Universitat Oberta de Catalunya

uoc.edu

Análisis de sistemas WAFs con herramientas Open Source

Alumno: Olavo Gabriel Pavón Ipiales

Plan de Estudios del Estudiante: “Máster Universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones”. (MISTIC)

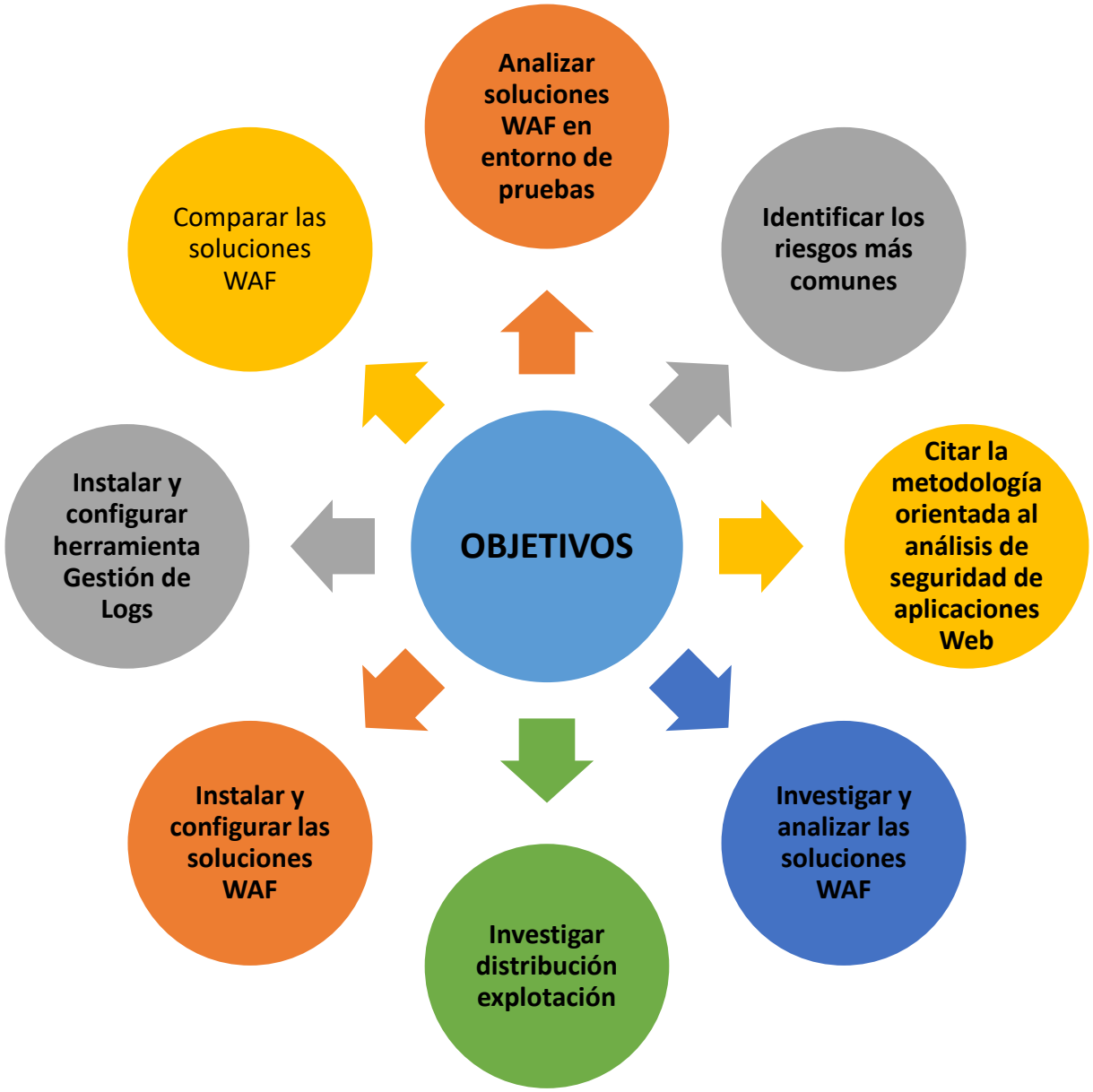
Área del trabajo final: Hacking

Nombre Consultor: Manuel Mendoza

Nombre Profesor responsable de la asignatura: Victor Garcia

AGENDA

- **1. Introducción, contexto, justificación del trabajo, objetivos planteados, planificación del trabajo.**
- 2. Fase de Investigación.
- 3. Implementación
- 4. Resultados Obtenidos
- 5. Video - Funcionamiento del laboratorio implementado
- 6. Conclusiones – Lecciones Aprendidas


OBJETIVOS
Analizar soluciones WAF en entorno de pruebas
Identificar los riesgos más comunes
Citar la metodología orientada al análisis de seguridad de aplicaciones Web
Investigar y analizar las soluciones WAF
Investigar distribución explotación
Instalar y configurar las soluciones WAF
Instalar y configurar herramienta Gestión de Logs
Comparar las soluciones WAF

