Universidad de Buenos Aires

Facultades de Ciencias Económicas,
Cs. Exactas y Naturales e Ingeniería

Carrera de especialización en Seguridad Informática

Trabajo Final

Tema

Demostración de los beneficios de Fortiweb en una organización

Autor:

Carlos Sulca Galarza

Tutor:

Pedro Hecht

Año: 2019

Cohorte 2018

Resumen

En la actualidad el crecimiento de internet ha impactado directamente en la seguridad de la información manejada cotidianamente, sitios de comercio electrónico, servicios, bancos e incluso redes sociales contienen información sensible que en la mayoría de los casos resulta ser muy importante una debida protección.

Se puede decir que uno de los puntos más críticos de la seguridad en Internet son las herramientas que interactúan de forma directa con los usuarios, en este caso los servidores web. Es común escuchar sobre fallas en los sistemas de protección de los servidores más frecuentemente utilizados, por ejemplo, Apache, NGINX, IIS, etc o en los lenguajes de programación en que son escritas las aplicaciones. Sin embargo, la mayoría de los problemas detectados en servicios web no son provocados por fallas de ninguna de estas partes, si no que los problemas se generan por malas prácticas de parte de los programadores.

Debemos entender que programar aplicaciones web seguras no es una tarea fácil, ya que requiere por parte del programador, no sólo cumplir con el objetivo funcional básico de la aplicación, sino una concepción general de los riesgos que puede correr la información procesada por el sistema, ante estas posibles falencias podemos describir el top 10 OWASP sobre las principales vulnerabilidades de aplicaciones web:

- Inyecciones SQL.
- Autenticación rota y gestión de sesión.
- Secuencias de comandos entre sitios (XSS).
- Referencias inseguras y directas a objetos.
- Mala configuración de seguridad.
- Exposición de datos sensibles.
- Falta de control de acceso a nivel de función.
- Solicitud de falsificación entre sitios (CSRF).
- Usando componentes vulnerables conocidos.

- Redirecciones no validadas.

Para solventar estas falencias podemos usar los beneficios que nos brinda Fortiweb para la mitigación del TOP 10 de amenazas definidas por OWASP.

Tabla de contenido

1. Introducción

El increíble mundo de la web hoy en día se ha vuelto un complemento de nuestra vida cotidiana y también en entornos corporativos por esta razón se crea la necesidad de implementar un sistema que permita mantener más seguro este ámbito web, esto no quiere decir que la implementación de una aplicación especializada en filtrar ataques web sea un sustitutivo de otras medidas de protección que debe llevar a cabo los desarrolladores. Las aplicaciones web sin protección son el punto más fácil de entrada para los hackers y son vulnerables a muchos tipos de ataques.

El WAF (Web Application Firewall) es un firewall distinto a los de usos convencionales, que traen seguridad al perímetro de la red corporativa. Un WAF crea una barrera entre un servicio web y todo el resto de internet. Es un muro que impide que cualquier usuario malintencionado tenga acceso a su sitio web o aplicación de forma no autorizada.

El Web Application Firewall bloquea y protege su aplicación contra acciones malintencionadas como manipulación de contenido visualizado, inyecciones indebidas en la base de datos de SQL estándar, conocida como “inyección de SQL”, algunos tipos de fraudes en el acceso administrativo y otras tantas especies de ciberataques.

WAF tiene la capacidad de monitorear, filtrar y bloquear automáticamente el tráfico de datos potencialmente malicioso a través de configuraciones y reglas predeterminadas que pueden impedir fácilmente los ataques más comunes. De esta forma su empresa evita trastornos con robo de información, aplicación fuera del aire debido a ataques DDoS, además de disminuir los gastos con infraestructura y recursos operacionales.

2. Objetivos

El objetivo de este trabajo es describir los beneficios que nos ofrece implementar en nuestras organizaciones un Fortiweb para la protección de nuestros servicios publicados en Internet como en Intranet.

Se realizará un breve análisis de sus funcionalidades y como nos protege de las vulnerabilidades más comunes definidas en el TOP 10 de OWASP, también validaremos como Fortiweb se alinea a las exigencias de cumplimiento que pueden ser requeridos por distintas organizaciones, a continuación, objetivos secundarios:

- Presentación de los beneficios de Fortiweb.
- Mecanismos de mitigación referentes al TOP 10 de vulnerabilidades de OWASP.
- Cumplimiento de Fortiweb con el estándar PCI DSS.
- Cumplimientos de políticas de seguridad de la información de una organización.
- Con la puesta en producción de Fortiweb se optimizará los recursos de los servidores backend.
- Mejora en la experiencia de los usuarios al usar los servicios web de una organización.

3. Marco Teórico

Un WAF (Web Application Firewall) es un dispositivo hardware o software que permite proteger los servidores de aplicaciones web de determinados ataques específicos en Internet. Se controlan las transacciones al servidor web de nuestro negocio. La familia de firewall de aplicaciones web constituida por los appliances FortiWeb proporciona protección completa y especializada a todos los niveles para las aplicaciones y servicios Web de pequeñas, medianas y grandes empresas, proveedores de servicios de aplicaciones y proveedores de Software como servicio.

Los módulos de protección de aplicaciones web y de firewall XML de los que dispone FortiWeb protegen las aplicaciones web y los datos e información sensibles publicados en Internet frente a ataques y pérdidas de información y datos críticos. Mediante el uso de avanzadas técnicas para ofrecer protección bidireccional contra ataques complejos y sofisticados como por ejemplo “SQL injection” y “Cross-site scripting”, las plataformas FortiWeb ayudan a prevenir robos de identidad, fraudes económicos y espionaje corporativo. Los dispositivos FortiWeb proporcionan la tecnología imprescindible para monitorizar y aplicar las distintas normativas regulatorias, tanto gubernamentales como de iniciativa privada, asegurar la implantación de las prácticas de seguridad recomendadas y las adecuadas políticas internas.

Mediante el uso de las herramientas automáticas de generación de informes preconfiguradas y totalmente personalizables que proporciona FortiWeb es muy sencillo medir el cumplimiento de las normativas regulatorias, como por ejemplo los estándares PCI DSS (Payment Card Industry’s Data Security Standard).

Las amenazas a la seguridad de las redes e infraestructuras de las organizaciones han evolucionado enormemente y suponen una gran amenaza para las aplicaciones y servicios web, que constituyen el punto de acceso a la información confidencial y crítica guardada en las bases de datos de backend. En respuesta a todas estas amenazas se constituyeron los estándares regulatorios PCI. Sin embargo, asegurar que las aplicaciones web están

completamente libres de vulnerabilidades es complicado debido a la constante aparición de nuevas vulnerabilidades, necesidades de parcheos, revisiones de código, a las presiones del mercado sobre los plazos temporales de las aplicaciones, la dificultad de la identificación de vulnerabilidades e incluso las dificultades de acceso al código de las aplicaciones.

Los appliances FortiWeb reducen de forma drástica el tiempo necesario para proteger de forma completa los recursos públicos y privados accesibles a través de Internet de cada organización y simplifica enormemente las tareas asociadas con la implementación de políticas de seguridad y el cumplimiento de las distintas normativas regulatorias.

4. Equipos de laboratorio

4.1 Sistema DVWA

Para demostrar el funcionamiento de Fortiweb usaremos el sistema web vulnerable llamado DVWA, ejecutada desde **Metasploitable 2** el cual tiene como base el sistema operativo Ubuntu, este entorno de pruebas cuenta con diversos sistemas vulnerables para entrenamiento.

Damn Vulnerable Web App (DVWA) es un reconocido entorno de entrenamiento en explotación de seguridad web escrito en PHP y MySQL cuyo objetivo principal es permitir a programadores y técnicos estudiar e investigar sobre las diferentes temáticas involucradas en dicho campo en un entorno completamente legal.

Dvwa permite analizar diversos ataques como:

- Brute Force
- Command Injection
- CSRF
- File Inclusion
- File Upload
- Insecure CAPTCHA
- SQL Injection
- SQL Injection (Blind)

- XSS (Reflected)
- XSS (Stored)

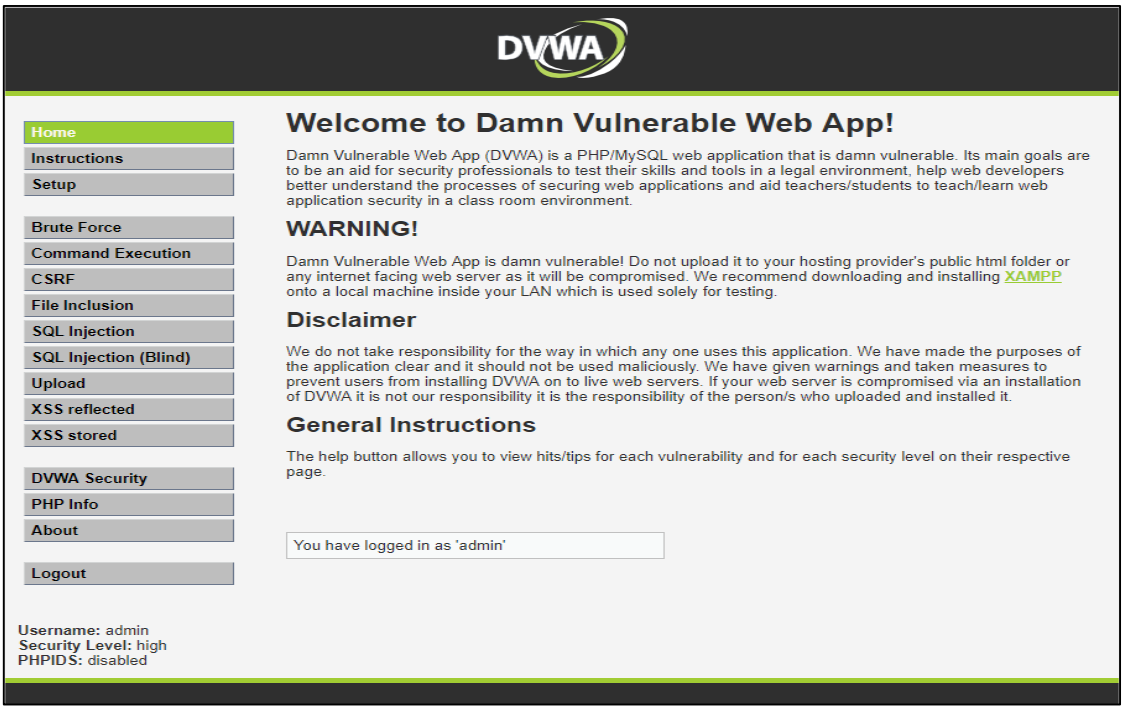

Figura 1. Bienvenida del sistema dvwa

DVWA nos permite trabajar con 3 niveles de seguridad Low, Medium y High.

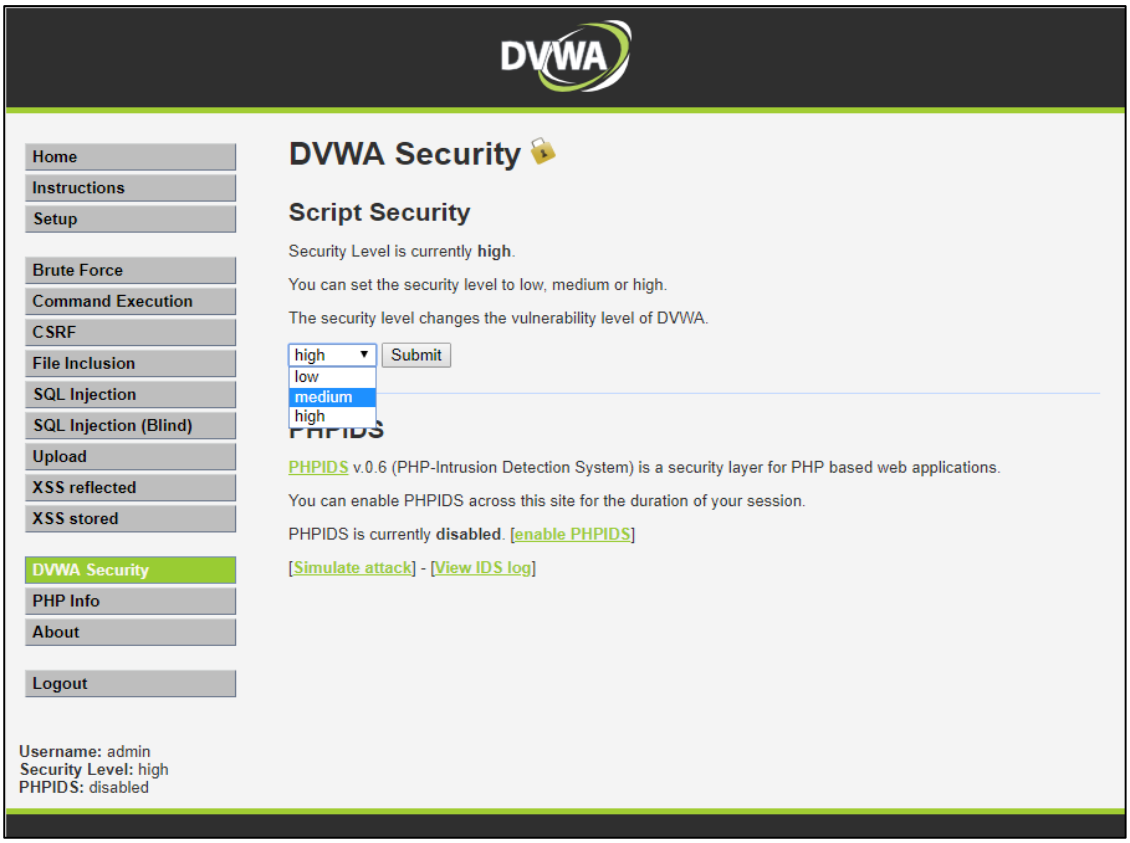

Figura 2. Niveles de seguridad del sistema dvwa

Como se comprobará más adelante, estas malas prácticas existentes en el sistema DVWA pueden llevar a brechas de seguridad graves generando situaciones de peligro para cualquier organización. La versión del sistema web DVWA es 1.0.7, vamos a demostrar cómo Fortiweb puede hacer un hardening de esta aplicación.

4.2 Firewall Fortigate

FortiGate es un Firewall basado en hardware o software desarrollado por Fortinet. El sistema de FortiGate es el único sistema que puede detectar y eliminar virus, gusanos y otras amenazas basadas en contenido, sin afectar al rendimiento de la red, incluso para aplicaciones en tiempo real como la navegación Web. Las soluciones de Fortigate también incluyen:

- Firewall
- Filtrado de contenido
- VPN
- Antivirus
- Antispam
- Detección y prevención de intrusos y gestor de tráfico
- Balanceo de carga
- Alertas por e-mail

Estas características hacen de Fortigate la más rentable, conveniente, potente y segura de las soluciones de seguridad de red disponibles.

La solución de seguridad del firewall Fortinet ofrece cobertura a tres capas: Infraestructura, Usuarios y Aplicaciones. Ofrece un fuerte control y seguridad desde los endpoints al centro de datos, la nube privada e híbrida y desde el perímetro a las aplicaciones, proporcionando una gran flexibilidad para la evolución de la infraestructura TI de las organizaciones.

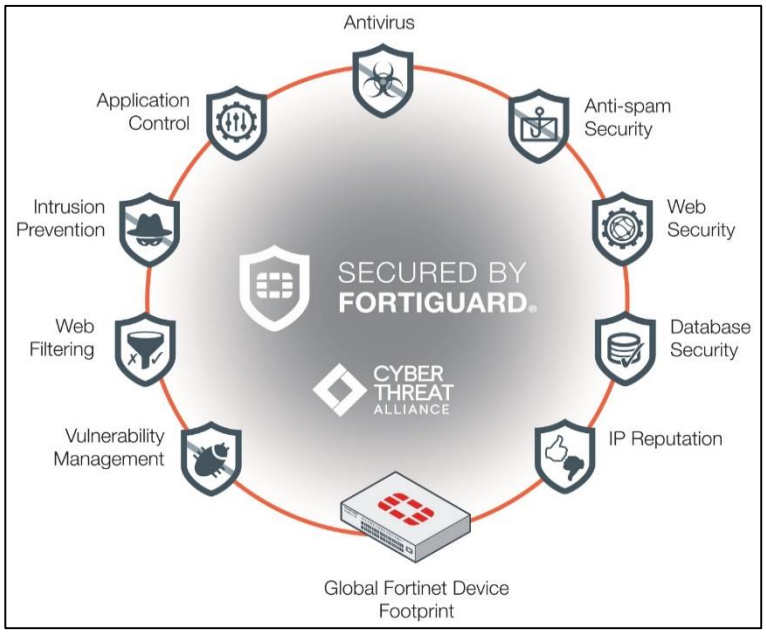

Figura 3. Funcionalidades de Fortigate

4.3 WAF Fortiweb

FortiWeb reduce el tiempo de implementación y simplifican la gestión de la seguridad de su empresa y protege aplicaciones web de ataques dirigidos a exploits -sean conocidos o desconocidos- mediante la utilización de métodos de detección multicapa y correlacionados.

FortiWeb de obtuvo la certificación ICSA Web Application Firewall, galardón que representa una prueba del compromiso por defender los más altos estándares de seguridad de la industria. El logro de esta certificación garantiza que los clientes FortiWeb puedan beneficiarse de las mejores prácticas en la industria de la seguridad informática para todas sus necesidades de aplicaciones Web.

Características y beneficios de FortiWeb:

- Único producto que proporciona un módulo Vulnerability Scanner dentro del firewall de aplicaciones web que completa una solución completa para PCI DSS 6.6.
- Garantiza la seguridad de las aplicaciones web y

contenido de la base de datos. Incluye bloqueo de amenazas tales como cross-site scripting, inyección SQL, desbordamientos de búfer, denegación de servicios, y envenenamiento de cookies, entre otros.

- Cifrado SSL.
- Servidor de equilibrio de carga.
- Compresión de datos.
- Análisis de datos en tiempo real.

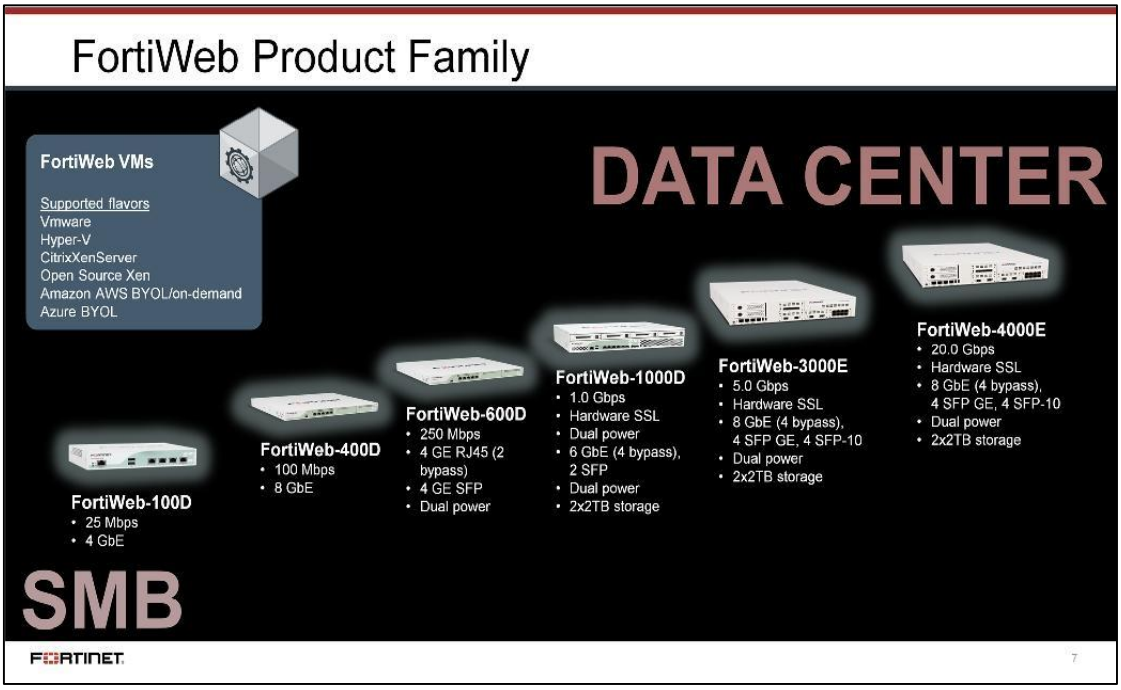

Figura 4. Familia de productos Fortiweb

4.4 Windows 8

Sistema operativo Windows 8 es usado para simular el equipo de un cliente ubicado en internet, este cliente realizará los ataques al sistema web DVWA.

Características del equipo virtual:

- Disco duro: 20 GB
- Memoria ram: 4 GB
- Navegador: Firefox, Google Chrome.

Figura 5. Windows 8

5. OWASP

OWASP, es una organización sin fines de lucro a nivel mundial. Su objetivo es promover la codificación y el endurecimiento de aplicaciones seguras.

OWASP ha estado creciendo de forma constante desde 2004, y ha contribuido a proyectos como la aplicación de educación de seguridad **WebGoat**. Ahora puede encontrar presentadores de OWASP en todas partes, desde las conferencias oficiales AppSec de OWASP en California.

Figura 6. Logo de Owasp

5.1 OWASP Top 10

El top 10 de OWASP es una lista de vulnerabilidades que los expertos en seguridad consideran las más serias amenazas de seguridad web. OWASP los actualiza periódicamente en función de los datos de ataque disponibles.

Muchas organizaciones grandes, incluida PCI, recomiendan que analice estas vulnerabilidades y las corrija o defienda en su contra, a continuación, el top 10:

- Inyecciones SQL.
- Autenticación rota y gestión de sesión.
- Secuencias de comandos entre sitios (XSS).

- Referencias inseguras y directas a objetos.
- Mala configuración de seguridad.
- Exposición de datos sensibles.
- Falta de control de acceso a nivel de función.
- Solicitud de falsificación entre sitios (CSRF).
- Usando componentes vulnerables conocidos.
- Redirecciones no validadas.

6. Arquitectura de red

FortiWeb estará implementado en modo Reverse Proxy, al tener capacidades de firewall específicas de TCP y HTTP, FortiWeb no está diseñado para proporcionar seguridad a aplicaciones que no son HTTP, debe implementarse detrás de un firewall, como FortiGate, que se enfoca en la seguridad de otros protocolos que pueden reenviarse a los servidores back-end, como FTP y SSH.

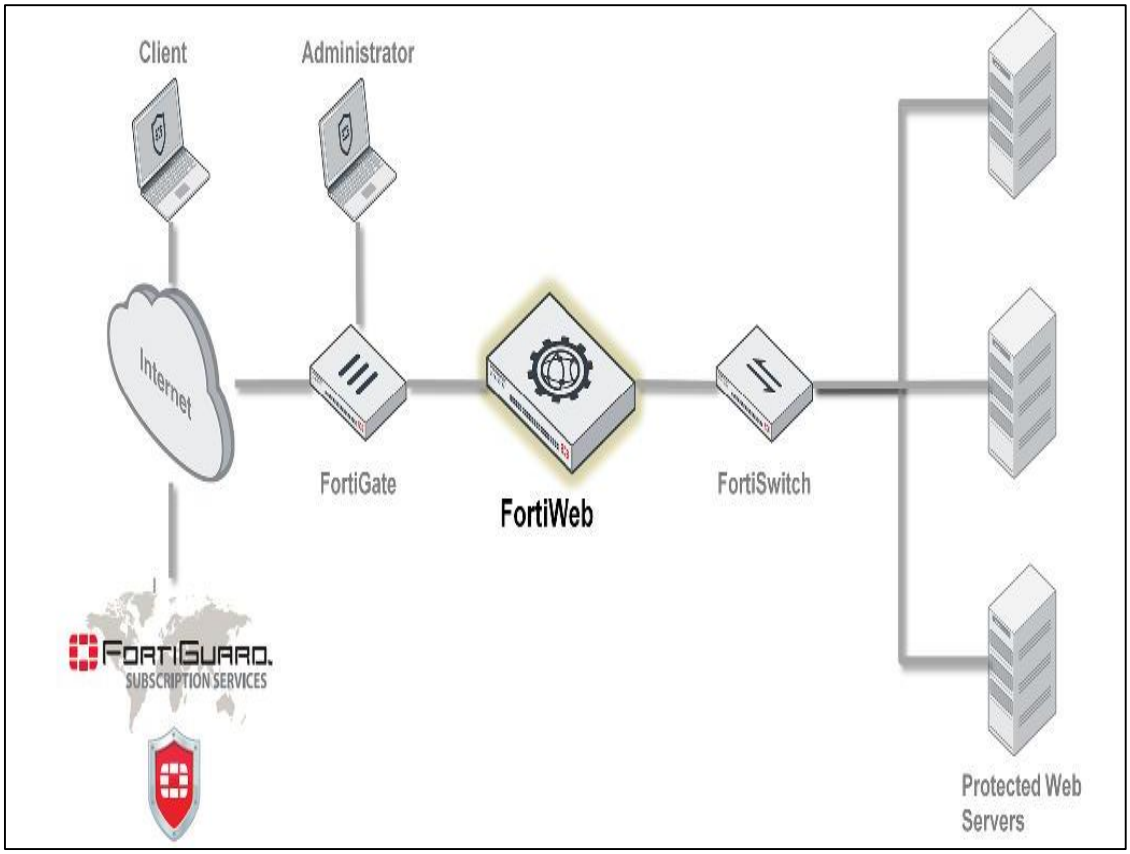

Figura 7. Arquitectura de red

7. Ataques de inyección SQL

7.1 Inyección SQL

Esta vulnerabilidad consiste en inyectar código SQL invasor dentro del código SQL programado con el objetivo de alterar la funcionalidad del sistema. Este tipo de intrusión normalmente es de carácter malicioso y, por tanto, un problema de seguridad informática. Un programa o sistema web que no haya sido validado de forma correcta podrá ser vulnerable y la seguridad del sistema (base de datos) no podrá ser asegurada.

La intrusión se puede llevar a cabo al ejecutar un programa vulnerable, ya sea, en ordenadores de escritorio o bien en sitios Web. La vulnerabilidad se puede producir cuando el programador use parámetros a ingresar por parte del usuario, para realizar una consultar de la base de datos. En esos parámetros es donde se puede incluir el código SQL intruso para que sea ejecutado en dicha consulta.

El objetivo más común del código intruso es extraer toda la información posible de la base de datos, aunque tienen más usos como puede ser el iniciar sesión con la cuenta otro usuario, subir una shell al servidor, etc.

A continuación, veremos como el sistema DVWA es vulnerable a este tipo de ataque, Una forma rápida y sencilla de ver si la página tiene una vulnerabilidad del tipo SQL Injection es introducir un **apóstrofe**, en el caso de tener este tipo de vulnerabilidad va a devolver un error de SQL:

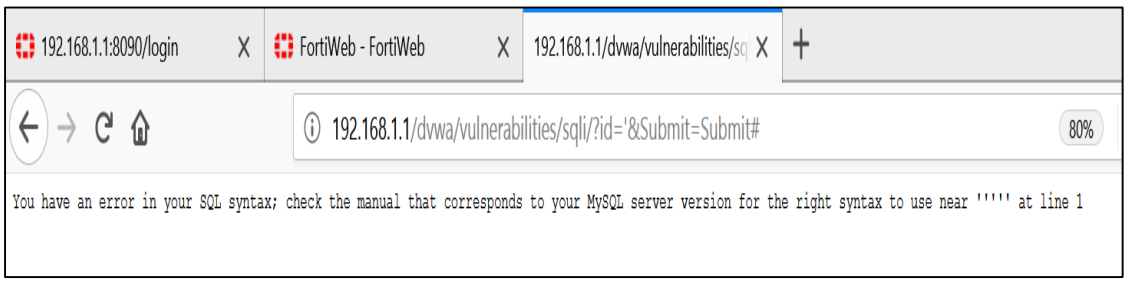

Figura 8. Vulnerabilidad sql

En la anterior figura se observa que el mensaje devuelto por el servidor es que hay un error de sintaxis en la consulta de SQL, por lo tanto, es vulnerable y se puede proceder con una inyección. Hay que tener en cuenta que no en todos los casos aparece el error, dependerá del tipo de vulnerabilidad que tenga, ya que en inyecciones avanzadas no muestran nada enviando una comilla.

A continuación, se muestran consultas básicas que se usan al hacer este tipo de ataque:

- 1' OR 1=1--'
- 1' OR '1' = '1
- ' OR"='
- ' OR 1=1--'
- ' OR 0=0 --'
- ' OR 'x'='x

Al probar la inyección con **1' OR 1=1--'** se muestra todos los usuarios que hay en la base de datos:

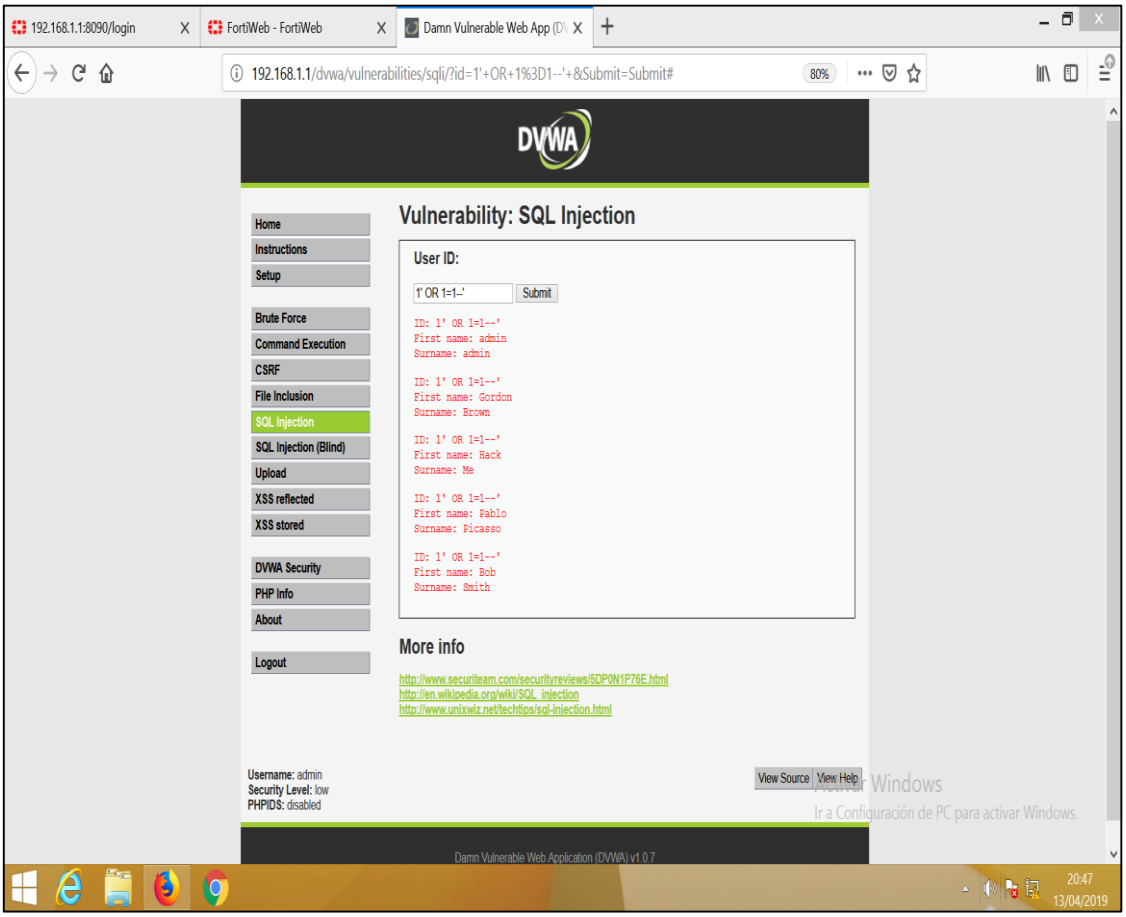

Figura 9. Muestra de información después de ataque

Al usar probar la inyección **% OR '0'='0** tambien se muestras todos los usuarios de la base de datos:

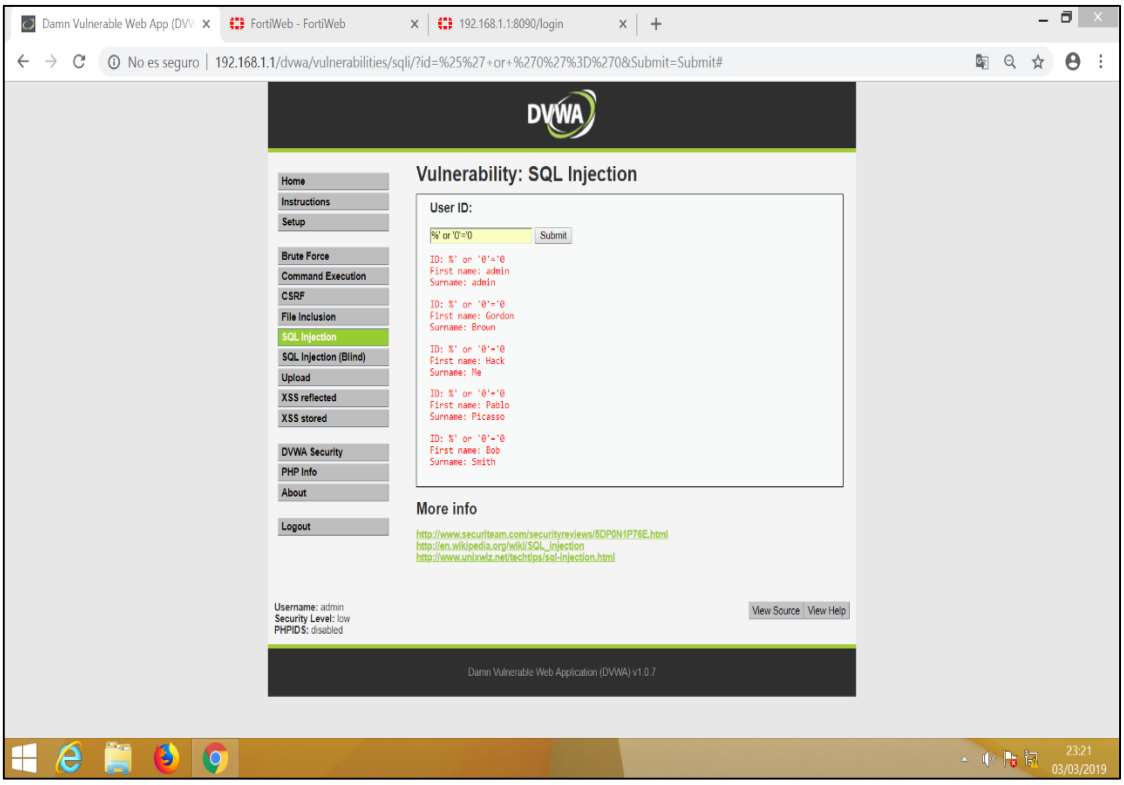

Figura 10. Muestra de información después de ataque

Vamos averiguar el nombre de la base de datos como de las tablas que existen con el código **' union select database(),group_concat(table_name) from information_schema.tables where table_schema=database()#**

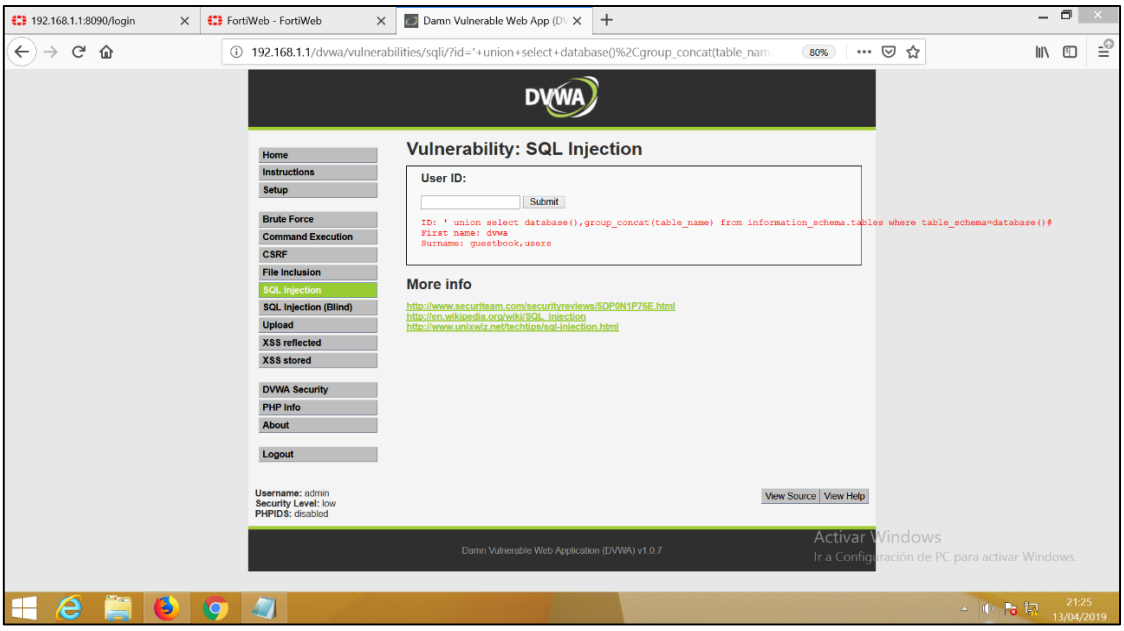

Figura 11. Muestra de información después de ataque

Tenemos como resultado base de datos llamada dvwa y tablas guestbook, users. Ahora obtendremos los nombres de las columnas de la tabla usuarios con el código **' union select 1,group_concat(column_name) from information_schema.columns where table_schema='dvwa' and table_name='users'#**

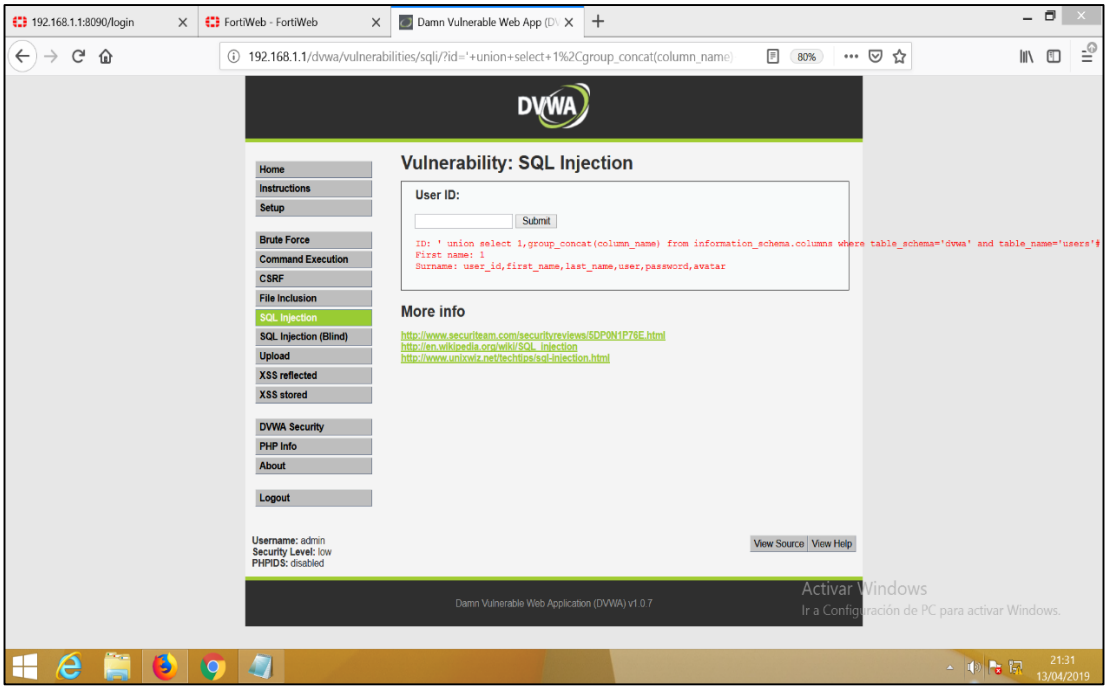

Figura 12. Muestra de información después de ataque

Por ultimo usaremos el codigo **'or 1=0 union select user,password from dvwa.users#** para poder obtener los usuarios y sus claves obteniendo:

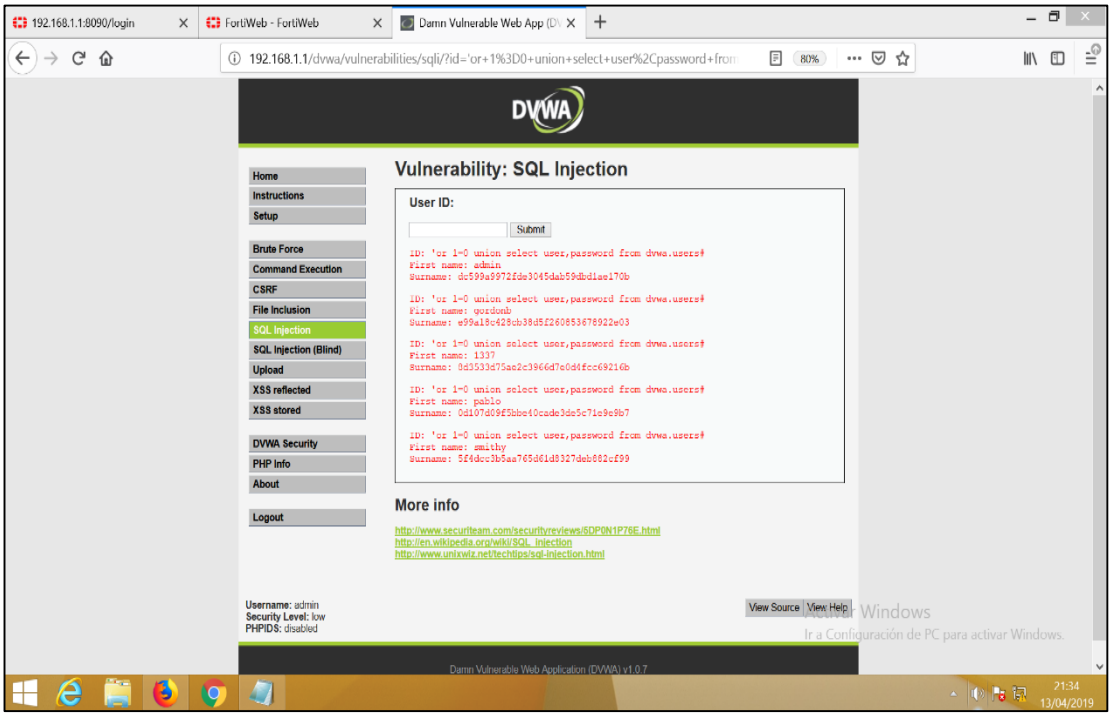

Figura 13. Muestra de información después de ataque

Como logramos observar obtuvimos las claves de los usuarios cifrados, ahora copiaremos la clave correspondiente a Pablo para interntar descifrarlos en alguna página web que ofresca el servicio como https://md5online.es/.

Figura 14. Descifrando clave de usuario

Con esto podemos validar como una aplicación sin buenas practicas de seguridad puede poner en riesgo el activo más importante de una organización su **información**.

7.2 Mitigación de inyección SQL

Fortiweb es capaz de bloquear ataques de inyección sql a continuación vamos a proteger el sistema DVWA de este tipo de ataques, usaremos la siguiente consulta sql **'or 1=0 union select user,password from dvwa.users#** para verificar si el ataque lograr su objetivo o no:

Figura 15. Explotando vulnerabilidad

Como vemos Fortiweb bloquea el ataque del cliente al ejecutar el codigo de inyección sql.

