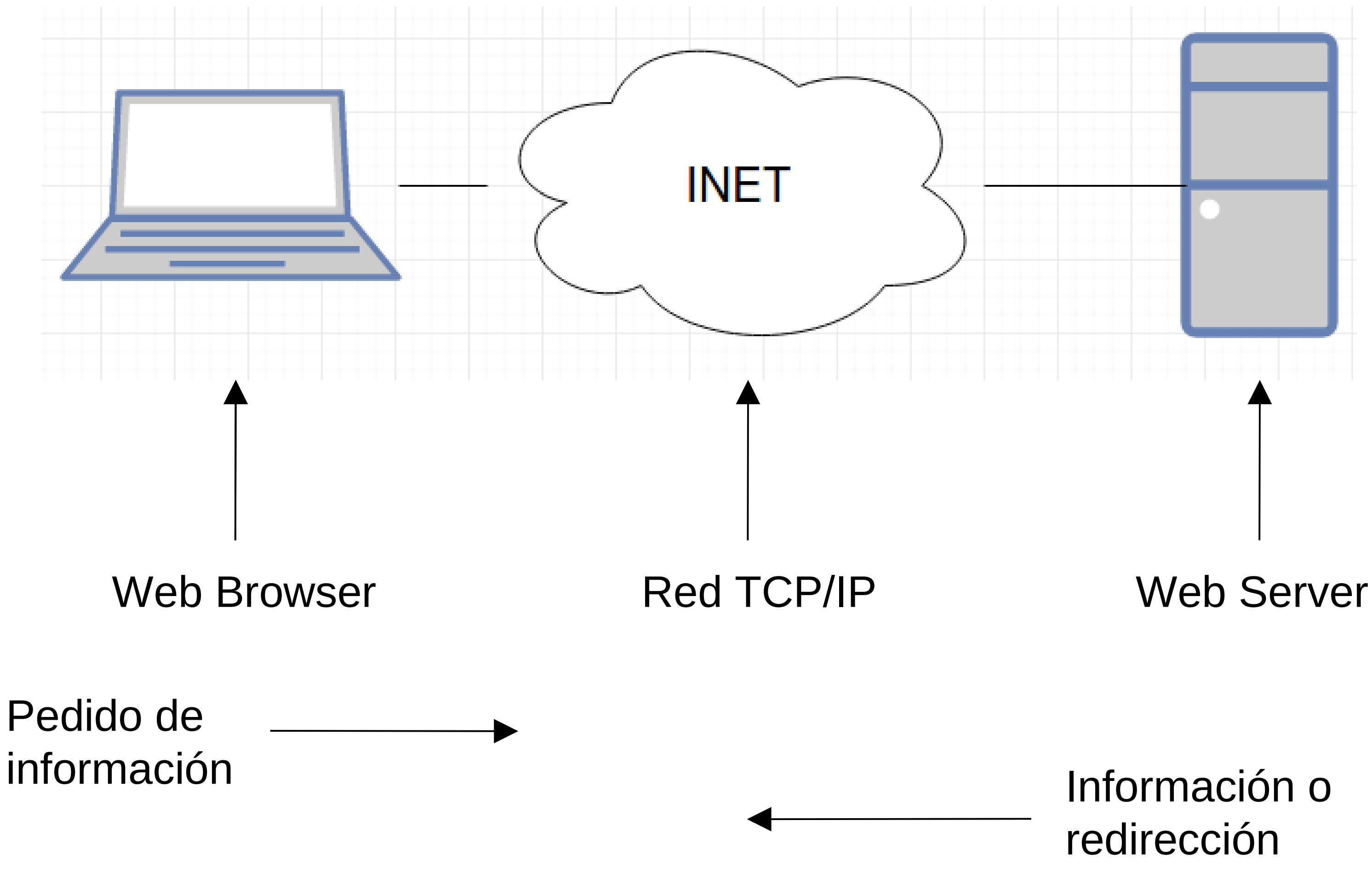
# 6669 Criptografía y Seguridad Informática

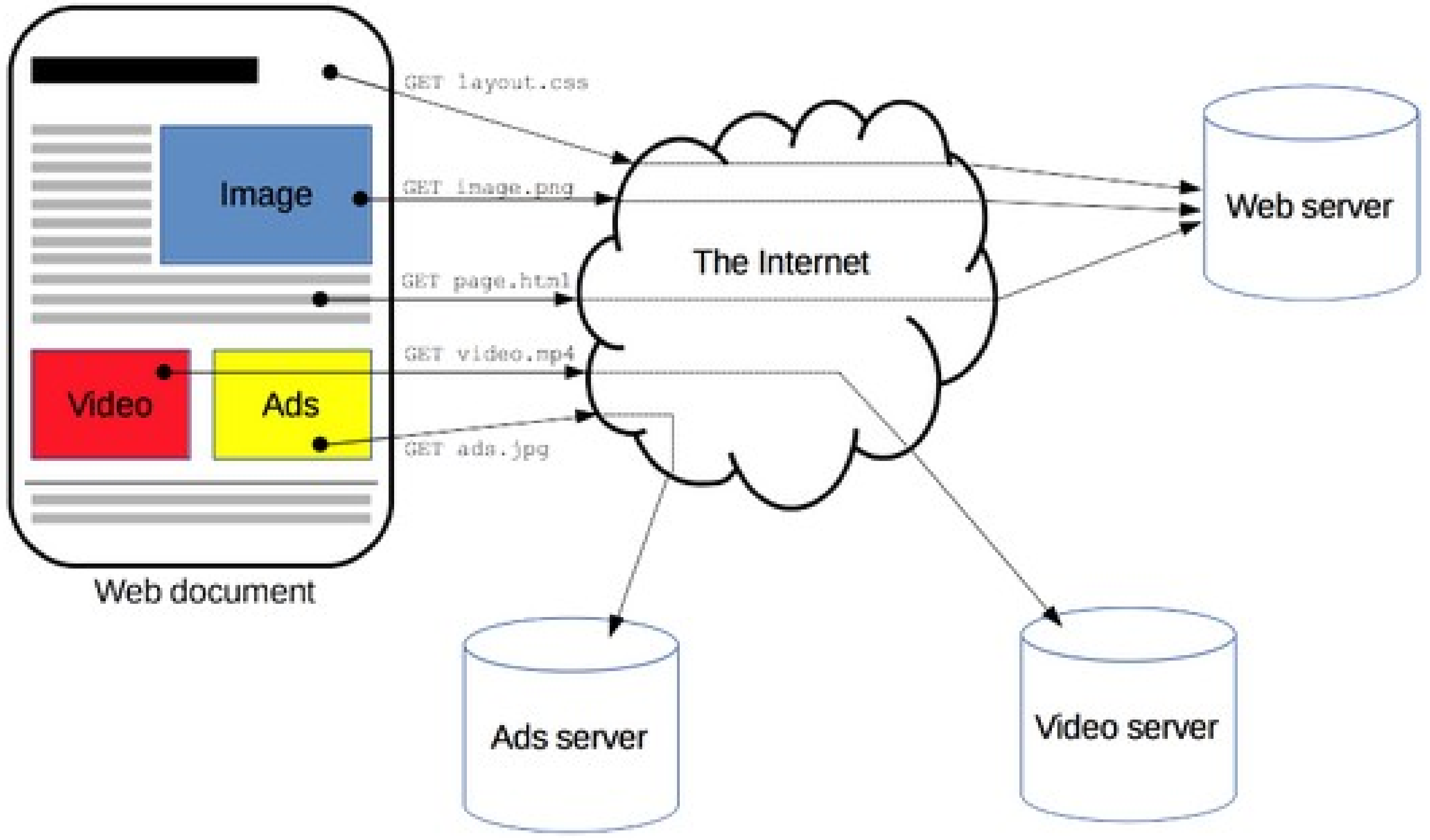
## Seguridad en Aplicaciones Web

# HTTP

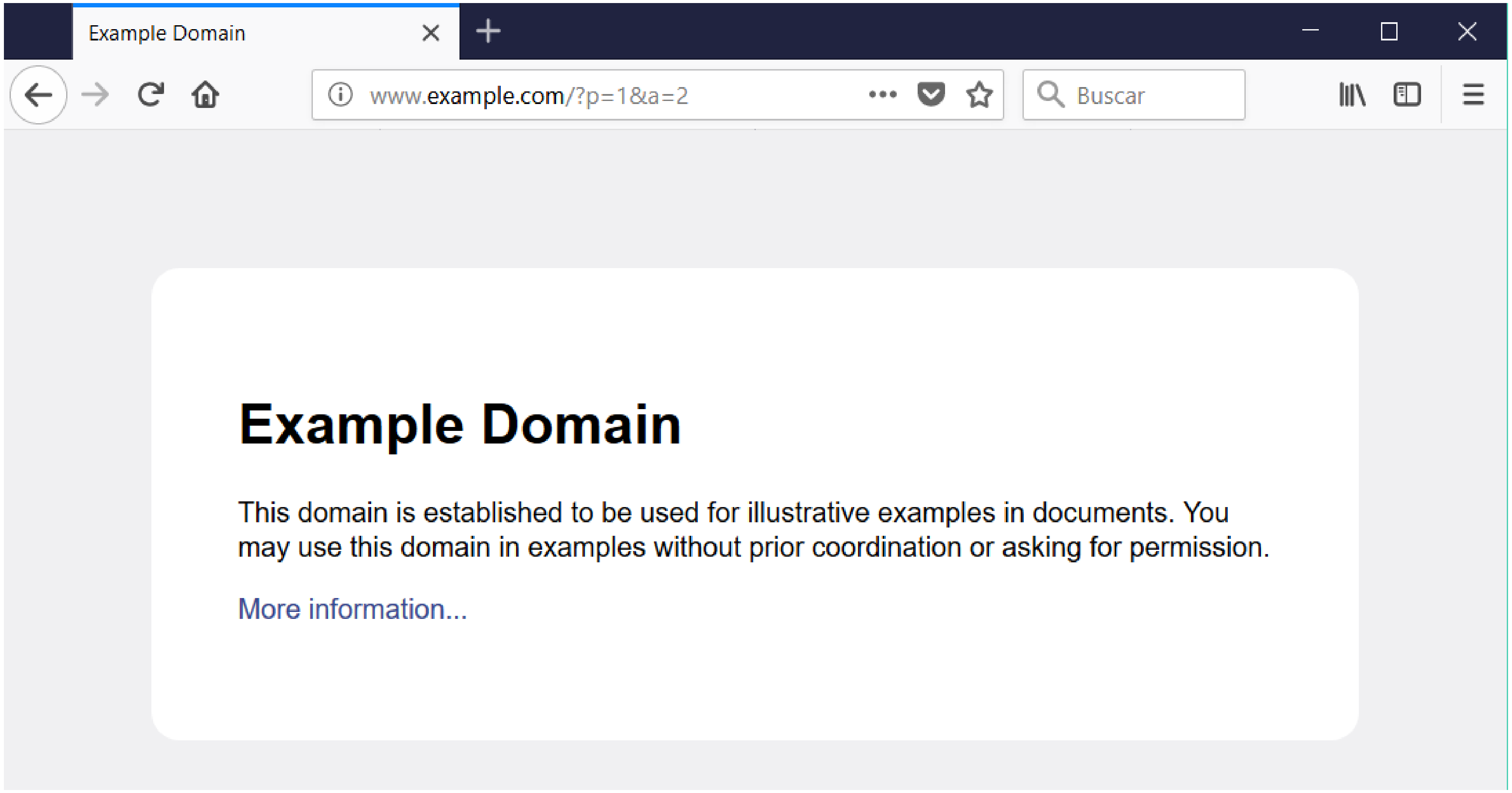
# HTTP


INET
Web Browser
Red TCP/IP
Web Server
Pedido de información
Información o redirección

## HTTP


GET layout.css
Image
GET image.png
GET page.html
GET video.mp4
Video
Ads
GET ads.jpg
Web document
The Internet
Web server
Ads server
Video server

## HTTP - Request

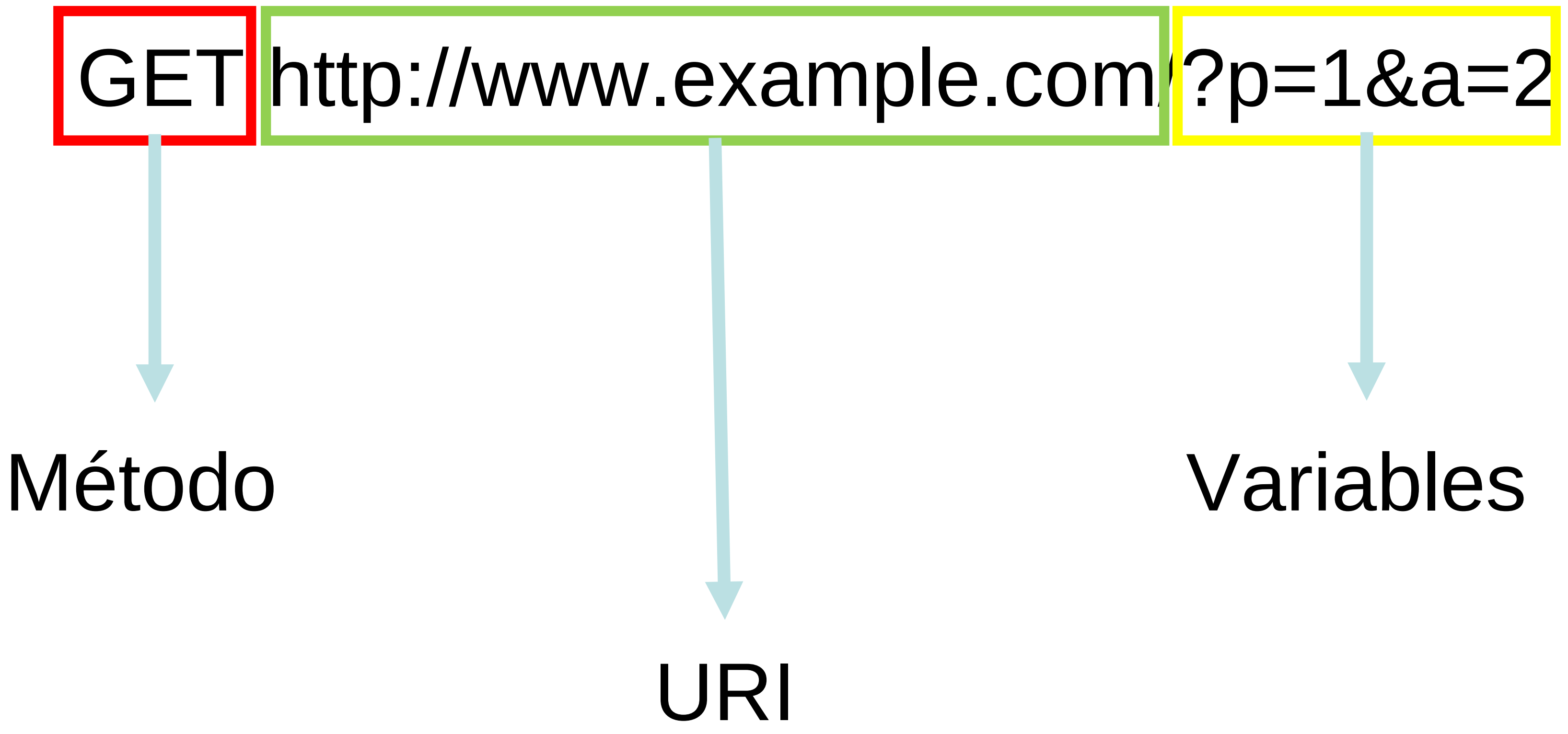


Example Domain

This domain is established to be used for illustrative examples in documents. You may use this domain in examples without prior coordination or asking for permission.

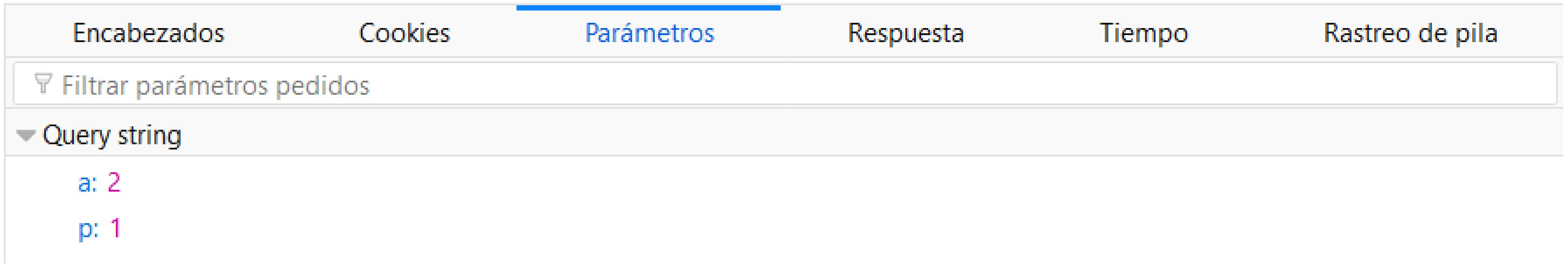
More information...

