Generación automática de reglas de seguridad en base a registros de sistema

Asier Salueña

Directores:
Ricardo J. Rodríguez, Víctor Pérez
Ponente:
Raquel Trillo

Departamento de Informática e Ingeniería de Sistemas
Escuela de Ingeniería y Arquitectura
Universidad de Zaragoza

Diciembre 2018

Universidad Zaragoza
1542

Índice

1. Introducción
2. Sistema propuesto
3. Ataques contemplados
4. Arquitectura
   - ELK
   - Detección de ataques
     - Clasificador
     - Sistema de reglas
   - Generación de reglas
5. Caso de estudio
6. Conclusiones

Universidad Zaragoza
1542

Índice

1. Introducción
2. Sistema propuesto
3. Ataques contemplados
4. Arquitectura
   - ELK
   - Detección de ataques
     - Clasificador
     - Sistema de reglas
   - Generación de reglas
5. Caso de estudio
6. Conclusiones

Introducción

- **Incremento de las amenazas en ciberseguridad**
  - 2015: Más de 250K denuncias por delitos informáticos gestionados por el FBI
  - **Pérdidas valoradas en más de US$ 1000M**
  - En todo el mundo $\approx 400000M$
- **Tendencia actual hacia la Inteligencia Artificial y el aprendizaje automático**
  - Ejemplo: Proyecto AI^2 (PatternEx + MIT)
- Soluciones empresariales tienen un **coste elevado**

Índice

1. Introducción
2. Sistema propuesto
3. Ataques contemplados
4. Arquitectura
   - ELK
   - Detección de ataques
     - Clasificador
     - Sistema de reglas
   - Generación de reglas
5. Caso de estudio
6. Conclusiones

Sistema propuesto

- **Determinar el comportamiento normal de una aplicación web**
- En base a sus registros de sistema **desarrollar un analizador de comportamientos**
- **Correlación de eventos anómalos**

Sistema propuesto

- **Determinar el comportamiento normal de una aplicación web**
- En base a sus registros de sistema **desarrollar un analizador de comportamientos**
- **Correlación de eventos anómalos**

Problemas

- Determinar el comportamiento normal en una aplicación web y compararlo con el del usuario $\rightarrow$ **Problema no trivial**
- Sistema cerrado, **difícil de extender** a otras aplicaciones
- **Falta de información** en los logs de sistema

Solución final

Desarrollo de un sistema que **clasifica las peticiones de manera automática en base a los registros de sistema**

1. Recoge las peticiones recibidas
2. Analiza y clasifica
3. Crea regla de seguridad en consecuencia

Índice

1. Introducción
2. Sistema propuesto
3. Ataques contemplados
4. Arquitectura
   - ELK
   - Detección de ataques
     - Clasificador
     - Sistema de reglas
   - Generación de reglas
5. Caso de estudio
6. Conclusiones

Ataques contemplados

Se han estudiado dos ataques, debido a que son los dos **tipos de ataques más habituales:**

- **DoS: Ataques de denegación de servicio**

Herramienta	Tipo de Ataque	Anónima
LOIC	*Packet Flooding Attack, Request flooding attack*	NO
HOIC	*Request flooding attack*	SI
hping	*Reflection-Based flooding attack*	SI
Slowloris	*Slow request/response attack*	SI
R u Dead Yet? (R.U.D.Y.)	*Slow request/response attack*	NO
#Refref	*Session flooding attack*	SI
HULK	*Asymmetric attack, Slow request/response attack*	SI
DDOSIM-Layer7	*Protocol exploitation flooding attack*	SI

- **SQLi: Ataques de inyección sql**
  - sqlmap
  - BSQL Hacker
  - SQLNinja

Ataques contemplados

Se han estudiado dos ataques, debido a que son los dos **tipos de ataques más habituales:**

