

eCTHP V2

suspicious Traffic Hunting

by: Ahmad Abdennasser Soliman

- Index of chapter:

- **Introduction**.....**1-2**
- **ARP traffic**.....**2-6**
- **ICMP traffic**.....**6-17**
- **TCP traffic**.....**17-22**
- **DHCP traffic**.....**22-29**
- **DNS traffic**.....**29-37**
- **HTTP, HTTPS traffic**.....**37-67**
- **Unknown traffic**.....**67-72**

Introduction:

- ف الاول لازم نشوف ال normal traffic وال suspicious traffic عشان نعرف نميز بينهم ونعمل analyzing traffic بيمر في الشبكة.

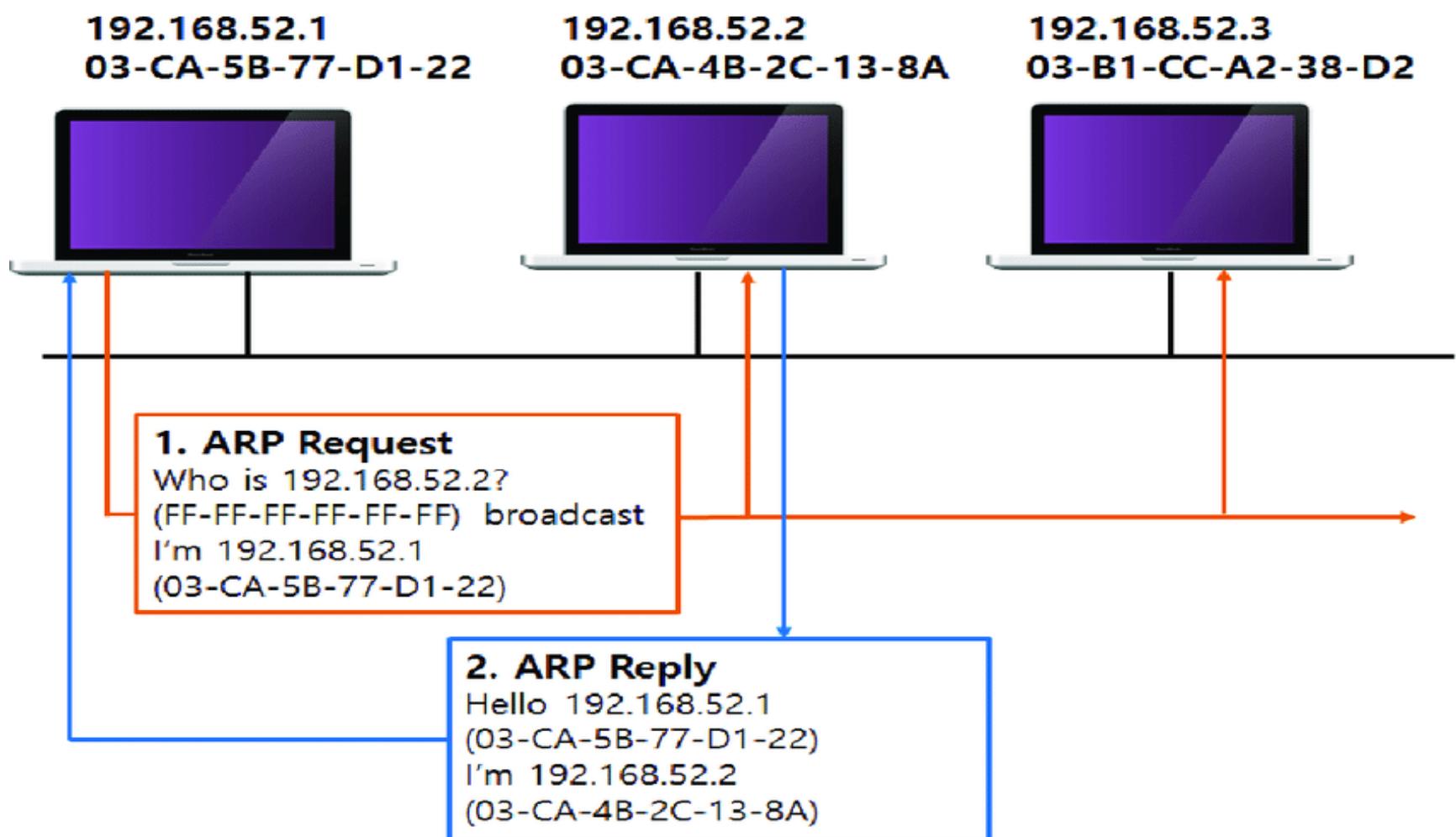
- ممکن يجي شخص ما يقول هو احنا ليه هنعمل **hunting** لـ **traffic** بنفسنا **manual** بتعمل الحاجات دي ؟؟

- الغرض من دا كله اننا نعود عيننا انها تطلع وتصطاد ال **suspicious traffic** عشان في بعض الشركات الكبيره هتحتاج **investigation** بایدك حاجات وتدخل جوا **packets** وتعملها بنفسك .

ARP traffic:

- برتوکول ال **ARP** اختصار ل **Address Resolution protocol** يعتبر من اهم البرتوكولات عندك ف الشبکه وهو المسؤول انه بیحول ال **Ip** ل **mac address** بمعنى لو انت معاك **Ip** ل **mac address** معین وعاوز تعرف ال **destination** بستخدم برتوکول ال **ARP** ودا بيتم من خلال **ARP Request** و **ARP Reply** تمام كدا موجود ف **layer 2** من ال **tcp/IP model** .

- ال **ARP Request** هو الرقم التعريفي لـ **operation code** عشان هتشوفه كتير فال **wire shark** هو {1} ولل **header** هو {2}... عشان هتشوفه وانت بتحلل ال



ال : normal ARP traffic

Arp broadcasts are normal from both clients and servers at reasonable flow.

- تقدر تقول هي رسائل **Arp** بتتبع من **destination source** لـ **Arp** ولكن بشكل معقول بمعنى متكرش رسائل كثيره ورا بعضها ف وقت قليل.

عشان فال **suspicious** هتلaci رسائل ال **ARP** بتيجي كثير و بتوصل للاف ف وقت قليل فانت هنا تعرف ان في حاجه مش تمام و دا ف الغالب بيبيقي **attacker** بيعمله ال **scanning behavior** باستخدام اداه ال **Nmap** عشان يعرف ال **ports** المفتوحه عندك .

- تاني حاجه فال **normal ARP** هتلaci الطبيعي انك تبعت

ARP ب **destination** يترض عليك من ال **Arp request**

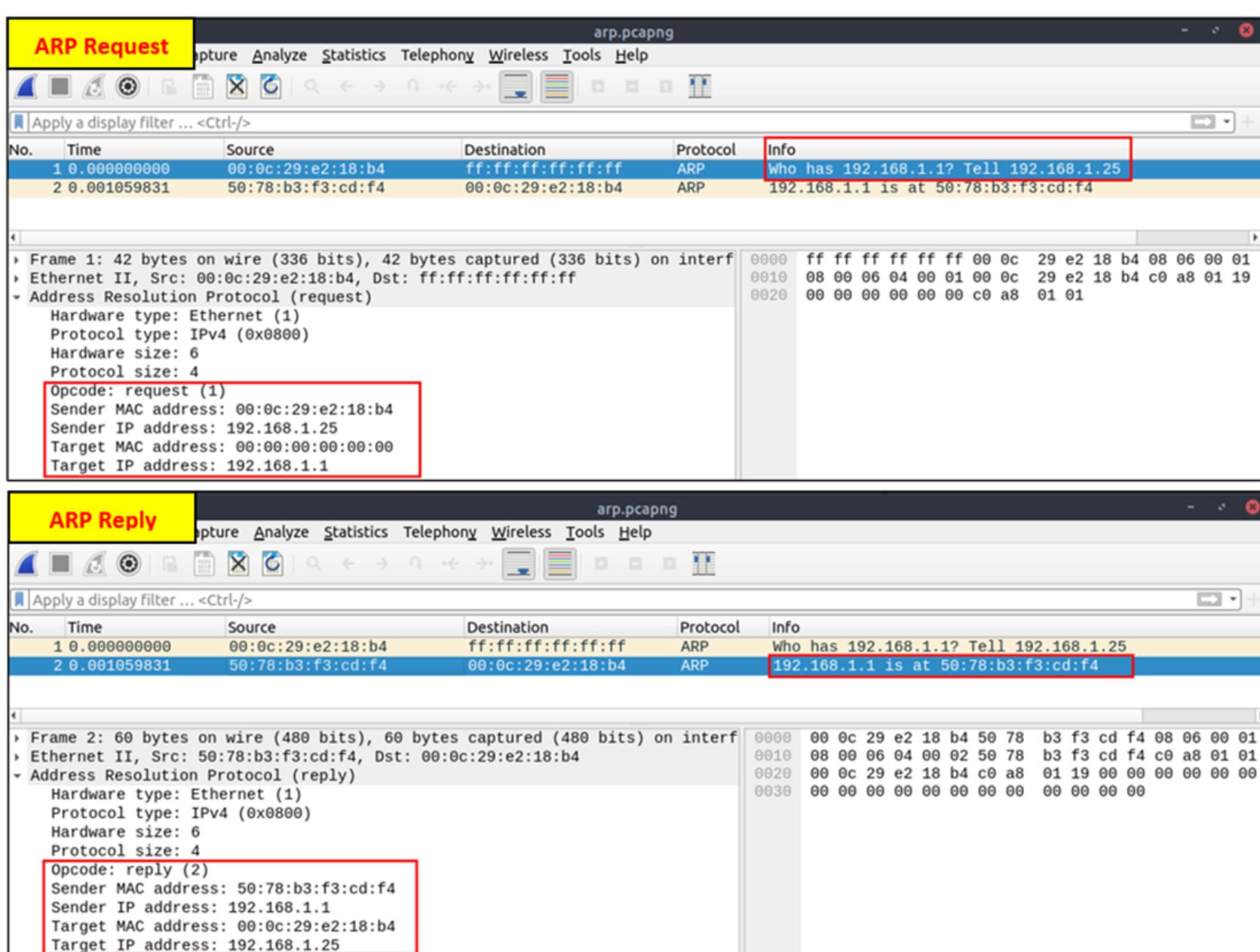
واحد فقط بمعنى يرض عليك **reply** **mac address**

mac address هتلaci نفس ال **suspicious** عشان فال

بس موجود عندك **two different IP address** فانت كدا عندك

فتاخد بالك من الحته دي ودا هنوضحه **Arp spoofing attack**

ف الصور كما سيأتي لاحقا ...



- الجزء الثالث هتلaci ان ف الطبيعي **وال normal** مبيجلکش

الا اما تكون باعت **Arp request** مضبوط كدا

- بس هتلاقي **فال suspicious** ال **gratuitous Arp reply**

بمعنى لذك هتلاقي ال **Arp reply** بيعتلك **attacker** لوحده كدا من غير متبعتك **mac address** ويقولك روح غير ال **request** الموجود فال **Arp table** او ال **Arp cache** بال **mac** الجديد ال هو بتاع ال **attacker** فهو كدا بينتحل شخصيه جهاز تاني معاك ع الشبكه فبعد كدا اما تيجي تبعت هتلاقي نفسك بتبعن لـ **attacker** بدل ال **user** ودا بنسميه ال **Arp spoofing** او ال **ARP poisoning** وهتلاقي ال **Arp reply** كل 30 ثانية بيعتلك عشان تحدث ال **attacker** ال عندك عشان ال **Session** الخاصه بال **ARP cache** بيتعملها كل 30 ثانية فيعمل كدا عشان يحافظ على الاتصال .

- لو بصينا ف المثال دا هنشوف ال **normal Arp reply** قبل ميحصل اي **attack** فال **normal case** بتعته ورد عليه الجهاز المطلوب التواصل معاه بال **mac** الخاص بي، وبال **operation code** زي مقولنا والدنيا تمام وزي الفل.

- تعالى نبص عال **suspicious** هتلاقي الدنيا مختلفه هتلاقي نفس ال **source** شغال بيعت **broadcast message** للناس ال معاه ع الشبكه كلهم ولو بصيت ع التوقيت هتلاقيه كله ورا بعضه حتى مبيستناش يسلام ال **reply** فتعرف ان دا **behavior scanning** شغال عليك - ودا تميزه من الوقت ما بين كل **packet** وال وراها لو لقيت الوقت قليل جدا تعرف ان دا **behavior scanning** وهتلاقي برضه نفس ال **IP** بيعت لاجهزه مختلفه ع الشبكه دون انتظار اي **.... reply**

- حطّلک صوره توضیح فکرہ انک ممکن تلاقی ال mac بیستخدمه 2

ARP Spoofing فتعرف من خلاله ان دا different IP

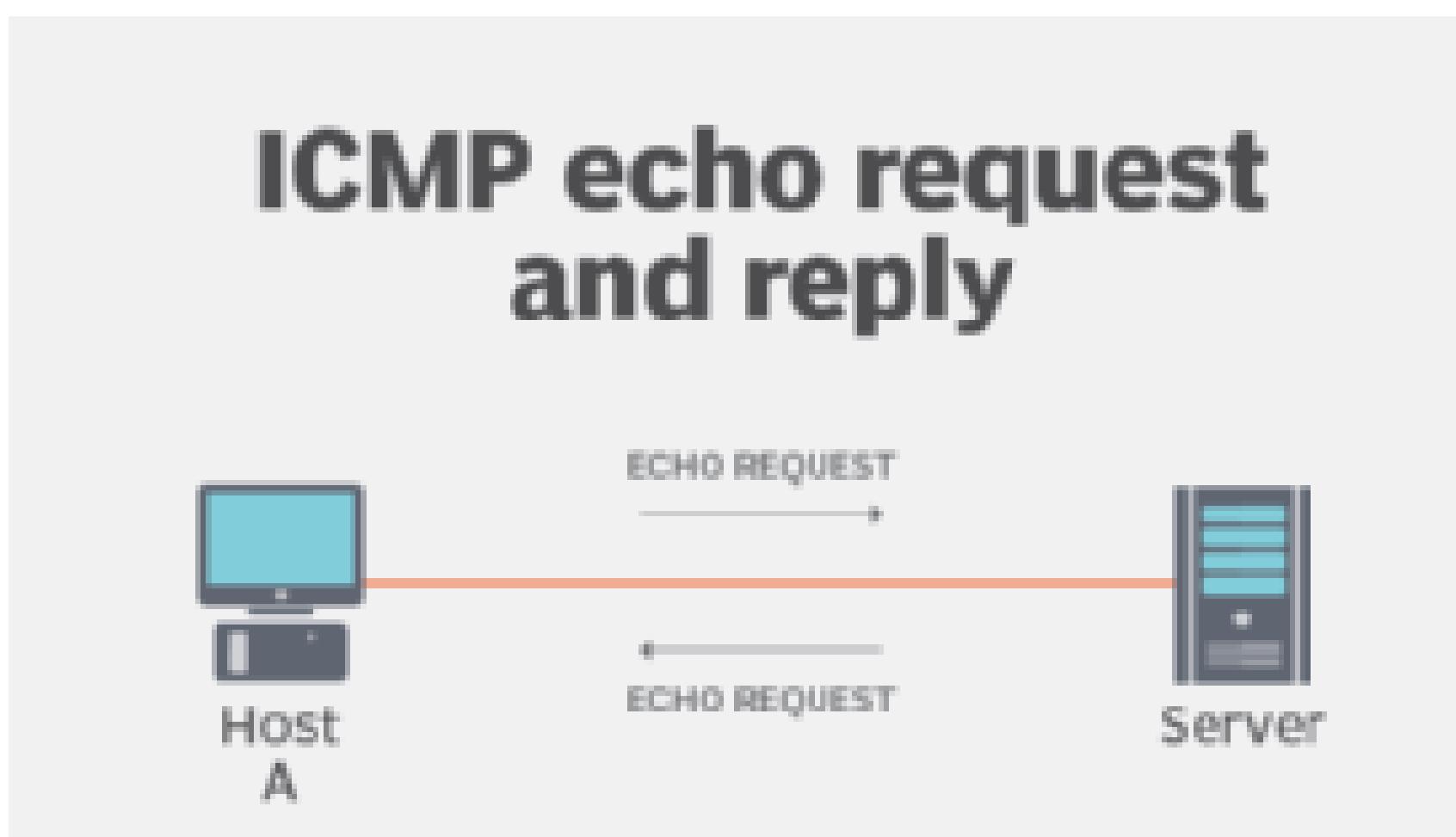
No.	Time	Source	Destination	Protocol	Info
1	0.0000000000	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.1? Tell 192.168.1.25
2	0.001059831	50:78:b3:f3:cd:f4	00:0c:29:e2:18:b4	ARP	192.168.1.1 is at 50:78:b3:f3:cd:f4
3	0.010490253	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.37? Tell 192.168.1.25
4	0.020876839	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.158? Tell 192.168.1.25
5	0.031275021	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.212? Tell 192.168.1.25
6	0.041848453	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.176? Tell 192.168.1.25
7	0.052746298	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.73? Tell 192.168.1.25
8	0.063388601	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.216? Tell 192.168.1.25
9	0.073905794	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.181? Tell 192.168.1.25
10	0.084401792	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.217? Tell 192.168.1.25
11	0.095003040	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.173? Tell 192.168.1.25
12	0.105417559	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.136? Tell 192.168.1.25
13	0.115638938	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.132? Tell 192.168.1.25
14	0.125920898	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.130? Tell 192.168.1.25
15	0.136708415	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.254? Tell 192.168.1.25
16	0.147294383	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.232? Tell 192.168.1.25
17	0.157926474	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.162? Tell 192.168.1.25
18	0.168416850	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.109? Tell 192.168.1.25
19	0.178936116	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.253? Tell 192.168.1.25
20	0.189453050	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.169? Tell 192.168.1.25

No.	Time	Source	Destination	Protocol	Info
1	0.0000000000	00:0c:29:e2:18:b4	50:78:b3:f3:cd:f4	ARP	Who has 192.168.1.1? Tell 192.168.1.25
2	0.001271501	50:78:b3:f3:cd:f4	00:0c:29:e2:18:b4	ARP	192.168.1.1 is at 50:78:b3:f3:cd:f4
3	0.393554684	00:0c:29:e2:18:b4	00:0c:29:98:c7:a8	ARP	192.168.1.1 is at 00:0c:29:e2:18:b4

Frame 3: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface eth0, id 0
Ethernet II, Src: 00:0c:29:e2:18:b4, Dst: 00:0c:29:98:c7:a8
Address Resolution Protocol (reply)
Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: reply (2)
Sender MAC address: 00:0c:29:e2:18:b4
Sender IP address: 192.168.1.1
Target MAC address: 00:0c:29:98:c7:a8
Target IP address: 192.168.1.12
[Duplicate IP address detected for 192.168.1.1 (00:0c:29:e2:18:b4) - also in use by 50:78:b3:f3:cd:f4 (frame 2)]

ICMP Traffic:

- اختصار لي **internet control message protocol** موجود ف **layer 2** ال هي **IP layer** وملوش **port** معين بيستغل عليه ودا برتوكول هتلاقيه مشهور لل **network trouble shooting** زي ان جهازك فيه مشكله ف الاتصال بالانترنت وزى انك عاوز تاخذ **access** على جهاز اخر فبقولك تعالى نستخدم ال **Icmp protocol** وهو المسؤول عن ال **ping** على الشبكة و هتلاقى نفسك استخدمت ال **Icmp** وانت بتعمل **ping** او **trace route** او **ping** وزى منتا عارف ان ال **ping** بنسخدمه عشان نشوف ال **destination** ال عاوزين نتواصل معاه متصل بالانترنت ولا لاء فبنستخدم **ping** ال اسمها **tool** ال بتسخدم ال **Icmp** عشان تعمل كدا ولو جالك **reply** من الجهاز يبقا كدا موجود فعلا والعكس صحيح



Command Prompt

```
C:>ping -n 10 -l 2000 www.meridianoutpost.com

Pinging www.meridianoutpost.com [72.47.244.140] with 2000 bytes of data:
Reply from 72.47.244.140: bytes=2000 time=73ms TTL=55
Reply from 72.47.244.140: bytes=2000 time=65ms TTL=55
Reply from 72.47.244.140: bytes=2000 time=66ms TTL=55
Reply from 72.47.244.140: bytes=2000 time=72ms TTL=55
Reply from 72.47.244.140: bytes=2000 time=69ms TTL=55
Reply from 72.47.244.140: bytes=2000 time=69ms TTL=55
Reply from 72.47.244.140: bytes=2000 time=70ms TTL=55
Reply from 72.47.244.140: bytes=2000 time=68ms TTL=55
Reply from 72.47.244.140: bytes=2000 time=65ms TTL=55
Reply from 72.47.244.140: bytes=2000 time=68ms TTL=55

Ping statistics for 72.47.244.140:
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 65ms, Maximum = 73ms, Average = 68ms

C:>_
```

- وممكن نستخدم ال **Trace route** معين باستخدام بروتوكول **ICMP** عشان نعرف ال **gate ways** ال عدي عليها اثناء رحله وصوله لل **destination** زي مثلا انك تعمل **google** لل **route**

```
C:>tracert google.com

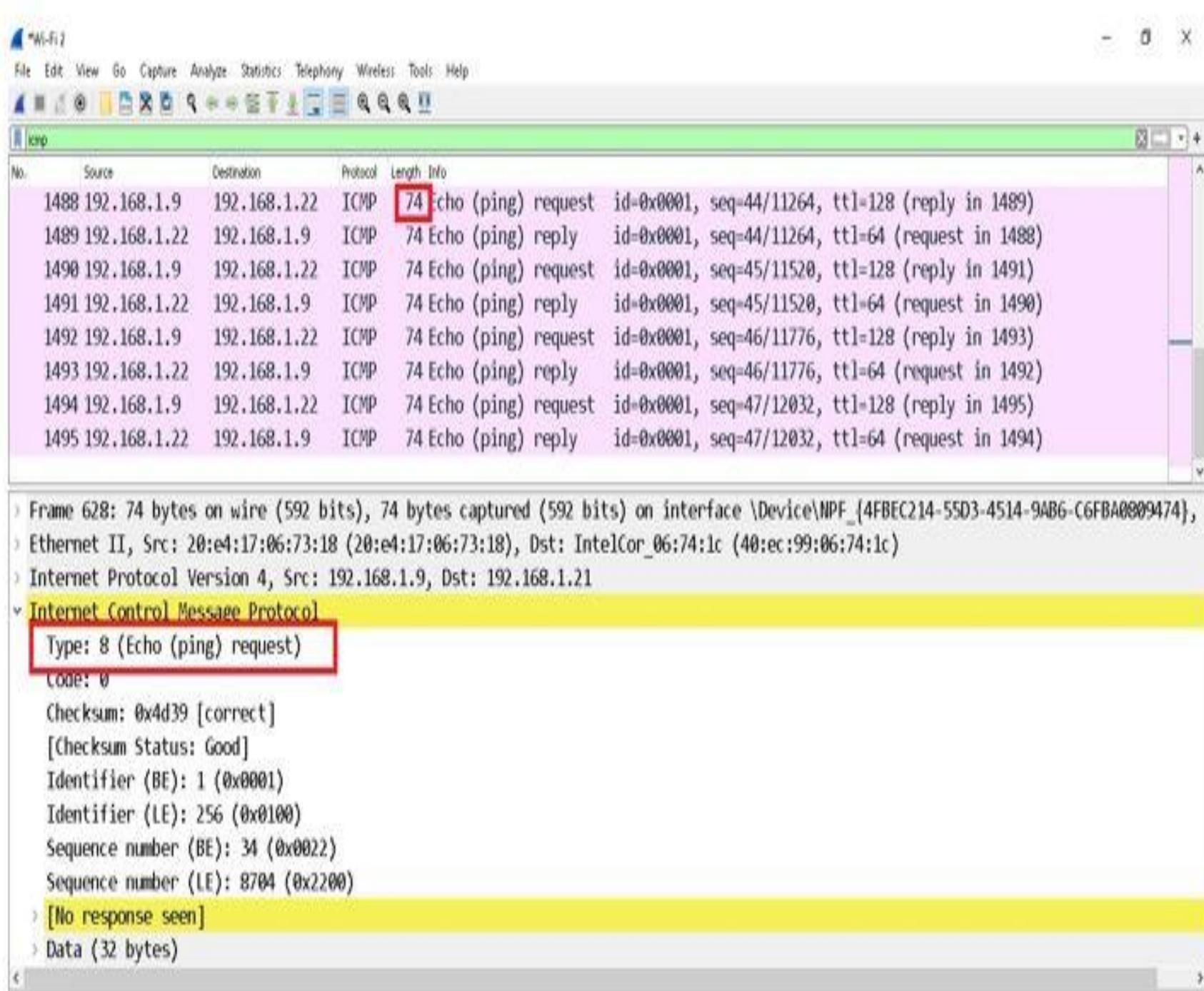
Tracing route to google.com [64.233.167.99]
over a maximum of 30 hops:
 1    <1 ms      <1 ms      <1 ms    10.9.9.1
 2    <1 ms      <1 ms      <1 ms  66.252.236.25.appliedi.net [66.252.236.25]
 3        3 ms      3 ms      3 ms  66.252.236.1.appliedi.net [66.252.236.1]
 4        1 ms      <1 ms      <1 ms  fc-agg1.fastcolo.net.97.82.216.in-addr.arpa [216
.82.97.33]
 5        2 ms      2 ms      1 ms  fc-nap-core-1.appliedi.net.97.82.216.in-addr.arp
a [216.82.97.209]
 6        2 ms      2 ms      1 ms  nap01.mia01.appliedi.net [66.165.166.193]
 7        2 ms      2 ms      1 ms  t6-0-0.core2.mia.terremark.net [66.165.161.198]
 8        2 ms      2 ms      2 ms  core1-1-0-0.mia.net.google.com [198.32.124.133]
 9        28 ms     18 ms      17 ms  72.14.238.57
10        42 ms     41 ms      40 ms  64.233.175.98
11        41 ms     41 ms      44 ms  66.249.94.133
12        52 ms     52 ms      39 ms  64.233.175.26
13        42 ms     41 ms      42 ms  py-in-f99.google.com [64.233.167.99]

Trace complete.

C:>>
```

- تعالى نشوف مع بعض شكل ال **Normal Icmp** هنلاقي ان

{ 0 } **code** ال **echo request** بتابعه { 8 } وال **type** بتابعه **ICMP** معين بيميزه عن غيره من ال **requests** الاخري عشان ال **type** **network** فلازم تكون عارف ال **normal** **request** من ال **packets** ودا هنشوفه من خلال امثله ال **Icmp** **header** ونبص فال **capture** **shark**



- تعالى نبص على ال **type echo reply** هتلاقی ان ال **type** بتابعه { 0
وال **threat hunter** لو لقيت غير ... يبقا انت ک **code** بتابعه { 0 ...
کدا تعرف انها **suspicious packet** ومع الوقت هتلاقی نفسك حفظت
الارقام دي متسلسل هم

ودي معظم ال **default Icmp types** بال **codes** بتابعتها ال
هتحاجها اثناء عملک وال هتشوفها كتير

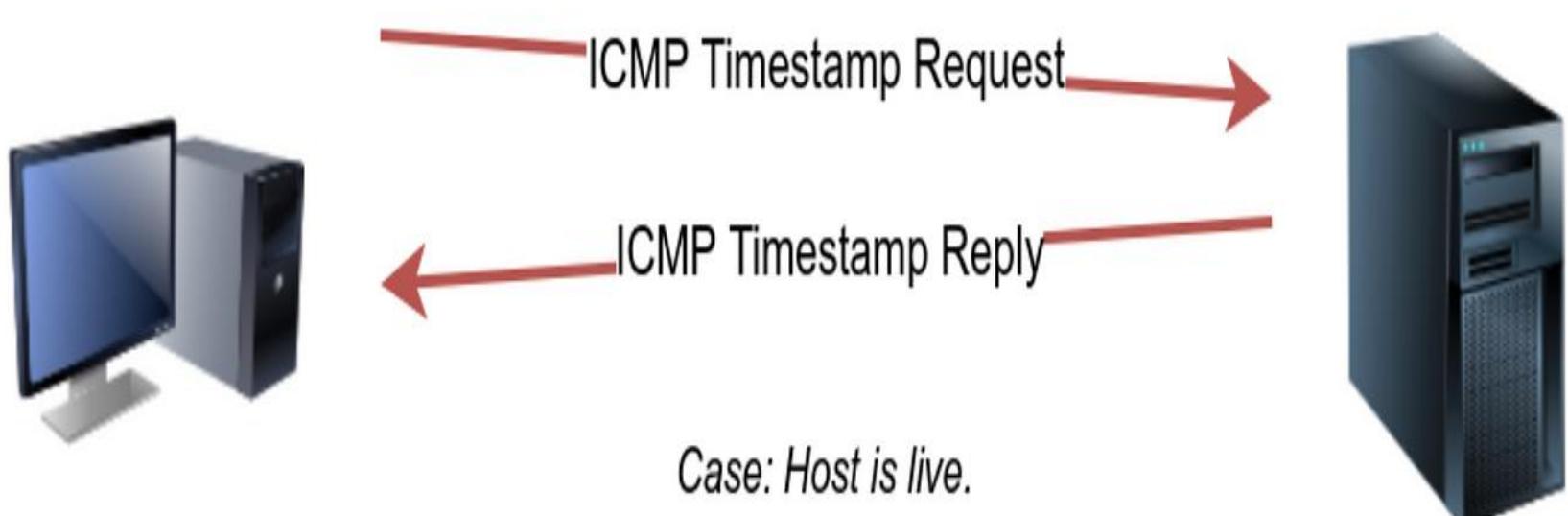
ICMP Type	Code	Description
0	0	echo reply (to ping)
3	0	destination network unreachable
3	1	destination host unreachable
3	2	destination protocol unreachable
3	3	destination port unreachable
3	6	destination network unknown
3	7	destination host unknown
4	0	source quench (congestion control)
8	0	echo request
9	0	router advertisement
10	0	router discovery
11	0	TTL expired
12	0	IP header bad

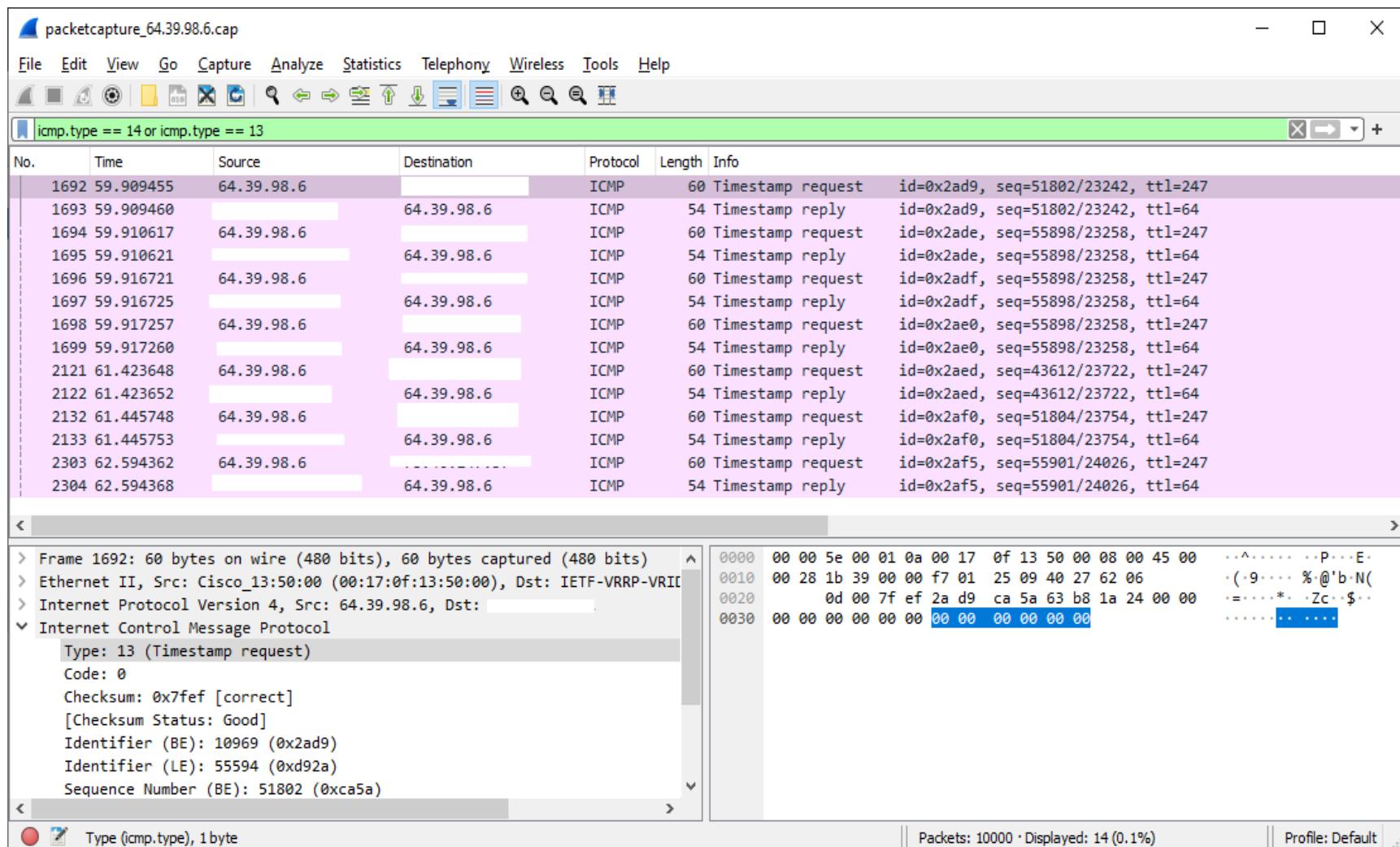
- تعالى نبص ع ال **Icmp suspicious Icmp** هتلاقی ان ال **data packets** بتتبعت بشكل كبير فانت تشک علطول ان دا ممكن يكون **traffic** تسریب للداتا من عندك فحجم ال **traffic** ال عندك **exfiltration** فالشبکه كبير او مش دا المعتاد بتاع ال **traffic** ال بينتقل عندك فالشبکه من ال **source** لـ **destination** وبرضه تبص على ال **length** فال

100 هو حجم ال traffic هتلاقیه ف المعتاد بیبقي wire shark انما لو لقیت traffic طوله 1000 مثلا فتبده تشك فيه معقوله فيه حد بیعمل data بالحجم دا ل packet واحده ؟؟ فتعرف ان دا ICMP cover على شكل exfiltration attack وبيمشي ف tunnel detect فتعمله traffic دی.