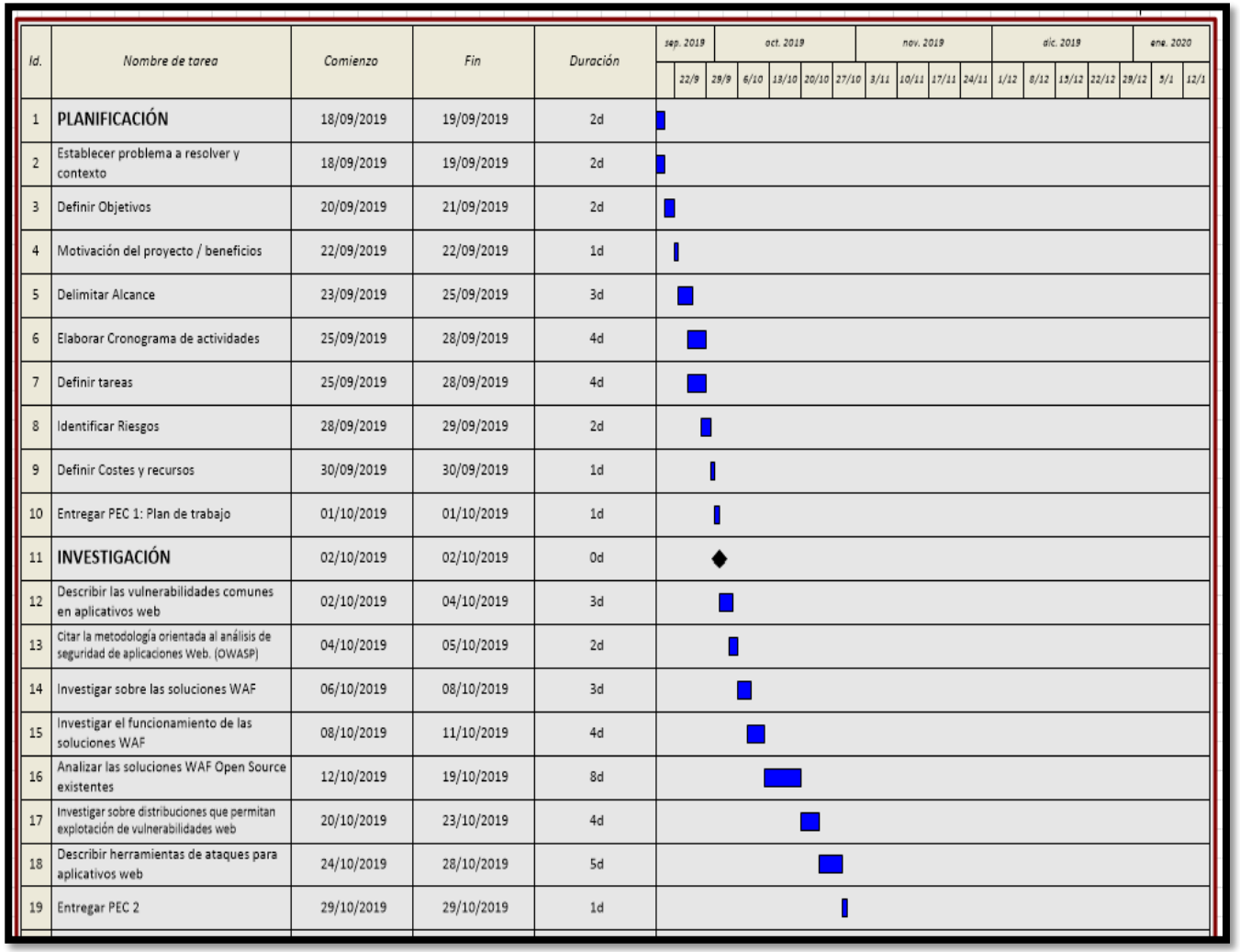
Ciclo de vida del presente proyecto

N. de serie	Fase	Detalle
	Creación	✓ Plan de trabajo ✓ Establecer problema, definir objetivos, beneficios, alcance, riesgos, costes y recursos.
	Inicio de Proyecto	✓ Diagramas de Gantt para revisión y control de tiempos ✓ Entrega de PEC 1.
	Análisis (Investigación)	✓ Investigar sobre todos los conceptos, metodologías y herramientas que se van a utilizar en el presente proyecto. ✓ Entrega PEC 2.
	Diseño, Instalación, Configuración y Pruebas.	✓ Diseñar la topología adecuada. ✓ Preparar el entorno de virtualización para el entorno de pruebas. ✓ Instalar, configurar todas las herramientas necesarias para la ejecución del proyecto. ✓ Realizar todas las pruebas necesarias, para describir los resultados obtenidos. ✓ Entrega PEC 3.
	Cierre de Proyecto	✓ Conclusiones del trabajo realizado. ✓ Redacción de la memoria final. ✓ Entrega PEC 4. ✓ Preparación de video y diapositivas. ✓ Entrega PEC 5. ✓ Defensa de TFM.

Planificación del Trabajo - Gantt Preliminar:

Planificación temporal detallada de las tareas y sus dependencias:

Id.	Nombre de tarea	Comienzo	Fin	Duración
1	**PLANIFICACIÓN**	18/09/2019	19/09/2019	2d
2	Establecer problema a resolver y contexto	18/09/2019	19/09/2019	2d
3	Definir Objetivos	20/09/2019	21/09/2019	2d
4	Motivación del proyecto / beneficios	22/09/2019	22/09/2019	1d
5	Delimitar Alcance	23/09/2019	25/09/2019	3d
6	Elaborar Cronograma de actividades	25/09/2019	28/09/2019	4d
7	Definir tareas	25/09/2019	28/09/2019	4d
8	Identificar Riesgos	28/09/2019	29/09/2019	2d
9	Definir Costes y recursos	30/09/2019	30/09/2019	1d
10	Entregar PEC 1: Plan de trabajo	01/10/2019	01/10/2019	1d
11	**INVESTIGACIÓN**	02/10/2019	02/10/2019	0d
12	Describir las vulnerabilidades comunes en aplicativos web	02/10/2019	04/10/2019	3d
13	Citar la metodología orientada al análisis de seguridad de aplicaciones Web. (OWASP)	04/10/2019	05/10/2019	2d
14	Investigar sobre las soluciones WAF	06/10/2019	08/10/2019	3d
15	Investigar el funcionamiento de las soluciones WAF	08/10/2019	11/10/2019	4d
16	Analizar las soluciones WAF Open Source existentes	12/10/2019	19/10/2019	8d
17	Investigar sobre distribuciones que permitan explotación de vulnerabilidades web	20/10/2019	23/10/2019	4d
18	Describir herramientas de ataques para aplicativos web	24/10/2019	28/10/2019	5d
19	Entregar PEC 2	29/10/2019	29/10/2019	1d

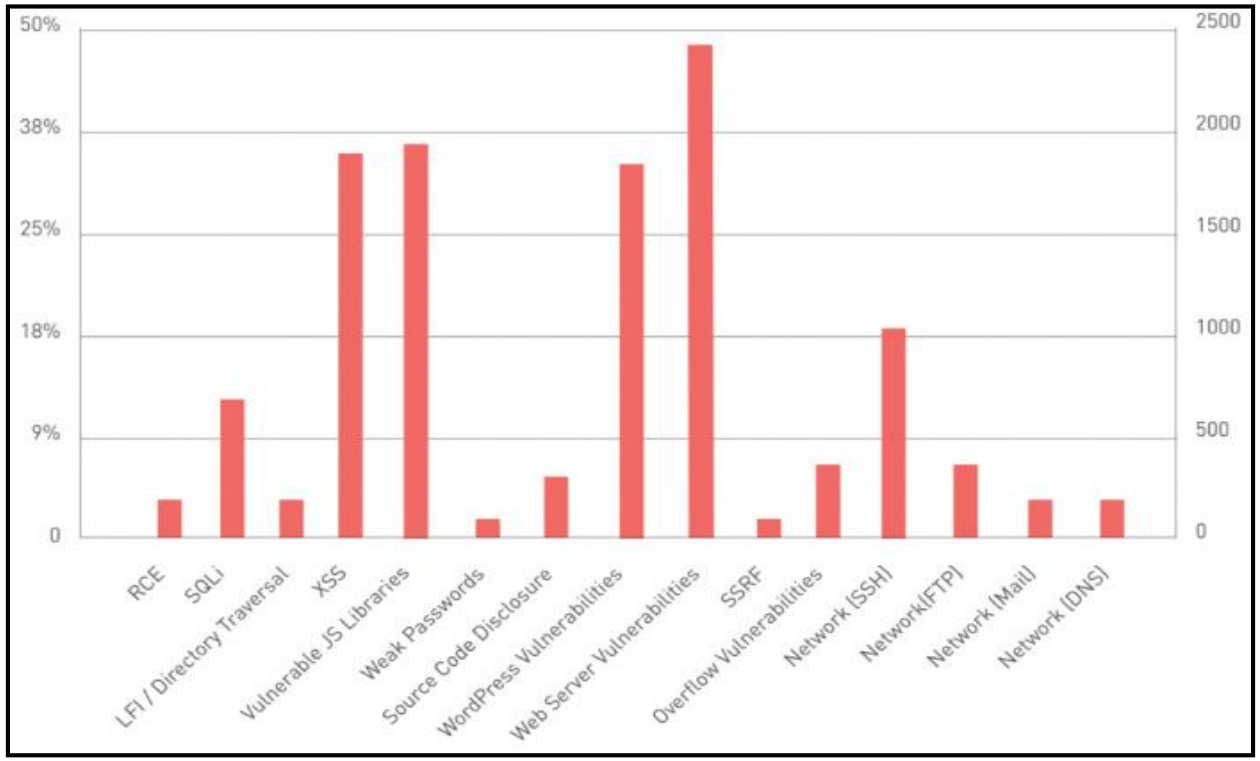


AGENDA

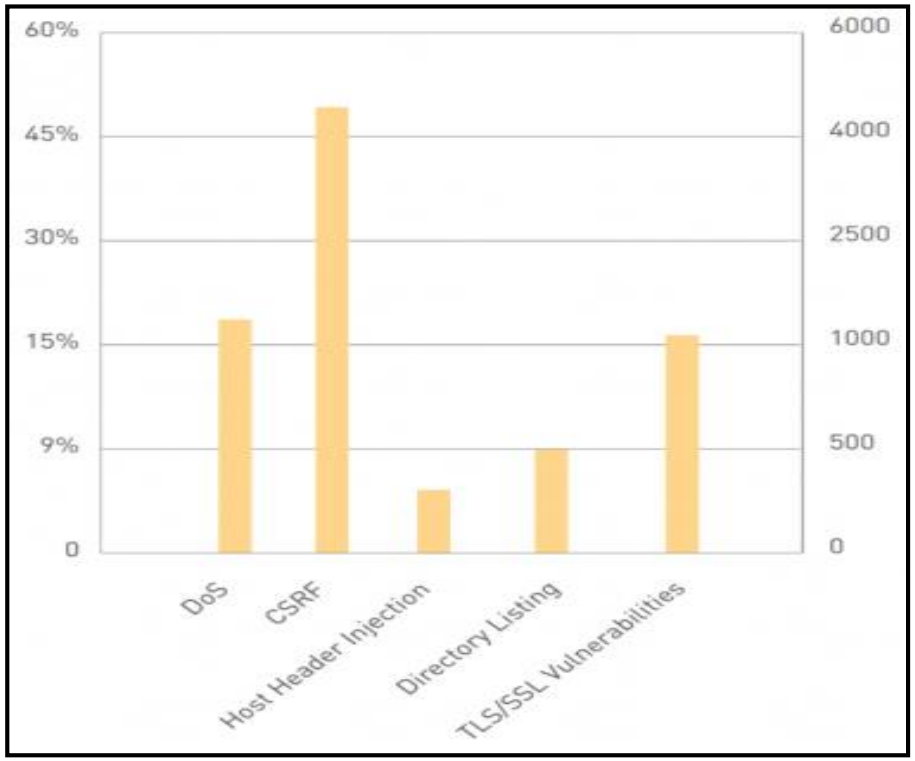
- 1. Introducción, contexto, justificación del trabajo, objetivos planteados, planificación del trabajo
- **2. Fase de Investigación**
- 3. Implementación
- 4. Resultados Obtenidos
- 5. Video - Funcionamiento del laboratorio implementado
- 6. Conclusiones – Lecciones Aprendidas