Figura 16. Bloqueo de FortiWeb

Como segunda prueba usaremos la consulta de inyección sql **% OR '0'='0** y veremos que sucede:

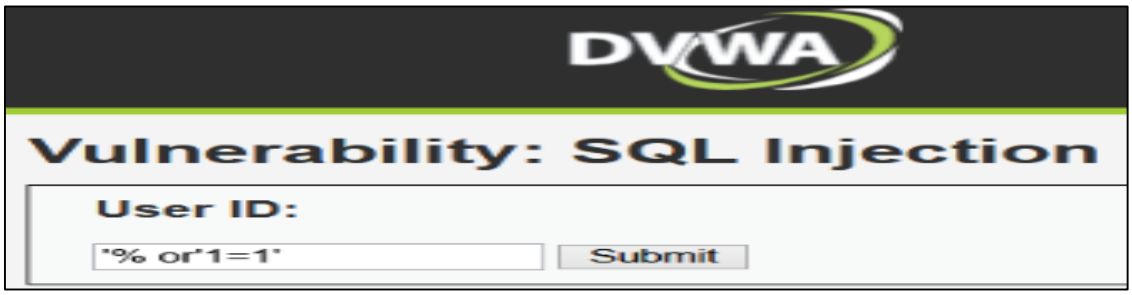

Figura 17. Explotando vulnerabilidad

Como vemos nuevamente Fortiweb bloqueo el ataque sql.

Figura 18. Bloqueo de FortiWeb

8. Ataque de perdida de autenticación y gestión de sesiones

8.1 Perdida de autenticación y gestión de sesiones

Englobaría a todo lo referido en el tratamiento y técnicas de protección de credenciales y cómo las implementan las aplicaciones. Con lo cual, tendría que ver con fallos de implementación como serían:

- La utilización de canales no cifrados para transmitir credenciales.
- No proteger debidamente las credenciales de usuarios.
- Mala gestión en la recuperación de credenciales.
- Exposición de los ID de sesión (Cookies de sesión).

Estos fallos nos podrían llevar a que usuarios malintencionados o algún atacante se hiciera con las credenciales de un usuario o un incluso un administrador del sistema con lo que ello conllevaría.

Serían muchas las formas de conseguir explotar este tipo de vulnerabilidades puesto que son muchos los fallos los que podrían llevar a que un atacante se pudiera hacer con las credencial ilícitamente, dichos fallos podrían ser:

- No utilización de HTTPS.
- El envío de contraseñas en texto claro.
- URL con ID de sesión implícitas.
- Credenciales almacenadas sin cifrar.

Veremos como el sistema DVWA no usa una conexión segura y como un usuario con algunos conocimientos podria obtener la cookie que el servidor le asigna al iniciar sesión.

Para este tipo de ataque haremos un man in the midle usando la herramienta **Burp Suite** la siguiente imagen muestra la cookie proporcionada por el sistema la cual es una cookie PHPSESSID.

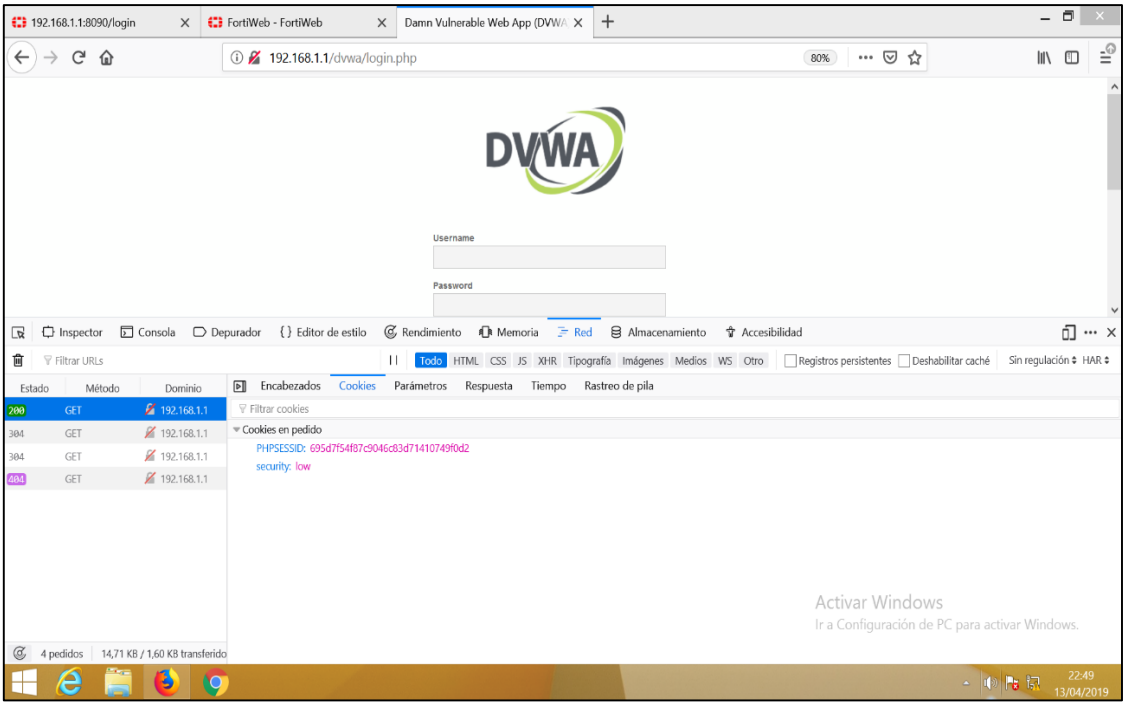

Figura 19. Detectando cookie del sistema

Realizaremos el login con el usuario admin y vemos lo siguiente.

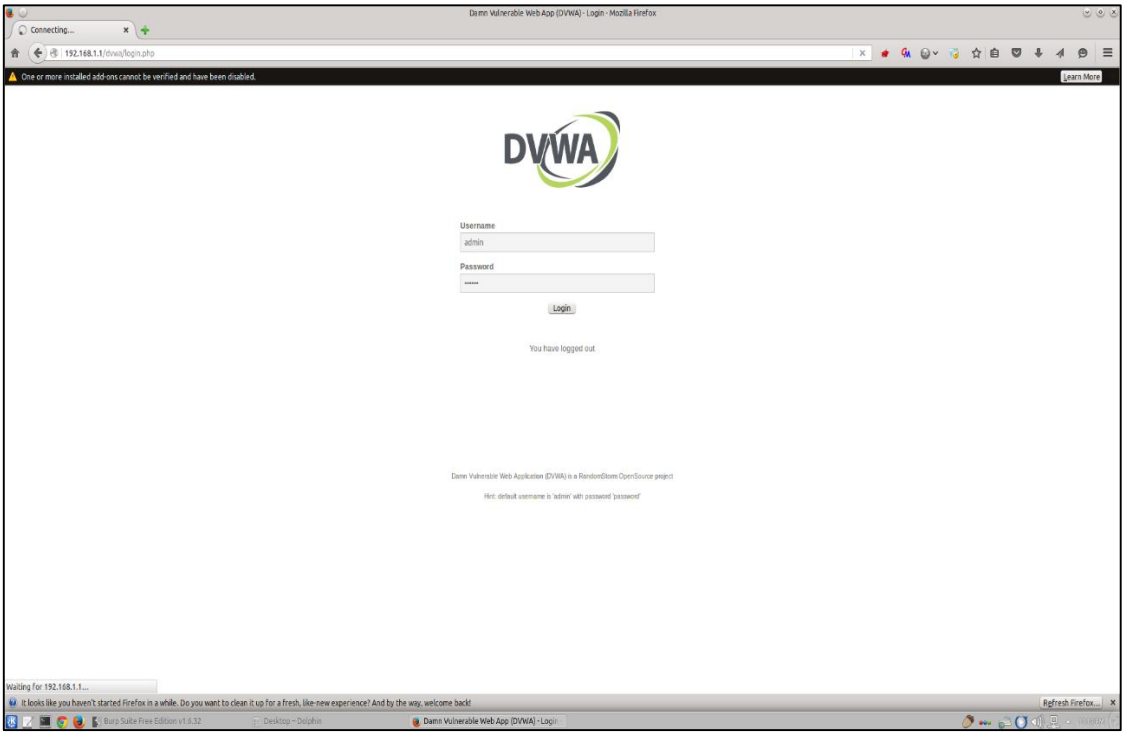

Figura 20. Realizando login en el sistema

A continuación vemos como podemos capturar facilmente las **credenciales** y tambien vemos la **cookie** en texto plano enviados por el metodo POST al servidor backend.

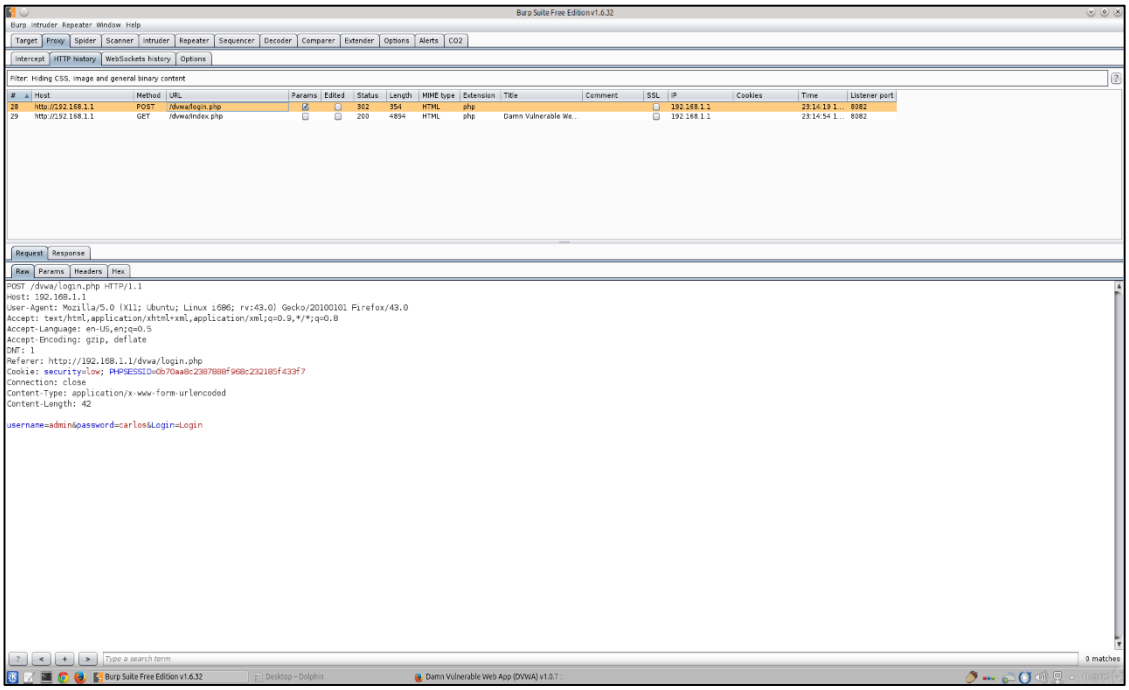

Figura 21. Interceptando tráfico con Burp Suite

Ahora vamos a copiar la cookie **PHPSESSID=0b70aa8c2387888f968c232185f433f7** vamos a instalar una extensión para el navegador Mozzila Firefox llamado **Cookie Manager** en el cual vamos a pegar nuestra cookie**.**

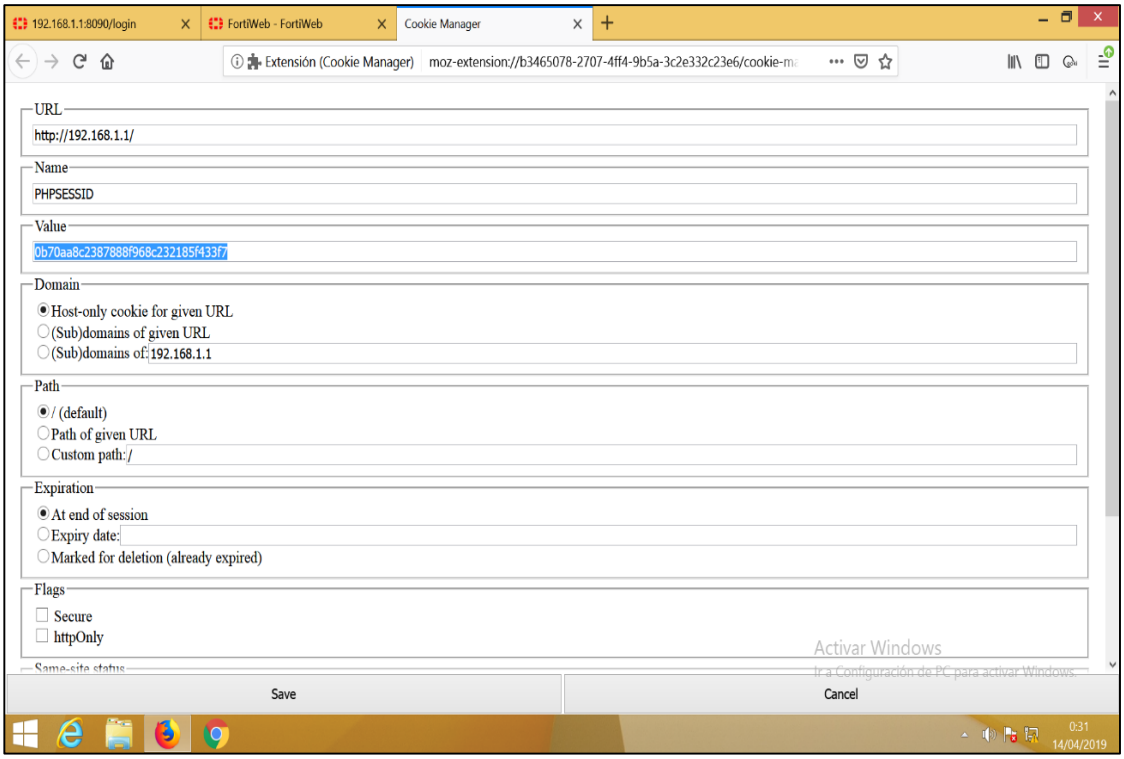

Figura 22. Modificando valor de cookie en la extensión

En otro equipo cliente ingresamos la url del sistema web 192.168.1.1/dvwa/instructions **"saltamos la página login.php"**

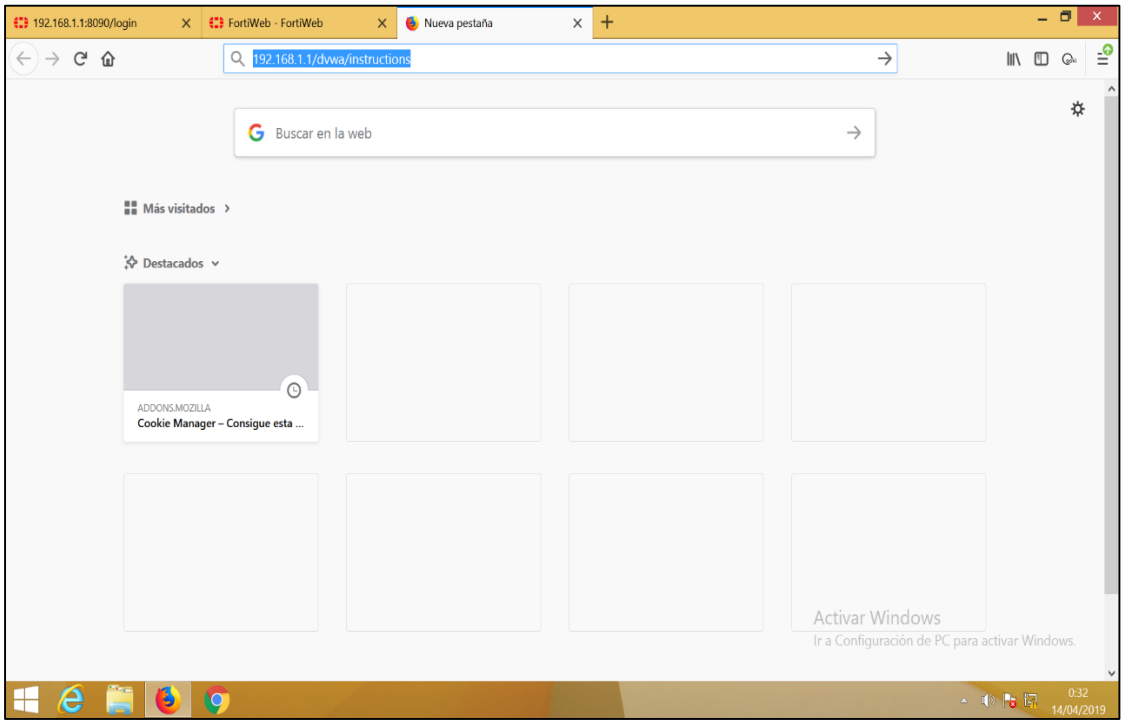

Figura 23. Ingresando al sistema web

Validamos como sin ingresar las credenciales logramos acceder al sistema web.

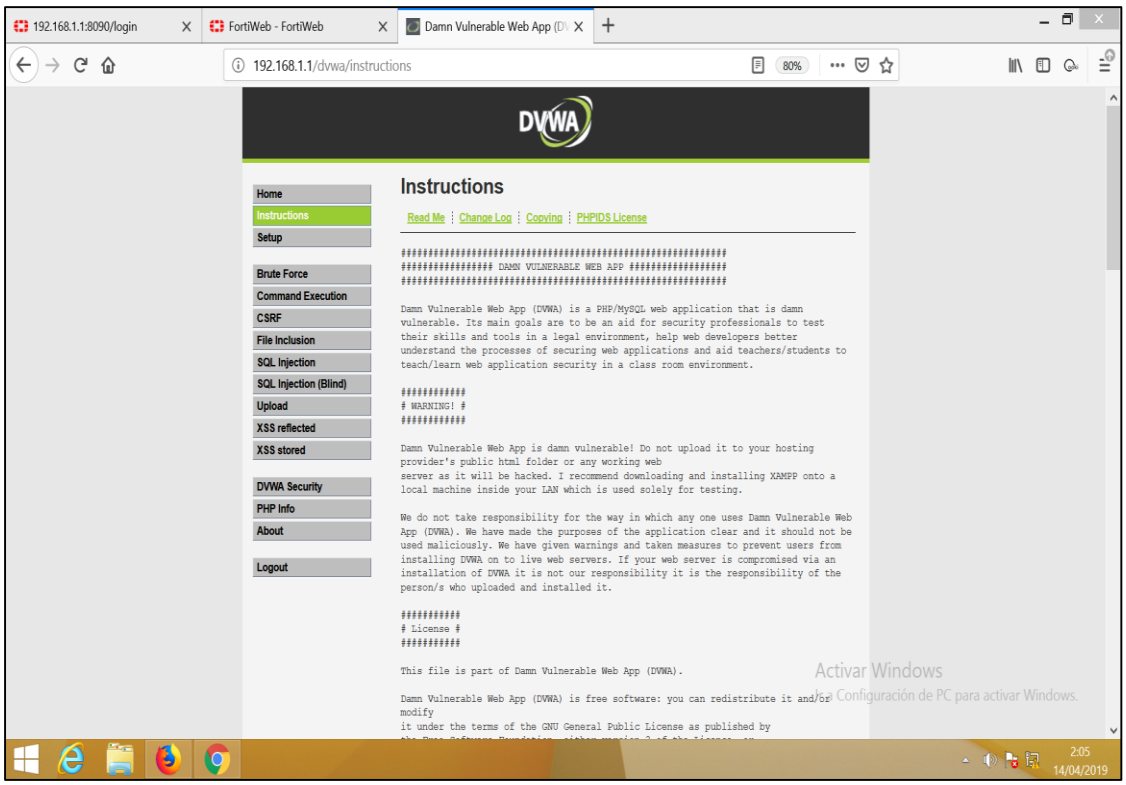

Figura 24. Ingreso correcto al sistema

8.2 Mitigación de perdida de autenticación y gestión de sesiones

Fortiweb nos permite bloquear ataques basados en cookies y aplicarlas en un perfil de protección. Por ejemplo, una política puede habilitar la detección de envenenamiento de cookies, cifrar las cookies emitidas por un servidor de servicios de fondo y agregar atributos de seguridad a las cookies.

Fortiweb nos brinda la opción de insertar una cookie de seguimiento adicional a la cookie proporcionada por el servidor backend a continuación veremos la **cookiesession1** añadida por nuestro waf.

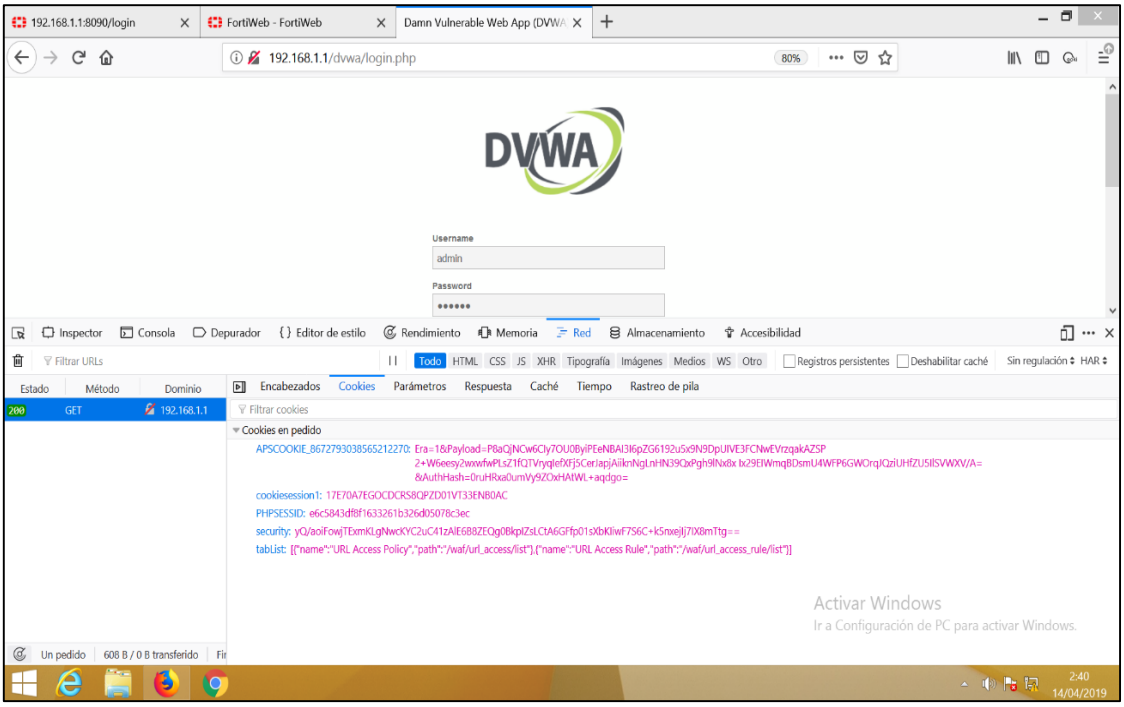

Figura 25. Cookie insertada por Fortiweb

Como vemos en la imagen anterior se añade una cookiesession1, pero adicionalmente Fortiweb brinda la opción de cifrar la cookie que envía el servidor backend dándonos la segura ante posibles ataques de envenenamiento de cookies y otros tipos de ataques, Fortiweb adicional proporciona estas técnicas para mitigar este tipo de ataque Autenticación https, doble factor de autenticación and kerberos support), página de inicio y orden de páginas, enmascaramiento de información sensible en los registros y Padding oracle protection.

Ahora veremos como agregamos una autenticación adicional antes del inicio en el sistema web, en un **Windows Server 2012** tendremos creados los usuarios que accederán a los recursos, si el usuario no puede autenticarse no tendrá acceso al sistema web.

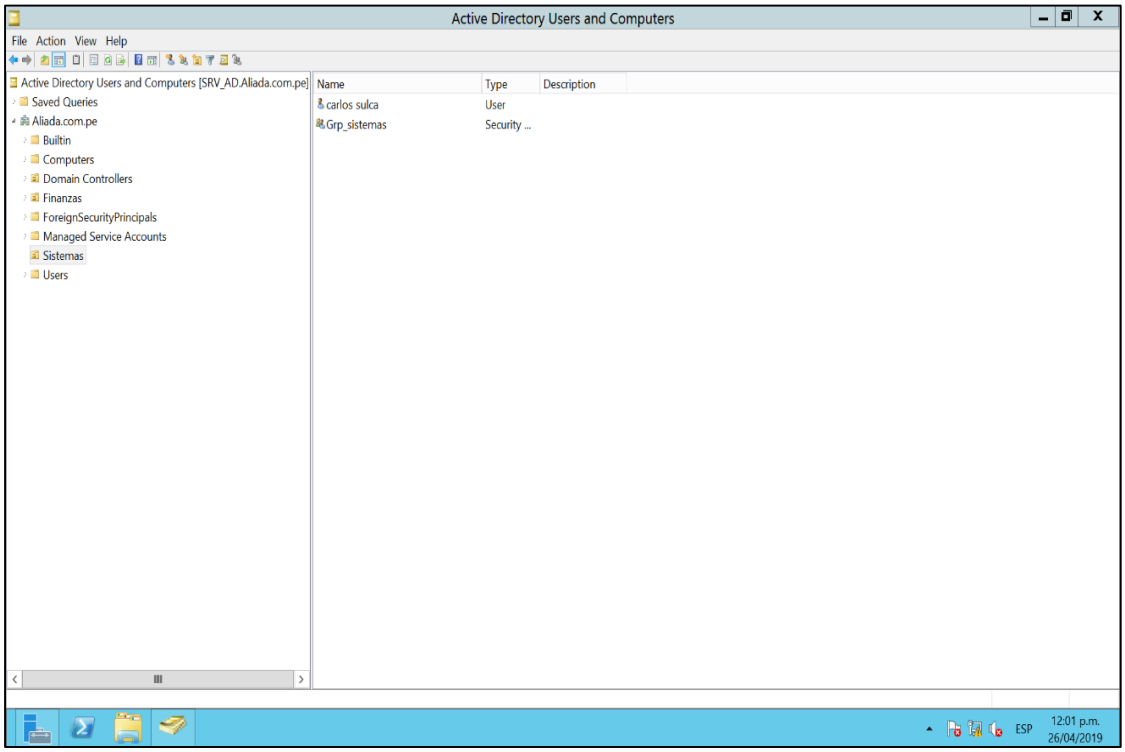

Figura 26. Usuarios en Windows server

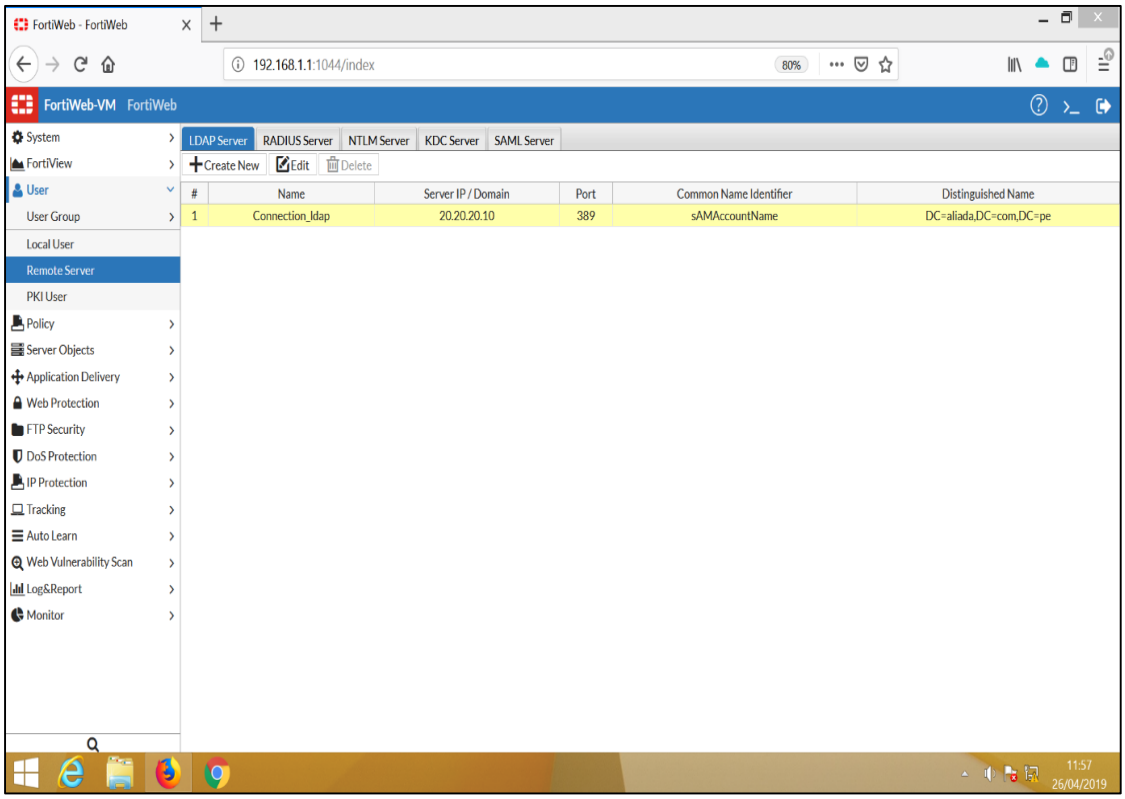

Figura 27. Conexión ldap a Windows server

Podremos visualizar que al solicitar las credenciales recibimos el código de estado **401 unauthorized** que significa que se requieren sus credenciales.

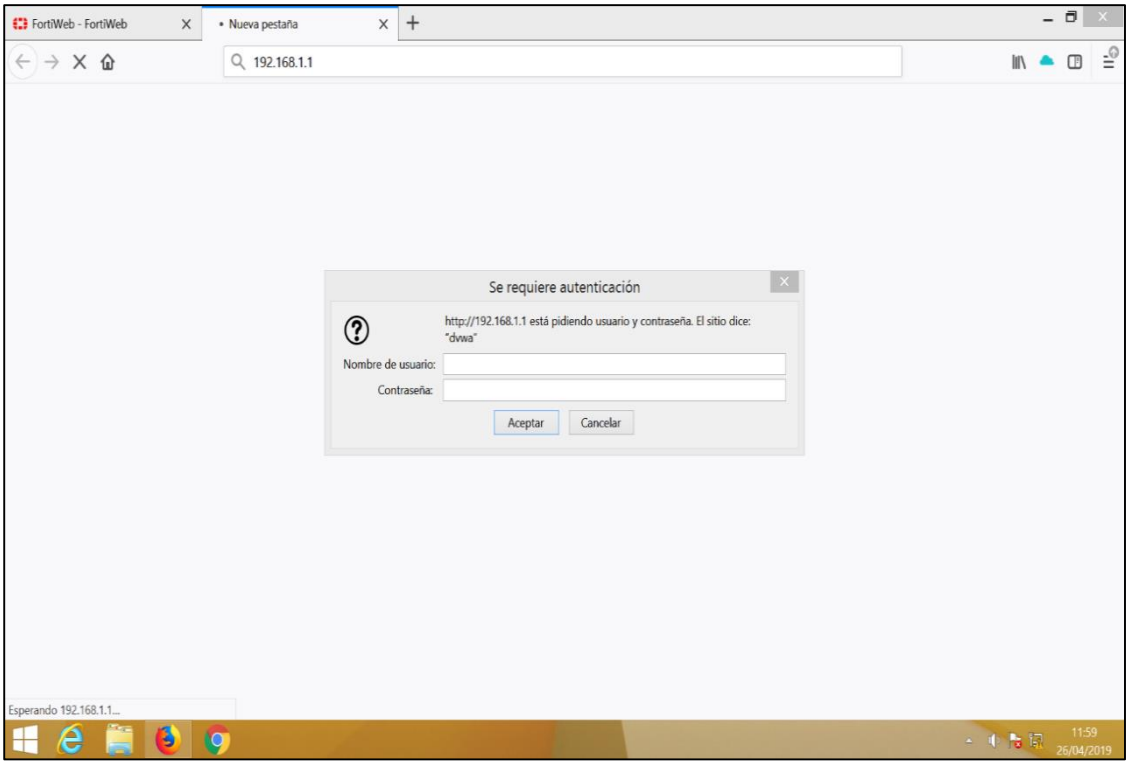

Figura 28. Autenticación adicional agregada por Fortiweb

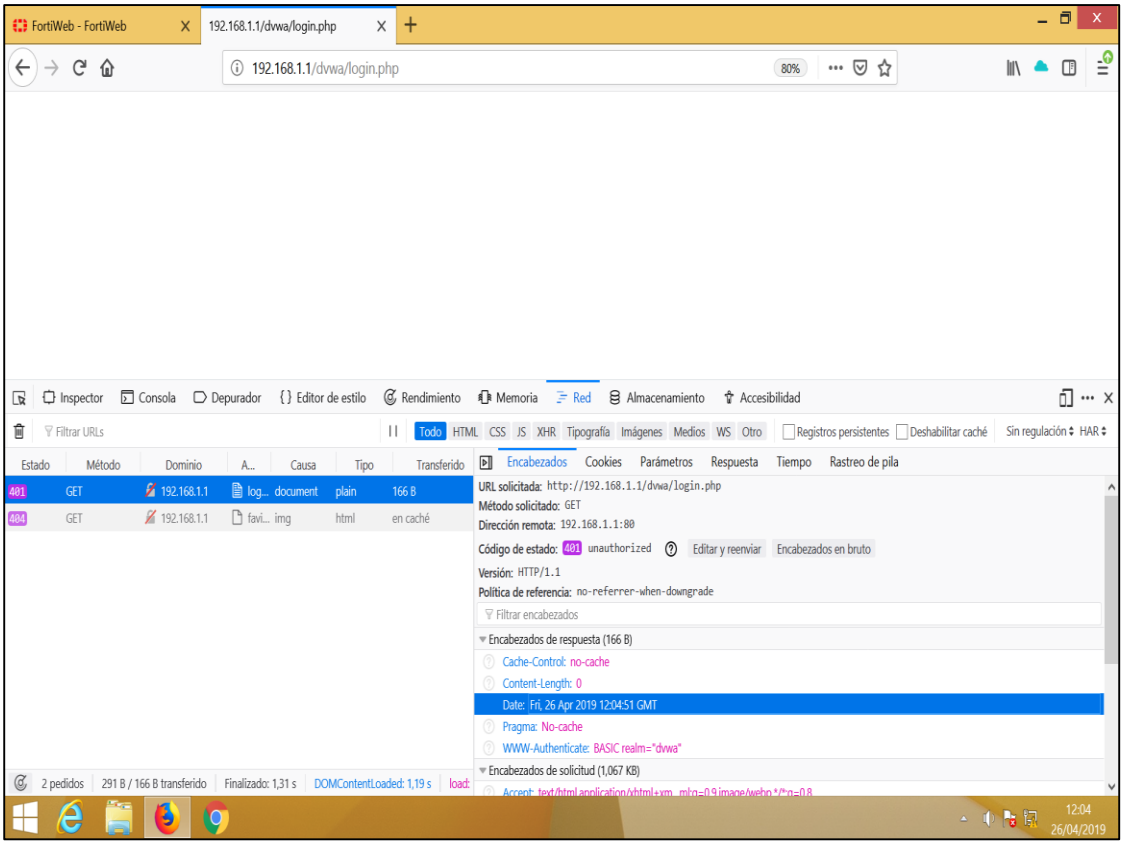

Figura 29. Código de estado 401

9. Ataque secuencia de comandos en sitios cruzados XSS

9.1 Secuencia de comandos en sitios cruzados XSS

Es importante tener en cuenta que, con esta vulnerabilidad, los atacantes explotan la confianza que un usuario tiene en un sitio en particular, y esto nos da una dimensión del impacto que puede tener.

Este tipo de vulnerabilidad puede ser explotada de dos maneras: de forma reflejada y de forma almacenada. A continuación, haré una breve explicación de cada una.

XSS reflejada

Consiste en modificar valores que la aplicación web usa para pasar variables entre dos páginas. Un clásico ejemplo de esto es hacer que a través de un buscador se ejecute un mensaje de alerta en JavaScript. Con XSS reflejado, el atacante podría robar las cookies para luego robar la identidad, pero para esto, debe lograr que su víctima ejecute un determinado comando dentro de su dirección web.

Para esto, los cibercriminales suelen enviar correos engañosos para que sus víctimas hagan clic en un enlace disfrazado y así se produzca el robo.

Para nuestro ejemplo usaremos un equipo **Kali Linux** para poder generar un archivo malicioso llamado **documento.php**

Figura 30. Generando archivo malicioso

Este documento será subido en el sistema **DVWA** en la opción de **Upload,** pero antes de realizar la carga del archivo dejaremos en modo de escucha la puerta trasera que estamos generando desde metasploit a

continuación se ve lo indicado.

```
msf exploit(multi/handler) > exploit

[-] Handler failed to bind to 192.168.1.1:8080:-  -
[*] Started reverse TCP handler on 0.0.0.0:8080
```

Figura 31. Ejecutando exploit

Ahora vamos al sistema DVWA y cargaremos el archivo **documento.php** en **Upload** veremos el mensaje de carga completada**.**

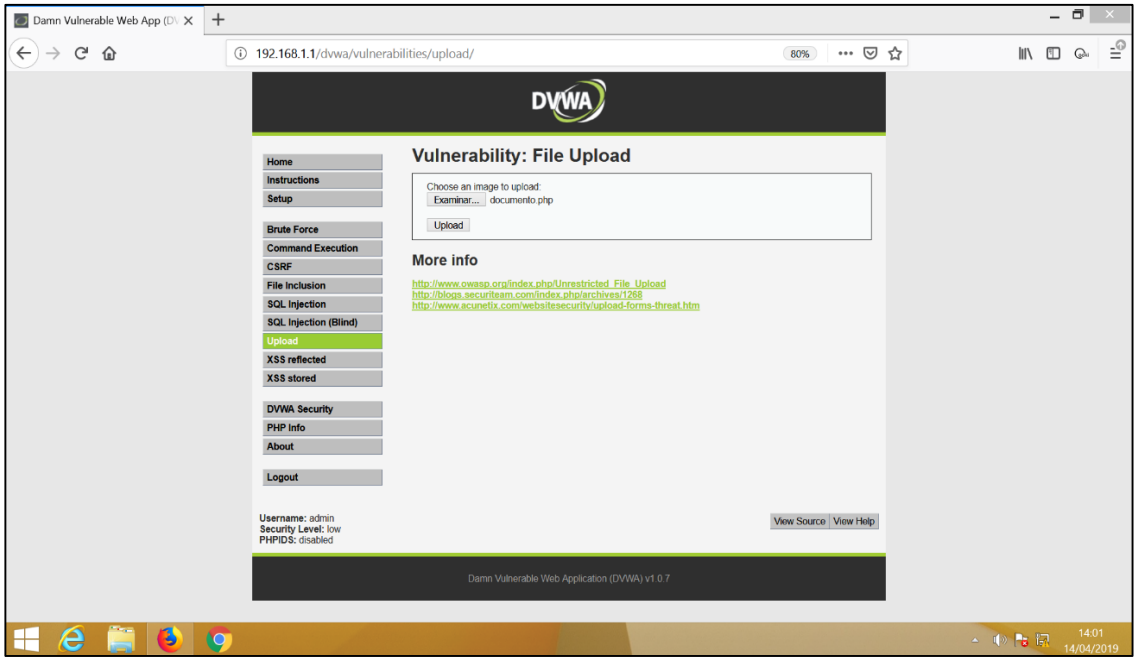

Figura 32. Subiendo archivo malicioso

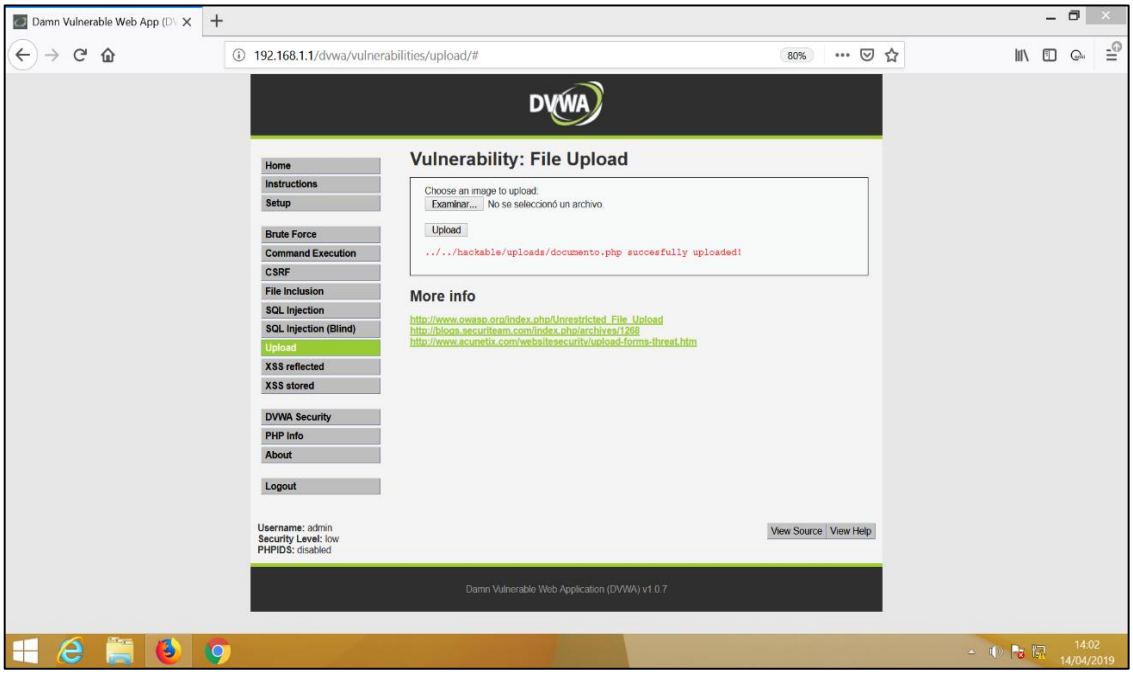

Figura 33. Mensaje de carga correcta

Ahora iremos a **XSS reflected** y agregaremos el siguiente script **<script>window.location="http://192.168.1.1/dvwa/hackable/uploads/documento.php"</script>** y daremos clic en **Submit.**

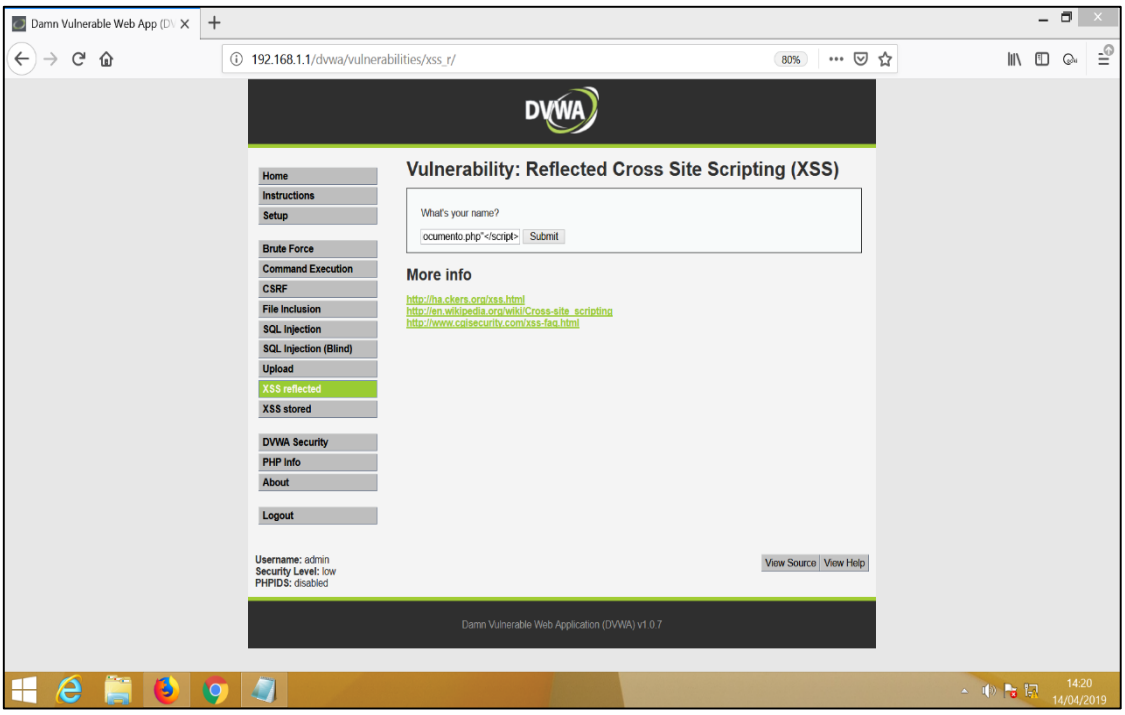

Figura 34. Ejecutando script malicioso

Veremos como la página web se queda cargando sin mostrar información alguna.