- **DoS: Ataques de denegación de servicio**

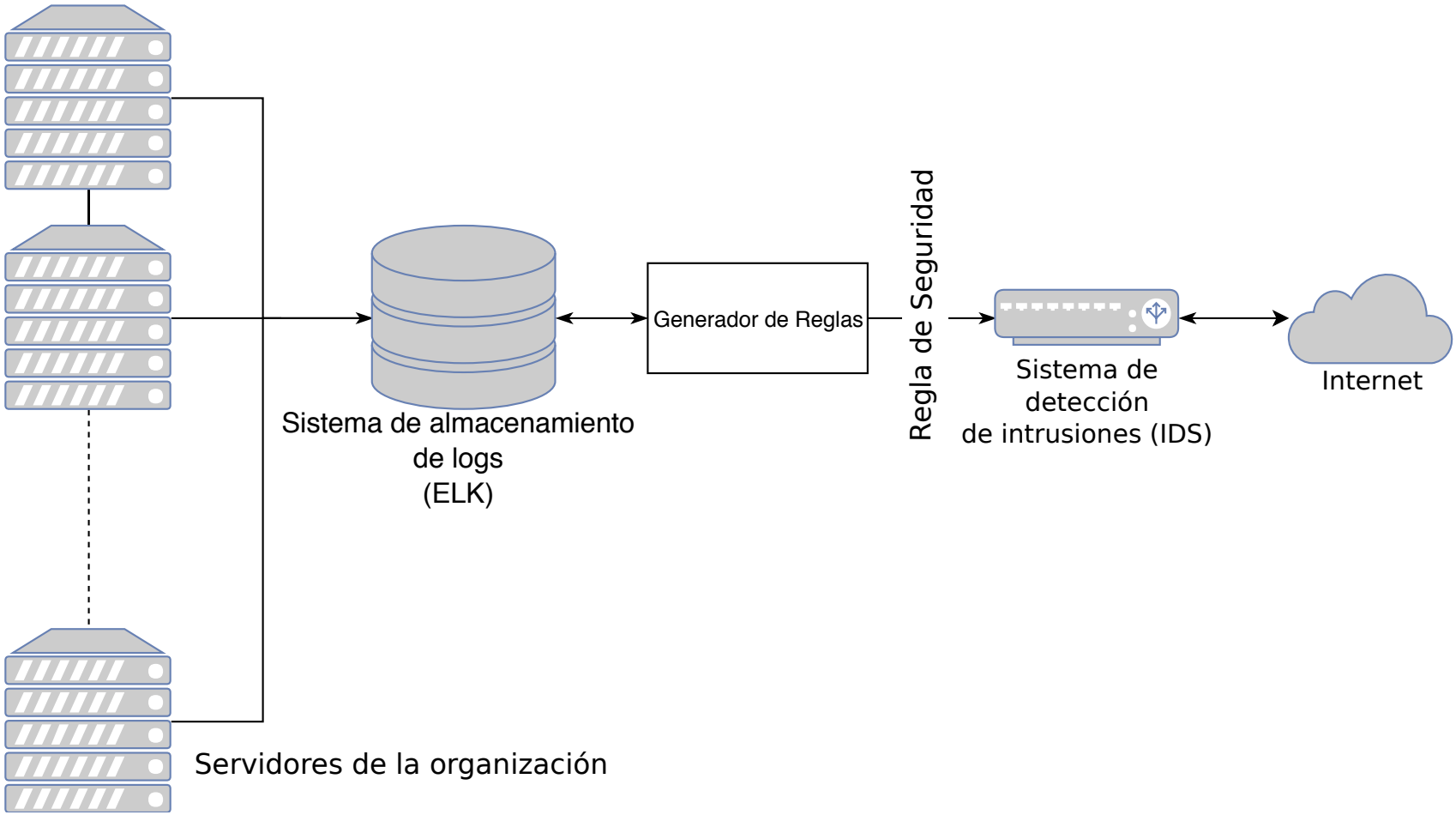
Herramienta	Tipo de Ataque	Anónima
LOIC	*Packet Flooding Attack, Request flooding attack*	NO
HOIC	*Request flooding attack*	SI
hping	*Reflection-Based flooding attack*	SI
Slowloris	*Slow request/response attack*	SI
R u Dead Yet? (R.U.D.Y.)	*Slow request/response attack*	NO
#Refref	*Session flooding attack*	SI
HULK	*Asymmetric attack, Slow request/response attack*	SI
DDOSIM-Layer7	*Protocol exploitation flooding attack*	SI