Vulnerabilidades comunes en aplicativos web

- De acuerdo con el reporte de Acunetix “Web Application Vulnerability 2019:



Vulnerabilidades: Criticas



Medias

Metodología orientada al análisis de seguridad de aplicaciones Web. (OWASP)

- OWASP (Open Web Application Security Project) TOP 10 (2017):

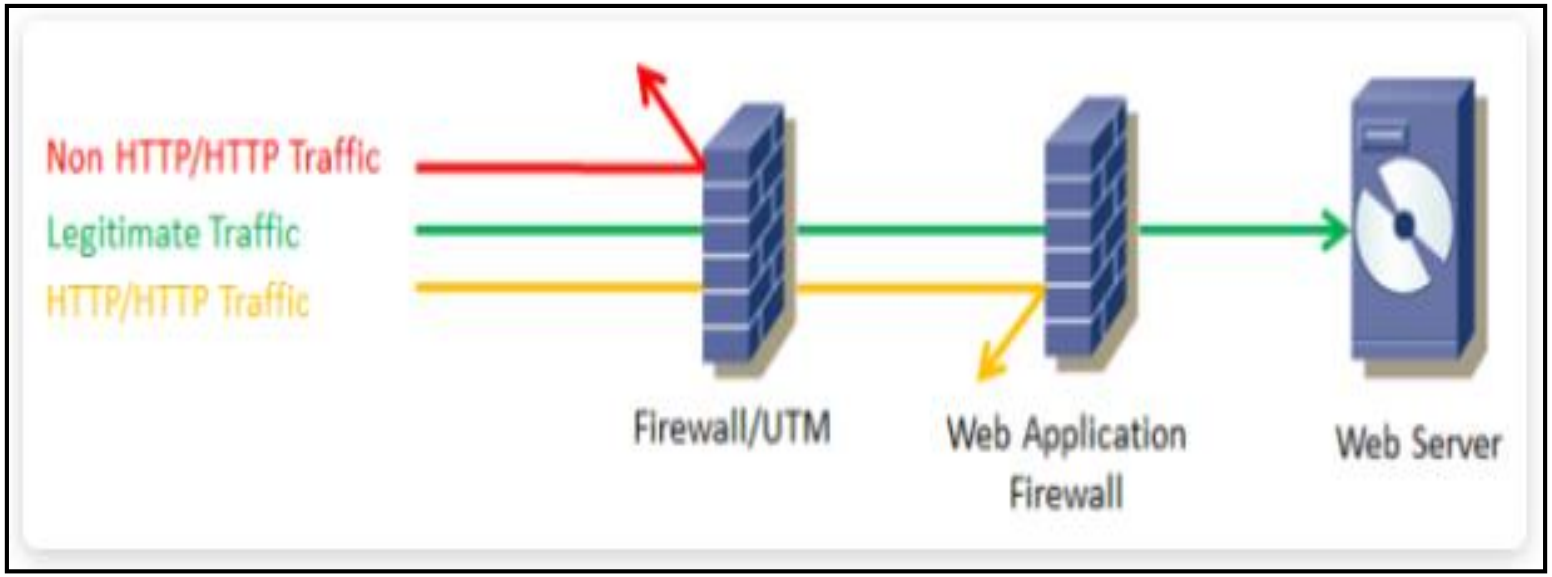
ITEM	Control	Explotabilidad	Prevalencia	Impacto
A1	Inyección	fácil	común	severo
A2	Autenticación rota y gestión de sesiones	fácil	común	severo
A3	Exposición de datos sensibles	promedio	generalizada	severo
A4	Entidad externa de XML (XXE)	promedio	común	severo
A5	Control de acceso inseguro	promedio	común	severo
A6	Configuración incorrecta de seguridad	fácil	generalizada	moderado
A7	Cross-Site Scripting (XSS)	fácil	generalizada	moderado
A8	Deserialización insegura	difícil	común	severo
A9	Uso de componentes con vulnerabilidades conocidas	promedio	generalizada	moderado
A10	Registro y monitoreo insuficientes	promedio	generalizada	moderado

Soluciones WAF

Beneficios

- Protección contra explotaciones de día cero
- Parches Temporales Automatizados
- Protección de ataques web comunes
- Previene fuga de información
- Balanceo de carga del tráfico
- Plataforma en alta disponibilidad
- Dar respuesta inmediata

Funcionamiento de las soluciones WAF



Posee firmas de ataques para:

- Sistemas Operativos.
- Servidores web.
- Lenguajes, frameworks y aplicaciones.
- Servicio de base de datos.

WAFs

Modos de implementación

- **WAF en modo Puente Transparente (Bridge)**
- **WAF en modo Proxy Inverso**
- **WAF End-point en modo embebido o plugin**
- **Cloud-Based**