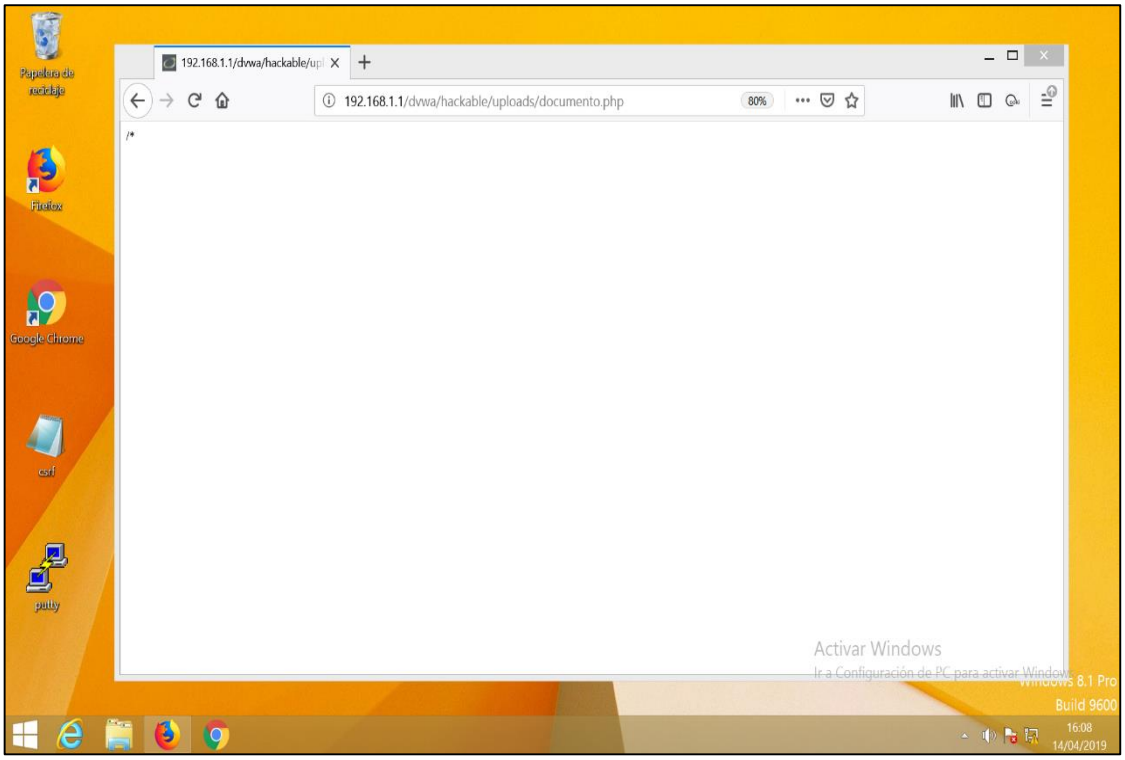

Figura 35. Página no muestra información

Vamos a Kali Linux para ver la apertura de la conexión hacia el servidor backend.

Figura 36. Conexión establecida con el servidor

Ahora ejecutamos unos comandos de prueba donde obtenemos información del servidor backend.

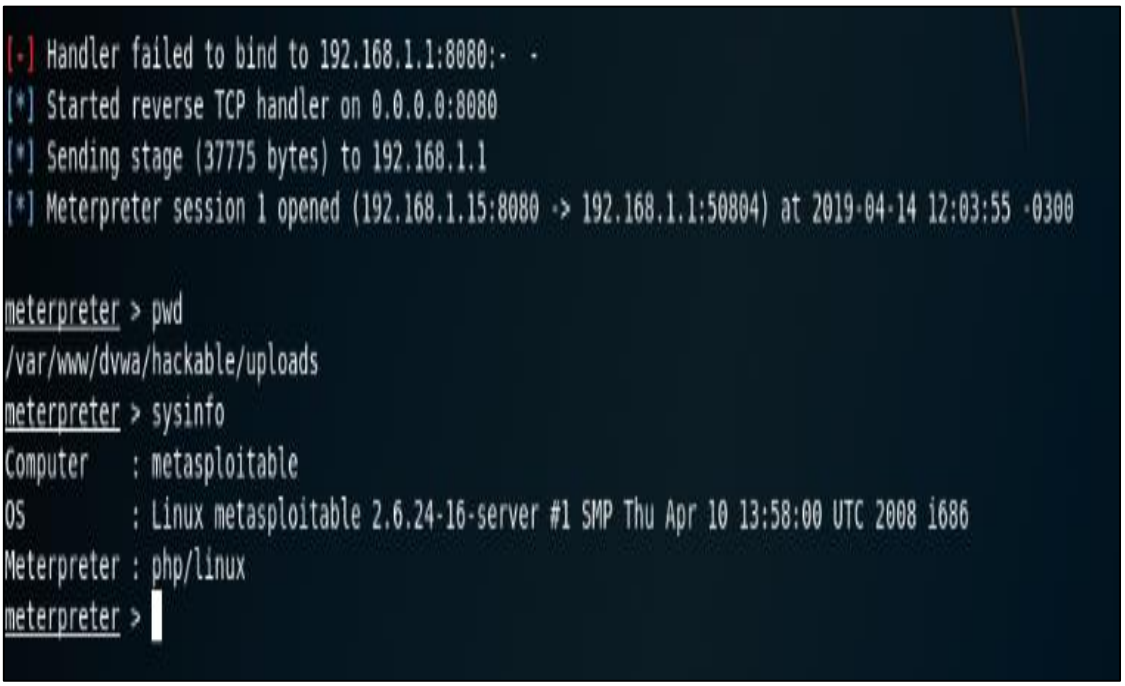

Figura 37. Ejecución de comandos al servidor

Con estos sencillos pasos, ejecutamos una puerta trasera en la máquina en un equipo cliente que nos permite acceder al servidor backend usando Metasploit.

XSS almacenada

Consiste en insertar código HTML (programación web) peligroso en sitios que lo permitan; de esta forma quedará visible a los usuarios que ingresen en el sitio modificado.

Inyectamos el siguiente script en **XSS stored** **<script>alert(document.cookie)</script>** daremos clic en **Sign Guestbook** para poder obtener la ok actual de nuestra sesión y veremos que sucede.

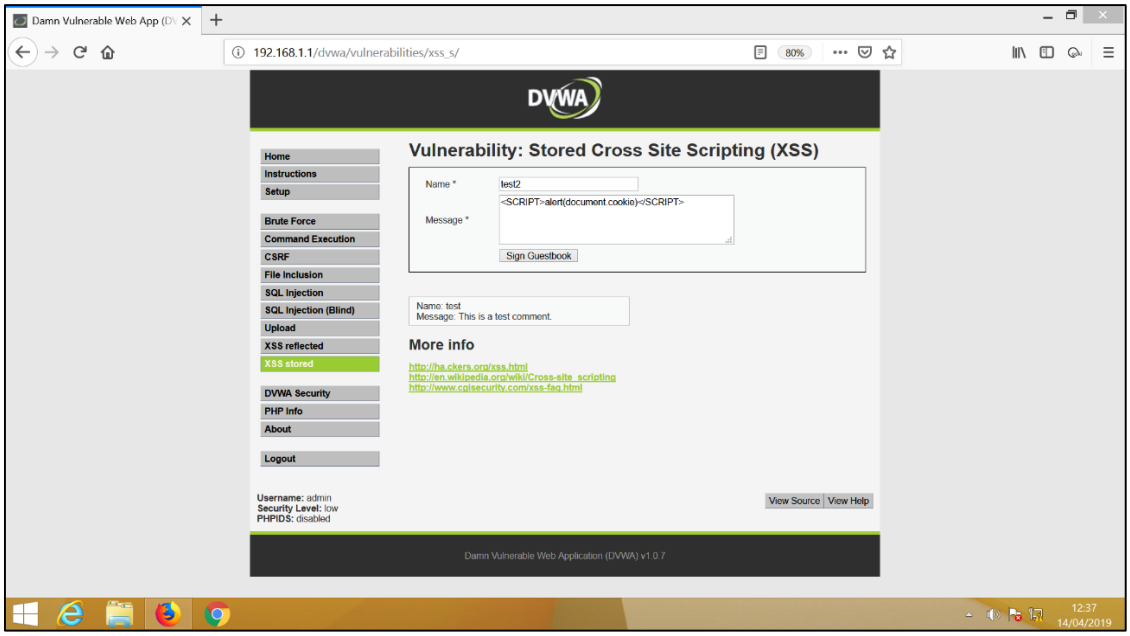

Figura 38. Ejecutando script malicioso

Obtenemos como resultado el valor de la cookie del usuario.

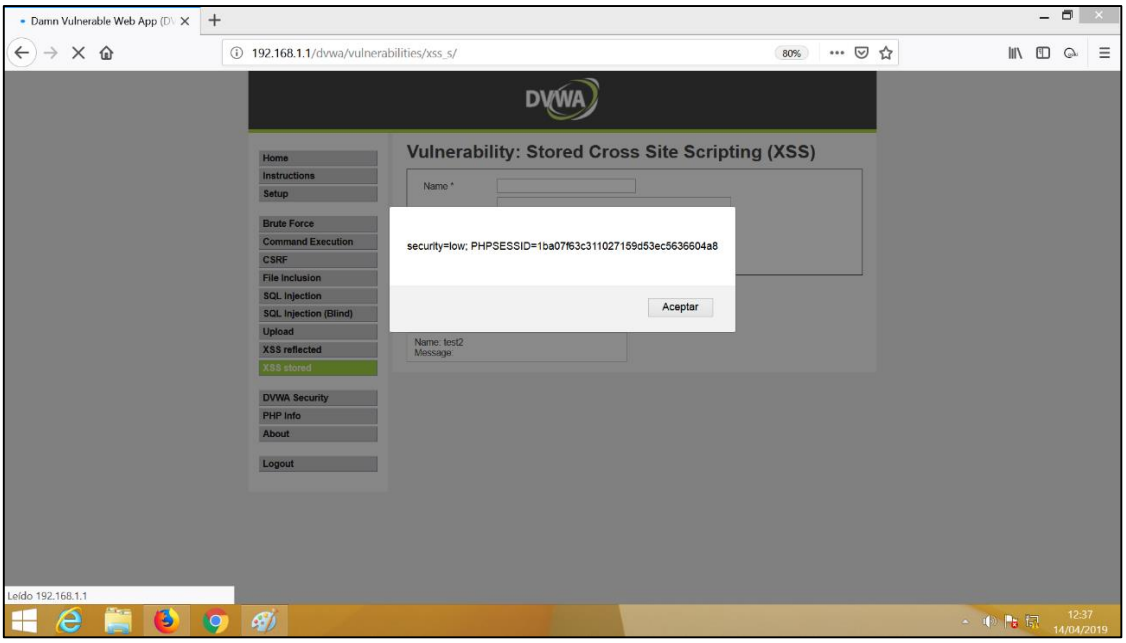

Figura 39. Obteniendo cookie después del ataque

Para la siguiente prueba iremos nuevamente a **XSS Stored** inspeccionaremos la página web y modificaremos ambas cajas de texto las cuales están limitadas a 50 palabras lo que haremos es modificar a 500.

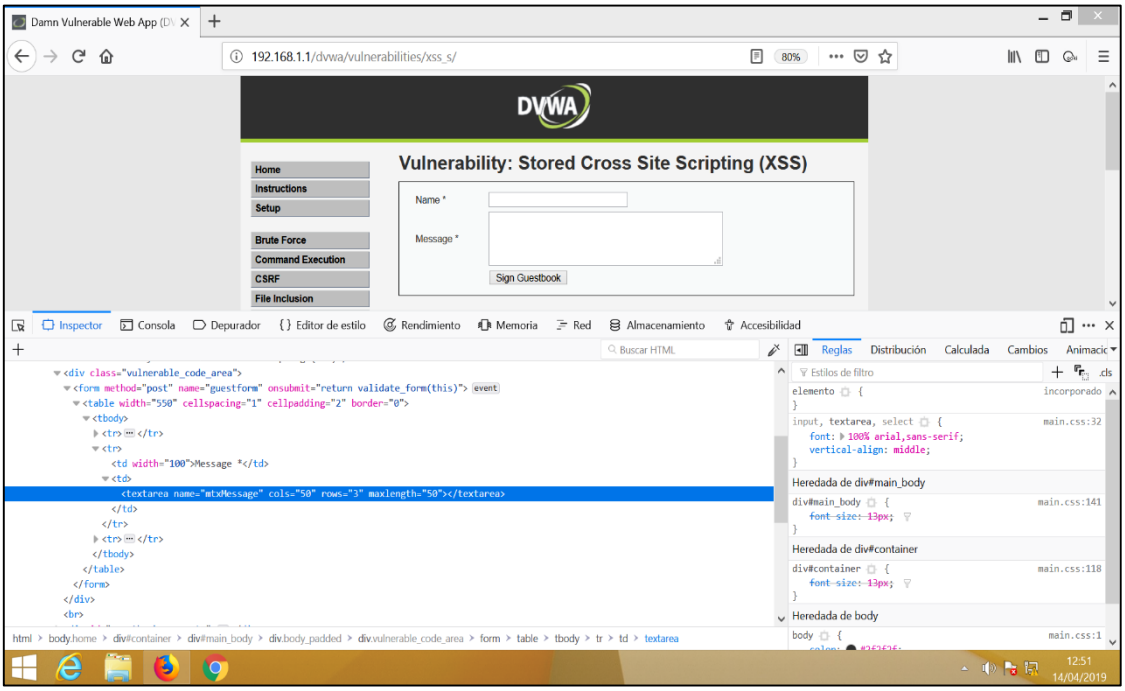

Figura 40. Modificando tamaño de campo

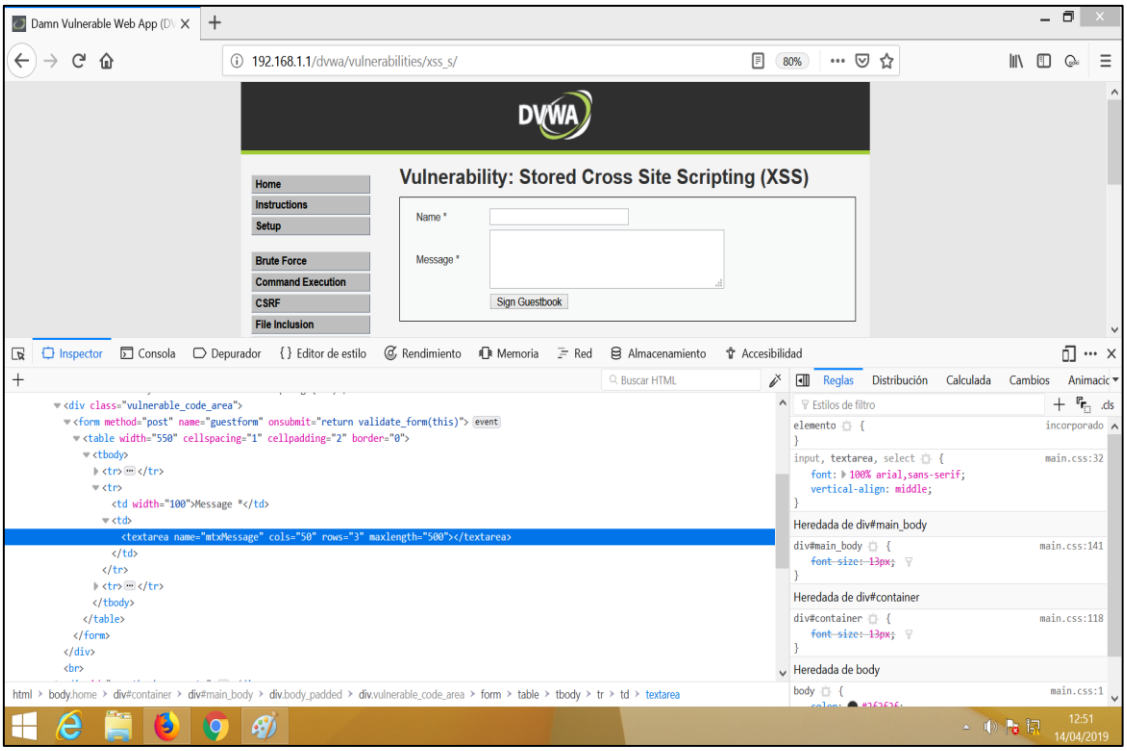

Figura 41. Nuevo tamaño de campo

Ahora con el campo admitiendo más cantidad de caracteres agregaremos el siguiente script:

<SCRIPT language="javascript">window.location="http://www.taringa.com";</SCRIPT>

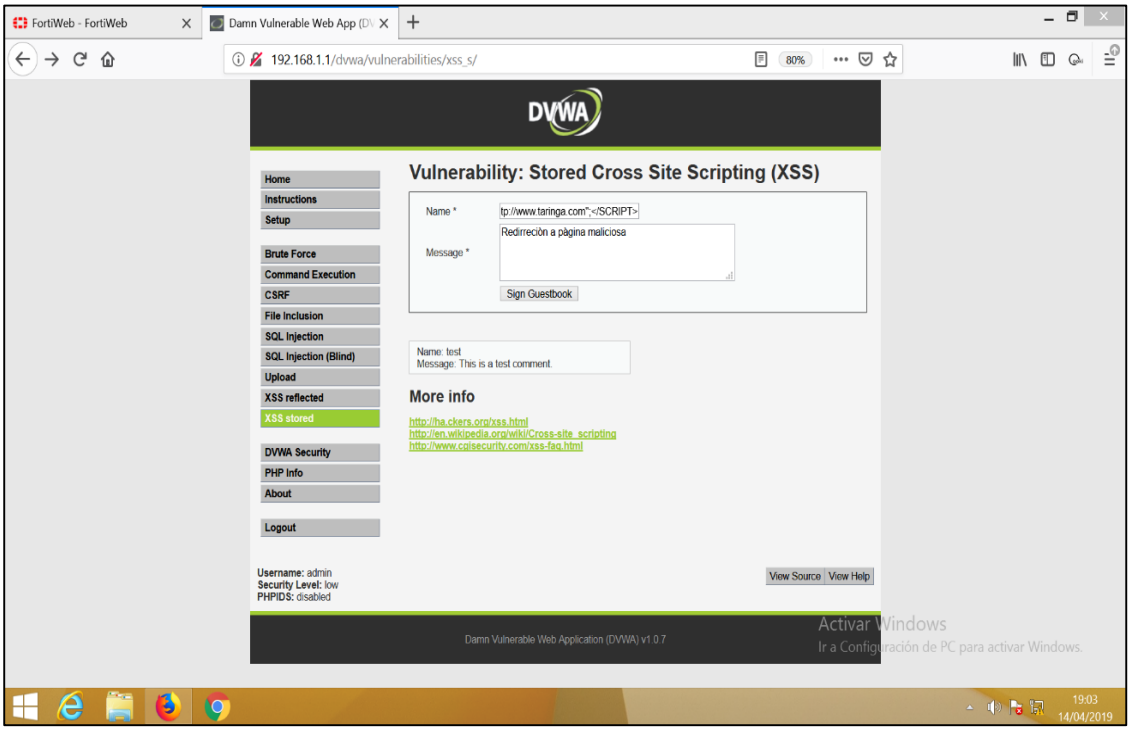

Figura 42. Ejecutando script malicioso

Al dar clic en **Sign Guestbook** veremos cómo intenta enviarnos a una página que podría ser manipulada por un atacante para robar información.

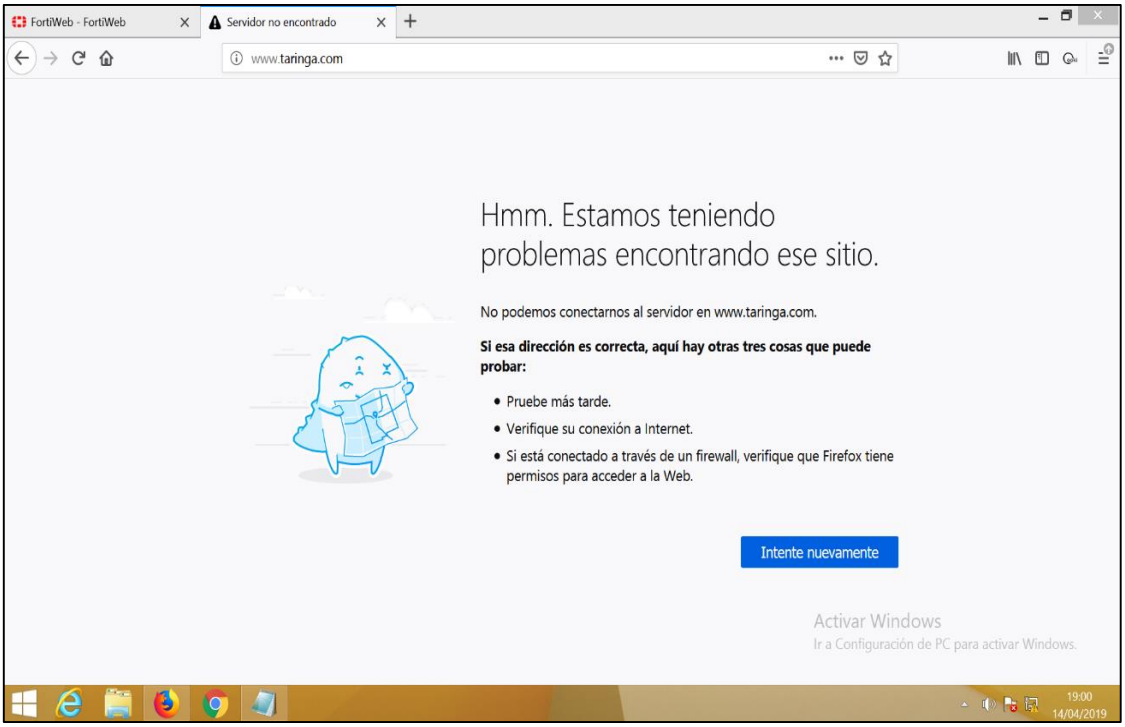

Figura 43. Script se ejecuta correctamente

9.2 Mitigación de ataques secuencia de comandos en sitios cruzados XSS

Fortiweb puede detectar muchos ataques de XSS y fugas de datos con firmas, estas firmas son actualizadas mediante **Fortiguard**.

Veremos como replicaremos el ataque **XSS reflected** y Fortiweb lo bloquea correctamente.

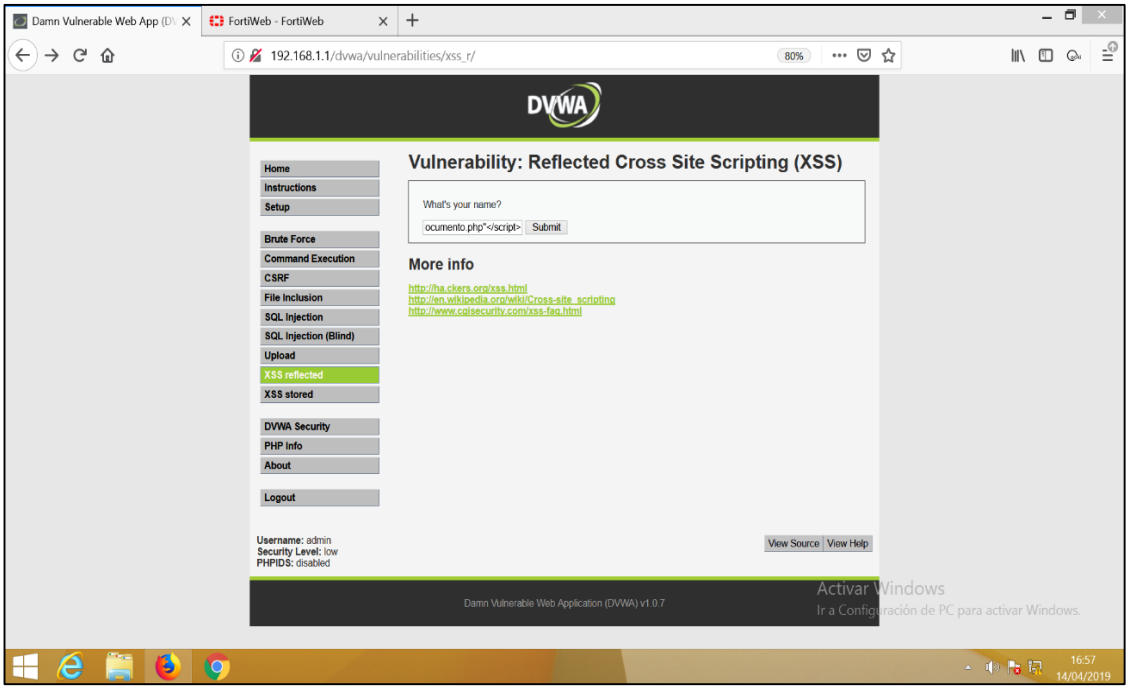

Figura 44. Ejecutando script malicioso

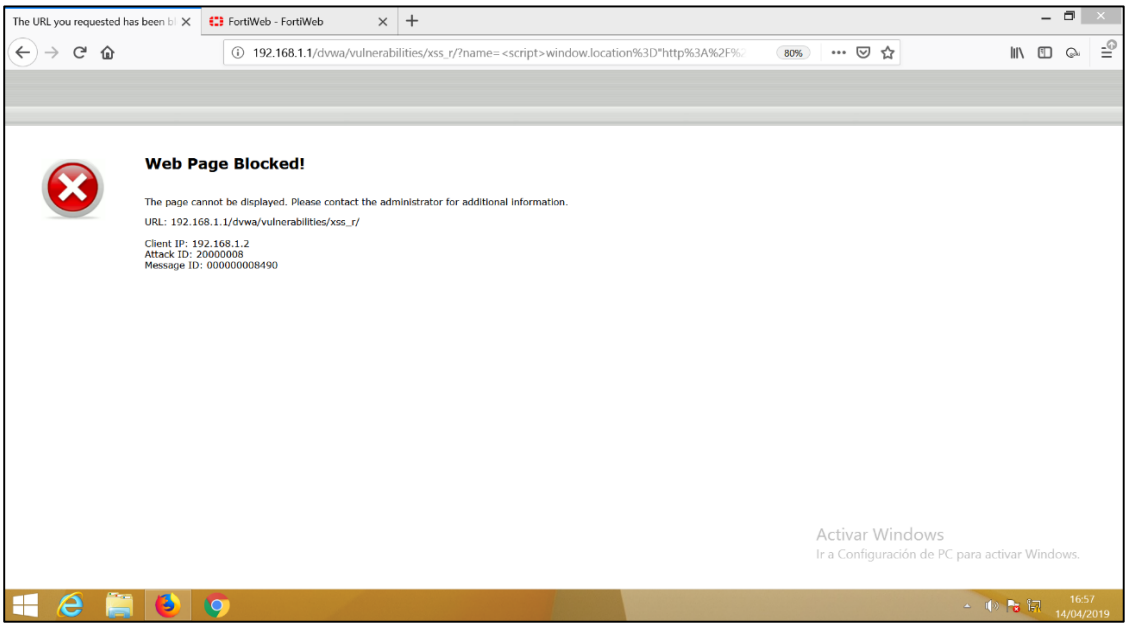

Figura 45. Bloqueo de FortiWeb

Ahora veremos cómo bloquea un ataque **XSS Stored** trataremos de obtener la cookie del usuario mediante el siguiente script **<SCRIPT>alert(document.cookie)</SCRIPT>** damos clic en **Sign Guestbook** y vemos como Fortiweb bloquea el ataque.

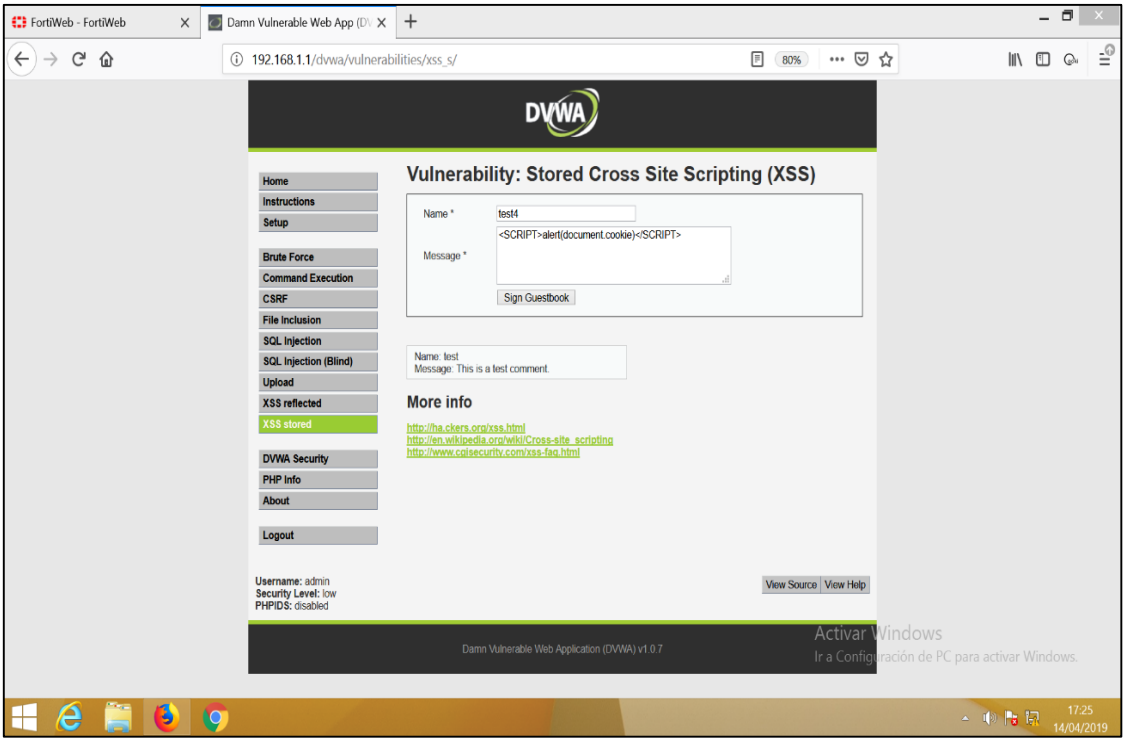

Figura 46. Ejecutando script malicioso

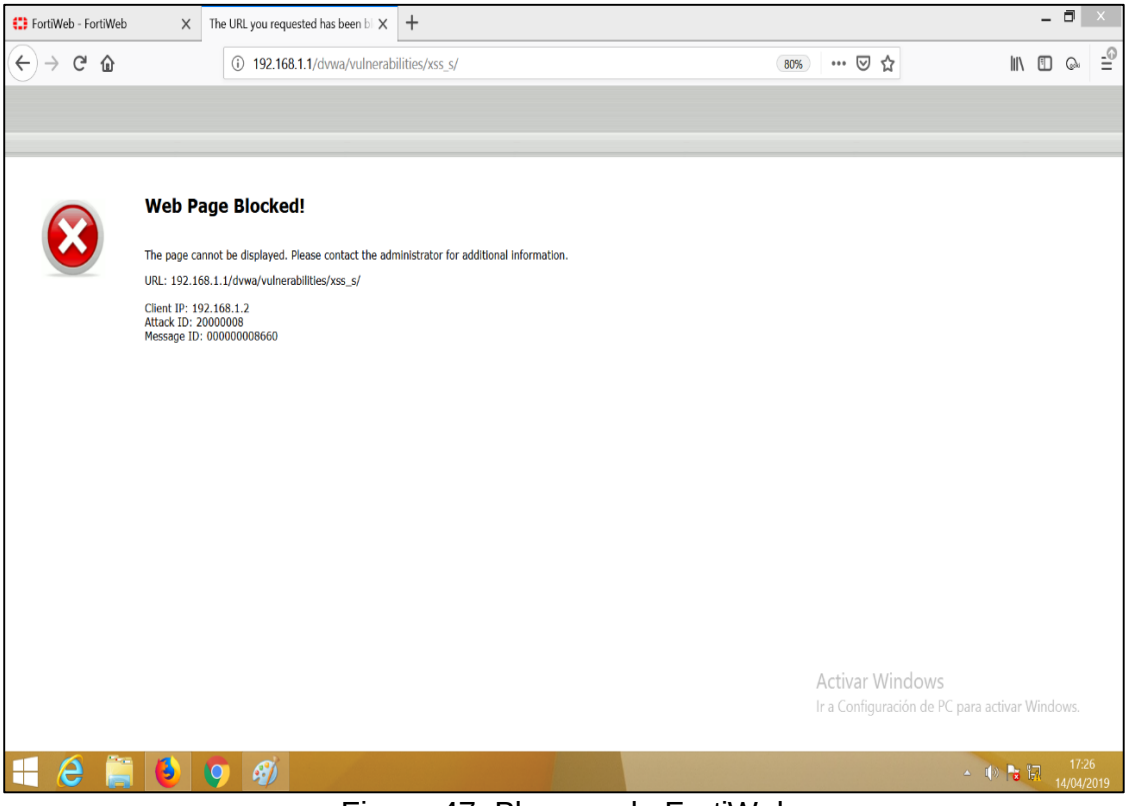

Figura 47. Bloqueo de FortiWeb

Ahora ejecutaremos nuevamente en XSS reflected el siguiente script **<SCRIPT language="javascript">window.location="http://www.taringa.com";</SCRIPT>** previamente modificamos el campo del texto para permitirnos admitir 500 caracteres damos clic en **Sign Guestbook** y vemos que sucede**.**

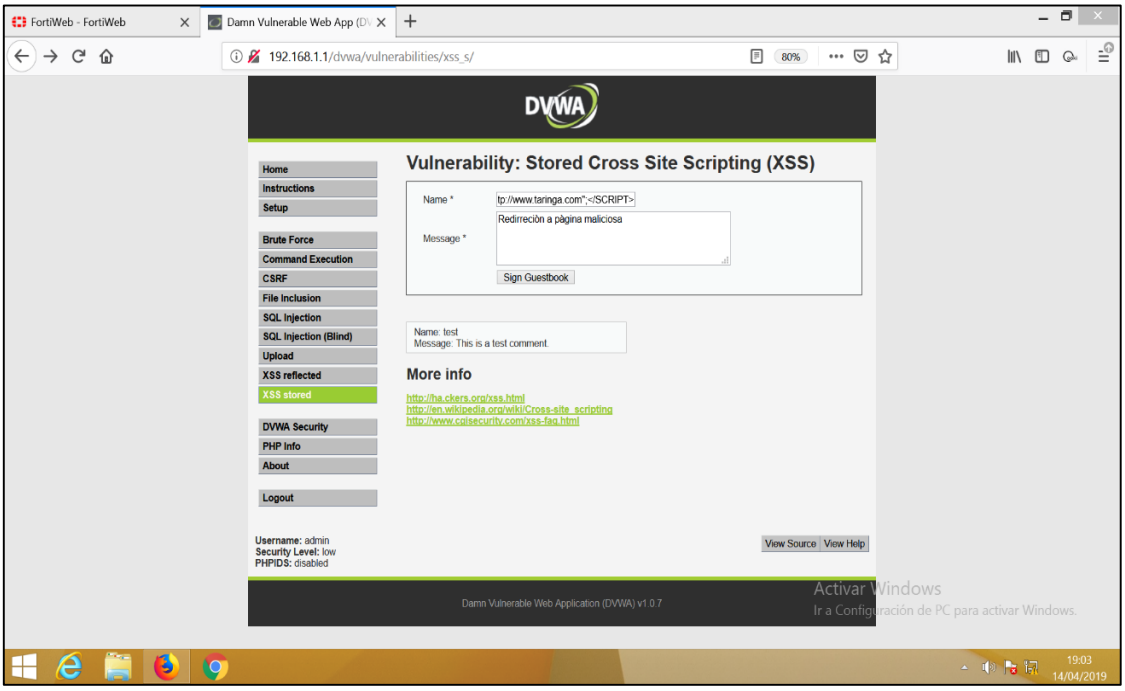

Figura 48. Ejecutando script malicioso

Como vemos el ataque fue correctamente bloqueado.

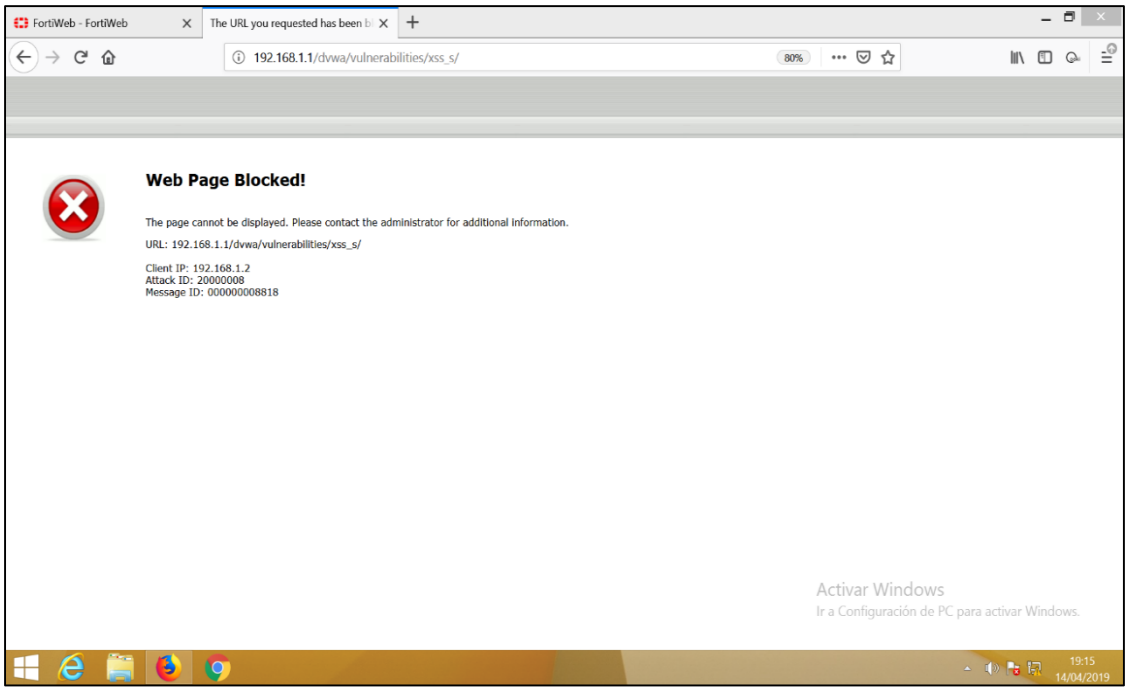

Figura 49. Bloqueo de FortiWeb

10. Ataque de referencia directa insegura a objetos

10.1 Referencia directa insegura a objetos

Las referencias inseguras a los objetos pueden ocurrir donde la aplicación web no valide la autorización a esos objetos. El hecho de que un cliente haya enviado un nombre de usuario y una contraseña no significa que deba ser universalmente confiable. Cada solicitud debe validarse para determinar si esa persona está autorizada para cambiar las contraseñas de los demás o no.

Este concepto es similar a A7 y A10, que verá más adelante. La diferencia es que A4 trata específicamente con parámetros en lugar de páginas y entradas de aplicaciones en lugar de redirigir las URL.

Linux permiten ejecutar comandos en una sola línea a través del uso de una serie de operadores:

- Ampersand (“&”) nos permite que dos o más comandos se ejecuten de manera simultánea.
  $ cd /tmp & mkdir nombre_directorio
- Barra (“|”): la salida del primero se convierte en la entrada del segundo.
  $ find . | xargs grep cadena_a_buscar
- Doble ampersand (“&&”) u operador AND: el segundo comando sólo se ejecutará si el primero termina con éxito.
  $ make && make install
- Doble barra (“||”) u operador OR: el segundo comando se ejecutará si el primero NO termina con éxito.
  $ cp /home/pepe/*.doc /backup/usuarios/pepe || echo "Sin hacer nada"

Vamos al Sistema DVWA a la opción de **Command Execution** y escribiremos el siguiente comando **& cat /etc/passwd** veremos que resultado obtenemos.

Como podemos validar el campo que debería permitir ingresar una dirección ip no valida los parámetros que se ingresan, con el comando ingresado podemos visualizar el contenido del archivo **passwd** donde se registran las cuentas de usuarios, así como las claves de accesos y privilegios.

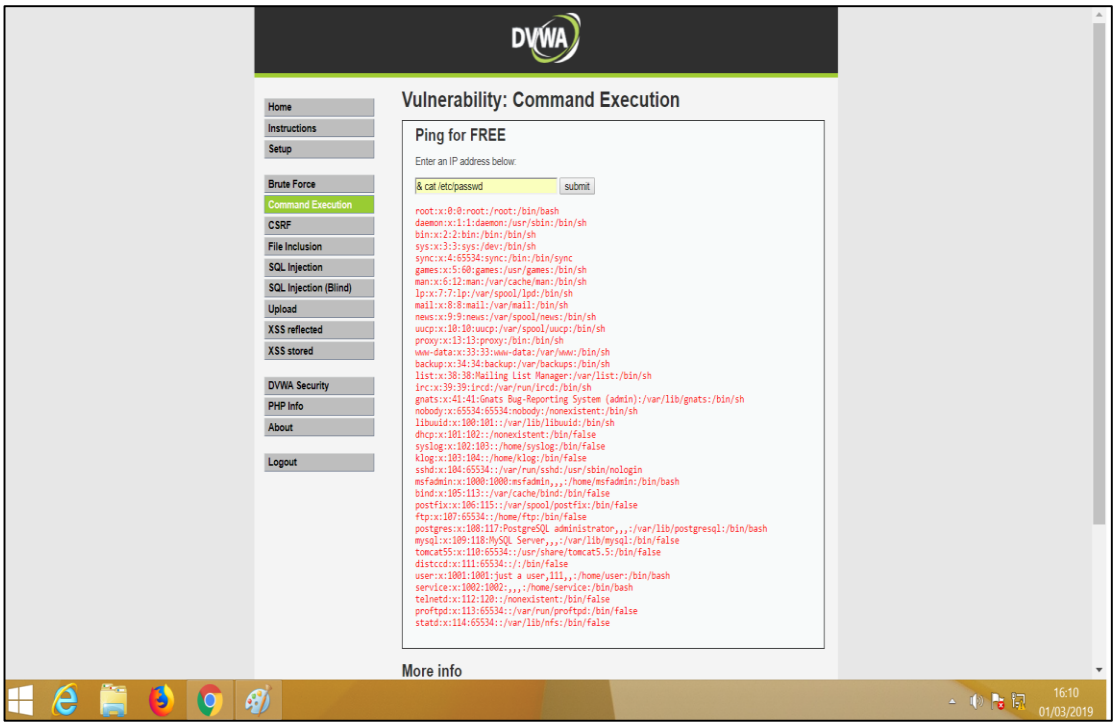

Figura 50. Ejecutando comandos maliciosos

Ahora vamos a listar los directorios del servidor mediante el siguiente comando || ls -l / veamos que sucede.