- لازم تاخد بالک برضه من ال unusual types / codes من خلال ال time stamp request عشان دا بیتبعد ما بین ال servers فقط ... مینفعش pc عادي بیعنه لان دا هيكون attack بیبقي عاوز يعرف ال time ال عند ال server عشان لو نزل اي target exploit يعرف ي zero day vulnerable فلازم تاخد بالک من ال Icmp requests المسموح بیها والغير مسموح packets هتلاقیها بتلونلك ال wire shark بیها ودا انت هتلاقیها فال attacker بیعملها ازاي

nmap -PP -sn TARGET





- عدنا تاني ممكن **attacker** يقدر يعمله ال **attack** من خلال ال

Icmp ال هو ممكن نستخدم ال **ICMP** و هو ال **Smurf Attack**

عشان نعمل **DDOS Attack** ال هو هجمات الحجب من الخدمه

بمعنى ان ال **spoofing** هيعمل **attacker** لـ **IP** بتاع ال

عنه و هيقوم باعت رساله **broad cast** باستخدام ال **victim**

بتاع **victim** لكل الناس الموجودين معاه فالشبكه **address**

وبالتالي لما تيجي الاجهزه او السيرفرات ترد على ال **requests** ال

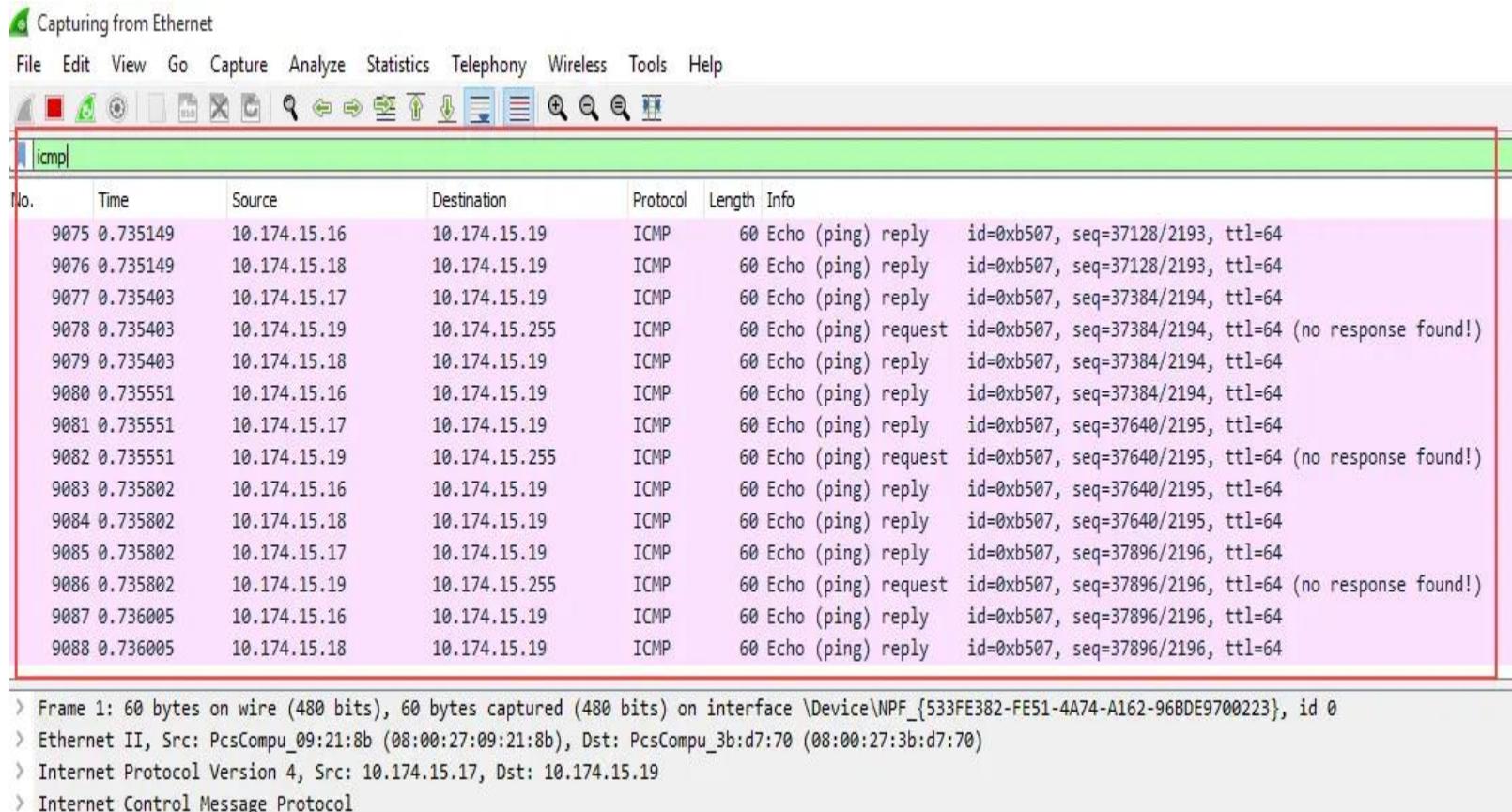
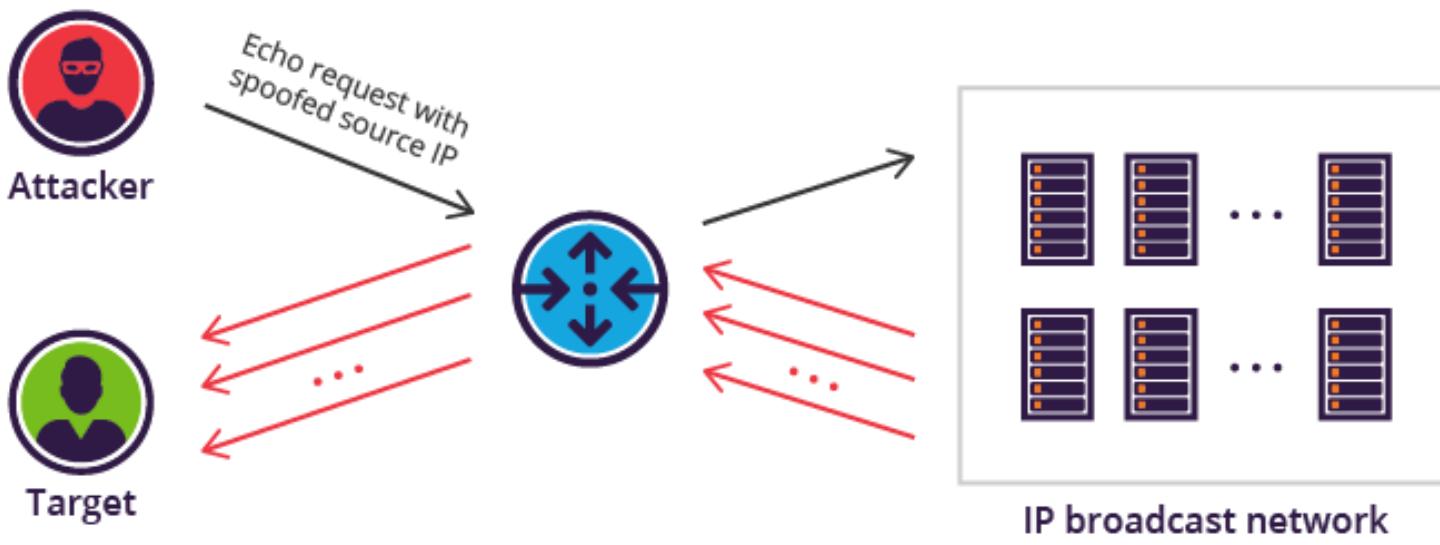
جاتلها هترض ع الجهاز الحقيقي وبالتالي مش هيستحمل يرد على

كميه ال **network** ال بتجيله وهيقع من ال **Reply's** بتاعتك

ويحصله **DDOS**

- ودا مثال لـ **wire shark** الموجود فال **capture** بيوضح ال

وهتلaci ان **Ip** 19 بيعت رساله **broad cast** للشبكه كلها **attack**



- عدنا **attack** تاني اسمه **ICMP tunneling attack** بمعني

انى ممکن استخدم **traffic** ال **ICMP** عشان احمل جواه

ips **firewall** او ال **IDS** يكون تاني يكون **malicious traffic**

وا ال **IDS** يكتشفه زى **http traffic** احطه جوا **tunnel** ل

attacker وا ال **ICMP traffic** والا داه ال ممکن يستخدمها ال

عشان يعمل ال **attack** دا اسمها **ptunnel** ودا انت ک

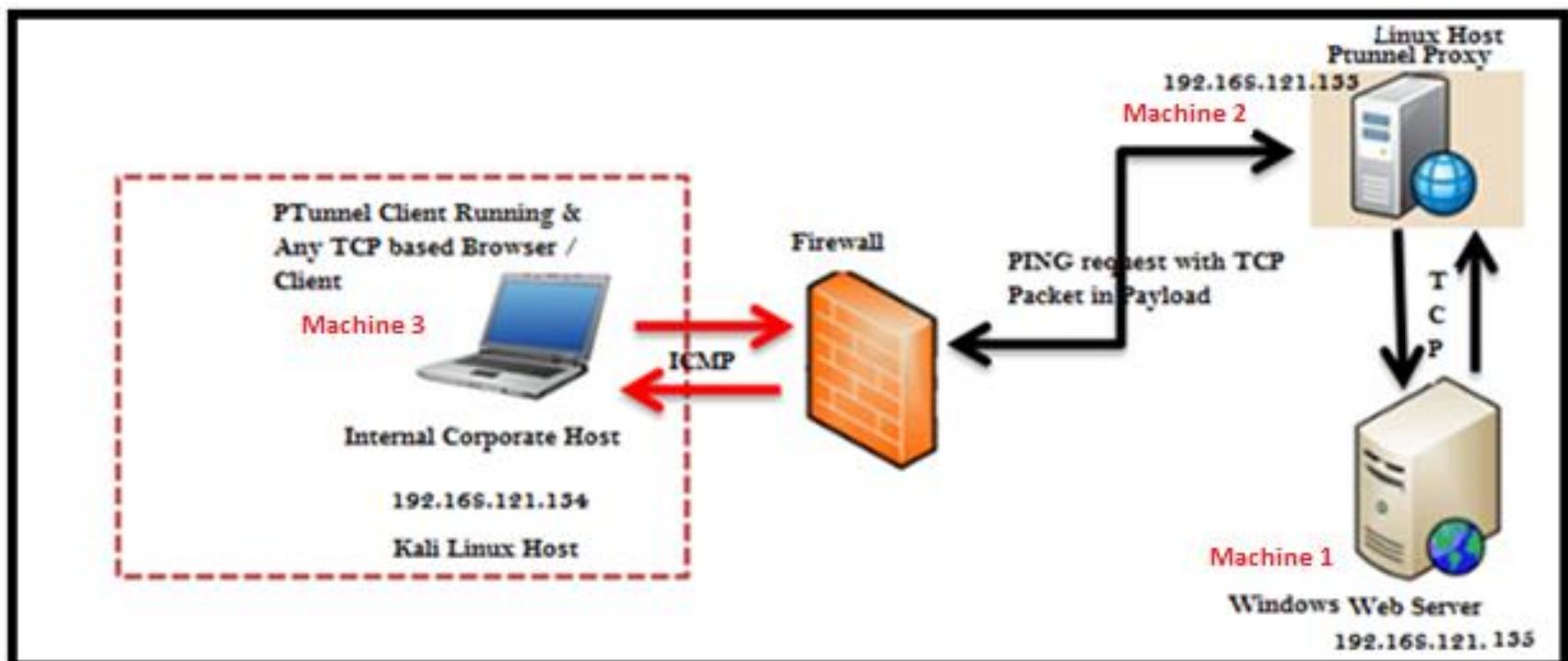
تقدر تكتشفه عن طریق ال **length** بتاعت ال **threat hunter**

وانت بتحلله فال **wire shark** هتلافي حجم ال **packet**

بیختلف كل مره وبیزید عن ال قبلها وهنسوف دا فلامثله **packets**

مع بعض وانت بتحلل ال **wire shark** فال **packets** بص

على جزء ال **data** لان اداه ال **ptunnel** ليها **sequence** معين
 ال بتغلف فيه ال **traffic** وهي تتبعه بتلاقي ال **sequence**
 فجزء ال **data** ودا بيكون الرقم ال لو شفته تعرف ان ال **traffic**
 دا استخدم ال **ptunnel** ال هو {**0xD5200880**} و هتلاقيني
Ptunnel حاططاك مثال يوضح ال



- ودا مثال لـ **Icmp tunneling** يوضح ال **ptunnel tool**
 المكتوب بيـه الاداه **attack** مكتوبه بلـغـه ال **C** لو فتحـتـ الـكـوـد
 suspicious المكتوب بيـه الاداه على اي **editor** هـتـلـاقـيـ الـكـوـدـ الـ **wire shark** فال **packet** فـالـ **suspicious** يـبـقـيـ دـاـ **investigation** وـانتـ بـتـعـمـلـ

```

191 #include <sys/types.h>
192 // Periodic packets after this interval (in seconds)
193 #define CHECKED_INTERVAL 1.5
194
195 /* ping_tunnel_pkt_t: This data structure represents the header of a ptunnel
196 packet, consisting of a magic number, the tunnel's destination IP and port,
197 as well as some other fields. Note that the dest IP and port is only valid
198 in packets from the client to the proxy.
199 */
200
201 typedef struct {
202     uint32_t      magic;           // magic number, used to identify ptunnel packets
203     uint32_t      dest_ip;         // destination IP and port (used by proxy to figure
204     uint16_t      dest_port;       // out where to tunnel to)
205     uint8_t       state;          // current connection status: see constants above.
206     uint8_t       ack;            // sequence number of last packet received from other end
207     uint16_t      data_len;        // length of data buffer
208     uint16_t      seq_no;         // sequence number of this packet
209     uint32_t      id_no;          // id number, used to separate different tunnels from each other
210     char          data[0];        // optional data buffer
211 } __attribute__((packed)) ping_tunnel_pkt_t;

```

Time	Source IP	Destination IP	Protocol	Source Port	Destination Port	Information	Details
1 6.000000	192.168.1.100	192.168.1.100	ICMP	78	Echo (ping) request	id=0x9a32, seq=0/0, ttl=64 (request in 1)	
2 6.000048	192.168.1.4	192.168.1.100	ICMP	78	Echo (ping) reply	id=0x9a32, seq=0/0, ttl=64 (request in 1)	
3 6.014309	192.168.1.4	192.168.1.100	ICMP	82	Echo (ping) reply	id=0x9a32, seq=0/0, ttl=64	
4 6.015240	192.168.1.100	192.168.1.4	ICMP	82	Echo (ping) request	id=0x9a32, seq=1/256, ttl=64 (reply in 5)	
5 6.015262	192.168.1.4	192.168.1.100	ICMP	82	Echo (ping) reply	id=0x9a32, seq=1/256, ttl=64 (request in 4)	
6 6.015319	192.168.1.4	192.168.1.100	ICMP	78	Echo (ping) reply	id=0x9a32, seq=1/256, ttl=64	

Code: 0
 Checksum: 0xe3d3 [correct]
 [Checksum Status: Good]
 Identifier (BE): 39474 (0x9a32)
 Identifier (LE): 12954 (0x329a)
 Sequence number (BE): 0 (0x0000)
 Sequence number (LE): 0 (0x0000)
 [Response frame: 2]
 * Data (28 bytes)

هتبص هنا فال **request** هتلaci فاl **data** تحت القيمه ال
ال بتعادل قيمه او كود ال **ptunnel** ال قولنا عليه فوق
انه **suspicious** هتلaci ال **wire shark** مطلعهولك فجزء ال
Data

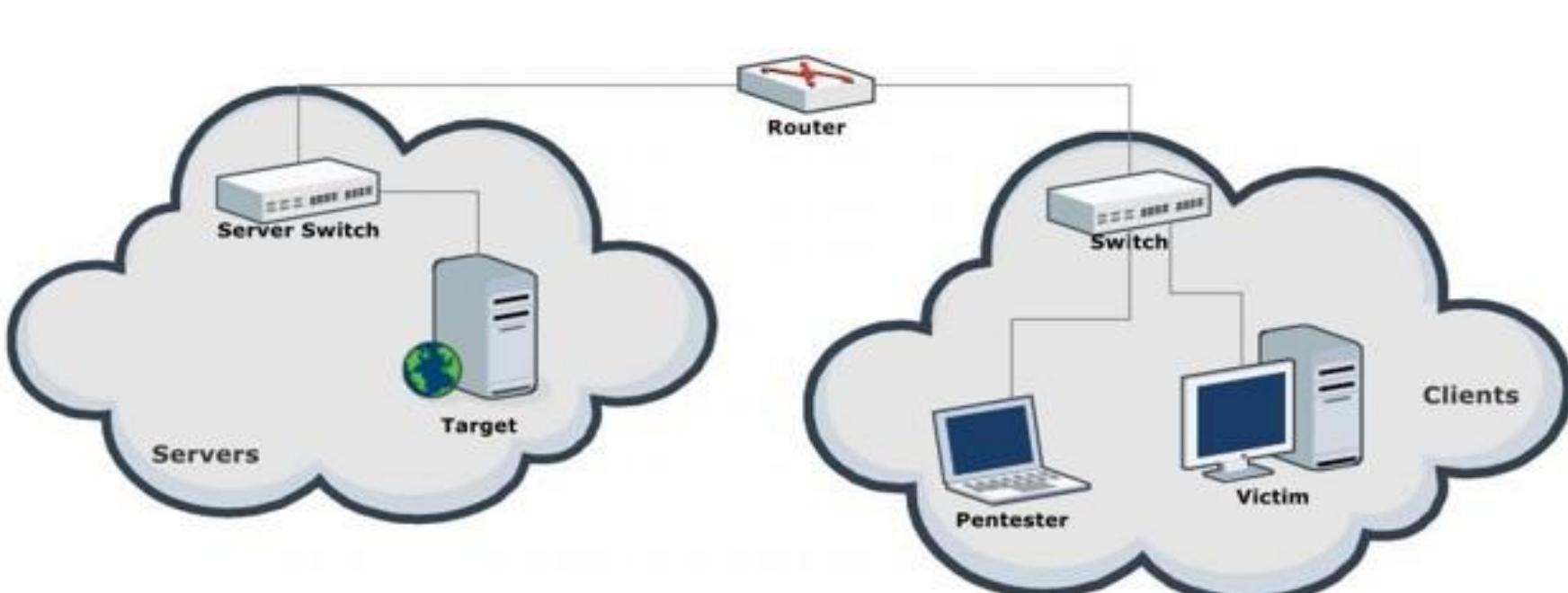
دا برضه مثال ICMP wire shark خدناها بال packet tunneling بتوضح ال

1	0.000000	192.168.1.100	192.168.1.4	ICMP	70 Echo (ping) request	id=0x9a32, seq=0/0, ttl=64 (reply in 2)
2	0.000046	192.168.1.4	192.168.1.100	ICMP	70 Echo (ping) reply	id=0x9a32, seq=0/0, ttl=64 (request in 1)
3	0.014989	192.168.1.4	192.168.1.100	ICMP	82 Echo (ping) reply	id=0x9a32, seq=0/0, ttl=64
4	0.015246	192.168.1.100	192.168.1.4	ICMP	82 Echo (ping) request	id=0x9a32, seq=1/256, ttl=64 (reply in 5)
5	0.015262	192.168.1.4	192.168.1.100	ICMP	82 Echo (ping) reply	id=0x9a32, seq=1/256, ttl=64 (request in 4)
6	0.015319	192.168.1.4	192.168.1.100	ICMP	70 Echo (ping) reply	id=0x9a32, seq=1/256, ttl=64
7	0.015439	192.168.1.4	192.168.1.100	ICMP	94 Echo (ping) reply	id=0x9a32, seq=2/512, ttl=64
→	8 0.015612	192.168.1.100	192.168.1.4	ICMP	170 Echo (ping) request	id=0x9a32, seq=2/512, ttl=64 (reply in 9)
←	9 0.015622	192.168.1.4	192.168.1.100	ICMP	170 Echo (ping) reply	id=0x9a32, seq=2/512, ttl=64 (request in 8)
10	0.017822	192.168.1.4	192.168.1.100	ICMP	86 Echo (ping) reply	id=0x9a32, seq=3/768, ttl=64
11	0.017958	192.168.1.100	192.168.1.4	ICMP	94 Echo (ping) request	id=0x9a32, seq=3/768, ttl=64 (reply in 12)
12	0.017977	192.168.1.4	192.168.1.100	ICMP	94 Echo (ping) reply	id=0x9a32, seq=3/768, ttl=64 (request in 11)
13	0.018171	192.168.1.4	192.168.1.100	ICMP	74 Echo (ping) reply	id=0x9a32, seq=4/1024, ttl=64
14	0.018282	192.168.1.100	192.168.1.4	ICMP	74 Echo (ping) request	id=0x9a32, seq=4/1024, ttl=64 (reply in 15)
15	0.018295	192.168.1.4	192.168.1.100	ICMP	74 Echo (ping) reply	id=0x9a32, seq=4/1024, ttl=64 (request in 14)
16	0.018440	192.168.1.4	192.168.1.100	ICMP	690 Echo (ping) reply	id=0x9a32, seq=5/1280, ttl=64

هتبص هنا فال هتلaci الوضع مش تمام لان فيه reply كتير
علي request واحد فتبده تشك فال packet وكمان هتلaci حجم ال
packet او ال length بتاعت ال packet مختلفه

هتلaci مرة ال echo request جاي ب 60 وال reply بتاعه جاي ب 690 !!! غريبه دي !!! هو مش المفروض يكون نفس ال length packet عادي!! فتبده تشک برضه فال ping وهكذا تقدر تعمل suspicious packet فال investigation traffic .

ال abuse Icmp redirect attack بمعنى الاخير معانا هو ال connection لوعنك جهازين عاوزين يعملو مع بعض طريقي اسهل بيبدئ يبعث رساله source redirect لـ destination ... تمام كدا ... تخييل ان ال attacker هو ال يبعث رساله source على اساس انه الرواتر يعني ويقوله source عندي طريقي اسهل عشان توصل لـ destination تعالى وانا هوديك ليه .. فجهاز ال traffic هيروحله فال sniffing هيعمله attacker source - ودا مثال لـ Icmp redirect Attack من ال wire shark وطريقه عمله برضه



ودا شكل ال ICMP Redirect Attack من ال wire shark

The screenshot shows a network capture in Wireshark. The list view displays several ICMP Redirect messages (Type 5) from source IP 10.100.13.1 to destination IP 10.100.13.126. The details view for the third message shows the following information:

- Frame 3: 82 bytes on wire (656 bits), 82 bytes captured (656 bits)
- Ethernet II, Src: VMware_a1:cb:34 (72:9b:2f:a0:90:91), Dst: VMware_a1:cb:34 (00:50:56:a1:cb:34)
- Internet Protocol Version 4, Src: 10.100.13.1, Dst: 10.100.13.126
- Internet Control Message Protocol
 - Type: 5 (Redirect)
 - Code: 1 (Redirect for host)
 - Checksum: 0x3dfe [correct]
 - [Checksum Status: Good]
- Gateway address: 10.100.13.20
- Internet Protocol Version 4, Src: 10.100.13.126, Dst: 10.23.56.100
- Transmission Control Protocol, Src Port: 55555, Dst Port: 80

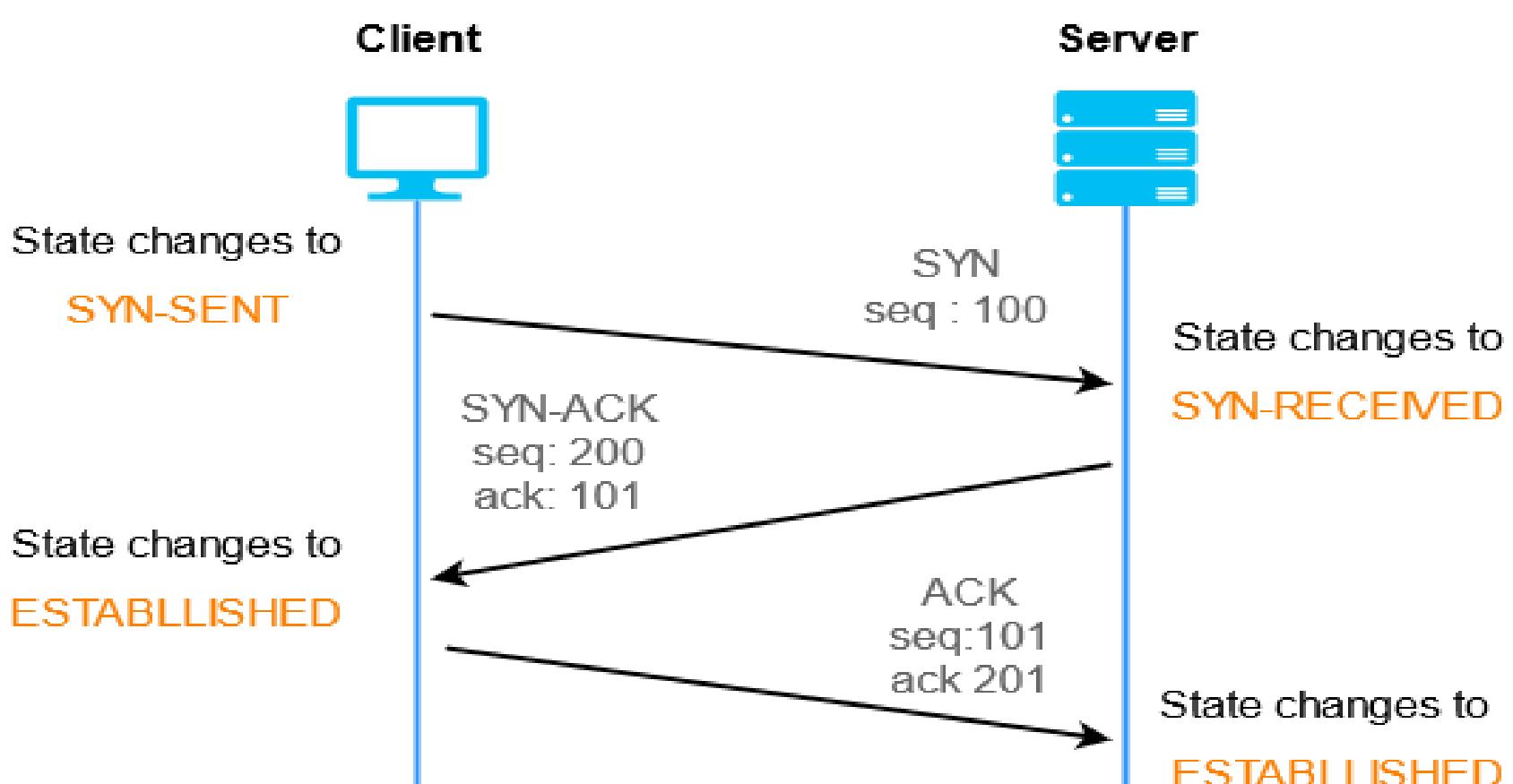
و هتلاظ رسائل redirect كتير ورا بعض و هتبص تلاقي ال IP ال هو 13.1 كان بيكلم ال IP ال هو 13.126 و فجأه دخل IP جديد ال هو 13.20 بتاع ال attacker و بيقول الجهاز 13.1 لو عاوز تروح للرواتر تعالىي انا هوديك ليه اسرع فعمل Redirect

TCP Traffic:

- اختصار ال transmission control protocol وهو المسؤول عن انه يوصل ال source او ال data packets ويهدلها ويوصلها من ال destination لـ packet error و هو حصل لـ data alert بالكلام دا و بيعرفك ان ال data موصلتشر لـ destination او حصلها lose في الطريق و دا انت بتشوفه اما بتيجي تبع Gmail لو وصل هتلافي ال email ببيعتاك رساله بالكلام دا اما لو موصلتشر هتلافي بيديك alert بالكلام ده برضه

- فدا يعتبر ال **source packet** من ال **delivery** بتاع ال **destination**

- ال **TCP** قبل ميرسل اي **data** بيقوم بعمليه ال **3way hand shake** عشان يطمئن ان ال **data** وصلت وكله تمام وعشان ي保證 **connection** ما بين ال **2 devices**



- خد بالك من ال **normal Tcp** هتلافي ال **syn** ال بيتبع زى م الصورة موضحة بيترد عليه بال **Syn-Ack** وبعد كدا ال **Ack** ويحصل الاتصال عادي

- عشان فال **suspicious Tcp** هتلافي **Syn** بتتبعه كتير بدون ميرجعها رد ال هو **Ack** فدا **scanning behavior** لـ **tool** **Nmap** زي ال **single host** وهتلافي برضه **multiple ports** والافتراض انك بتاعه ل **destination**

تطع من **port** معين تروح ل **port** معين مش لكل ال **ports** هو
دا بتكلم عليه ودي طريقة شغل ال **scanning behavior**