## HTTP - Request



# HTTP – Request – Parámetros

## GET

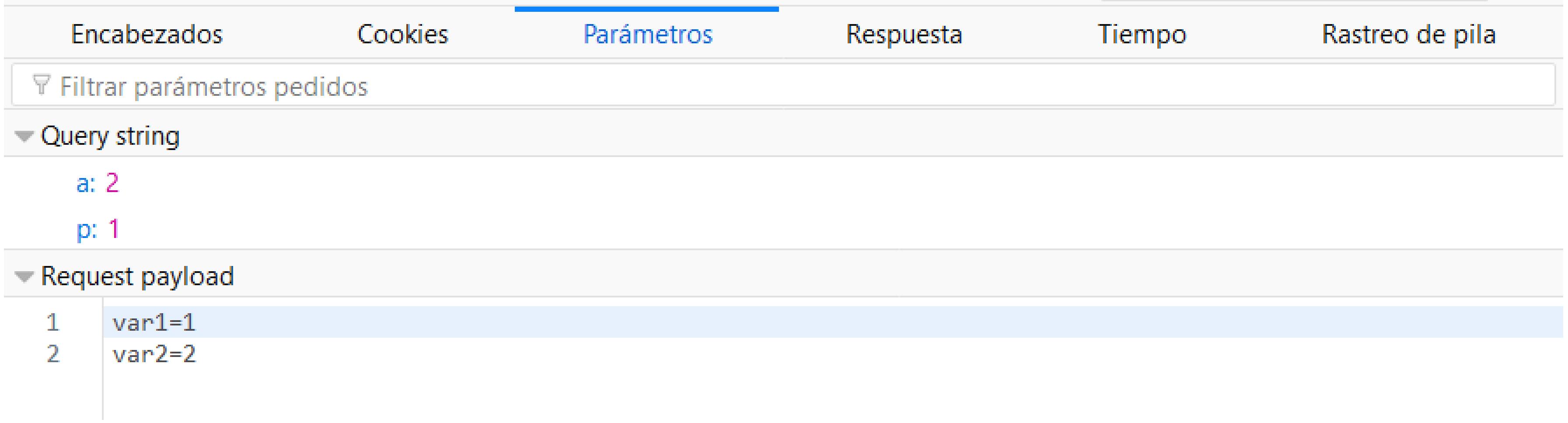


Query string

a: 2

p: 1

## POST



Query string

a: 2

p: 1

Request payload

```
var1=1
var2=2
```

## HTTP – Request

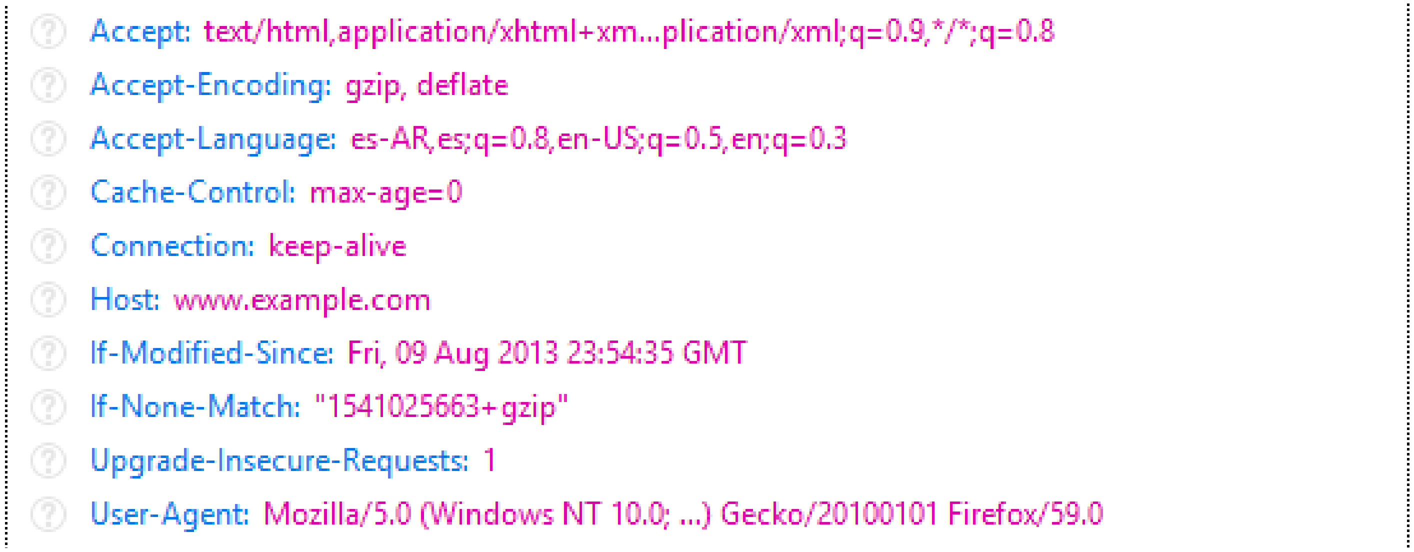
- Indica la acción que debe realizar el servidor.
  - GET: Petición de información sobre un recurso.
  - POST: Envío de datos a procesar.
  - DELETE: Borrar el recurso especificado.
  - OPTIONS: Devuelve los métodos soportados por el servidor.

Existen otros: PUT, CONNECT, TRACE, PATCH

- URI: Ubicación del recurso (http://example.com/index.html).

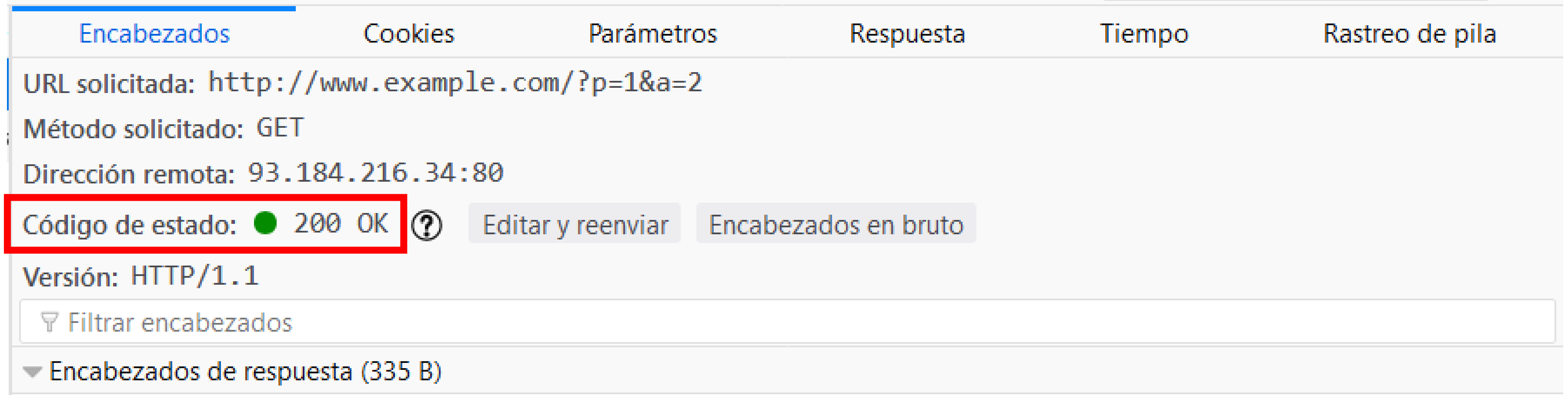
- Todo lo que viene despues de “?” son variables separadas por “&”.

## HTTP – Request – Headers


```
Accept: text/html,application/xhtml+xm...plication/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: es-AR,es;q=0.8,en-US;q=0.5,en;q=0.3
Cache-Control: max-age=0
Connection: keep-alive
Host: www.example.com
If-Modified-Since: Fri, 09 Aug 2013 23:54:35 GMT
If-None-Match: "1541025663+gzip"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; ...) Gecko/20100101 Firefox/59.0
```

Variables del cliente, dan información sobre la comunicación y sobre quién contacta. Permiten al cliente y al servidor enviar información adicional junto a una petición o respuesta.

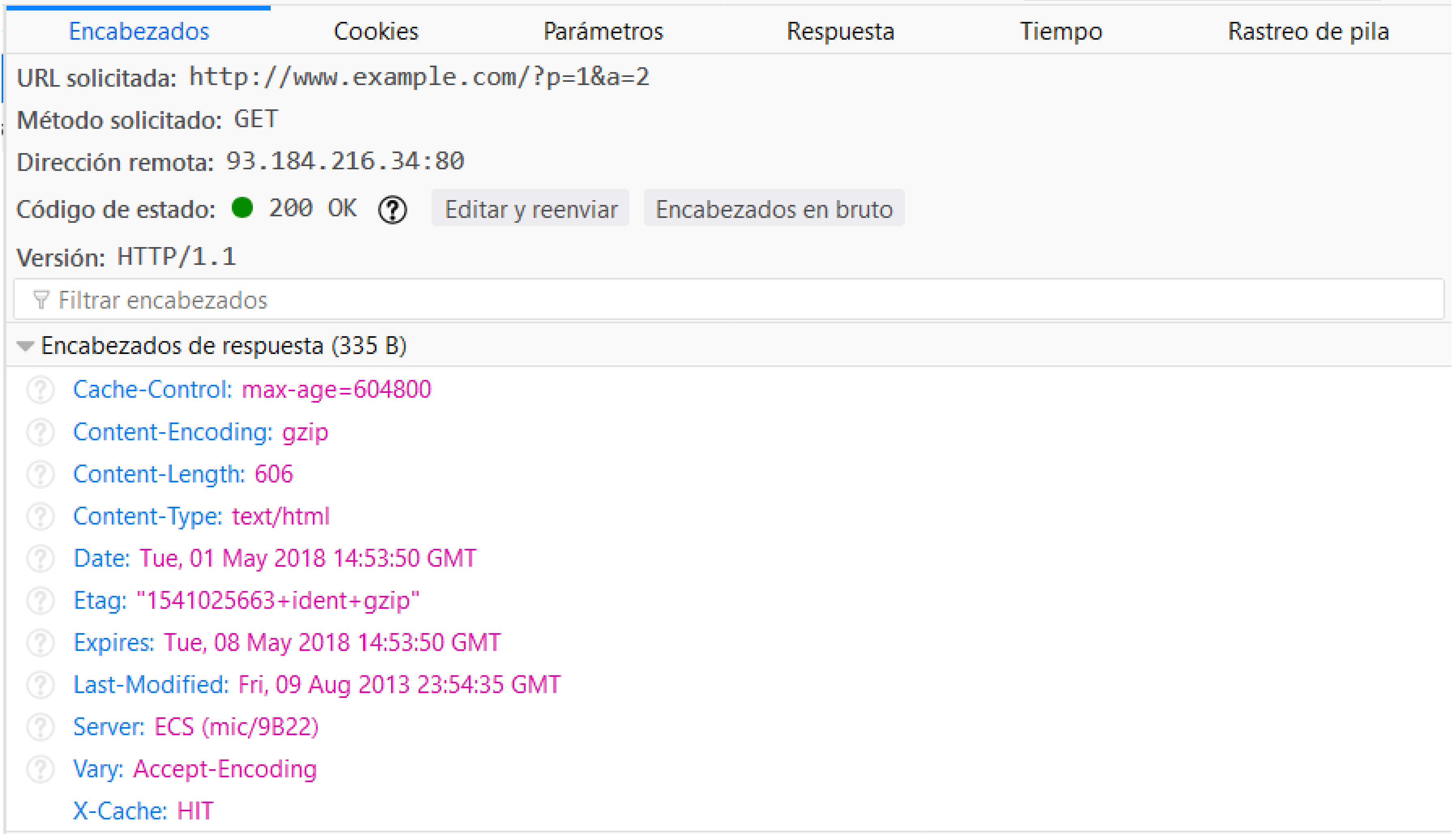
## HTTP – Response – Códigos



URL solicitada: `http://www.example.com/?p=1&a=2`
Método solicitado: GET
Dirección remota: 93.184.216.34:80
Código de estado: ● 200 OK
Versión: HTTP/1.1
Encabezados de respuesta (335 B)

- Código de respuesta a la acción.
  - 2xx: Éxito
  - 3xx: Redireción
  - 4xx: Error del cliente
  - 5xx: Error del servidor

# HTTP – Response – Headers

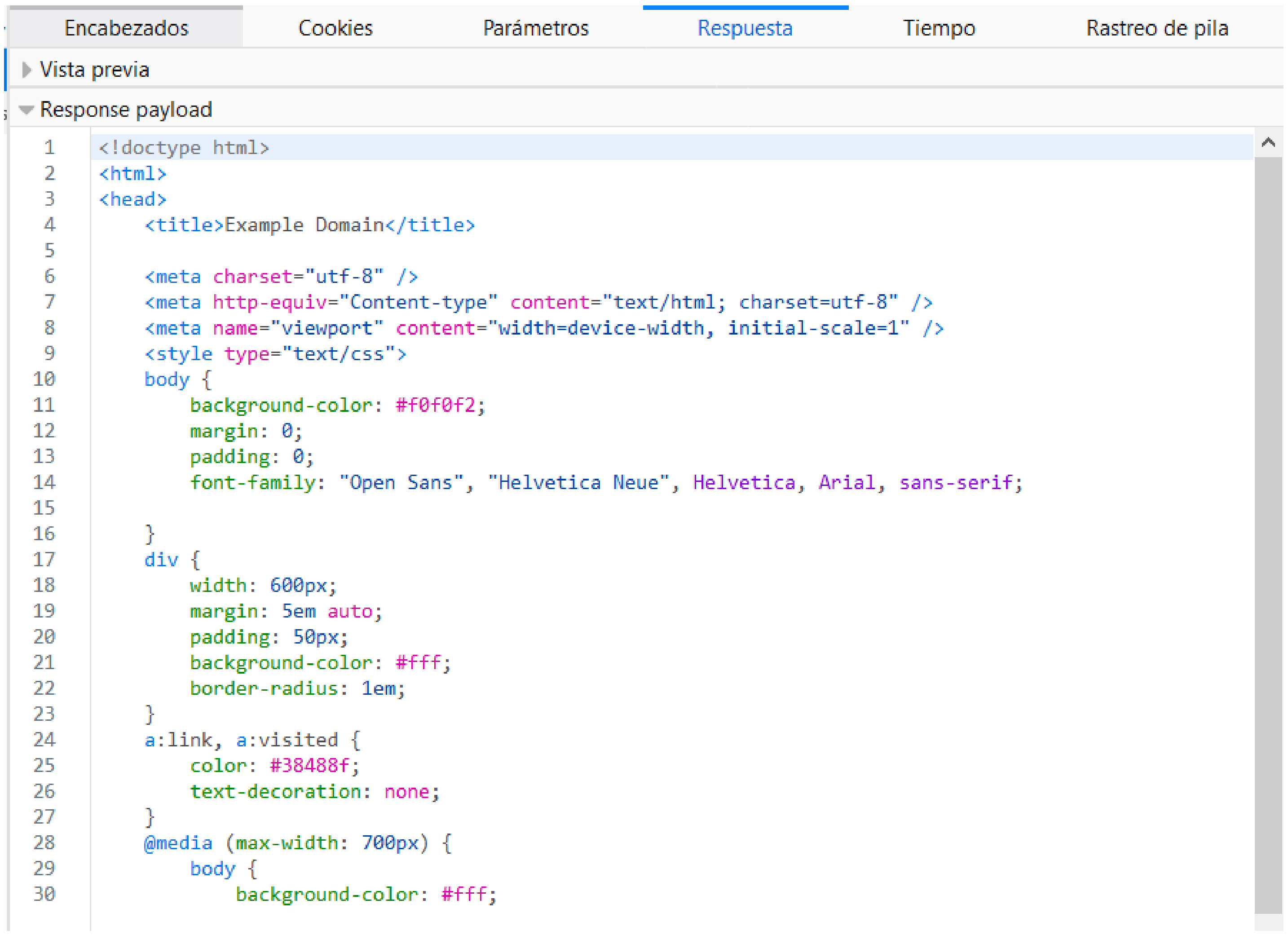


URL solicitada: http://www.example.com/?p=1&a=2
Método solicitado: GET
Dirección remota: 93.184.216.34:80
Código de estado: ● 200 OK
Versión: HTTP/1.1

Encabezados de respuesta (335 B)

```
Cache-Control: max-age=604800
Content-Encoding: gzip
Content-Length: 606
Content-Type: text/html
Date: Tue, 01 May 2018 14:53:50 GMT
Etag: "1541025663+ident+gzip"
Expires: Tue, 08 May 2018 14:53:50 GMT
Last-Modified: Fri, 09 Aug 2013 23:54:35 GMT
Server: ECS (mic/9B22)
Vary: Accept-Encoding
X-Cache: HIT
```

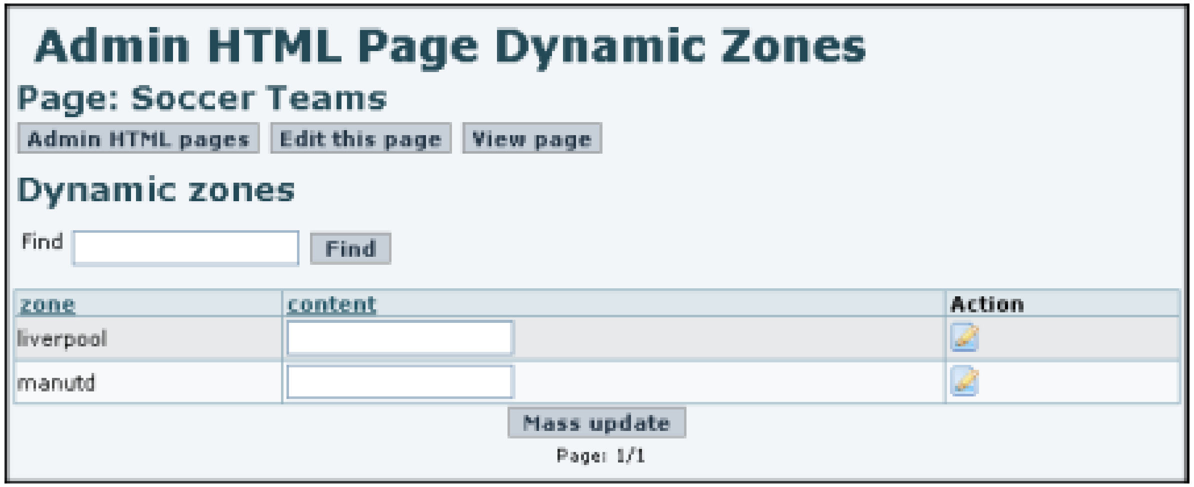
# HTTP – Response – Body



```
<!doctype html>
<html>
<head>
    <title>Example Domain</title>

    <meta charset="utf-8" />
    <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <style type="text/css">
    body {
        background-color: #f0f0f2;
        margin: 0;
        padding: 0;
        font-family: "Open Sans", "Helvetica Neue", Helvetica, Arial, sans-serif;

    }
    div {
        width: 600px;
        margin: 5em auto;
        padding: 50px;
        background-color: #fff;
        border-radius: 1em;
    }
    a:link, a:visited {
        color: #38488f;
        text-decoration: none;
    }
    @media (max-width: 700px) {
        body {
            background-color: #fff;
```

## HTML

- HyperText Markup Language.
- Diseño de texto ordenado por “tags”.
- Cada “tag” le informa al Web Browser como debe mostrar la información.
- Si el Browser entiende HTML entonces muestra un contenido formateado.