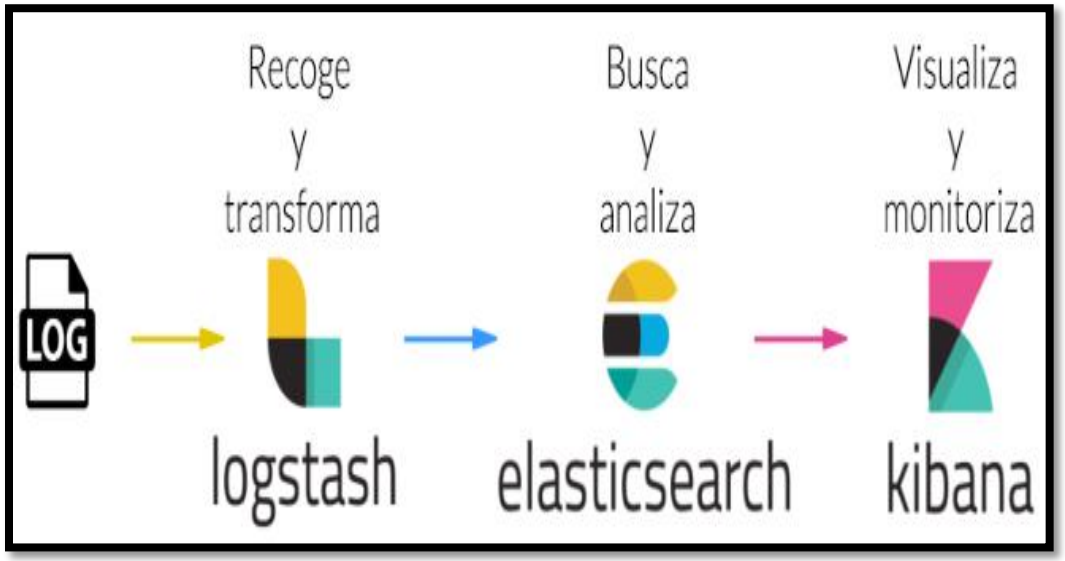
Modelo de seguridad

- **Modelo de Seguridad Positiva**
- **Modelo de Seguridad Negativa**

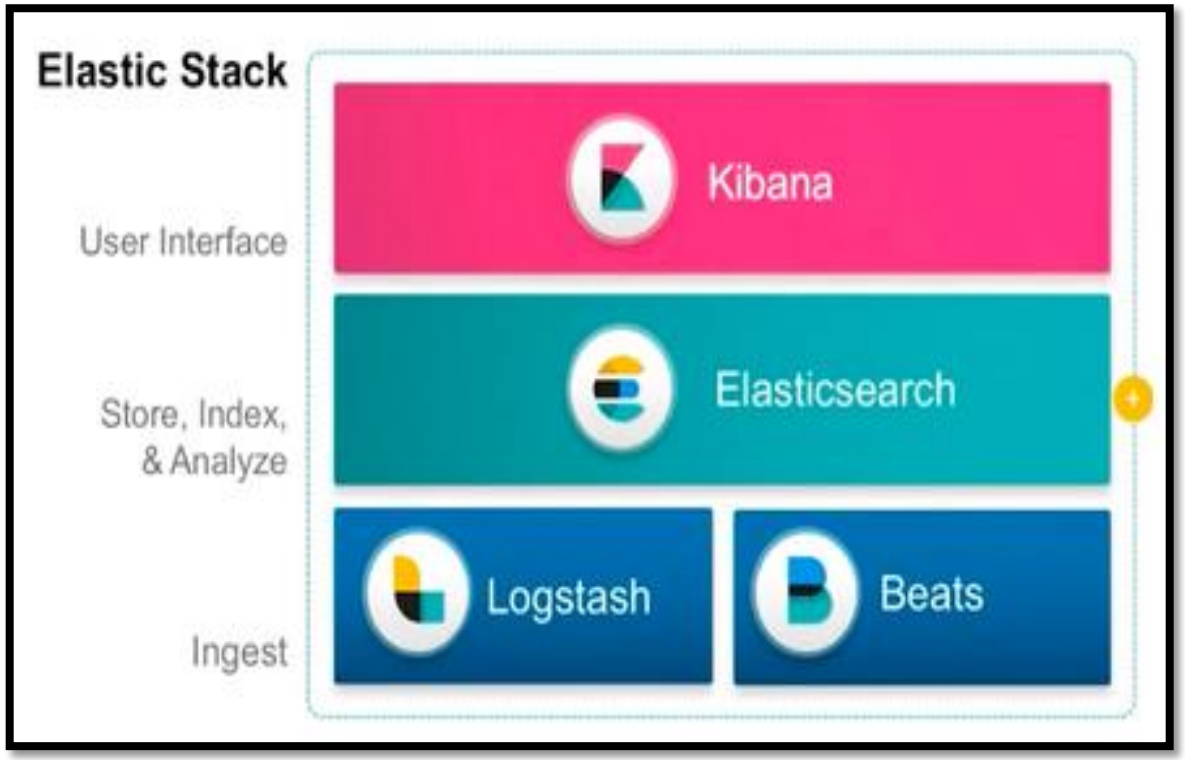
Riesgos que implica emplear un WAF

- **No estar configurados correctamente, transacciones licitas denegadas y pérdida de capital**
- **Retardo en las transferencias**

Herramienta Open Source para la gestión de logs – ELK STACK



Ingesta de datos en ELK



Arquitectura ELK

AGENDA

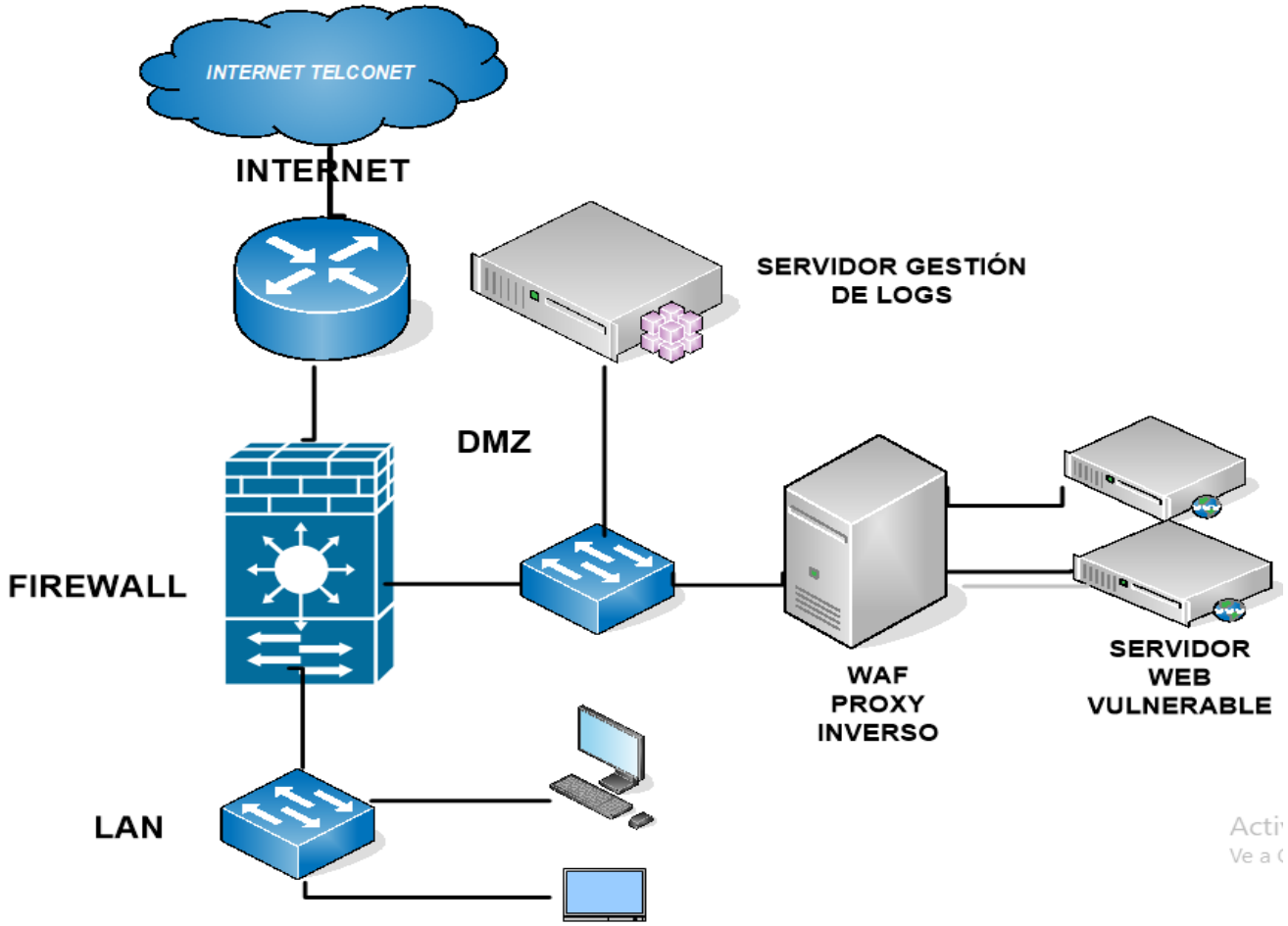
- 1. Introducción, contexto, justificación del trabajo, objetivos planteados, planificación del trabajo
- 2. Fase de Investigación
- **3. Implementación**
- 4. Resultados Obtenidos
- 5. Video - Funcionamiento del laboratorio implementado
- 6. Conclusiones – Lecciones Aprendidas

Soluciones Open Source seleccionadas para la Implementación del laboratorio:

Distribución para explotación de vulnerabilidades web	• **Web Security Dojo-3.3 - Portal Web DVWA**
Soluciones WAFs	• **ModSecurity para apache** • **Vulture.** • **Waf2Py.** • **Nginx**
Solución de Gestión de Logs	• **ELK Stack**

Implementación – Topología de red Propuesta:

- La topología de red propuesta para el laboratorio es la siguiente:



Herramientas de Hardware y Software:

- **Herramientas Hardware necesarias:**
- Computador portátil Intel Core i7 @2.40 GHz, 8GB de memoria RAM.

- **Herramientas Software necesarias:**
- VMware Workstation Pro 14.1.2 build-8497320.
- Sistema Operativo Centos 7.
- Sistema Operativo Debian 9.
- Sistema Operativo FreeBSD 12.0-Release.
- Distribución Web Security Dojo-3.3
- Distribución Kali Linux
- Distribución bitnami-elk-7.3.2
- Sistema Operativo Windows 10 (Cliente)
- Herramienta Filebeat-oss 7.3.2

Limitante de recursos en el hardware

- Se ha excluido la implementación del firewall en el laboratorio, ya que no abastece el recurso necesario para ejecutar 4 o 5 máquinas virtuales al mismo tiempo. El direccionamiento IP utilizado es:

Plataforma	Dirección IP	Dirección IP de escucha
DOJO - Portal Web DVWA	192.168.1.41	NA
KALI LINUX	192.168.1.10	NA
MODSECURITY	192.168.1.40	NA
VULTURE	192.168.1.103:8000	192.168.1.40
WAF2PY	192.168.1.101:62443	192.168.1.40
NGINX	192.168.1.107	NA
Cliente Windows	192.168.1.110	NA
Servidor Gestión Logs ELK Stack	192.168.1.142	NA

- El dominio que se va a utilizar para el aplicativo web vulnerable es: **proxy-waf.com.ec**. Este dominio es falso, y solo se lo utiliza para realizar las pruebas respectivas en la red DMZ.