- **SQLi: Ataques de inyección sql**
  - **sqlmap**
  - BSQL Hacker
  - SQLNinja

Índice

1. Introducción
2. Sistema propuesto
3. Ataques contemplados
4. Arquitectura
   - ELK
   - Detección de ataques
     - Clasificador
     - Sistema de reglas
   - Generación de reglas
5. Caso de estudio
6. Conclusiones

Vista del sistema

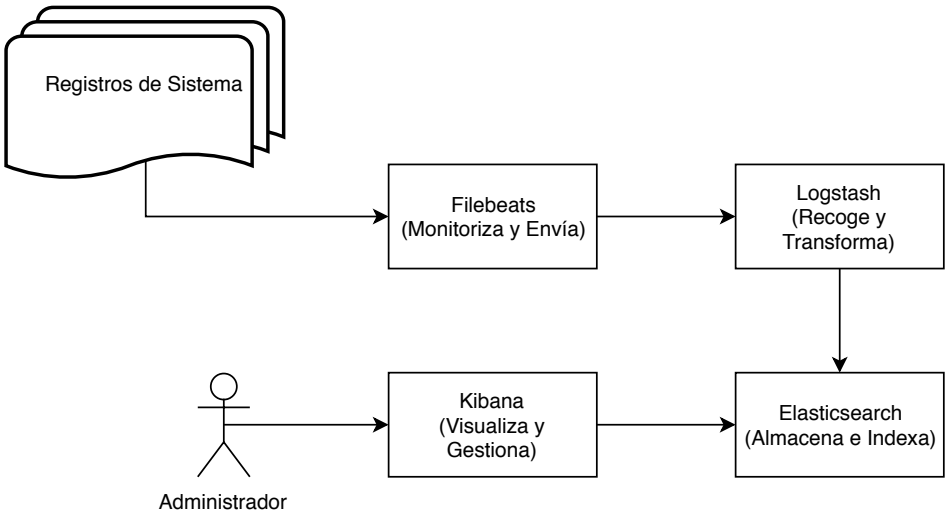


Índice

1. Introducción
2. Sistema propuesto
3. Ataques contemplados
4. Arquitectura
   - ELK
   - Detección de ataques
     - Clasificador
     - Sistema de reglas
   - Generación de reglas
5. Caso de estudio
6. Conclusiones

Sistema de almacenamiento de logs

ELK Stack



Sistema de normalización y almacenamiento centralizado de logs

- **Elasticsearch:** Motor de búsqueda y Análisis
- **Logstash:** Recopila datos y envía al destino deseado
- **Kibana:** Herramienta de exploración y visualización de datos

Índice

1. Introducción
2. Sistema propuesto
3. Ataques contemplados
4. Arquitectura
   - ELK
   - Detección de ataques
     - Clasificador
     - Sistema de reglas
   - Generación de reglas
5. Caso de estudio
6. Conclusiones

El sistema de detección de ataques

- **Clasificador**: Basado en campos de petición HTTP
- **Sistema de reglas**

Cabecera	Valor
Accept	text/html.application/xhtml+xm.../xml:q-0.9,*/*;q=0.8
Accept-Encoding	gzip, deflate
Accept-Language	es-ES,es;q-0.8,en-US;q-0.5,en:q-0.3
Connection	keep-alive
Cookie	https%3a%2f%2frrhh.unizar.es%2...dashboard; has_js-1
Host	www.unizar.es
If-None-Match	"1537345731-1"
Referer	"http://www.unizar.es/"
Upgrade-Insecure-Requests	1
User-Agent	Mozilla/5.0(X11, Linux x86_64...) ...Firefox/60.0