Admin HTML Page Dynamic Zones

Page: Soccer Teams

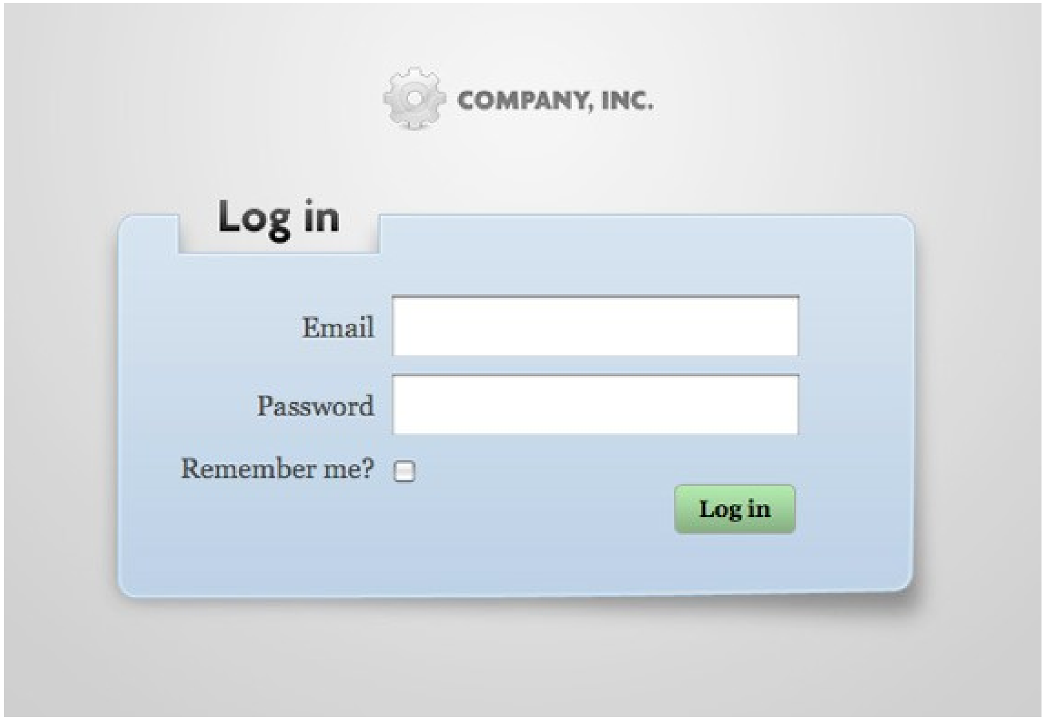
Dynamic zones

| zone | content | Action |
|---|---|---|
| liverpool | | |
| manutd | | |

Page: 1/1

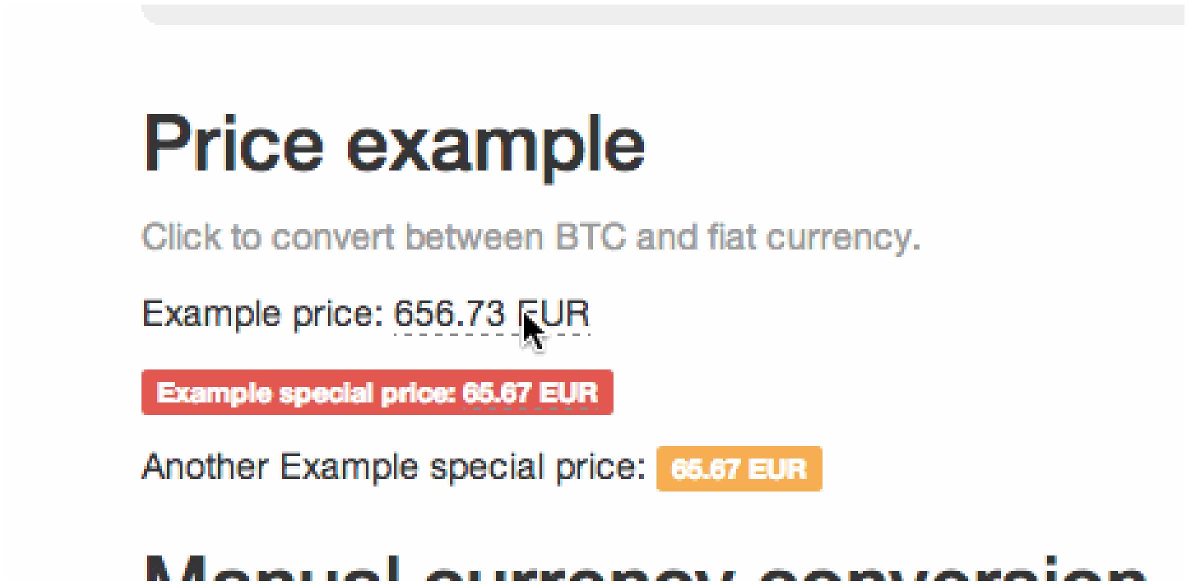
## HTML + CSS

- CSS (Cascading Style Sheet).
- Páginas solo con HTML son aburridas, necesitan estilos.
- Describe los colores, formatos, tipo de letra, bordes, etc...



## HTML + CSS + Javascript

- La página es estática, quiero que sea interactiva.
- Javascript permite ejecutar codigo dinámico en el cliente.



Price example

Click to convert between BTC and fiat currency.

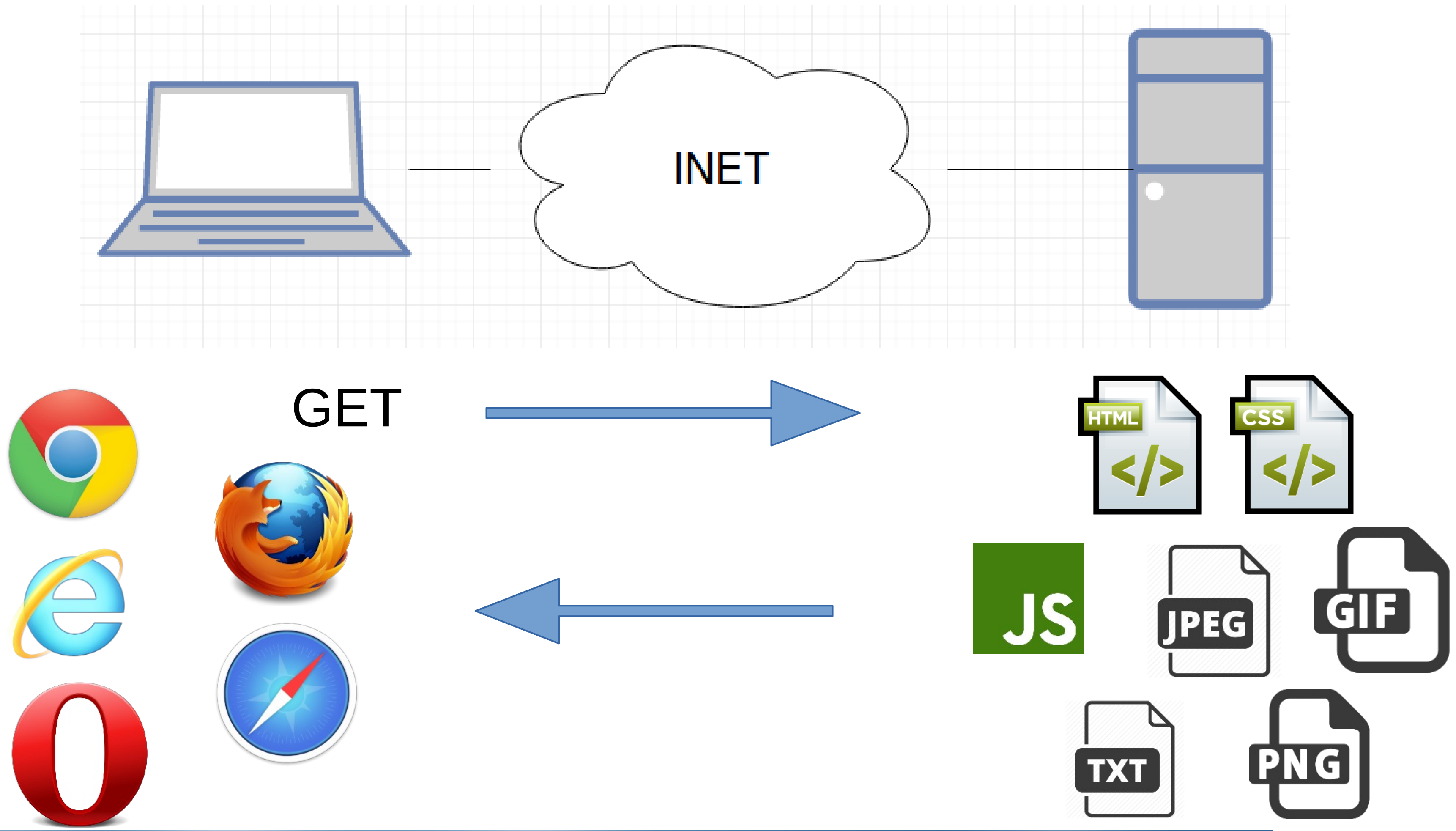
Example price: 656.73 EUR

Example special price: 65.67 EUR

Another Example special price: 65.67 EUR

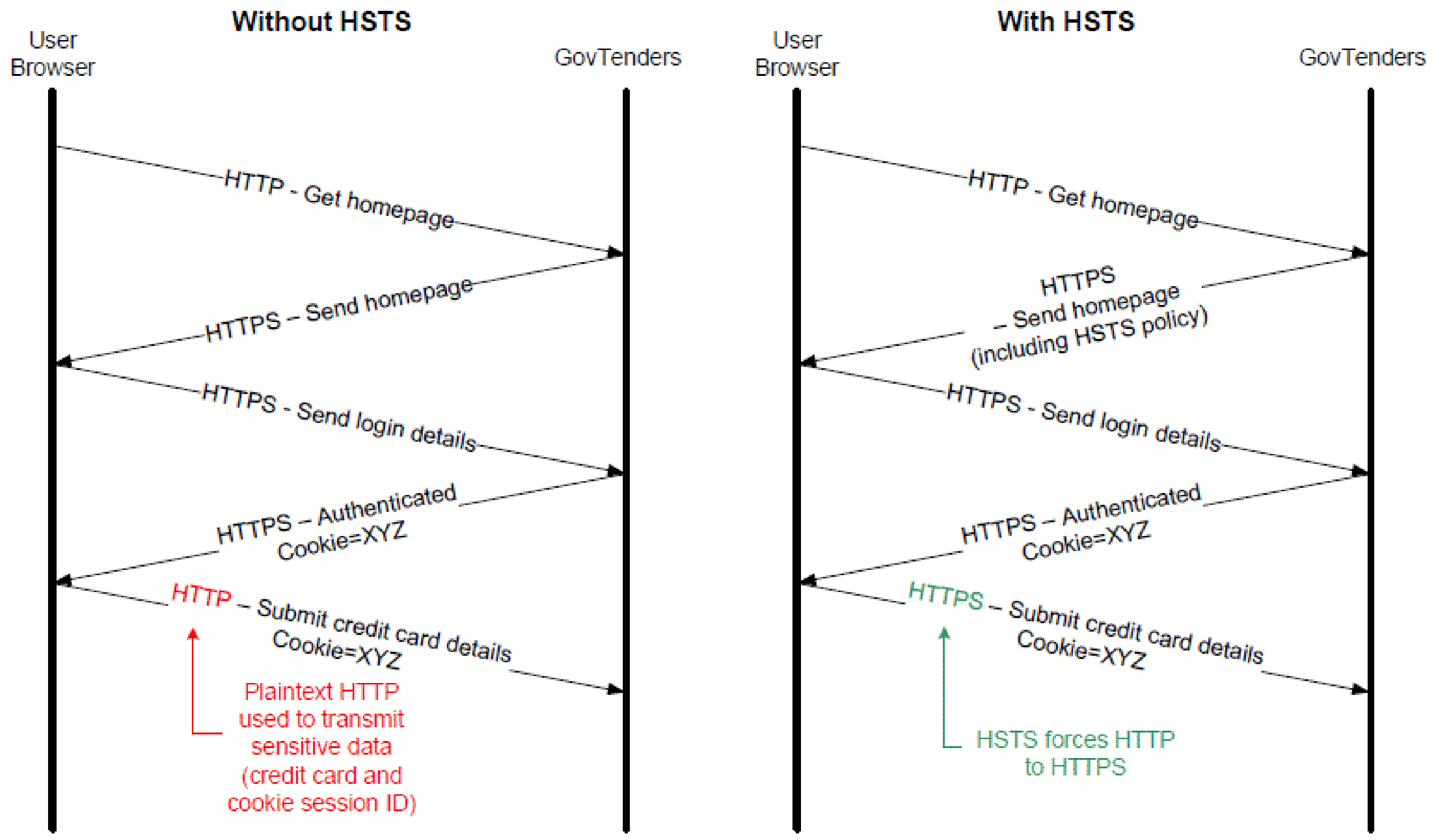
# Conexión web

- Todos estos recursos son archivos del servidor.



## Seguridad Web

# HSTS (HTTP Strict Transport Security)


Without HSTS
User Browser
GovTenders
HTTP - Get homepage
HTTPS – Send homepage
HTTPS - Send login details
HTTPS – Authenticated Cookie=XYZ
HTTP – Submit credit card details Cookie=XYZ
Plaintext HTTP used to transmit sensitive data (credit card and cookie session ID)
With HSTS
User Browser
GovTenders
HTTP - Get homepage
HTTPS – Send homepage (including HSTS policy)
HTTPS - Send login details
HTTPS – Authenticated Cookie=XYZ
HTTPS – Submit credit card details Cookie=XYZ
HSTS forces HTTP to HTTPS

## HSTS

**Ejemplo para Apache:**

```
# Optionally load the headers module:
LoadModule headers_module modules/mod_headers.so

<VirtualHost 67.89.123.45:443>
    Header always set Strict-Transport-Security "max-age=63072000; includeSubdomains;"
</VirtualHost>
```

## Ejemplo Redirect para Apache:

```
<VirtualHost *:80>
  [...]
  ServerName example.com
  Redirect permanent / https://example.com/
</VirtualHost>
```

## Ejemplo Rewrite para Apache:

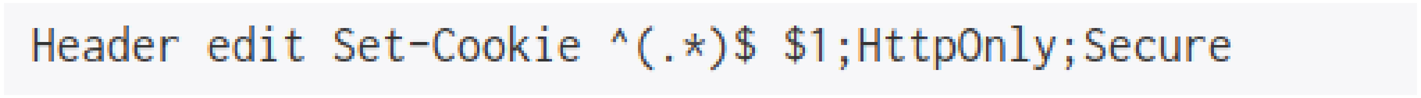
```
<VirtualHost *:80>
  [...]
  <IfModule mod_rewrite.c>
    RewriteEngine On
    RewriteCond %{HTTPS} off
    RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI}
  </IfModule>
</VirtualHost>
```

## Cookies Attribs

## Cookies Attribs: HttpOnly & Secure

- *HttpOnly*: Evita que la cookie sea extraída por algún script malicioso
- *Secure*: Indica que esta cookie únicamente viajará por HTTPS

Config en Apache:



```
Header edit Set-Cookie ^(.*)$ $1;HttpOnly;Secure
```

## X-Frame-Options

Evita que caigamos en la técnica Clickjacking

Valores posibles:

"DENY", "SAMEORIGIN" o "ALLOW-FROM uri"

Ejemplo para Apache:

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

## Ataques Web Comunes

## HTTP

- HTTP es puro texto, es fácil de modificarlo.
- Se podrían modificar los parámetros de los request HTTP.

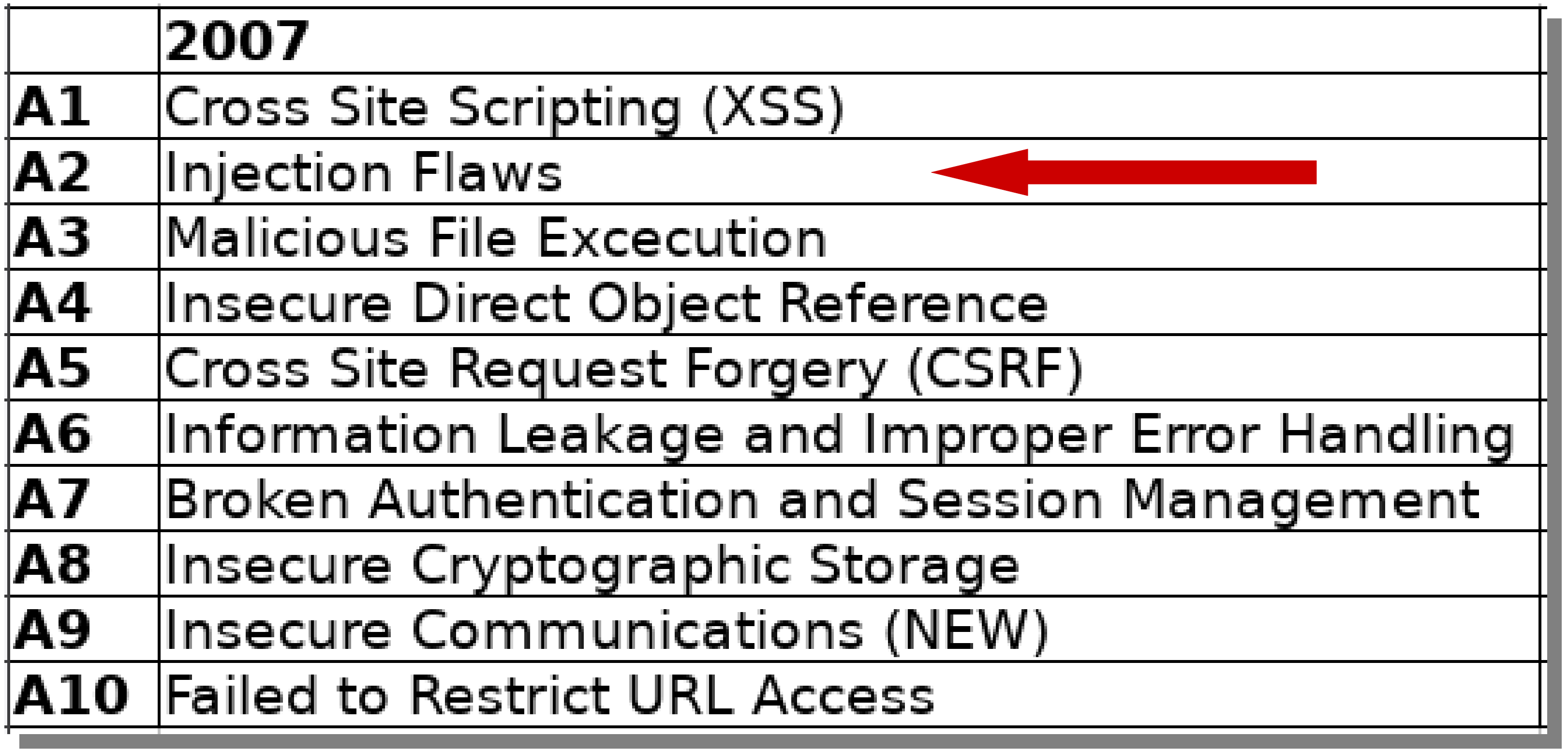
```
GETTTTTT /sitio/index.html HTTP/1.1
Host: www.misitio.com
User-Agent: Mozilla Firefox
```

```
GET /sitio/index.html?p=<script>alert(1);</script>&a=2 HTTP/1.1
Host: www.misitio.com
User-Agent: Mozilla Firefox
```

```
GET ../../../../etc/passwd HTTP/1.1
Host: www.misitio.com
User-Agent: Mozilla Firefox
```