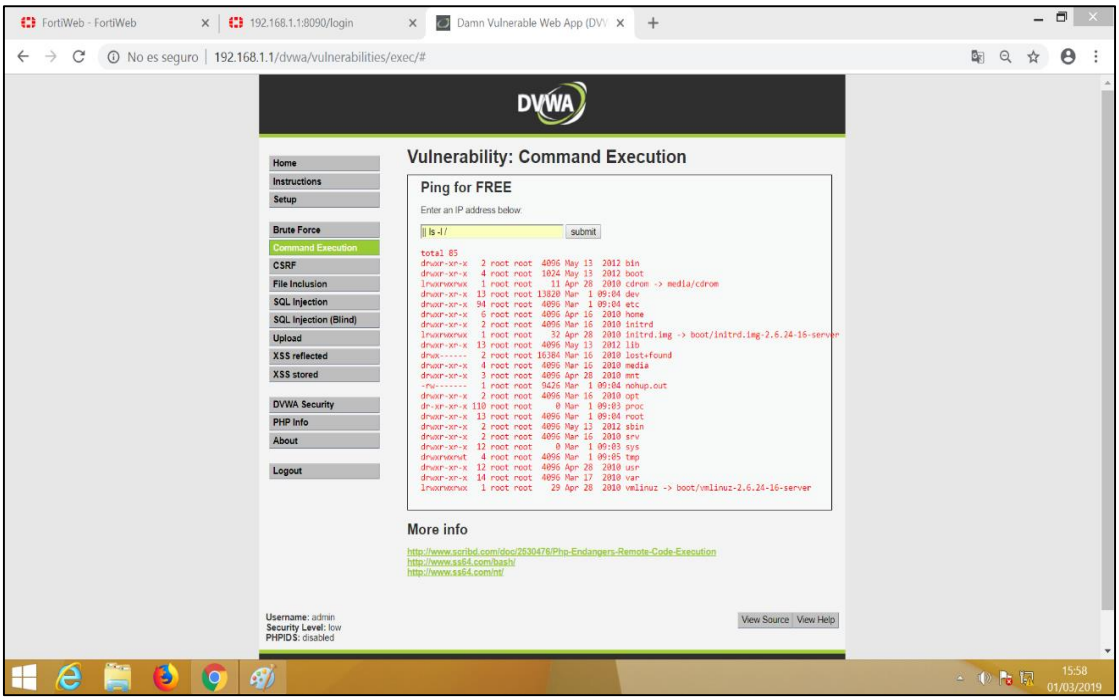

Figura 51. Ejecutando comandos maliciosos

En estos tipos de ataques también puede realizarse mediante envenenamiento de cookies que ya hemos visto anteriormente, otro tipo de ataque que podemos ser vulnerables es por la divulgación de información que podemos observar en la siguiente URL http://192.168.1.1/dvwa/instructions.php una extensión de archivo **.php**, por ejemplo, es una entrada que le dice al servidor web que debe usar su preprocesador PHP para representar la página HTML antes de responder a los clientes. Si permite que un cliente cargue un archivo PHP arbitrario y luego vaya a esa URL, puede acceder a cualquier información a la que tenga acceso el módulo **PHP**.

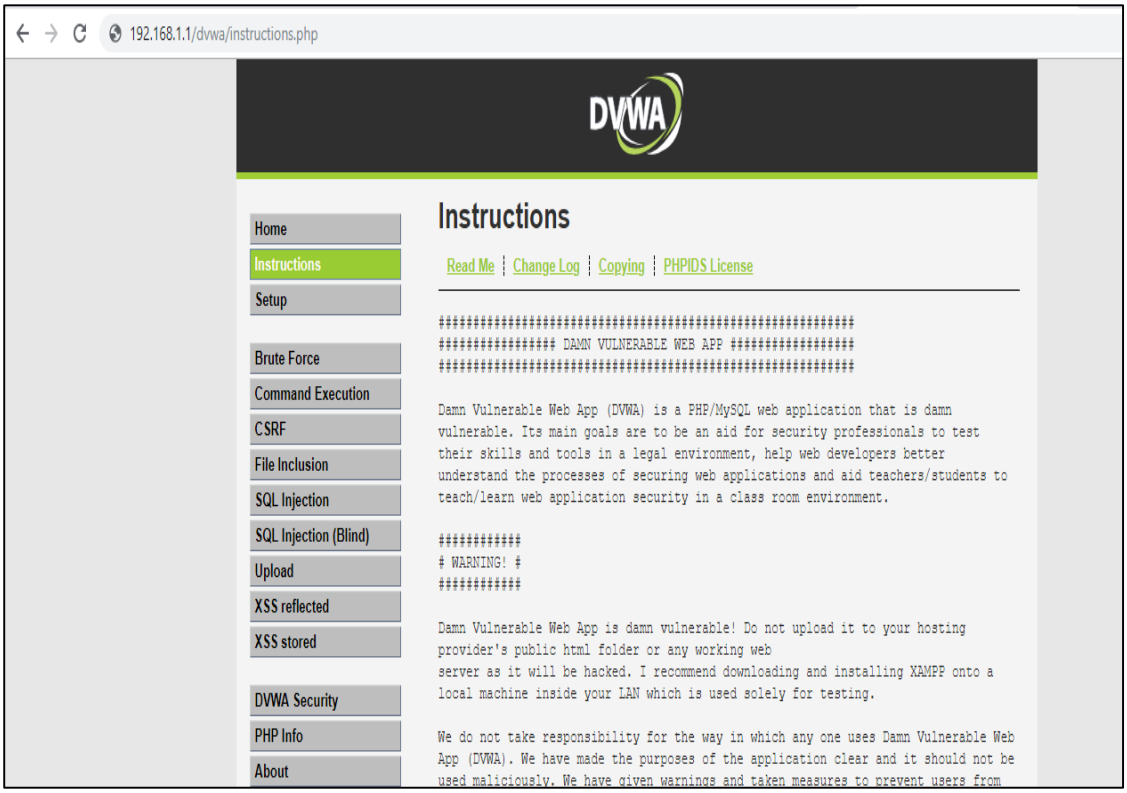

Figura 52. Sistema web muestra extensión de archivos

10.2 Mitigación referencia directa insegura a objetos

Fortiweb nos brinda la posibilidad de aplicar reglas de validación de parámetros, podemos definir longitudes y tipos de datos para los campos que necesiten su respectiva validación, a continuación, vamos a brindar seguridad al campo de la pestaña **Command Execution**, veremos cómo nos protege de estos tipos de ataques.

Ejecutando el comando **& cat /etc/passwd** vemos lo siguiente**:**

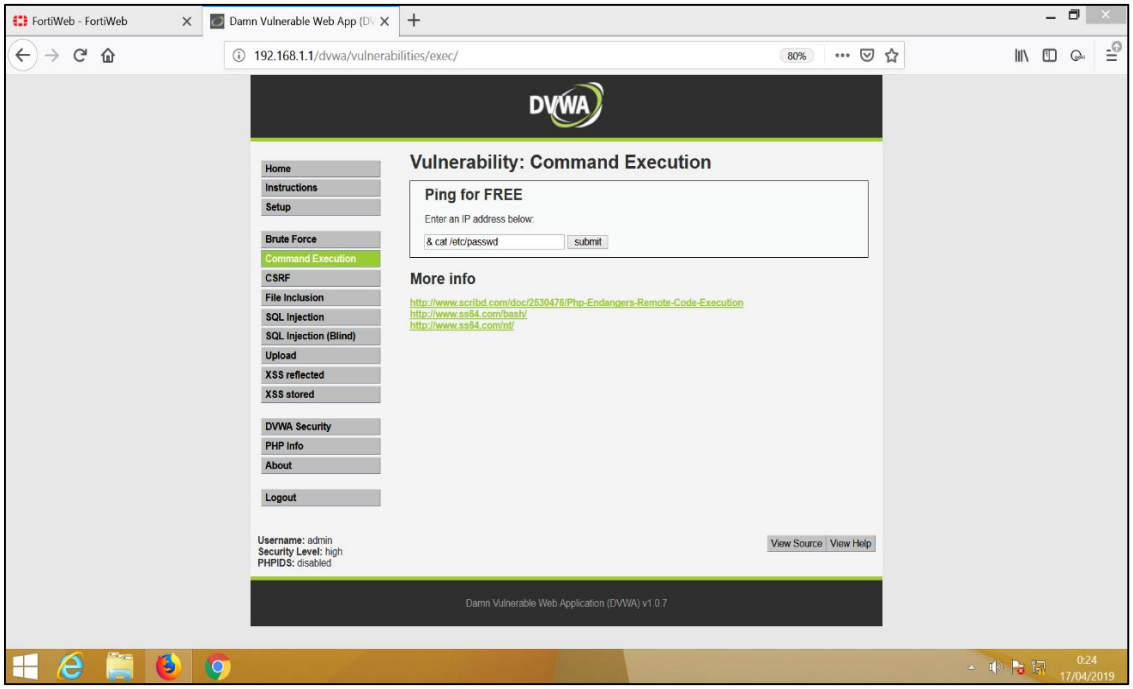

Figura 53. Ejecutando comandos maliciosos

Podemos ver como Fortiweb bloquea el ataque de manera satisfactoria.

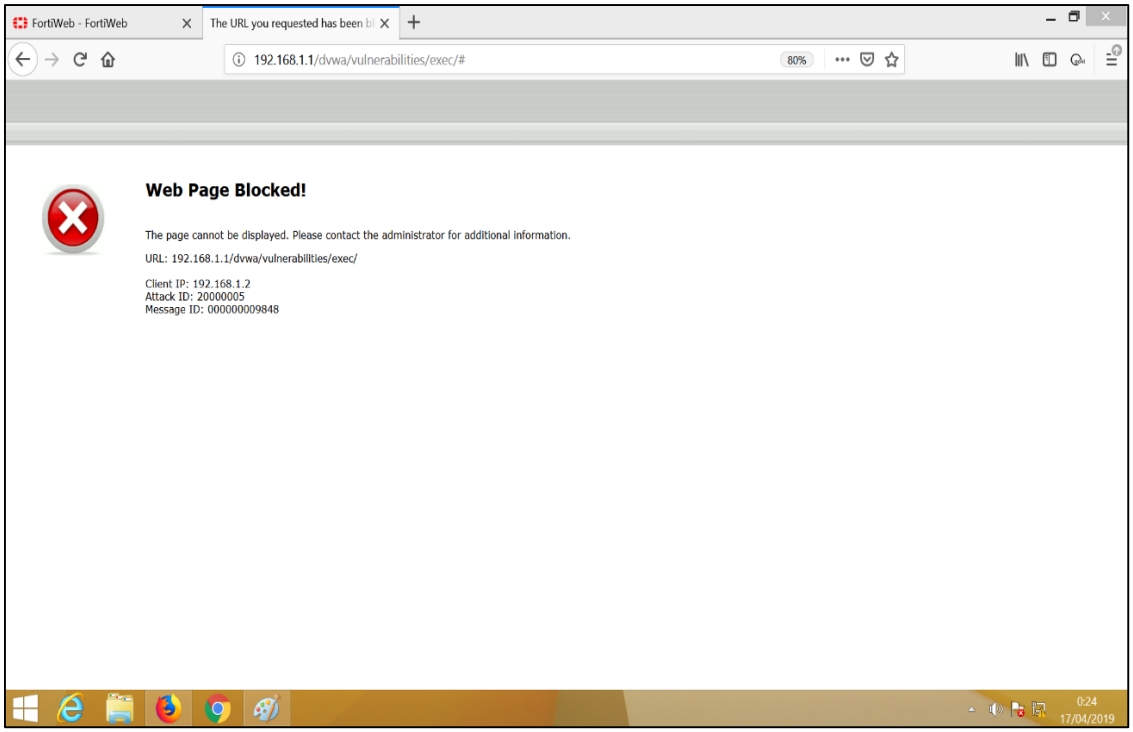

Figura 54. Bloqueo de FortiWeb

Otra opción de mitigar este tipo de ataque es aplicando una validación del campo para solo permitir el ingreso de una dirección ip a continuación se

ve como ingresamos el nombre de una persona.

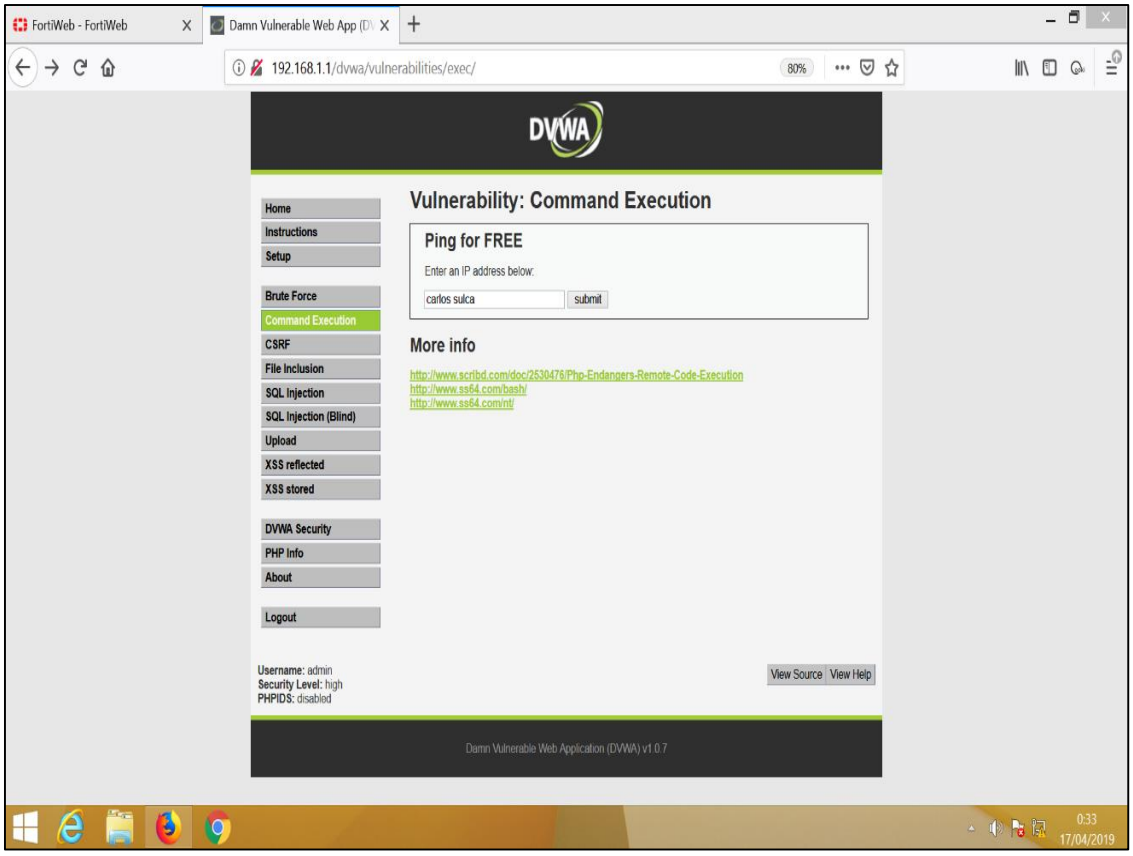

Figura 55. Ejecutando comandos maliciosos

Vemos el log de Fortiweb donde se evidencia el bloqueo de un parámetro inválido.

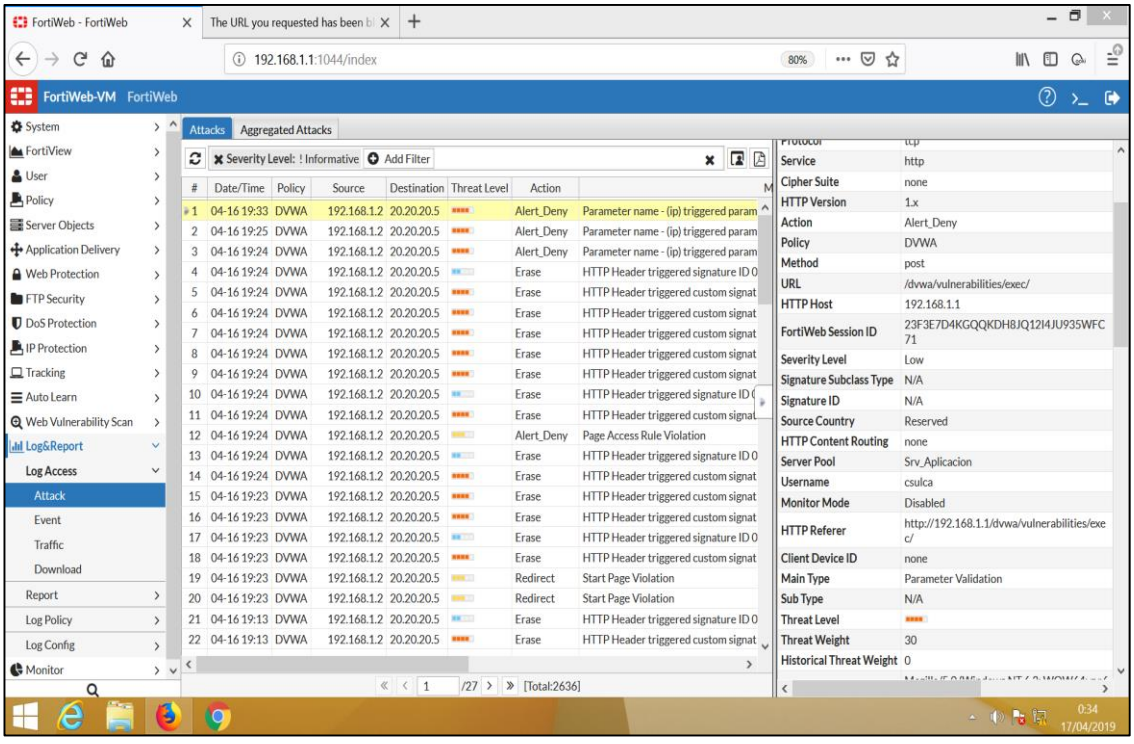

Figura 56. Revisando logs generados

11. Ataque de configuración de seguridad incorrecta

11.1 Configuración de seguridad incorrecta

La mayoría de los servidores web tienen páginas predeterminadas. Cuando configura el servidor web por primera vez, esto ayuda a confirmar rápidamente que el software se está ejecutando. Sin embargo, estos archivos nunca deberían exponerse en servidores de producción. Este es esencialmente el mensaje de A5. Estos archivos proporcionan información que puede ser útil para los atacantes. Si sus permisos son incorrectos, estos archivos también pueden ser un vector de explotación.

Ingresando a sistema DVWA podemos ver como la extensión de la página de inicio de sesión como la de la pestaña **PHP info** muestra información sensible del servidor y de la configuración del sistema web.

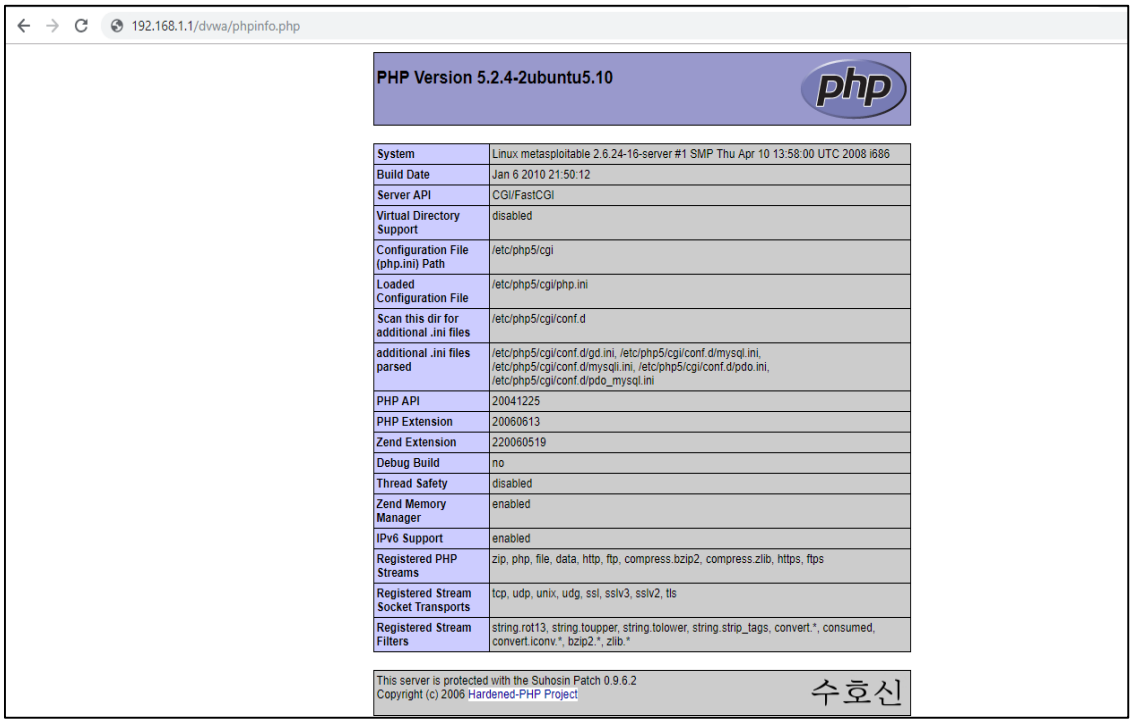

Figura 57. Divulgación de información

El archivo **phpinfo.php** generalmente tiene una función simple que muestra todas las configuraciones de PHP. Cada aplicación podría tener su propio archivo php.ini y .htaccess. IIS, Apache o cualquiera que sea su servidor web puede insertar un encabezado **X-Powered-By: y Server:** que

indica qué versiones de servidor y parche están instaladas. Las huellas dactilares de la pila de software son útiles para crear ataques o incluso para comprar ataques predefinidos en el mercado negro.

Si cualquier archivo de configuración puede ser leído, escrito o ejecutado por los usuarios en Internet, los atacantes pueden obtener información sobre cómo explotar los servidores no reparados, reescribir la configuración y más.

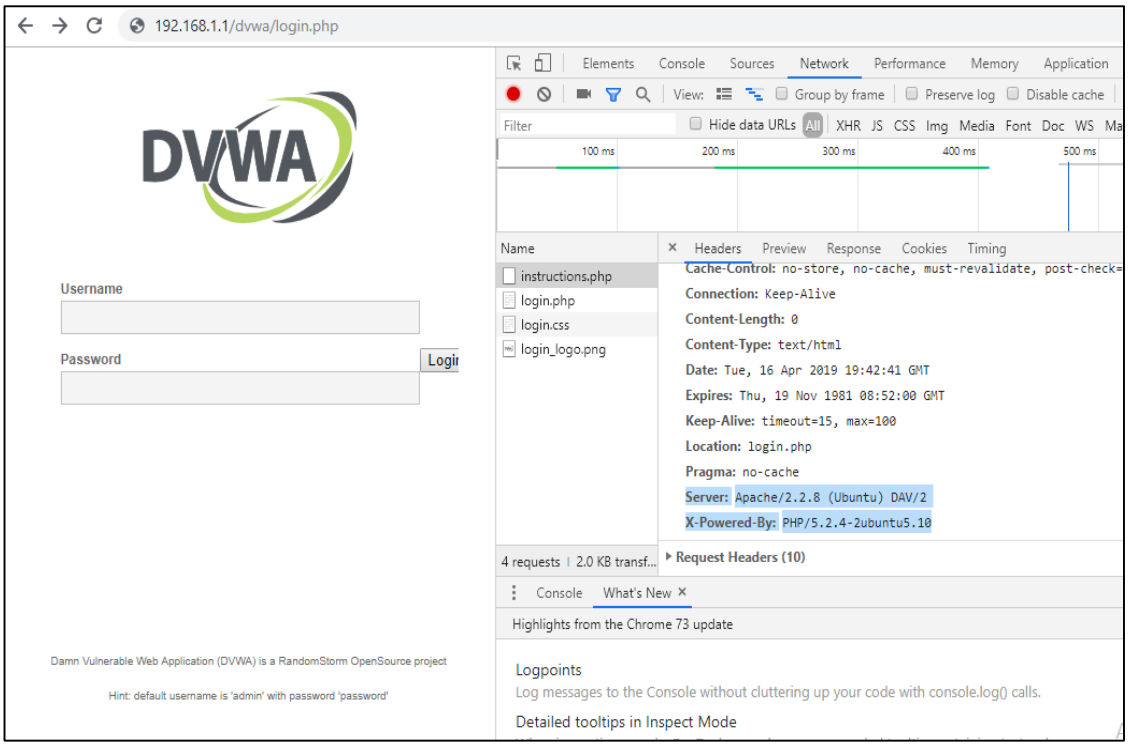

Figura 58. Divulgación de información

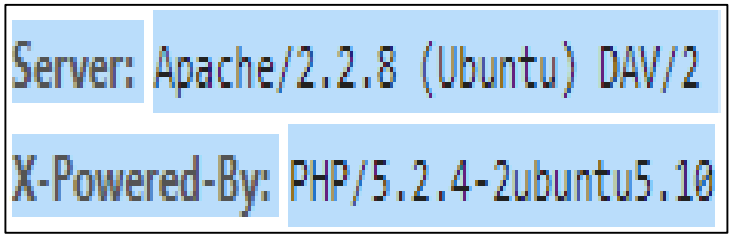

Figura 59. Divulgación de información

11.2 Mitigación de ataques por configuración de seguridad incorrecta

Las pruebas creativas de penetración realizadas por especialistas humanos siempre deben ser parte de sus auditorías de

seguridad, pero las **pruebas automáticas** también deben formar parte de su arsenal. Es imprácticamente lento encontrar vulnerabilidades comunes usando solo pruebas manuales. Para esto, podemos usar **el escáner de vulnerabilidad web** integrado de FortiWeb, utiliza HTTP (como sus usuarios) e intenta encontrar módulos no parcheados, divulgación de código fuente, más vulnerabilidades a tres de las 10 amenazas principales de OWASP.

- Inyección sql.
- Secuencia de comandos en sitios cruzados XSS.
- Ejecución de comandos.
- Divulgación de información.

Las vulnerabilidades A5 se enumerarán en la Información y a veces, en las categorías de vulnerabilidades de divulgación de fuentes. También puede usar la categoría de firmas de Divulgación de información para encontrar vulnerabilidades A5. Otra mitigación de Fortiweb es usar el **autoaprendizaje** el cual detecta las direcciones URL sensibles de cada servidor y por último **definir reglas de acceso a URL sensibles**.

Para bloquear los ataques de divulgación de información activaremos en firmas de Fortiweb como veremos a continuación:

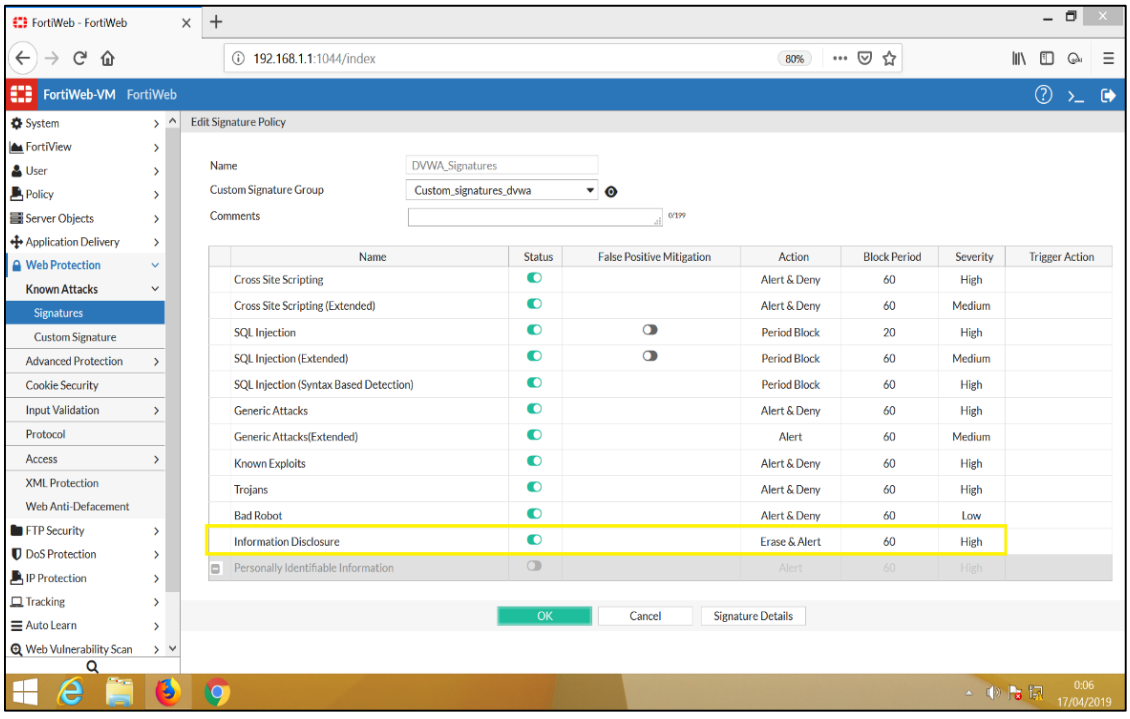

Figura 60. Activando firmas de seguridad

Adicionalmente vamos a crear una firma personalizada para poder ocultar la información de **X-Powered-By: y Server:**

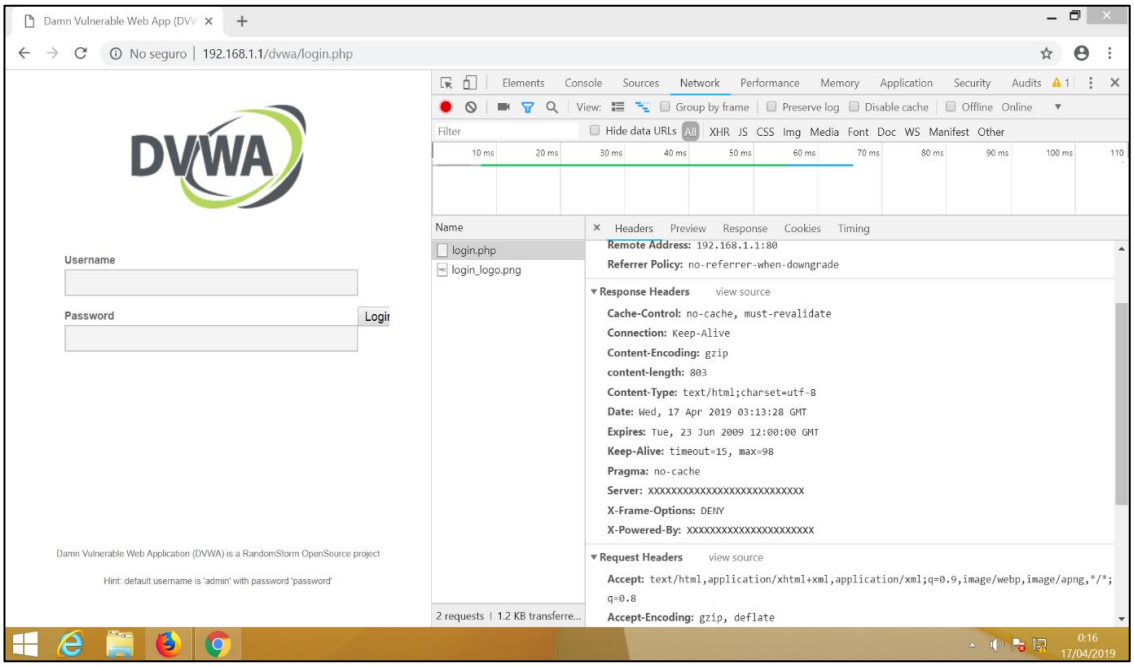

Figura 61. Ocultando información sensible

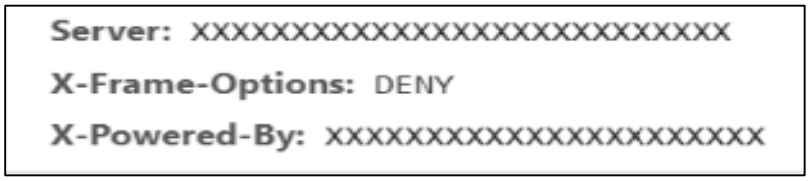

Figura 62. Ocultando información sensible

Como se comentó anteriormente podemos ejecutar el escáner de vulnerabilidades que cuenta Fortiweb para poder visualizar brechas de seguridad y recomendaciones para mitigarlas en nuestro sistema web.

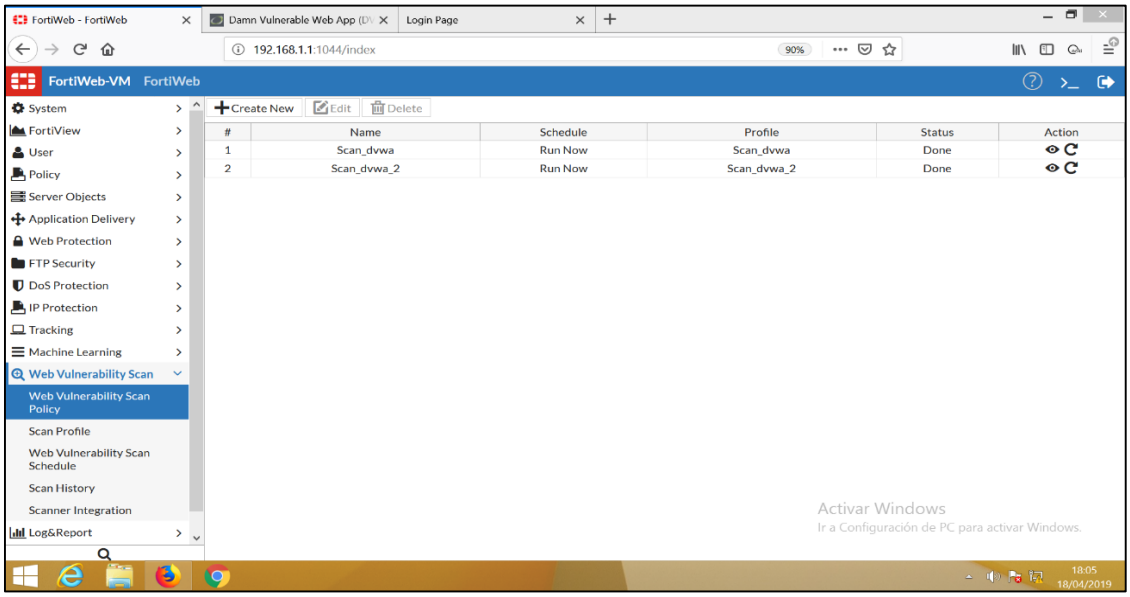

Figura 63. Análisis de vulnerabilidades de Fortiweb

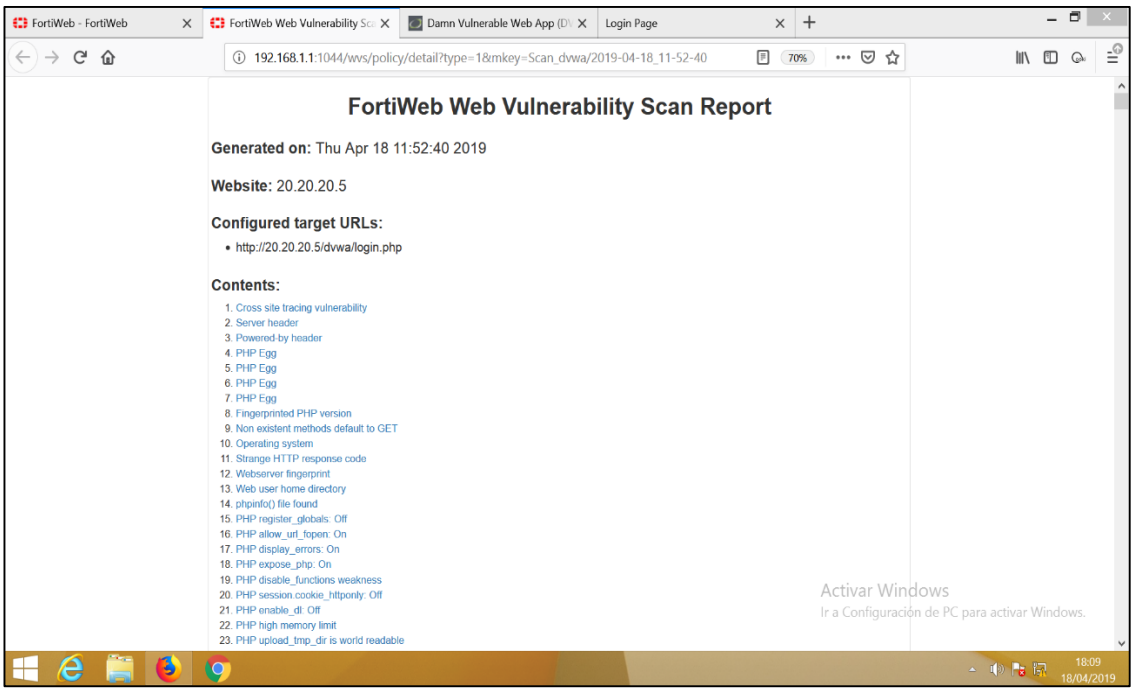

Figura 64. Reporte de vulnerabilidades de Fortiweb

El autoaprendizaje puede enseñarle mucho sobre las amenazas a las que se enfrentan sus activos web. También le ayuda a comprender las estructuras de sus aplicaciones web y cómo las usan los usuarios finales. Sin embargo, lo más importante es que el aprendizaje automático puede ayudarlo a adaptar rápidamente la configuración de FortiWeb para que se adapte a sus aplicaciones web.

El autoaprendizaje descubre las URL y otras características de las sesiones HTTP y / o HTTPS al observar el tráfico que pasa a sus servidores web. Para saber si la solicitud es legítima o un posible intento de ataque, realiza las siguientes tareas:

- Compara la solicitud para atacar firmas
- Observa entradas tales como cookies y parámetros de URL.
- Rastrea la respuesta de sus servidores web a cada solicitud, como 401 Unauthorizedo500 Internal Server Error
- Captura la tasa de solicitudes de archivos (hits) por dirección IP y tipo de contenido

Al aprender de su tráfico, el dispositivo FortiWeb puede sugerir

configuraciones apropiadas y ayudarlo a generar rápidamente perfiles diseñados específicamente para su tráfico único.

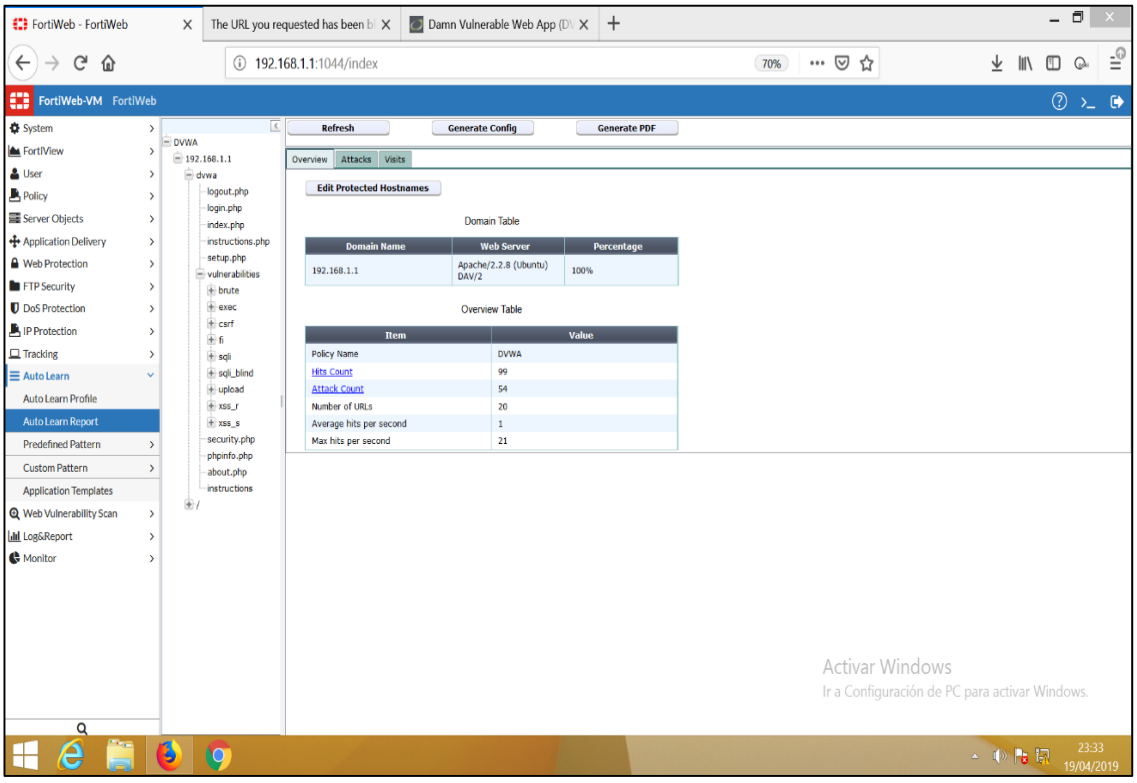

Figura 65. Análisis de tráfico por Fortiweb

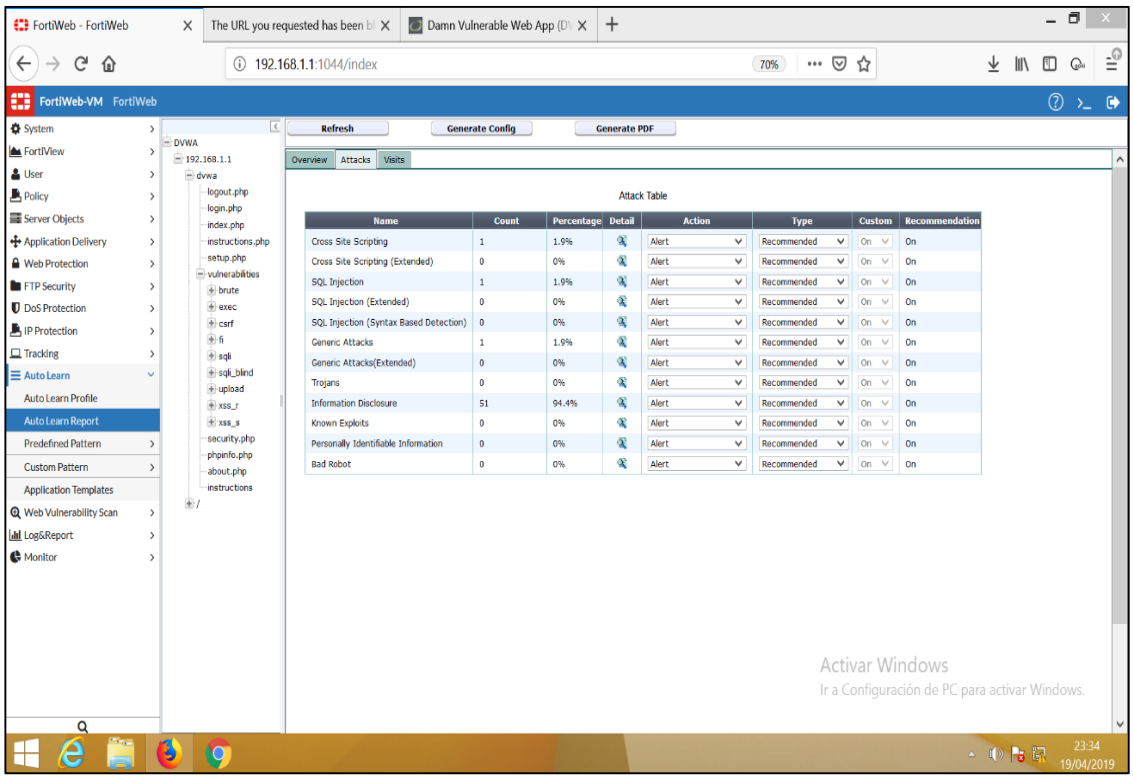

Figura 66. Clasificación de ataques revisados por AL

Otra técnica que Fortiweb nos permite usar para la protección de

posibles ataques es la **reescritura de url**, pero en este caso va en conjunto con una reescritura del cuerpo html tanto en los **request y response**, veremos cómo nuestras páginas tienen la extensión **.php** dando información muy importante para usuarios maliciosos.