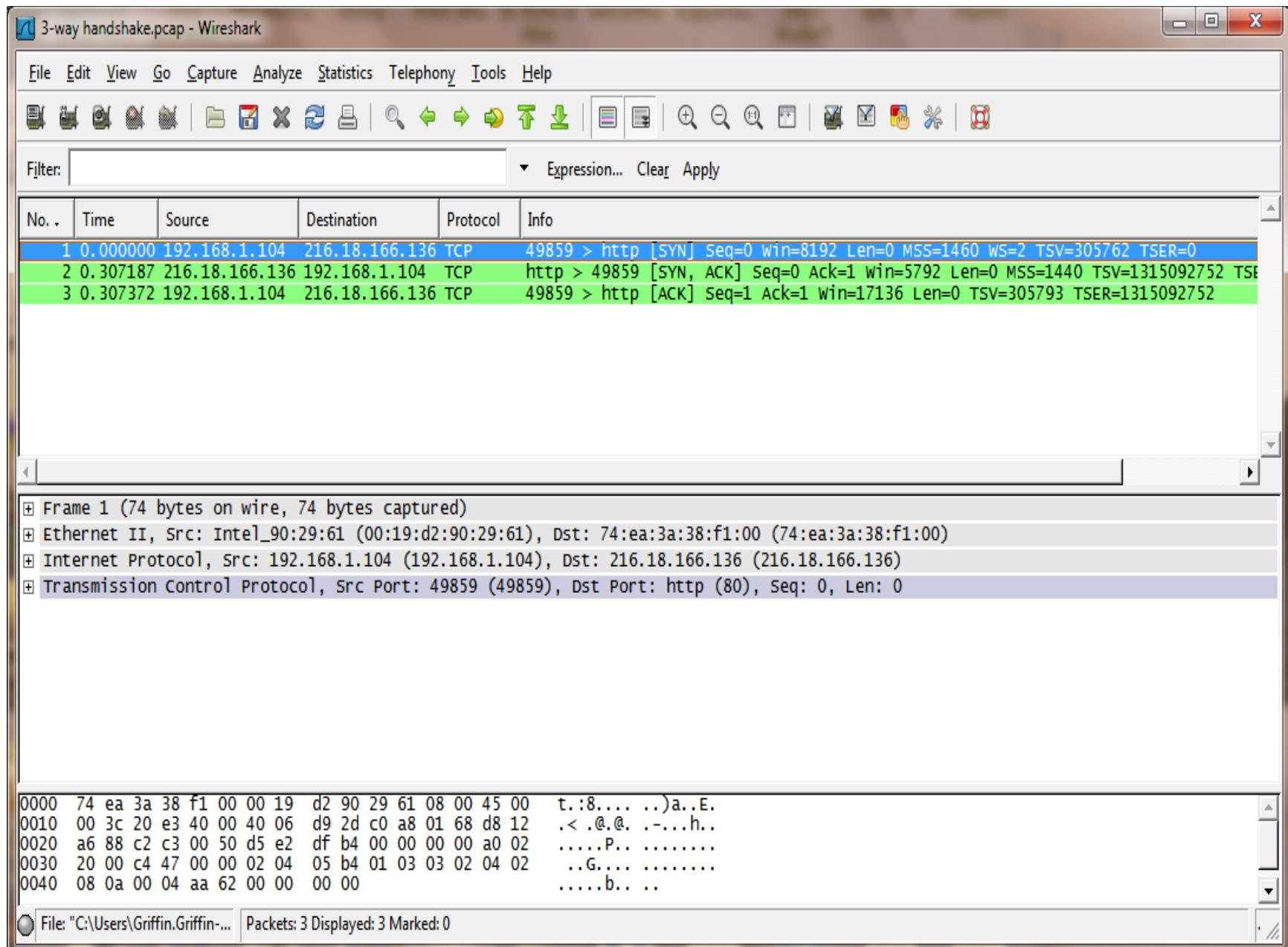
Nmap

- وبرضه لو شفت **IP** معين بيروح ل **multiple ip** تعرف برضه انت ک
عندک **scanning behavior** ان دا **threat hunter**
والمعلومات دي بتخليك سابق ال **attackers** بخطوة لانک لو شفت
عندک **فالشبکه scanning behavior** مش هستنی اما ال
penetration testing يروح يکمل عليك باقی خطوات ال **attacker**
فمجرد انک سابقه بخطوة هتقدر تقلل من خطر ال **attack** عليك وتعمله
عندک **action** او **block** او **reset** او **drop**

- نقطه تانيه هتلaci برضه هتلaci فال **normal** ان عشان يحصل
عندک **FLAGS** هتلaci بنسخدم **connection establish**
وال **Ack** وال **Syn** ال هما ضروري عشان يحصل الاتصال

- انما ف ال **flags traffic** هتلaci **suspicious** تانيه
مهاش علاقه بال **connection** اساسا زى ال **reset** وال **finish**
فدا معناه انه مش ال فکدا في **normal Tcp behavior**
وهو **suspicious traffic**

- بص کدا ع الصوره دی هتشوف ال **normal** عشان تميز ال **suspicious**

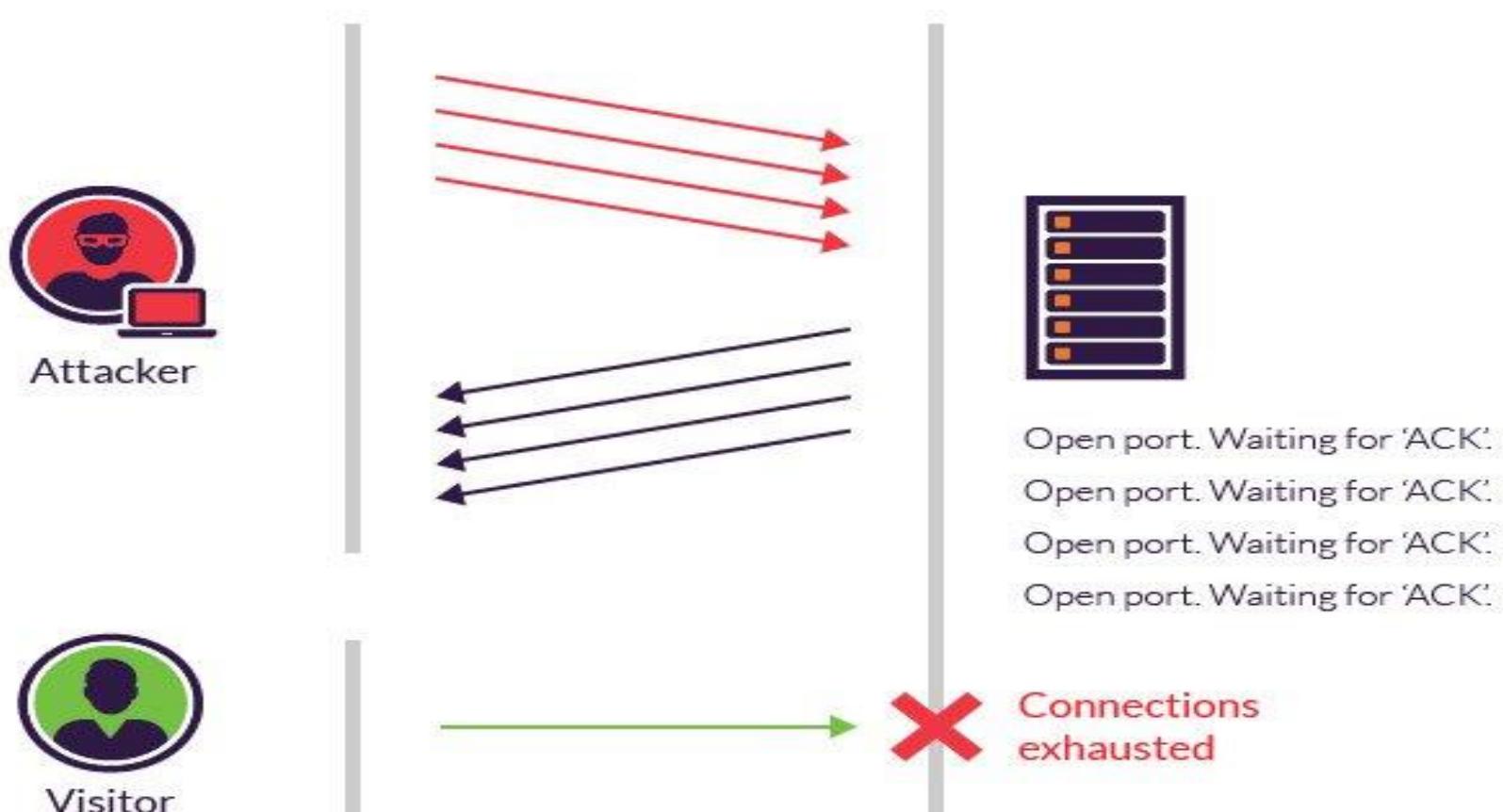


- هتلaci ال **Syn** شغال عادي بعت **normal Tcp** بعد کدا اترض عليه ب **Ack** وبعد کدا اترض عليه ب **Syn-Ack** والدنيا تمام .

تعالي نبع ع ال suspicious Tcp عشان نعرف نميذه

No.	Time	Source	Destination	Protocol	Length	Info
1. 0.000000000	184.21.221.175	10.10.10.6		TCP	54	2870 → 80 [SYN] Seq=0 Win=512 Len=0
2. 0.000113203	180.110.36.117	10.10.10.6		TCP	54	2871 → 80 [SYN] Seq=0 Win=512 Len=0
3. 0.000180211	58.232.251.64	10.10.10.6		TCP	54	2872 → 80 [SYN] Seq=0 Win=512 Len=0
4. 0.000250709	168.40.137.131	10.10.10.6		TCP	54	2873 → 80 [SYN] Seq=0 Win=512 Len=0
5. 0.000310804	92.253.225.15	10.10.10.6		TCP	54	2874 → 80 [SYN] Seq=0 Win=512 Len=0
6. 0.000365148	PcsCompu_99:b1:5	Broadcast		ARP	60	Who has 10.10.10.1? Tell 10.10.10.6
7. 0.000416478	101.43.92.96	10.10.10.6		TCP	54	2875 → 80 [SYN] Seq=0 Win=512 Len=0
8. 0.000479694	249.74.117.161	10.10.10.6		TCP	54	2876 → 80 [SYN] Seq=0 Win=512 Len=0
9. 0.000538088	224.221.126.3	10.10.10.6		TCP	54	2877 → 80 [SYN] Seq=0 Win=512 Len=0
10. 0.000595377	212.180.15.246	10.10.10.6		TCP	54	2878 → 80 [SYN] Seq=0 Win=512 Len=0
11. 0.000685119	53.84.12.250	10.10.10.6		TCP	54	2879 → 80 [SYN] Seq=0 Win=512 Len=0
12. 0.000761756	115.143.29.208	10.10.10.6		TCP	54	2880 → 80 [SYN] Seq=0 Win=512 Len=0
13. 0.000821969	141.188.194.69	10.10.10.6		TCP	54	2881 → 80 [SYN] Seq=0 Win=512 Len=0
14. 0.000894381	148.154.3.138	10.10.10.6		TCP	54	2882 → 80 [SYN] Seq=0 Win=512 Len=0
15. 0.000954276	240.70.161.136	10.10.10.6		TCP	54	2883 → 80 [SYN] Seq=0 Win=512 Len=0
16. 0.001012189	74.128.175.245	10.10.10.6		TCP	54	2884 → 80 [SYN] Seq=0 Win=512 Len=0
17. 0.001068786	48.116.104.50	10.10.10.6		TCP	54	2885 → 80 [SYN] Seq=0 Win=512 Len=0
18. 0.001124649	253.173.202.202	10.10.10.6		TCP	54	2886 → 80 [SYN] Seq=0 Win=512 Len=0
19. 0.001181709	168.247.131.71	10.10.10.6		TCP	54	2887 → 80 [SYN] Seq=0 Win=512 Len=0
20. 0.001237559	38.49.171.60	10.10.10.6		TCP	54	2888 → 80 [SYN] Seq=0 Win=512 Len=0
21. 0.001310497	247.255.115.123	10.10.10.6		TCP	54	2889 → 80 [SYN] Seq=0 Win=512 Len=0
22. 0.001371675	60.28.191.19	10.10.10.6		TCP	54	2890 → 80 [SYN] Seq=0 Win=512 Len=0
23. 0.001429062	57.110.175.235	10.10.10.6		TCP	54	2891 → 80 [SYN] Seq=0 Win=512 Len=0
24. 0.001500470	21.6.124.189	10.10.10.6		TCP	54	2892 → 80 [SYN] Seq=0 Win=512 Len=0

- هن بص هنا هنلاقي **syn request** كتير ورا بعض بدون الرد عليها فكدا دا **scanning behavior** لـ **port** وكمان هنلاقيه جاي من **different ports** وجاي من **Ip** واحد رايح ل **network** فدا ياكدلك انه **scanning** عندك ع ال **different Ip** وكمان هنلاقيه بيبعت طلبات **Syn** كتير ورا بعض ودا ال **attack** قولنا عليه ال اسمه **DDOS Attack** عشان يعملك **Syn flooding**



- سيناريو اخر لـ **suspicious** ... المفروض فال **normal** زي مقولنا يحصل **SYN** وبعده **ACK** وبعده **SYN-ACK** عشان يحصل ال **connection** هنلاقي بعض الحالات ان بيحصل **SYN** ثم بيحصل بعده ال **SYN-ACK** بس بيجيلك **RST** فالآخر بمعني رفض او قطع الاتصال ودا ال بيعمله ال **scanning tools** هو دا السيناريو الخاص بيها

No.	Time	Source	Destination	Protocol	Info
1 0.000000	10.0.0.107	10.0.0.221	TCP	1500-9999	[SYN] Seq=100 Win=8192 Len=0
2 0.000020	10.0.0.221	10.0.0.107	TCP	9999-1500	[SYN, ACK] Seq=1404263211 Ack=101 Win=26883
3 0.000325	10.0.0.107	10.0.0.221	TCP	1500-9999	[RST] Seq=101 Win=0 Len=0

first two handshake happen

Connection rested during final handshake process

```

> Ethernet II, Src: 00:0C:47:00:01:17 (00:0C:47:00:01:17), Dst: 00:0C:01:39:2E:17 (00:0C:01:39:2E:17)
> Internet Protocol Version 4, Src: 10.0.0.107 (10.0.0.107), Dst: 10.0.0.221 (10.0.0.221)
> Transmission Control Protocol, Src Port: 1500 (1500), Dst Port: 9999 (9999), Seq: 101, Len: 0
  Source Port: 1500 (1500)
  Destination Port: 9999 (9999)
  [Stream index: 0]
  [TCP Segment Len: 0]
  Sequence number: 101
  Acknowledgment number: 0
  Header Length: 20 bytes
  > ... 0000 0000 0100 = Flags: 0x004 (RST) ← RST flag Set
  Window size value: 0
  [Calculated window size: 0]
  [Window size scaling factor: -2 (no window scaling used)]

```

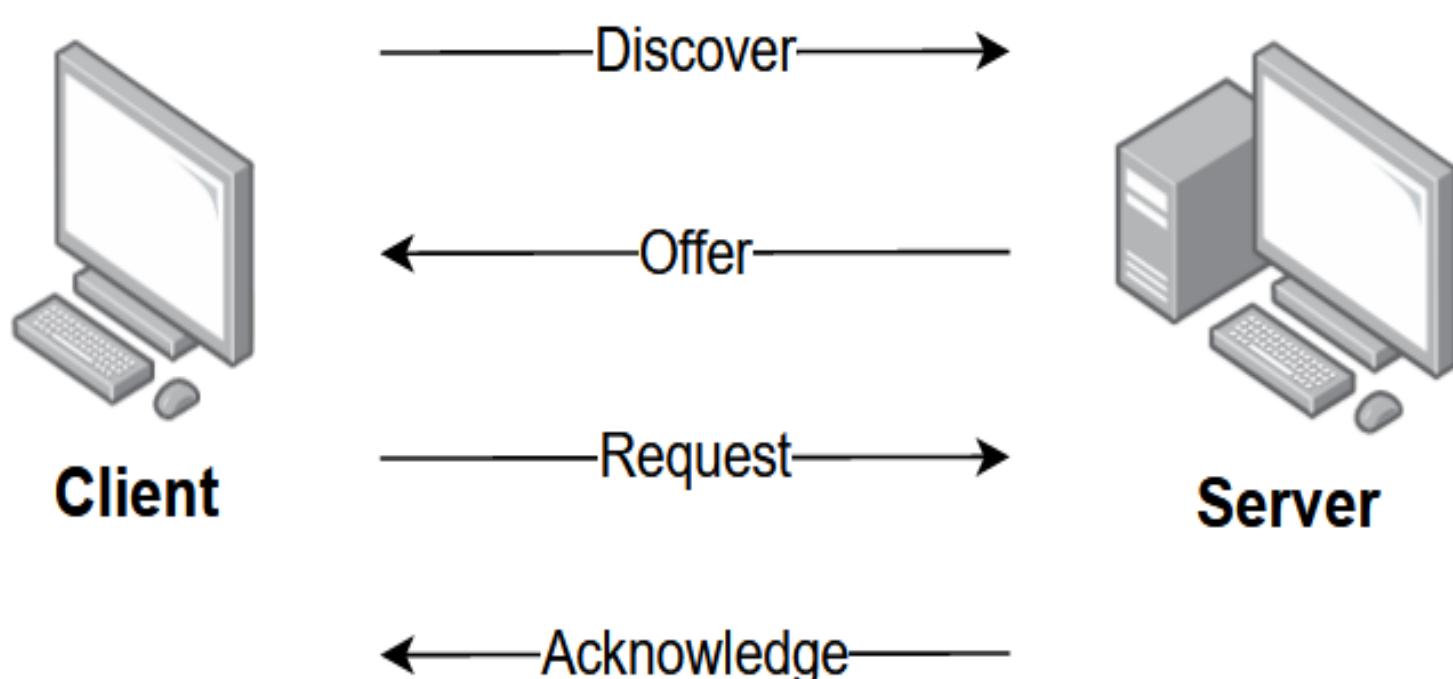
DHCP Traffic:

- اختصار dynamic host configuration protocol و دا البرتوكول المسؤول انه يوزع IP للاجهزه وال الموجوده عندك و دا بيستغل جوا ال LAN ال هي الشبكه الداخلية يعني .

احنا عارفين عشان ال user يحصل IP فيه طريقتين انه ياخذ Ip طبعا علي IP address assign manual ال انت واقع فيه و علي حسب ال هيقولك عليه subnet mask او الطريقه الثانيه والمشهورة network ال administrator اكتر انك تاخذ ال IP بشكل اتوماتيك عن طريق برتوكول ال DHCP ولازم يكون موجود ليه سيرفر على الشبكة اسمه DHCP Server

عشنان يوزع IP ع الاجهزه وممكن تلاقي ال server موجود عادي على IP او على router مش شرط بس اهم حاجه تاخد ال IP بشكل اتوماتيكي وبيشتغل على بورتات 67 , 68 وبيستخدم بروتوكول ال UDP من طبقه ال transport layer تمام كدا .

ال Discover بيشتغل ب process اسمها DORA ال هما و هننشر حهم برضه و acknowledgement و request و offer عشنان تبقا معايا قدام



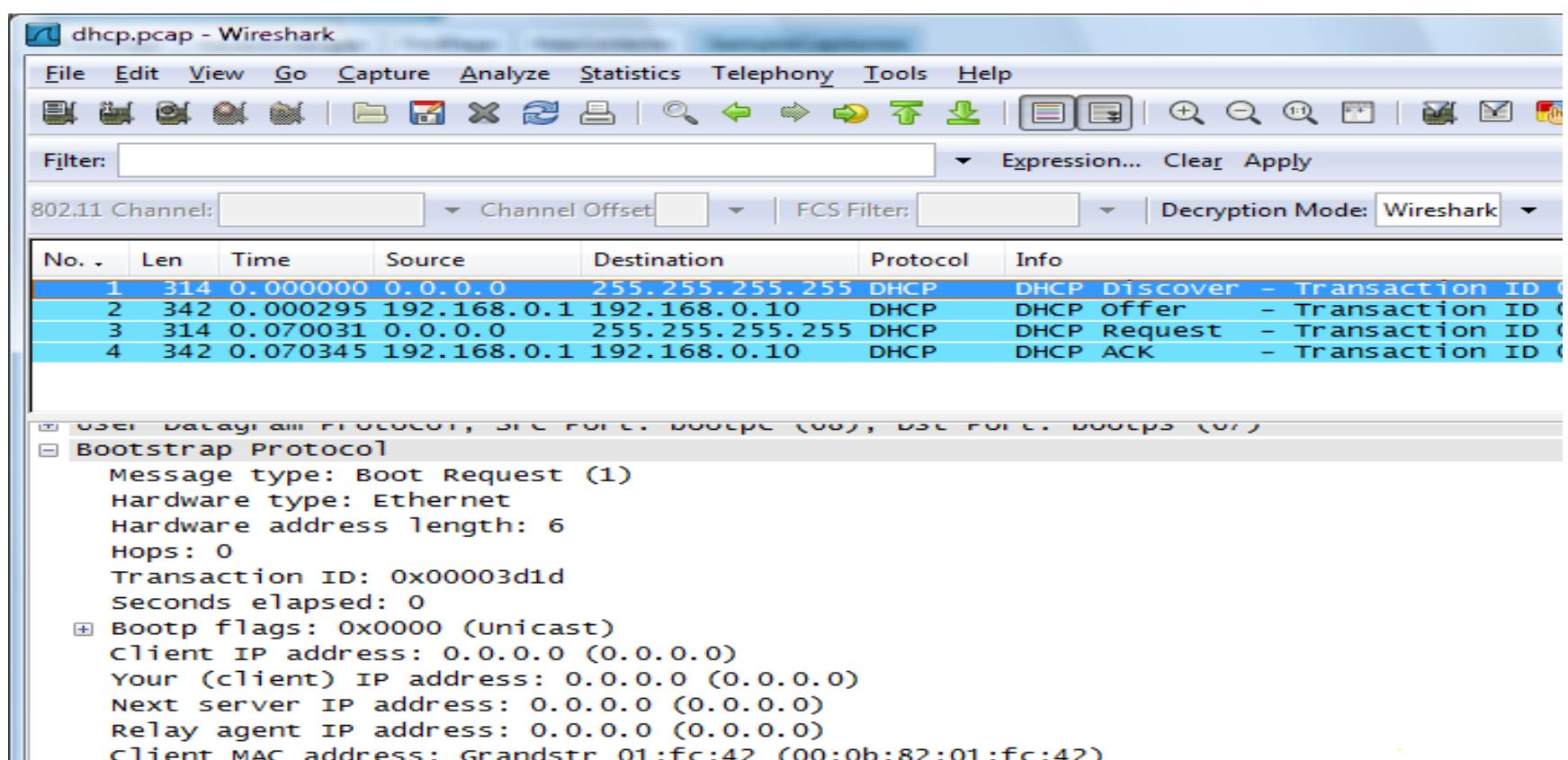
- هنفترض ان جهازك معهوش IP وعاوز يتواصل مع الناس ال معاه الشبكة فبيبدئ بيعت رساله ال broadcast DHCP discover بشكل mac كل الناس الموجودين معاه ع الشبكة ... بيعتلهم بواسطه ال الخاص بيده ويقول لهم ي جماعه عاوز عنوان IP على امل ان فالشبكه يسمعك ويرض عليك ويديك IP ... DHCP server

- المفروض خد بالك ان ال **DHCP server** يرض عليك برساله **offer** ويقولك خد ال **IP** دا شوفه مناسب ليك ولا لاء بس ممكن ال **spoofing** يرض بدل ال **server** ويعمله **attacker** بعدين .

- انت بترض ع ال **DHCP Request** بتفول بعد كدا برساله **server** لل **IP** احجزلي ال **server** دا تمام عشان محدث تاني ياخده

- وقتها ال **server** بيرض عليك برساله **DHCP process** بمعنى انه هيقوم باتمام ال **Acknowledgement** وهيجزلك ال **IP** ال وافت عليه لمده معينه ال بيحدها هو ال **DHCP server** والمده دي بنسميها **administrator** بتعالى وقت محدد ليك لاستخدام ال **IP** وبعد كدا هيتوقف .

..... **Normal DHCP**



- هبص ف packet رقم 1 هنلاقي الجهاز مكنش معاه IP بعت رساله DHCP للشبكه كلها ورض عليه ال IP ال هو 0.1 بال DORA وهذا الى نهايه ال DORA ال شرحنها فوق ودا السليم والمعتاد .

```

> Frame 1: 314 bytes on wire (2512 bits), 314 bytes captured (2512 bits)
> Ethernet II, Src: Grandstr_01:fc:42 (00:0b:82:01:fc:42), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255
User Datagram Protocol, Src Port: 68, Dst Port: 67
# Bootstrap Protocol (Discover)
  Message type: Boot Request (1)
  Hardware type: Ethernet (0x01)
  Hardware address length: 6
  Hops: 0
  Transaction ID: 0x00003d1d
  Seconds elapsed: 0
  Bootp flags: 0x0000 (Unicast)
  Client IP address: 0.0.0.0
  Your (client) IP address: 0.0.0.0
  Next server IP address: 0.0.0.0
  Relay agent IP address: 0.0.0.0
  Client MAC address: Grandstr_01:fc:42 (00:0b:82:01:fc:42)
  Client hardware address padding: 000000000000000000000000
  Server host name not given
  Boot file name not given
  Magic cookie: DHCP
# Option: (53) DHCP Message Type (Discover)
  Length: 1
  DHCP: Discover (1)
  Option: (61) Client Identifier
> Option: (50) Requested IP Address
> Option: (55) Parameter Request List

```

- فالصورة ال فاتت موضحه شكل ال discover جوا ال packet ماشي ازاي وهتلاقي ال DHCP discover وهو بيطلب ال user بيطلب معاه حاجات تانيه زى ال subnet mask و حاجات تانيه كتير بس الاساسي هو طلب ال request او ال بيان لـ Boot strap و هتلاقي packet فال DHCP server Discover .

- ممكن تلاقي بروتوكول ال DHCP ليه مسمى اخر وهو عشان لو شفته فال wire shark

```
# Option: (53) DHCP Message Type (Discover)
    Length: 1
    DHCP: Discover (1)
# Option: (61) Client identifier
    Length: 7
    Hardware type: Ethernet (0x01)
    Client MAC address: Grandstr_01:fc:42 (00:0b:82:01:fc:42)
# Option: (50) Requested IP Address
    Length: 4
    Requested IP Address: 0.0.0.0
# Option: (55) Parameter Request List
    Length: 4
    Parameter Request List Item: (1) Subnet Mask
    Parameter Request List Item: (3) Router
    Parameter Request List Item: (6) Domain Name Server
    Parameter Request List Item: (42) Network Time Protocol Servers
# Option: (255) End
    Option End: 255
Padding: 00000000000000
```

تعالى نشوف الخطوة ال بعدها جوا ال packet وهي ال DHCPoffer

```
Frame 2: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits)
Ethernet II, Src: Dell_ad:f1:9b (00:08:74:ad:f1:9b), Dst: Grandstr_01:fc:42 (00:0b:82:01:fc:42)
Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.0.10
User Datagram Protocol, Src Port: 67, Dst Port: 68
Bootstrap Protocol (Offer)
    Message type: Boot Reply (2)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0x00003d1d
    Seconds elapsed: 0
    Bootp flags: 0x0000 (Unicast)
    Client IP address: 0.0.0.0
    Your (client) IP address: 192.168.0.10
    Next server IP address: 192.168.0.1
    Relay agent IP address: 0.0.0.0
    Client MAC address: Grandstr_01:fc:42 (00:0b:82:01:fc:42)
    Client hardware address padding: 000000000000000000000000
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
Option: (53) DHCP Message Type (Offer)
    Length: 1
    DHCP: Offer (2)
```

- هنبعص فال **packet** تحت هنلاقي ال **offer** بتاعها **type** وهنلاقيه بيعرض عليك تاخد ال **IP** { 0.10 } زي م الصورة موضحة .

هتلaci ف المثال دا ال **DHCP** بيرض عليك برضه بال حاجات الزياذه
ال كنت طلبتها مع ال **request** زي ال **subnet** **gate way** وال **lease Time** **mask**
وكل دا احنا لسه فال **normal** عشان متو هش مني !!

```

    # Option: (53) DHCP Message Type (Offer)
        Length: 1
        DHCP: Offer (2)
    # Option: (1) Subnet Mask
        Length: 4
        Subnet Mask: 255.255.255.0
    # Option: (58) Renewal Time Value
        Length: 4
        Renewal Time Value: (1800s) 30 minutes
    # Option: (59) Rebinding Time Value
        Length: 4
        Rebinding Time Value: (3150s) 52 minutes, 30 seconds
    # Option: (51) IP Address Lease Time
        Length: 4
        IP Address Lease Time: (3600s) 1 hour
    # Option: (54) DHCP Server Identifier
        Length: 4
        DHCP Server Identifier: 192.168.0.1
    # Option: (255) End
        Option End: 255
    Padding: 000000000000000000000000000000000000000000000000...

```

- هنبع ف الخطوة الثالثه هنلقي ال **DHCP request** خرج من عند ال **client** راح لل **server** وبيقوله انا موافق عال **IP** ال انت عطتهولي لـ **DHCP server** لك

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	0.0.0.0	255.255.255.255	DHCP	314	DHCP Discover - Transaction ID 0x3d1d
2	0.000295	192.168.0.1	192.168.0.10	DHCP	342	DHCP Offer - Transaction ID 0x3d1d
3	0.070031	0.0.0.0	255.255.255.255	DHCP	314	DHCP Request - Transaction ID 0x3d1e
4	0.070345	192.168.0.1	192.168.0.10	DHCP	342	DHCP ACK - Transaction ID 0x3d1e

0.... = Broadcast flag: Unicast
.000 0000 0000 0000 = Reserved flags: 0x0000
Client IP address: 0.0.0.0
Your (client) IP address: 0.0.0.0
Next server IP address: 0.0.0.0
Relay agent IP address: 0.0.0.0
Client MAC address: Grandstr_01:fc:42 (00:0b:82:01:fc:42)
Client hardware address padding: 0000000000000000
Server host name not given
Boot file name not given
Magic cookie: 0x696E696E
Option: (53) DHCP Message Type (Request)
Length: 1
DHCP: Request (3)
Option: (62) Client Identifier
Length: 7
Hardware type: Ethernet (0x01)
Client MAC address: Grandstr_01:fc:42 (00:0b:82:01:fc:42)
Option: (50) Requested IP Address
Length: 4
Requested IP Address: 192.168.0.10
Option: (54) DHCP Server Identifier
Length: 4
DHCP Server Identifier: 192.168.0.1
Option: (55) Parameter Request List
Option: (255) End
Padding: 00

- الرساله الرابعه والاخيره هي رساله ال **DHCP ACK** بيقولوك ال **server** فيها كله تمام و **done** سجلتك بال **mac** بتاعك و عطيتك ال **IP** ال انت وافتت عليه ويربطه بال **mac** الخاص بجهازك وبيديك ال **IP** ال هو ساعه يكون فيها ال **IP** دا معاك فيها وبعد كدا سيرفر ال **DHCP** لو لاقاك لسه شغال ومتصل بنفس ال **IP** هسيبيهولك انما لو

عملت للاتصال هتلacie خد ال IP دا ووجه ل user تاني lose يستخدمه بما انك يعني فقدت الاتصال .