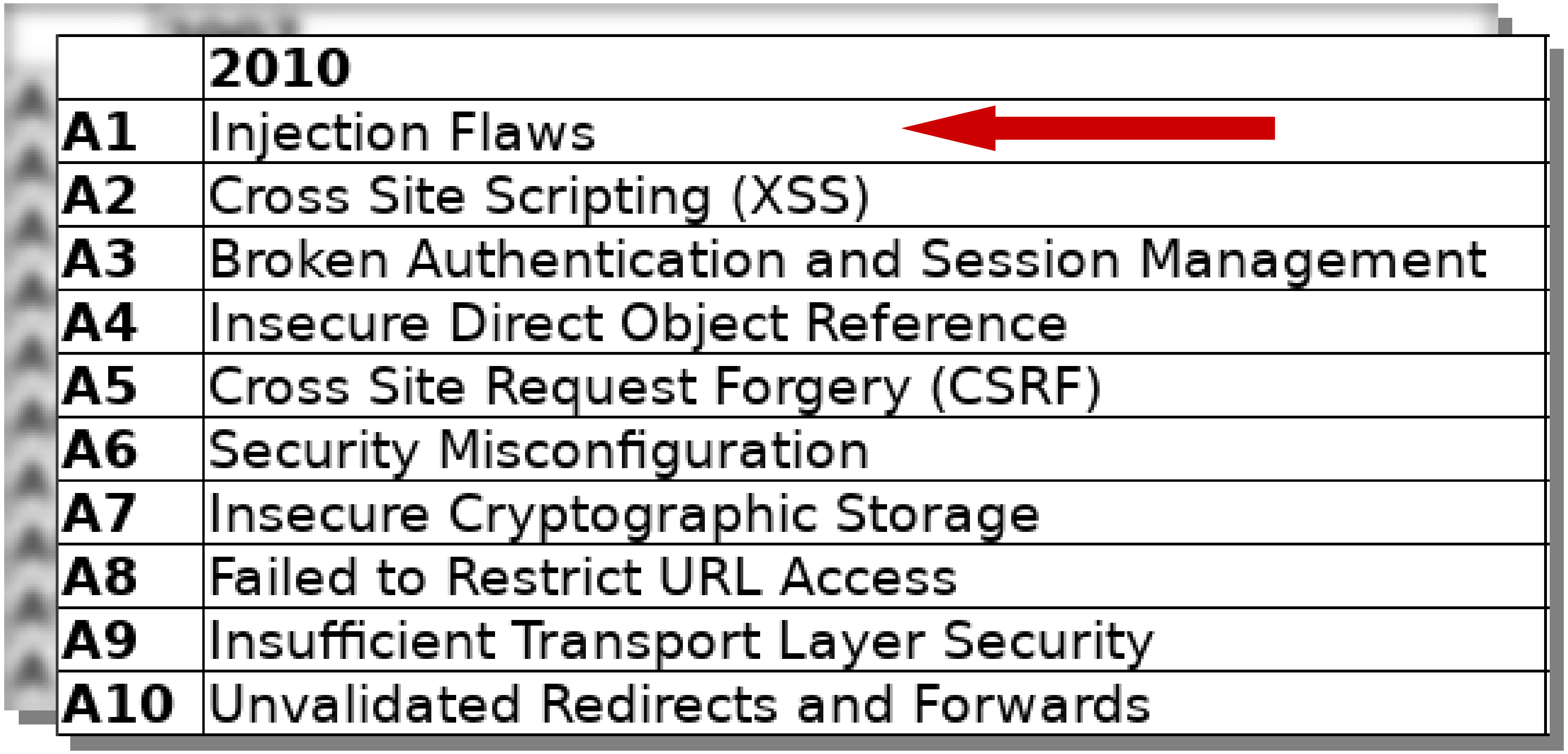
```
POST  /sitio/login.php HTTP/1.1
Host: www.misitio.com
User-Agent: Mozilla Firefox

username=’ or 1=1; – &password=pass
```

# Evolución del OWASP Top 10



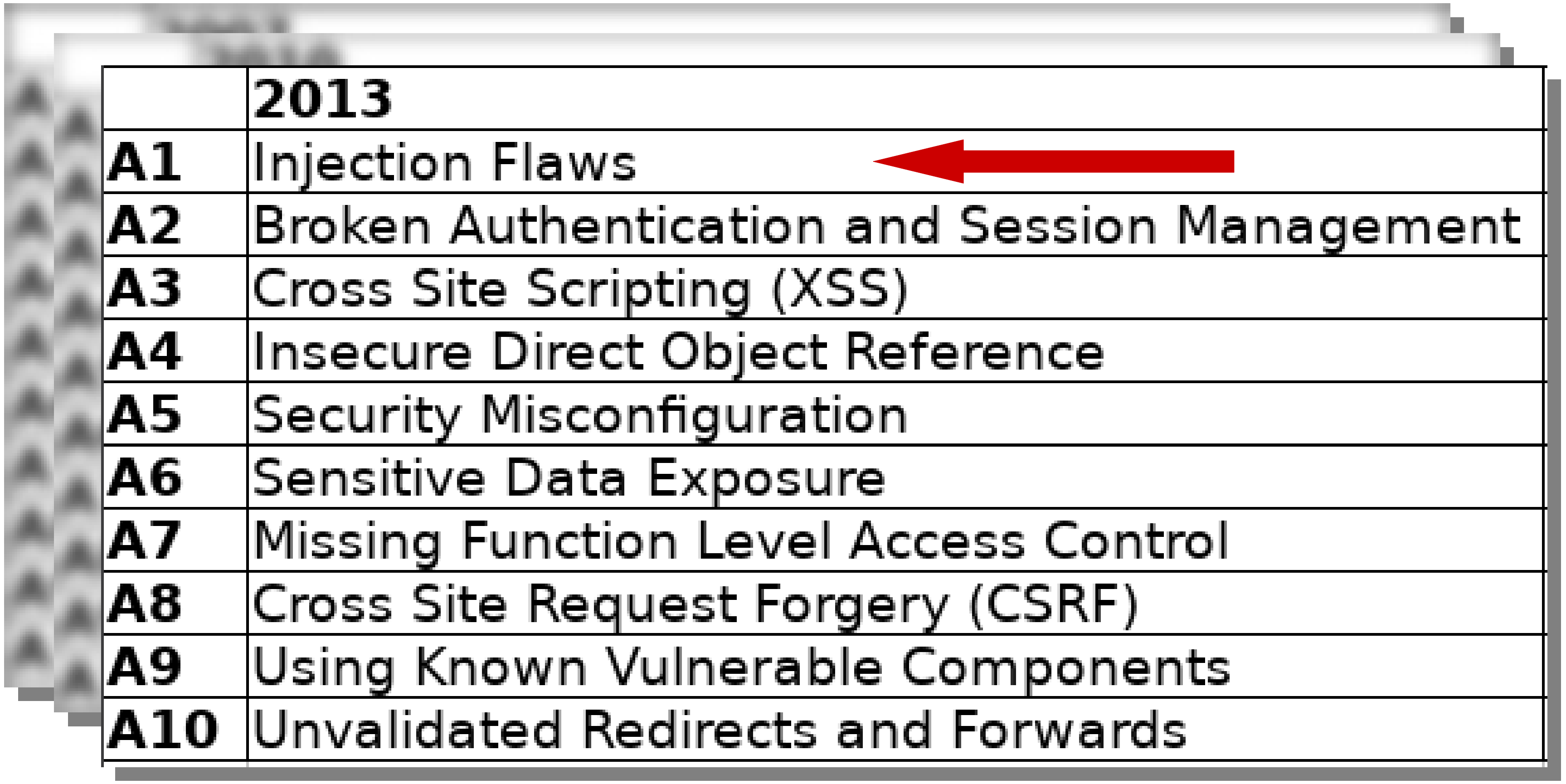
| | 2007 |
|---|---|
| A1 | Cross Site Scripting (XSS) |
| A2 | Injection Flaws |
| A3 | Malicious File Excecution |
| A4 | Insecure Direct Object Reference |
| A5 | Cross Site Request Forgery (CSRF) |
| A6 | Information Leakage and Improper Error Handling |
| A7 | Broken Authentication and Session Management |
| A8 | Insecure Cryptographic Storage |
| A9 | Insecure Communications (NEW) |
| A10 | Failed to Restrict URL Access |

# Evolución del OWASP Top 10



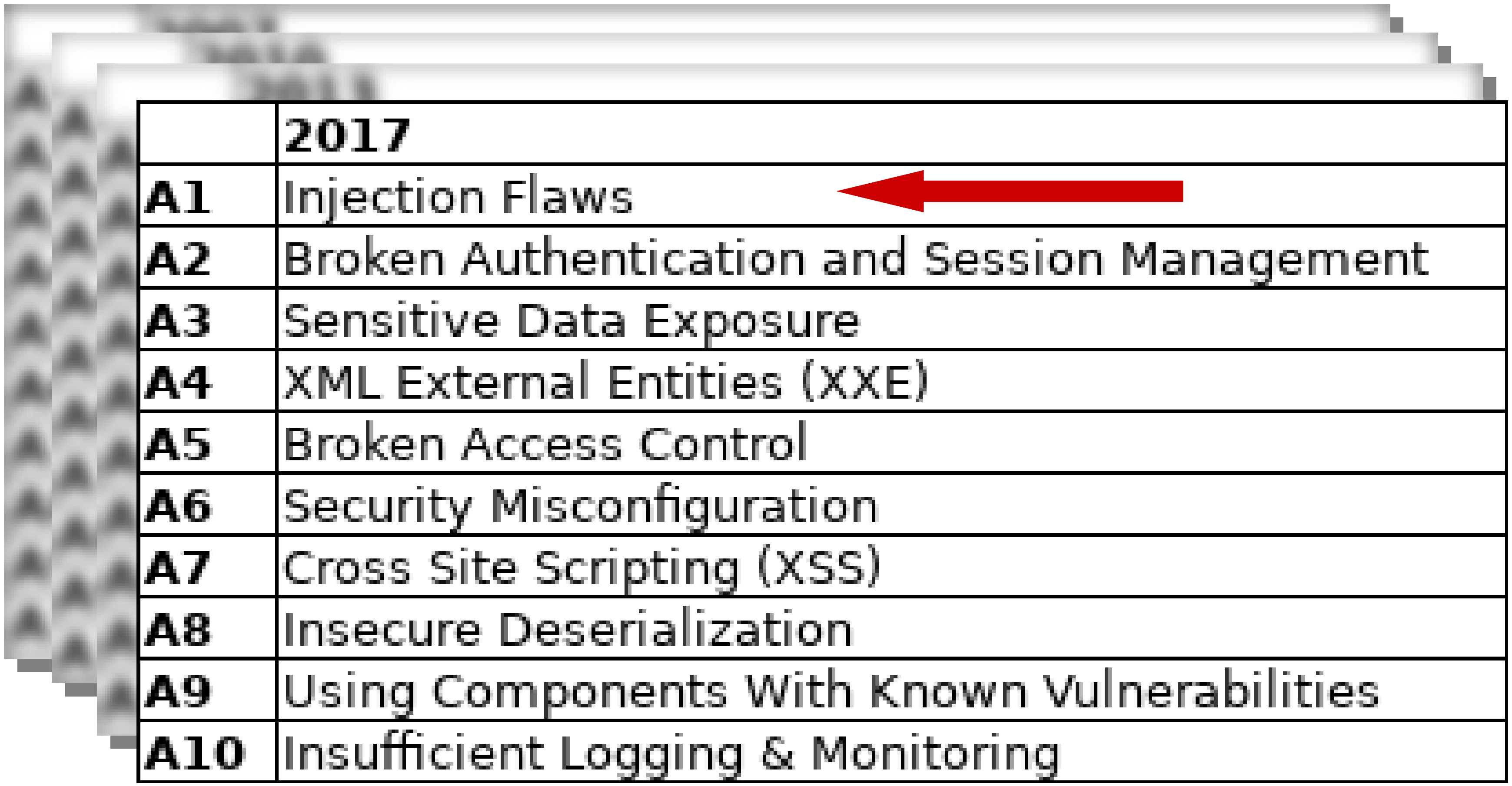
| | 2010 |
|---|---|
| **A1** | Injection Flaws |
| **A2** | Cross Site Scripting (XSS) |
| **A3** | Broken Authentication and Session Management |
| **A4** | Insecure Direct Object Reference |
| **A5** | Cross Site Request Forgery (CSRF) |
| **A6** | Security Misconfiguration |
| **A7** | Insecure Cryptographic Storage |
| **A8** | Failed to Restrict URL Access |
| **A9** | Insufficient Transport Layer Security |
| **A10** | Unvalidated Redirects and Forwards |

## Evolución del OWASP Top 10



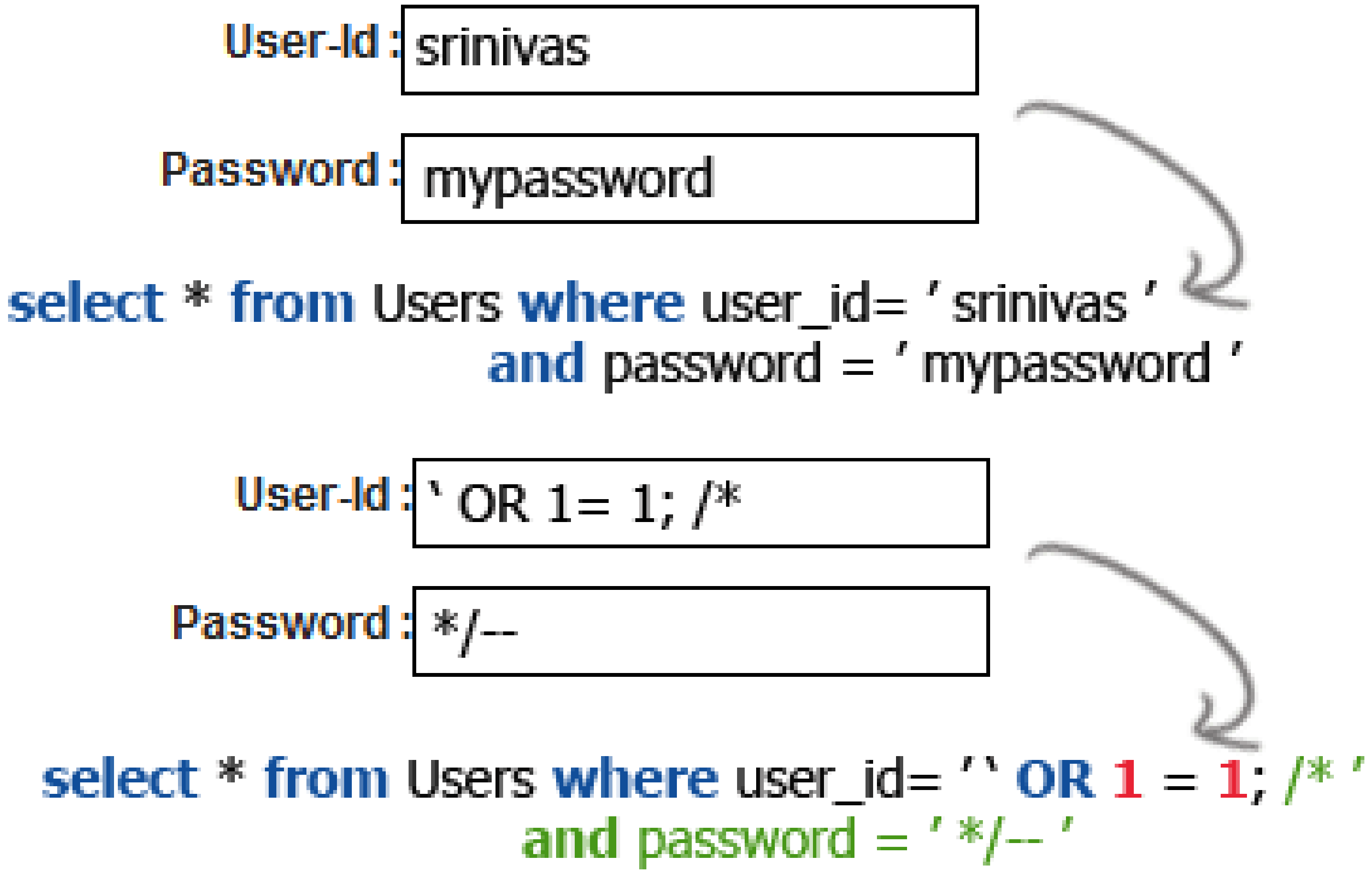
| | 2013 |
|---|---|
| **A1** | Injection Flaws |
| **A2** | Broken Authentication and Session Management |
| **A3** | Cross Site Scripting (XSS) |
| **A4** | Insecure Direct Object Reference |
| **A5** | Security Misconfiguration |
| **A6** | Sensitive Data Exposure |
| **A7** | Missing Function Level Access Control |
| **A8** | Cross Site Request Forgery (CSRF) |
| **A9** | Using Known Vulnerable Components |
| **A10** | Unvalidated Redirects and Forwards |