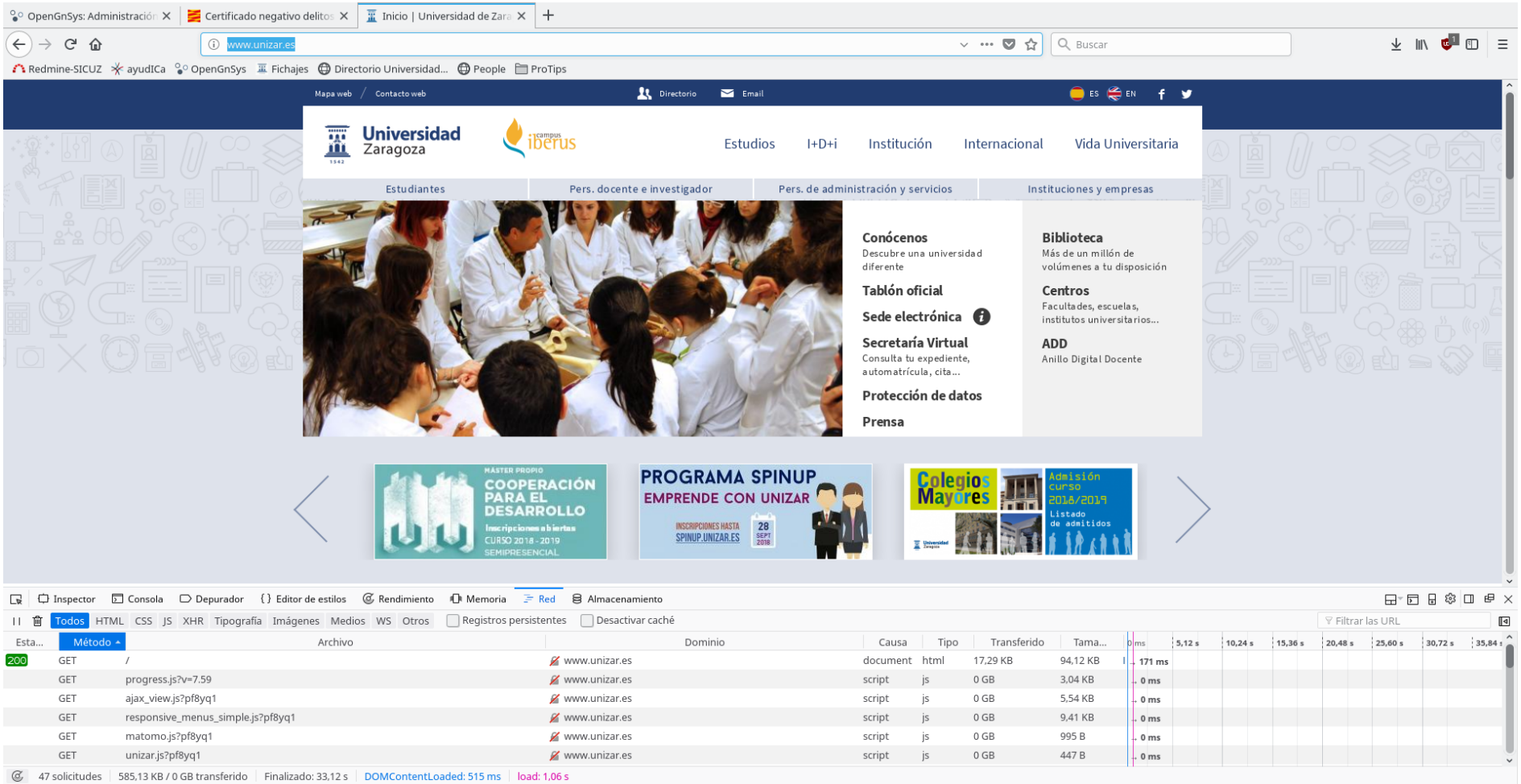
El sistema de detección de ataques

- **Clasificador**: Basado en campos de petición HTTP
- **Sistema de reglas**

Cabecera	Valor
Accept	text/html.application/xhtml+xm.../xml:q-0.9,*/*;q=0.8
Accept-Encoding	gzip, deflate
Accept-Language	es-ES,es;q-0.8,en-US;q-0.5,en:q-0.3
Connection	keep-alive
Cookie	https %3a %2f %2frrhh.unizar.es %2...dashboard; has_js-1
Host	www.unizar.es
If-None-Match	"1537345731-1"
Referer	"http://www.unizar.es/"
Upgrade-Insecure-Requests	1
User-Agent	Mozilla/5.0(X11, Linux x86_64...) ...Firefox/60.0