Requerimientos necesarios para implementar WAFs Open Source

	ModSecurity + Apache
Sistema Operativo	Centos 7
Interfaz	Mínimal, sin entorno gráfico
Interza Web para administar y configurar	NO
Tamaño en disco duro	50 GB
Memoria RAM	2 GB
Procesadores	2
Paquete	httpd-2.4.6-90
Modulo	proxy_module
Paquete	mod_secutiry-2.9.3
Modulo	security2_module
Paquete	**mod_secutiry-crs**

	Vulture
Sistema Operativo	FreeBSD 12.0-Release
Interfaz	Minimal, sin entorno gráfico
Interza Web para administar y configurar	SI
Tamaño en disco duro	50 GB
Memoria RAM	2 GB
Procesadores	2
Paquete	apache2
Modulo	mod_secutiry
Modulo	mod_svm (Machine learning basado en Support Vector Machines)
Modulo	mod_defender
Paquete	**mod_secutiry-crs**

	Waf2Py
Sistema Operativo	Debian 9.3
Interfaz	Mínimal, sin entorno gráfico
Interza Web para administar y configurar	SI
Tamaño en disco duro	50 GB
Memoria RAM	2 GB
Procesadores	2
Paquete	apache2
Modulo	mod_secutiry v3
Modulo	Web2Py 2.17.2
Modulo	Modsecurity Nginx connector
Paquete	**mod_secutiry-crs**
Paquete	openresty-1.13.6.2

	NGINX ModSecurity
Sistema Operativo	Centos 7
Interfaz	Minimal, sin entorno gráfico
Interza Web para administar y configurar	NO
Tamaño en disco duro	50 GB
Memoria RAM	2 GB
Procesadores	2
Paquete	nginx-1.15.12
Modulo	mod_security
Modulo	**owasp-modsecurity-crs**

Reglas en las soluciones WAF

- Se describe el formato de una regla básica que utilizan los sistemas WAFs:

  SecRule VARIABLES OPERATOR ACTIONS

- Ejemplo: SecRule ARGS "@rx <script>" \
  "id:2000,log,deny,status:404"

Los tres parámetros tienen los siguientes significados:

- El parámetro “VARIABLES” le dice a ModSecurity donde buscar. La variable “ARGS”, indica todos los parámetros de solicitud “requests”.
- El parámetro “OPERATOR” le dice a ModSecurity como mirar. La variable "@rx <script>" es un patrón de expresión regular, que se comparará con ARGS.
- El parámetro “ACTIONS” se usa para agregar metadatos a las reglas y para especificar qué debe hacer ModSecurity cuando ocurre una coincidencia.

AGENDA

- 1. Introducción, contexto, justificación del trabajo, objetivos planteados, planificación del trabajo
- 2. Fase de Investigación
- 3. Implementación
- **4. Resultados Obtenidos**
- 5. Video - Funcionamiento del laboratorio implementado
- 6. Conclusiones – Lecciones Aprendidas

Integrar el laboratorio y realizar las pruebas respectivas

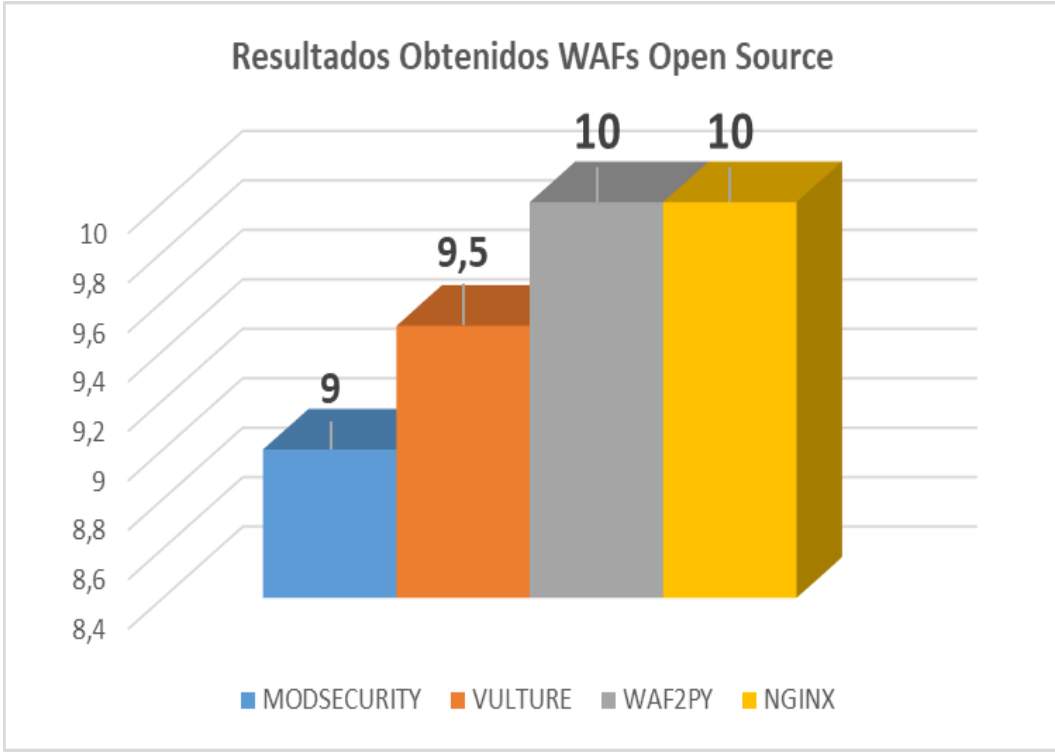
- El CRS proporciona protección contra muchas categorías de ataque comunes, entre ellos se va a realizar los siguientes ataques comunes sobre el target DVWA de la herramienta Web Security Dojo, se ha propuesto dar una puntuación a cada ataque, y así poder evaluar la efectividad de detección y bloqueo de cada WAF:

ITEM	ATAQUES	PUNTAJE TOTAL
1	Fuerza Bruta	1
2	Ejecución de comandos	1
3	File Inclusión local	1
4	File Inclusión externo	1
5.1	Inyección SQL Manual	0,33
5.2	Inyección SQL Manual	0,33
5.3	Inyección SQL Automatizado	0,34
6.1	Inyección SQL (BLIND) Manual	0,5
6.2	Inyección SQL (BLIND) Automatizado	0,5
7	File Upload	1
8	XSS DOM	1
9	XSS Reflected	1
10	XSS Stored	1
	TOTAL=	10

Resultados obtenidos de los ataques:

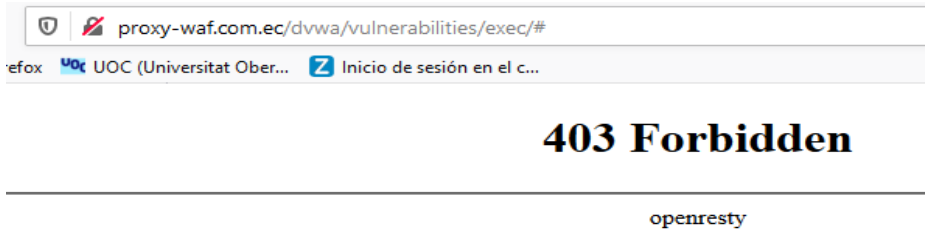
Los resultados obtenidos de la detección y bloqueo de la efectividad de cada WAF Open Source, se detallan a continuación:

			WAF OPEN SOURCE			
ITEM	ATAQUES	PUNTAJE TOTAL	MODSECURITY	VULTURE	WAF2PY	NGINX
1	Fuerza Bruta	1	1	1	1	1
2	Ejecución de comandos	1	1	1	1	1
3	File Inclusión local	1	1	1	1	1
4	File Inclusión externo	1	1	1	1	1
5.1	Inyección SQL Manual	0,33	0,33	0,33	0,33	0,33
5.2	Inyección SQL Manual	0,33	0,33	0,33	0,33	0,33
5.3	Inyección SQL Automatizado	0,34	0,34	0,34	0,34	0,34
6.1	Inyección SQL (BLIND) Manual	0,5	0,5	0,5	0,5	0,5
6.2	Inyección SQL (BLIND) Automatizado	0,5	0,5	0,5	0,5	0,5
7	File Upload	1	0	0,5	1	1
8	XSS DOM	1	1	1	1	1
9	XSS Reflected	1	1	1	1	1
10	XSS Stored	1	1	1	1	1
	TOTAL=	**10**	**9**	**9,5**	**10**	**10**