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	0.0.0.0	255.255.255.255	DHCP	314	DHCP Discover - Transaction ID 0x3d1d
2	0.000295	192.168.0.1	192.168.0.10	DHCP	342	DHCP Offer - Transaction ID 0x3d1d
3	0.070831	0.0.0.0	255.255.255.255	DHCP	314	DHCP Request - Transaction ID 0x3d1e
4	0.070845	192.168.0.1	192.168.0.10	DHCP	342	DHCP ACK - Transaction ID 0x3d1e

```

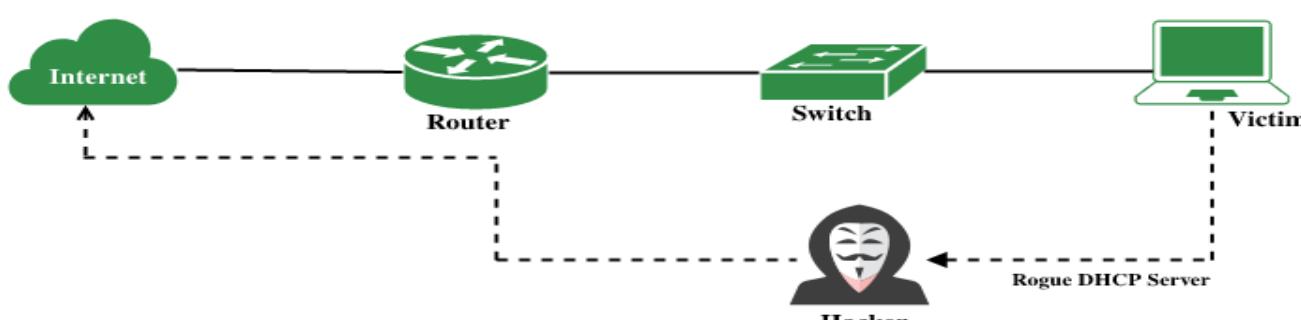
Seconds elapsed: 0
> Bootp flags: 0x0000 (Unicast)
Client IP address: 0.0.0.0
Your (client) IP address: 192.168.0.10
Next server IP address: 0.0.0.0
Relay agent IP address: 0.0.0.0
Client MAC address: Grandstr_01:fc:42 (00:0b:82:01:fc:42)
Client hardware address padding: 00000000000000000000000000000000
Server host name not given
Boot file name not given
Magic cookie: DHCP
# Option: (53) DHCP Message Type (ACK)
Length: 1
DHCP: ACK (5)
> Option: (58) Renewal Time Value
> Option: (59) Rebinding Time Value
# Option: (51) IP Address Lease Time
Length: 4
IP Address Lease Time: (3600s) 1 hour
# Option: (54) DHCP Server Identifier
Length: 4
DHCP Server Identifier: 192.168.0.1
# Option: (1) Subnet Mask
Length: 4
Subnet Mask: 255.255.255.0
> Option: (255) End
Padding: 000000000000000000000000000000000000000000000000000000000000000 ...

```

كل ال شفناه دا كان ال **Normal DHCP** تعالى نشوف الدنيا ماشيء ازاي فال **Suspicious DHCP**

- ممكن ال **rogue server** يتحل شخصيه ال **attacker** فيقتعك انه ال **DHCP Server** الحقيقي وهو **man in the middle** ولو عرف يقتعك بکدا هيبقا نفذ عليك **fake DHCP server** فانت كل حاجه هتعملها هترووح لل **attack** فهوبي يشوف كل ال بتعمله ودا

Rogue DHCP Server Attack



- ودا ال DHCP Rogue لل wire shark ال خدناه بال capture هتلaci ال offer لما بعتله discover بعتلك attacker بدل ال man in the middle attack الاصلی ال اخر عملیه ال server

No.	Source	Destination	Protocol	Info
1	0.0.0.0	255.255.255.255	DHCP	DHCP Discover - Transaction ID 0x816
2	192.168.1.1	255.255.255.255	DHCP	DHCP Offer - Transaction ID 0x18f
3	172.16.1.1	255.255.255.255	DHCP	DHCP Offer - Transaction ID 0x816
4	0.0.0.0	255.255.255.255	DHCP	DHCP Request - Transaction ID 0x816
5	192.168.1.1	255.255.255.255	DHCP	DHCP ACK - Transaction ID 0x816
6	172.16.1.1	255.255.255.255	DHCP	DHCP NAK - Transaction ID 0x816

```

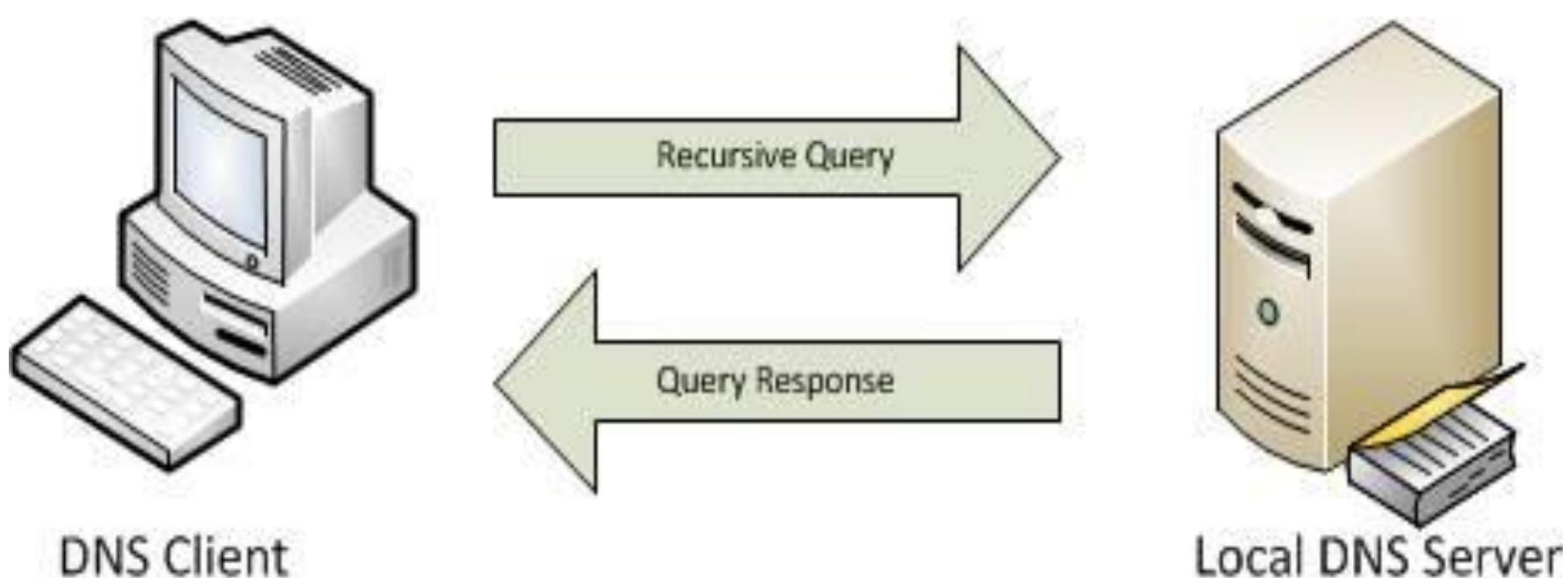
> Frame 2: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits) on interface -, id 0
> Ethernet II, Src: ca:03:46:64:00:00 (ca:03:46:64:00:00), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Internet Protocol Version 4, Src: 192.168.1.1 (192.168.1.1), Dst: 255.255.255.255 (255.255.255.255)
> User Datagram Protocol, Src Port: 67, Dst Port: 58
└ Dynamic Host Configuration Protocol (Offer)
    Message type: Boot Reply (2)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0x00000018f
    Seconds elapsed: 0
    > Bootp flags: 0x8000, Broadcast flag (Broadcast)
    Client IP address: 0.0.0.0 (0.0.0.0)
    Your (client) IP address: 192.168.1.4 (192.168.1.4)
    Next server IP address: 0.0.0.0 (0.0.0.0)
    Relay agent IP address: 0.0.0.0 (0.0.0.0)
    Client MAC address: ca:04:0d:64:00:00 (ca:04:0d:64:00:00)
    Client hardware address padding: 000000000000000000000000
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
    > Option: (53) DHCP Message Type (Offer)
    > Option: (54) DHCP Server Identifier (192.168.1.1)
    > Option: (51) IP Address Lease Time
    > Option: (58) Renewal Time Value
    > Option: (59) Rebinding Time Value
    > Option: (1) Subnet Mask (255.255.255.0)
    > Option: (3) Router
    > Option: (6) Domain Name Server
    > Option: (15) Domain Name
        Length: 10
        Domain Name: hacker.com
    > Option: (255) End
    Padding: 0000
  
```

DNS Traffic:

- اختصار ل layer 4 شغال ف Domain name System ال هي
ال على بورت 53 باستخدام بروتوكول UDP من application layer طبقه ال names resolve لل transport بيعمل الخاصه بالمواقع
ل زى مثلا لو انت عاوز تروح لجوجل بتروح
فالمتتصفح عندك وتكتب DNS الطلب دا بيروح لل google.com

ال متسجل عنده ال website انت رايحله عن طريق ال DNS protocol ويوجهاك لـ IP ويفهم انه لان مفيش حاجه ع النت اسمها google.com او بمعنى اصح مبيفهمهاش ال DNS بيفهم ارقام فقط ال هي ال IP لكن ال DNS server يعرف ال IP فهتلacie مسجل عنده فال website كل IP بال DNS server بتعاه.

- واي domain جديد بيظهر ع الانترنت بتلاقيه اتسجل تلقائي فال عشان تروح لموقع ما لازم الاول يطلع من جهازك DNS server بيعتها بروتوكول ال DNS ويترد عليه من ال DNS Query بال respond ومعاه بيوجهاك لاسم الموقع ال كتبته ف المتصفح



وال normal DNS server بتاع ال client هتلacie بيتبعت من client عشان يعرف عنوان موقع معين رايحله مش هتلacie بيتبعت من client آخر ولو لقيت كدا اعرف ان في جهاز ما suspicious او بيسبت DNS traffic هو ال عمال يبعث ال malware من جهاز لآخر .

تعالى نفرق ما مابين ال **suspicious** وال **normal** من خلال الجدول دا

Normal DNS Traffic	Suspicious DNS Traffic
Port 53, UDP	Traffic on port 53 but using TCP instead of UDP
Should only go to DNS Servers	DNS traffic not going to DNS Servers
Should see DNS Responses to DNS Queries	A lot of DNS Queries with no DNS responses or vice versa

- ال **normal** هتلaciه شغال على **port 53** باستخدام ال **UDP** وزي مقولنا هيروح ال **traffic** لـ **DNS server** فقط وهتشوف لكل **DNS query** الخاص بيها ... دا كدا فال **DNS response**

Normal

- انما هتلaci ال **suspicious** بـ يستخدم نفس ال **port** ولكن ببرتوكول ال **TCP** وليس ال **UDP** زي مشوفنا فال **normal** وهتلaci كمان ال **TCP** دا **normal** مفيهوش حاجه تشكي فيها وخد بالك من نقطه هنا ال **normal traffic** عادي ومش **suspicious** ولا حاجه فانت لو لقيته شغال بال **UDP** تمام ---

- مفيش مشكله انما لو لقيته شغال بال **TCP** تبدء تشک فيه وتروح
تعمل **investigation** فال **packet** بنفسك وارد تكون **normal** وال
ال رايح تحتاج ال **TCP** فعلا عشان متظلمش حد !!

- برضه فال **DNS traffic** هتلacci ان ال **suspicious** ال بيطلع من
جهازك مش بيوصل لـ **DNS server** فتبدء تشک علطول انه ممكن
يكون **suspicious** وبيتبع من جهاز لآخر زي مقولنا فوق

- فال **DNS query** هتلacci برضه **suspicious** كتير بدون بيترد
suspicious او العكس غترف ان دا **DNS response**
عشان ال **DNS query** كل **normal** بيترد عليها بـ **DNS response**

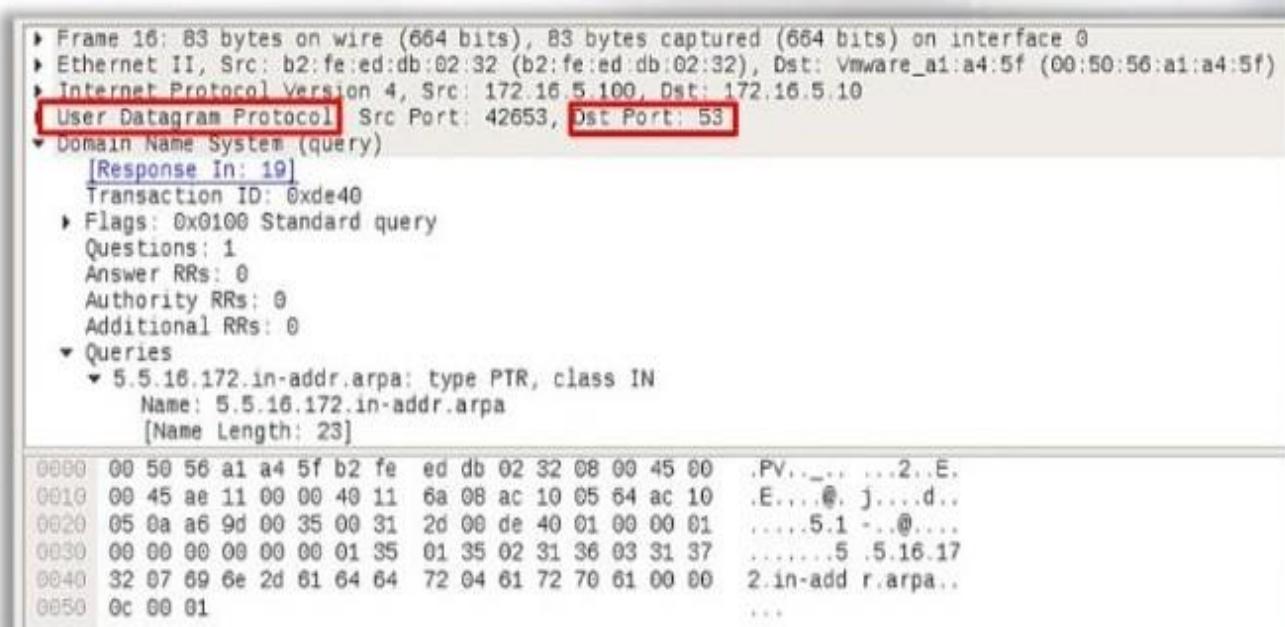
- ومن خلال الامثله الجايه هنحل ال **normal** وال **suspicious** من
خلال **wire shark** هنلقطها بال **packet capture**

No.	Time	Source	Destination	Protocol	Length	Info
16	26.200151138	172.16.5.100	172.16.5.10	DNS	83	Standard query 0xde40 PTR 5.5.16.172.in-addr.arpa
19	26.272980431	172.16.5.10	172.16.5.100	DNS	127	Standard query response 0xde40 PTR 5.5.16.172.in-addr.arpa PTR wkst-techsupport.sportsfoo.com
41	50.605405613	172.16.5.100	172.16.5.10	DNS	94	Standard query 0xa620 PTR 5.5.16.172.in-addr.arpa OPT
42	50.639061726	172.16.5.10	172.16.5.100	DNS	138	Standard query response 0xa620 PTR 5.5.16.172.in-addr.arpa PTR wkst-techsupport.sportsfoo.com OPT

Also notice that there are 2 different transaction IDs for the packets: 0xde40 & 0xa620.

- بص كدا ع ال **DNS query** هتلaci ليها رقمتعريفي اسمه ال **query** ال هو انت اما بتيجي تبعت ال **Transaction ID** معها الرقم دا ولازم اما يترد عليك فال **response** يرجعلك معاها نفس ال **Transaction ID**.

Below we can verify that this is a UDP packet and it's using an expected port, 53.



هنا هتلaciه بيقولك الاتصال تم فعلا بشكل **normal** عشان جهاز ال **client** راح لل **server** على **port 53** وكمان على بروتوكول ال **connection** **UDP** سليم.

```

> Frame 19: 127 bytes on wire (1016 bits), 127 bytes captured (1016 bits) on interface 0
> Ethernet II, Src: VMware_a1:a4:5f (00:50:56:a1:a4:5f), Dst: b2:fe:ed:db:02:32 (b2:fe:ed:db:02:32)
> Internet Protocol Version 4, Src: 172.16.5.10, Dst: 172.16.5.100
> User Datagram Protocol, Src Port: 53, Dst Port: 42653
-> Domain Name System (response)
  [Request In: 16]
  [Time: 0.072829293 seconds]
  Transaction ID: 0xde40
  Flags: 0x8580 Standard query response, No error
  Questions: 1
  Answer RRs: 1
  Authority RRs: 0
  Additional RRs: 0
  Queries
    * 5.5.16.172.in-addr.arpa: type PTR, class IN
      Name: 5.5.16.172.in-addr.arpa
      [Name Length: 23]
      [Label Count: 6]
      Type: PTR (domain name PoinTeR) (12)
      Class: IN (0x0001)
  Answers
    * 5.5.16.172.in-addr.arpa: type PTR, class IN, wkst-techsupport.sportsfoo.com
      Name: 5.5.16.172.in-addr.arpa
      Type: PTR (domain name PoinTeR) (12)
      Class: IN (0x0001)
      Time to live: 3600
      Data length: 32
      Domain Name: wkst-techsupport.sportsfoo.com

```

هتلaci هنا ال **query** جالك بتاع ال **response** ال بعثتها و هتلaci ال
DNS server ال بعثتها انت لل **query** بتاع ال **answer**

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.170.8	192.168.170.20	DNS	70	Standard query 0x1032 TXT google.com
2	0.000530	192.168.170.20	192.168.170.8	DNS	96	Standard query response 0x1032 TXT google.com TXT
3	4.005222	192.168.170.8	192.168.170.20	DNS	70	Standard query 0xf76f MX google.com
4	4.037355	192.168.170.20	192.168.170.8	DNS	290	Standard query response 0xf76f MX google.com MX 40 smtp4.google.com MX 10 smtp5.google.com..
5	12.017185	192.168.170.8	192.168.170.20	DNS	70	Standard query 0x49a1 LOC google.com
6	12.056209	192.168.170.20	192.168.170.8	DNS	70	Standard query response 0x49a1 LOC google.com
7	20.024827	192.168.170.8	192.168.170.20	DNS	85	Standard query 0x9bb5 PTR 104.9.192.66.in-addr.arpa
8	20.025333	192.168.170.20	192.168.170.8	DNS	129	Standard query response 0x9bb5 PTR 104.9.192.66.in-addr.arpa PTR 66-192-9-104.gen.twteleco..
9	92.109905	192.168.170.8	192.168.170.20	DNS	74	Standard query 0x75c0 A www.netbsd.org
10	92.238816	192.168.170.20	192.168.170.8	DNS	90	Standard query response 0x75c0 A www.netbsd.org A 204.152.190.12
11	100.965135	192.168.170.8	192.168.170.20	DNS	74	Standard query 0xf0d4 AAAA www.netbsd.org
12	109.292893	192.168.170.20	192.168.170.8	DNS	102	Standard query response 0xf0d4 AAAA www.netbsd.org AAAA 2001:4f8:4:7:2e0:81ff:fe52:9a6b
13	109.027394	192.168.170.8	192.168.170.20	DNS	74	Standard query 0x7f39 AAAA www.netbsd.org
14	109.027781	192.168.170.20	192.168.170.8	DNS	102	Standard query response 0x7f39 AAAA www.netbsd.org AAAA 2001:4f8:4:7:2e0:81ff:fe52:9a6b

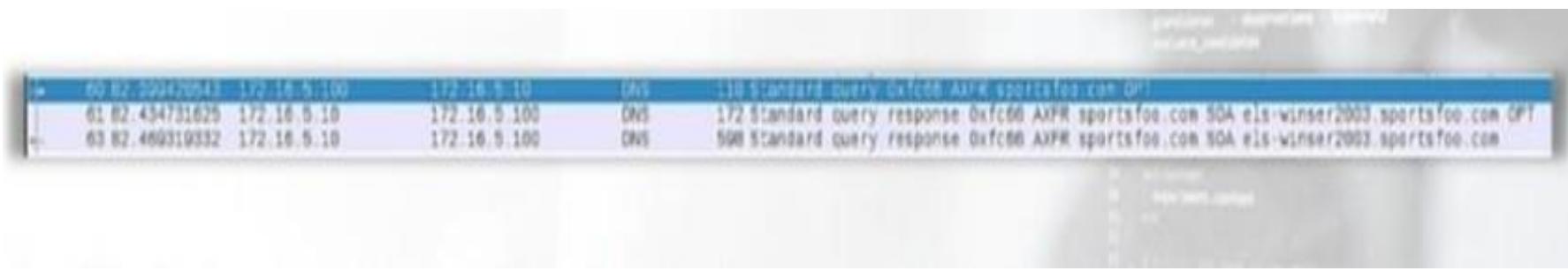
```

> Frame 1: 70 bytes on wire (560 bits), 70 bytes captured (560 bits)
> Ethernet II, Src: AsustekI_b1:0c:ad (00:e0:18:b1:0c:ad), Dst: QuantaCo_32:41:8c (00:c0:9f:32:41:8c)
> Internet Protocol Version 4, Src: 192.168.170.8, Dst: 192.168.170.20
> User Datagram Protocol, Src Port: 32795, Dst Port: 53
> Domain Name System (query)

```

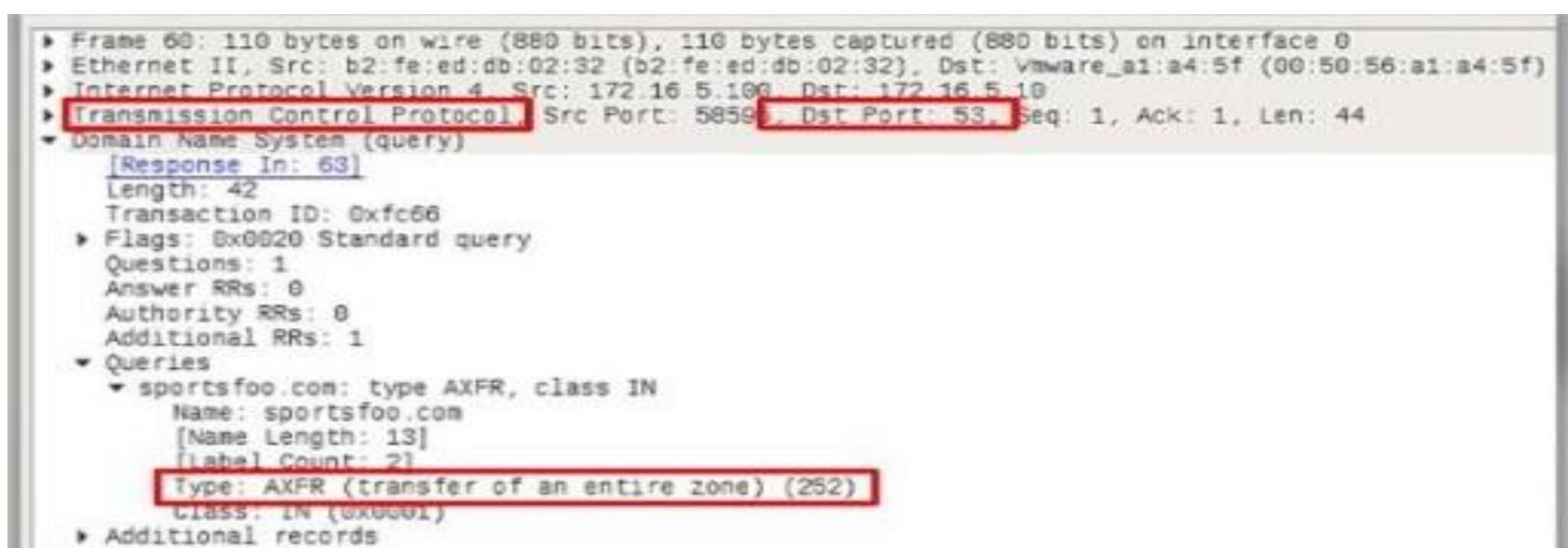
بص هنا برضه هتلaci **DNS query** وكل **normal DNS Traffic** **DNS response** بيقابها

تعالى نبص ع الـ **suspicious** كدا



- هتبص تلاقي **attack** بيحصل عندك اسمه **DNS Zone Transfer** دي بتحصل ما بين ال **servers** وبتحصل ما بين ال **client** وبعضها ميعملهاش **servers** وبتحصل ما بين ال **client** عشان يتشاركو المعلومات فلما شوف **IP** بتاع **client** بيبعد **zone** عشان يغيرها او يتلاعب بيها .
وبيعمل كدا عشان يسحب العناوين الموجوده فال **DNS servers** .

- وخد بالك من نقطه ال **zone transfer** اما تيجي تتبعت بتتبع
بروتوكول ال **TCP** ال كنا قللنا عليه فوق ممكن حالات يكون فيها اهي دي حاله منهم ودا هنشوفه ف المثال الجي مع بعض **suspicious**



- لو بصينا هنا هنلاقي فعلا استخدم ال **TCP** ودا عشان ال **request** مبعوت بال **zone transfer** بيبقا عاوز يعرف كل ال **IP** فحجم ال **traffic** بيكون كبير عشان كدا بيستخدم ال **TCP** عكس ال **UDP**

- ال هتلائيه حتى اما يبعث traffic بيكون حجمه صغير على قد الاتصال وبيبقا عاوز IP لجهاز واحد فهتلائي الدنيا تمام عكس ال TCP
ال بيшиيل traffic كبير نوعاما فبنبدء نشك فيه

- فانت ك traffic هتبده تبع على اي threat hunter شغال بال server لو لقيته من TCP
لديك suspicion كدا دا ال server ل client

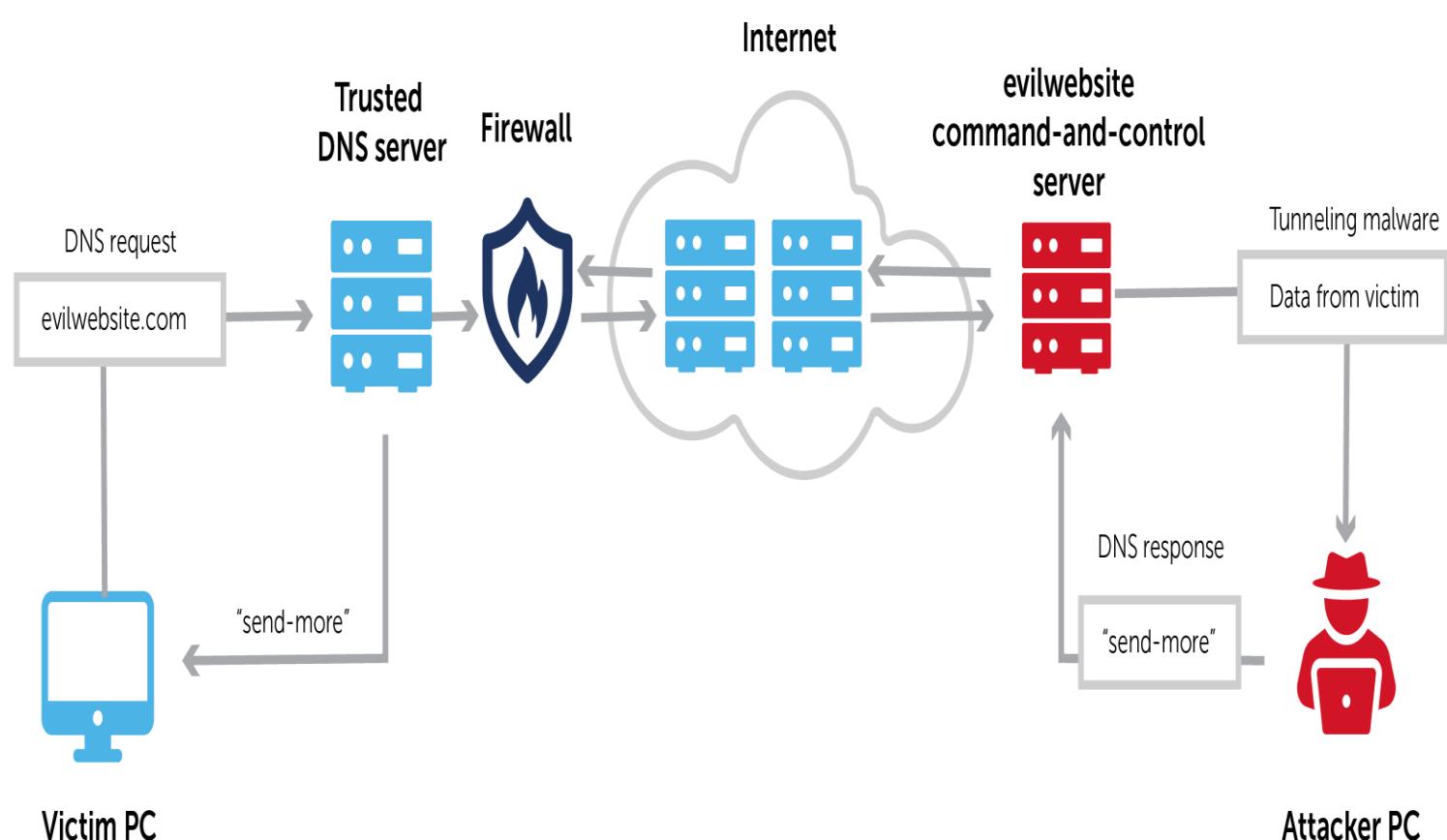
- ودا شكل ال attacker ال بيرجعهولك ال response
DNS Transfer Zone Attack عليك

```
► Frame 61: 172 bytes on wire (1376 bits), 172 bytes captured (1376 bits) on interface 0
► Ethernet II, Src: VMware_A1:A4:5F (00:50:56:a1:a4:5f), Dst: B2:FE:ED:D0:02:32 (b2:fe:ed:db:02:32)
► Internet Protocol Version 4, Src: 172.16.5.10, Dst: 172.16.5.100
► Transmission Control Protocol, Src Port: 53, Dst Port: 50595, Seq: 1, Ack: 45, Len: 106
└ Domain Name System (response)

[Request ID: 60]
[Time: 0.035303082 seconds]
Length: 104
Transaction ID: 0xfc66
Flags: 0x8000 Standard query response, No error
Questions: 1
Answer RRs: 1
Authority RRs: 0
Additional RRs: 1
* Queries
  * sportsfoo.com: type AXFR, class IN
    Name: sportsfoo.com
    [Name Length: 13]
    [Label Count: 2]
    Type: AXFR (transfer of an entire zone) (252)
    Class: IN (0x0001)
* Answers
  * sportsfoo.com: type SOA, class IN, mname els-winserv2003.sportsfoo.com
    Name: sportsfoo.com
    Type: SOA (Start Of a zone of Authority) (6)
    Class: IN (0x0001)
    Time to live: 3600
    Data length: 50
    Primary name server: els-winserv2003.sportsfoo.com
    Responsible authority's mailbox: hostmaster.sportsfoo.com
    Serial Number: 19
    Refresh Interval: 900 (15 minutes)
    Retry Interval: 600 (10 minutes)
    Expire limit: 86400 (1 day)
    Minimum TTL: 3600 (1 hour)
* Additional records
```

- عدنا **attack** تاني اسمه ال **DNS Tunneling** بنسخدمه فال **attacker** عمل عليك **data exfiltration attack** ال هو بعد اما ال **exploitation attack** وعملك **attacker** بيعمل حاجه اسمها تسريب الداتا زي ال بتشفوفها دايما على ال **Dark web** لان ال **fire wall** معاهم **copy – paste** لان ال **data** هيسكه فيقوم عامل ممرات او انفاق عباره عن **DNS Traffic** لكن **exfiltration** ال هو عاملها **Data**

DNS tunneling



HTTP & HTTPS Traffic:

- اختصار ل **Hypertext Transfer Protocol** ودا برتوكول شغال ف 4 layer ال هي **application layer** ودا برتوكول التصفح لو عاوز تشو夫 فيديو او تدخل موقع ما لازم تتعدي على ال **HTTP** والفرق بيشه وبين ال **HTTP** بسيط جدا ال **clear text** يعني اي حد يقدر يشوف محتوي ال **packet** وهي بتترسل من ال **source** لـ **destination** فبيقا معرض لل **attack** كتير عكس ال **HTTPS** ال بيكون **certificate** فال **data** تكون مشفره بواسطه **secure** بيشرط تبادلها بين طرفي الاتصال قبل ارسال المعلومات وبتكون ال **SSL certificate** فبتضمن نقل البيانات بسلام بين الطرفين وعدم حدوث **attack** عليها بالمنتصف وعدم تعرضها للاختراق و مجرد فهمك لواحد الثاني هيقيا بالنسبة لك سهل ان شاء الله .

