux split-window -v ;
tmux select-pane -U
tmux send-keys -t c2 'autossh -M 0 root@[REDIRECTOR IP] -p 22 -N -R 2222:127.0.0.1:53 -vvv' C-m
tmux select-pane -D
tmux split-window -v ;
tmux select-pane -U
tmux send-keys -t c2 'autossh -M 0 root@[REDICRECTOR IP] -p 22 -N -R 2223:127.0.0.1:443 -vvv' C-m
tmux select-pane -D
tmux split-window -v ;
tmux select-pane -U
tmux send-keys -t c2 'sudo socat tcp4-LISTEN:53,fork udp:localhost:53' C-m
tmux select-pane -D
tmux send-keys -t c2 'cd /opt/cobaltstrike; sudo ./teamserver [C2 IP] [PASSWORD]' C-m
tmux attach-session -d -t c2
```

Cloud Redirector tmux config file

```
#!/bin/sh

tmux new-session -d -s redirect ;
tmux split-window -v ;
tmux select-pane -U
tmux send-keys -t redirect 'socat udp4-LISTEN:53,fork tcp4:localhost:2222' C-m
tmux select-pane -D
tmux send-keys -t main 'socat tcp4-LISTEN:443,fork tcp4:localhost:2223' C-m
tmux attach-session -d -t redirect
```

Data at rest

- Even though the C2 is locked down, data at rest should still be protected
- Linux offers different options depending if you want to so full disk encryption or file encryption:
  - Veracrypt
  - eCryptfs
  - GPG
  - Tomb
    - standard filesystem tools (GNU)
    - cryptographic API of the Linux kernel (cryptsetup and LUKS).
    - AES256

```
tomb  .  Commanded to dig tomb bsides.tomb
tomb (*) Creating a new tomb in bsides.tomb
tomb  .  Generating bsides.tomb of 30MiB
30+0 records in
30+0 records out
31457280 bytes (31 MB, 30 MiB) copied, 0.20581 s, 153 MB/s
-rw------- 1 root root 30M Sep  3 07:43 bsides.tomb
tomb (*) Done digging bsides
tomb  .  Your tomb is not yet ready, you need to forge a key and lock it:
tomb  .  tomb forge bsides.tomb.key
tomb  .  tomb lock bsides.tomb -k bsides.tomb.key
```

```
tomb  .  Commanded to forge key bsides.tomb.key with cipher algorithm AES256
tomb [W] This operation takes time. Keep using this computer on other tasks.
tomb [W] Once done you will be asked to choose a password for your tomb.
tomb [W] To make it faster you can move the mouse around.
tomb [W] If you are on a server, you can use an Entropy Generation Daemon.
512+0 records in
512+0 records out
512 bytes copied, 222.536 s, 0.0 kB/s
tomb (*) Choose the password of your key: bsides.tomb.key
```


Questions?