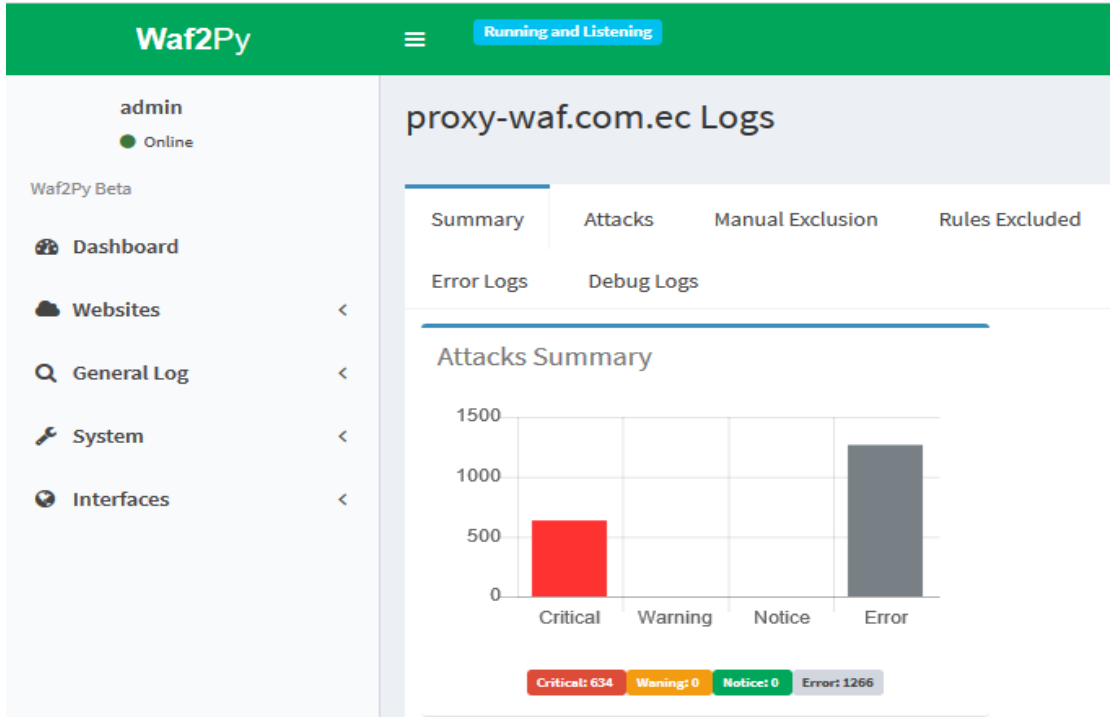
Monitorizar los logs obtenidos de los WAFs Open Source

- **Respuesta de detección y bloqueo obtenidos en WAF Waf2Py**



Mensaje de bloqueo

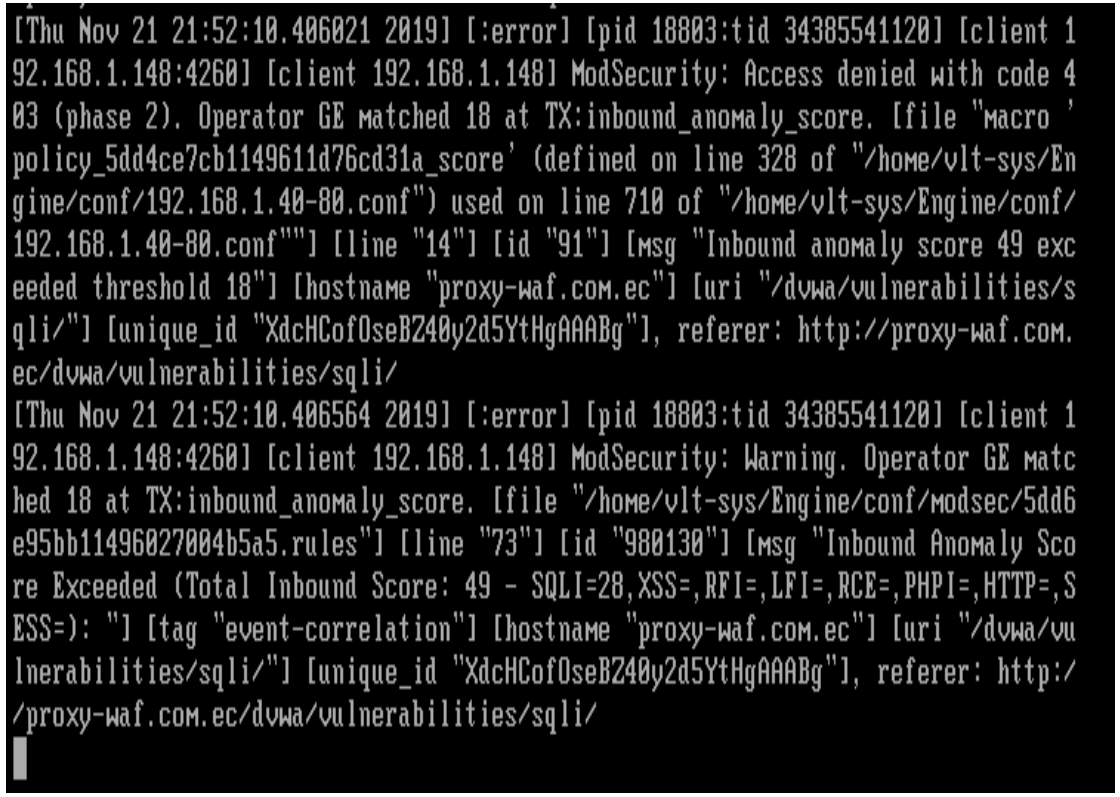
Registro de logs



Resumen estadísticas

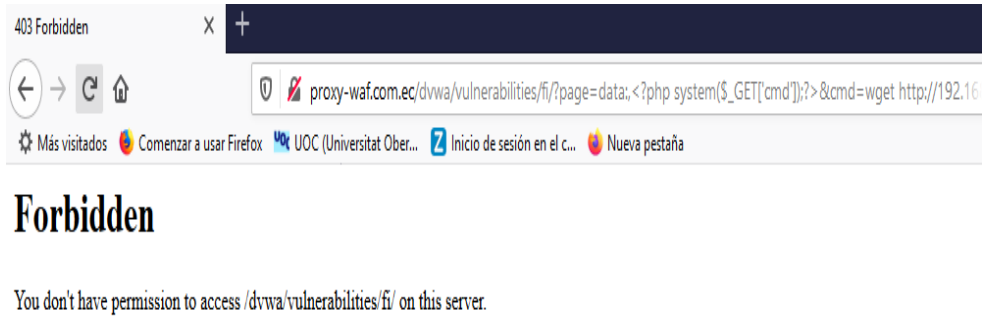
- **Respuesta de detección y bloqueo obtenidos en WAF Vulture:**