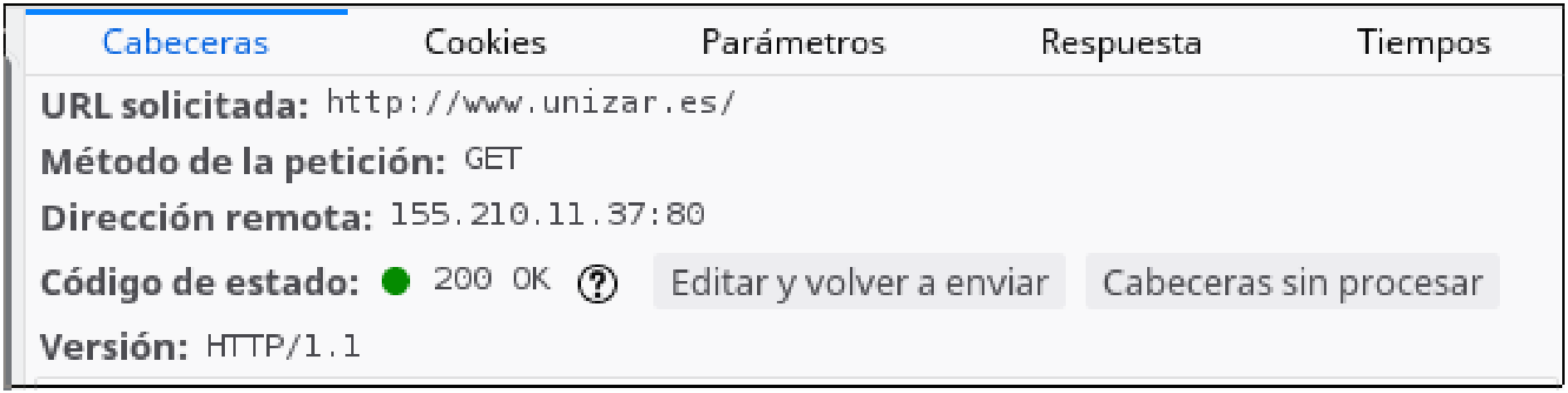
El sistema de detección de ataques

Ejemplo de una petición web



El sistema de detección de ataques

Ejemplo de una petición web



Detección de ataques - Clasificador

Sistema automático de clasificación que evalúa el campo Agent y Referrer de una petición HTTP

- **Clasificador Bayesiano Ingenuo**, clasificador probabilista basado en el teorema de Bayes que supone independencia entre los predictores
- **Algoritmo de validación cruzada**
- **Cuatro variaciones diferentes:**
  - Distribución Multinomial / Bernoulli
  - Bolsa de palabras / Bigramas

Detección de ataques - Sistema de reglas

- **Actúa tras la clasificación automática**
- Tiene en cuenta los resultados de la clasificación
- **Considera los campos Continent_Code, Country_Name, Response**
- Además, **cuenta las peticiones web realizadas desde una misma dirección IP**

Ponderación del sistema de reglas

Sistema de reglas	
Clasificación positiva del campo *Agent*	25 %
Clasificación positiva del campo *Referrer*	25 %
Resto de campos:	25 %
- *Continent_Code* $\neq$ *UE*	8.33 %
- *Country_Name* $\in$ {"*EEUU*","*Holanda*","*China*","*Brasil*","*Rusia*"}	8.33 %
- *Response* $\neq$ 200	8.33 %
+50 peticiones en 60 segundos	25 %

Tabla: Ponderaciones del sistema de reglas

Índice

1. Introducción
2. Sistema propuesto
3. Ataques contemplados
4. Arquitectura
   - ELK
   - Detección de ataques
     - Clasificador
     - Sistema de reglas
   - Generación de reglas
5. Caso de estudio
6. Conclusiones