- هنمهد الاول نقطتين هنعزهم بعدين اول حاجه لازم تعرفها عن ال **request** انه بيشتغل بنظام ال **Reply** وال **Request** فكل **HTTP** لازم يترد عليه بال **reply** المناسب... فمثلا لو **PC1** عاوز يقراء حاجه عند **PC2** بيعتله **GET Request** اسمه **request** انما لو **PC2** عاوز يدخل **data** عند **PC2** هيعتله حاجه اسمها **POST** ولو عاوز يمسح من عنده حاجه هيعتله **DELETE** ولهذا عشان كدا ال **HTTP** بيشتغل ب **method** معينه ودا المتصفح بتاعك بيكون عارفها وبيعرف يوجهك للطريقه المناسبه ...

وېرضه هتلاقى ال **Reply** لىه رد معین بىردىك بىه بىتفهم منه رد ال
علي ال انت بىعنهوله . **server**

وكل **request** و **Reply** لىه **status code** بىعبر عنه او بىفهمه ال
وېرض علیك بىه ودا ھنشوفە فالمثال الاتي

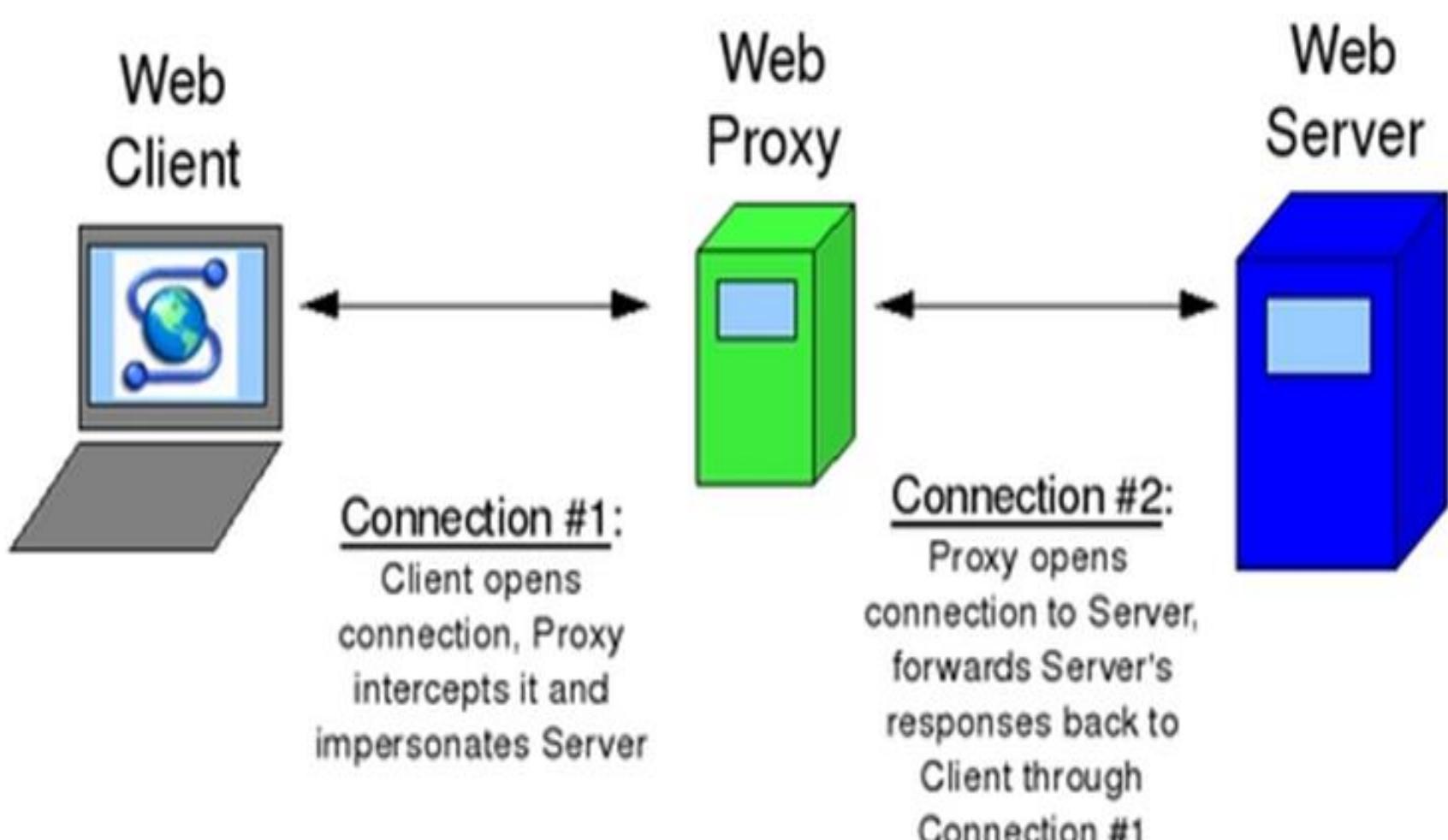
HTTP Status Codes		
Level 200	Level 400	Level 500
200: OK 201: Created 202: Accepted 203: Non-Authoritative Information 204: No content	400: Bad Request 401: Unauthorized 403: Forbidden 404: Not Found 409: Conflict	500: Internal Server Error 501: Not Implemented 502: Bad Gateway 503: Service Unavailable 504: Gateway Timeout 599: Network Timeout

- ھنبىص نلاقي بعض من الاکواد ال بىبىعنهالك ال **destination** بىدل ع
الحاله ال هو فيها يعنى مثلا لو فتحت صفحه المتصفح وكتبت فيه
GET ولاقيت صفحه جوجل ظهرتاك يېقا انت بىع **google.com**
لوجل ووصل لىل **server** ورض علیك ب **reply** ال هو
يافق انك تتصفح جوجل وقام فاتحلك الصفحه
وھكذا هتلاقى كل **Reply** لىه **request** معين باى **status code 200**
وھكذا هتلاقى كل **Reply** لىه **request** معين باى **status code 201**
بىعنهالك **Facebook** على **comment**
. **server** بىتېعنى **code 201** من ال

تعالي نعرف الفرق ما بين ال **normal** وال **suspicious**

Normal HTTP Traffic	Suspicious HTTP Traffic
Port 80, TCP Port 8080, TCP (used as alternate) Port 8088, TCP (used as alternate)	Malicious binaries (backdoors), scripts, web shells, etc. will use this port because typically, in all corporate environments, the port is open.
Plaintext traffic	If the traffic is encrypted, then most likely it's being used for malicious traffic. Malicious traffic can be in plaintext as well.
Web server typically in FQDN format.	The server will point to an IP address instead of FQDN format.

عندنا اول حاجه ال **HTTP Request** وهو رايح من ال جهازك لـ **HTTP** بيروح على **port 80** عند السيرفر ولو هتشغل **server** **Proxy Server** بمعنى يكون **WEB Server** ال هو **Server** وسيط بينك وبين شبكه الانترنت نفسها ... ال **HTTP Request** على منفذ **8080** او **8088** تمام كدا



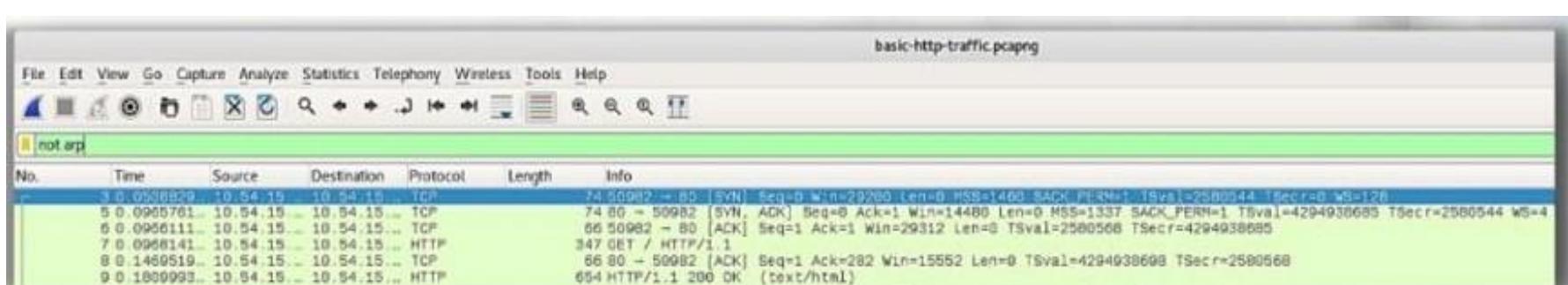
- كمان هتلاقي ال **HTTP Traffic** يعني اي حد لو عمل **Traffic Intercept** لـ **Clear text** بيكون **HTTP Traffic** هيرفع يقرؤه عادي ودا اشهر البرتوكولات ال هتلاقيه بيتعمل عليه **Man in The Middle Attack** كمان هتلاقي ال **Web Server** ال انت رايحله على فورمات اسمه **Fully Qualified Domain Name** ال **FQDN** يعني اسم الموقع ال انت رايحله زي **google** كدا ... انما لو لقيتك رايح لموقع زي كدا هقولك اوقف !! ايه الموقع ال انت رايحله الغريب دا فين ال **FQDN** بتاعته !! عشان ممكن يكون يكون موقع مشبوه ! وممكن يكون سليم اه عادي ممكن تروح لموقع عادي باستخدام ال **IP** بس انت لو شفت دا تشک فيه وتعمله **investigation** **Threat hunter** عشان تتأكد من سلامه ال **Traffic** ده .

- تعالى نبص عال **suspicious** عشان نعرف الفرق بينهم ودا كله هنشوفه من خال امثاله عمليه بال **wire shark**
- هتلاقي فال **HTTP** برضه ال **suspicious** شغال على نفس ال **destination** مش **port 80** ولكن هتلاقيه مفتوح دايما عند ال **port 80** عادي بيعمل **request and reply** ويخلص هتلاقيه دايما عاوز ال **port 80** مفتوح ودا لان ال **attacker** بيكون عامل **port 80** مثلا ومبرمجه يروحك على **back door malicious** فهتلاقيه دايما عاوز يروح لـ **port 80** ال شغال عليه ال **HTTP**

- الحاجه الثانيه انك هتلaci ال **HTTP Traffic** ال جايلك
مشفرو وممکن تلaci جزء منه بس هو ال مشفر والمفروض
انه يكون **tools** ودا نتيجه لـ **clear text** ال بيستخدمها ال
malicious Traffic عن طريق ال **attacker**
..... **encryption**

- الحاجه الاخيرة انك هتلaci ال **HTTP Traffic** رايج ل **IP** وليس
زي مقولنا قبل كدا ... فانت لو لقيت حاجه زي كدا علي غير
المعتاد تبدئ تشكي فال **traffic** وتوقفه وتعمل فيه **Investigation**.

تعالي نشوف امثاله عمليه بال **wire Shark** عشان نعرف الفرق



Here is a brief summary of the above image:

- We are seeing 6 packets (4 relating to TCP and 2 on HTTP).
- Packets 3-6 is the **TCP Handshake**. HTTP relies on TCP for reliability.
- In packet 7, we notice an **HTTP method** (GET).
- In packet 9, we see an **HTTP response code** (200 OK).

- عشان يحصل **HTTP traffic** ويتنقل لازم يكون حصل قبل منه ال **TCP** بتوع ال **establish** الخاص بال **three way hand shake**

ويحصل **SYN** و **ACK** و **SYN_ACK** زى مقولنا قبل كدا وبعد كدا
يبدء **HTTP Traffic** ف الارسال

- لو بصيت فال **packet** رقم **7** هتلاقى بعد اما حصل ال **3 way**
GET method انه يبعث **HTTP** هيبدء برتوكول ال **hand shake**
وهيرض عليه ال **server** بمعناه موافق على
. **Normal Traffic** كدا اول علامات ال

- تاني علامه عندك هتلاقى جهاز ال **source** رايح ل **destination** **port 80**

No.	Time	Source	Destination	Protocol	Length	Info
3	0.0500029	10.54.15...	10.54.15...	TCP	74	50982 - 80 [S,N] Seq=0 Win=29200 Len=0 MSS=1400 SACK_PERM=1 TSval=2580544 TSecr=0 WS=120
5	0.0965761	10.54.15...	10.54.15...	TCP	74	80 - 50982 [S,N, ACK] Seq=0 Ack=1 Win=14400 Len=0 MSS=1337 SACK_PERM=1 TSval=4294938685 TSecr=2580544 WS=4
6	0.0966111	10.54.15...	10.54.15...	TCP	66	50982 - 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=2580560 TSecr=4294938685
7	0.0966141	10.54.15...	10.54.15...	TCP	34	80 - 50982 [ACK]
8	0.1460510	10.54.15...	10.54.15...	TCP	66	80 - 50982 [ACK] Seq=1 Ack=282 Win=15552 Len=0 TSval=4294938690 TSecr=2580560
9	0.1800093	10.54.15...	10.54.15...	HTTP	654	HTTP/1.1 200 OK (text/html)

Frame 7: 347 bytes on wire (2776 bits), 347 bytes captured (2776 bits) on interface 0
Ethernet II, Src: 26:11:59:88:53:02 (26:11:59:88:53:02), Dst: VMware_a1:61:66 (00:50:56:a1:61:66)
Internet Protocol Version 4, Src: 10.54.15.100, Dst: 10.54.15.68
Transmission Control Protocol, Src Port: 50982, Dst Port: 80, Seq: 1, Ack: 1, Len: 281
Hypertext Transfer Protocol

- انت ممكن تدخل جوا ال **HTTP packet** بتاعت ال **HTTP** وتشوف الدنيا
ماشيء ازاي جوا بمعنى تعمل **investigation** بشكل متعمق شويه

```
Frame 7: 347 bytes on wire (2776 bits), 347 bytes captured (2776 bits) on interface 0
Ethernet II, Src: 26:11:59:88:53:02 (26:11:59:88:53:02), Dst: VMware_a1:61:66 (00:50:56:a1:61:66)
Internet Protocol Version 4, Src: 10.54.15.100, Dst: 10.54.15.68
Transmission Control Protocol, Src Port: 50982, Dst Port: 80, Seq: 1, Ack: 1, Len: 281
Hypertext Transfer Protocol
  GET / HTTP/1.1\r\n
  Host: 10.54.15.68\r\n
  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0\r\n
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
  Accept-Language: en-US,en;q=0.5\r\n
  Accept-Encoding: gzip, deflate\r\n
  Connection: keep-alive\r\n
\r\n
[Full request URI: http://10.54.15.68/] [HTTP request 1/2]
[Response in frame: 9]
[Next request in frame: 11]
```

- هنا مثلا عند السهم المشار اليه هتلاقى فمحتوى ال **HTTP** انه بعث

ولكن ل **IP** وليس **FQDN** ودا ممكن يكون !!
زى مقولنا بس ممكن يكون **normal** برضه !! **suspicious**

- بس الحاله ال عندنا دي **Normal** طب ليه !! عشان ال **IP** ال انت رايحله دا **IP** داخلي واقع جوا نفس ال **subnet mask** بتاعك او ال **10.54.15.68** IP بمعنى اصح فهتلاقى ال **IP** **request** المضار اليه بالسهم ال هو ال **server** فحالتنا ال كان عاوز يبعث له **10.54.15.100** هو ال **Ip request** . تمام كدا فدي حاله بنستثنى منها ال **IP** لو كان فال **request**

- امتى تشكيشك فال **Traffic** ال رايحله **server** ال كان ال **IP** **subnet** بيكون شبكه تانيه مش واقع معاك نفس ال **public IP** . وصلت كدا

تعالي نبص ع ال **..... HTTP Request response** بتاع



- هتلاقي ال packet number 7 Response فال packet number 9 بيكوله GET 200 يعني تمام وقبل ال request .

تعالي نشوف المحتوي بتاع ال Response ال جي عندنا

```
Line-based text data: text/html
<!DOCTYPE html>
<html>
</head>
</head>
<body>
<div class="login-card">
<h1>Log-in</h1><br>
<form method="post" action="login.php">
<input type="text" name="user" autocomplete="off" placeholder="Username">
<input type="password" name="pass" placeholder="Password">
<input type="submit" name="login" class="login login-submit" value="login">
</form>
</div>
</body>
</html>
```

- هنلاقي ال wire shark جايبلنا ال Raw Data مكتوبه بال web export انت ممكن تعملها Html- text عادي وتفتحلك ال content وتقراه عادي .

- بس لو بصيت فال login هنلاقي ان دي صفحه content موجود فيها username and password معينه لـ Tap

لو حابب تعرف تفاصيل اكتر عن ال packet ممكن تروح تقف بالماوس ع ال packet right click فال wire shark تضغط وتختر packet هنلاقي طلعلك تفاصيل اكتر عن ال Follow TCP stream



تعالي نشوف ال **Suspicious HTTP packet** بتكون عامله ازاي
..... **Wire shark** فال

No.	Time	Source	Destination	Protocol	Length	Info
10.000000	10.124.211.96	10.124.211.96	TCP	74	33020 - 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=125613 TSecr=0 WS=128	
20.056720	10.124.211.96	10.124.211.200	TCP	74	80 - 33020 [SYN, ACK] Seq=0 Ack=1 Win=14400 Len=0 MSS=1337 SACK_PERM=1 TSval=222206 TSecr=125613 WS=4	
30.056747	10.124.211.96	10.124.211.96	TCP	66	33020 - 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=125627 TSecr=222206	
40.056824	10.124.211.200	10.124.211.96	HTTP	249	GET / HTTP/1.1	
50.133531	10.124.211.96	10.124.211.200	TCP	66	80 - 33020 [ACK] Seq=1 Ack=284 Win=15592 Len=0 TSval=222223 TSecr=125627	
60.133549	10.124.211.96	10.124.211.200	HTTP	101	[TCP Previous segment not captured] Continuation	
70.0.133556	10.124.211.96	10.124.211.96	TCP	78	[TCP Window Update] 33020 - 80 [ACK] Seq=284 Ack=1 Win=30336 Len=0 TSval=125647 TSecr=222223 SRE=1361	
80.0.156230	10.124.211.96	10.124.211.200	TCP	1391	[TCP OUT-OF-Order] 80 - 33020 [ACK] Seq=1 Ack=284 Win=15592 Len=1325 TSval=125627 TSecr=222224	
90.0.156259	10.124.211.200	10.124.211.96	TCP	66	33020 - 80 [ACK] Seq=1361 Ack=1361 Win=33280 Len=0 TSval=125657 TSecr=222224	
10.0.481485	10.124.211.200	10.124.211.96	HTTP	389	GET /news.php HTTP/1.1	
11.0.552272	10.124.211.96	10.124.211.200	TCP	1391	[TCP segment of a reassembled PDU]	
12.0.552305	10.124.211.200	10.124.211.96	TCP	66	33020 - 80 [ACK] Seq=607 Ack=2606 Win=36006 Len=0 TSval=126501 TSecr=223003	
13.0.552678	10.124.211.96	10.124.211.200	HTTP	1393	HTTP/1.1 200 OK (text/html)	
14.0.552589	10.124.211.200	10.124.211.96	TCP	66	33020 - 80 [ACK] Seq=607 Ack=3973 Win=30040 Len=0 TSval=126501 TSecr=223003	
15.0.960093	10.124.211.200	10.124.211.96	HTTP	410	GET /newsdetails.php?id=26 HTTP/1.1	
16.0.020439	10.124.211.96	10.124.211.200	TCP	1391	[TCP segment of a reassembled PDU]	
17.0.020461	10.124.211.200	10.124.211.96	TCP	66	33020 - 80 [ACK] Seq=951 Ack=5298 Win=41886 Len=0 TSval=127118 TSecr=223701	
18.0.020470	10.124.211.96	10.124.211.200	HTTP	83	HTTP/1.1 200 OK (text/html)	
19.0.020471	10.124.211.200	10.124.211.96	TCP	66	33020 - 80 [ACK] Seq=951 Ack=5315 Win=41886 Len=0 TSval=127118 TSecr=223701	
20.0.453656	10.124.211.200	10.124.211.96	HTTP	373	GET /newsdetails.php?id=2627 HTTP/1.1	
21.0.517192	10.124.211.96	10.124.211.200	TCP	1391	[TCP segment of a reassembled PDU]	
22.0.517210	10.124.211.200	10.124.211.96	TCP	66	33020 - 80 [ACK] Seq=1258 Ack=6640 Win=44800 Len=0 TSval=127992 TSecr=224575	
23.0.517216	10.124.211.96	10.124.211.200	HTTP	68	HTTP/1.1 200 OK (text/html)	
24.0.517218	10.124.211.200	10.124.211.96	TCP	66	33020 - 80 [ACK] Seq=1258 Ack=6642 Win=44800 Len=0 TSval=127992 TSecr=224575	
25.0.520484	10.124.211.96	10.124.211.200	TCP	66	80 - 33020 [FIN, ACK] Seq=6642 Ack=1258 Win=18768 Len=0 TSval=225826 TSecr=127992	
26.0.520662	10.124.211.200	10.124.211.96	TCP	66	33020 - 80 [FIN, ACK] Seq=1258 Ack=6643 Win=44800 Len=0 TSval=129243 TSecr=225826	
27.0.582082	10.124.211.96	10.124.211.200	TCP	66	80 - 33020 [ACK] Seq=6643 Ack=1259 Win=18768 Len=0 TSval=225842 TSecr=129243	
28.0.437742	10.124.211.200	10.124.211.96	TCP	74	33022 - 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=129973 TSecr=0 WS=128	
29.0.503661	10.124.211.96	10.124.211.200	TCP	74	80 - 33022 [SYN, ACK] Seq=0 Ack=1 Win=14400 Len=0 MSS=1337 SACK_PERM=1 TSval=226572 TSecr=129973 WS=4	

- فال **packet** دی عندنا **Traffic** هتلaci ال **source** بیبعت

انه عاوز يروح لعنويين زی کدا **requests**

news details.php?id=26%27 **news details.php?id=26**

- فلو بصيت عال **traffic** الاخير خصوصا لان التانيين **normal** واحد

عاوز يفتح صفحه **php** والثاني عاوز يفتح صفحه **php** رقمها 26

انما تيجي لل **request** **26%27** التالت هتلaci **!!!**

- باختصار دا **SQL Injection** وخصوصا **suspicious code**

Attack لان فيه رموز زی ال **# , ^ , % , &** بتبقی محظوظة

للسيرفرات الخاصة بالـ **HTTP** مينفعش **user** عادي يستخدمهم ويحطهم فالمتصفح لأن دي ببساطه بيتعملها تحويل لارقام ورموز **injection** وتعمل لل **Destination** ال **server** تانيه اما توصل لل ... فدا نوع **traffic** بنصفه **suspicious** لو لقيت الرموز دي ... عموما اي **SQL injection** تلاقيه زي كدا اعرف انه **request**

No.	Time	Source	Destination	Protocol	Length	Info
20	9.43050	10.124.211.200	10.124.211.96	HTTP	373	GET /newsdetails.php?id=20&CategoryID=1 HTTP/1.1
21	9.517192	10.124.211.96	10.124.211.200	TCP	1391	[TCP segment of a reassembled PDU]
22	9.517210	10.124.211.200	10.124.211.96	TCP	66	33020 - 80 [ACK] Seq=1258 Ack=6640 Win=44800 Len=0 TSeqval=127992 TSeqcr=224575
23	9.517216	10.124.211.96	10.124.211.200	HTTP	68	HTTP/1.1 200 OK (text/html)
24	9.517218	10.124.211.200	10.124.211.96	TCP	66	33020 - 80 [ACK] Seq=1260 Ack=6642 Win=44800 Len=0 TSeqval=127992 TSeqcr=224575
25	14.520484	10.124.211.96	10.124.211.200	TCP	66	80 - 33020 [FIN, ACK] Seq=6642 Ack=1258 Win=0 TSeqval=225826 TSeqcr=127992
26	14.520662	10.124.211.200	10.124.211.96	TCP	66	33020 - 80 [FIN, ACK] Seq=1258 Ack=6643 Win=44800 Len=0 TSeqval=129243 TSeqcr=225826
27	14.582082	10.124.211.96	10.124.211.200	TCP	66	80 - 33020 [ACK] Seq=6643 Ack=1258 Win=18768 Len=0 TSeqval=228842 TSeqcr=129243
28	17.437742	10.124.211.200	10.124.211.96	TCP	74	33022 - 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1400 SACK_PERM=1 TSeqval=129973 TSeqcr=0 WS=128
29	17.503661	10.124.211.96	10.124.211.200	TCP	74	80 - 33022 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1337 SACK_PERM=1 TSeqval=226572 TSeqcr=129973 WS=4
30	17.503697	10.124.211.200	10.124.211.96	TCP	66	33022 - 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSeqval=129989 TSeqcr=226572
31	17.504112	10.124.211.200	10.124.211.96	HTTP	389	GET /newsdetails.php?id=20&CategoryID=1; --%20- HTTP/1.1
32	17.575934	10.124.211.96	10.124.211.200	TCP	66	80 - 33022 [ACK] Seq=1 Ack=324 Win=15552 Len=0 TSeqval=226590 TSeqcr=126980
33	17.578773	10.124.211.96	10.124.211.200	TCP	84	[TCP Previous segment not captured] (TCP segment of a reassembled PDU)
34	17.578787	10.124.211.200	10.124.211.96	TCP	78	[TCP Windows Update] 33022 - 80 [ACK] Seq=324 Ack=1 Win=38336 Len=0 TSeqval=130000 TSeqcr=226590 SLE=1326 SRE=1344
35	17.579066	10.124.211.96	10.124.211.200	TCP	1391	[TCP Cut-Off-Officer] 80 - 33022 [ACK] Seq=1 Ack=324 Win=15552 Len=1325 TSeqval=226591 TSeqcr=129909
36	17.579085	10.124.211.200	10.124.211.96	TCP	66	33022 - 80 [ACK] Seq=324 Ack=1344 win=33280 Len=0 TSeqval=130008 TSeqcr=226591
37	22.579448	10.124.211.200	10.124.211.96	TCP	66	33022 - 80 [FIN, ACK] Seq=324 Ack=1344 Win=33280 Len=0 TSeqval=131258 TSeqcr=226591
38	22.584131	10.124.211.96	10.124.211.200	TCP	66	80 - 33022 [FIN, ACK] Seq=1344 Ack=324 Win=15552 Len=0 TSeqval=227042 TSeqcr=130008
39	22.584144	10.124.211.200	10.124.211.96	TCP	66	33022 - 80 [ACK] Seq=325 Ack=1345 Win=33280 Len=0 TSeqval=131259 TSeqcr=227842
40	22.625046	10.124.211.96	10.124.211.200	TCP	66	80 - 33022 [ACK] Seq=1345 Ack=325 Win=15552 Len=0 TSeqval=227852 TSeqcr=131258
41	25.101234	10.124.211.200	10.124.211.96	TCP	74	33024 - 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1400 SACK_PERM=1 TSeqval=131888 TSeqcr=0 WS=128
42	25.141132	10.124.211.96	10.124.211.200	TCP	74	80 - 33024 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1337 SACK_PERM=1 TSeqval=228481 TSeqcr=131888 WS=4
43	25.141157	10.124.211.200	10.124.211.96	TCP	66	33024 - 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSeqval=131898 TSeqcr=228481
44	25.141361	10.124.211.200	10.124.211.96	HTTP	389	GET /newsdetails.php?id=20&CategoryID=1; --%20- HTTP/1.1
45	25.186026	10.124.211.96	10.124.211.200	TCP	66	80 - 33024 [ACK] Seq=1 Ack=324 Win=15552 Len=0 TSeqval=228492 TSeqcr=131899
46	25.187157	10.124.211.96	10.124.211.200	HTTP	1289	HTTP/1.1 200 OK (text/html)
47	25.187168	10.124.211.200	10.124.211.96	TCP	66	33024 - 80 [ACK] Seq=324 Ack=1220 Win=32128 Len=0 TSeqval=131810 TSeqcr=228493
48	30.188296	10.124.211.200	10.124.211.96	TCP	66	33024 - 80 [FIN, ACK] Seq=324 Ack=1220 Win=32128 Len=0 TSeqval=133160 TSeqcr=228493

- هنا برضه هتلاني محاولات مستمرة من **attacker** بيحاول يعمل **Wire shark** وال **server** لـ **SQL Injection** لقطها .

طب احنا ک Threat Hunter دا attacker محتاجین نعمل نعرف ال عامل ال Attack دا manual Tool ولا عن طریق معینه ودا عshan احنا مطالبین نعرف دا عshan هنتبه فال report ال هنقدمه لل manager ال method فتعالی نعرف مع بعض ازای نعرف ال Attacker

لَوْ حِنَالَّا، **packet** دے، مثلاً و هندخا، فَالْتَفَاصِلُ، بِتَاعِتِهَا

No.	Time	Source	Destination	Protocol	Length	Info
20	9.453050	10.124.211.200	10.124.211.96	HTTP	373	GET /newsdetails.php?id=26%27 HTTP/1.1
21	9.517192	10.124.211.96	10.124.211.200	TCP	1391	[TCP segment of a reassembled PDU]
22	9.517210	10.124.211.200	10.124.211.96	TCP	66	33020 - 80 [ACK] Seq=1258 Ack=6640 Win=44800 Len=0 TStamp=127992 TSecr=224575
23	9.517216	10.124.211.96	10.124.211.200	HTTP	68	HTTP/1.1 200 OK (text/html)
24	9.517218	10.124.211.200	10.124.211.96	TCP	66	33020 - 80 [ACK] Seq=1258 Ack=6642 Win=44800 Len=0 TStamp=127992 TSecr=224575
25	14.520484	10.124.211.96	10.124.211.200	TCP	66	80 - 33020 [FIN, ACK] Seq=6642 Ack=1258 Win=18768 Len=0 TStamp=225826 TSecr=127992
26	14.520662	10.124.211.200	10.124.211.96	TCP	66	33020 - 80 [FIN, ACK] Seq=6643 Ack=1258 Win=44800 Len=0 TStamp=129243 TSecr=225826
27	14.582082	10.124.211.96	10.124.211.200	TCP	66	80 - 33020 [ACK] Seq=6643 Ack=1259 Win=18768 Len=0 TStamp=225842 TSecr=129243
28	17.437742	10.124.211.200	10.124.211.96	TCP	74	33022 - 80 [SYN] Seq=0 Win=20480 Len=0 MSS=1460 SACK_PERM=1 TStamp=129973 TSecr=0 WS=128
29	17.503661	10.124.211.96	10.124.211.200	TCP	74	80 - 33022 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1337 SACK_PERM=1 TStamp=129972 TSecr=129973 WS=4
30	17.503697	10.124.211.200	10.124.211.96	TCP	66	33022 - 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TStamp=129989 TSecr=226572
31	17.504112	10.124.211.200	10.124.211.96	HTTP	389	GET /newsdetails.php?id=26%20and%201=1--%20- HTTP/1.1
32	17.575934	10.124.211.96	10.124.211.200	TCP	66	80 - 33022 [ACK] Seq=1 Ack=324 Win=15662 Len=0 TStamp=226590 TSecr=129989
33	17.578773	10.124.211.96	10.124.211.200	TCP	84	[TCP Previous segment not captured] [TCP segment of a reassembled PDU]
34	17.578787	10.124.211.200	10.124.211.96	TCP	78	[TCP Window Update] 33022 - 84 [ACK] Seq=324 Ack=1 Win=30338 Len=0 TStamp=130000 TSecr=226590 SLE=1326 SRE=1044
35	17.579066	10.124.211.96	10.124.211.200	TCP	1391	[TCP Out-of-Order] 80 - 33022 [ACK] Seq=1 Ack=324 Win=15662 Len=0 TStamp=226591 TSecr=129989
36	17.579095	10.124.211.200	10.124.211.96	TCP	66	33022 - 80 [ACK] Seq=324 Ack=1344 Win=33280 Len=0 TStamp=130008 TSecr=226591
37	22.579448	10.124.211.200	10.124.211.96	TCP	66	33022 - 80 [FIN, ACK] Seq=324 Ack=1344 Win=33280 Len=0 TStamp=131258 TSecr=226591
38	22.584131	10.124.211.96	10.124.211.200	TCP	66	80 - 33022 [FIN, ACK] Seq=1344 Ack=324 Win=15552 Len=0 TStamp=227042 TSecr=130008
39	22.584144	10.124.211.200	10.124.211.96	TCP	66	33022 - 80 [ACK] Seq=325 Ack=1345 Win=33280 Len=0 TStamp=131259 TSecr=227042
40	22.625048	10.124.211.96	10.124.211.200	TCP	66	80 - 33022 [ACK] Seq=1345 Ack=325 Win=15662 Len=0 TStamp=227852 TSecr=131258
41	25.101234	10.124.211.200	10.124.211.96	TCP	74	33024 - 80 [SYN] Seq=0 Win=20480 Len=0 MSS=1460 SACK_PERM=1 TStamp=131888 TSecr=0 WS=128
42	25.141132	10.124.211.96	10.124.211.200	TCP	74	80 - 33024 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1337 SACK_PERM=1 TStamp=131888 TSecr=131888 WS=4
43	25.141157	10.124.211.200	10.124.211.96	TCP	66	33024 - 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TStamp=131899 TSecr=228481
44	25.141361	10.124.211.200	10.124.211.96	HTTP	389	GET /newsdetails.php?id=26%20and%201=2--%20- HTTP/1.1
45	25.189026	10.124.211.96	10.124.211.200	TCP	66	80 - 33024 [ACK] Seq=1 Ack=324 Win=15662 Len=0 TStamp=228492 TSecr=131899
46	25.187157	10.124.211.96	10.124.211.200	HTTP	1285	HTTP/1.1 200 OK (text/html)
47	25.187168	10.124.211.200	10.124.211.96	TCP	66	33024 - 80 [ACK] Seq=324 Ack=1220 Win=32128 Len=0 TStamp=131910 TSecr=228493
48	30.188296	10.124.211.200	10.124.211.96	TCP	66	33024 - 80 [FIN, ACK] Seq=324 Ack=1220 Win=32128 Len=0 TStamp=133160 TSecr=228493

هتبص هنا فالتفاصيل بتاعت ال packet

```

Frame 20: 373 bytes on wire (2984 bits), 373 bytes captured (2984 bits)
Ethernet II, Src: 1a:3a:46:bf:43:91 (1a:3a:46:bf:43:91), Dst: VMware_a1:4e:f0 (00:50:56:a1:4e:f0)
Internet Protocol Version 4, Src: 10.124.211.200, Dst: 10.124.211.96
Transmission Control Protocol, Src Port: 33020, Dst Port: 80, Seq: 951, Ack: 5315, Len: 307
Hypertext Transfer Protocol
    GET /newsdetails.php?id=26%27 HTTP/1.1\r\n
    Host: 10.124.211.96\r\n
    User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    Connection: keep-alive\r\n
\r\n
[Full request URI: http://10.124.211.96/newsdetails.php?id=26%27]
[HTTP request 4/4]
[Prev request in frame: 15]
[Response in frame: 23]

```

- هتبص عند جزء ال user - Agent زي مالصورة موضحه هنلاقي
ان ال attack استخدم متصفح ال fire fox وهو بيتعت ال attacker او ال SQL injection Attack فدا بيستخدم طريقة manual .