# Evolución del OWASP Top 10



| | 2017 |
|---|---|
| **A1** | Injection Flaws |
| **A2** | Broken Authentication and Session Management |
| **A3** | Sensitive Data Exposure |
| **A4** | XML External Entities (XXE) |
| **A5** | Broken Access Control |
| **A6** | Security Misconfiguration |
| **A7** | Cross Site Scripting (XSS) |
| **A8** | Insecure Deserialization |
| **A9** | Using Components With Known Vulnerabilities |
| **A10** | Insufficient Logging & Monitoring |

SQL Injection



User-Id : srinivas

Password : mypassword

```
select * from Users where user_id= ' srinivas '
                          and password = ' mypassword '
```

User-Id : ` OR 1= 1; /*

Password : */--

```
select * from Users where user_id= '` OR 1 = 1; /* '
                          and password = ' */-- '
```

KALI LINUX

Apache

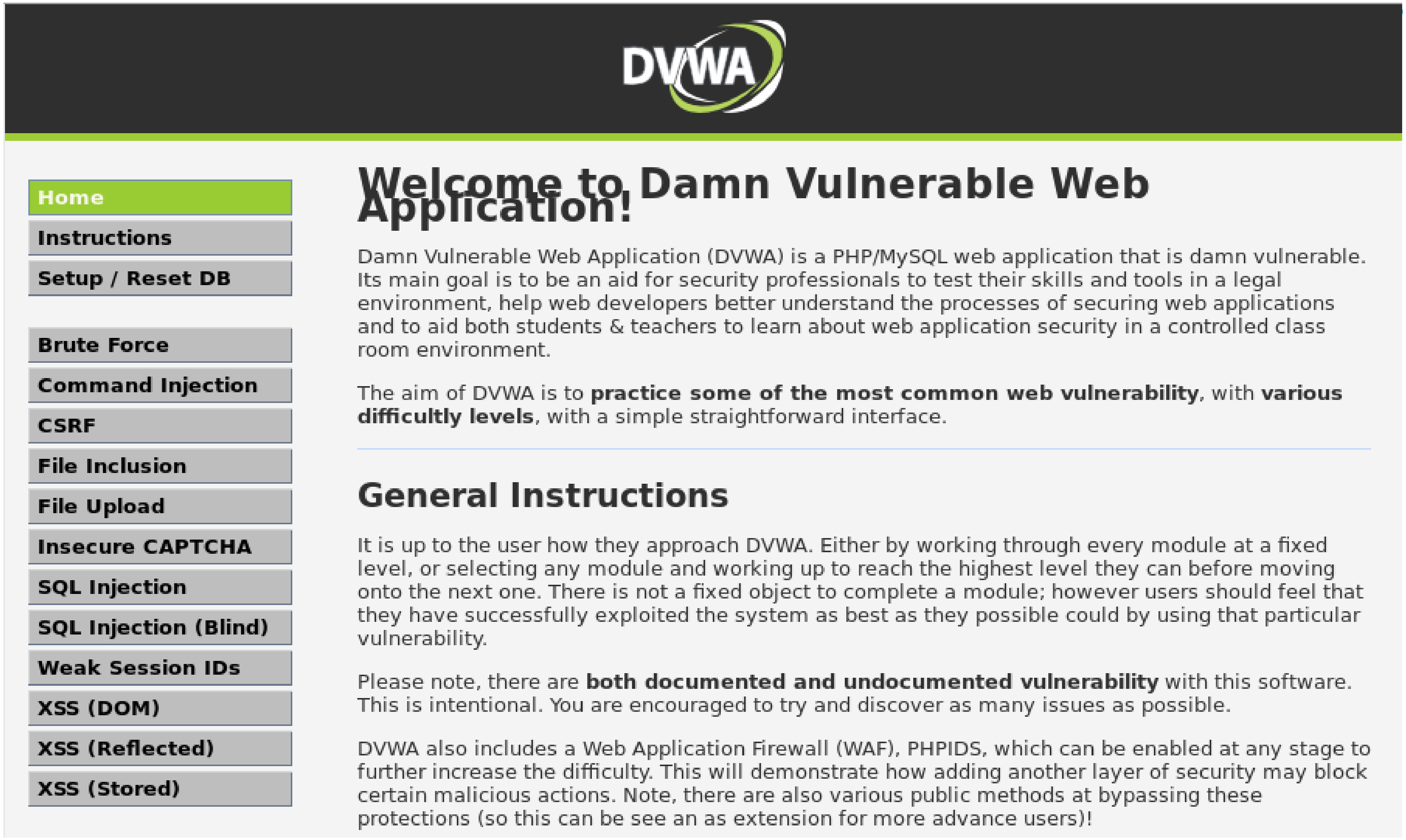
php

MariaDB

DVWA

modsecurity

Open Source Web Application Firewall



## Welcome to Damn Vulnerable Web Application!

Damn Vulnerable Web Application (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goal is to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and to aid both students & teachers to learn about web application security in a controlled class room environment.

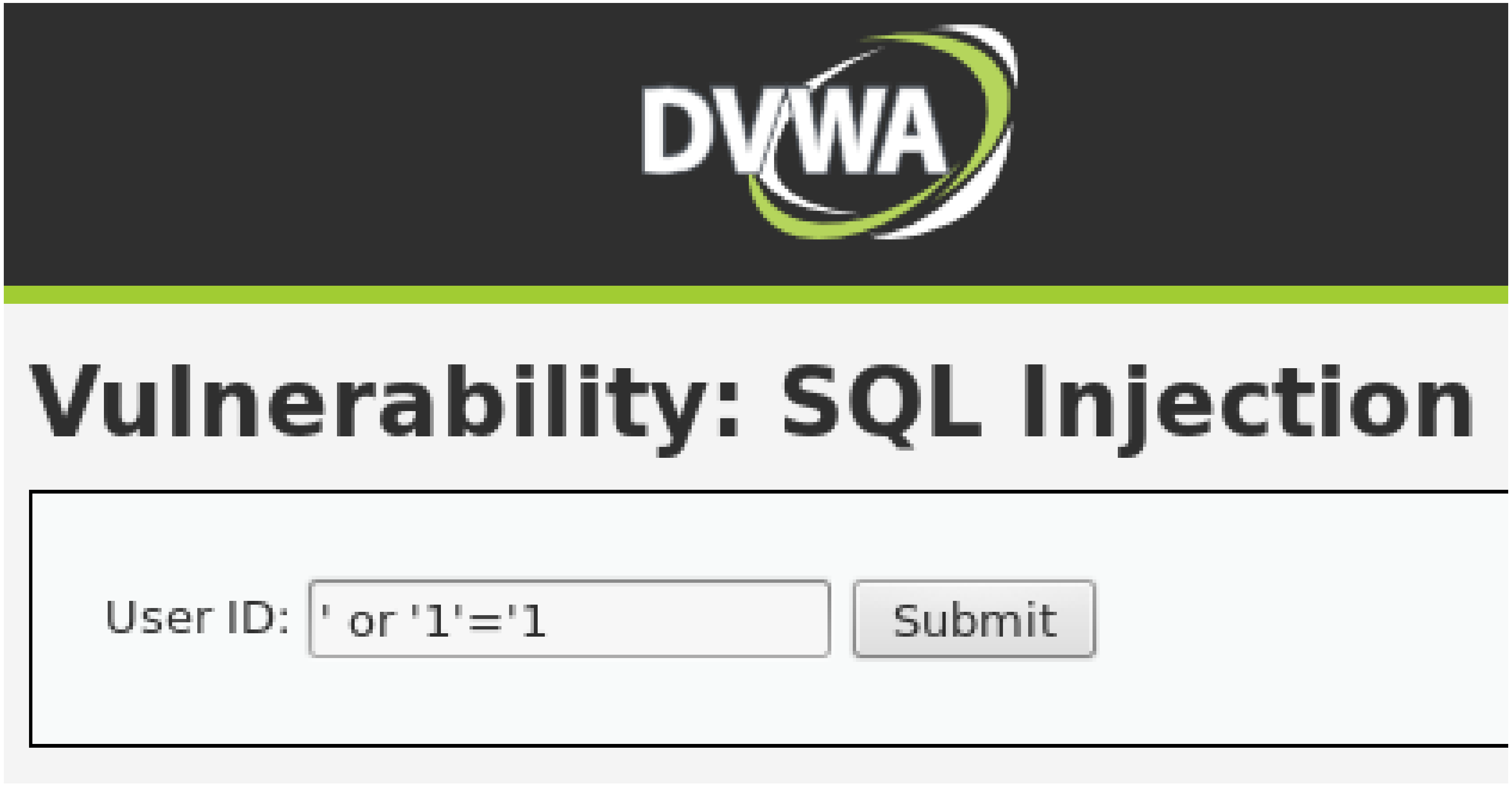
The aim of DVWA is to **practice some of the most common web vulnerability**, with **various difficultly levels**, with a simple straightforward interface.

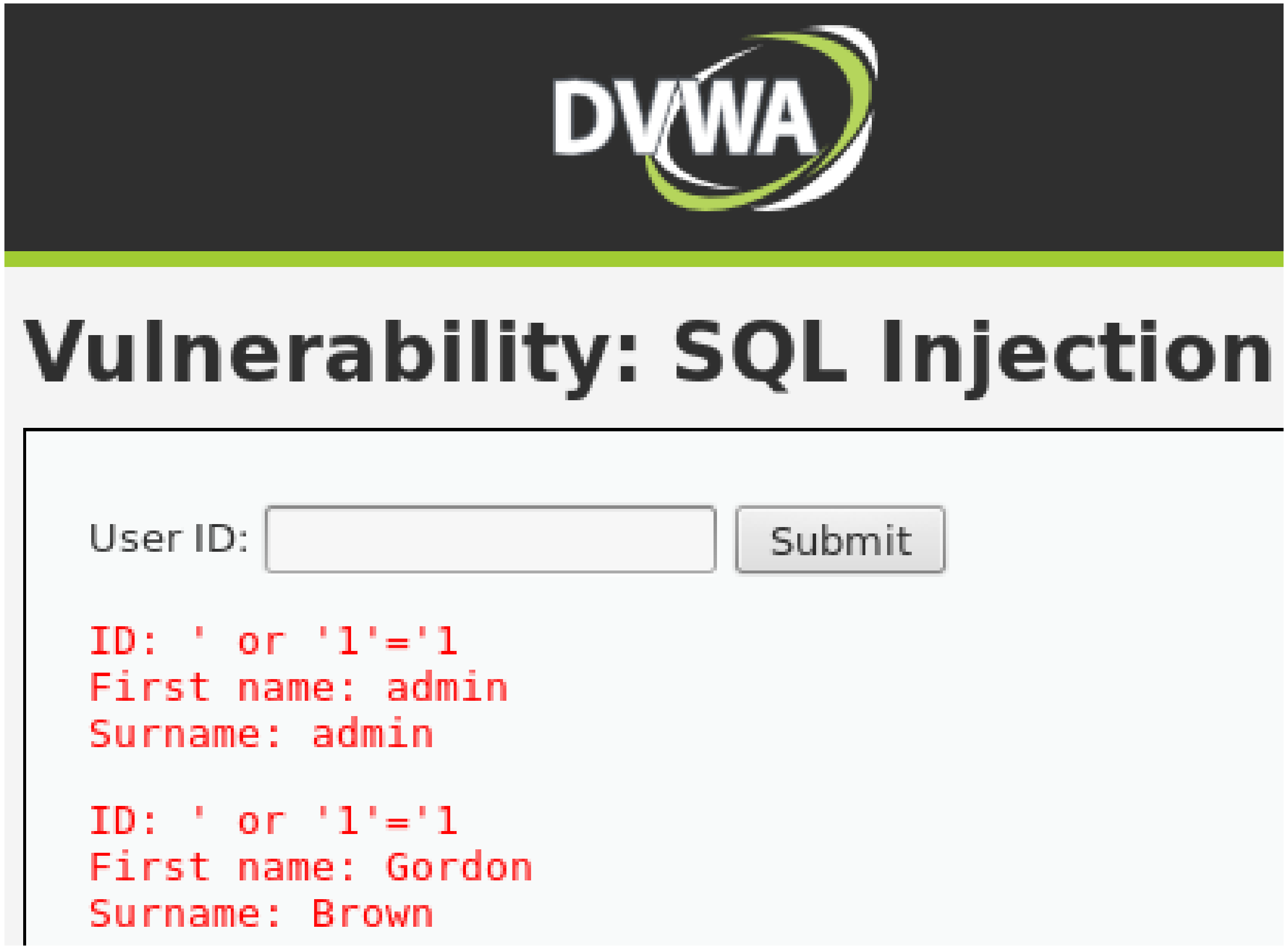
### General Instructions

It is up to the user how they approach DVWA. Either by working through every module at a fixed level, or selecting any module and working up to reach the highest level they can before moving onto the next one. There is not a fixed object to complete a module; however users should feel that they have successfully exploited the system as best as they possible could by using that particular vulnerability.

Please note, there are **both documented and undocumented vulnerability** with this software. This is intentional. You are encouraged to try and discover as many issues as possible.

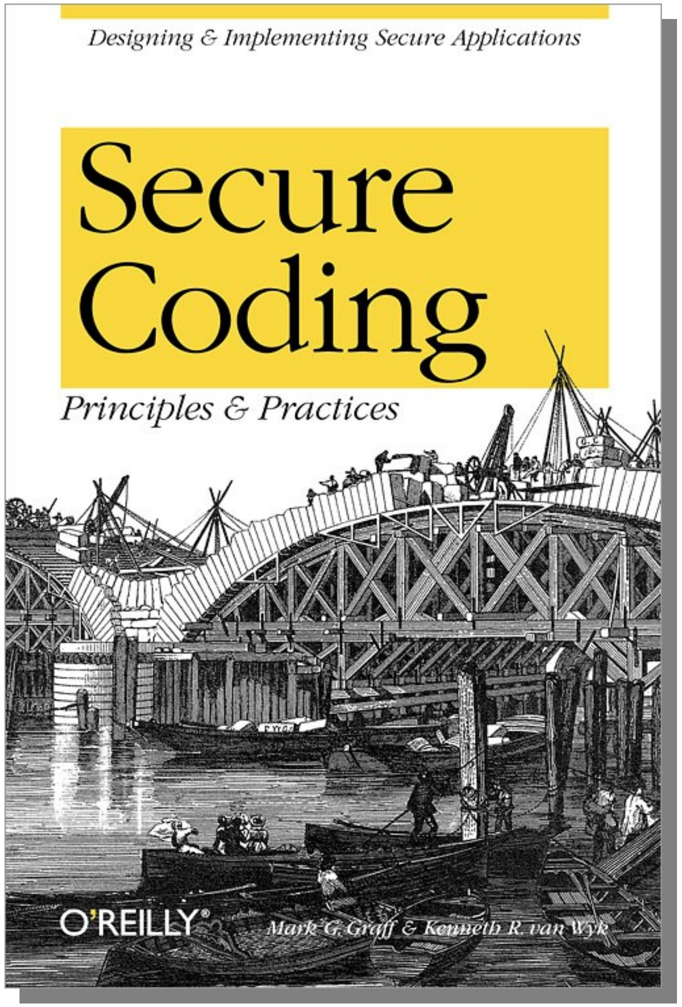
DVWA also includes a Web Application Firewall (WAF), PHPIDS, which can be enabled at any stage to further increase the difficulty. This will demonstrate how adding another layer of security may block certain malicious actions. Note, there are also various public methods at bypassing these protections (so this can be see an as extension for more advance users)!


Vulnerability: SQL Injection



# Vulnerability: SQL Injection

```
ID: ' or '1'='1
First name: admin
Surname: admin

ID: ' or '1'='1
First name: Gordon
Surname: Brown
```

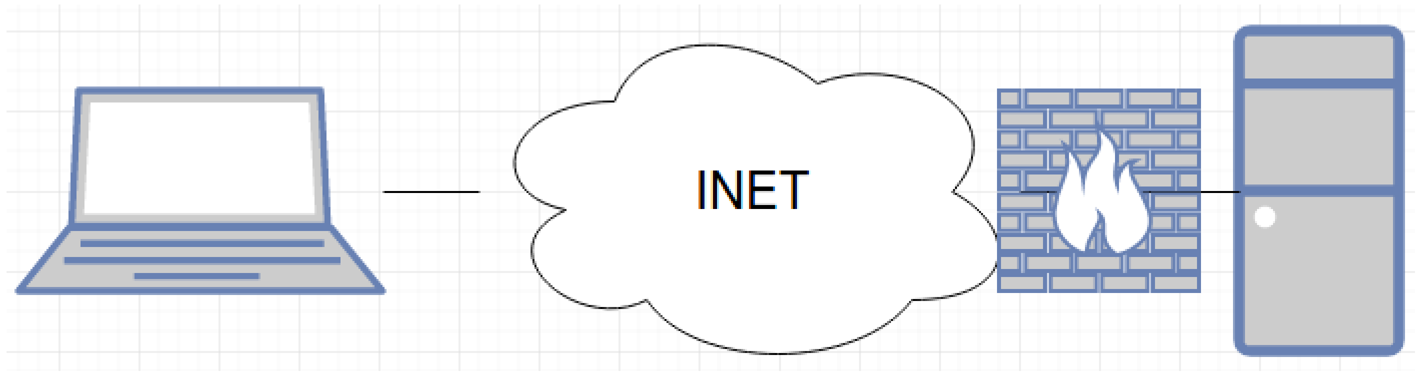

Designing & Implementing Secure Applications
Secure Coding
Principles & Practices
O'REILLY
Mark G. Graff & Kenneth R. van Wyk

## WAF

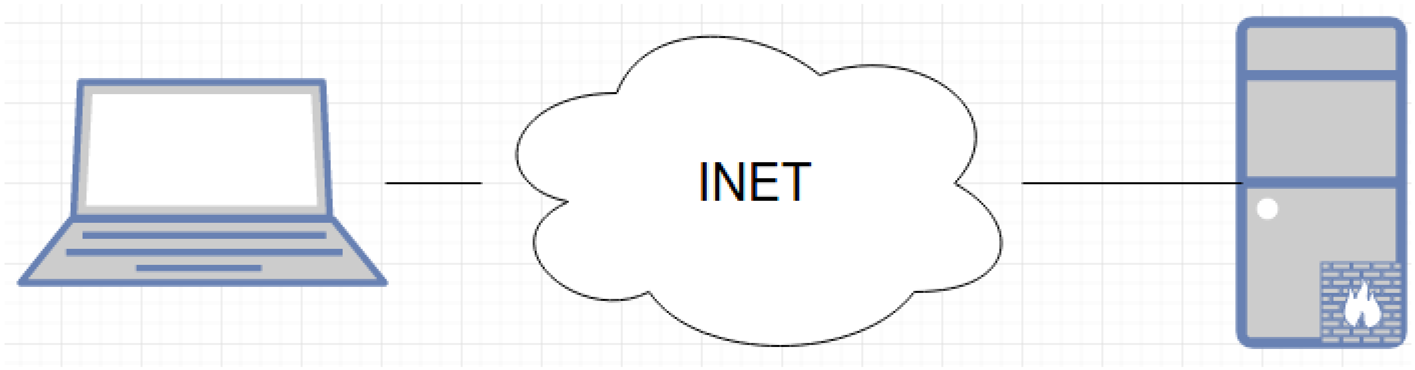
- Web Application Firewall/Filter.
- Es un IPS/IDS específicamente diseñado para HTTP.
- Contiene un set de reglas que detectan ataques web en los Request y Response HTTP.
- Versión OpenSource: ModSecurity.
  - Módulo de Apache o NGINX (también IIS).
  - Intercepta los mensajes HTTP antes de que lleguen a destino.