Mensaje de bloqueo



Registro de logs

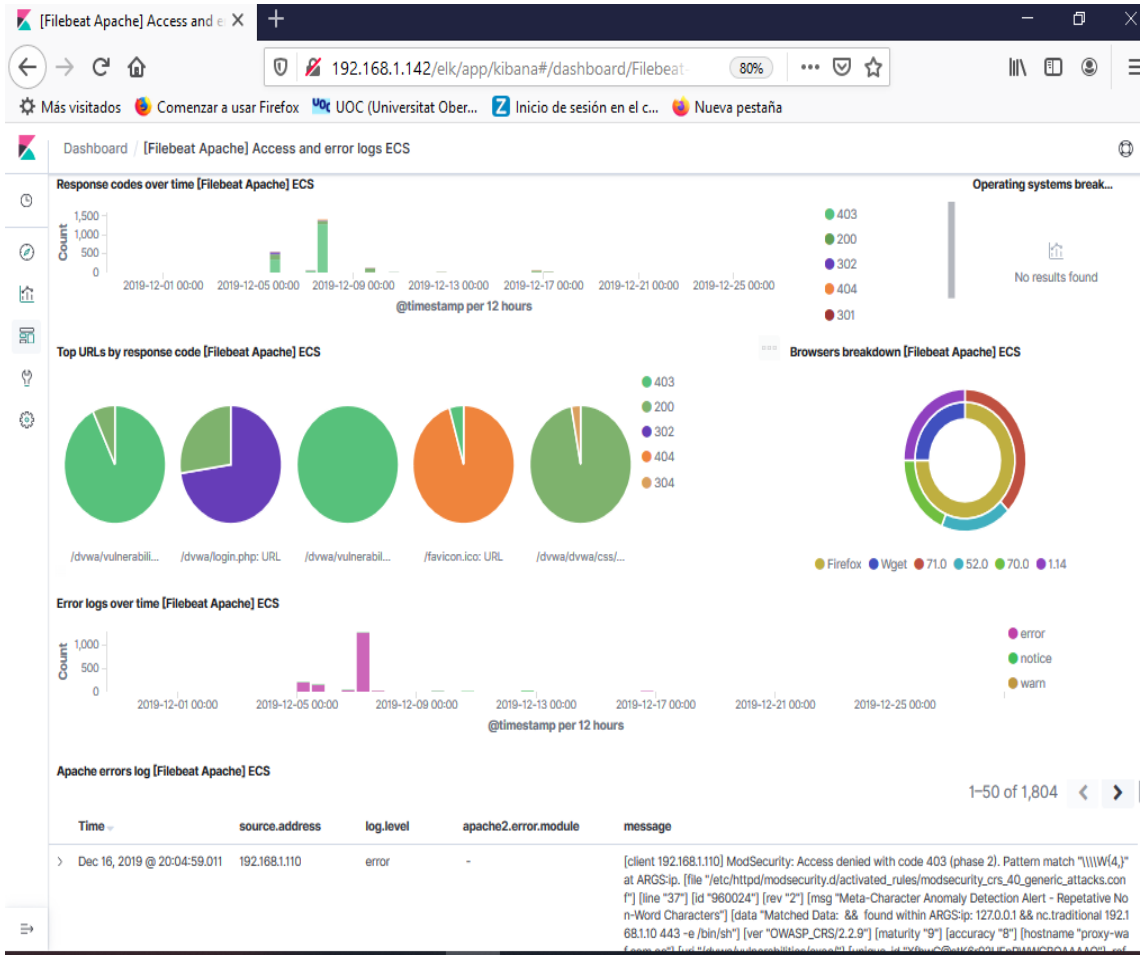
- **Respuesta de detección y bloqueo obtenidos en WAF ModSecurity:**

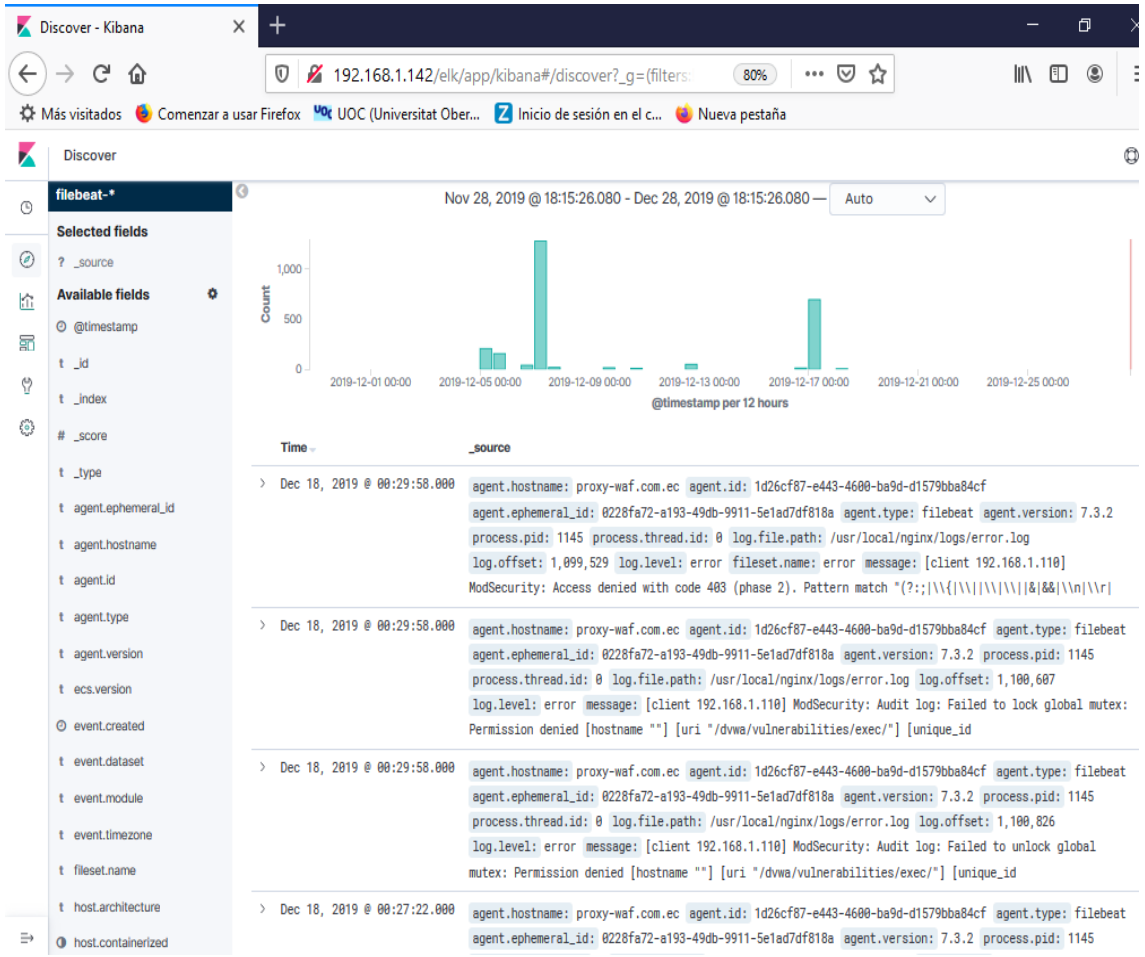


Mensaje de bloqueo

Registro de logs

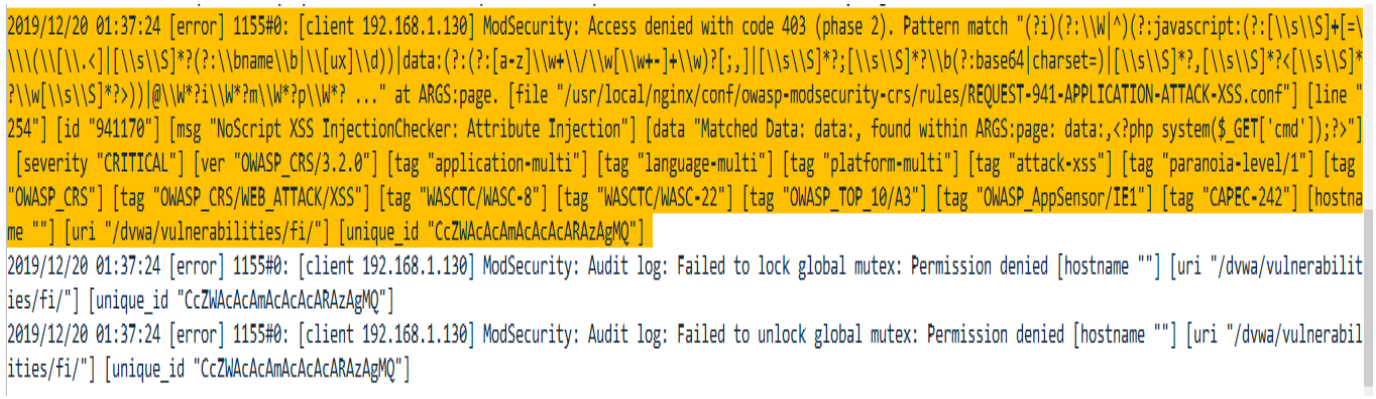
- **Dashboard (Filebeat Apache) en servidor de gestión logs ELK Stack – WAF ModSecurity:**