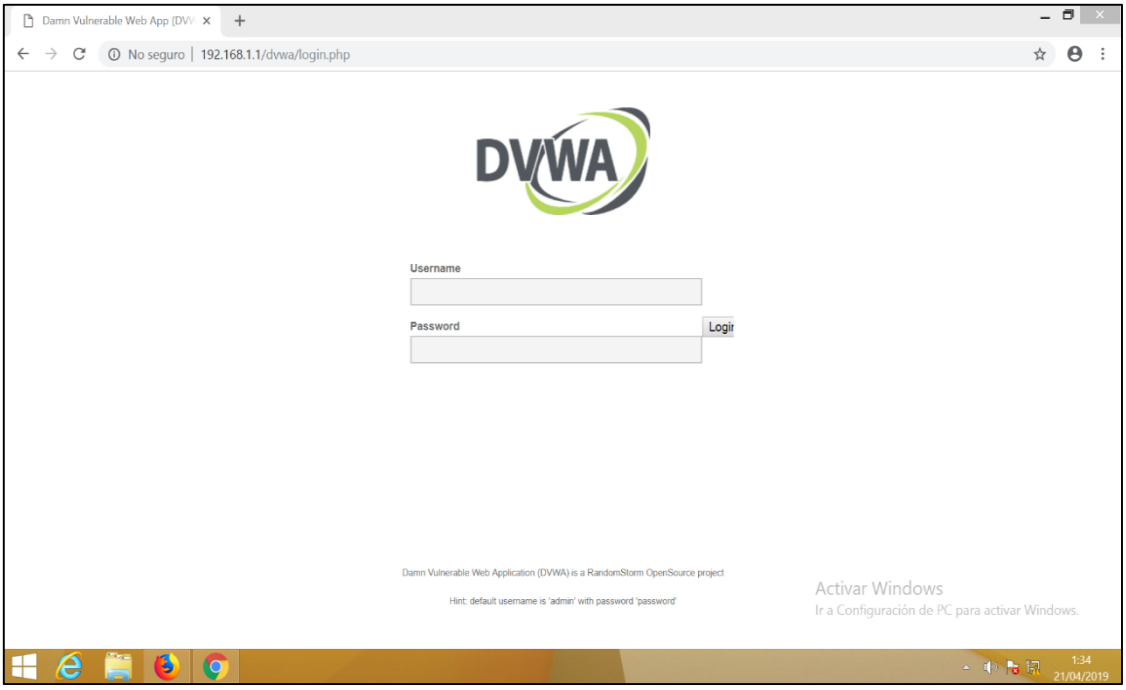

Figura 67. Divulgación de información

Ahora veremos cómo las url ya no muestran su extensión **.php** ayudándonos a no divulgar información de manera innecesaria.

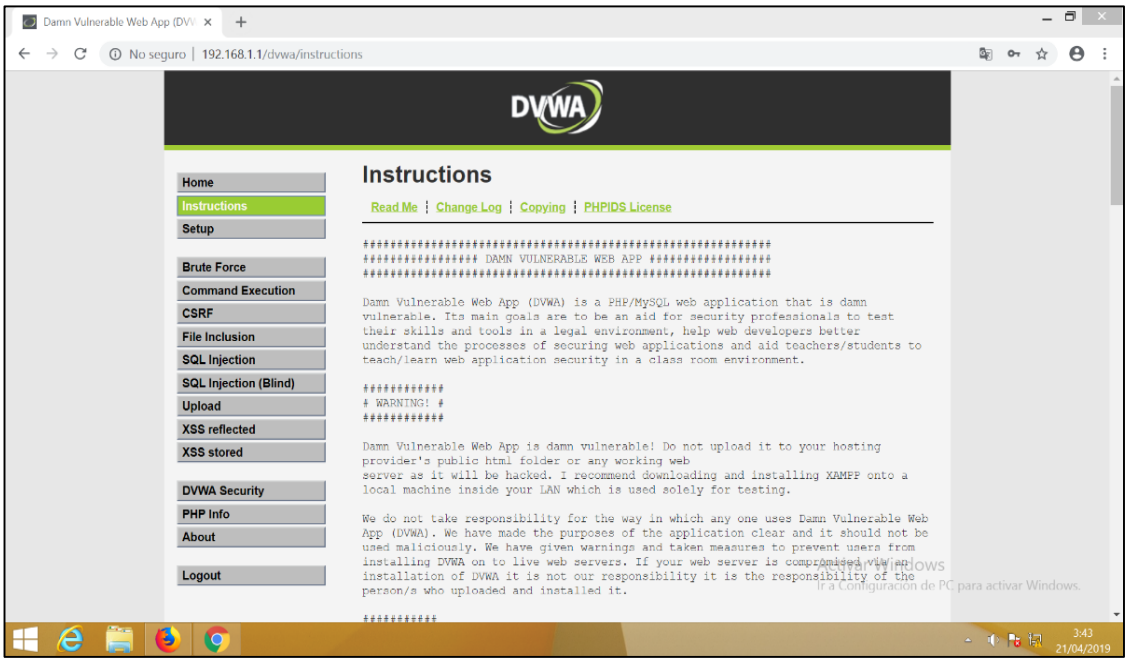

Figura 68. Ocultando la extensión de archivos del sistema

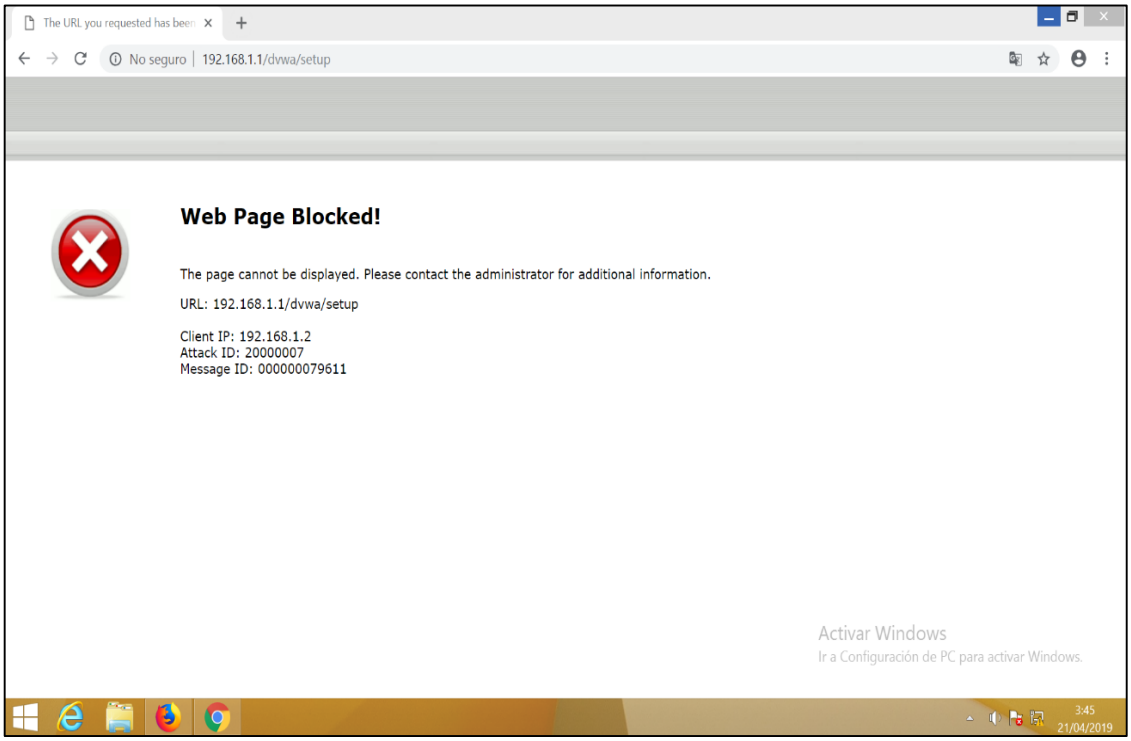

Figura 69. Ocultando la extensión de archivos del sistema

12. Ataques por exposición de datos sensibles

12.1 Exposición de datos sensibles

Este objetivo es el corazón del cumplimiento de **PCI DSS**. Las otras 10 principales amenazas de OWASP pueden afectar la seguridad de los datos almacenados de la tarjeta de pago, una razón más para eliminar dicha función de la aplicación web si es posible pero esta amenaza es específicamente sobre los datos mientras está en tránsito, en los cables.

Al igual que FortiGate, FortiWeb tiene protección de fuga de datos para detectar fugas de tarjetas de crédito en las respuestas del servidor. Idealmente, los servidores deberían aceptar números de tarjeta, pero nunca aumentar el riesgo repitiéndolos al cliente. FortiWeb va más allá, sin embargo. Como terminador SSL o TLS, FortiWeb solo puede ofrecer a sus clientes las versiones de protocolo y las suites de cifrado más seguras. Esto ayuda a mantener sus servidores y clientes más seguros**. Si se registran los ataques, puede enmascarar fácilmente las contraseñas y los números de las tarjetas de crédito para que no aparezcan en los registros sin encriptar**.

A continuación, vemos como en una conexión http las credenciales de un usuario va en texto plano.

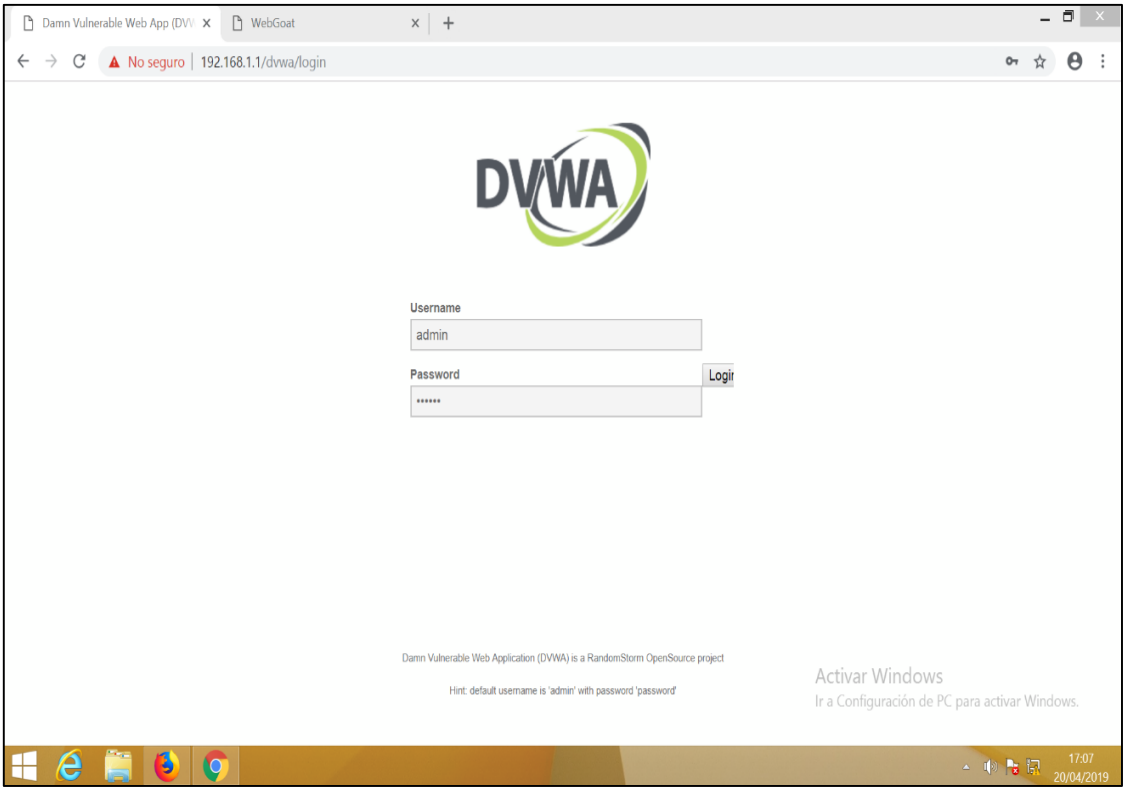

Figura 70. Iniciando sesión en el sistema web

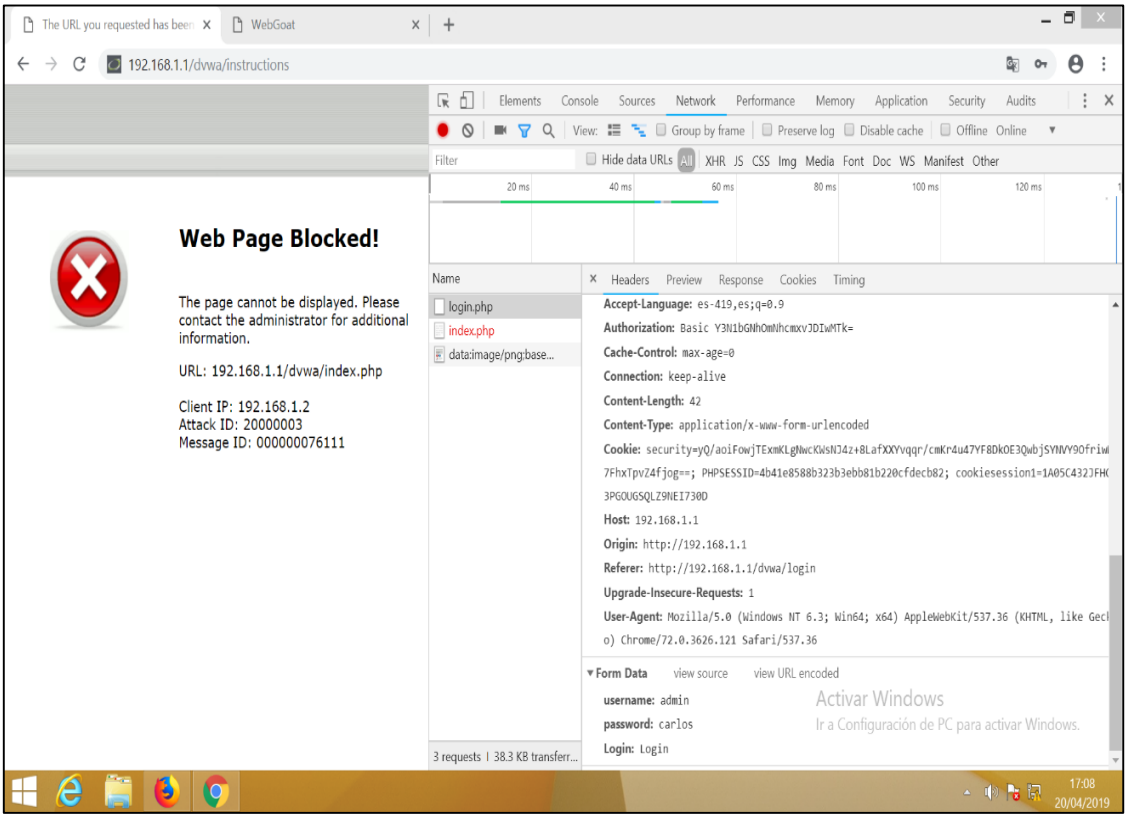

Figura 71. Visualizamos las credenciales en texto plano

12.2 Mitigaciones a la exposición de datos sensibles

Fortiweb nos permite habilitar DLP, editamos una política de firma, podemos configurar FortiWeb para detectar una infracción basada en un número específico de números de tarjetas de pago en la página web, si es necesario.

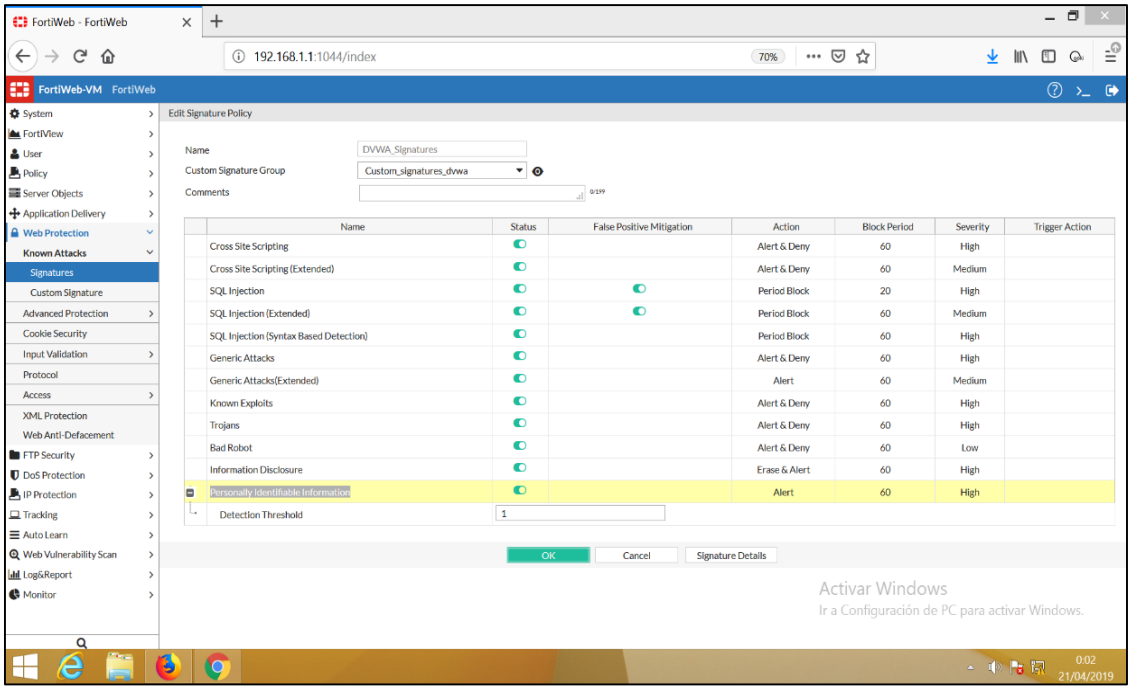

Figura 72. Activando firmas de seguridad

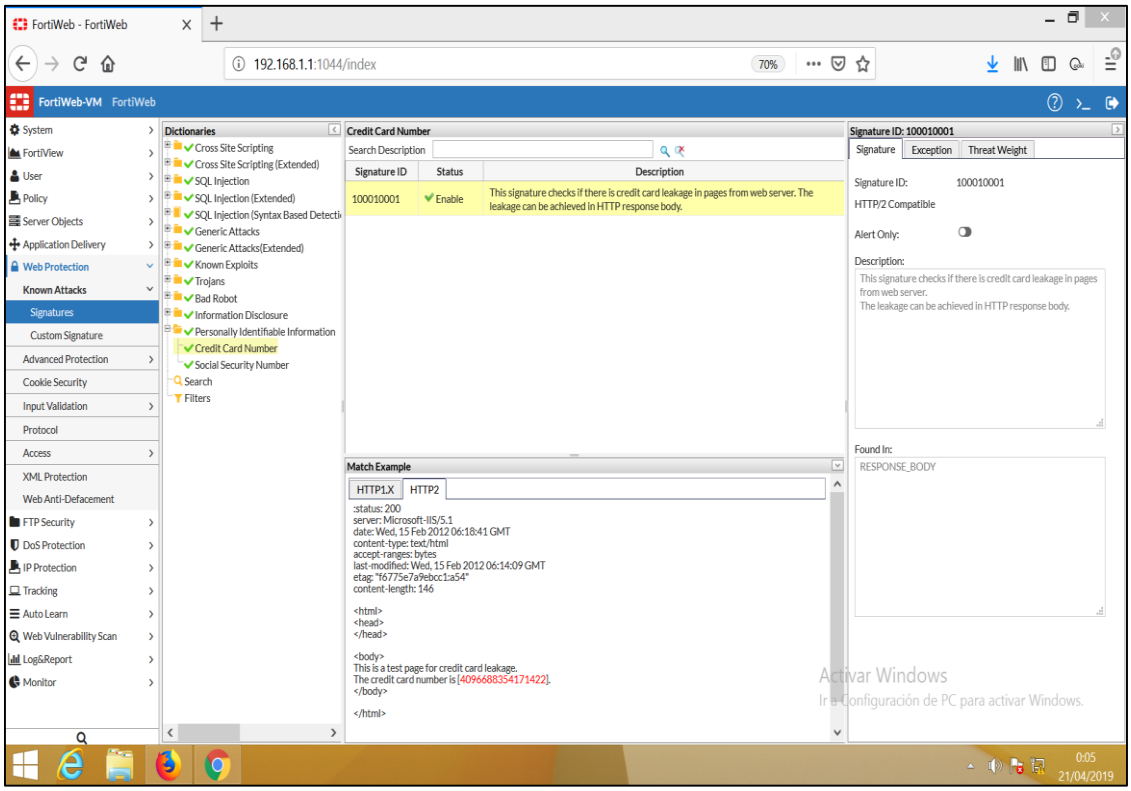

Figura 73. Activando la protección para número de tarjetas de crédito

Recuerde: tanto A6 como PCI DSS requieren seguridad mientras el pago está en tránsito. Para asegurar los datos de la tarjeta de pago mientras está en tránsito, la autenticación y el cifrado son fundamentales. Tenga en cuenta que las versiones anteriores de SSL 2.0 y SSL 3.0 tienen muchas vulnerabilidades conocidas, y deben evitarse, si es posible.

Si se requiere que cumpla con las normas PCI DSS, SSL 2.0, fortaleza de clave débil y hash MD5 se consideran violaciones de PCI DSS. Pronto, SSL 3.0, TLS 1.0 y SHA-1 también pueden ser violaciones. Esto es cada vez más probable, porque los estilos de renegociación y las suites de cifrado especificados ahora tienen exploits.

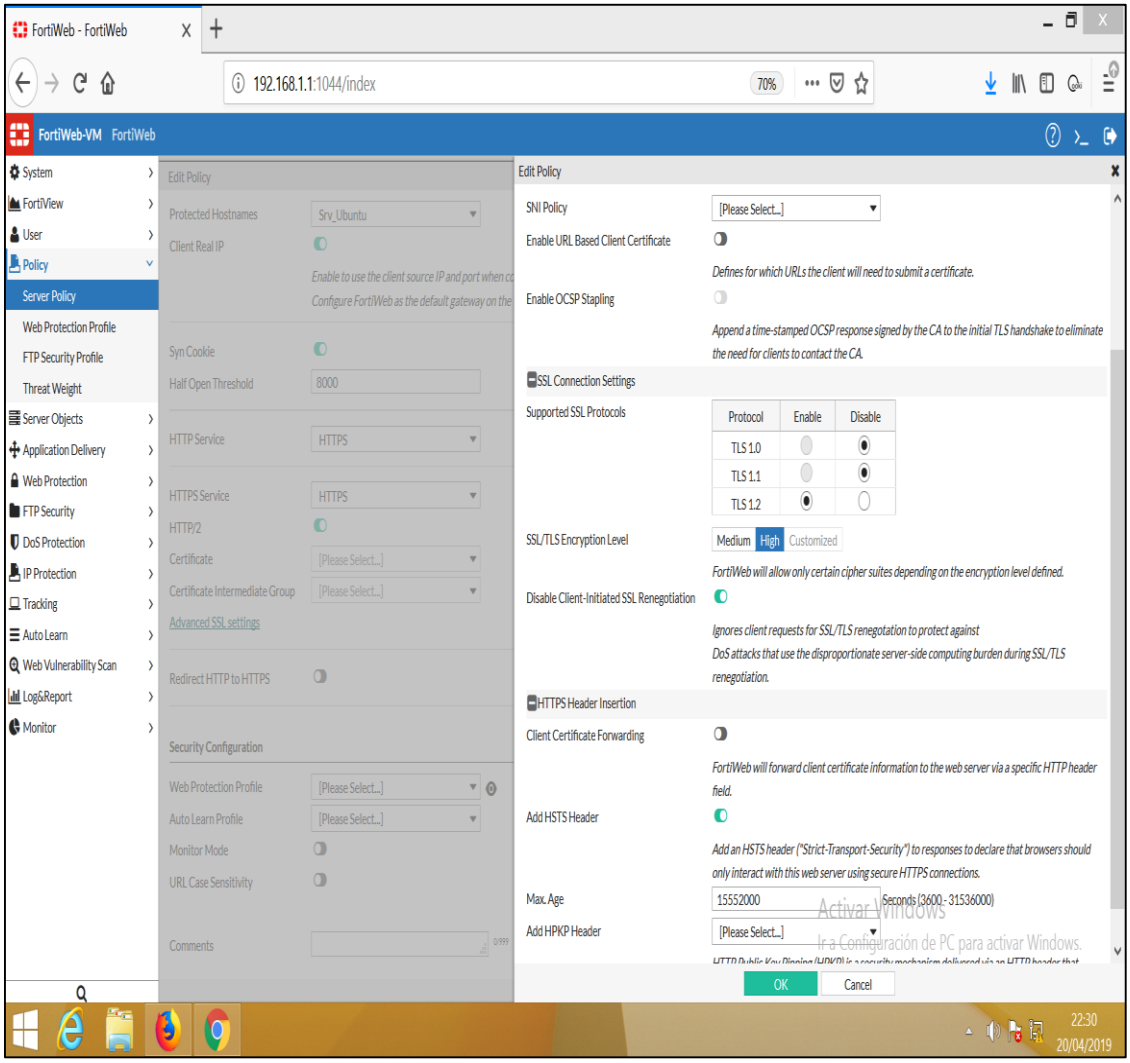

Figura 74. Activar TLS para las comunicaciones seguras

A continuación, vemos la configuración para cifrar las claves de los usuarios en los logs.

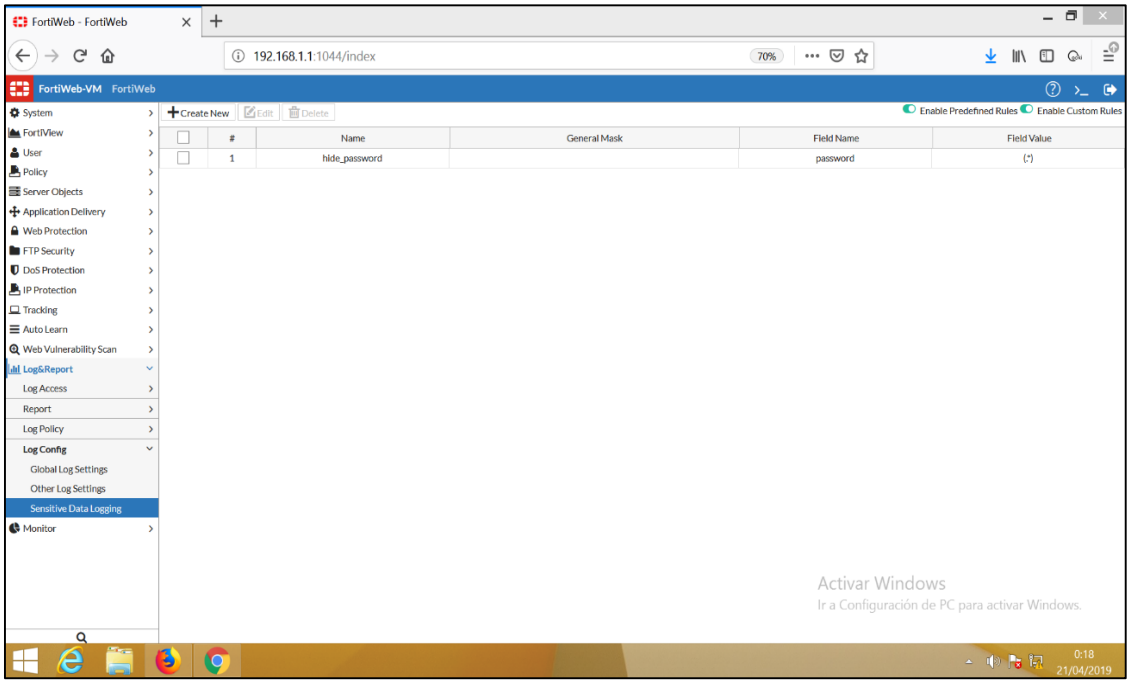

Figura 75. Activando el cifrado de información sensible en los logs

Validamos como la clave es correctamente cifrada en los logs almacenados en nuestro Fortiweb.

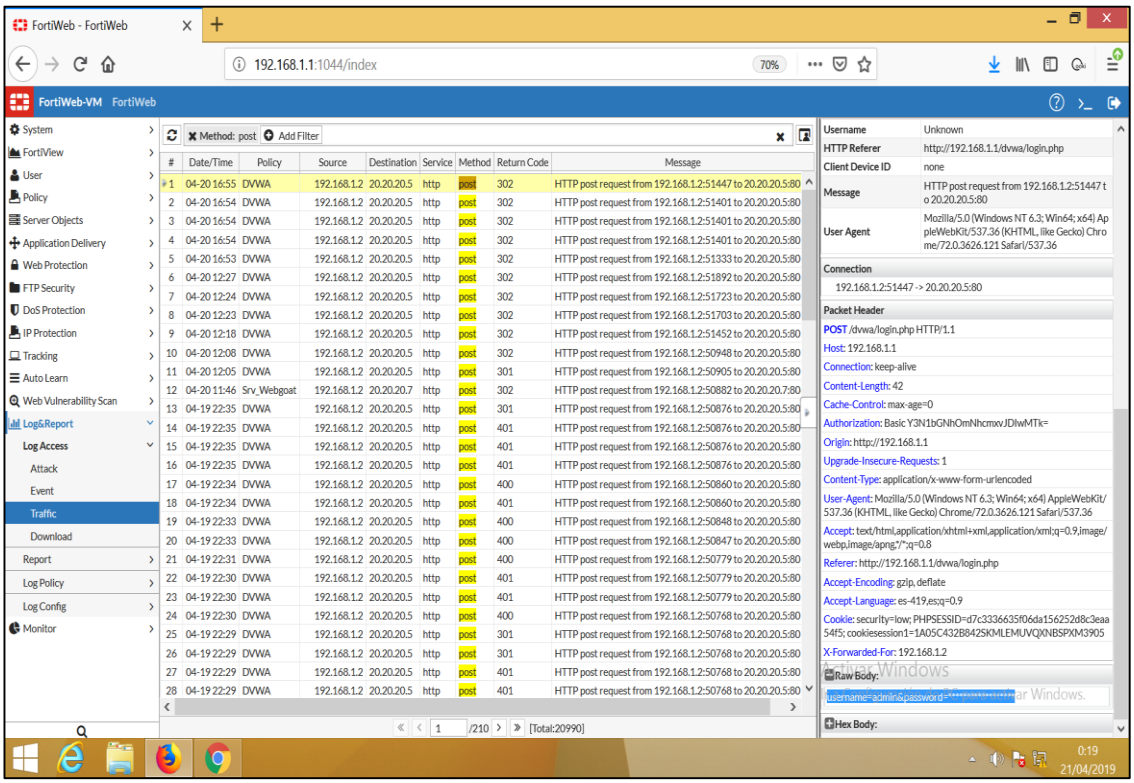

Figura 76. Clave cifrada en los logs

13. Ataques por ausencia de control de acceso a las funciones

13.1 Ausencia de control de acceso a las funciones

El hecho de que no haya hipervínculos públicos a una página web administrativa no significa que no se encuentre y explote. Si las reglas de acceso de su servidor web no lo prohíben, un atacante podría incluso acceder a archivos fuera del directorio de su aplicación web e incluso dentro de la propia aplicación web, no se puede suponer que los clientes siempre accedan a las páginas web de forma autorizada.

Algunos de los hacks más famosos se han ejecutado simplemente editando la URL que estaba en la barra de URL del navegador, tratando de acceder a las URL donde la aplicación no verificó la autorización. Esto se llama **navegación forzada.**

Recuerde incluso si un usuario está autenticado, no están necesariamente autorizados para cada URL. Para cada solicitud, la aplicación debe verificar que el cliente esté autorizado para ese dominio, como lo hace FortiWeb, y en ese paso de la sesión. La aplicación también debe verificar que la URL debe ser accesible desde esa dirección IP.

Los archivos de configuración no deberían ser accesibles a través de Internet. El acceso debe estar restringido a una red de gestión privada. Podría haber otras URL, usadas internamente por la aplicación web, cuyos permisos deberían establecerse para que no se pueda acceder directamente desde Internet.

A continuación, veremos cómo desde la página inicial del sistema donde necesitamos iniciar sesión para poder dar uso al sistema podemos moficar la url a **192.168.1.1/dvwa/setup.php** y accedemos sin problemas y resetamos la base de datos.

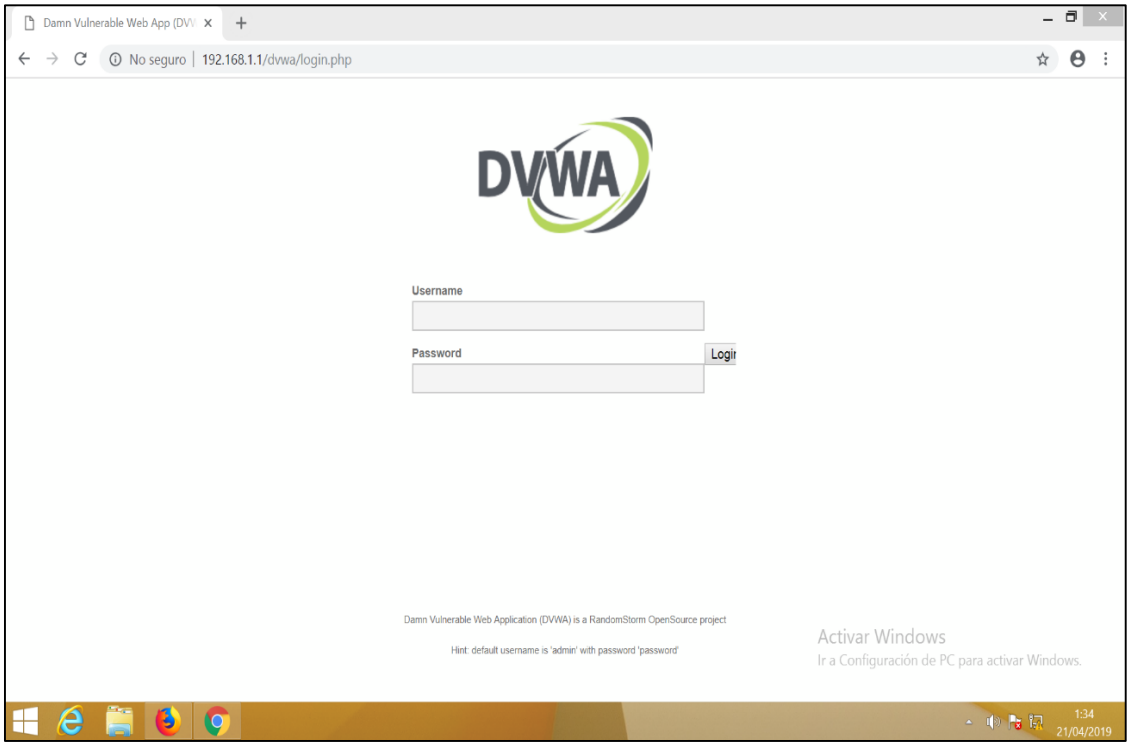

Figura 77. Página de inicio de sesión del sistema web

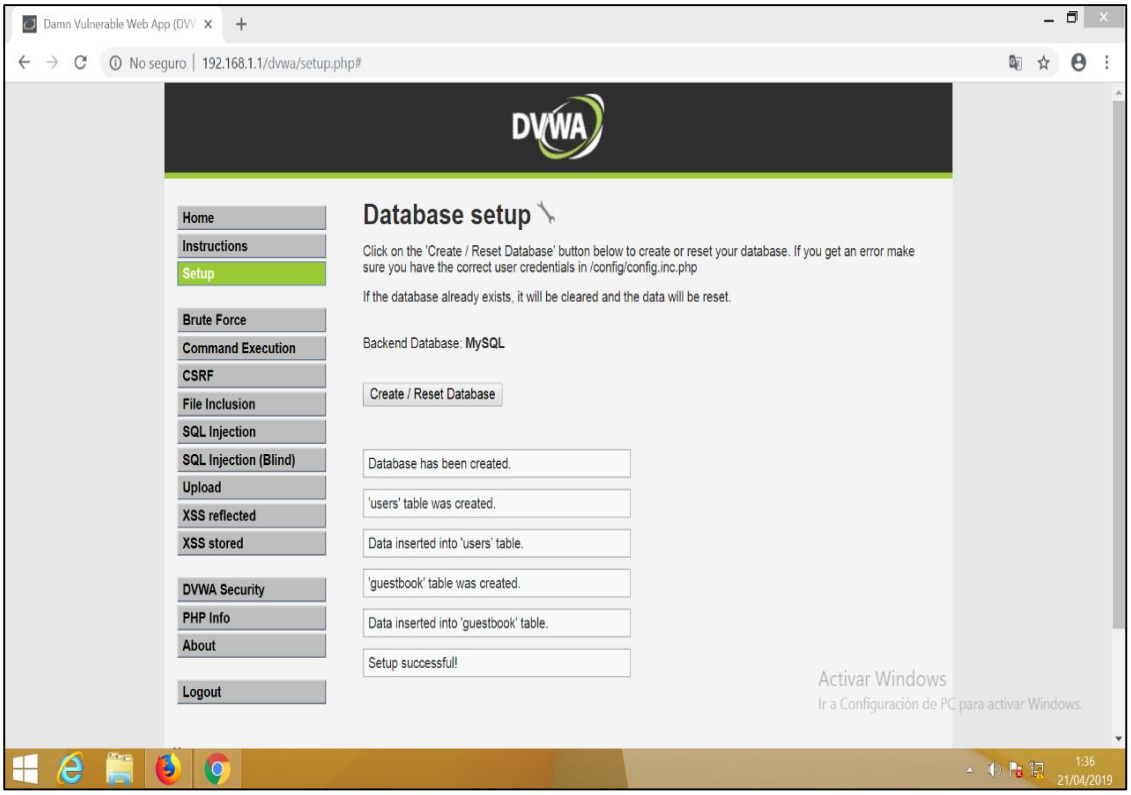

Figura 78. Borrado de la base de datos por acceso inválido

13.2 Mitigación en ausencia de control de acceso a las funciones

Fortiweb nos permite combinar diversas técnicas para protegernos de este tipo de ataques como usar las reglas de control de acceso, incluidas las llamadas reglas personalizadas, que le permiten combinar múltiples factores, como **User-Agent:** y la limitación de velocidad, se pueden seleccionar en el perfil de protección. También lo pueden hacer las **páginas de inicio**, **las reglas de orden de página y las firmas**.

Crearemos una regla de página de inicio con la url correcta **/dvwa/login** ahora vemos como al intentar saltar el login y intentar ingresar a **setup.php** es bloqueado su solicitud por intentar acceder a una url no autorizada para los usuarios, pero si intentan acceder a otra url como **/dvwa/instructions.php** los usuarios son redireccionados a la página de inicio.