## WAF – Modos de uso

### Proxy reverso



### Interno al servidor



modsecurity

Open Source Web Application Firewall

## Capacidades de ModSecurity

- Interpretación completa del estándar
- Loggear el contenido de HTTP requests / responses
- Detección y bloqueo de ataques (basado en reglas)
- Funciones adicionales (cálculos de hashes, acceso a archivos, etc)
- Análisis de HTTP y HTTPS

## Virtual Patching

- Creación de reglas para bloqueo de ataques específicos
- Bloqueo de ataques para los que no hay parche (zero day)
- Menor ventana de oportunidad para los atacantes

## Modo Detection-Only

- No se interfiere en el tráfico
- Se dejan registros de todas las reglas que concuerdan
- Sirve para depurar y probar configuraciones
- Trabaja de forma similar a un IDS (solamente detecta)

## Modo Self-Contained

- Si una regla concuerda, se bloquea la petición
- Las reglas siguientes no se ejecutan
- Pros: Consume menos recursos y es más fácil de interpretar
- Contras: No hay punto intermedio, las reglas bloquean o no

## Modo Anomaly Score

- Cada regla que concuerda eleva el puntaje de “anomalía”
- Al final, si $score >= $threshold, se bloquea la petición
- Pros: Es más flexible y personalizable
- Contras: Consume más memoria y es más complejo de depurar
- setvar:tx.anomaly_score=+%{tx.warning_anomaly_score}
- setvar:tx.anomaly_score=+%{tx.critical_anomaly_score}

## Fases de Request / Response

1) Request headers (REQUEST_HEADERS)

2) Request body (REQUEST_BODY)

3) Response headers (RESPONSE_HEADERS)

4) Response body (RESPONSE_BODY)

5) Logging (LOGGING)

## Configuración de tipo de filtrado

```
# Self-contained
SecDefaultAction "phase:1,log,auditlog,deny,status:403"
SecDefaultAction "phase:2,log,auditlog,deny,status:403"

# Anomaly Score
SecDefaultAction "phase:1,log,auditlog,pass"
SecDefaultAction "phase:2,log,auditlog,pass"
```

## Sintaxis de Reglas (SecRule)

```
SecRule VARIABLES OPERATOR [ACTIONS]
```

## Variables Importantes

- ARGS
- ARGS_NAMES
- GEO
- REMOTE_ADDR
- REQUEST_BODY
- REQUEST_COOKIES
- REQUEST_HEADERS
- REQUEST_URI
- RESPONSE_BODY

## Operadores Importantes

- contains: contiene la cadena
- eq: igual a
- ge: mayor o igual a
- le: menor o igual a
- rx: expresión regular (operador por defecto)

## Acciones Importantes

- allow: deja pasar y no evalúa mas reglas
- block: hacer lo que diga SecDefaultAction
- chain: encadenar regla con la regla siguiente
- deny: bloquear la petición (devolver por defecto 403)
- drop: cerrar la conexión TCP (con flag FIN)
- pass: seguir procesando las demás reglas

## Sintaxis de Reglas (SecRule)

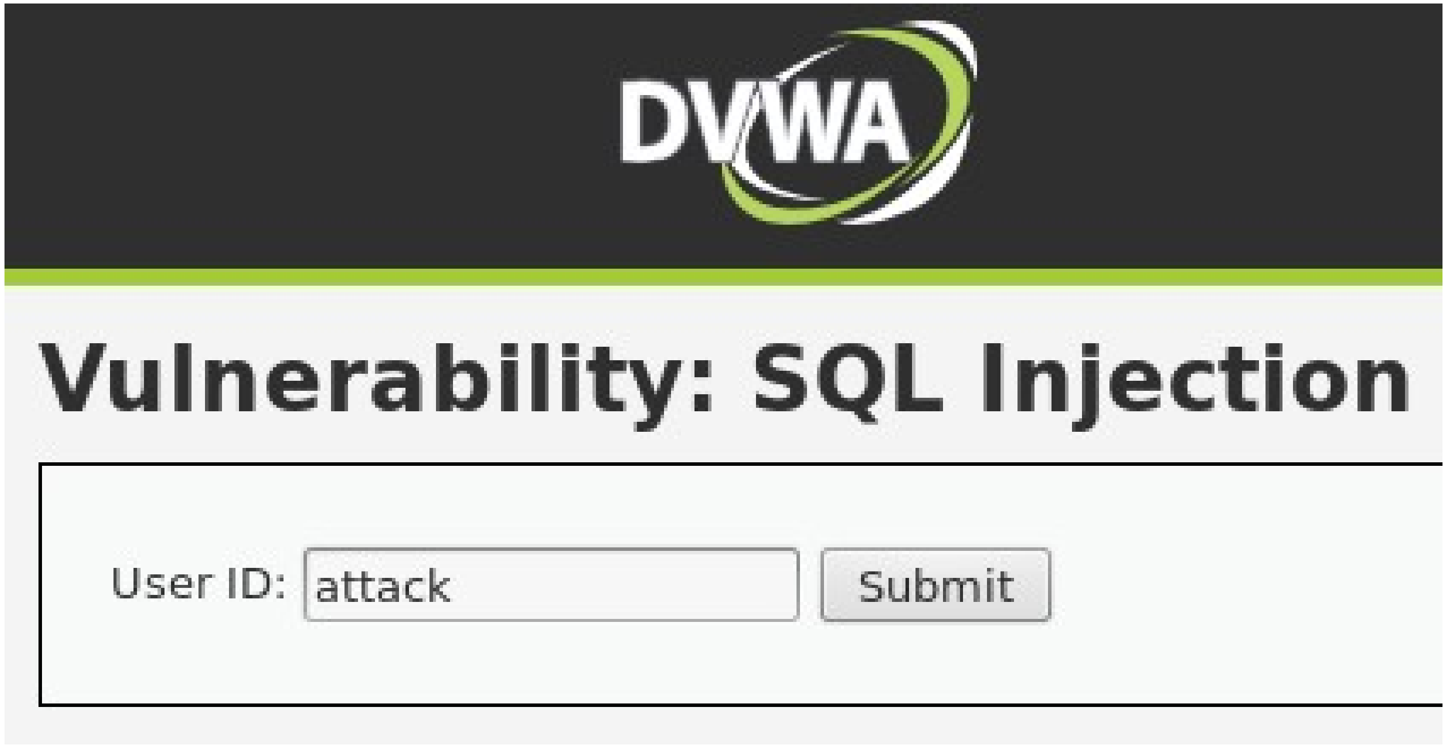
```
SecRule VARIABLES OPERATOR [ACTIONS]

SecRule ARGS "attack" "phase:1,log,deny,id:1"
```

## Habilitar / Deshabilitar

```
a2enmod security2
```

```
apache2ctl restart
```

## Sintaxis de Reglas (SecRule)

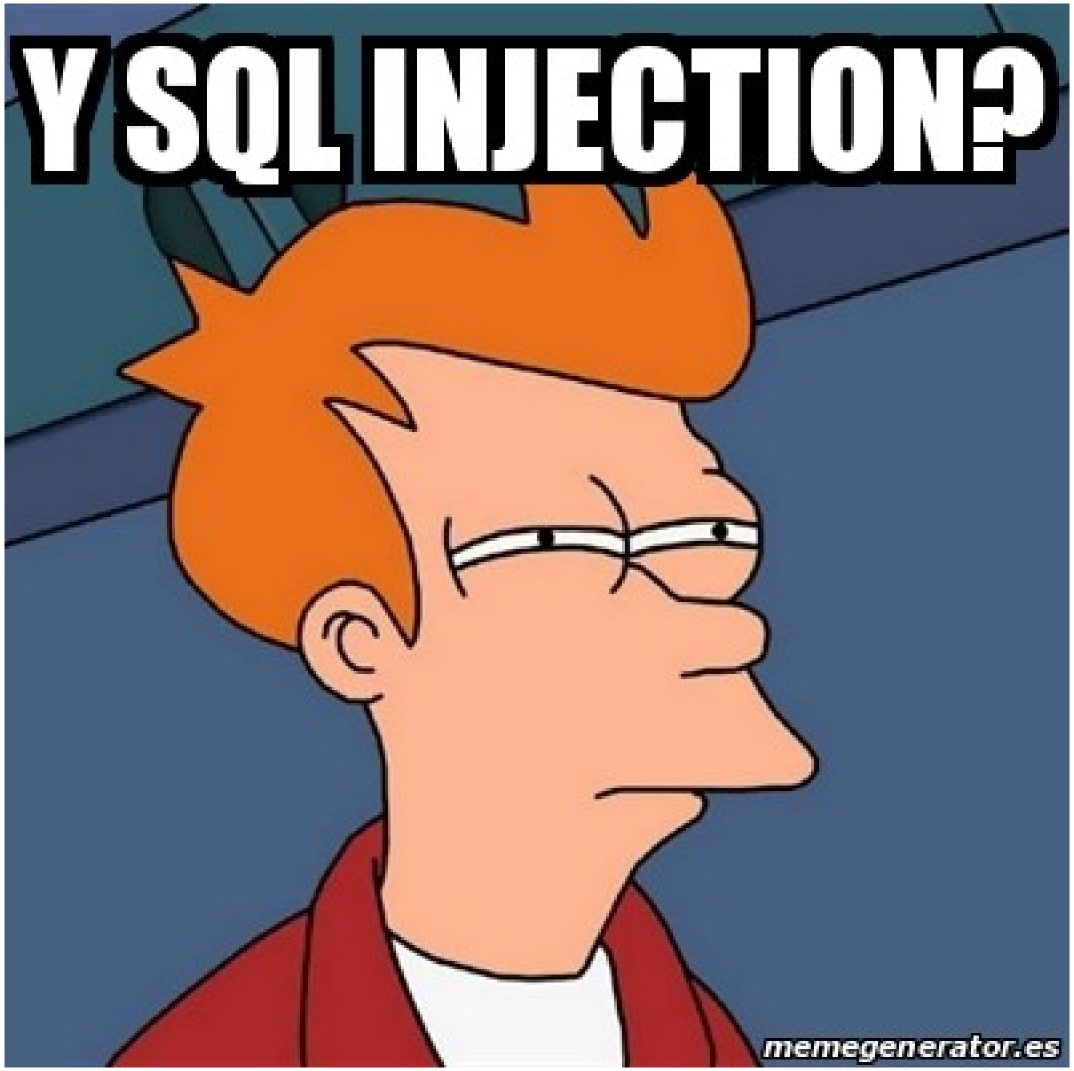


## Definición de Excepciones

- SecRuleRemoveByMsg
- SecRuleRemoveByTag
- SecRuleRemoveById

## Definición de Excepciones

```
SecRuleRemoveByID 700001

SecRuleRemoveByMsg “Host header is a numeric IP address”
```


Y SQL INJECTION?
memegenerator.es

## Regla de Ejemplo SQLi

```
SecRule ARGS "(or|and|not|like)"
   "phase:request,
   block, id:700001,
   msg:'SQL Injection Attack: Keywords',
   severity:'WARNING'"
```

Trabajar sobre argumentos (GET y POST)

## Regla de Ejemplo SQLi

```
SecRule ARGS "(or|and|not|like)"
   "phase:request,
   block, id:700001,
   msg:'SQL Injection Attack: Keywords',
   severity:'WARNING'"
```

Concordar expresión regular

## Regla de Ejemplo SQLi

```
SecRule ARGS "(or|and|not|like)"
   "phase:request,
   block, id:700001,
   msg:' SQL Injection Attack: Keywords' ,
   severity:' WARNING' "
```

Trabajar sobre la fase de petición (Fase 2)

## Regla de Ejemplo SQLi

Acción a tomar

```
SecRule ARGS "(or|and|not|like)"
   "phase:request,
   block, id:700001,
   msg:'SQL Injection Attack: Keywords',
   severity:'WARNING'"
```

## Regla de Ejemplo SQLi

ID de la regla

```
SecRule ARGS "(or|and|not|like)"
   "phase:request,
   block, id:700001,
   msg:' SQL Injection Attack: Keywords' ,
   severity:' WARNING' "
```

## Regla de Ejemplo SQLi

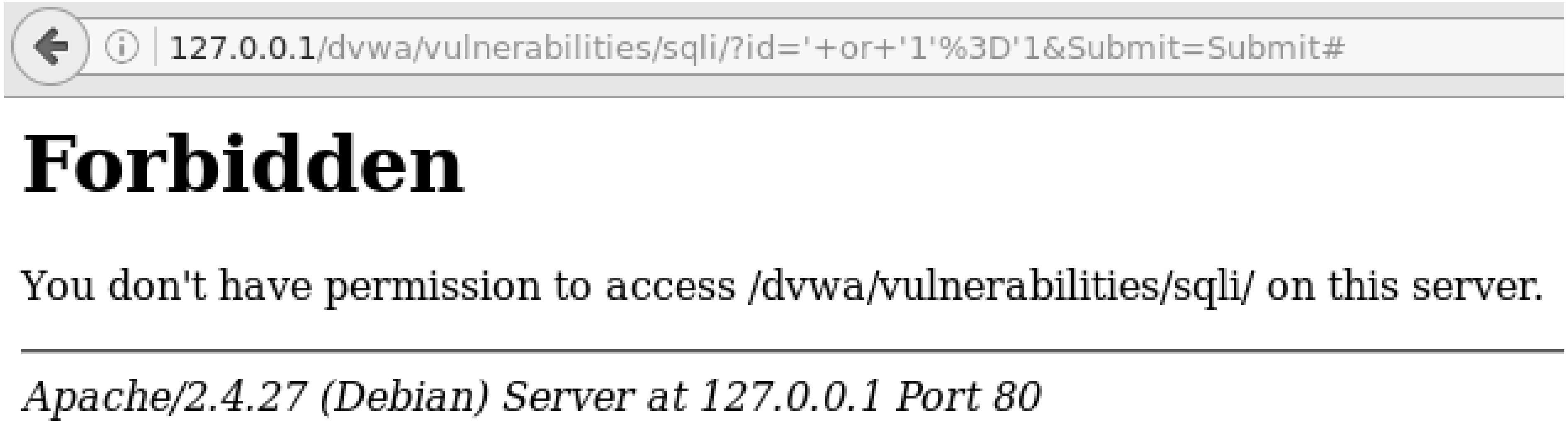
Mensaje de la regla

```
SecRule ARGS "(or|and|not|like)"
  "phase:request,
  block, id:700001,
  msg:'SQL Injection Attack: Keywords',
  severity:'WARNING'"
```

## Regla de Ejemplo SQLi

Severidad de la regla

```
SecRule ARGS "(or|and|not|like)"
   "phase:request,
   block, id:700001,
   msg:'SQL Injection Attack: Keywords',
   severity:'WARNING'"
```



```
127.0.0.1/dvwa/vulnerabilities/sqli/?id='+or+'1'%3D'1&Submit=Submit#
```

```
Forbidden

You don't have permission to access /dvwa/vulnerabilities/sqli/ on this server.

Apache/2.4.27 (Debian) Server at 127.0.0.1 Port 80
```

## Logs

Lo que hizo
ModSecurity

```
ModSecurity: Access denied with code 403 (phase 2).
Pattern match "(or|and|not|like)" at ARGS:id.
[file "/etc/modsecurity/custom/securetia.conf"]
[line "38"] [id "700001"] [msg "SQL Injection Attack:
Keywords"]
[severity "WARNING"] [hostname "127.0.0.1"]
[uri "/dvwa/vulnerabilities/sqli/"]
[unique_id "WcLY8X8AAQEAADgpuZ8AAAAC"]
```

## Logs

Por qué lo hizo

```
ModSecurity: Access denied with code 403 (phase 2).
Pattern match "(or|and|not|like)" at ARGS:id.
[file "/etc/modsecurity/custom/securetia.conf"]
[line "38"] [id "700001"] [msg "SQL Injection Attack:
Keywords"]
[severity "WARNING"] [hostname "127.0.0.1"]
[uri "/dvwa/vulnerabilities/sqli/"]
[unique_id "WcLY8X8AAQEAADgpuZ8AAAAC"]
```

## Logs

Donde dice que lo tenía que hacer

```
ModSecurity: Access denied with code 403 (phase 2).
Pattern match "(or|and|not|like)" at ARGS:id.
[file "/etc/modsecurity/custom/securetia.conf"]
[line "38"] [id "700001"] [msg "SQL Injection Attack:
Keywords"]
[severity "WARNING"] [hostname "127.0.0.1"]
[uri "/dvwa/vulnerabilities/sqli/"]
[unique_id "WcLY8X8AAQEAADgpuZ8AAAAC"]
```

## Logs

ID y mensaje de la regla en cuestión

```
ModSecurity: Access denied with code 403 (phase 2).
Pattern match "(or|and|not|like)" at ARGS:id.
[file "/etc/modsecurity/custom/securetia.conf"]
[line "38"] [id "700001"] [msg "SQL Injection Attack:
Keywords"]
[severity "WARNING"] [hostname "127.0.0.1"]
[uri "/dvwa/vulnerabilities/sqli/"]
[unique_id "WcLY8X8AAQEAADgpuZ8AAAAC"]
```