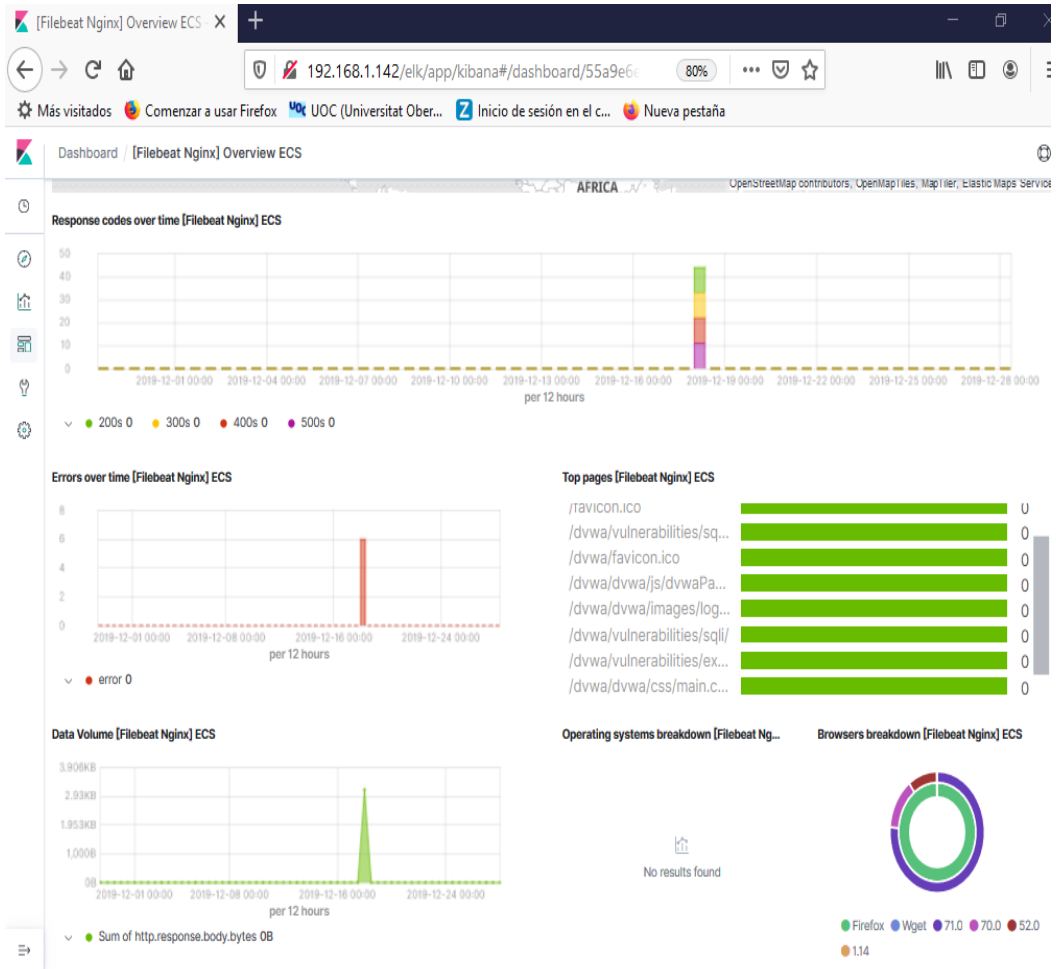
- **Respuesta de detección y bloqueo obtenidos en WAF Nginx:**

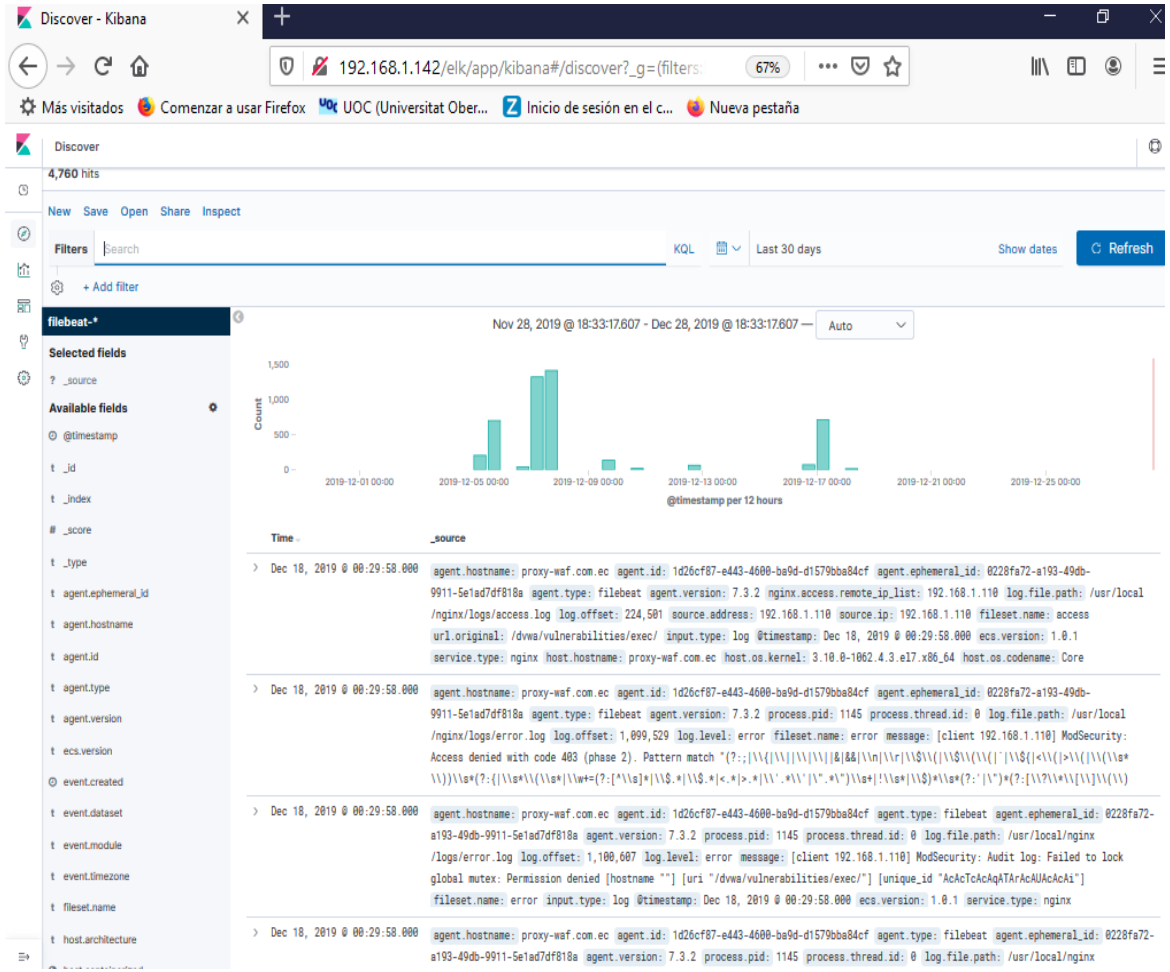
Mensaje de bloqueo



Registro de logs

- Dashboard (Filebeat Nginx) en servidor de gestión logs ELK Stack – WAF NGINX:





AGENDA

- 1. Introducción, contexto, justificación del trabajo, objetivos planteados, planificación del trabajo
- 2. Fase de Investigación
- 3. Implementación
- 4. Resultados Obtenidos
- **5. Video - Funcionamiento del laboratorio implementado**
- 6. Conclusiones – Lecciones Aprendidas

FUNCIONAMIENTO DE LOS WAFs OPEN SOURCE - VIDEO

AGENDA

- 1. Introducción, contexto, justificación del trabajo, objetivos planteados, planificación del trabajo
- 2. Fase de Investigación
- 3. Implementación
- 4. Resultados Obtenidos
- 5. Video - Funcionamiento del laboratorio implementado
- **6. Conclusiones – Lecciones Aprendidas**

CONCLUSIONES

Lecciones Aprendidas

- Los WAFs son indispensables en la actualidad
- Implementación de los WAFs con herramientas Open Source en empresas que no cuentan con capital
- Los WAFs se adaptan a cualquier entorno web
- Nudo crítico al implementar el laboratorio fue la falta de capacidad del hardware
- La etapa de implementación de los WAFs Open Source, no se cumplió en los tiempos establecidos de acuerdo a la planificación temporal
- Poner en producción la implementación de los WAFs
- Implementar el WAF Vulture con el módulo Virtual Paching y SVM (Support Vector Machine)
- Todos los objetivos planteados en el presente proyecto fueron cumplidos a satisfacción

MUCHAS GRACIAS POR SU ATENCIÓN