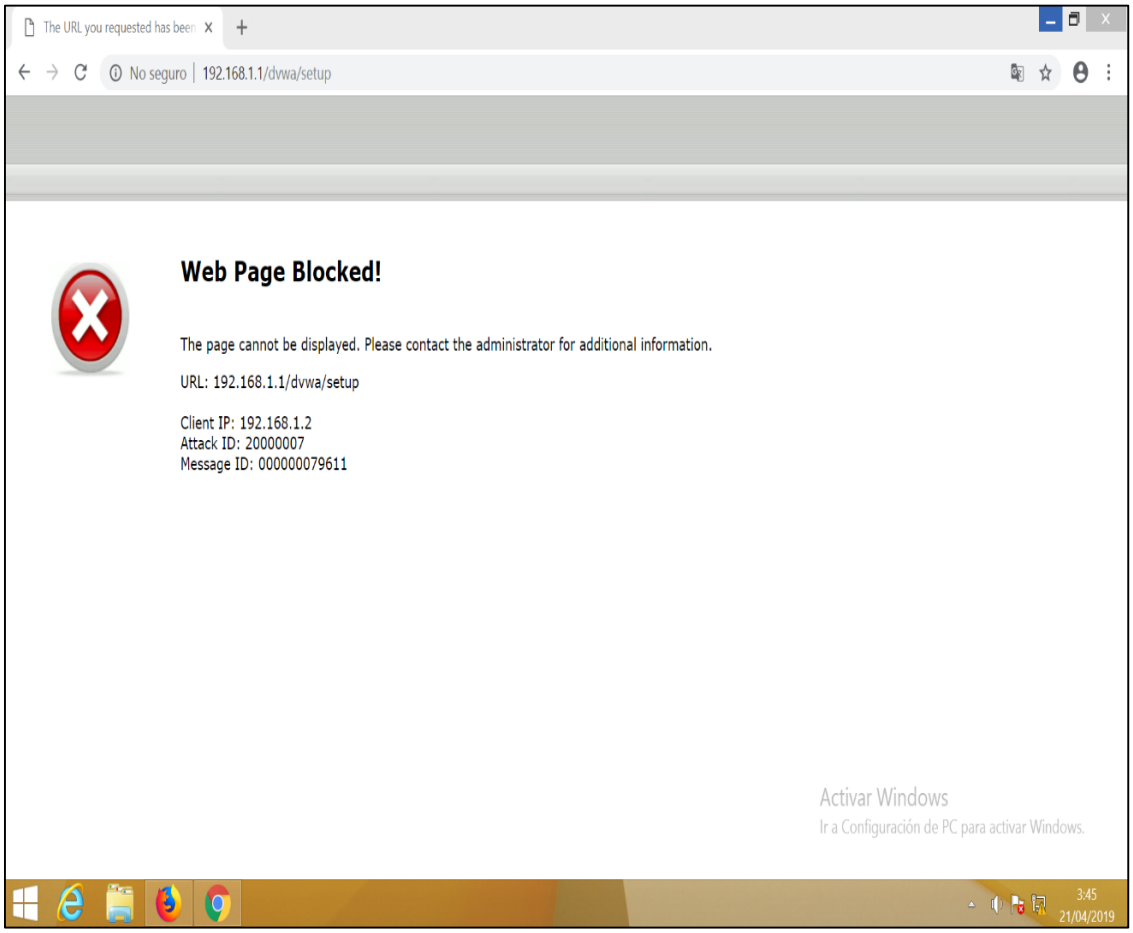

Figura 79. Bloqueo de FortiWeb

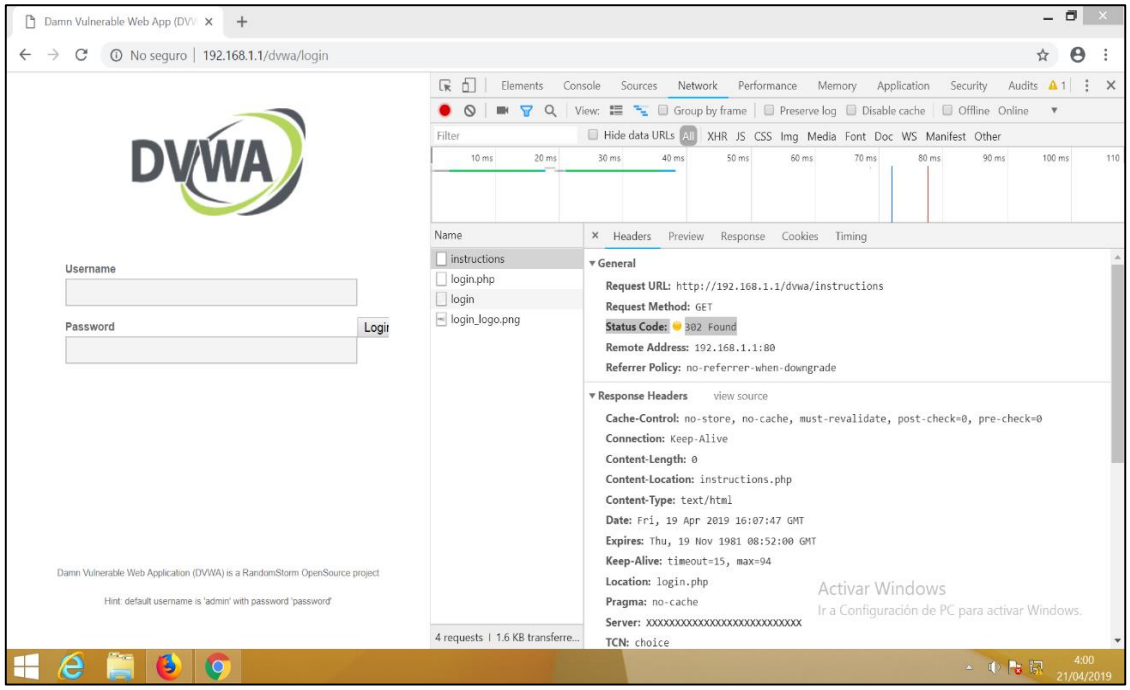

Figura 80. Redireccionamiento de FortiWeb

Fortiweb nos permite aplicar un orden lógico para el sistema web a continuación forzaremos al sistema para que acceda a una determinada url después de realizar el inicio de sesión, deberá ingresar a **instructions.php** como vemos muestra un bloqueo ya que el sistema tiene predeterminado ir a **index.php**

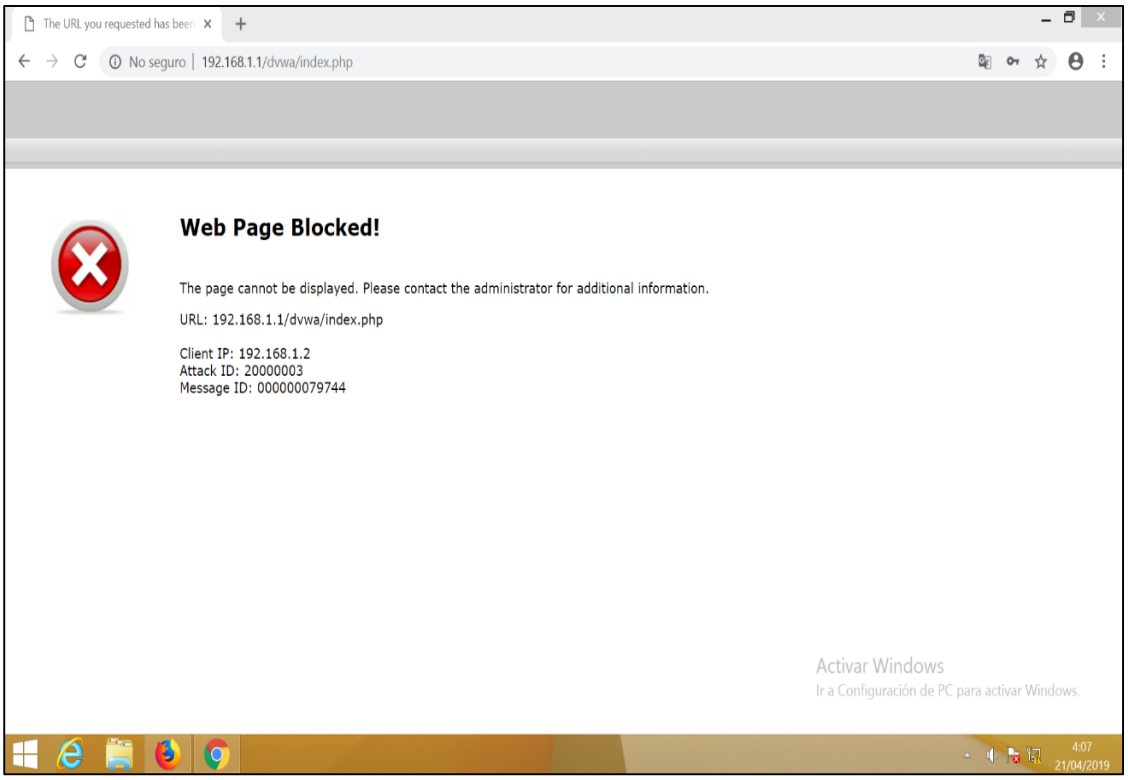

Figura 81. Bloqueo de FortiWeb

14. Ataques de falsificación de peticiones entre sitios cruzados CSRF

14.1 Falsificación de peticiones entre sitios cruzados CSRF

En A8, el cliente está correctamente autenticado y correctamente autorizado, pero un atacante inyecta código malicioso, elabora un correo electrónico de phishing o utiliza ingeniería social para engañar al usuario para que ejecute una acción.

Como era de esperar, FortiWeb puede detectar y desinfectar algunas formas de CSRF, como **clickjacking**. Pero otras formas de este ataque son actualmente demasiado intensivas en CPU y memoria para que la prevención sea práctica en tiempo real. A menudo requieren un gran número de reglas de seguridad tanto positivas como negativas.

De este modo, y dado que los navegadores ejecutan simultáneamente código enviado por múltiples sitios web, existe el riesgo de que un sitio web (sin el conocimiento del usuario) envíe una solicitud a un segundo sitio web y éste interprete que la acción ha sido autorizada por el propio usuario.

Por ejemplo, supongamos que Jane recibió un correo electrónico que parece ser de su banco. Se le inyectó un código para que cuando Jane visite el sitio web del banco real e intente transferir fondos, no se envíen al destinatario previsto. En cambio, los fondos se transfieren al atacante. Para detectar este tipo de ataque, un WAF debería recordar a todos los destinatarios de la transferencia autorizados y bloquear los no autorizados, para cada usuario, en cada aplicación web. Además, esta es solo una entrada, en una página web el WAF necesitaría comprender cada entrada, en cada página, de cada aplicación web protegida. Por lo tanto, si es posible, su aplicación debería intentar salvaguardar las transiciones críticas de las máquinas de estados como esta. Los auditores de seguridad de código pueden ayudarlo a encontrarlos, y las bibliotecas de CSRF existen para ayudar a eliminar esto. Esto ayudará a protegerse contra la ingeniería social o los vectores de chat. También puede habilitar firmas FortiWeb para evitar inyecciones de código CSRF.

Realizaremos el siguiente ataque primero ingresaremos al sistema y nos dirigimos a **CSRF** vamos a copiar el código html marcado.

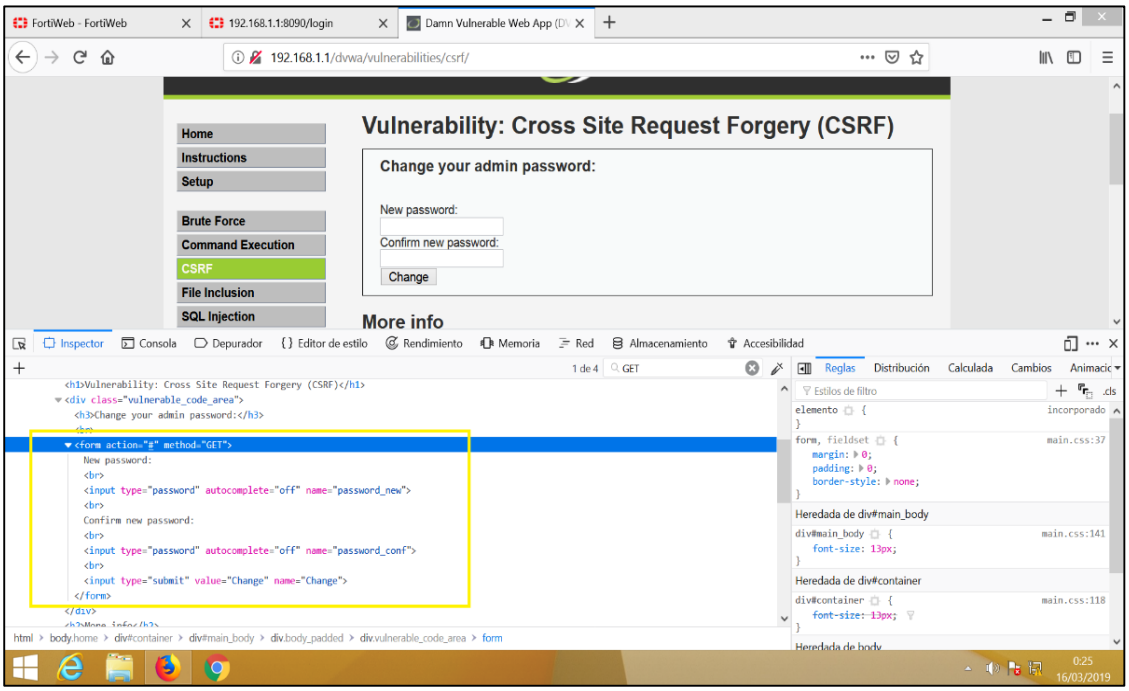

Figura 82. Copiando código html

El código copiado lo guardaremos en un notepad con extensión .html y como visualizamos muestra los campos para poder realizar el cambio de contraseña en el **sistema DVWA**.

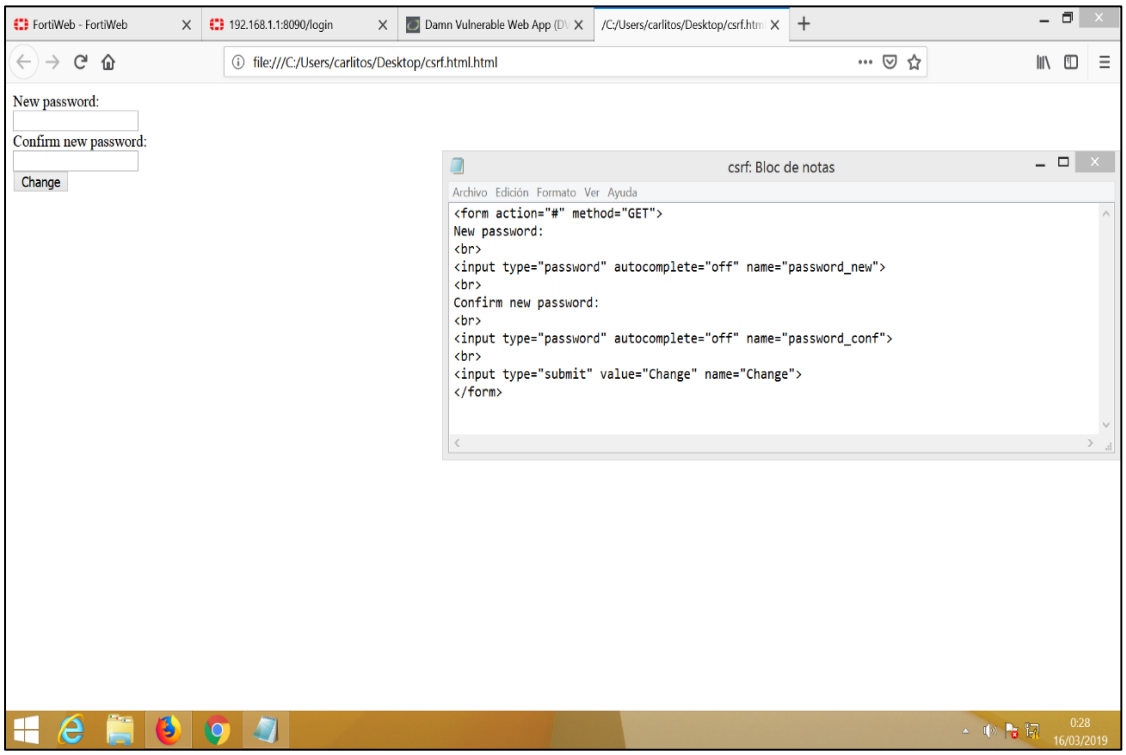

Figura 83. Creando archivo html malicioso

Editamos el archivo csrf.html y le pondremos la clave **hacker** por defecto, ahora este archivo es enviado a un usuario.

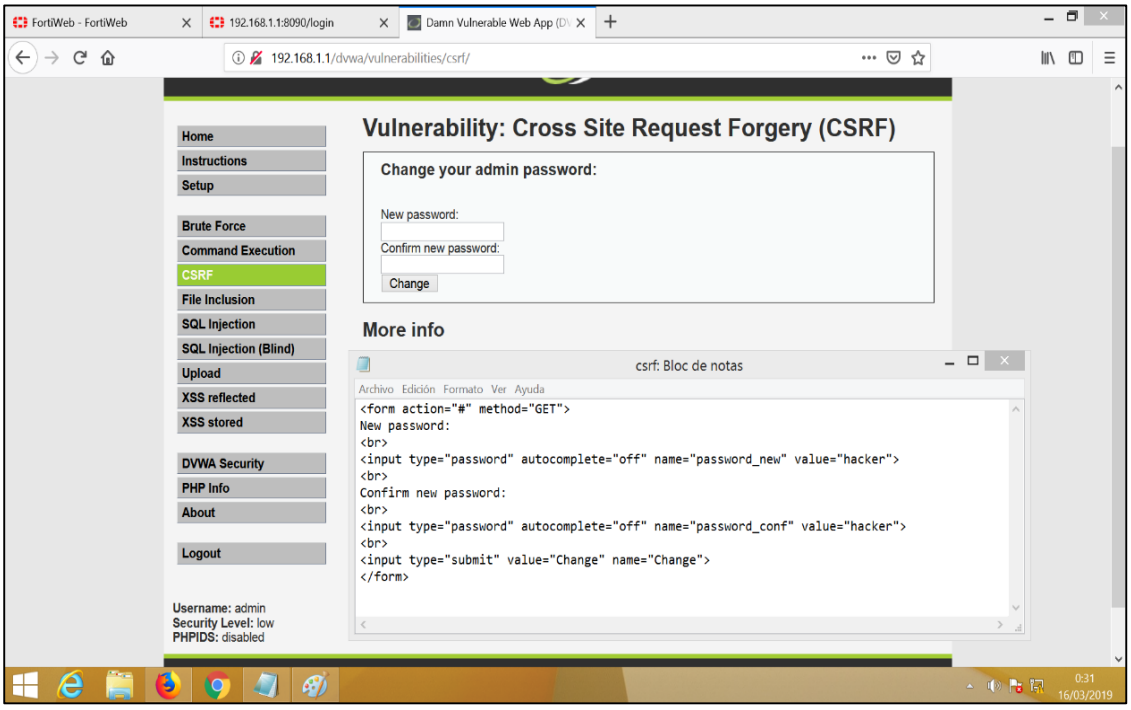

Figura 84. Agregando clave en archivo malicioso

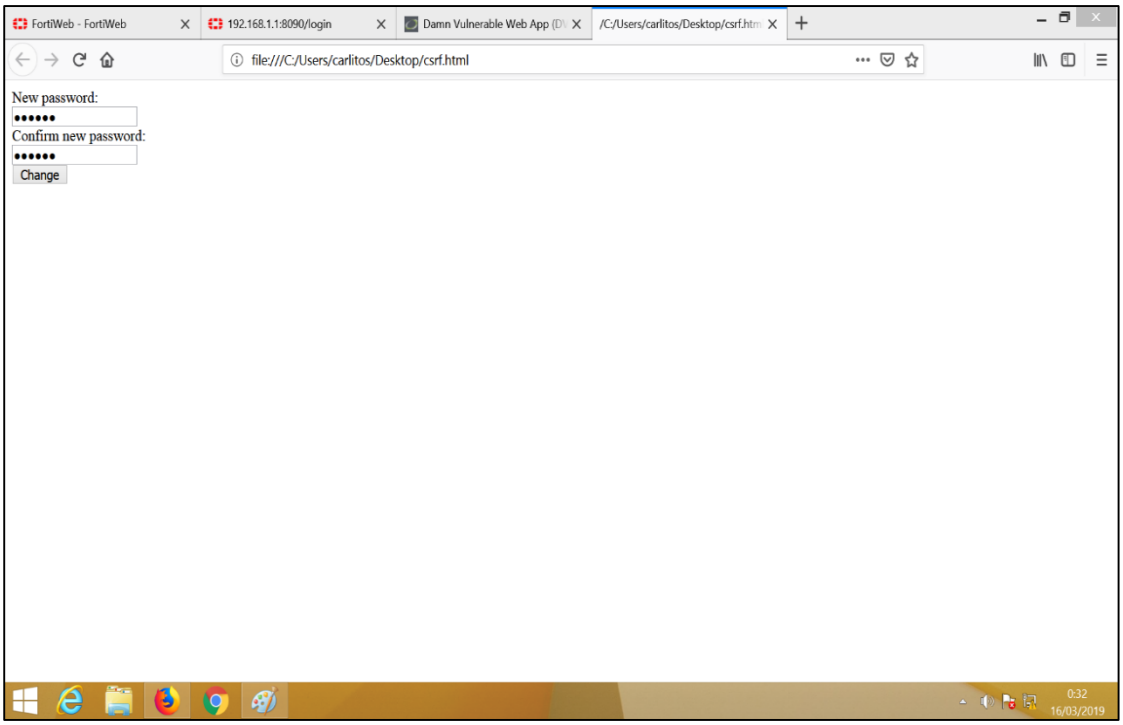

Figura 85. Ejecución del archivo malicioso

Para tener listo nuestro archivo csrf.html vamos a copiar la url original y veremos que sucede.

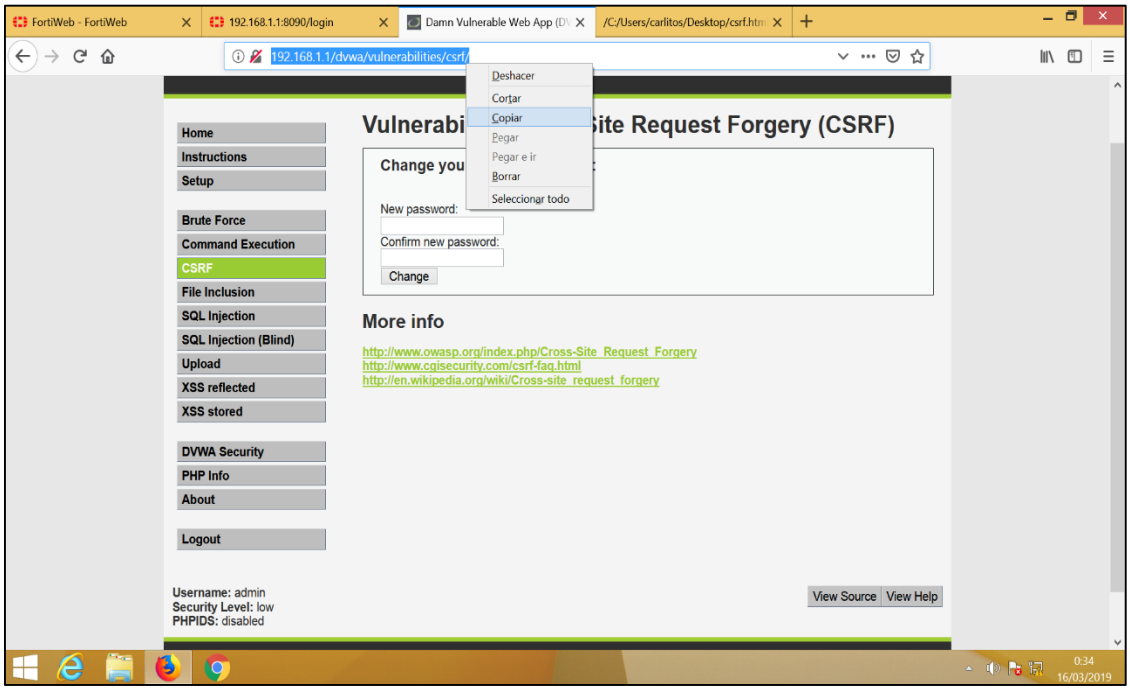

Figura 86. Copiando url del sistema web

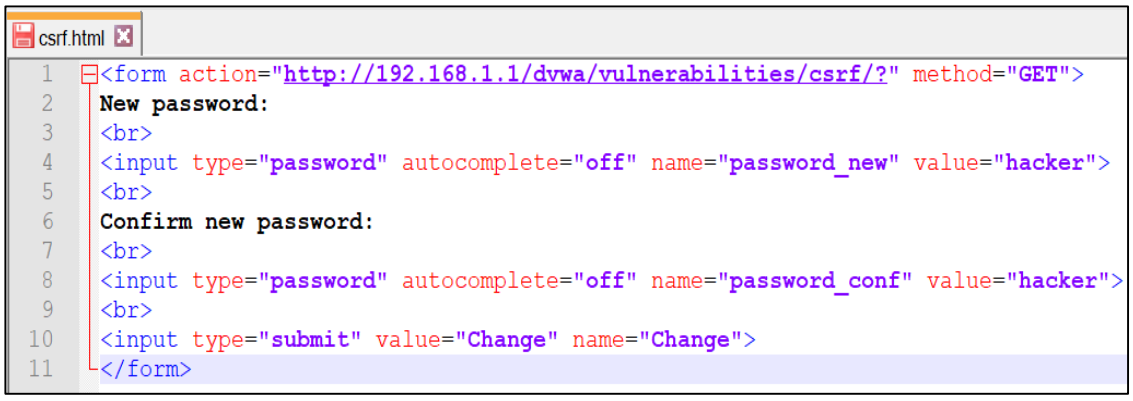

Figura 87. Modificando archivo malicioso

En este ejemplo se hará uso de la ingeniería social y la confianza que el usuario tiene en el atacante para que haga clic en el enlace al documento HTML que llevará a cabo el ataque CSRF.

En caso de no contar con esta ventaja, se podrían emplear otros métodos como por ejemplo combinando un ataque Man in the middle con DNS Spoofing para servirle una web falsa enlazando con nuestro documento HTML en lugar del sitio web original al que pretendía acceder el usuario.

El usuario al recibir el archivo csrf.html ve el formulario y hace clic en **change** realizando correctamente el cambio ya definido en el archivo.

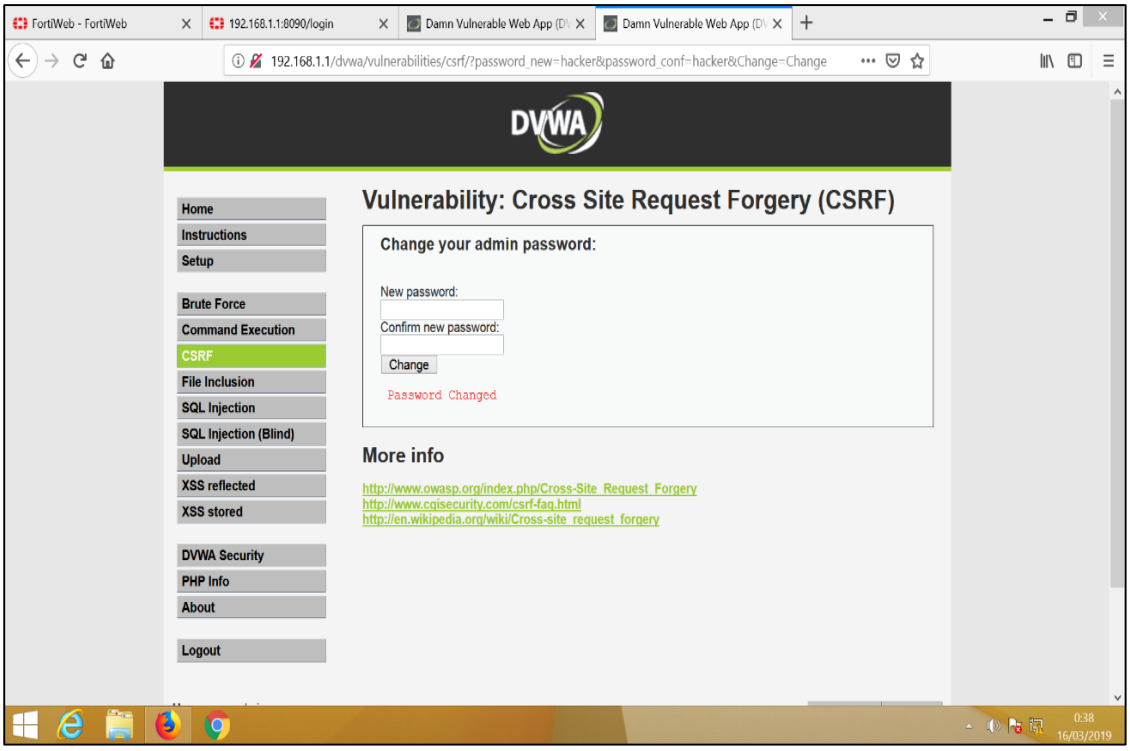

Figura 88. Usuario ejecuta el archivo malicioso, ejecutándose con éxito

Ahora el usuario al intentar acceder con su clave vemos como no logra realizar su ingreso comúnmente.

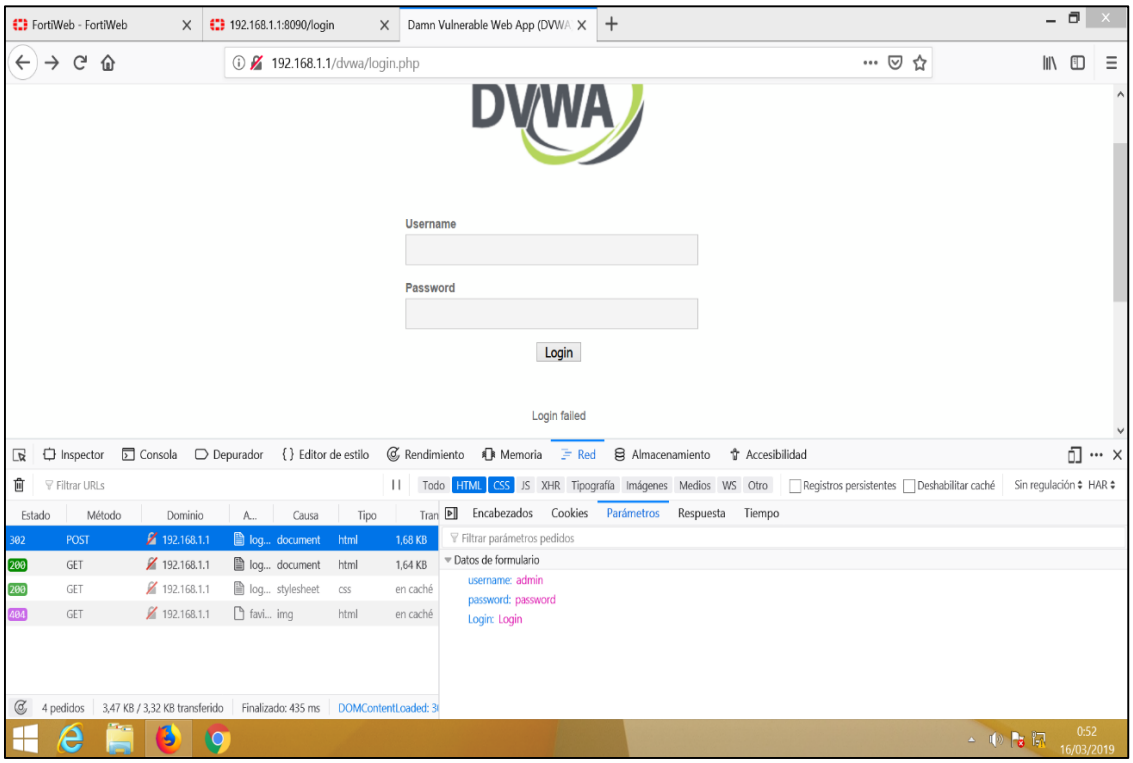

Figura 89. Usuario no puede iniciar sesión con sus credenciales

Ahora intentamos ingresar con el clave **hacker** e ingresamos correctamente.

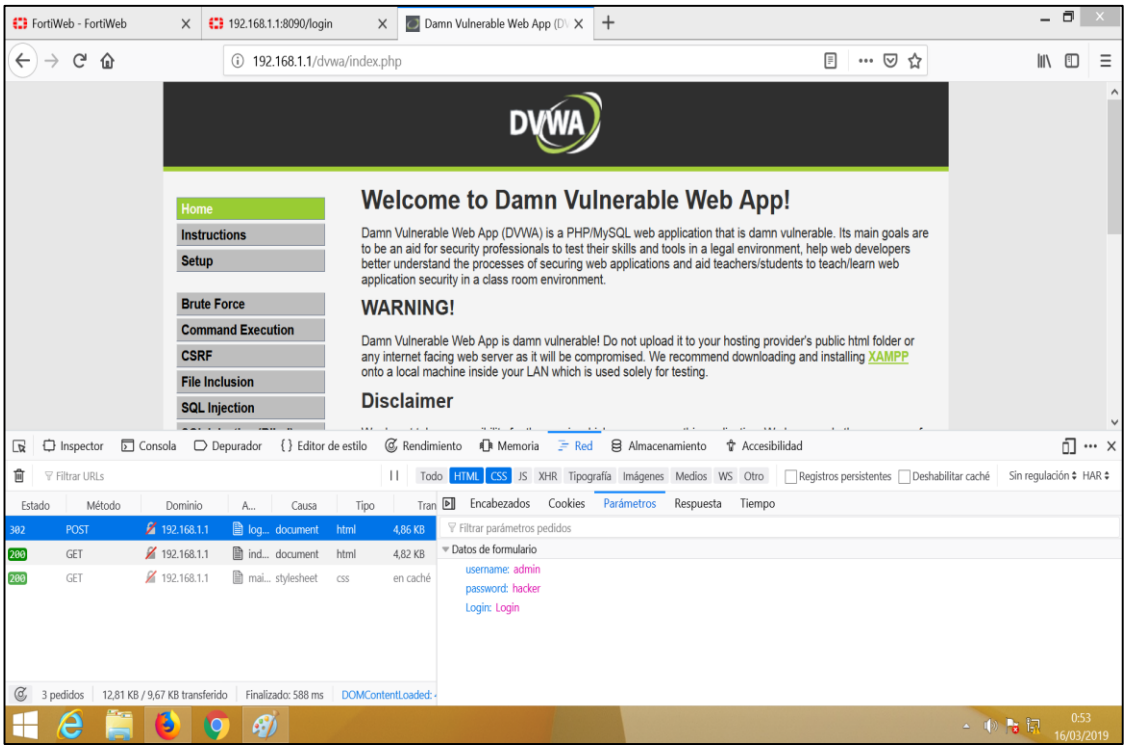

Figura 90. El atacante ingresa con las nuevas credenciales

14.2 Mitigación de falsificación de peticiones entre sitios cruzados CSRF

Fortiweb nos permite protegernos de estos ataques con las firmas que son actualizadas mediante Fortiguard, también podemos unir reglas de acceso a determinadas url y forzar un orden lógico. A continuación, veremos cómo Fortiweb bloquea el ataque **csrf**.

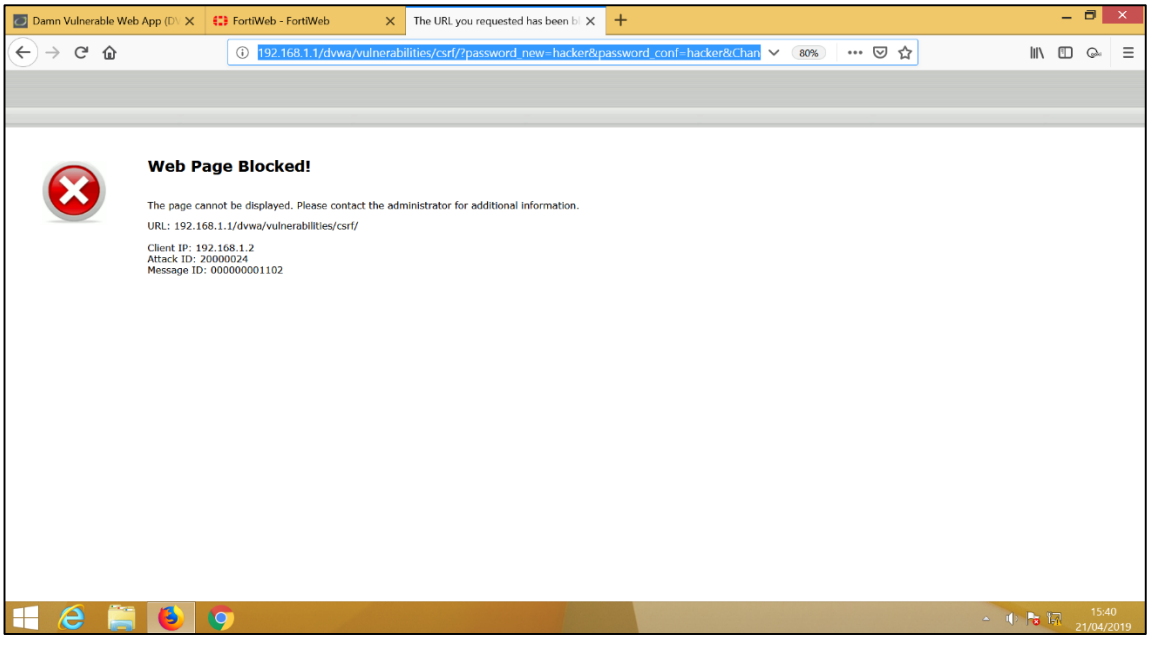

Figura 91. Bloqueo de FortiWeb

Vemos las firmas seleccionadas para protección de este tipo de ataques.

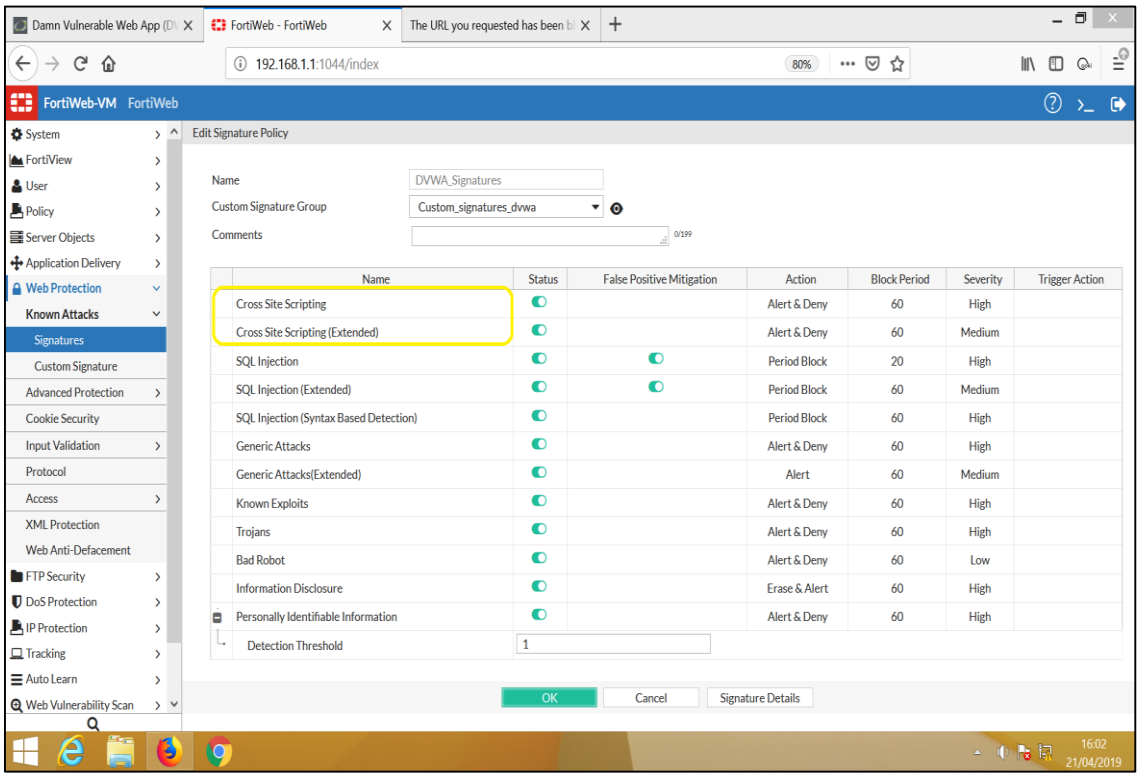

Figura 92. Activando firmas de seguridad

15. Ataques de uso de componentes con vulnerabilidades conocidas

15.1 Uso de componentes con vulnerabilidades conocidas

Puede sorprendernos saber que OWASP no considera que el software sin parche sea la amenaza más grave. Solo ocupa el noveno lugar en la lista de sus 10 amenazas de seguridad más serias, a pesar de que es una de las más comunes. Este ranking se debe en parte a que es el más fácil de defender. Si FortiWeb está buscando exploits y troyanos conocidos, y está bloqueando las fugas HTTPS de Heartbleed, entonces esto le permite un poco de tiempo para parchear sus servidores. Las actualizaciones automáticas en muchos componentes de software de servidor también hacen que esta amenaza sea fácil de combatir.

Ahora vamos a subir un código malicioso, como por ejemplo una **shell PHP**, por medio de un formulario que originalmente ha sido creado para

permitir únicamente la subida de tipos de archivos determinados (generalmente multimedia como imágenes o vídeos), pudiendo llegar a convertirse en una puerta abierta a todo el sistema donde esté alojada la página web. Este tipo de formularios de subida son comúnmente encontrados en redes sociales, foros, blogs e incluso en las bancas electrónicas de algunos bancos, en este caso vamos atacar la librería usada en PHP llamada Upload donde vemos que no existe una correcta **validación** de los archivos que se suben al servidor, como veremos subimos el archivo **csrf.html** y el archivo **urls dvwa.txt**.

```php
<?php
    if (isset($_POST['Upload'])) {

            $target_path = DVWA_WEB_PAGE_TO_ROOT."hackable/uploads/";
            $target_path = $target_path . basename( $_FILES['uploaded']['name']);

            if(!move_uploaded_file($_FILES['uploaded']['tmp_name'], $target_path)) {

                echo '<pre>';
                echo 'Your image was not uploaded.';
                echo '</pre>';

              } else {

                echo '<pre>';
                echo $target_path . ' succesfully uploaded!';
                echo '</pre>';

            }

        }
?>
```

Figura 93. Código php en nivel low

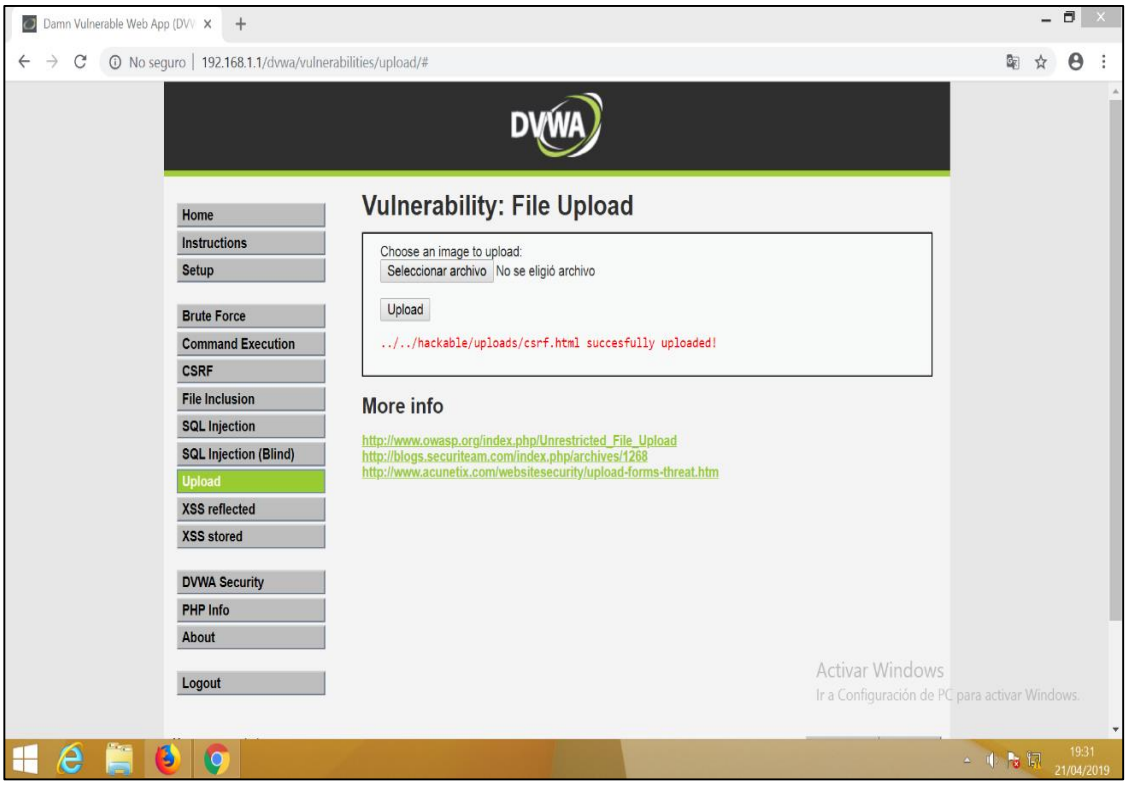

Figura 94. Carga correcta de archivo malicioso

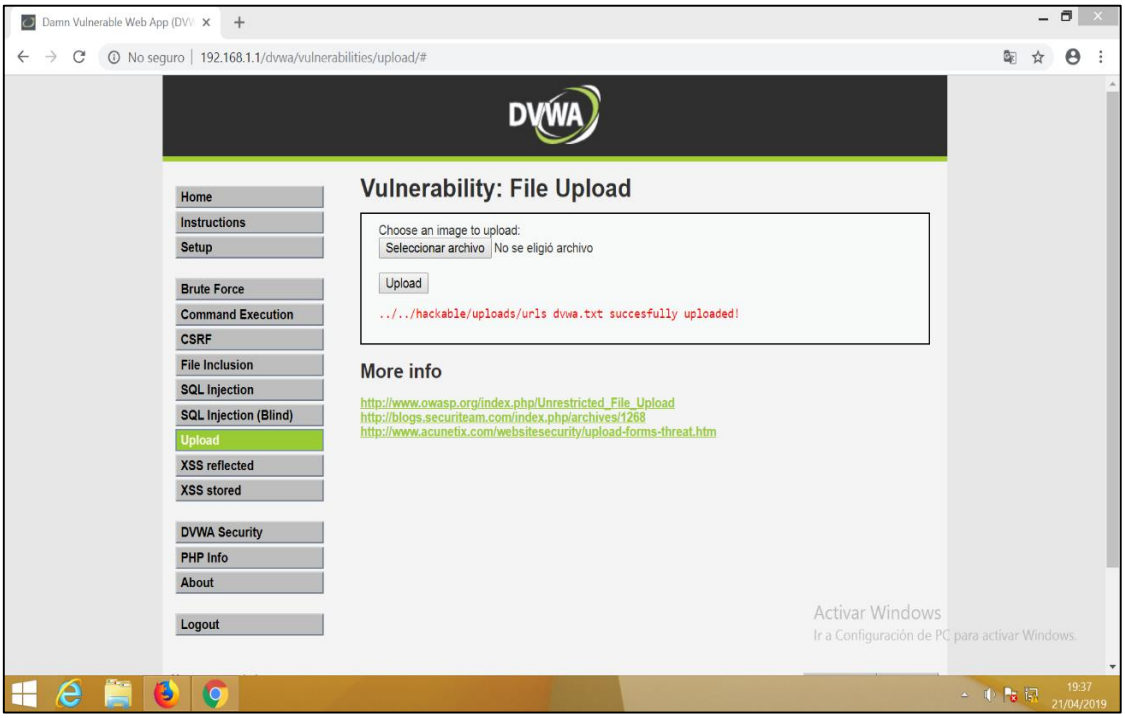

Figura 95. Carga correcta de archivo malicioso

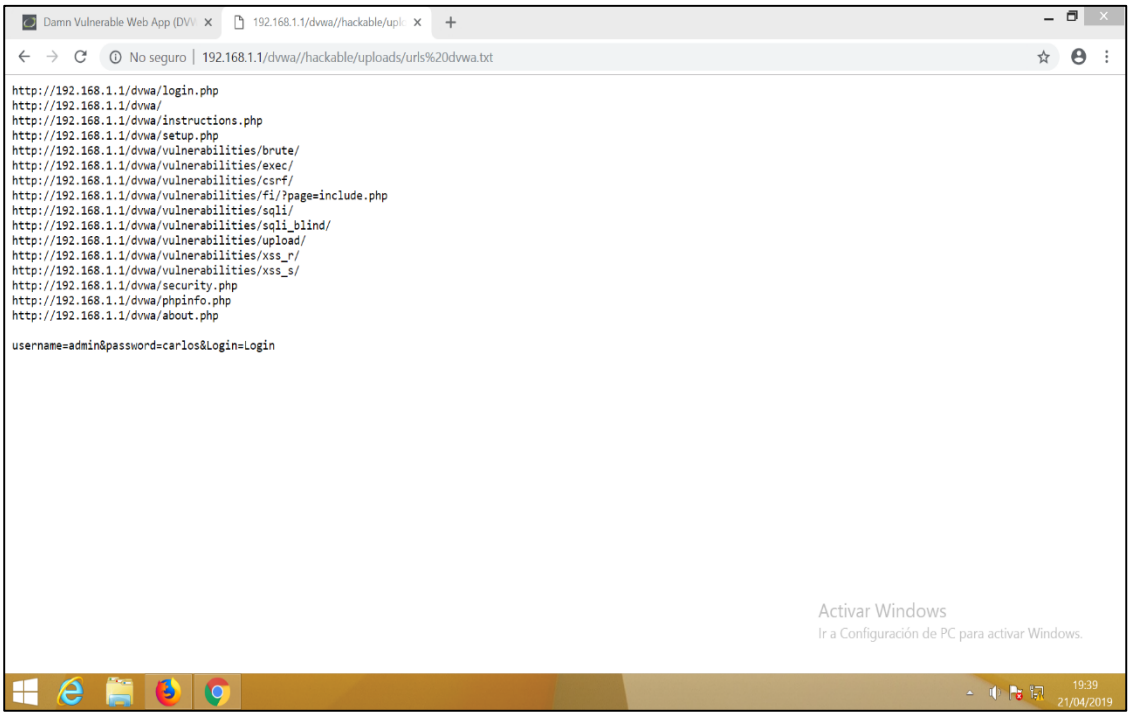

Figura 96. Viendo contenido de archivo malicioso

Ahora vamos a subir un archivo malicioso llamado **.php**, como veremos en la siguiente imagen muestra un mensaje de error a pesar de estar en nivel de seguridad bajo, para poder subir el archivo al servidor vamos a

modificar el **request**.