- طب عاوزين نعرف هل ال attack نجح فال attacker بتاعه ولا لاء ؟!

- هنروح لل packet دى هنلاقي ال attacker لسه شغال بيعمل SQL injection attack بس في حاجه غريبه !!!

No.	Time	Source	Destination	Protocol	Length Info
56	34.163959	10.124.211.200	10.124.211.96	HTTP	266 GET /newsdetails.php?id=1 HTTP/1.1
57	34.169322	10.124.211.96	10.124.211.200	TCP	74 80 - 33028 [SYN, ACK] Seq=0 Ack=1 Win=14400 Len=0 MSS=1337 SACK_PERM=1 TSeqval=230739 TSeqcr=134145 WS=4
58	34.169364	10.124.211.200	10.124.211.96	TCP	66 33028 - 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSeqval=134150 TSeqcr=230739
59	34.180457	10.124.211.200	10.124.211.96	TCP	74 33030 - 80 [SYN, ACK] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSeqval=134158 TSeqcr=0 WS=128
60	34.202983	10.124.211.96	10.124.211.200	TCP	66 80 - 33026 [ACK] Seq=1 Ack=201 Win=15552 Len=0 TSeqval=230747 TSeqcr=134154
61	34.206414	10.124.211.96	10.124.211.200	TCP	1391 [TCP segment of a reassembled PDU]
62	34.206432	10.124.211.200	10.124.211.96	TCP	66 33026 - 80 [ACK] Seq=201 Ack=1328 Win=32128 Len=0 TSeqval=134165 TSeqcr=230748
63	34.206495	10.124.211.96	10.124.211.200	HTTP	82 HTTP/1.1 200 OK (text/html)
64	34.206501	10.124.211.200	10.124.211.96	TCP	66 33026 - 80 [ACK] Seq=201 Ack=1342 Win=32128 Len=0 TSeqval=134166 TSeqcr=230748
65	34.206500	10.124.211.96	10.124.211.200	TCP	66 80 - 33026 [FIN, ACK] Seq=1342 Ack=201 Win=15552 Len=0 TSeqval=230748 TSeqcr=134154
66	34.207131	10.124.211.200	10.124.211.96	TCP	66 33026 - 80 [FIN, ACK] Seq=201 Ack=1348 Win=32128 Len=0 TSeqval=134168 TSeqcr=230748
67	34.218945	10.124.211.96	10.124.211.200	TCP	74 80 - 33030 [SYN, ACK] Seq=0 Ack=1 Win=14400 Len=0 MSS=1337 SACK_PERM=1 TSeqval=230751 TSeqcr=134158 WS=4
68	34.218972	10.124.211.200	10.124.211.96	TCP	66 33030 - 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSeqval=134168 TSeqcr=230751
69	34.230022	10.124.211.200	10.124.211.96	TCP	74 33032 - 80 [SYN, ACK] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSeqval=134171 TSeqcr=0 WS=128
70	34.248217	10.124.211.96	10.124.211.200	TCP	66 80 - 33026 [ACK] Seq=1348 Ack=202 Win=15552 Len=0 TSeqval=230750 TSeqcr=134166
71	34.269905	10.124.211.96	10.124.211.200	TCP	74 80 - 33032 [SYN, ACK] Seq=0 Ack=1 Win=14400 Len=0 MSS=1337 SACK_PERM=1 TSeqval=230764 TSeqcr=134171 WS=4
72	34.269934	10.124.211.200	10.124.211.96	TCP	66 33032 - 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSeqval=134181 TSeqcr=230764
73	35.126264	10.124.211.200	10.124.211.96	HTTP	266 GET /newsdetails.php?id=1 HTTP/1.1
74	35.126569	10.124.211.200	10.124.211.96	TCP	74 33034 - 80 [SYN, ACK] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSeqval=134395 TSeqcr=0 WS=128
75	35.166226	10.124.211.96	10.124.211.200	TCP	66 80 - 33028 [ACK] Seq=1 Ack=201 Win=15552 Len=0 TSeqval=230988 TSeqcr=134395
76	35.166267	10.124.211.96	10.124.211.200	TCP	74 80 - 33034 [SYN, ACK] Seq=0 Ack=1 Win=14400 Len=0 MSS=1337 SACK_PERM=1 TSeqval=230988 TSeqcr=134395 WS=4
77	35.166341	10.124.211.200	10.124.211.96	TCP	66 33034 - 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSeqval=134405 TSeqcr=230988
78	35.166906	10.124.211.96	10.124.211.200	TCP	1391 [TCP segment of a reassembled PDU]
79	35.168835	10.124.211.200	10.124.211.96	TCP	66 33028 - 80 [ACK] Seq=201 Ack=1328 Win=32128 Len=0 TSeqval=134405 TSeqcr=230988
80	35.168926	10.124.211.96	10.124.211.200	HTTP	82 HTTP/1.1 200 OK (text/html)
81	35.168943	10.124.211.200	10.124.211.96	TCP	66 33028 - 80 [ACK] Seq=201 Ack=1342 Win=32128 Len=0 TSeqval=134405 TSeqcr=230988
82	35.168962	10.124.211.96	10.124.211.200	TCP	66 80 - 33026 [FIN, ACK] Seq=1342 Ack=201 Win=15552 Len=0 TSeqval=230988 TSeqcr=134395
83	35.170154	10.124.211.200	10.124.211.96	TCP	66 23028 - 80 [FIN, ACK] Seq=201 Ack=1342 Win=32128 Len=0 TSeqval=134405 TSeqcr=230988

- لو بصيت فال **packet** ال فوق هتلaci ان ال **GET Request** جايلك باين انه عادي مفيهوش اكواود ال **SQL Injection** ال كنا قولنا **attack** و**attack** فتقول ال **manual** علها فوق وبيعملها **attack** وقف **packet** عشان ندخل في تفاصيلها هتلaci ان الكلام مختلف حبتن متيجي نشوف !

```

Frame 56: 266 bytes on wire (2128 bits), 266 bytes captured (2128 bits)
Ethernet II, Src: 1a:3a:46:bf:43:91 (1a:3a:46:bf:43:91), Dst: VMware_a1:4e:f0 (00:50:56:a1:4e:f0)
Internet Protocol Version 4, Src: 10.124.211.200, Dst: 10.124.211.96
Transmission Control Protocol, Src Port: 33026, Dst Port: 80, Seq: 1, Ack: 1, Len: 260
Hypertext Transfer Protocol
  GET /newsdetails.php?id=1 HTTP/1.1\r\n
    Accept-Encoding: gzip,deflate\r\n
    Host: 10.124.211.96\r\n
    Accept: */*\r\n
    User-Agent: sqlmap/1.1.4#stable (http://sqlmap.org)\r\n
    Connection: close\r\n
    Cache-Control: no-cache\r\n
\r\n
[Full request URI: http://10.124.211.96/newsdetails.php?id=1]
[HTTP request 1/1]
[Response in frame: 63]

```

- هتبص تلaci فال **user -Agent** ال استخدمنا ال **tool** ال وهي ال **SQL map** ودي من أشهر الادوات لاختراق ال **SQL injection** وعمل ال **data base** بس عن طريق **Tool** وليس **attacker** وال **manual** عموما بيبقا عنده

سيناريو ماشي عليه انه يجرب **manual** الاول وبعدين لو لاقى ال **SQL map** دى معرضه للاختراق بيدع يروح لل **tools** زي **page** ويديها ال **exploits** وهي تقدر تجرب ال **page URL** لحد ميحصل ال **attack** فعلا.

- انت ك **attack** كدا عرفت ال **IP** المتسبب فال **Threat hunter** ممكن تروح تعمله **rule** على حسب ال **block** او **deny** ال عندك.

HTTPS Traffic:

- ال هو **Hyper Text Transfer Protocol Secure** شغال ف **Layer 4** ال هي **Application Layer** ودا النسخه ال **HTTP** من ال **secure** قولنا عليه **SSL Protocol** عشان بيستخدم ال **Secure** ال هو **Secure Socket Layer**.

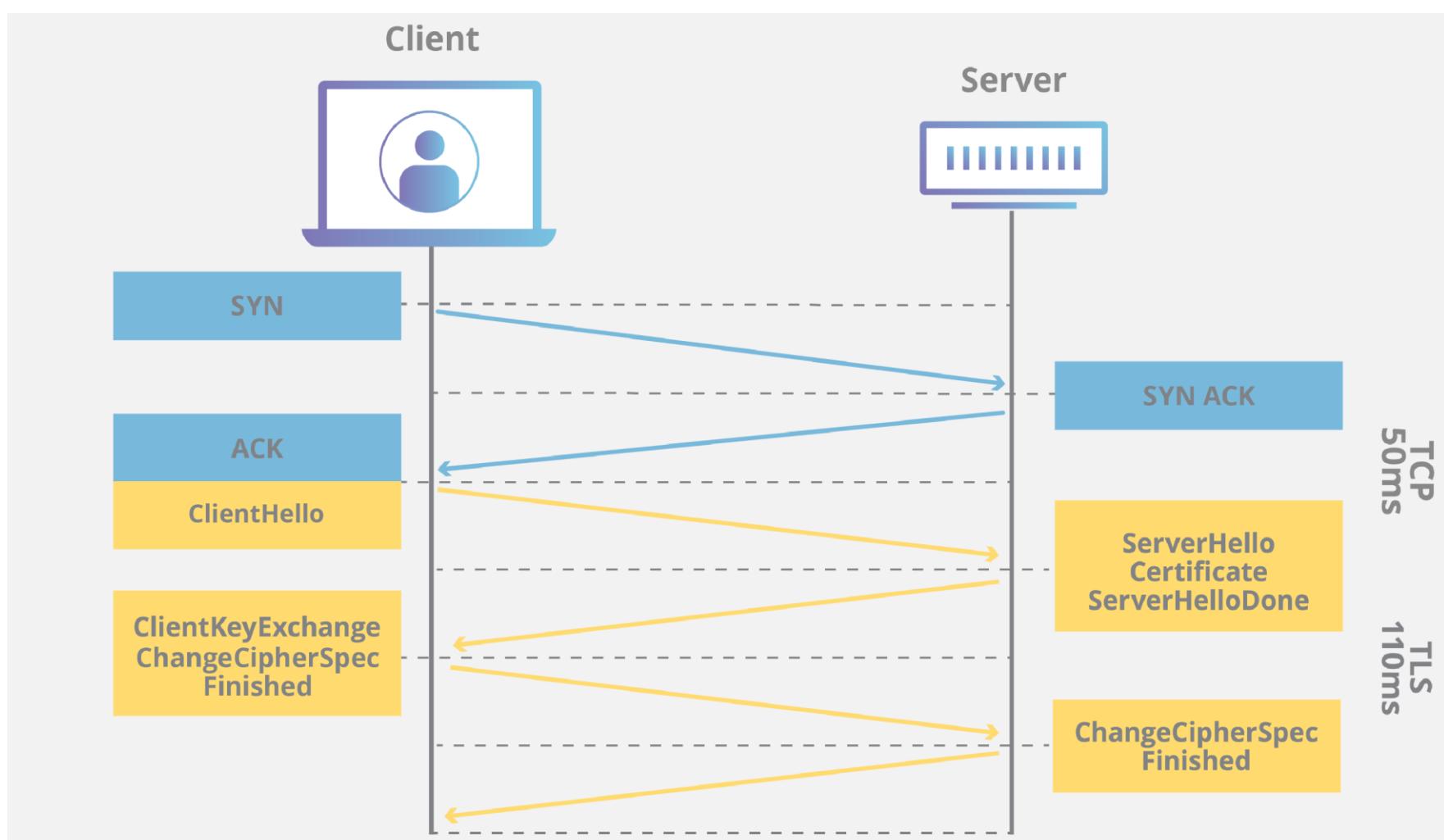
- ال **HTTPS** بيعمل **Hand shake** عن طريق **establish** زى بتاعت ال **TCP** ال قولنا عليها فوق ولكن اكثر تعقيدا منها ودا المسؤول عنه بروتوكول ال **SSL** بمعنى لازم ال **client** وال **server** يتتفقوا على نفس **SSL Version** قبل ميحصل الاتصال بينهم ولازم يتتفقوا على ال **Cryptography Algorithm** قبل ميحصل الاتصال

بينهم وال بحدد الكلام دا هو ال **SSL** لانه المسؤول انه يعمل **secure** . **HTTPS** بتاعت ال **Session** لل **client**

- فال **SSL** مسؤول انه يوفق مفاتيح التشفير بين الطرفين ال عند ال **server** والا عند ال **client**

- ولازم قبل ميحصل الاتصال يحصل **Authentication** مابين الطرفين

- ولازم قبل ميحصل الاتصال يتتفقو الطرفين على **public Key** . ولازم قبل ميحصل الاتصال يستخدموه عشان يعملو **establish** للاتصال.



- الحاجات دي كان لازم تتقابل عشان هنعزها بعدين واحنا بنحل ال traffic عالي نعرف الفرق ما بين ال **Normal** وال **Suspicious** عشان نفرقهم

Normal HTTPS Traffic	Suspicious HTTPS Traffic
Port 443, TCP Port 8443, TCP (used as alternate)	Malicious binaries (backdoors), scripts, web shells, etc. will use this port because typically in all corporate environments the port is open.
Encrypted traffic	If the traffic is not encrypted and Secure Sockets Layer packet details are empty within packet details, then that will fall under suspicious.
Web server typically in FQDN format.	Server will point to an IP address instead of FQDN format.

- اول حاجه فال **Normal** هتلاقیه بیشتعل على **port 443** او لو شغال **port** زی مشرحا فال **HTTP** فوق هتلاقیه شغال على **port 8443** وبیستخدم برتوکول **TCP** من الطبقه ال تحته ال هي **. Transport Layer**

- وهتلاقي **encrypted** بیکون **normal** ال **HTTPS** ال **traffic**

- واما تيجي تفتح موقع ما بال **HTTPS** هتلاقیك بتفتحه بال **FQDN** مش بال **IP**.

- تعالی نشووف ايه النظام فال **Suspicious** هتلاقي انه بیجیاک برضه عن طريق ال **HTTPS** المفتوحه عندك ال هي **443 , 8443** بس بدل میبعثتك **normal traffic** هتلاقیه بیبعثتك **web shell** فعشان تطلع ال **traffic** على كل ال **investigation** لازم تعمل **suspicious** **HTTPS** الخاصه بال

- کمان هتلaci ال suspicious فیه بیکون not traffic ال suspicious

port يعني هتلaci مستخدم ال SSL و هتلaci رایح ل encrypted

ولكن لو وقفت ع ال packet وضعفت right click و منها 443

هتلaci المحتوي بتاع ال packet مقروء مش مشفر .

- و هتلaci ال suspicious بعض الاحيان بیستخدم ال IP بدل من ال FQDN

فانت لو لقيت traffic جايلاك ب IP داخلي واقع نفس ال

دا عادي ممکن يكون عاوز يروح ل server subdomain معاك نفس الشبکه انما لو لقيته عاوز يروح ل IP خارجي خارج شبكتك

انت تبدع تشک فال suspicious traffic دا ممکن يكون packet فتروح تعمل جوا ال Deep Investigation اکتر .

- تعالی ناخد wire shark بال capture عشان الصورة توضح اکتر

....

No.	Time	Source	Destination	Protocol	Length	Info
3 0.049179	10.54.15.100	10.54.15.15	TCP	74 39678 - 80 [Syn] Seq=0 Win=29296 Len=0 MSS=1460 SACK_PEEK=1 TSval=2640960 TSecr=0 WS=128		
4 11.050121	10.54.15.100	10.54.15.15	TCP	74 40112 - 81 [Syn] Seq=0 Win=29296 Len=0 MSS=1460 SACK_PEEK=1 TSval=2641000 TSecr=0 WS=128		
5 14.560130	10.54.15.100	10.54.15.15	TCP	74 441 - 40112 [Syn] ACK=1 Seq=1 Win=1460 MSS=1460 TSval=2641037 TSecr=2641037 WS=14		
11.14.560374	10.54.15.100	10.54.15.15	TCP	66 45112 - 443 [ACK] Seq=1 ACK=158 Win=1552 Len=0 TSval=2644612 TSecr=2644612		
12.14.860618	10.54.15.100	10.54.15.15	TLSv1.2	232 Client Hello		
13.14.654485	10.54.15.15	10.54.15.100	TCP	66 443 - 45112 [ACK] Seq=1 ACK=158 Win=1552 Len=0 TSval=2644612 TSecr=2644612		
14.14.658808	10.54.15.15	10.54.15.100	TLSv1.2	195 Server Hello, Certificate		
15.14.858839	10.54.15.100	10.54.15.15	TCP	66 45112 - 443 [ACK] Seq=158 Win=32128 Len=0 TSval=2644624 TSecr=2644624		
16.14.860183	10.54.15.100	10.54.15.15	TLSv1.2	72 Alert (Level: Fatal, Description: Unknown CA)		
17.14.661622	10.54.15.100	10.54.15.15	TCP	66 45112 - 443 [Syn] ACK=158 Win=32128 Len=0 TSval=2644625 TSecr=2644625		
18.14.661618	10.54.15.15	10.54.15.100	TLSv1.2	119 Server Key Exchange,Server Hello Done		
19.14.861102	10.54.15.100	10.54.15.15	TCP	66 45112 - 443 [Syn] ACK=158 Win=32128 Len=0 TSval=2644626 TSecr=2644626		
20.14.706659	10.54.15.15	10.54.15.100	TCP	66 443 - 45112 [ACK] Seq=1404 ACK=1552 Win=1460 TSval=2644627 TSecr=2644627		
21.14.706658	10.54.15.100	10.54.15.15	TCP	66 45112 - 443 [ACK] Seq=178 Win=1460		
22.22.541446	10.54.15.100	10.54.15.15	TCP	74 45114 - 443 [Syn] ACK=158 Win=29296 Len=0 MSS=1460 SACK_PEEK=1 TSval=2644630 TSecr=2644630 WS=128		
23.22.584353	10.54.15.15	10.54.15.100	TCP	66 45114 - 40112 [Syn] ACK=158 Win=29296 Len=0 TSval=2644630 TSecr=2644630 WS=14		
24.22.594291	10.54.15.100	10.54.15.15	TCP	66 45114 - 443 [ACK] Seq=1 Win=29296 Len=0 TSval=2644630 TSecr=2644630		
25.22.584487	10.54.15.100	10.54.15.15	TLSv1.2	232 Client Hello		
26.22.643672	10.54.15.15	10.54.15.100	TCP	66 443 - 45114 [ACK] Seq=1 ACK=158 Win=1552 Len=0 TSval=2644632 TSecr=2644632		
27.22.654983	10.54.15.15	10.54.15.100	TLSv1.2	195 Server Hello, Certificate		
28.22.646975	10.54.15.100	10.54.15.15	TCP	66 45114 - 443 [ACK] Seq=158 Win=32128 Len=0 TSval=2644631 TSecr=2644631		
29.22.649560	10.54.15.15	10.54.15.100	TLSv1.2	170 Server Key Exchange,Server Hello Done		
30.22.649666	10.54.15.100	10.54.15.15	TCP	66 45114 - 443 [Syn] ACK=158 Win=32128 Len=0 TSval=2644632 TSecr=2644632		
31.22.651191	10.54.15.100	10.54.15.15	TLSv1.2	192 Client Key Exchange, Change Cipher Spec, Hello Request, Hello Response		
32.22.651230	10.54.15.100	10.54.15.15	TLSv1.2	370 Application Data		
33.22.650218	10.54.15.15	10.54.15.100	TCP	66 443 - 45114 [ACK] Seq=158 Win=1460 Len=0 TSval=2644632 TSecr=2644632		
34.22.706636	10.54.15.15	10.54.15.100	TLSv1.2	324 New Session Ticket, Change Cipher Spec, Encrypted Handshake Message		
35.22.706683	10.54.15.15	10.54.15.100	TLSv1.2	370 Application Data, Application Data, Application Data, Application Data		
36.22.704476	10.54.15.100	10.54.15.15	TCP	66 45114 - 443 [ACK] Seq=904 Ack=2382 Win=32128 Len=0 TSval=2644636 TSecr=2644636		
37.22.715752	10.54.15.100	10.54.15.15	TLSv1.2	370 Application Data		
38.22.706697	10.54.15.15	10.54.15.100	TLSv1.2	370 Application Data, Application Data, Application Data, Application Data		
39.22.807472	10.54.15.100	10.54.15.15	TCP	66 45114 - 443 [ACK] Seq=913 Ack=2382 Win=32128 Len=0 TSval=2644632 TSecr=2644632		
40.27.787067	10.54.15.100	10.54.15.15	TLSv1.2	97 Encrypted Alert		
41.27.706613	10.54.15.100	10.54.15.15	TCP	66 45114 - 443 [Syn] ACK=158 Win=1460 TSval=2644632 TSecr=2644632		
42.27.771140	10.54.15.15	10.54.15.100	TLSv1.2	97 Encrypted Alert		
43.27.771140	10.54.15.100	10.54.15.15	TCP	66 45114 - 443 [Syn] ACK=158 Win=1460 TSval=2644632 TSecr=2644632		
44.27.771140	10.54.15.15	10.54.15.100	TCP	66 45114 - 443 [Syn] ACK=158 Win=1460 TSval=2644632 TSecr=2644632		

- هتبص عندك فال **TCP** هتلافقیان اول حاجه حصل **capture** عادي وبعده بده برتوکول ال **SSL** يدخل ع ال **connection** وبعد ال **server** بين ال **client** وال **server** بده ال **Hello** يبعث لل **client** ال **certificate** ورض عليه ال **Client** بال **connection** فاحنا قولنا بتاعته وهكذا الي نهاية ال **certificate** عال **traffic** مشفر ورایح ل **port 443** فدا زی مشرحنا فوق تمام.

- تعالى نتعمق جوا ال **packet** شويه عشان نتعرف عال **Normal packet** **specific** بشكل

```

Frame 25: 233 bytes on wire (1864 bits), 233 bytes captured (1864 bits)
Ethernet II, Src: 26:11:59:88:02 (26:11:59:88:02), Dst: vmware_a1:61:66 (00:50:56:a1:61:66)
Internet Protocol Version 4, Src: 10.54.15.100, Dst: 10.54.15.15
Transmission Control Protocol, Src Port: 45114, Dst Port: 443, Seq: 1, Ack: 1, Len: 167
Secure Sockets Layer
  * TLSv1.2 Record Layer: Handshake Protocol: Client Hello
    Content Type: Handshake (22) ←
    Version: TLS 1.0 (0x0301) ←
    Length: 162 ←
  * Handshake Protocol: Client Hello ←
    Handshake Type: Client Hello (1) ←
    Length: 158 ←
    Version: TLS 1.2 (0x0303) ←
  * Random ←
    Session ID Length: 0 ←
    Cipher Suites Length: 22 ←
  * Cipher Suites (22 suites) ←
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b) ←
    Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f) ←
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a) ←
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009) ←
    Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013) ←
    Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) ←
    Cipher Suite: TLS_DHE_RSA_WITH_AES_128_CBC_SHA (0x0033) ←
    Cipher Suite: TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x0039) ←
    Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f) ←
    Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035) ←
    Cipher Suite: TLS_RSA_WITH_3DES_EDE_CBC_SHA (0x000a) ←
    Compression Methods Length: 1 ←
  - Compression Methods (1 method) ←
    Compression Method: null (0) ←
    Extensions Length: 96 ←
  * Extension: renegotiation_info ←
  * Extension: elliptic_curves ←
  * Extension: ec_point_formats ←
  * Extension: SessionTicket TLS ←
  * Extension: next_protocol_negotiation ←
  * Extension: Application Layer Protocol Negotiation ←
  * Extension: status_request ←
  * Extension: signature_algorithms ←

```

- فال **packet** ال قدامك دي هتلافقی ال **IP** ال هو **15.100** رایح لل **IP** ال هو **15.15** وطالع من **source port 45114** رایح ل **SSL** وهتلافقیه مستخدم برتوکول ال **destination port 443** فدي اول اشارة على ان ال **traffic** دا سليم .

- وکمان هتلaci ال content packet علی عکس ال traffic عادي لكن لوبصيت على ال suspicious هتلaci ال traffic بداع ال packet هتلaciه فارغ لانه دي بتكون content مصنعيه بواسطه ادوات ال Attacker crafted packet

- و هتلaci ال client hello server فيه محتويات زي version ال مبينهم وال hand shake ال هي ال content type بداع ال SSL ال هو " SSL " بس دا الاسم الحديث بداعه ال هتشوفه فال wire shark

تعالي نبص عال server ال جاييه من ال packet

```

* Frame 27: 1391 bytes on wire (11128 bits), 1391 bytes captured (11128 bits)
* Ethernet II, Src: vmware_81: f4:d0 (00:50:56:81:f4:d0), Dst: 26:11:59:58:53:02 (26:11:59:58:53:02)
* Internet Protocol Version 4, Src: 10.54.15.15, Dst: 10.54.15.100
* Transmission Control Protocol, Src Port: 443, Dst Port: 45114, Seq: 1, Ack: 166, Len: 1325
* SECURITY OVERLAYS Layer
  - TLSv1.2 Record Layer: Handshake Protocol: Server Hello
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 61
  - Handshake Protocol: Server Hello
    Handshake Type: Server Hello (2)
    Length: 57
    Version: TLS 1.2 (0x0303)
  - Random
    GMT Unix Time: May 23, 2017 13:27:38.600000000 EDT
    Random Bytes: 2000d7125e0e0022e9441d5121c77b5e3cb00e6b5fd2242e...
    Session ID Length: 0
    Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc002f1)
    Compression Method: null (0)
    Extensions Length: 17
    * Extension: renegotiation_info
    * Extension: ec_point_formats
    * Extension: SessionTicket TLS
  - TLSv1.2 Record Layer: Handshake Protocol: Certificate
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 1011
  - Handshake Protocol: Certificate
    Handshake Type: Certificate
    Length: 1007
    Certificates Length: 1004
    - Certificates (1004 bytes)
      Certificate Length: 1001
      - Certificate: 308203e5308202cda0030020102020000d9e303c1875e1375... (pkcs-9-at-emailAddress)
        - signedCertificate
          version: v3 (2)
          serialNumber: -27733007900000007947
          - signature (sha1WithRSAEncryption)
          - issuer: rdnSequence (0)
          - validity
          - subject: rdnSequence (0)
          - subjectPublicKeyInfo
          - extensions: 3 items
        - algorithmIdentifier (sha1WithRSAEncryption)
        Padding: 0
        encrypted: 2326c8138a0c0a00ff004ee0e6900ca06f34ae0dcf343ae0...