Sistema de actuación

$$Acción = \begin{cases} Bloqueo, & si \quad r \geq 80\,\% \\ Alerta, & si \quad 75\,\% \leq r < 80\,\% \\ Normal, & si \quad r < 75\,\% \end{cases}$$

Índice

1. Introducción
2. Sistema propuesto
3. Ataques contemplados
4. Arquitectura
   - ELK
   - Detección de ataques
     - Clasificador
     - Sistema de reglas
   - Generación de reglas
5. Caso de estudio
6. Conclusiones

Caso de estudio

- Despliegue del sistema desarrollado sobre una **copia parcial de Moodle**
  - Plataforma web de apoyo docente usada por la Universidad de Zaragoza
- **Escenarios de ataque contemplados:**
  1. Sin el sistema de defensa propuesto
  2. Con el sistema de defensa propuesto

Caso de estudio

Entrenamiento del clasificador desplegado en la plataforma

Fase de entrenamiento:

- 65K ataques diferentes generados con las herramientas nombradas, 66 % DDoS y 33 % SQLi
  - 80 % datos de entrenamiento
  - 10 % datos de validación
  - 10 % datos de test

(FP: Falsos Positivos; FN: Falsos Negativos)

Variación	**FP**	**FN**
Bolsa palabras/Multinomial	4,49 %	6,28 %
Bolsa palabras/Bernoulli	4,51 %	2 %
Bigramas/Multinomial	4,49 %	6,28 %
Bigramas/Bernoulli	0 %	58,79 %

Tabla: Falsos positivos y falsos negativos para el campo *Agent*

Variación	**FP**	**FN**
Bolsa palabras/Multinomial	4,80 %	6,20 %
Bolsa palabras/Bernoulli	4,21 %	0,13 %
Bigramas/Multinomial	4,33 %	5,89 %
Bigramas/Bernoulli	0 %	66,49 %

Tabla: Falsos positivos y falsos negativos para el campo *Referrer*

(FP: Falsos Positivos; FN: Falsos Negativos)

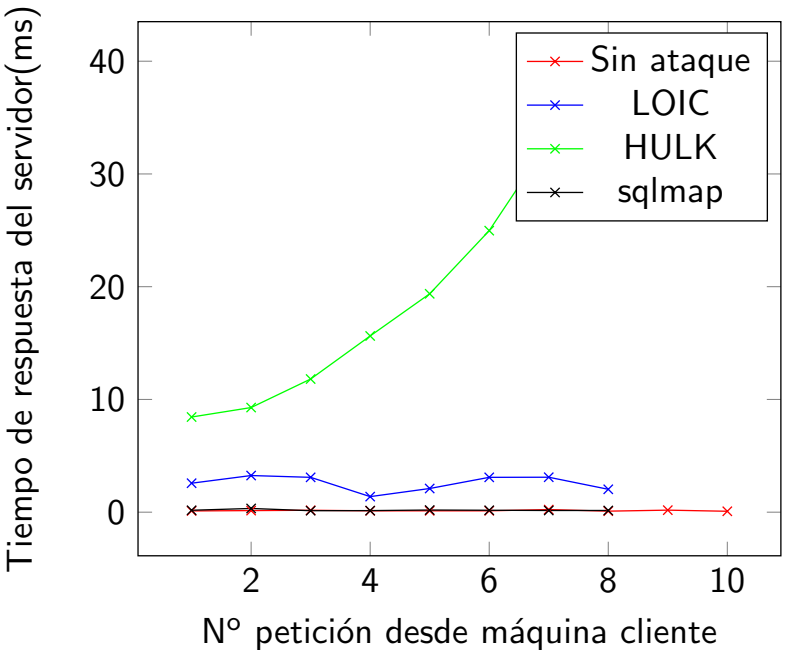
Variación	**FP**	**FN**
Bolsa palabras/Multinomial	4,49 %	6,28 %
Bolsa palabras/Bernoulli	4,51 %	2 %
Bigramas/Multinomial	4,49 %	6,28 %
Bigramas/Bernoulli	0 %	58,79 %