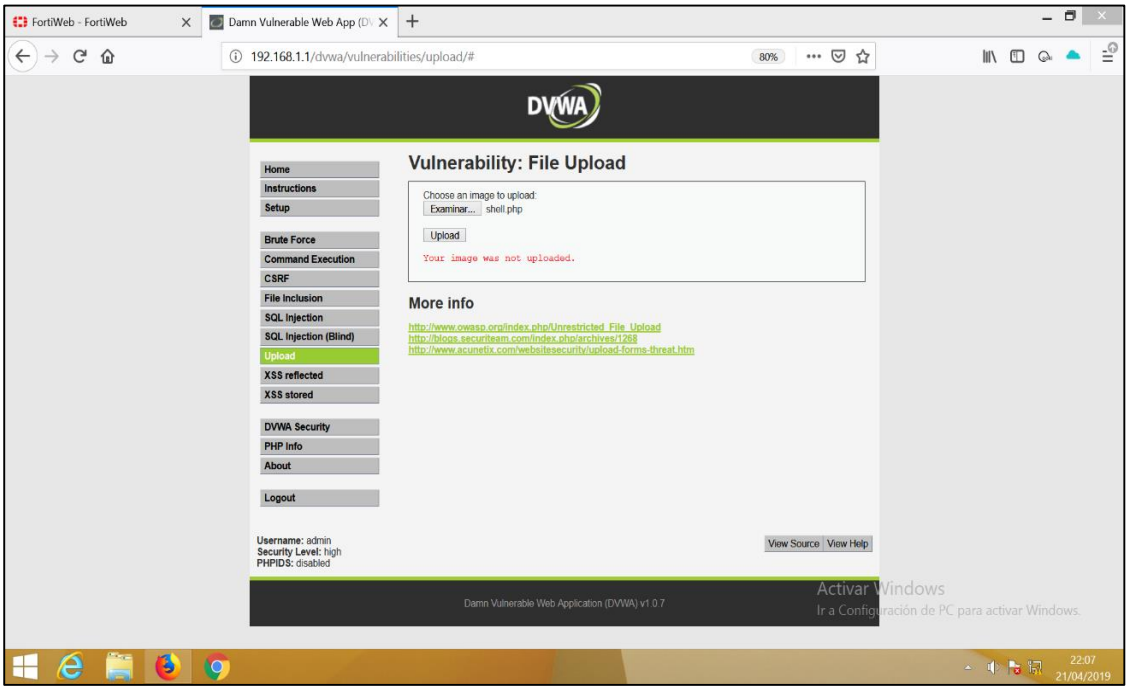

Figura 97. Error al subir archivo php

Para modificar la solicitud request vamos a capturar el tráfico con la herramienta Burp Suite, al momento de subir el archivo **.php** vamos a modificar el request agregando en **content-type application/x-php** , como se ve el archivo se carga correctamente siendo una puerta trasera para futuros ataques.

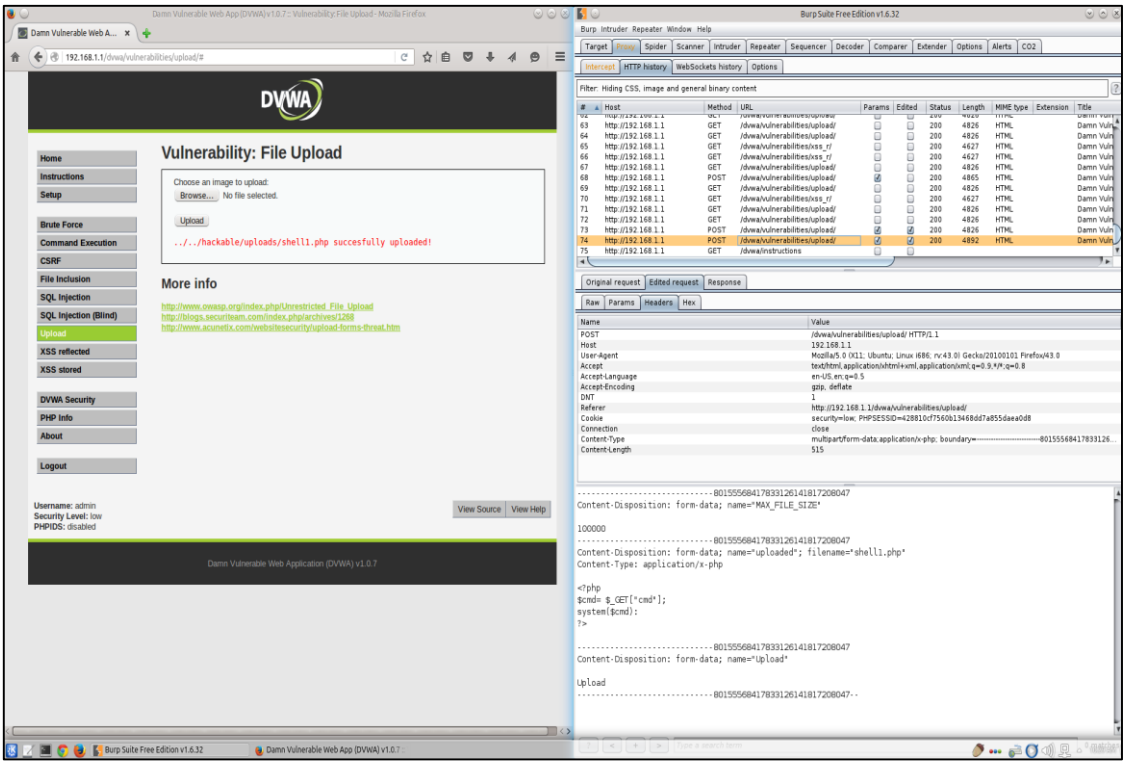

Figura 98. Interceptando tráfico y modificando el request

Ahora vamos a realizar un ataque con **metasploit** desde Kali Linux atacando una vulnerabilidad para PHP que afecta a instalaciones que usan el intérprete CGI.

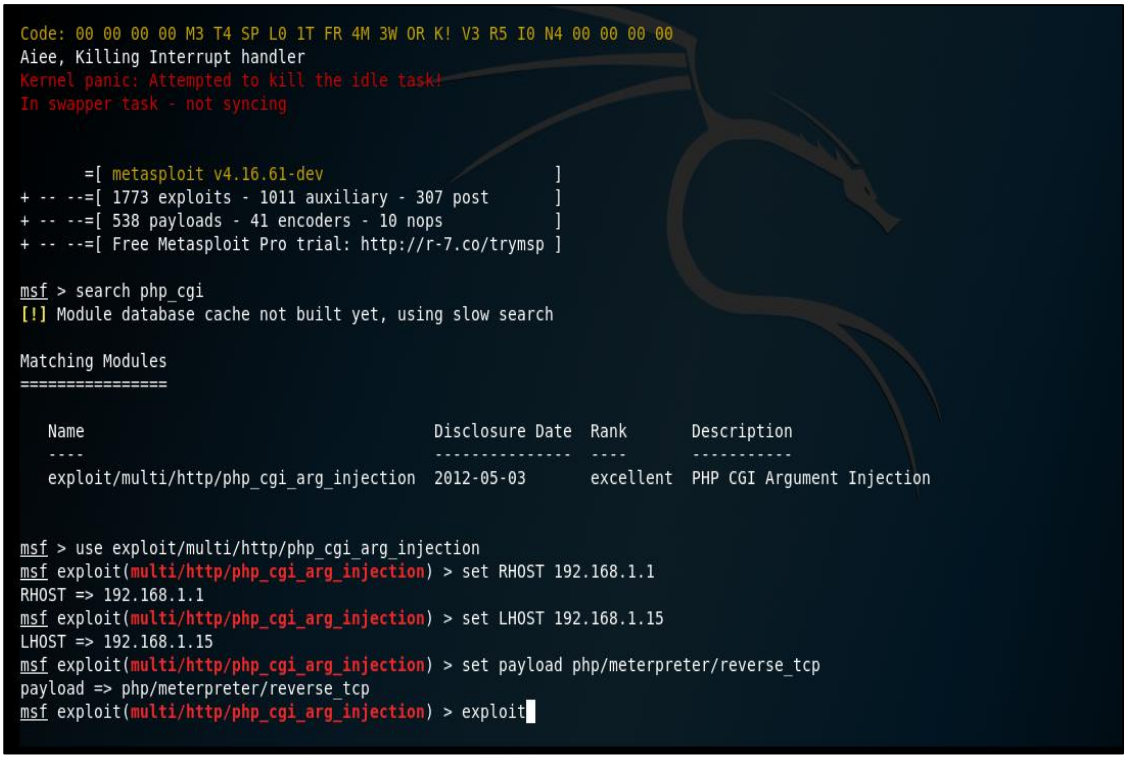

Figura 99. Ejecutamos exploit

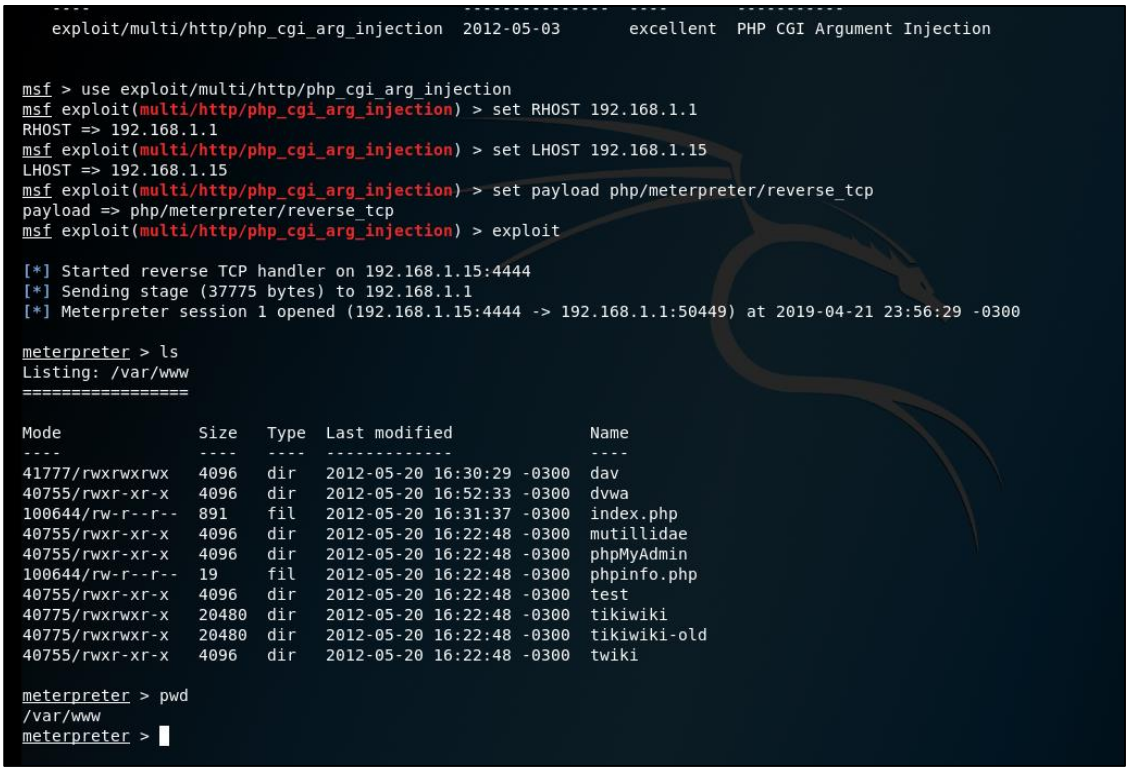

Figura 100. Obtenemos conexión con el servidor web

Como vemos logramos ejecutar un shell en el servidor y podemos ver los directorios y en que carpeta actualmente estamos, ahora subiremos el archivo **c99.php** el cual es un shell php que nos permite usar diferentes funciones para manejar un servidor sin necesidad de entrar en su panel de control.

```
meterpreter > upload '/root/Escritorio/c99.php' /var/www/
[*] uploading  : /root/Escritorio/c99.php -> /var/www/
[*] uploaded   : /root/Escritorio/c99.php -> /var/www//c99.php
meterpreter > ls
Listing: /var/www
==================

Mode              Size    Type  Last modified              Name
----              ----    ----  -------------              ----
100644/rw-r--r--  665712  fil   2019-04-22 12:28:55 -0300  c99.php
41777/rwxrwxrwx   4096    dir   2012-05-20 16:30:29 -0300  dav
40755/rwxr-xr-x   4096    dir   2012-05-20 16:52:33 -0300  dvwa
100644/rw-r--r--  891     fil   2012-05-20 16:31:37 -0300  index.php
40755/rwxr-xr-x   4096    dir   2012-05-20 16:22:48 -0300  mutillidae
40755/rwxr-xr-x   4096    dir   2012-05-20 16:22:48 -0300  phpMyAdmin
100644/rw-r--r--  19      fil   2012-05-20 16:22:48 -0300  phpinfo.php
40755/rwxr-xr-x   4096    dir   2012-05-20 16:22:48 -0300  test
40775/rwxrwxr-x   20480   dir   2012-05-20 16:22:48 -0300  tikiwiki
40775/rwxrwxr-x   20480   dir   2012-05-20 16:22:48 -0300  tikiwiki-old
40755/rwxr-xr-x   4096    dir   2012-05-20 16:22:48 -0300  twiki

meterpreter >
```

Figura 101. Subiendo shell en el servidor

Ahora ingresaremos al shell donde vemos los directorios del servidor web, a continuación, vamos a modificar los permisos del archivo **login.php** dentro de la carpeta **DVWA**.

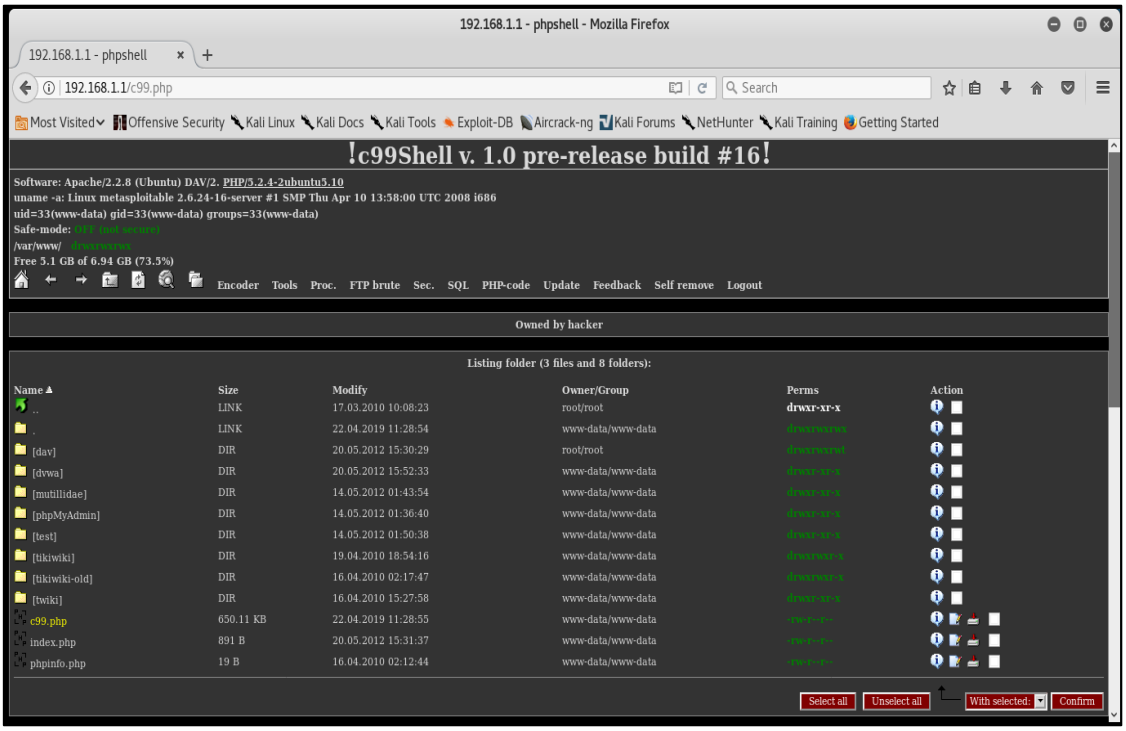

Figura 102. Ejecución de shell desde Kali Linux

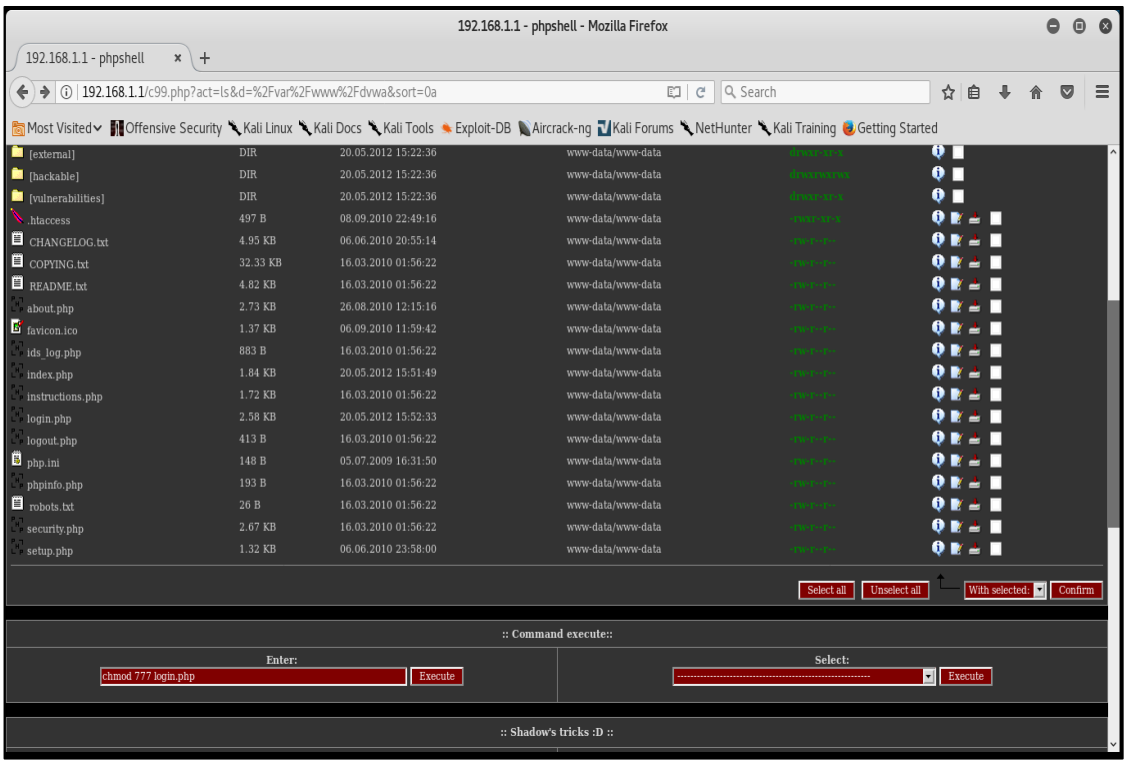

Figura 103. Modificando permisos de archivo login.php

Ahora modificamos el archivo **login.php** para poder realizar un defacement, al acceder al sistema vemos que el inicio de sesión fue modificado.

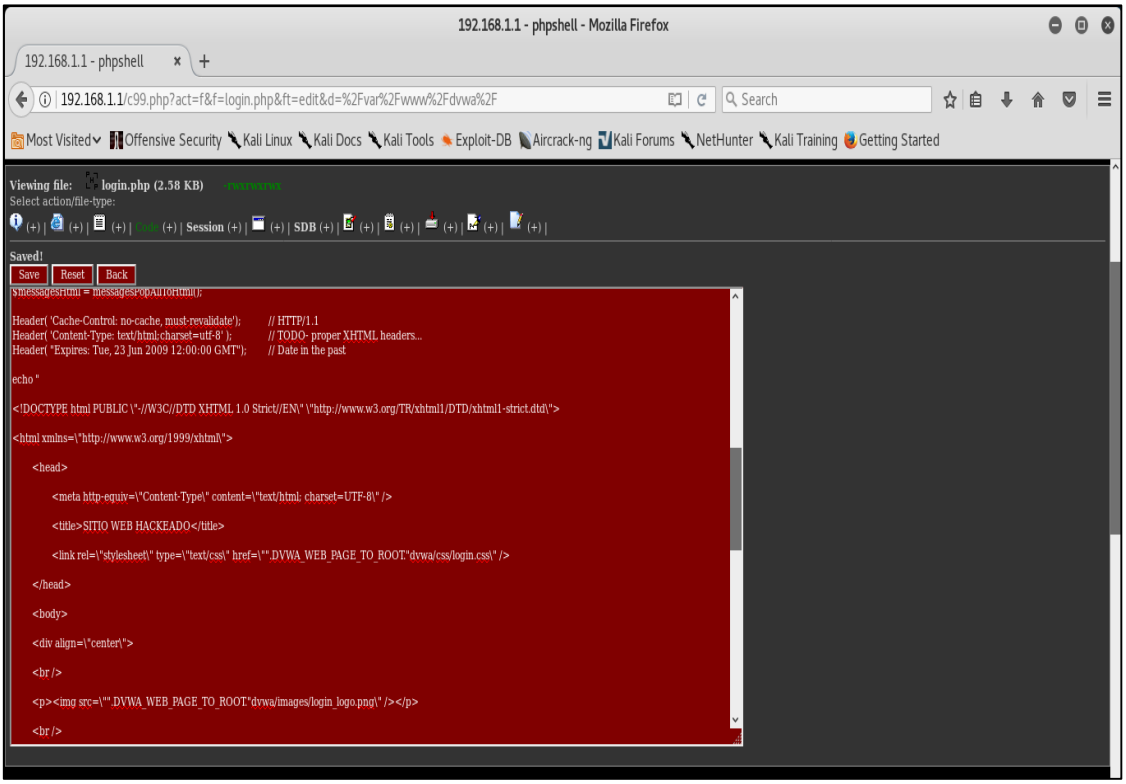

Figura 104. Modificando archivo login.php

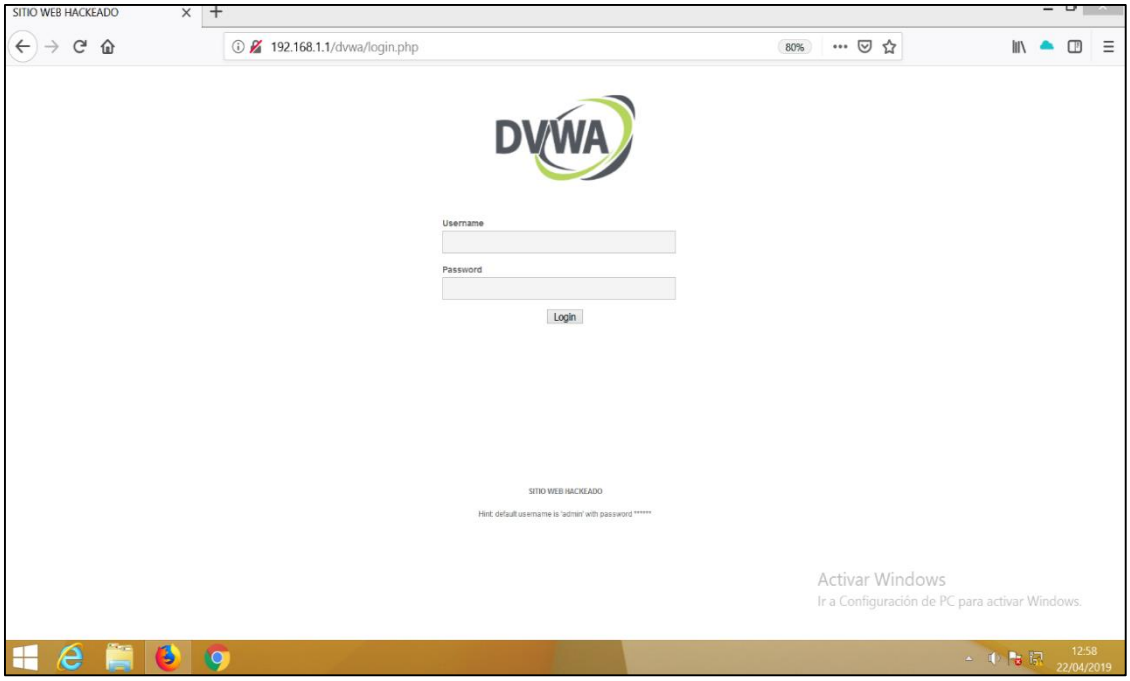

Figura 105. Visualizamos que la página web fue modificada

15.2 Mitigación en uso de componentes con vulnerabilidades conocidas

Fortiweb nos permite la descarga de HTTPS esto se configura directamente en la política del servidor. La opción de bloquear exploits conocidos y subidas de troyanos se configura en dos lugares: el conjunto de firmas y las restricciones de carga de archivos.

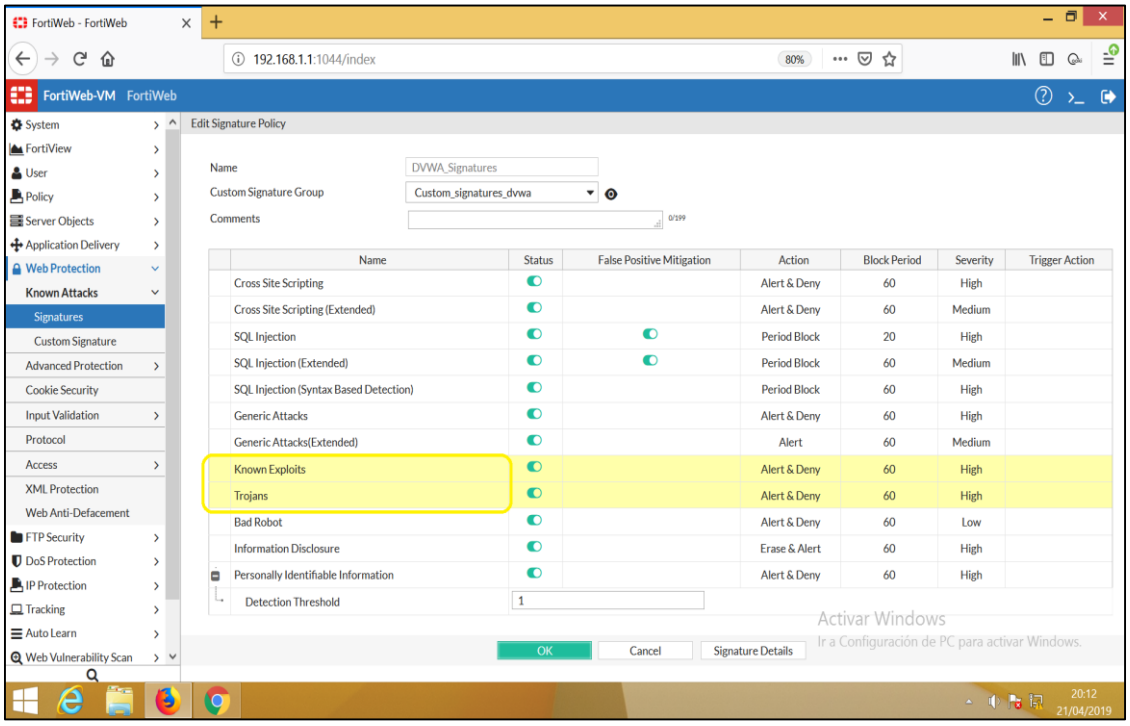

Figura 106. Activando firmas de seguridad

Ahora vamos a configurar la seguridad de archivos para solo permitir la carga de archivos JPG y PNG en la url **192.168.1.1/dvwa/** y activando el antivirus en la política.

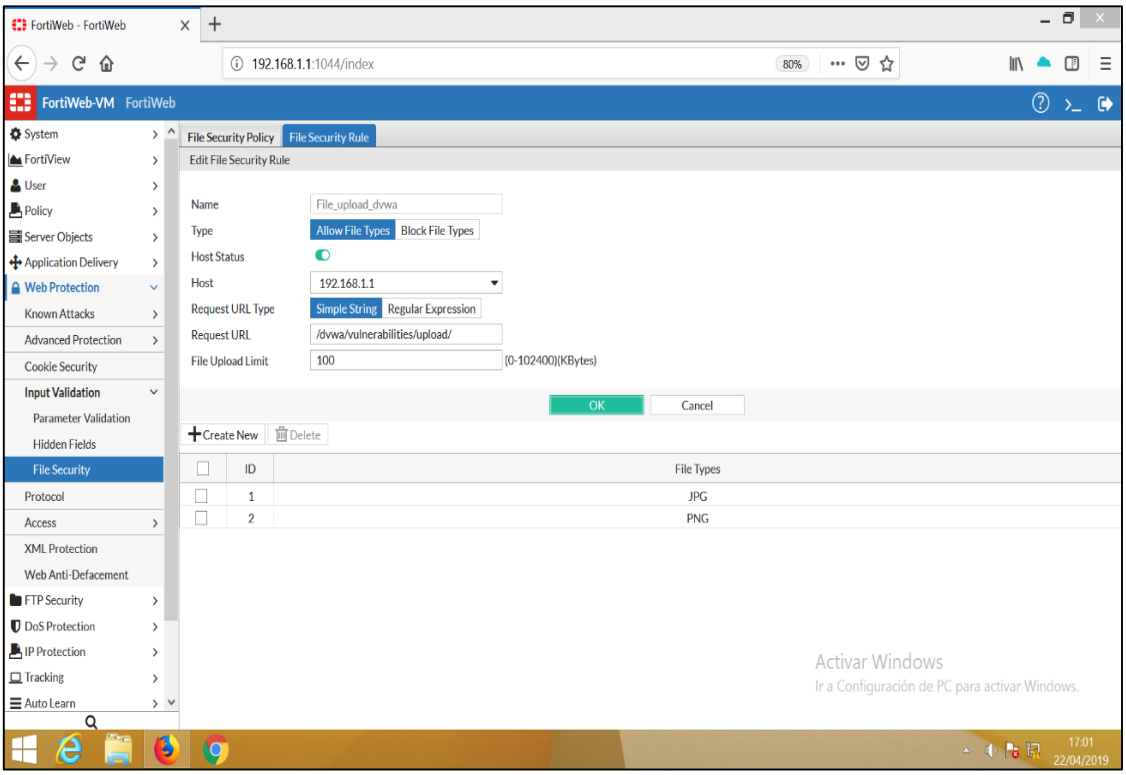

Figura 107. Creación de regla para permitir la carga de ciertos archivos

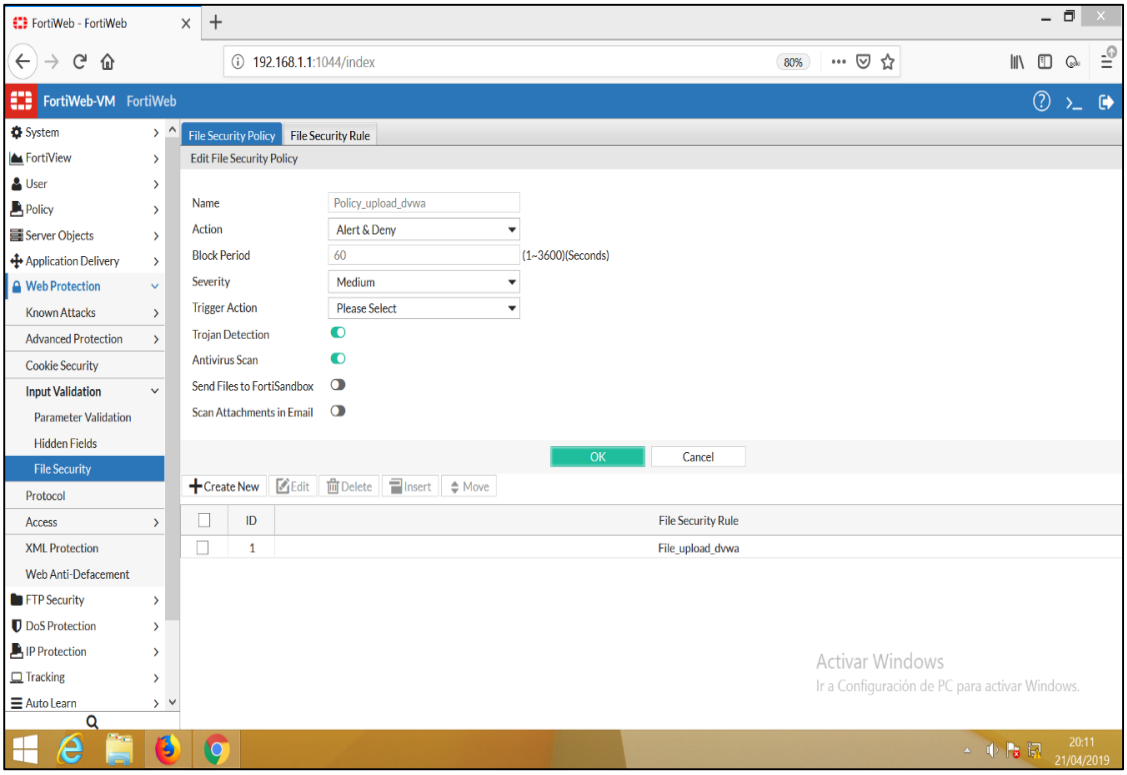

Figura 108. Activando escáner de antivirus y troyanos

Intentamos subir el archivo urls **dvwa.txt** y vemos como Fortiweb bloquea la carga del archivo.

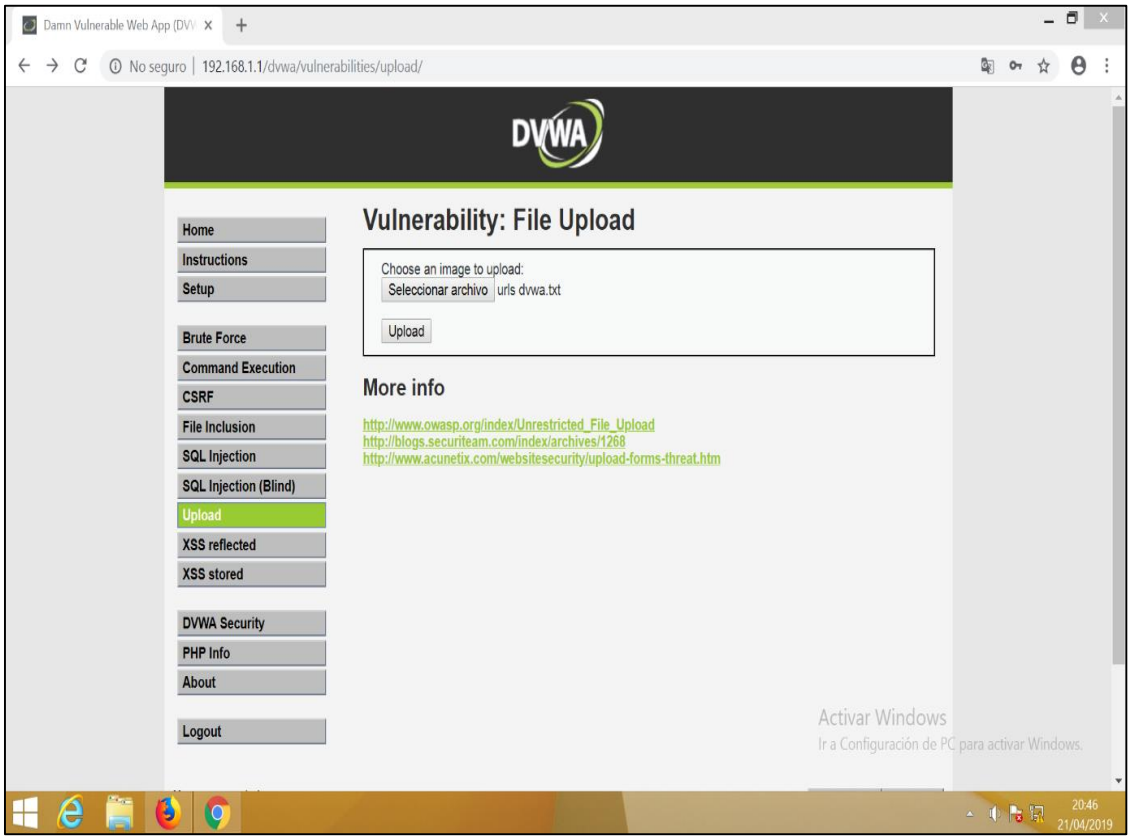

Figura 109. Subiendo archivo txt malicioso

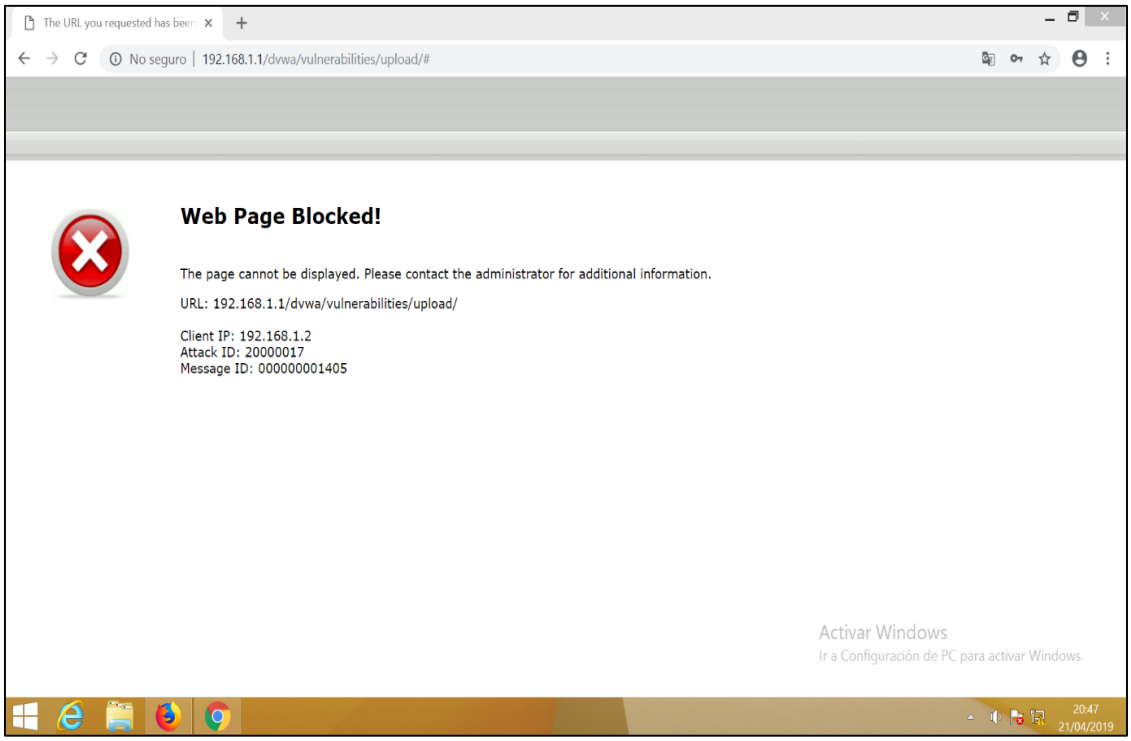

Figura 110. Bloqueo de FortiWeb

A continuación, se puede visualizar los logs producidos por el bloqueo de subir archivos no admitidos y cuando acceden el tamaño máximo establecido al servidor.