## Logs

Severidad de la regla

```
ModSecurity: Access denied with code 403 (phase 2).
Pattern match "(or|and|not|like)" at ARGS:id.
[file "/etc/modsecurity/custom/securetia.conf"]
[line "38"] [id "700001"] [msg "SQL Injection Attack:
Keywords"]
[severity "WARNING"] [hostname "127.0.0.1"]
[uri "/dvwa/vulnerabilities/sqli/"]
[unique_id "WcLY8X8AAQEAADgpuZ8AAAAC"]
```

## Logs

Hostname del cliente

```
ModSecurity: Access denied with code 403 (phase 2).
Pattern match "(or|and|not|like)" at ARGS:id.
[file "/etc/modsecurity/custom/securetia.conf"]
[line "38"] [id "700001"] [msg "SQL Injection Attack:
Keywords"]
[severity "WARNING"] [hostname "127.0.0.1"]
[uri "/dvwa/vulnerabilities/sqli/"]
[unique_id "WcLY8X8AAQEAADgpuZ8AAAAC"]
```

## Logs

URI solicitada

```
ModSecurity: Access denied with code 403 (phase 2).
Pattern match "(or|and|not|like)" at ARGS:id.
[file "/etc/modsecurity/custom/securetia.conf"]
[line "38"] [id "700001"] [msg "SQL Injection Attack:
Keywords"]
[severity "WARNING"] [hostname "127.0.0.1"]
[uri "/dvwa/vulnerabilities/sqli/"]
[unique_id "WcLY8X8AAQEAADgpuZ8AAAAC"]
```

## Logs

ID del log detallado
de peticiones

```
ModSecurity: Access denied with code 403 (phase 2).
Pattern match "(or|and|not|like)" at ARGS:id.
[file "/etc/modsecurity/custom/securetia.conf"]
[line "38"] [id "700001"] [msg "SQL Injection Attack:
Keywords"]
[severity "WARNING"] [hostname "127.0.0.1"]
[uri "/dvwa/vulnerabilities/sqli/"]
[unique_id "WcLY8X8AAQEAADgpuZ8AAAAC"]
```

## Audit Logs

A: Encabezado de la entrada de log
B: Request headers
C: Request body
D: No implementado
E: Intermediary response body
F: Response headers
G: No implementado
H: Audit log trailer
I: Igual a C, pero sin los archivos
J: Archivos multipart/form-data encoding
K: Reglas que concordaron
Z: Indica el final de la entrada del log

## Audit Logs

```
--3e65ef33-A--
[26/Sep/2017:19:00:05] WcrN5X8-@55oAAAAE ::1 57776 ::1 80

--3e65ef33-B--
GET /dvwa/vulnerabilities/sqli/?id=%27+or+%271%27+%3D+...Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:45.0) ...
Accept: text/html,application/xhtml+xml ...
Accept-Encoding: gzip, deflate
Cookie: security=low; PHPSESSID=9up6j61hc9d6cp23svrptm3tg2
Connection: keep-alive
Cache-Control: max-age=0
```

## Audit Logs

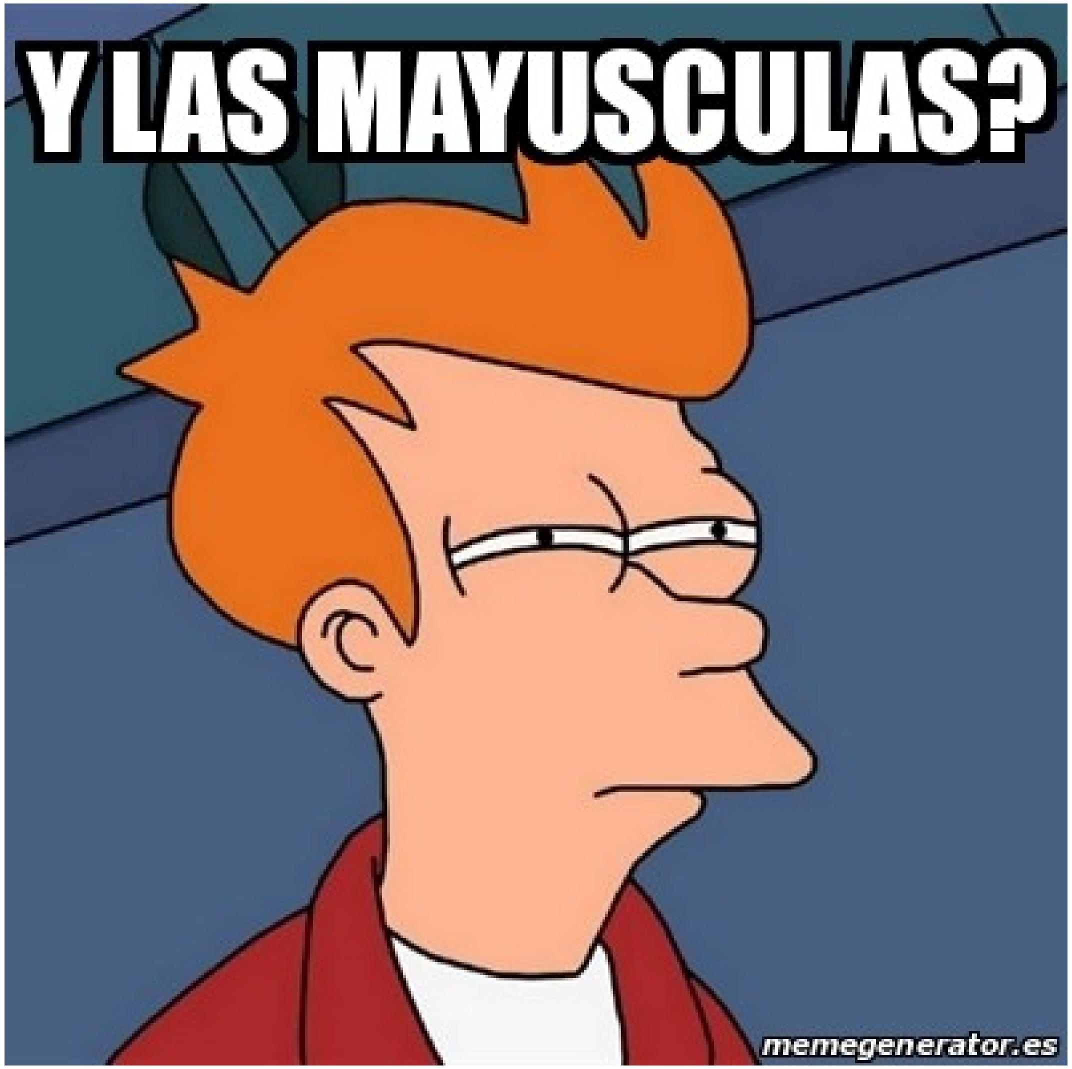
```
--3e65ef33-F--
HTTP/1.1 403 Forbidden
Content-Length: 310
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=iso-8859-1
```

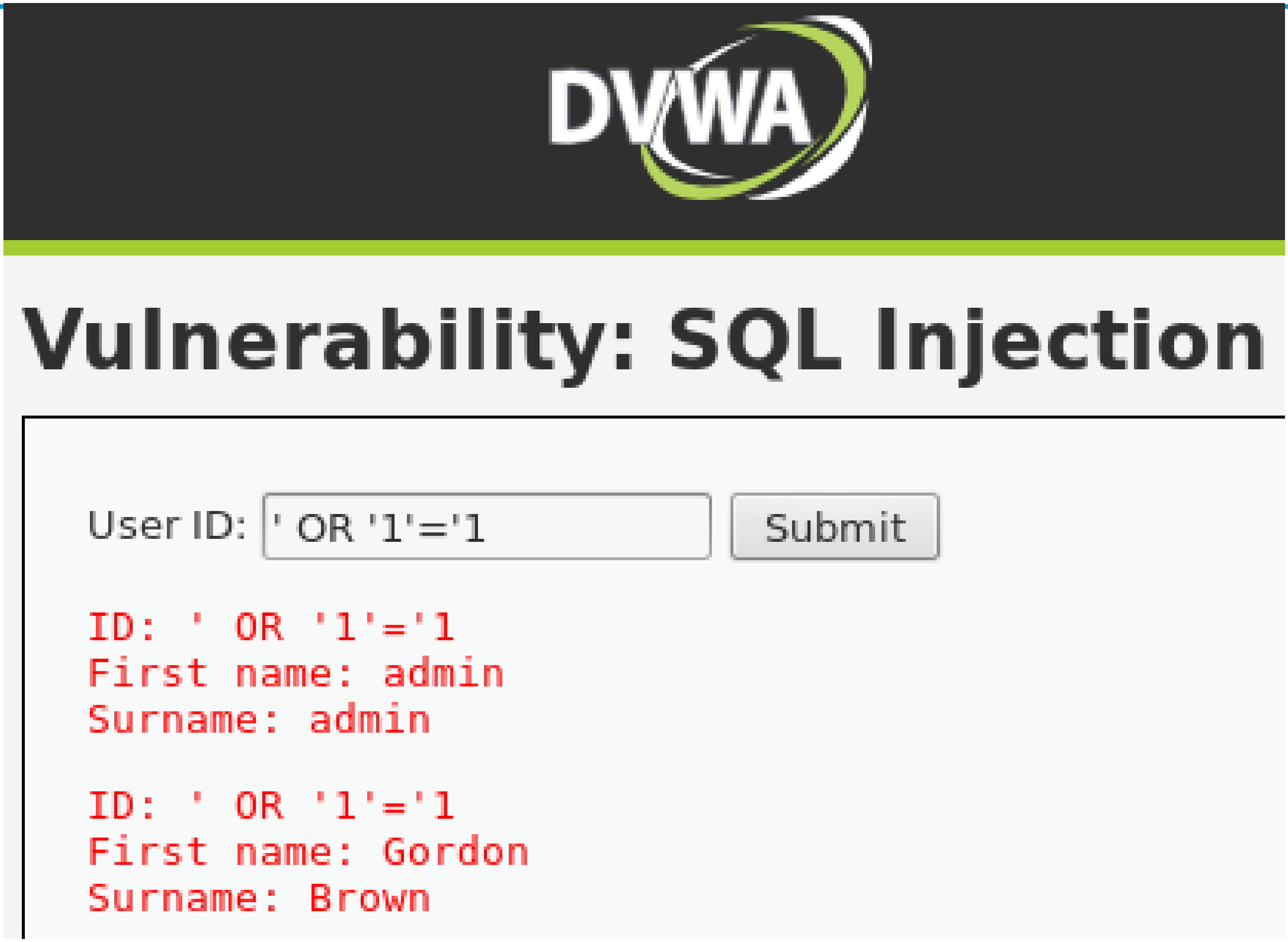
## Audit Logs

```
--3e65ef33-E--
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
<p>You don' t have permission to access /dvwa/vulnerabilities/sqli/...
</p>
<hr>
<address>Apache/2.4.27 (Debian) Server at localhost Port
80</address>
</body></html>
```

## Audit Logs

```
--3e65ef33-H--
Message: Access denied with code 403 (phase 2). Pattern match "(or|
and|not|like)" at ARGS:id. [file
"/etc/modsecurity/custom/securetia.conf"] [line "32"] [id "700001"]
[msg "SQL Injection Attack: Keywords"] [severity "WARNING"]
Action: Intercepted (phase 2)
Producer: ModSecurity for Apache/2.9.1
(http://www.modsecurity.org/).
Server: Apache/2.4.27 (Debian)
Engine-Mode: "ENABLED"

--3e65ef33-Z--
```


Y LAS MAYUSCULAS?
memegenerator.es



Vulnerability: SQL Injection

```
ID: ' OR '1'='1
First name: admin
Surname: admin

ID: ' OR '1'='1
First name: Gordon
Surname: Brown
```

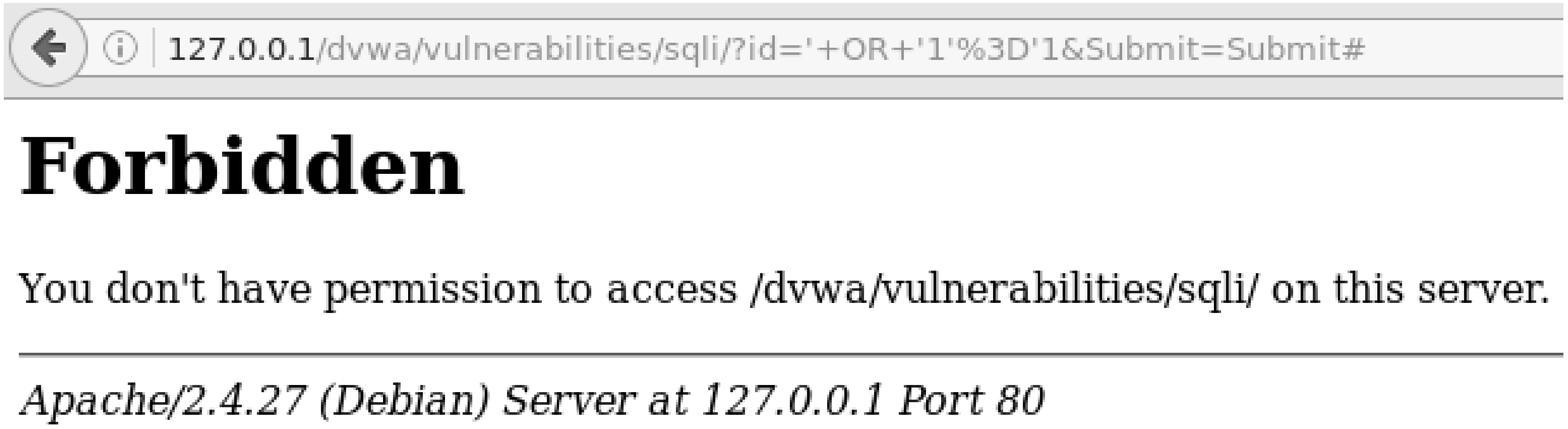
## Transformaciones Importantes

- base64Decode
- sqlHexDecode
- cmdLine
- compressWhitespace
- cssDecode
- hexDecode
- htmlEntityDecode
- length
- lowercase
- md5

## Regla de Ejemplo

Transformaciones
a aplicar

```
SecRule ARGS "(or|and|not|like)"
   "phase:request,
   t:lowercase,
   block, id:700002,
   msg:'SQL Injection Attack: Keywords',
   severity:'WARNING'"
```



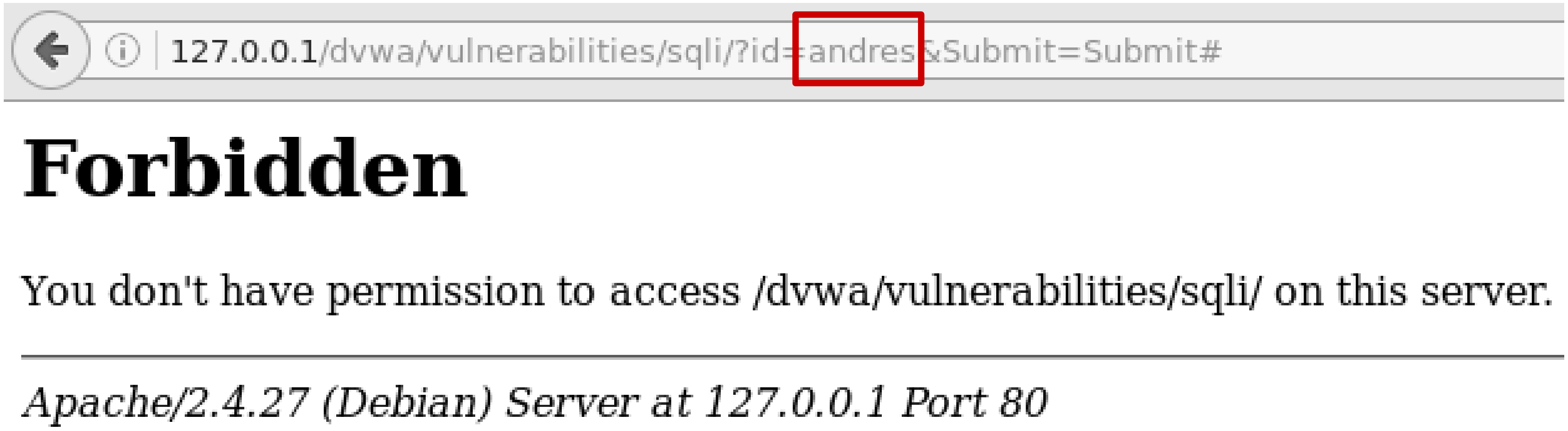
```
127.0.0.1/dvwa/vulnerabilities/sqli/?id='+OR+'1'%3D'1&Submit=Submit#
```

Forbidden

You don't have permission to access /dvwa/vulnerabilities/sqli/ on this server.

*Apache/2.4.27 (Debian) Server at 127.0.0.1 Port 80*

Y ANDRES?
memegenerator.es



```
127.0.0.1/dvwa/vulnerabilities/sqli/?id=andres&Submit=Submit#
```

Forbidden

You don't have permission to access /dvwa/vulnerabilities/sqli/ on this server.