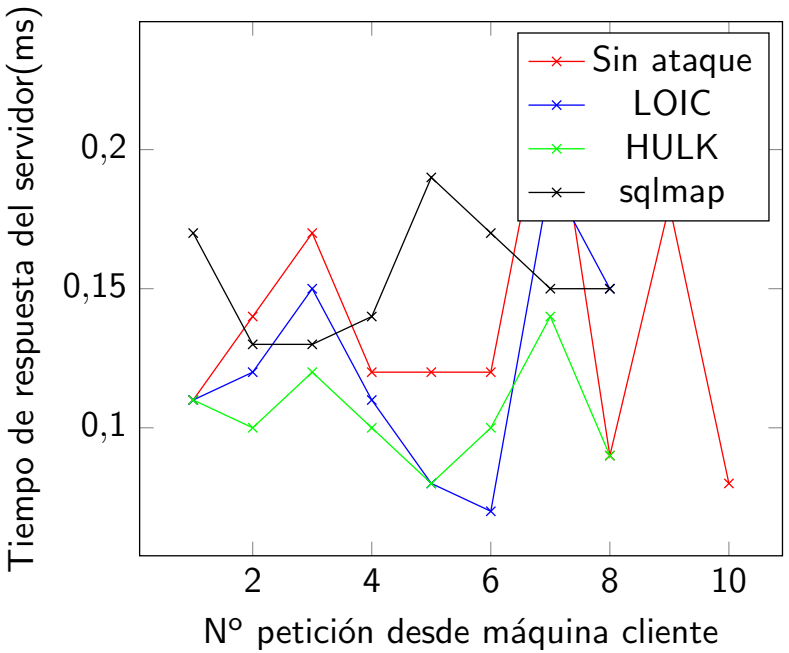
Tabla: Falsos positivos y falsos negativos para el campo *Agent*

Variación	**FP**	**FN**
Bolsa palabras/Multinomial	4,80 %	6,20 %
Bolsa palabras/Bernoulli	4,21 %	0,13 %
Bigramas/Multinomial	4,33 %	5,89 %
Bigramas/Bernoulli	0 %	66,49 %

Tabla: Falsos positivos y falsos negativos para el campo *Referrer*

- **Sistema montado: 3 máquinas**
  - 1 máquina servidor
  - 1 máquina atacante
  - 1 máquina cliente ⇒ Tiempo de respuesta





Índice

1. Introducción
2. Sistema propuesto
3. Ataques contemplados
4. Arquitectura
   - ELK
   - Detección de ataques
     - Clasificador
     - Sistema de reglas
   - Generación de reglas
5. Caso de estudio
6. Conclusiones

Conclusiones y Problemas encontrados

- **Es viable crear un sistema que gestione de manera automática la respuesta a eventos de ataque a una aplicación web**

Conclusiones y Problemas encontrados

- **Es viable crear un sistema que gestione de manera automática la respuesta a eventos de ataque a una aplicación web**
- Siempre y cuando se disponga de un volumen elevado de datos para entrenar al sistema clasificador

Conclusiones y Problemas encontrados

- **Es viable crear un sistema que gestione de manera automática la respuesta a eventos de ataque a una aplicación web**
- Siempre y cuando se disponga de un volumen elevado de datos para entrenar al sistema clasificador
- ¡Problema grave! **Difícil encontrar datos en materia de seguridad**

Conclusiones y Problemas encontrados

- **Es viable crear un sistema que gestione de manera automática la respuesta a eventos de ataque a una aplicación web**
- Siempre y cuando se disponga de un volumen elevado de datos para entrenar al sistema clasificador
- ¡Problema grave! **Difícil encontrar datos en materia de seguridad**

Trabajo futuro

- **Conseguir número mayor de datos para entrenar el sistema** → No tanto en cantidad, sino en variedad
- **Extender el uso del clasificador a otros campos** en el sistema de reglas
- **Afinar el sistema de actuación** basado en el nuevo clasificador entrenado con los nuevos datos

Gestión del proyecto