```

- هنبص نلاقي ال server hello رد عال client server وفيه محتوياته برضه ال هي نوع اصدار ال TLS ال وافق عليه ال

- وزى مالصورة موضحه هتلaci ال client باعت لى server هيسخدمها ال encryption algorithm وال certificate Cypher هى اثناء ارسال ال packet لى client ومفاتيح التشفير الخاصه بيها وهكذا

```

Frame 29: 170 bytes on wire (1360 bits), 170 bytes captured (1360 bits)
Ethernet II, Src: VMware_81:f4:d0 (00:50:56:a1:f4:d0), Dst: 26:11:59:88:53:02 (26:11:59:88:53:02)
Internet Protocol Version 4, Src: 10.54.15.15, Dst: 10.54.15.100
Transmission Control Protocol, Src Port: 443, Dst Port: 45114, Seq: 1326, Ack: 168, Len: 104
[2 Reassembled TCP Segments (338 bytes): #27(243), #29(95)]
Secure Sockets Layer
  * TLSv1.2 Record Layer: Handshake Protocol: Server Key Exchange
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 333
      - Handshake Protocol: Server Key Exchange
        Handshake Type: Server Key Exchange (12)
        Length: 329
          - EC Diffie-Hellman Server Params
            Curve Type: named_curve (0x03)
            Named Curve: secp256r1 (0x0017)
            Pubkey Length: 65
            Pubkey: 04481daa41bf8a836acffe3ce86c112c109af374b2ef1326...
          * Signature Hash Algorithm: 0x0401
            Signature Hash Algorithm Hash: SHA256 (4)
            Signature Hash Algorithm Signature: RSA (1)
            Signature Length: 256
            Signature: Bf3e65872e9d3bb17841322323a621d35e14faf32886ee82...
  * Secure Sockets Layer
  * TLSv1.2 Record Layer: Handshake Protocol: Server Hello Done
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 4
      - Handshake Protocol: Server Hello Done
        Handshake Type: Server Hello Done (14)
        Length: 0

```

- هتلaci ال server رض عليك برضه فال packet ال بعدها بال client بتاعه ال هو مفتاح التبادل ال بينه وبين ال key exchange اثناء ارسال ال data ودى الخطوة الثالثه ف عمليه انشاء ال . Establish connection عشان تكون SSL/TLS

```

Frame 31: 192 bytes on wire (1536 bits), 192 bytes captured (1536 bits)
Ethernet II, Src: 26:11:59:88:53:02 (26:11:59:88:53:02), Dst: VMware_81:61:66 (00:50:56:a1:61:66)
Internet Protocol Version 4, Src: 10.54.15.100, Dst: 10.54.15.15
Transmission Control Protocol, Src Port: 45114, Dst Port: 443, Seq: 168, Ack: 1430, Len: 126
Secure Sockets Layer
  * TLSv1.2 Record Layer: Handshake Protocol: Client Key Exchange
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 70
      - Handshake Protocol: Client Key Exchange
        Handshake Type: Client Key Exchange (16)
        Length: 66
          - EC Diffie-Hellman Client Params
            Pubkey Length: 65
            Pubkey: 04496c4e42312aa0f1b9855834438ee5d7f97745533bfc5e...
  * TLSv1.2 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
    Content Type: Change Cipher Spec (20)
    Version: TLS 1.2 (0x0303)
    Length: 1
    Change Cipher Spec Message
  * TLSv1.2 Record Layer: Handshake Protocol: Multiple Handshake Messages
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 40
      - Handshake Protocol: Hello Request
        Handshake Type: Hello Request (0)
        Length: 0
      - Handshake Protocol: Hello Request
        Handshake Type: Hello Request (0)
        Length: 0

```

- ودي اخر **SSL/TLS connection** فال **packet** عشان يبدع اتصال بال **client** ودا هتلافي ال **HTTPS** بيبعت **keys exchange** بتاعته لل **server** عشان يتم ال **connection** ما بينهم .

```

▶ Frame 34: 324 bytes on wire (2592 bits), 324 bytes captured (2592 bits)
▶ Ethernet II, Src: Vmware_a1:f4:d0 (00:50:56:a1:f4:d0), Dst: 26:11:59:88:53:02 (26:11:59:88:53:02)
▶ Internet Protocol Version 4, Src: 10.54.15.15, Dst: 10.54.15.100
▶ Transmission Control Protocol, Src Port: 443, Dst Port: 45114, Seq: 1430, Ack: 604, Len: 258
└ Secure Sockets Layer
    ▶ TLSv1.2 Record Layer: Handshake Protocol: New Session Ticket ←
        Content Type: Handshake (22)
        Version: TLS 1.2 (0x0303)
        Length: 202
    ▶ Handshake Protocol: New Session Ticket
        Handshake Type: New Session Ticket (4)
        Length: 198
    ▶ TLS Session Ticket
        Session Ticket Lifetime Hint: 300
        Session Ticket Length: 192
        Session Ticket: c87ec842e1a7e7c2fd503729435f618d50f9e59487ae8647...
    ▶ TLSv1.2 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec ←
        Content Type: Change Cipher Spec (20)
        Version: TLS 1.2 (0x0303)
        Length: 1
        Change Cipher Spec Message
    ▶ TLSv1.2 Record Layer: Handshake Protocol: Encrypted Handshake Message ←
        Content Type: Handshake (22)
        Version: TLS 1.2 (0x0303)
        Length: 40
        Handshake Protocol: Encrypted Handshake Message

```

- هبص هنا فنلاقي بعد اما ال **server** وال **client** اتفقو على كل حاجه وكله تمام بدء يفتح ما بينهم **ticket** او **Session** زي مال **packet** موضحة وبيكون ليها **time** معين تنتهي فيه.... لو قفلت الاتصال او فقدت ال **Session** دي بتحتاج تعمل الخطوات دي من الاول عشان تضمن سلامه وصول ال **HTTPS Traffic** بدون تعرضه للتسلل او الاختراق .

```

▶ Frame 35: 770 bytes on wire (6160 bits), 770 bytes captured (6160 bits)
▶ Ethernet II, Src: Vmware_a1:f4:d0 (00:50:56:a1:f4:d0), Dst: 26:11:59:88:53:02 (26:11:59:88:53:02)
▶ Internet Protocol Version 4, Src: 10.54.15.15, Dst: 10.54.15.100
▶ Transmission Control Protocol, Src Port: 443, Dst Port: 45114, Seq: 1688, Ack: 604, Len: 704
└ Secure Sockets Layer
    ▶ TLSv1.2 Record Layer: Application Data Protocol: http-over-tls
        Content Type: Application Data (23)
        Version: TLS 1.2 (0x0303)
        Length: 32
        Encrypted Application Data: bb006752c8e53aef6bb13c15ff590c829883685da8b96e44...

```

- بعد اما الخطوات ال شرحتها فالجزء ال فات هتلاقی ال **data** ال بتترسل مابین الطرفین **encrypted** ودا ال احنا عاوزين نوصله ودا ال موضحاه ال **packet**

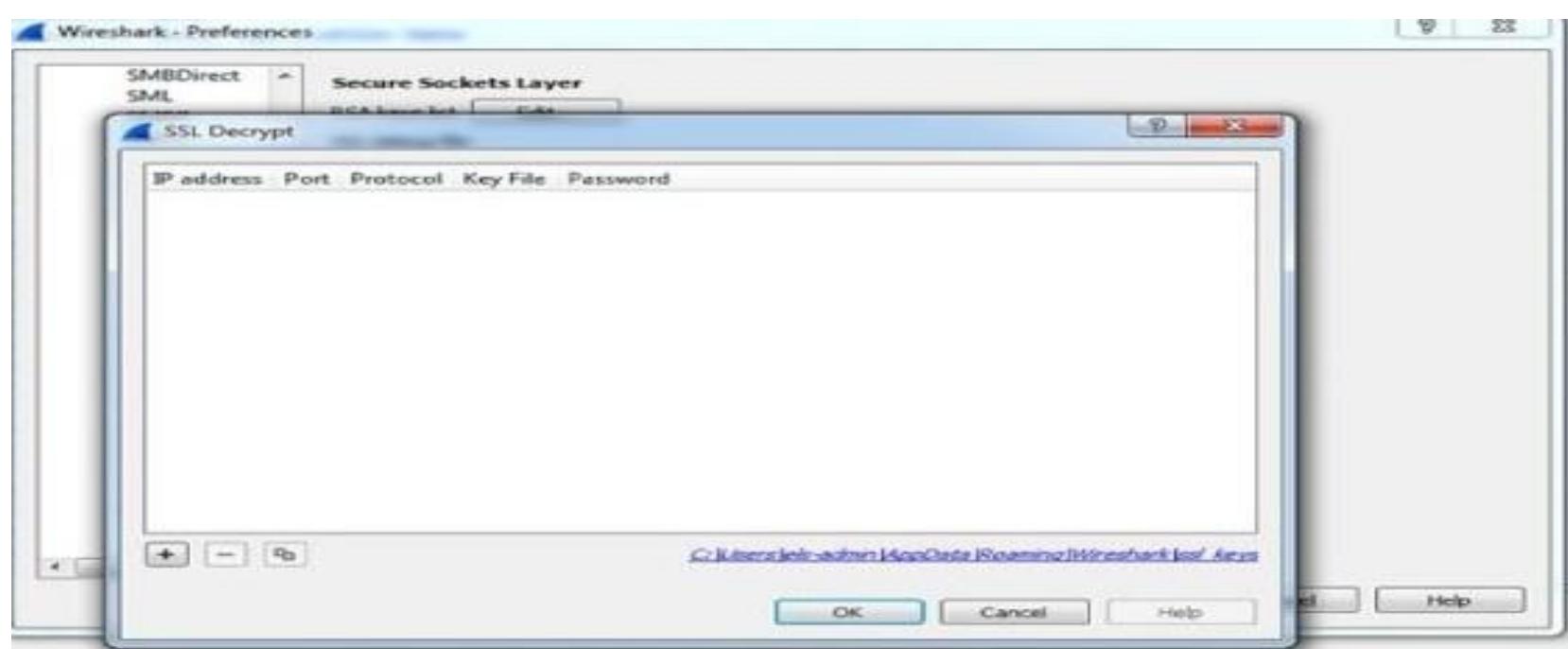
Time	Source	Destination	Protocol	Length	Info
1 0.000000	127.0.0.1	127.0.0.1	TCP	74	38713 → 443 [SYN] Seq=0 Win=32767 Len=0 MSS=16386 SACK_PERM=1 TSval=525562106 TSecr=0 NS=1
2 0.000021	127.0.0.1	127.0.0.1	TCP	74	443 → 38713 [SYN, ACK] Seq=0 Ack=1 Win=32767 Len=0 MSS=16386 SACK_PERM=1 TSval=525562115 TSecr=525562106 NS=1
3 0.000037	127.0.0.1	127.0.0.1	TCP	66	38713 → 443 [ACK] Seq=1 Ack=1 Win=32767 Len=0 TSval=525562115 TSecr=525562115
4 0.000158	127.0.0.1	127.0.0.1	SSLv2	171	Client Hello
5 0.000178	127.0.0.1	127.0.0.1	TCP	66	443 → 38713 [ACK] Seq=1 Ack=106 Win=32767 Len=0 TSval=525562115 TSecr=525562115
6 0.0002168	127.0.0.1	127.0.0.1	SSLv3	995	Server Hello, Certificate, Server Hello Done
7 0.0002599	127.0.0.1	127.0.0.1	TCP	66	38713 → 443 [ACK] Seq=106 Ack=930 Win=32767 Len=0 TSval=525562117 TSecr=525562117
8 2.008933	127.0.0.1	127.0.0.1	SSLv3	278	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
9 2.822778	127.0.0.1	127.0.0.1	SSLv3	141	Change Cipher Spec, Encrypted Handshake Message
10 2.822809	127.0.0.1	127.0.0.1	TCP	66	38713 → 443 [ACK] Seq=318 Ack=1005 Win=32767 Len=0 TSval=525564938 TSecr=525564938
11 2.833071	127.0.0.1	127.0.0.1	SSLv3	503	Application Data
12 2.873275	127.0.0.1	127.0.0.1	TCP	66	443 → 38713 [ACK] Seq=1005 Ack=755 Win=32767 Len=0 TSval=525564938 TSecr=525564948
13 2.938485	127.0.0.1	127.0.0.1	SSLv3	103	Encrypted Handshake Message
14 2.938758	127.0.0.1	127.0.0.1	SSLv3	183	Encrypted Handshake Message
15 2.938761	127.0.0.1	127.0.0.1	TCP	66	443 → 38713 [ACK] Seq=1042 Ack=872 Win=32767 Len=0 TSval=525565054 TSecr=525565054
16 2.938999	127.0.0.1	127.0.0.1	SSLv3	1073	Encrypted Handshake Message, Encrypted Handshake Message, Encrypted Handshake Message
17 2.940026	127.0.0.1	127.0.0.1	SSLv3	337	Encrypted Handshake Message, Change Cipher Spec, Encrypted Handshake Message
18 2.943406	127.0.0.1	127.0.0.1	SSLv3	172	Change Cipher Spec, Encrypted Handshake Message
19 2.944825	127.0.0.1	127.0.0.1	SSLv3	5756	Application Data, Application Data
20 2.944864	127.0.0.1	127.0.0.1	TCP	66	38713 → 443 [ACK] Seq=1143 Ack=7845 Win=32767 Len=0 TSval=525565060 TSecr=525565059
21 2.964424	127.0.0.1	127.0.0.1	SSLv3	471	Application Data

- ف المثال دا هتلاقی فال **client** رقم 4 ان ال **packet** باعت ل **server** بيكوله عاوزين نعمل الاتصال عن طريق **SSL V2** فال **server** ال بعدها هتلاقی ال **server** بيرض عليه بيكوله لاء استخدم ال الاحدث ال هو **SSL V3** عشان يتم تشفير الاتصال ما بين **version**

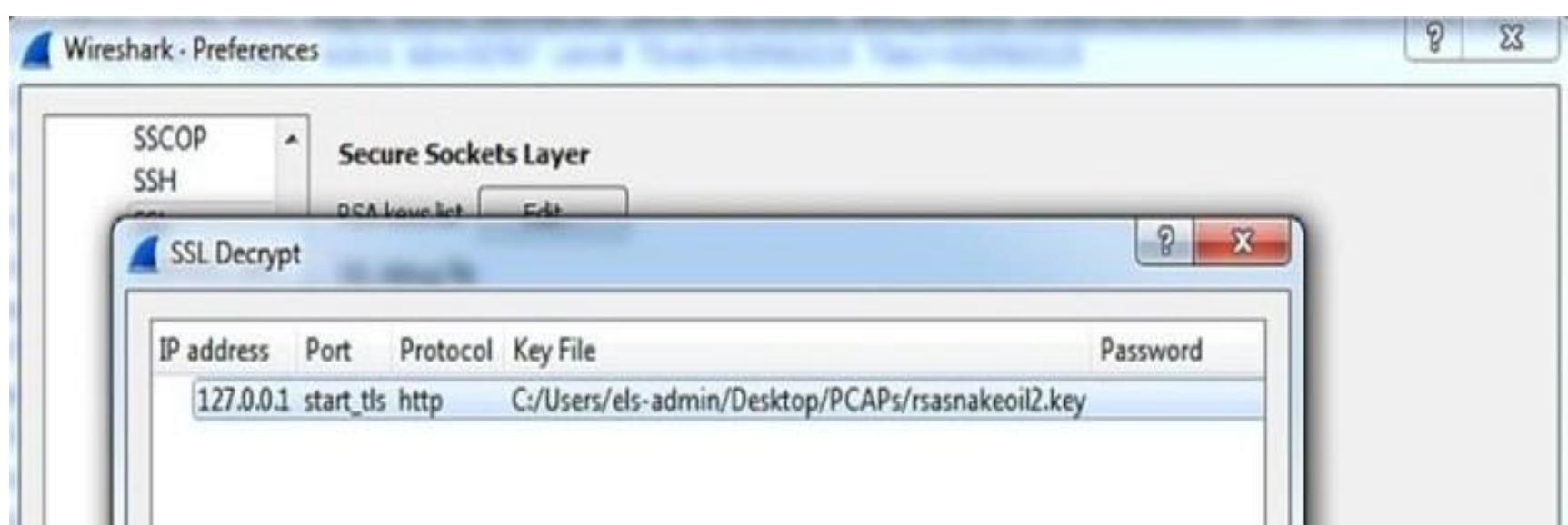
> Frame 11: 503 bytes on wire (4024 bits), 503 bytes captured (4024 bits)
> Ethernet II, Src: 00:00:00_00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00 (00:00:00:00:00:00)
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 38713, Dst Port: 443, Seq: 318, Ack: 1005, Len: 437
Secure Sockets Layer
SSLv3 Record Layer: Application Data Protocol: http-over-tls
Content Type: Application Data (23)
Version: SSL 3.0 (0x0300)
Length: 432
Encrypted Application Data: 4ac33e9d7778012cb4bc4c8a84d7b9980c2110f0fa007c16...
0040 7c 8a 17 03 00 01 b6 2a c3 3c 9d 77 79 88 2c b4
0050 b2 4c 9a 84 d7 09 5d 9c 23 3b 10 f0 40 7c 36 b6
0060 72 fd 22 42 4f ad 5d 4a d0 a0 6f aa 44 9c 62 9d
0070 1b c5 fc e9 3c 5c 0f e8 85 9b 9c 95 a4 38 6c d2 d5
0080 05 20 ab 81 fd 18 94 73 b6 d7 a6 c3 dd 74 d2 9d
0090 3e 8f 21 6d 78 24 ca 3c 2b 24 36 12 7a 36a 9c ac
00a0 4e 3c a8 fb 37 3b be 9a f4 2f 98 9b 89 0a a1 0e
00b0 74 7b e3 91 84 e7 94 88 cc f8 95 7b 4a 22 f2 75
00c0 22 e8 dd 38 9c fd e9 03 23 4c 20 94 6e d5 e3 72
00d0 9e a1 1b c9 86 d6 83 55 6e 67 5d 9c b6 25 81 b6
00e0 02 91 66 d2 44 18 bc ce 7e 10 39 a6 c4 37 f5
00f0 35 d4 b3 6d 93 23 84 99 6e 69 7e d8 c9 4c bf 3f
0100 33 68 37 b7 fd 44 b6 86 2c 91 d9 25 a2 6b 5b ed
0110 9f 64 69 9e 34 c8 29 87 d9 10 b6 99 38 9a 7b cc
0120 13 3f 76 4c e6 df 24 8a d5 5d b6 91 49 8d 64 42
0130 25 a8 64 27 d4 5e 1b 3a 65 a8 73 06 05 6e 17 54
0140 b4 6b 69 7a 32 28 5d cff 5a 06 13 de 73 71 9f
0150 c9 bd 79 2a c2 e5 9b 5e 32 e7 cb 97 6e 20 ea 99
0160 4e 6a 22 f9 37 29 08 27 6d 83 f3 08 1c dd 0e dc

- زي مالمثال موضح لو بصينا عال **traffic** هتلاقیه كلام غير مفهوم ودا لانه مشفر ومستخدم ال **TLS** او **SSL**

- لو ال **server** ليه موجود معاك HTTPS Traffic ال تم ارسال ال على نفس الشبكة فاكيد هيكون معاك ال **private key** بتاعه فتقدر تفاصيله تشفير ال **encrypted data** دي ودا هنشوفوه ف المثال الجي .
- طبعا انت محصل ال **private key** ال هيفاك تشفير الداتا المشفرة بواسطه ال **public key** ال بعتله بييه الداتا مشفرة فالاول وعشان نعمل **decrypt** نفك تشفير ال traffic هتروح عندك فال **Wire Shark** وتعمل الخطوات دي
- من **edit** هتروح ل **protocols** ومنها هتروح ل **preferences** ومنها **RSA keys List** وتختر **Edit** ومنها **SSL**



- هيطلب منك ال **IP Address** وال **port** بتاع ال **server** وال **key** زي مهشوف دلوقتي والمكان بتاع **protocol**



- هتلاقي بعد كدا ال **Traffic** بتاع ال **HTTPS** اتفك و هتلاقي بدل مكان فيه **HTTP traffic** لـ **SSL certificate** بدء يظهر **decrypt** تقدر تقراه وبكدا يبقا قدرنا نعمل **Clear text** عندك ال هو **HTTPS Traffic** عشان نشووف محتواه وهفكرك دا لو معاك ال ...
ال هيفاك التشفير **private key**

Time	Source	Destination	Protocol	Length	Info
1 0.000000	127.0.0.1	127.0.0.1	TCP	74	38713 → 443 [SYN] Seq=0 Win=32767 Len=0 MSS=16384 SACK_PERM=1 TSval=525562106 TSecr=0 WS=1
2 0.000021	127.0.0.1	127.0.0.1	TCP	74	38713 → 38713 [SYN, ACK] Seq=0 Ack=1 Win=32767 Len=0 MSS=16384 SACK_PERM=1 TSval=525562115 TSecr=525562106 WS=1
3 0.000037	127.0.0.1	127.0.0.1	TCP	66	38713 → 443 [ACK] Seq=1 Ack=1 Win=32767 Len=0 TSval=525562115 TSecr=525562115
4 0.000158	127.0.0.1	127.0.0.1	SSLv2	171	Client Hello
5 0.000178	127.0.0.1	127.0.0.1	TCP	66	443 → 38713 [ACK] Seq=1 Ack=106 Win=32767 Len=0 TSval=525562115 TSecr=525562115
6 0.002168	127.0.0.1	127.0.0.1	SSLv3	995	Server Hello, Certificate, Server Hello Done
7 0.002669	127.0.0.1	127.0.0.1	TCP	66	38713 → 443 [ACK] Seq=106 Ack=938 Win=32767 Len=0 TSval=525562117 TSecr=525562117
8 2.000933	127.0.0.1	127.0.0.1	SSLv3	278	Client Key Exchange, Change Cipher Spec, Finished
9 2.022770	127.0.0.1	127.0.0.1	SSLv3	141	Change Cipher Spec, Finished
10 2.022899	127.0.0.1	127.0.0.1	TCP	66	38713 → 443 [ACK] Seq=318 Ack=1005 Win=32767 Len=0 TSval=525564938 TSecr=525564938
11 2.033671	127.0.0.1	127.0.0.1	HTTP	583	GET / HTTP/1.1
12 2.073275	127.0.0.1	127.0.0.1	TCP	66	443 → 38713 [ACK] Seq=1005 Ack=755 Win=32767 Len=0 TSval=525564948 TSecr=525564948
13 2.938445	127.0.0.1	127.0.0.1	SSLv3	183	Hello Request
14 2.938758	127.0.0.1	127.0.0.1	SSLv3	183	Client Hello
15 2.938761	127.0.0.1	127.0.0.1	TCP	66	443 → 38713 [ACK] Seq=1042 Ack=872 Win=32767 Len=0 TSval=525565054 TSecr=525565054
16 2.938999	127.0.0.1	127.0.0.1	SSLv3	1873	Server Hello, Certificate, Server Hello Done
17 2.940026	127.0.0.1	127.0.0.1	SSLv3	337	Client Key Exchange, Change Cipher Spec, Finished
18 2.941486	127.0.0.1	127.0.0.1	SSLv3	172	Change Cipher Spec, Finished
19 2.944625	127.0.0.1	127.0.0.1	HTTP	5756	HTTP/1.1 200 OK (text/html)
20 2.944864	127.0.0.1	127.0.0.1	TCP	66	38713 → 443 [ACK] Seq=1143 Ack=7845 Win=32767 Len=0 TSval=525565080 TSecr=525565080
21 2.964424	127.0.0.1	127.0.0.1	HTTP	471	GET /icons/jhe061.png HTTP/1.1
22 2.964572	127.0.0.1	127.0.0.1	TCP	74	38714 → 443 [SYN] Seq=0 Win=32767 Len=0 MSS=16384 SACK_PERM=1 TSval=525565088 TSecr=0 WS=1
23 2.964588	127.0.0.1	127.0.0.1	TCP	74	443 → 38714 [SYN, ACK] Seq=0 Ack=1 Win=32767 Len=0 MSS=16384 SACK_PERM=1 TSval=525565088 TSecr=525565088 WS=1
24 2.964598	127.0.0.1	127.0.0.1	TCP	66	38714 → 443 [ACK] Seq=1 Ack=1 Win=32767 Len=0 TSval=525565088 TSecr=525565088
25 2.964818	127.0.0.1	127.0.0.1	SSLv3	188	Client Hello
26 2.964819	127.0.0.1	127.0.0.1	TCP	66	443 → 38714 [ACK] Seq=1 Ack=121 Win=32767 Len=0 TSval=525565088 TSecr=525565088
27 2.992274	127.0.0.1	127.0.0.1	SSLv3	220	Server Hello, Change Cipher Spec, Finished
28 2.992312	127.0.0.1	127.0.0.1	TCP	66	38714 → 443 [ACK] Seq=121 Ack=155 Win=32767 Len=0 TSval=525565108 TSecr=525565108
29 2.992855	127.0.0.1	127.0.0.1	HTTP	562	GET /icons/debian/openlogo-25.jpg HTTP/1.1

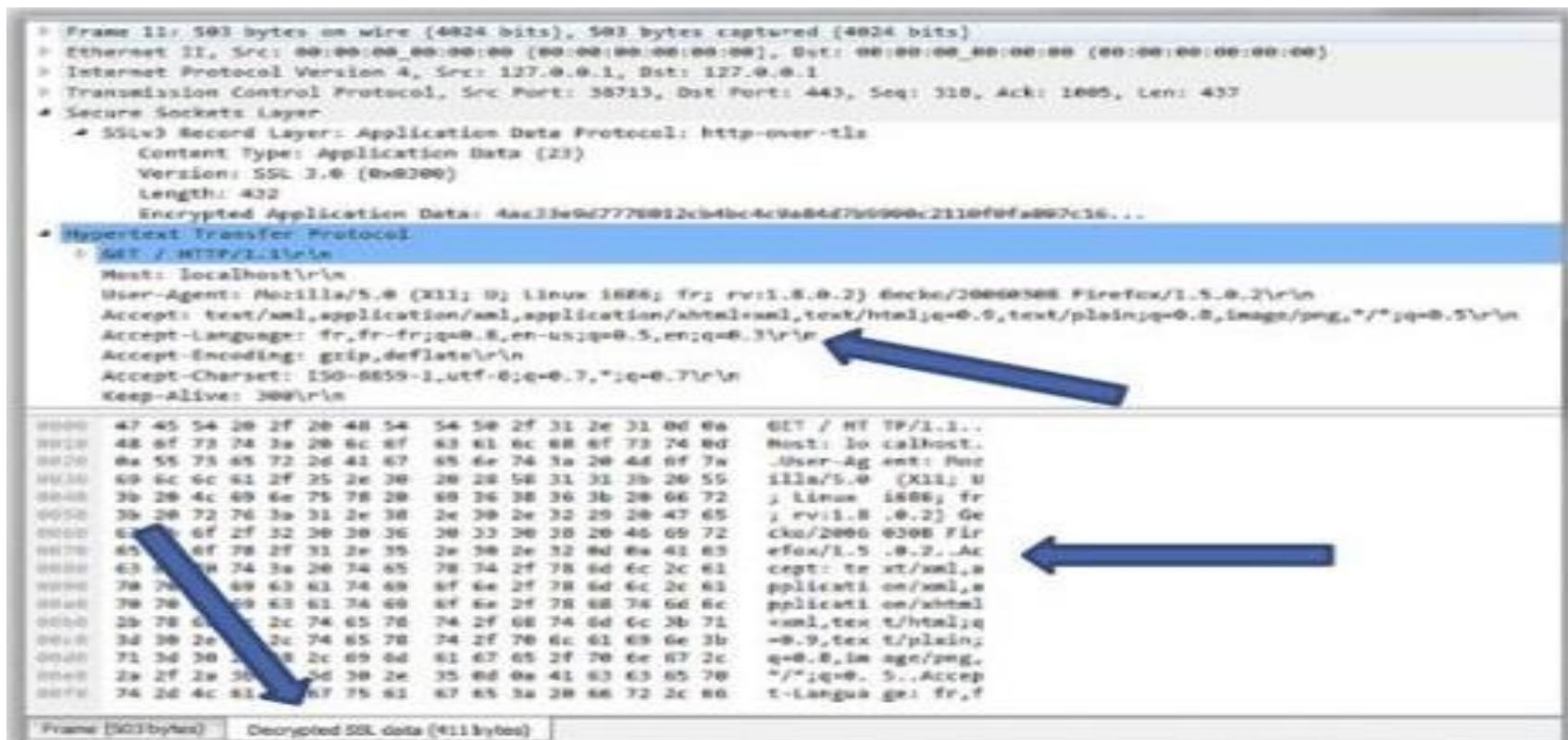
تعالي ندخل جوا ال **packet** ونشوف الدنيا ماشيء ازاي اكتر

```
> Frame 11: 583 bytes on wire (4624 bits), 583 bytes captured (4624 bits)
> Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 38713, Dst Port: 443, Seq: 318, Ack: 1005, Len: 432
# Secure Sockets Layer
# SSLv3 Record Layer: Application Data Protocol: http-over-tls
Content-Type: Application Data (23)
Version: SSL 3.0 (0x0300)
Length: 432
Encrypted Application Data: 4ec33e9d7778012cb4bc4c9a8a4d7b1998c2110fefa007c16...
# Hypertext Transfer Protocol
# HTTP / 1.1 [HTTP/1.1]
Host: localhost\r\n
User-Agent: Mozilla/5.0 (X11; U; Linux; fr; rv:1.8.0.2) Gecko/20060308 Firefox/1.5.0.2\r\n
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5\r\n
Accept-Language: fr,fr-fr;q=0.8,en-us;q=0.5,en;q=0.3\r\n
Accept-Encoding: gzip,deflate\r\n
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
Keep-Alive: 3000\r\n
0000  00 00 00 00 00 00 00 00 00 00 00 00 00 45 00  ...
0001  01 e9 49 72 48 00 48 00 71 0a 77 00 00 01 7f 00  .Ig@.0.
0002  00 01 97 39 01 bb 28 8c 3a d4 78 c5 28 c5 09 18  ..B..X. :.X.{...
0003  2f ff ff dd 00 00 01 01 00 00 53 7c 14 1f 53  .....S|..S
0004  7c 0a 17 03 00 01 b8 4a c3 3e 9d 77 78 01 2c b4  |...|...>.HX...
0005  bc 4c 9a 84 d7 b9 98 8c 21 10 19 fa 89 7c 16 bb  .L.....1....]
0006  77 fb 72 42 4f ad 58 4a d9 aa 0f aa 44 6c 62 94  W.FBO.P2 ..o.03b.
0007  1b c5 e9 1c 5e de 85 00 0c 05 e4 18 5e d2 03  ....^....n
0008  05 20 ab 81 fd 18 9a 75 0d d7 ef c3 dd 74 d7 9c  .....,5...9...
0009  3e 6f 21 fd 24 ca 3c 78 78 36 12 7a 9c ac 01 m...< p06.z...
000a  4c 1c a8 fb 27 30 ba 9a f4 2f 0a ab 80 6a a1 68  R...@. /...j...
000b  74 f8 e3 91 84 e7 98 88 cc f8 95 7b 8a 22 f2 f9  t.....{...
000c  27 e8 dd 38 9c fd e9 83 71 dc 28 a4 ee df e3 72  ..H... q.p.m.r...
000d  9e a1 f9 c9 98 d6 02 55 6a 67 5d 9c b8 75  .....U jg]..4...
000e  01 9f e6 d2 44 18 bc ca 7a 10 39 af 75  .....D... z.9.....
000f  55 d4 b3 8d 93 23 84 99 3a 00 00 00 4c bf 3f 5...m...R... ..L.P
```

Frame (583 bytes) | Decrypted SSL data (411 bytes)

هتبص على **packet 11** هتلاقي ان فعلا بدء **HTTP traffic** ال **clear text** يظهر عندك و هتلاقي ال **TLS** تشفير

- و هتلاقی عذر تخت tab ظهرت جدیده اسمها SSL Decrypted دوس علیها هتلاقی برضه ال data بقت ظاهره و تقدر شوفها Data



- تعالی نبص عال HTTPS لل Suspicious Traffic

هنشوف مثال على **CrypMic** ransomware اسمه ودا كان بيصيّب جهاز ال **Victim** عن طريق port 443 بتابع ال **HTTPS** وبيعمل عنده ال **C&C** servers هو **command and control** بيرفعه عند ال **victim** ويتحكم فيه عن طريقه ودا هنشوفه من خلال الامثله الجايه وهنعرف ازاى ممكن نقدر نحلله .