*Apache/2.4.27 (Debian) Server at 127.0.0.1 Port 80*

## Log

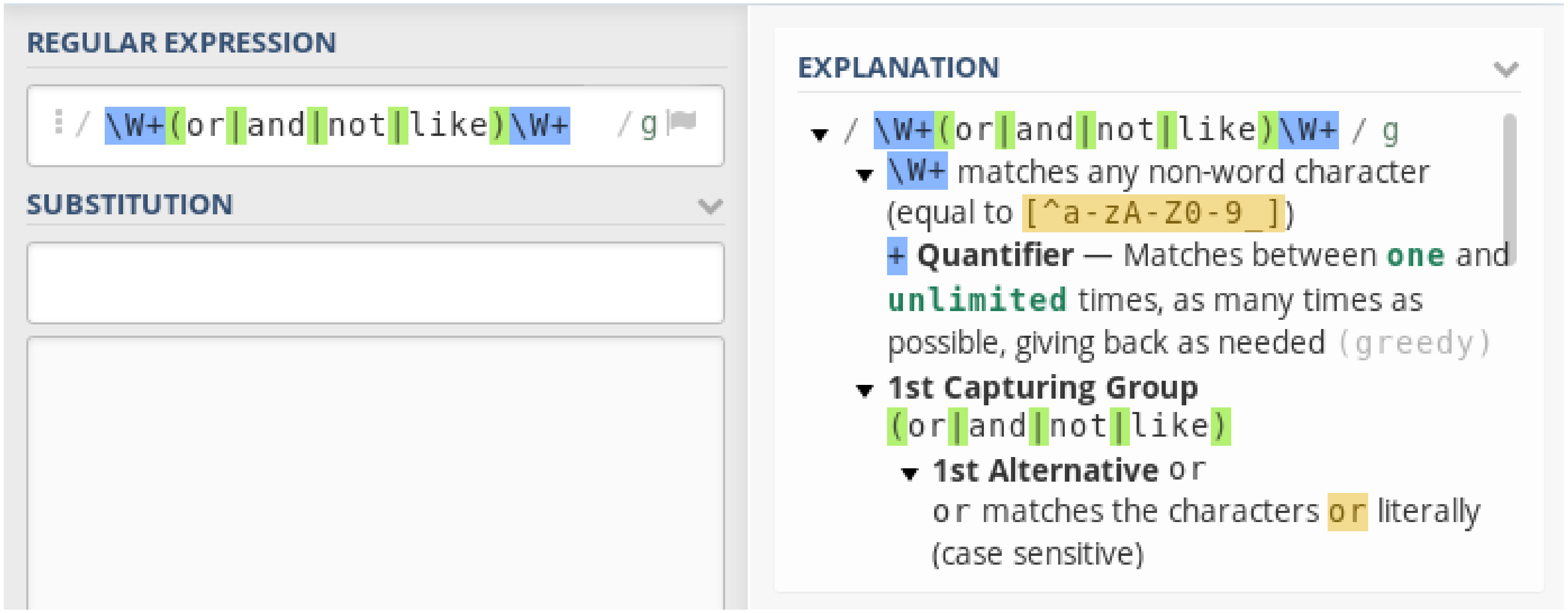
Ahí está el problema

```
ModSecurity: Access denied with code 403 (phase 2).
Pattern match "(or|and|not|like)" at ARGS:id.
[file "/etc/modsecurity/custom/securetia.conf"] [line "68"]
[id "700002"] [msg "SQL Injection Attack: Keywords"]
[severity "WARNING"] [hostname "127.0.0.1"]
[uri "/dvwa/vulnerabilities/sqli/"]
[unique_id "WcLp4n8AAQEAADqAlEgAAAAC"]
```

## Regla Mejorada

Regex más específica

```
SecRule ARGS "\W+(or|and|not|like)\W+"
   "phase:request,
   t:lowercase,
   block, id:700003,
   msg:'SQL Injection Attack: Keywords',
   severity:'WARNING'"
```


REGULAR EXPRESSION

```
/ \W+(or|and|not|like)\W+ /g
```

SUBSTITUTION

EXPLANATION

- / \W+(or|and|not|like)\W+ / g
  - \W+ matches any non-word character (equal to [^a-zA-Z0-9_])
    + Quantifier — Matches between one and unlimited times, as many times as possible, giving back as needed (greedy)
  - 1st Capturing Group (or|and|not|like)
    - 1st Alternative or
      or matches the characters or literally (case sensitive)

**https://regex101.com**

## Reglas Encadenadas

## Reglas Encadenadas

Acá está la magia

```
SecRule REQUEST_URI "@EndsWith /vulnerabilities/exec/"
    phase:1,chain,deny,t:none,id:700008
 SecRule REQUEST_HEADERS:User-Agent "@contains Firefox"
```

## Reglas Encadenadas

- Acciones disruptivas sólo en la 1er regla
- Acciones disruptivas se ejecutan si todas las reglas concuerdan
- Acciones no-disruptivas se pueden usar en cualquier regla
- Acciones no-disruptivas se ejecutan si esa regla concuerda
- Acciones de metadatos (id, rev, msg) sólo en la 1er regla

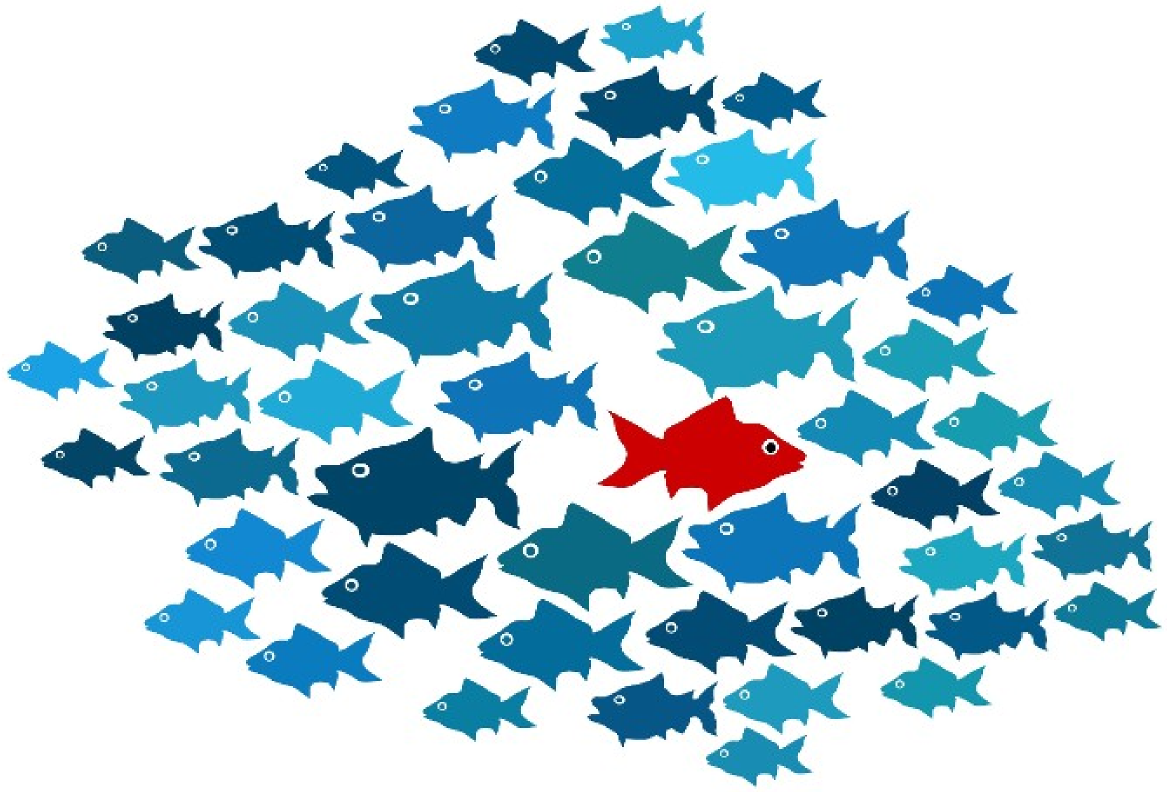
## Configuraciones al Vuelo

## Configuraciones al Vuelo

Acá está la magia

```
SecRule ARGS:dev "^true$"
     "phase:1,
     pass,
     id:700008,
     ctl:ruleEngine=DetectionOnly"
```

## Filtrado Basado en Anomalías



## Filtrado Basado en Anomalías

```
SecAction
 "id:800000,
 nolog, pass,
 setvar:tx.critical_anomaly_score=5,
 setvar:tx.error_anomaly_score=4,
 setvar:tx.warning_anomaly_score=3,
 setvar:tx.notice_anomaly_score=2,
 setvar:tx.anomaly_score_threshold=5"
```

Cuánto vale cada tipo de regla

## Filtrado Basado en Anomalías

```
SecAction
 "id:800000,
 nolog, pass,
 setvar:tx.critical_anomaly_score=5,
 setvar:tx.error_anomaly_score=4,
 setvar:tx.warning_anomaly_score=3,
 setvar:tx.notice_anomaly_score=2,
 setvar:tx.anomaly_score_threshold=5"
```

Cuánto puedo soportar antes de bloquear la petición

## Filtrado Basado en Anomalías

```
SecRule ARGS "\W+(or|and|not|like)\W+"
   "phase:request,
   t:lowercase,
   block, id:700004,
   msg:'SQL Injection Attack: Keywords',
   severity:'WARNING',
   setvar:tx.anomaly_score=+%{tx.warning_anomaly_score}"
```

Aumento la varible en base a lo que pesa un warning

# Filtrado Basado en Anomalías

```
SecRule ARGS "(\"|'|´|`)"
    "phase:request,
    t:lowercase,
    block, id:700005,
    msg:'SQL Injection Attack: Quotes',
    severity:'WARNING',
    setvar:tx.anomaly_score=+%{tx.warning_anomaly_score}"
```

Aumento la varible en base a lo que pesa un warning

## Filtrado Basado en Anomalías

```
SecRule TX:ANOMALY_SCORE "@ge
{tx.anomaly_score_threshold}"
   "msg:' Inbound Anomaly Score Exceeded
   (Total Score: %{TX.ANOMALY_SCORE})' ,
   severity:CRITICAL,
   phase:request,
   id:799999,
   deny, log"
```

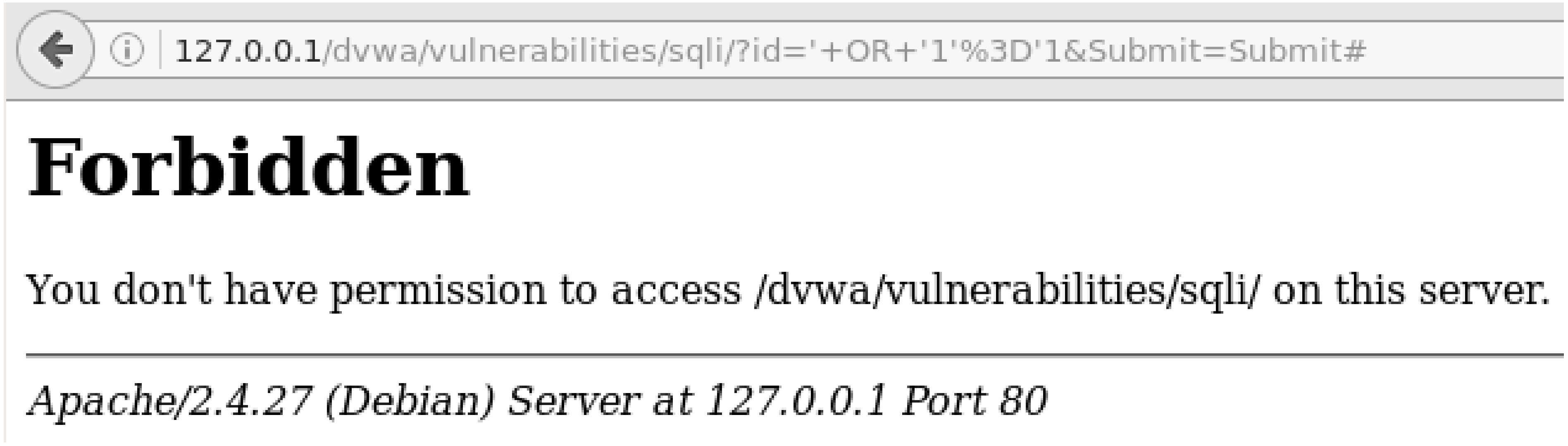
Trabajo sobre la variable TX:ANOMALY_SCORE

## Filtrado Basado en Anomalías

```
SecRule TX:ANOMALY_SCORE "@ge
{tx.anomaly_score_threshold}"
   "msg:' Inbound Anomaly Score Exceeded
   (Total Score: %{TX.ANOMALY_SCORE})' ,
   severity:CRITICAL,
   phase:request,
   id:799999,
   deny, log"
```

verifico si es igual o mayor que tx.anomaly_score

## Filtrado Basado en Anomalías



```
127.0.0.1/dvwa/vulnerabilities/sqli/?id='+OR+'1'%3D'1&Submit=Submit#
```

**Forbidden**

You don't have permission to access /dvwa/vulnerabilities/sqli/ on this server.

*Apache/2.4.27 (Debian) Server at 127.0.0.1 Port 80*

## Filtrado Basado en Anomalías

La regla no bloquea.
Alerta y aumenta el
nivel de anomalía.

```
ModSecurity: Warning.
Pattern match "\\\\W+(or|and|not|like)\\\\W+" at ARGS:id.
[file "/etc/modsecurity/custom/securetia.conf"] [line "49"]
[id "700004"] [msg "SQL Injection Attack: Keywords"]
[severity "WARNING"] [hostname "127.0.0.1"]
[uri "/dvwa/vulnerabilities/sqli/"]
[unique_id "WcL-a38AAQEAADrWE3YAAAAC"]
```

## Filtrado Basado en Anomalías

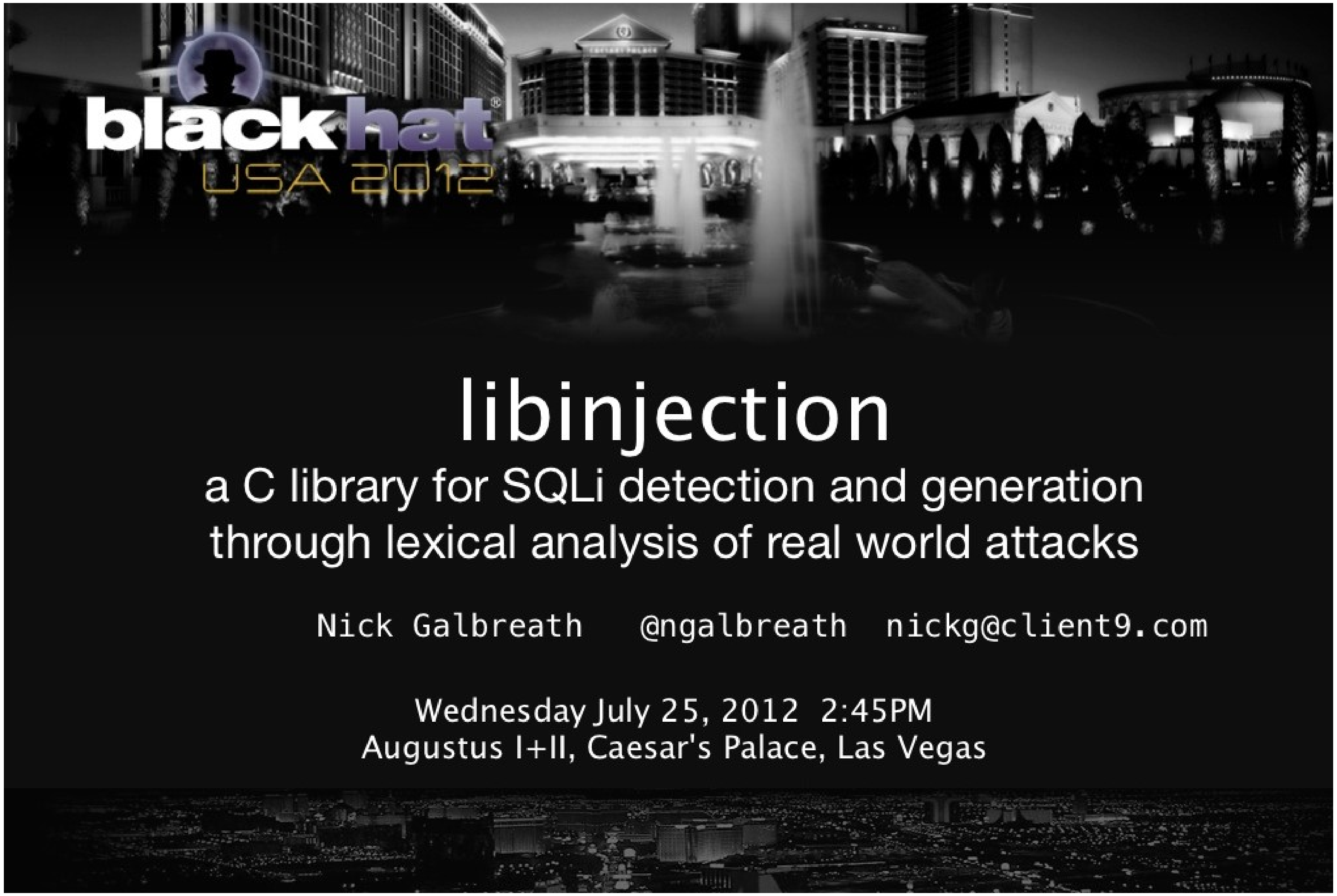
La regla no bloquea. Alerta y aumenta el nivel de anomalía.

```
ModSecurity: Warning.
Pattern match "(\\"|'|\\xc2\\xb4|`)" at ARGS:id.
[file "/etc/modsecurity/custom/securetia.conf"] [line "62"]
[id "700005"] [msg "SQL Injection Attack: Quotes"]
[severity "WARNING"] [hostname "127.0.0.1"]
[uri "/dvwa/vulnerabilities/sqli/"]
[unique_id "WcL-a38AAQEAADrWE3YAAAAC"]
```

## Filtrado Basado en Anomalías

Bloquea por exceder
el umbral definido

```
ModSecurity: Access denied with code 403 (phase 2).
Operator GE matched 4 at TX:anomaly_score.
[file "/etc/modsecurity/custom/securetia.conf"] [line "74"]
[id "799999"]
[msg "Inbound Anomaly Score Exceeded (Total Score: 6)"]
[severity "CRITICAL"] [hostname "127.0.0.1"]
[uri "/dvwa/vulnerabilities/sqli/"]
[unique_id "WcL-a38AAQEAADrWE3YAAAAC"]
```



libinjection

a C library for SQLi detection and generation through lexical analysis of real world attacks

Nick Galbreath @ngalbreath nickg@client9.com

Wednesday July 25, 2012 2:45PM
Augustus I+II, Caesar's Palace, Las Vegas

black hat
USA 2012

## Filtrado Basado en LibInjection

Ahí está toda la magia

```
SecRule ARGS "@detectSQLi"
    "phase:request,
    block,
    msg:'SQL Injection Attack: LibInjection',
    id:700006,
    severity:'CRITICAL',
    setvar:tx.anomaly_score=+%{tx.critical_anomaly_score}"
```

## OWASP Core Rule Set (CRS)

- Protección genérica contra ataques comunes
- Detección de anomalías (ej: Content-Lenght no numérico)
- Proyecto de código abierto y gratuito
- Reglas bien comentadas