- هنفترض اننا معانا ال **HTTPs** المصاّب بال **capture** بتاع ال **statistics** **wire shark** فان **ransomware** ومنها على **هتلaciّه** جايبلوك اكتر البروتوكولات على **network** ال تم استخدامها عندك فال

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	878	100.0	710011	23 k	0	0	0
Ethernet	100.0	878	1.7	12292	413	0	0	0
Internet Protocol Version 4	100.0	878	2.5	17560	590	0	0	0
Transmission Control Protocol	100.0	878	95.6	678479	22 k	676	429856	14 k
Secure Sockets Layer	4.0	35	5.0	35786	1203	35	35786	1203
Hypertext Transfer Protocol	1.3	11	12.2	86590	2912	6	2200	73
Media Type	0.1	1	10.9	77493	2606	1	77675	2612
Line-based text data	0.5	4	1.5	10709	360	4	6562	220
Data	17.8	156	28.8	204260	6869	156	204260	6869

- بصينا ع تحليل البروتوكولات عندي فالشبكة هتلaci اکتر traffic
 - عندك هو ال TCP يعني معناه انه فيه request و response رايم
 - جاي عندك فالشبكة وهتلaci برضه ال HTTPS ال هو ال SSL فيه
 - ليه برضه تمام كدا
- هروح بعد كدا ل packet نفس ال statistics ال انا فيها واروح
 - واختار منها End point عشان استخرج كل ال End points
 - traffic من ال capture من ال Traffic IP
 - وال المربوطه بيها زى مهنشوف فالامثله country

Ethernet	IPv4	IPv6	TCP	UDP	Port	Packets	Bytes	Packets A + B	Bytes A + B	Packets B + A	Bytes B + A	Country	AS Number	City	Latitude	Longitude
51.254.30.225	80	532	445 k		326	431 k	206		14 k	France	AS16276 OVH SAS	—	—	48.858200	2.338700	
83.217.27.178	80	24	2184		7	839	17		1345	Russian Federation	AS200161 DATAPRO Limited Liability Company	—	—	55.738602	37.606800	
85.14.243.0	443	311	257 k		200	250 k	111		6792	Germany	AS24951 myLoc managed IT AG	—	—	51.299301	9.490900	
89.36.89.80	80	11	4558		5	3861	6		697	Romania	AS43938 SC Gateway Telecom SRL	Bucharest, 10	44.700001	26.450001		
192.168.4.195	49284	11	4558		6	697	5		3861	—	—	—	—	—	—	
192.168.4.195	49288	6	366		4	246	7		120	—	—	—	—	—	—	
192.168.4.196	49289	18	1818		13	1099	5		719	—	—	—	—	—	—	
192.168.4.196	49293	126	88 k		58	4721	68		84 k	—	—	—	—	—	—	
192.168.4.196	49294	12	726		10	606	2		120	—	—	—	—	—	—	
192.168.4.196	49301	394	356 k		138	8799	256		347 k	—	—	—	—	—	—	
192.168.4.196	49310	8	526		5	352	3		174	—	—	—	—	—	—	
192.168.4.196	49311	13	2441		7	438	6		2003	—	—	—	—	—	—	
192.168.4.196	49312	273	253 k		89	5358	184		248 k	—	—	—	—	—	—	
192.168.4.196	49313	9	600		5	320	4		280	—	—	—	—	—	—	
192.168.4.196	49314	8	498		5	324	3		174	—	—	—	—	—	—	

- لو بصيت هنا فالاول هتلaci 3 عناوين IP رايحين ل 3 دول مختلفه
 - احنا هنركز علي ال رايم port 443 عشان دا ال عليه HTTPS

- هنرکز عال IP ال هو رایح Germany ال هو 85.14 هنبص نلاقي حجم ال packets ال طالع من ال source رایح لـ server الموجود في suspicious فبده نشك ان دا ما = 6792 كبير جدا في Germany Traffic .

- تعالى نكمـل statistics هتروج بعد كـدا لـ investigation و بعد كـدل نروح لـ conversations و عـاوزـين نـبـصـ مـيـنـ عنـدـنـاـ فالـشـبـكـهـ الدـاخـلـيهـ بيـتواـصلـ معـ الـ IPـ بـتـاعـ Germanyـ قـولـنـاـ عـلـيهـ Suspiciousـ .

Ethernet-1	IPv4-4	IPv6	TCP-11	UDP	Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
192.168.4.196	49284	89.36.89.80	80	11	4558	6	697	5	3861	0.000000	1.9715	2828	15 k					
192.168.4.196	49288	83.217.27.178	80	6	366	4	246	2	120	1.873187	174.1499	11	5					
192.168.4.196	49289	83.217.27.178	80	18	1818	13	1099	5	719	1.873282	235.9954	37	24					
192.168.4.196	49293	51.254.30.225	80	126	88 k	58	4721	68	84 k	3.462278	221.1622	170	3040					
192.168.4.196	49294	51.254.30.225	80	12	726	10	606	2	120	3.462375	232.5032	20	4					
192.168.4.196	49301	51.254.30.225	80	394	356 k	138	8799	256	347 k	7.199412	2.5818	27 k	1076 k					
192.168.4.196	49310	85.14.243.9	443	8	526	5	352	3	174	10.027150	2.1311	1321	653					
192.168.4.196	49311	85.14.243.9	443	13	2441	7	438	6	2003	11.845196	1.8945	1849	8457					
192.168.4.196	49312	85.14.243.9	443	273	253 k	89	5358	184	246 k	13.038160	2.5466	16 k	779 k					
192.168.4.196	49313	85.14.243.9	443	9	600	5	320	4	280	16.852456	1.3813	1853	1621					
192.168.4.196	49314	85.14.243.9	443	8	498	5	324	3	174	84.801036	1.6513	1569	842					

- هنبص نلاقي ال IP ال هو 192.168.4.196 ال طالع من عندي من الشـبـكـهـ الدـاخـلـيهـ هو ال باعـتـ trafficـ لـ IPـ الـ destinationـ وهو بـتـاعـ Germanyـ وهـتـلـاقـيـ نفسـ الـ IPـ الدـاخـلـيـ دـاـ فـاتـحـ كـذـاـ Sessionـ عـلـىـ نفسـ الـ portـ الـ 443ـ وـعـدـ الـ.....ـ هـتـلـاقـيـهـ بـيـزـيـدـ.....ـ Trafficـ كـلـ Packetـ .

- يـبقـاـ اـنـاـ اـلـ اوـلـ كـ Threat hunterـ روـحـتـ شـكـيـتـ فالـ public IPـ وبعدـ كـداـ روـحـتـ شـوـفـتـهـ بـيـكـلمـ مـيـنـ عنـدـيـ دـاـخـلـ الـ Networkـ وـطـلـعـتـ الـ Ipـ الدـاخـلـيـ الـ كانـ بـيـكـلمـهـ....ـ وـتـعـالـيـ نـكـمـلـ معـ investigationـ .

ip.addr==85.14.243.9 & & tcp.port==443						
No.	Time	Source	Destination	Protocol	Length	Info
538	19.027150	192.168.4. 85.14.243.	TCP	66.49310 -> 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1464 WS=4 SACK_PERM=1		
539	19.386978	85.14.243.	192.168.4.	TCP	66.443 -> 49310 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1310 WS=256 SACK_PERM=1	
540	19.387128	192.168.4. 85.14.243.	TCP	66.49310 -> 443 [ACK] Seq=1 Ack=1 Win=65900 Len=0		
541	19.142660	192.168.4. 85.14.243.	SSL	106 Continuation Data		
542	19.268880	85.14.243.	192.168.4.	TCP	54.443 -> 49310 [FIN, ACK] Seq=1 Ack=53 Win=65792 Len=0	
543	19.258995	192.168.4. 85.14.243.	TCP	60.49310 -> 443 [ACK] Seq=53 Ack=2 Win=65908 Len=0		
544	19.044058	192.168.4. 85.14.243.	TCP	60.49310 -> 443 [FIN, ACK] Seq=53 Ack=2 Win=65900 Len=0		
545	19.845190	192.168.4. 85.14.243.	TCP	66.49311 -> 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1464 WS=4 SACK_PERM=1		
546	19.051212	85.14.243.	192.168.4.	TCP	66.443 -> 49311 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1310 WS=256 SACK_PERM=1	
547	19.051360	192.168.4. 85.14.243.	TCP	60.49311 -> 443 [ACK] Seq=1 Ack=1 Win=65908 Len=0		
548	19.158204	85.14.243.	192.168.4.	TCP	54.443 -> 49310 [ACK] Seq=2 Ack=54 Win=65792 Len=0	
549	19.811870	192.168.4. 85.14.243.	SSL	72 Continuation Data		
550	19.025054	85.14.243.	192.168.4.	SSL	62 Continuation Data	
551	19.037337	85.14.243.	192.168.4.	TCP	1372.443 -> 49311 [ACK] Seq=9 Ack=19 Win=65792 Len=1318	
552	19.037410	85.14.243.	192.168.4.	TCP	395.443 -> 49311 [PSH, ACK] Seq=1327 Ack=19 Win=65792 Len=341	
553	19.037471	85.14.243.	192.168.4.	TCP	54.443 -> 49311 [FIN, ACK] Seq=1668 Ack=19 Win=65792 Len=0	
554	19.037485	192.168.4. 85.14.243.	TCP	60.49311 -> 443 [ACK] Seq=19 Ack=1668 Win=65900 Len=0		
555	19.037585	192.168.4. 85.14.243.	TCP	60.49311 -> 442 [FIN, ACK] Seq=19 Ack=1668 Win=65900 Len=0		
556	19.037760	192.168.4. 85.14.243.	TCP	60.49311 -> 443 [ACK] Seq=20 Ack=1669 Win=65900 Len=0		
557	19.038160	192.168.4. 85.14.243.	TCP	66.49312 -> 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1464 WS=4 SACK_PERM=1		
558	19.297291	85.14.243.	192.168.4.	TCP	66.443 -> 49312 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1310 WS=256 SACK_PERM=1	
559	19.297442	192.168.4. 85.14.243.	TCP	60.49312 -> 443 [ACK] Seq=1 Ack=1 Win=65900 Len=0		
560	19.023923	192.168.4. 85.14.243.	TCP	60. [TCP Spurious Retransmission] 49311 -> 443 [FIN, ACK] Seq=19 Ack=1669 Win=65900 Len=0		
561	19.739739	85.14.243.	192.168.4.	TCP	54. [TCP ZeroWindow] 443 -> 49311 [ACK] Seq=1669 Ack=20 Win=8 Len=0	
562	19.059827	192.168.4. 85.14.243.	SSL	72 Continuation Data		
563	19.180189	85.14.243.	192.168.4.	SSL	62 Continuation Data	
564	19.180897	85.14.243.	192.168.4.	TCP	1372.443 -> 49312 [ACK] Seq=0 Ack=19 Win=65792 Len=1318	

- كمان لو روحنا بصينا على ال filter wire shark traffic وعملنا باسم ال IP الخارجي ال هو في port Germany بال بيروح Wire shark ال هو 443 هنلاقي ال HTTPS Traffic مطلعى ال packet رقم 560 and 561 رقم packet فيه error فال !!!

- هنلاقي ال TCP Spurious error بتاعها ال هو TCP Retransmission بمعنى ان اما بتيجي تعمل اتصال بال SYN و بعد كدا بيحصل ال SYN من source فهنلاقي هنا ان ال Server destination بعث رساله لل destination ومرضش عليه يسكت !! هنلاقيه عمال يبعث تاني ف رسائل لل destination ف ساعتها ال wire shark بتصنف ان دا مشكله وبيفهم ان ال destination دا مش عاوز يرد

عليك ودا بيحصل فحاله ان ال **destination** بيكون مش عاوز يرد على اتصالك ياخذ منه الاتصال فقط وبعدين ميرضش عليك.

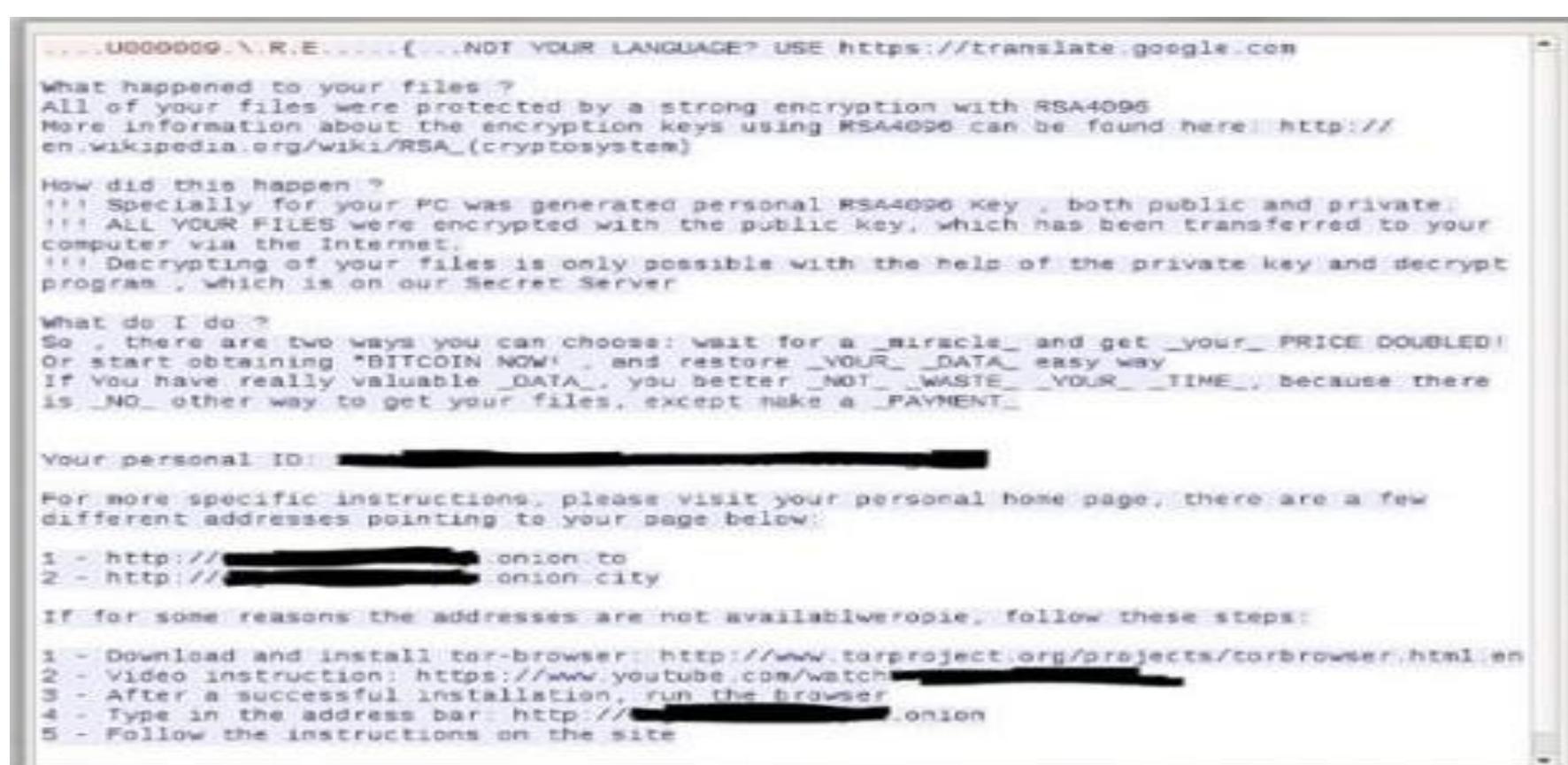


- دا هتلاقیه ال **command and C&C** ال **behavior** بتاع ال **control server** بمعنى ال **destination** بس ال يبعتلك انما اما تيجي انت تبعتله هتلاقیه مبيرضش عليك او يرض عليك بحاجه اسمها **Finish ACK**.
- هنبعص فال **packet** ال بعديها ال هي **561** هنلاقیه عاطيني **error** اسمه **TCP Zero Window** دي **packet** ودا هتلاقیه فحاله ان ال **request** واحد عمال يبعتلك **header** **payload** مش فال **packet** لان ال **header** **payload** و **Trailer** عندنا مكونه من ... **header** **payload** و **Trailer** ال بيكون فيها العناوين ال هي ال **IP** وال **payload** بيكون فيها ال **content** وال **Destination** وال **source** ال **header** **payload** **Trailer** دا بيكون فيه ال **error** وبيكون نهاية ال **packet**.



- فدا برضه معناه ان ال **C&C** هو ال بيعتها لان ال **C&C** مش بيملى
ال **wire shark** بيسها فاضيه ... فعلظول تلاقي ال **payload** بيديك
network **packet** عندها بتتبع فال **alert** ويقولك الحق فيه
فاضيه طب هو مين ال هيكلم حد فال **Network** وبيعتله **packet**
فاضيه !!!

- طيب لو احنا جينا عال **Right** **packet** رقم **561** وعملنا عليها **Click**
واختارنا منها **follow TCP Stream** اول حاجه هتشوفها ان
ال **content** دي فاضيه وكمان هتلaci ان ال **packet** ال
جواها كمان **traffic** !!! هو مش المفروض **clear Text** ال
يكون **encrypted HTTPS** مشفر طبيفي لانه بيستخدم ال
عشان يشفر المحادثه ما بين الطرفين . **SSL/TLS**



- ودي الرساله الطبيعيه ال بيرسلها لك ال ransomware مصاب بيها ال هي بتظهر لك user بالشكل دا



- وبتجيلك من ال C&C Servers اما تيجي تعمل live files بشكل ما يجي على اي حاجه على جهازك زي ال files يقوم مطلع لك الرساله دی

- فاحنا باستخدام ال wire shark فالشرح ال فات قدرنا نوصل للرساله user ال بطلع لل

- طبعاً دا مش كل ال HTTP & attacks ال بتحصل على ال HTTPS احنا ذكرنا منهم ال Common وال هتشوفه كتير اثناء عملك port 80 & 443 وتركز اهم شيء على threat hunter الخاصين بال attack عشان اي HTTP & HTTPS بالبروكولين دول هتلaciهم جايناك على البورات دا .

Unknown Traffic:

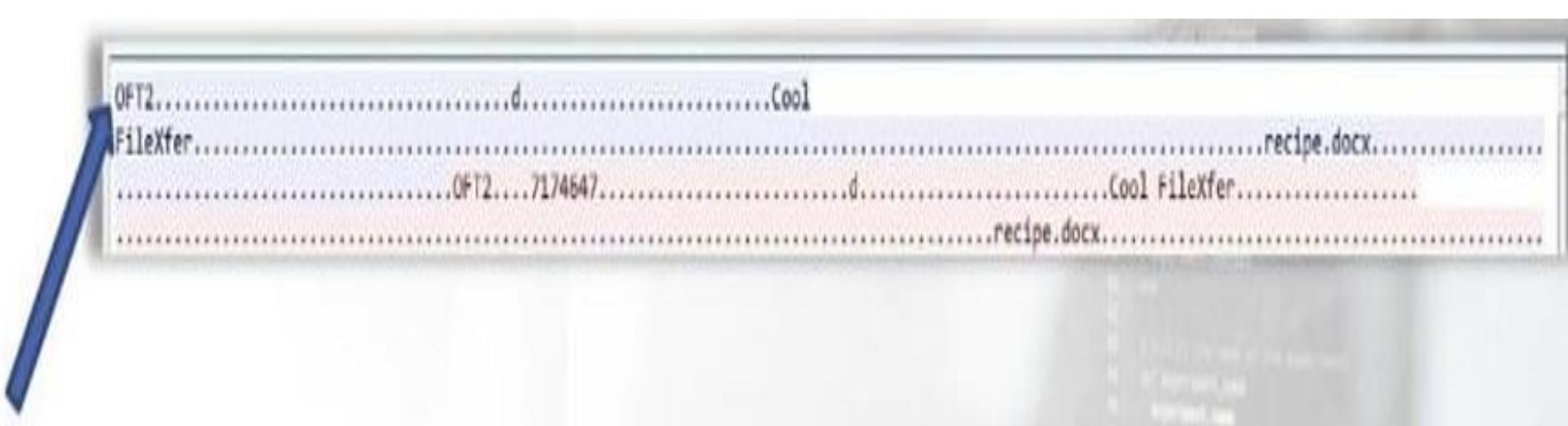
- ممکن تلاقي traffic زی کنا ذکرناه فالشرح ال فات بتاع ال traffic بیبقا جی عبی port 443 انت عارف انه بتاع ال HTTPS بس جی یعمل عليه حاجه تانیه زی ال C&C یبنالک ان ال traffic عادي لکن هو غیر کدا ... و هتلافي امثاله کتیر على کدا .
- المثال ال معانا عندنا SSL pcap file فيه برتوکول ال عشان ننقل من خلله instant messenger traffic والمفروض ان ال messages ملھوش دعوه بال HTTPS traffic ال جایه من ينقل ال yahoo او غيره فدا هنشوفه مع بعض ان شاء الله
- بعد اما عملنا wire shark filter لل 443 عشان نشتغل على نضافه ونطلع traffic ال TCP المعتمد عليه HTTPS لاقینا الاتي فال capture ال خدناها....

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.158	64.12.24.50	SSL	60	Continuation Data
2	0.000579	64.12.24.50	192.168.1.158	TCP	60	443 -> 51128 [ACK] Seq=1 Ack=7 Win=64240 Len=0
3	15.044068	192.168.1.158	64.12.24.50	SSL	243	Continuation Data
4	15.044588	64.12.24.50	192.168.1.158	TCP	60	443 -> 51128 [ACK] Seq=1 Ack=196 Win=64240 Len=0
5	15.135701	192.168.1.158	64.12.24.50	SSL	94	Continuation Data
6	15.135706	64.12.24.50	192.168.1.158	TCP	60	443 -> 51128 [ACK] Seq=1 Ack=236 Win=64240 Len=0
7	15.152349	64.12.24.50	192.168.1.158	SSL	263	Continuation Data
8	15.154639	64.12.24.50	192.168.1.158	SSL	92	Continuation Data
9	15.155906	192.168.1.158	64.12.24.50	TCP	60	51128 -> 443 [ACK] Seq=236 Ack=210 Win=62780 Len=0
10	15.155911	192.168.1.158	64.12.24.50	TCP	60	51128 -> 443 [ACK] Seq=236 Ack=248 Win=62742 Len=0
11	39.587870	192.168.1.158	64.12.24.50	SSL	182	Continuation Data
12	39.590958	64.12.24.50	192.168.1.158	TCP	60	443 -> 51128 [ACK] Seq=248 Ack=364 Win=64240 Len=0
13	39.697907	64.12.24.50	192.168.1.158	SSL	263	Continuation Data
14	39.698818	192.168.1.158	64.12.24.50	TCP	60	51128 -> 443 [ACK] Seq=364 Ack=457 Win=62742 Len=0
15	39.700370	64.12.24.50	192.168.1.158	SSL	92	Continuation Data
16	39.700370	192.168.1.158	64.12.24.50	TCP	60	51128 -> 443 [ACK] Seq=457 Ack=546 Win=62742 Len=0

Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
Ethernet II, Src: HewlettP_45:a4:bb (00:12:79:45:a4:bb), Dst: VMware_00:0d:62 (00:0c:29:00:0d:62)
Internet Protocol Version 4, Src: 192.168.1.158, Dst: 64.12.24.50
Transmission Control Protocol, Src Port: 51128, Dst Port: 443, Seq: 1, Ack: 1, Len: 6
Secure Sockets Layer

- وکمان هتلaci ان ال **Encrypted traffic** مش مشفر مش **traffic** ودي
کمان غریبه طب ازاي وهو مستخدم ال **SSH** دا اذا كان دا معمول عشان
یخلي ال **HTTPs** اساسا **Encryption** عن طريق ال **secure** ؟!!!
فکدا اتاكدنا ان دا مش **SSL Traffic** وان دا **fake normal**

- لو روحنا لل **TCP Stream** من ال **wire shark** لرقم 5 وال
برتوكول **session** هي ال المفتوحة بيني وبين الجهاز الآخر فهناقي
برتوكول ثاني مستخدم لنقل الملفات وهو ال **OFT2** ودا برتوکول
بیستخدم داخل برنامج اسمه **AOL message** برنامج زي **Yahoo** کدا
بنپعت پیه رسایل او بنسسل پیه ملفات

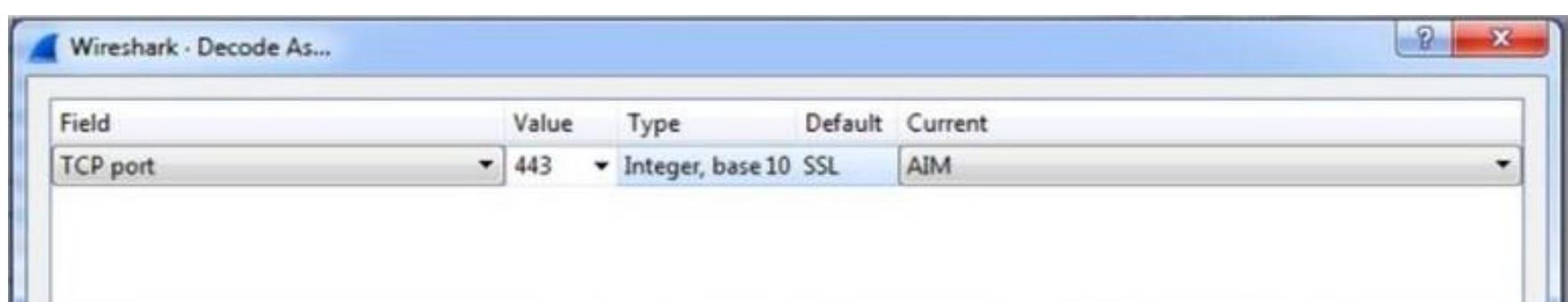


- طب فال wire shark لـ decode عندنا ميزة انه بيقدر يعمل لـ packet traffic ال انا شاكـ ويتاكدـ لي من محتوياتها وهـ لـ فعلاـ الـ Traffic وـ هو suspicious ولا ايـه النـظام !! وـ دا شـكل الـ Traffic فيه دـا طـلع

باین انه **decode** قبل ال منعمله عشان نبقا نعرف الفرق بعدين .

Before						
No.	Time	Source	Destination	Protocol	Length	Info
135	61.286572	64.12.25.91	192.168.1.159	TLSv1	284	Application Data
136	61.287474	64.12.24.50	192.168.1.158	SSL	183	Continuation Data
159	67.283887	192.168.1.159	64.12.25.91	TLSv1	115	Application Data
161	67.395187	64.12.24.50	192.168.1.158	SSL	94	Continuation Data
164	69.467940	192.168.1.159	64.12.25.91	TLSv1	236	Application Data
166	69.578661	64.12.25.91	192.168.1.159	TLSv1	284	Application Data
167	69.578667	64.12.24.50	192.168.1.158	SSL	280	Continuation Data
177	77.713532	192.168.1.159	64.12.25.91	TLSv1	115	Application Data
179	77.834511	64.12.24.50	192.168.1.158	SSL	94	Continuation Data
181	83.374949	192.168.1.159	64.12.25.91	TLSv1	254	Application Data
183	83.487629	64.12.25.91	192.168.1.159	TLSv1	284	Application Data
184	83.489593	64.12.24.50	192.168.1.158	SSL	298	Continuation Data
188	83.793749	192.168.1.159	64.12.25.91	TLSv1	115	Application Data
190	83.906524	64.12.24.50	192.168.1.158	SSL	94	Continuation Data
194	84.883656	192.168.1.159	64.12.25.91	TLSv1	115	Application Data
196	84.995484	64.12.24.50	192.168.1.158	SSL	94	Continuation Data
200	87.726166	192.168.1.158	64.12.24.50	SSL	94	Continuation Data
202	87.832875	64.12.25.91	192.168.1.159	TLSv1	115	Application Data
210	90.788876	192.168.1.158	64.12.24.50	SSL	64	Continuation Data

- هتجي على اي **packet SSL** وتقف عليها وتضغط **right Click** وتحتار منها **decode as** هنا بتبدء تدي لـ **wire shark** معطياته ال هيقوم بالعمل على اساسها عشان يعملك تshireح لـ **packet** دي ويعرفك على محتوياتها....



- هتجي عند ال **port** وتحتار ال **TCP** وبعدين تروح لـ **dissector** ال هو اداه التshireح لـ **packet** ال هيستخدمها ال هو بروتوكول اخر عشان تتأكد فانت كدا بتقول لـ **wire shark** لـ **AIM** **TCP Traffic** لـ **decoder** بتابعها ان يحولك

- تعالى نشوف ال traffic بعد اما استخدمنا ال dissector

After					
No.	Time	Source	Destination	Protocol	Length Info
51	34.761557	205.188.13.12	192.168.1.159	AIM Generic	831 AIM Generic, Rate Info
52	34.763039	192.168.1.159	205.188.13.12	AIM Generic	88 AIM Generic, Rate Info Ack
54	34.768974	192.168.1.159	205.188.13.12	AIM Generic	86 AIM Generic, Client Ready
56	34.770551	192.168.1.159	205.188.13.12	AIM SST	81 SNAC data, AIM SST, Subtype: 0x0008
63	35.313582	205.188.13.12	192.168.1.159	AIM SST	1395 SNAC data, AIM SST, Subtype: 0x0007
92	58.458766	192.168.1.158	64.12.24.50	AIM Messaging	182 AIM Messaging, Outgoing to: Sec558user1
94	58.568705	64.12.24.50	192.168.1.158	AIM Generic	263 AIM Generic, Rate Change
97	58.571268	64.12.24.50	192.168.1.158	AIM Messaging	92 AIM Messaging, Acknowledge
136	61.287474	64.12.24.50	192.168.1.158	AIM Messaging	183 AIM Messaging, Incoming
161	67.395187	64.12.24.50	192.168.1.158	AIM Messaging	94 AIM Messaging, Mini Typing Notifications (MTN)
167	69.578667	64.12.24.50	192.168.1.158	AIM Messaging	286 AIM Messaging, Incoming
179	77.834511	64.12.24.50	192.168.1.158	AIM Messaging	94 AIM Messaging, Mini Typing Notifications (MTN)
184	83.489593	64.12.24.50	192.168.1.158	AIM Messaging	298 AIM Messaging, Incoming
198	83.906524	64.12.24.50	192.168.1.158	AIM Messaging	94 AIM Messaging, Mini Typing Notifications (MTN)
196	84.995484	64.12.24.50	192.168.1.158	AIM Messaging	94 AIM Messaging, Mini Typing Notifications (MTN)
200	87.726166	192.168.1.158	64.12.24.50	AIM Messaging	94 AIM Messaging, Mini Typing Notifications (MTN)
212	98.816866	192.168.1.158	64.12.24.50	AIM Messaging	164 AIM Messaging, Outgoing to: Sec558user1AIM Messaging, Mini Typing Notifications (MTN)
214	91.003314	64.12.24.50	192.168.1.158	AIM Generic	263 AIM Generic, Rate Change
216	91.004650	64.12.24.50	192.168.1.158	AIM Messaging	92 AIM Messaging, Acknowledge

- بدء هنا يستبدلك ال SSL Protocol ال هو فعلا كان fakeProtocol و يستبدلك مكانه بال AIMProtocol فكدا فعلا ال packet و تعالى نشوف شكل ال suspicious traffic بقى عامل ازاي ؟

```

Frame 167: 280 bytes on wire (2240 bits), 280 bytes captured (2240 bits)
Ethernet II, Src: Vmware_b0:8d:62 (00:0c:29:b0:8d:62), Dst: HewlettP_45:a4:bb (00:12:79:45:a4:bb)
Internet Protocol Version 4, Src: 64.12.24.50, Dst: 192.168.1.158
Transmission Control Protocol, Src Port: 443, Dst Port: 51128, Seq: 664, Ack: 364, Len: 226
# AOL Instant Messenger
    Command Start: 0x2a
    Channel ID: SNAC Data (0x02)
    Sequence Number: 22234
    Data Field Length: 220
        ▶ FNAC: Family: AIM Messaging (0x0004), Subtype: Incoming (0x0007)
    # AIM Messaging, Incoming
        ICBM Cookie: d41b4ece87df1f77
        Message Channel ID: 0x0001
        ▶ Buddy: Sec558user1
        Warning Level: 0
        TLV Count: 5
        ▶ TLV: User class
        ▶ TLV: Member since
        ▶ TLV: BART Info
        ▶ TLV: Session Length (sec)
        ▶ TLV: Online since
        ▶ TLV: Message Block
        ▶ TLV: Unknown
        ▶ TLV: Non-direct connect typing notification
        ▶ TLV: Unknown

```

- فعلا هنلاقي ظهر ال protocol الحقيقي ال استخدمناه ال attacker فالارسال ال هو AIM و بدء ال wire shark يظهر لك محتوياته .

- وفي النهاية احنا كل ال اتكلمنا عليه باستخدام ال **wire shark**

انما احنا عندنا ادوات تانية بنعمل بيها زي ال **network hunting** زي ال **passive packets** ودي بتشتغل عال **miner** وتدليه **live capture** وهو يفكها لك مش بيعملك زي ال **packet**

wire shark

- وعندنا ادوات تانية زي ال **RSA NetworkMiner investigator** برضه زيها زي ال **capture** بتحلله وتدليه **network miner** وتحللها وتدليه **details** بتاعته بالتفاصيل وتقدر تتعقب فال **tools** دي وتعلم عنها اكتر وربنا يوفقنا جميعا.