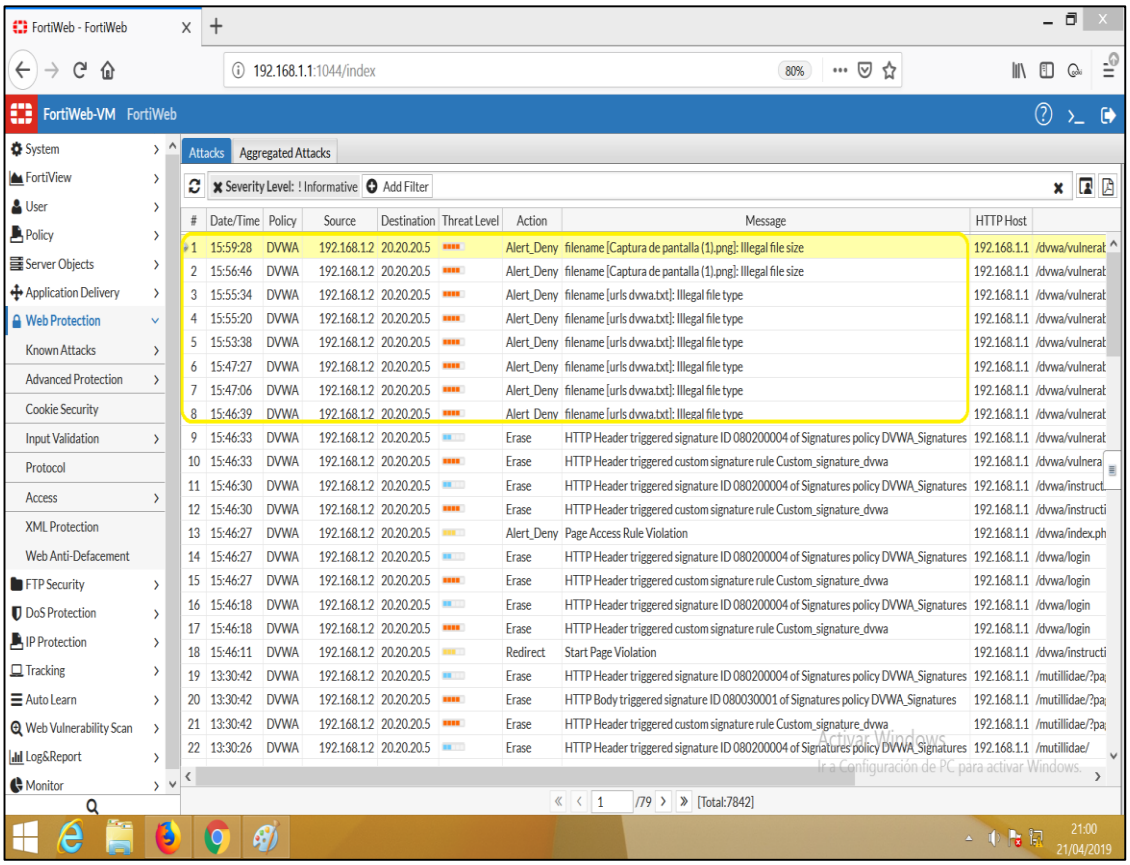

Figura 111. Logs que muestran el bloqueo en Fortiweb

Ahora intentaremos ejecutar el **exploit** desde Kali Linux y veremos cómo Fortiweb logra bloquear el ataque.

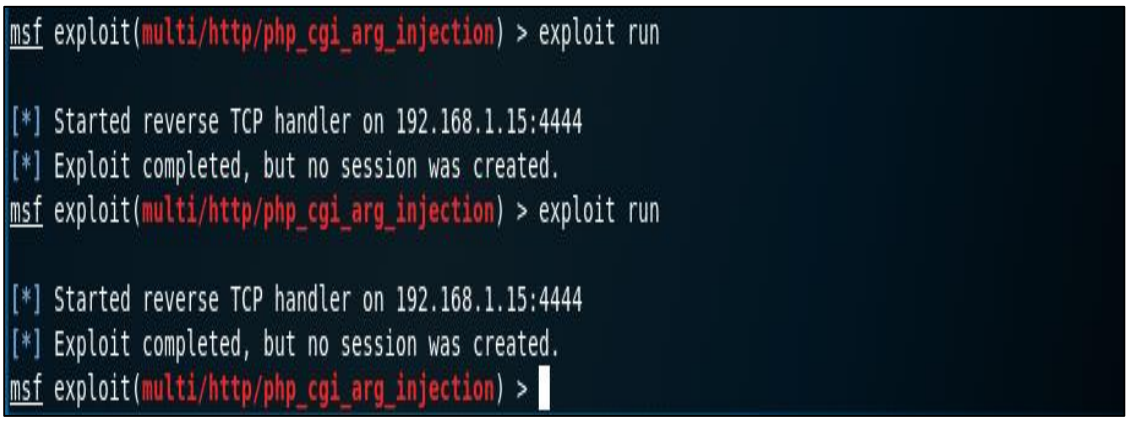

Figura 112. Bloqueo de Fortiweb al intento de ejecución de exploit

Ahora veremos logs en FortiWeb, donde visualizaremos el bloqueo y podremos ver la ip origen del atacante para poder agregarlo a una lista negra.

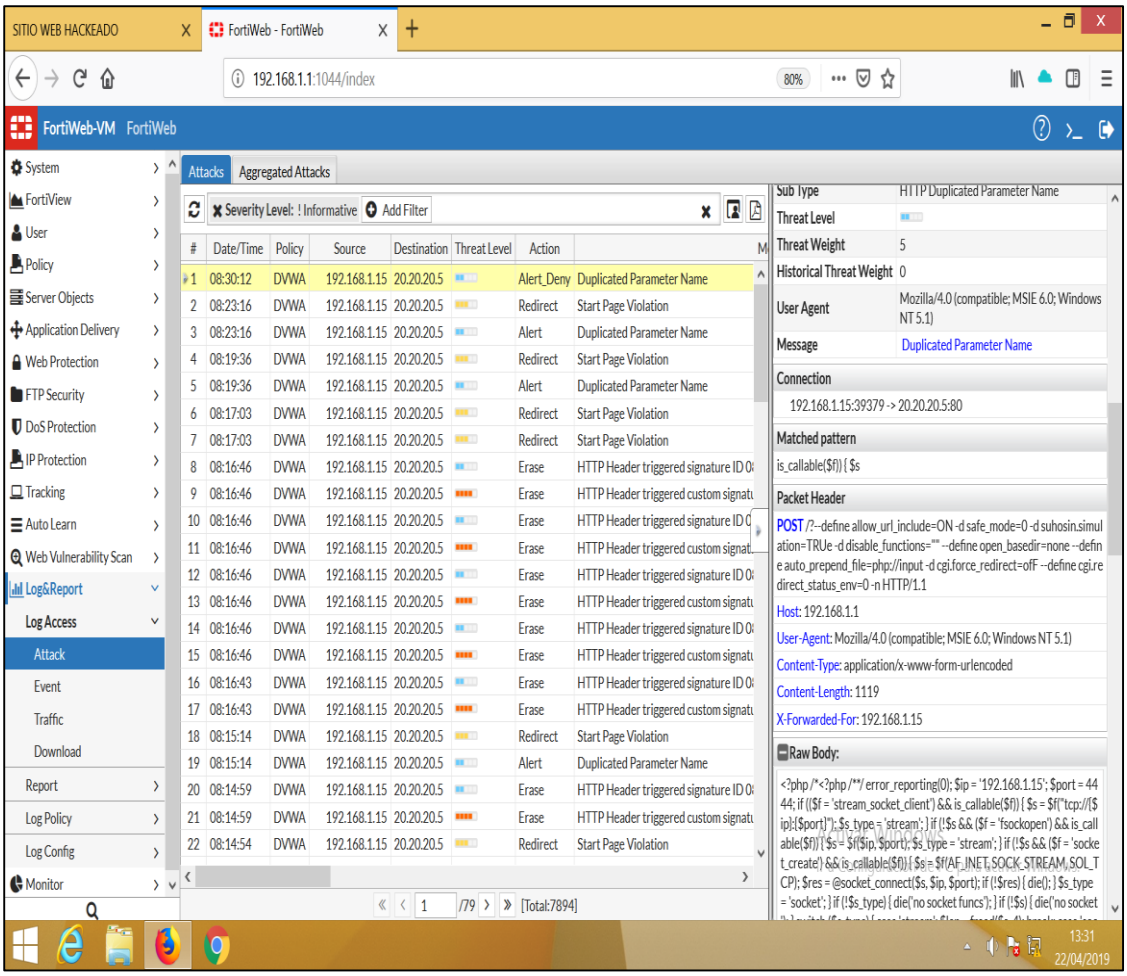

Figura 113. Log que muestra el bloqueo de Fortiweb a la conexión de exploit

Adicionalmente Fortiweb nos permite configurar un Anti defacemente, mantiene hashes de archivos en su Apache, IIS u otro directorio de sitio web. Periódicamente, FortiWeb se conecta al servidor para ver si los archivos han cambiado. Si detecta un cambio y usted no le dijo explícitamente a FortiWeb que ocurriría un cambio autorizado, entonces FortiWeb puede enviarle un correo electrónico o revertir automáticamente los archivos a una copia limpia. Esto puede ayudar a minimizar el impacto de desfiguraciones masivas de unidades de disco en los proveedores de hosting, mientras trabajas para descubrir y analizar el agujero de seguridad.

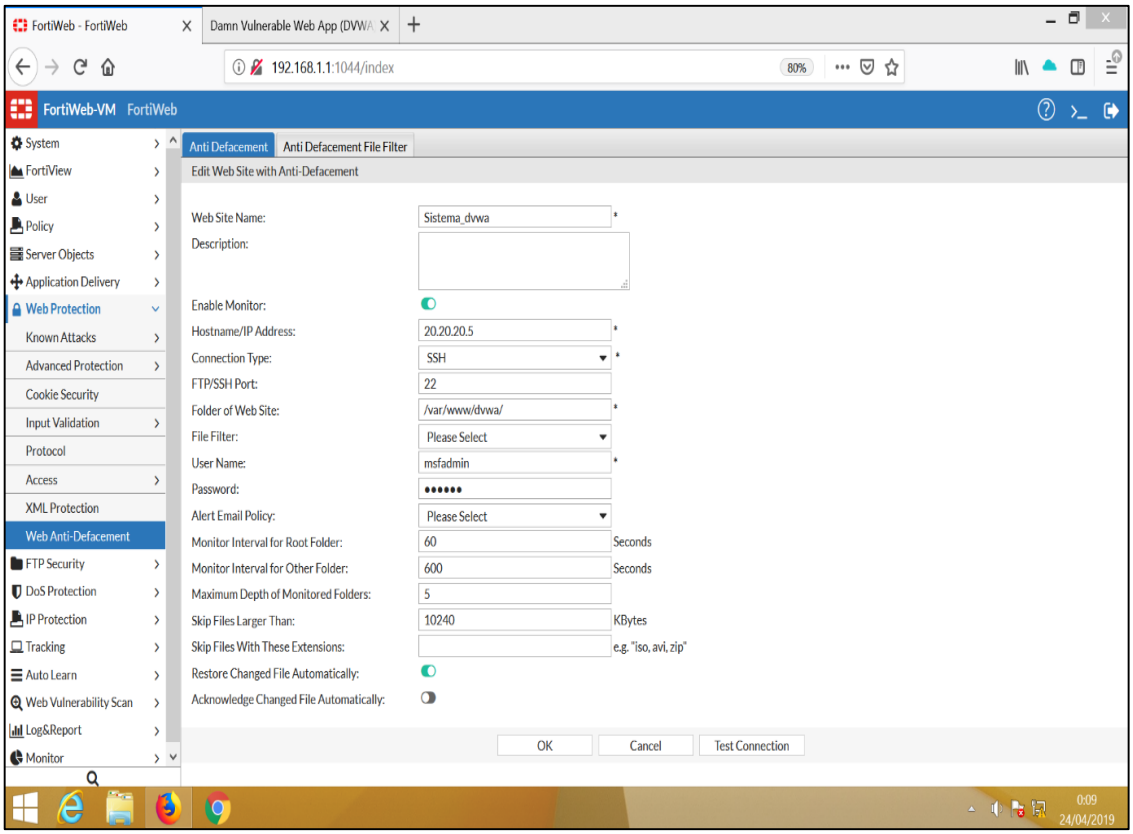

Figura 114. Configuración de antidefacement en Fortiweb

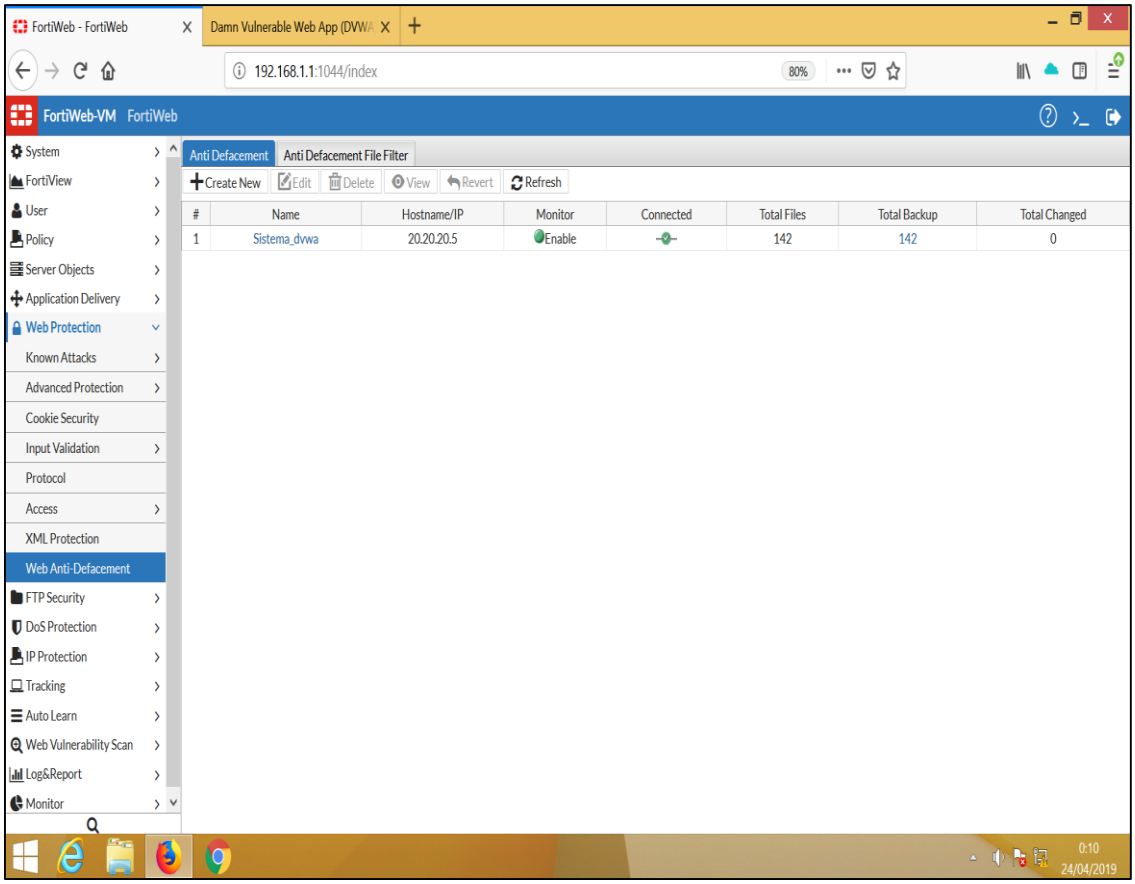

Figura 115. Verificando el estado de conexión con el servidor web

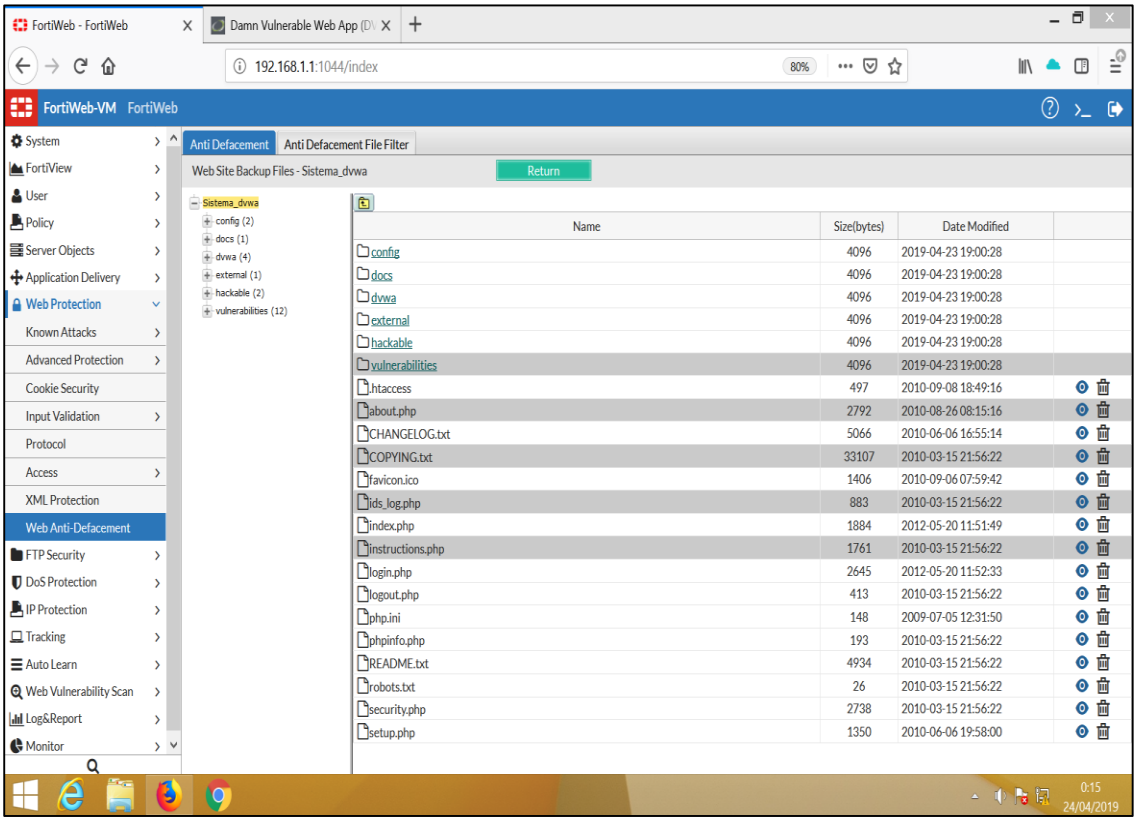

Figura 116. Archivos copiados por Fortiweb

16. Ataque de redirecciones y reenvíos no validados

16.1 Redirecciones y reenvíos no validados

Al igual que A4, A10 trata con las entradas que la aplicación web no ha validado. En este caso, es específicamente entradas para redirecciones. Las aplicaciones deben verificar ambos:

- Destino válido (si se permiten redireccionamientos externos).
- Autorización de la persona para ir a esas páginas web.

El atacante crea enlaces para que la víctima haga clic que luego son llevados a una aplicación de confianza y allí los atacantes instalan el código malicioso. Veremos cómo después de realizar el inicio de sesión en el sistema nos redirecciona a la url **index.php**, podemos ver el código de estado 302 que significa redireccionamiento.

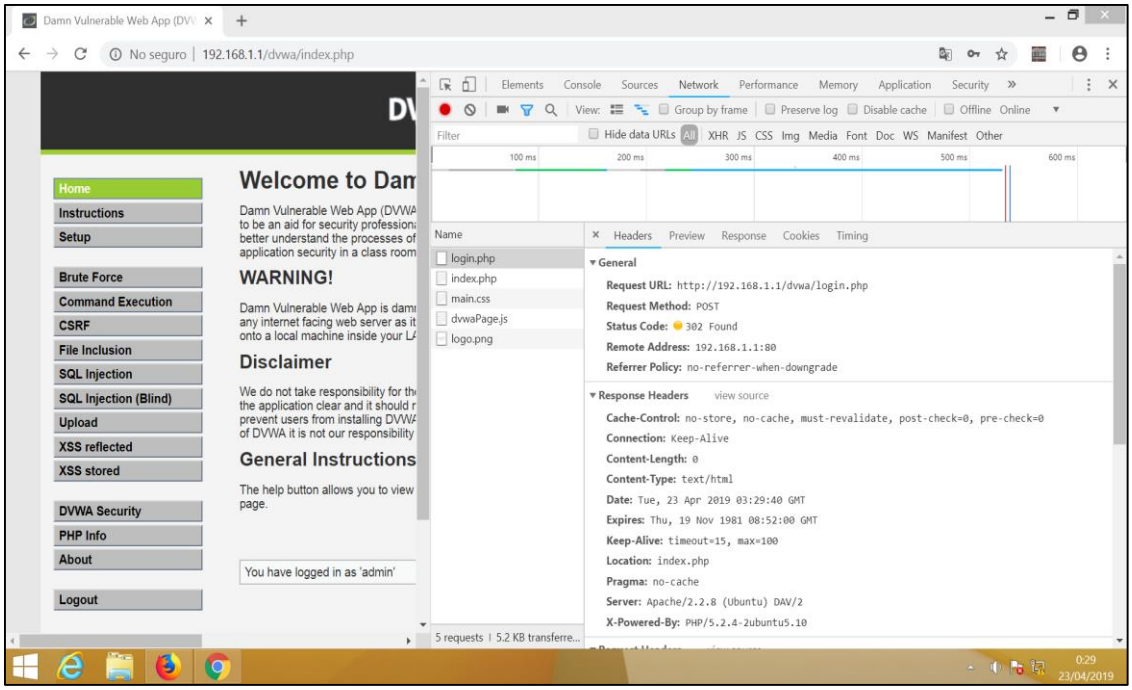

Figura 117. Código de estado 302

Veremos un ejemplo de como un **pishing** puede engañar fácilmente a un usuario para así poder obtener sus credenciales. Tenemos un usuario malicioso que tiene en su servidor web con 3 archivos, un archivo **hack.html** que tiene la apariencia del sistema web **DVWA**, también cuenta con un archivo **contra.php** el cual tiene definido poder capturar las credenciales ingresadas en la página falsa luego de ello los redirecciona hacia el sitio web real y por último tenemos un archivo **cuentas.txt** para poder almacenar las credenciales.

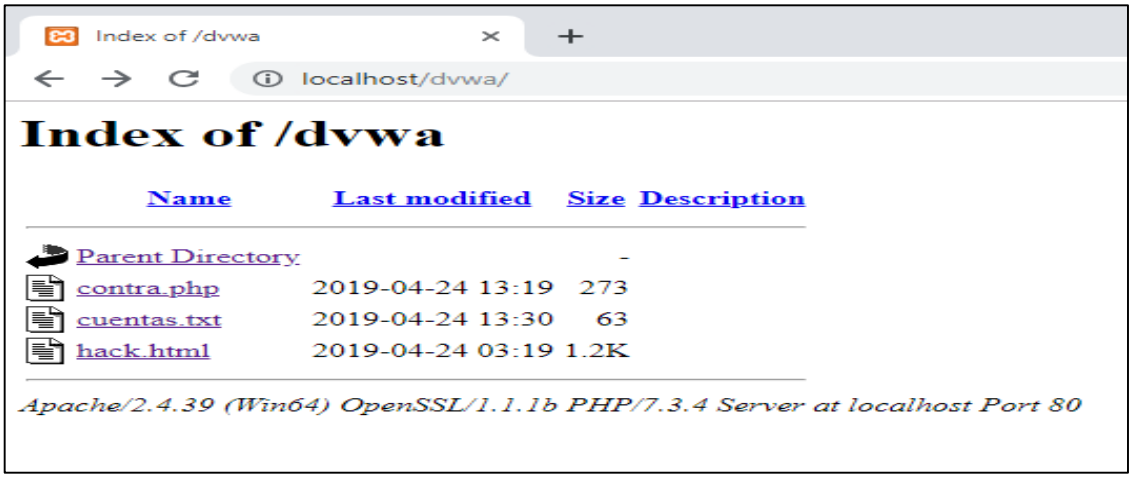

Figura 118. Archivos creados en servidor web malicioso

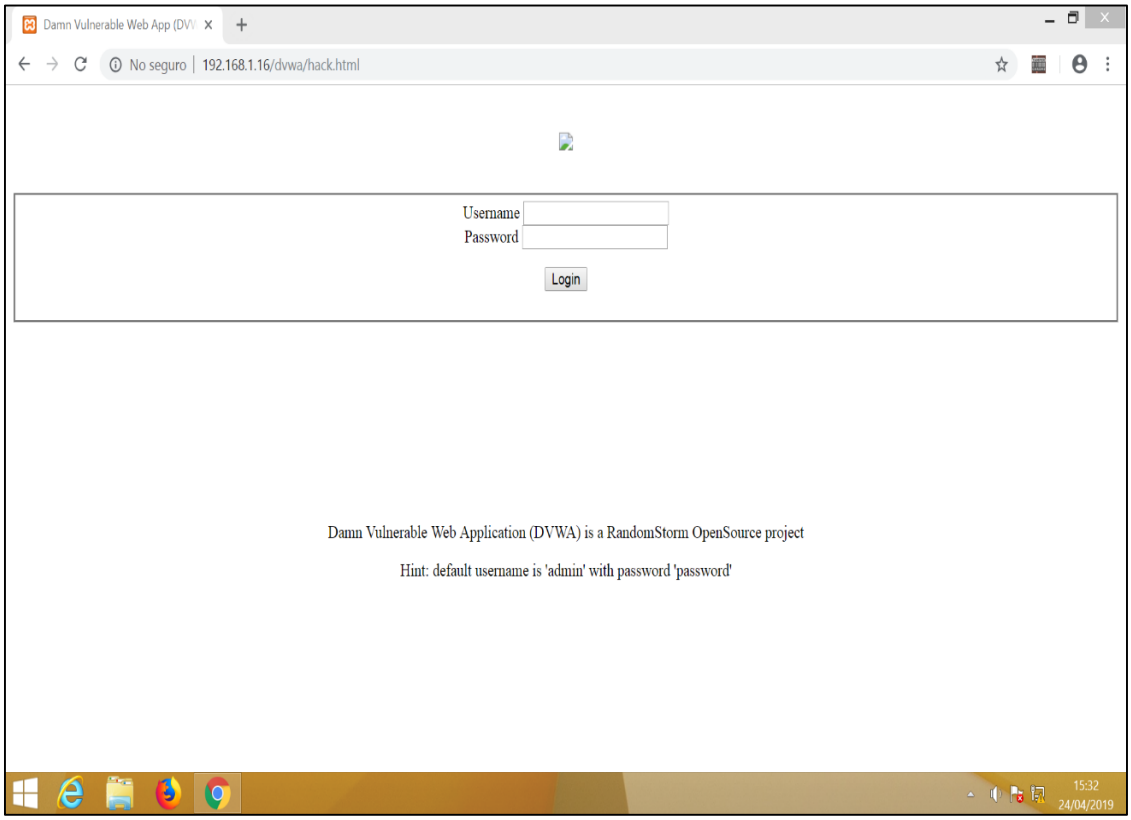

Figura 119. Usuario recibe url maliciosa e ingresa sus credenciales

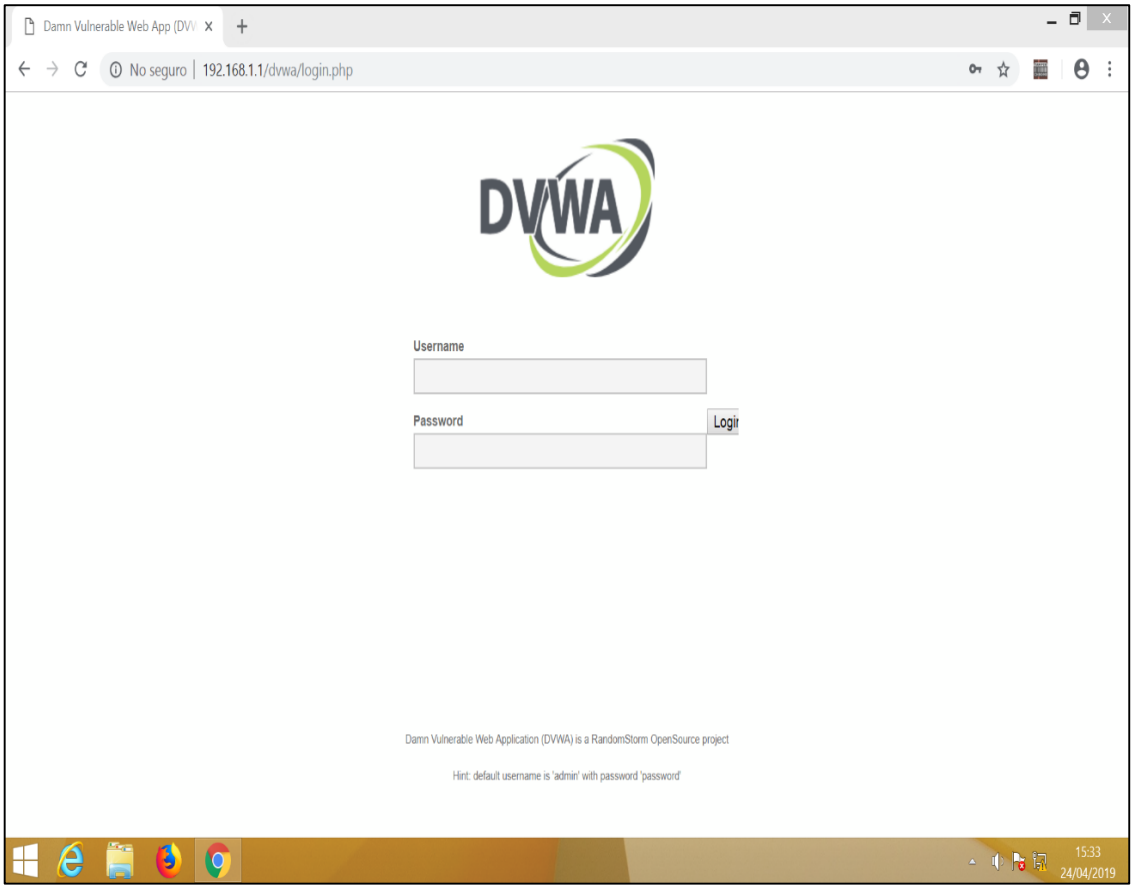

Figura 120. Usuario es redireccionado al verdadero sitio web

A continuación, vemos las credenciales grabadas en el archivo **cuentas.txt.**

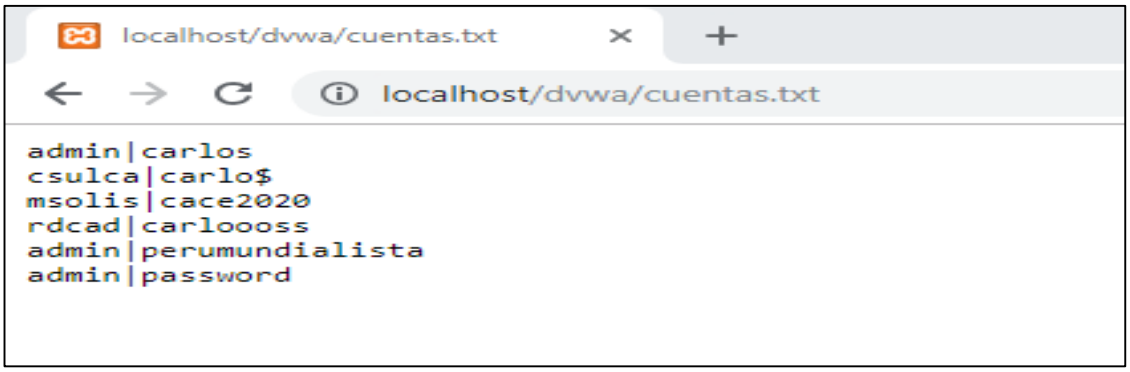

Figura 121. Atacante almacenó las credenciales del usuario

Ahora ingresaremos un script en **XSS stored** que nos permitirá redirigiros a un sitio web malicioso, como veremos se ejecuta con normalidad.

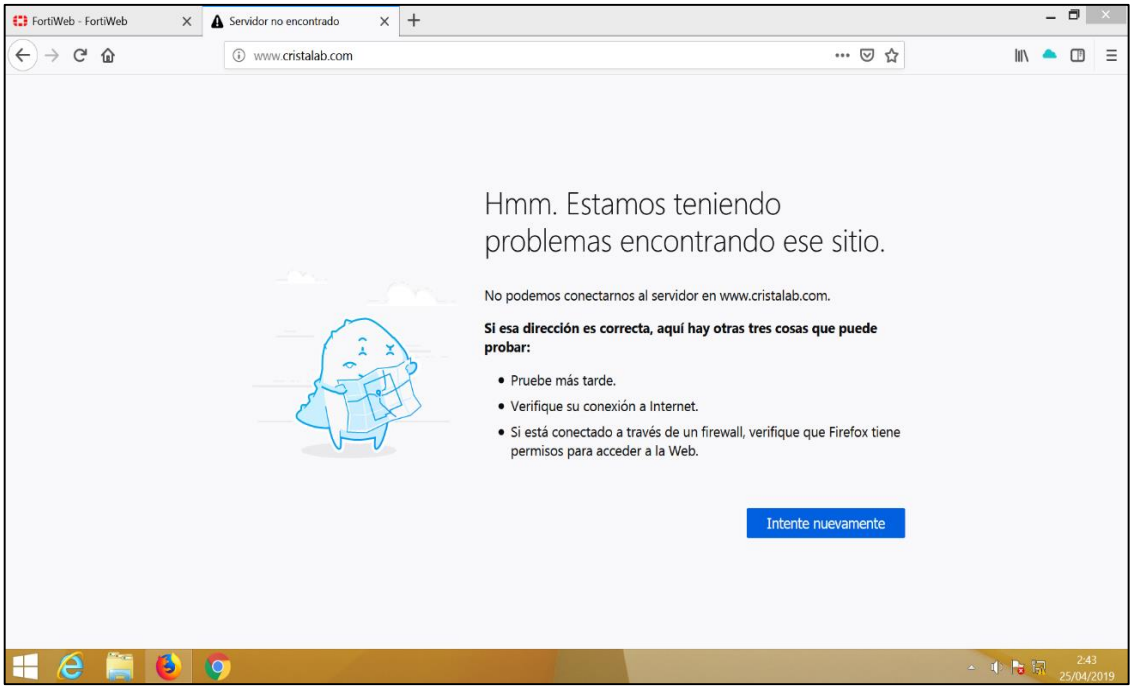

Figura 122. Script malicioso ejecuta un redireccionamiento

El script usado es el siguiente:

```
<body>
<script type="text/javascript">
window.location="http://www.cristalab.com";
</script>
</body>
```

Figura 123. Script malicioso

16.2 Mitigaciones para ataques de redirecciones y reenvíos no validados.

FortiWeb nos permite realizar reglas de validación de entradas, pero específicamente sobre temas de redireccionamiento, también podemos usar las restricciones de acceso a ciertas url como vimos con anterioridad.

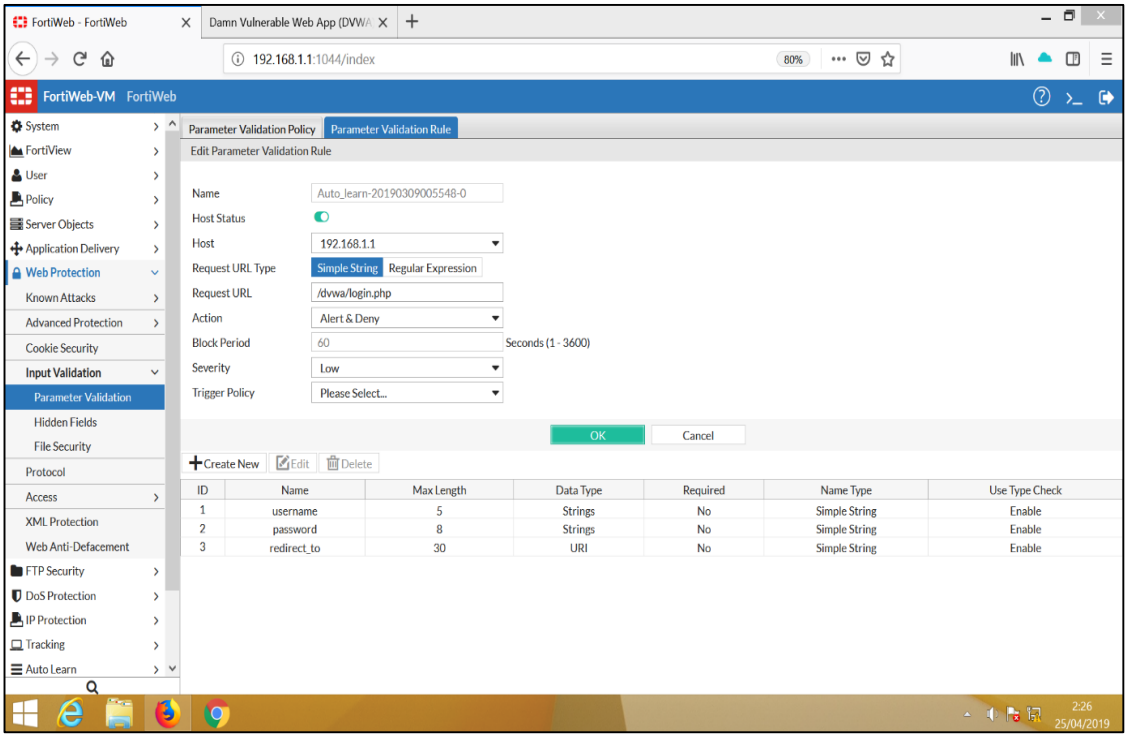

Figura 124. Reglas de validación

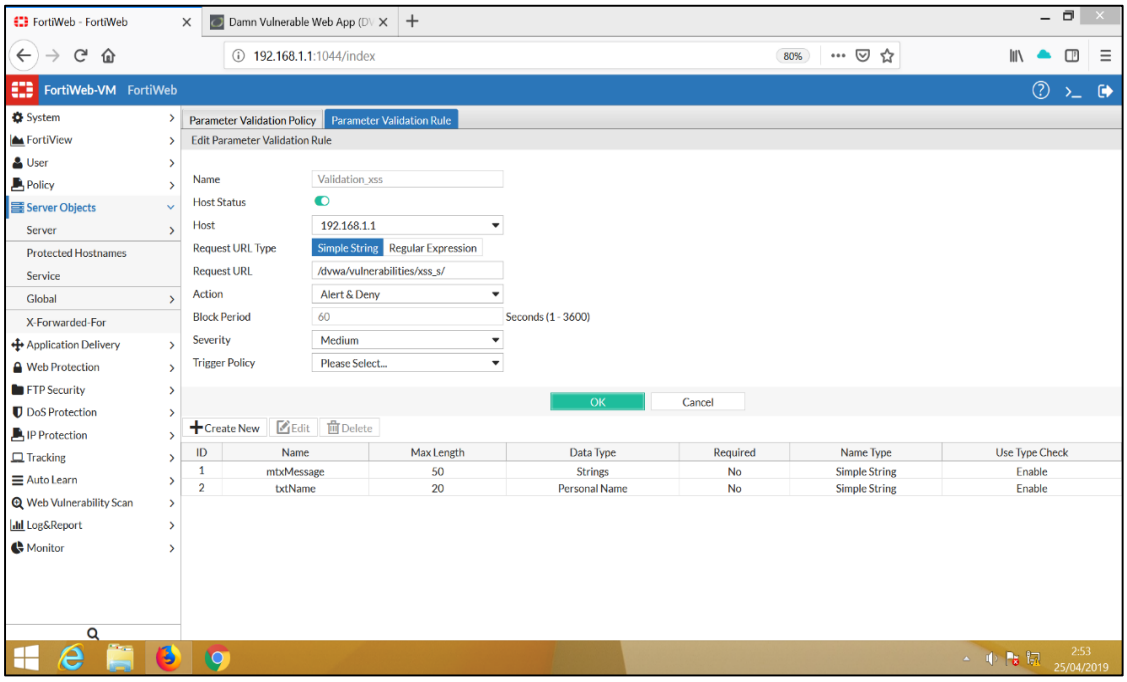

Figura 125. Reglas de validación

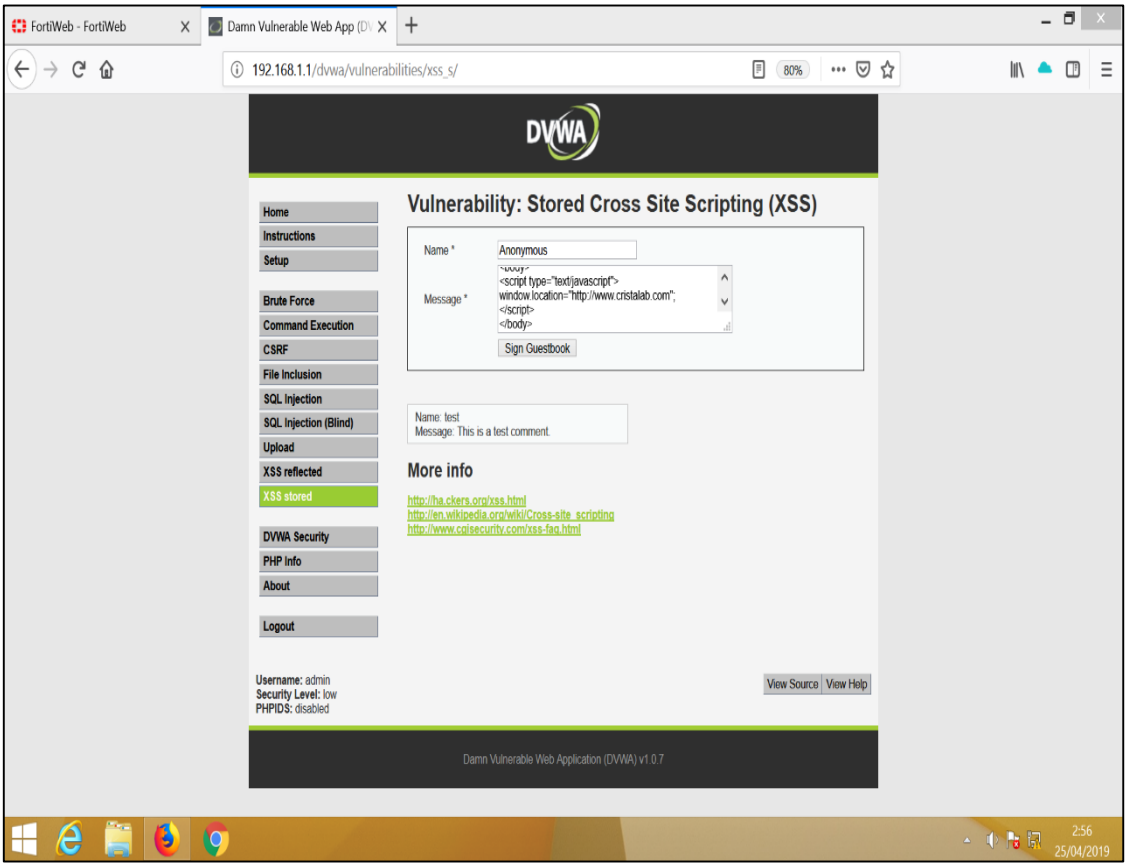

Figura 126. Ejecutando script malicioso

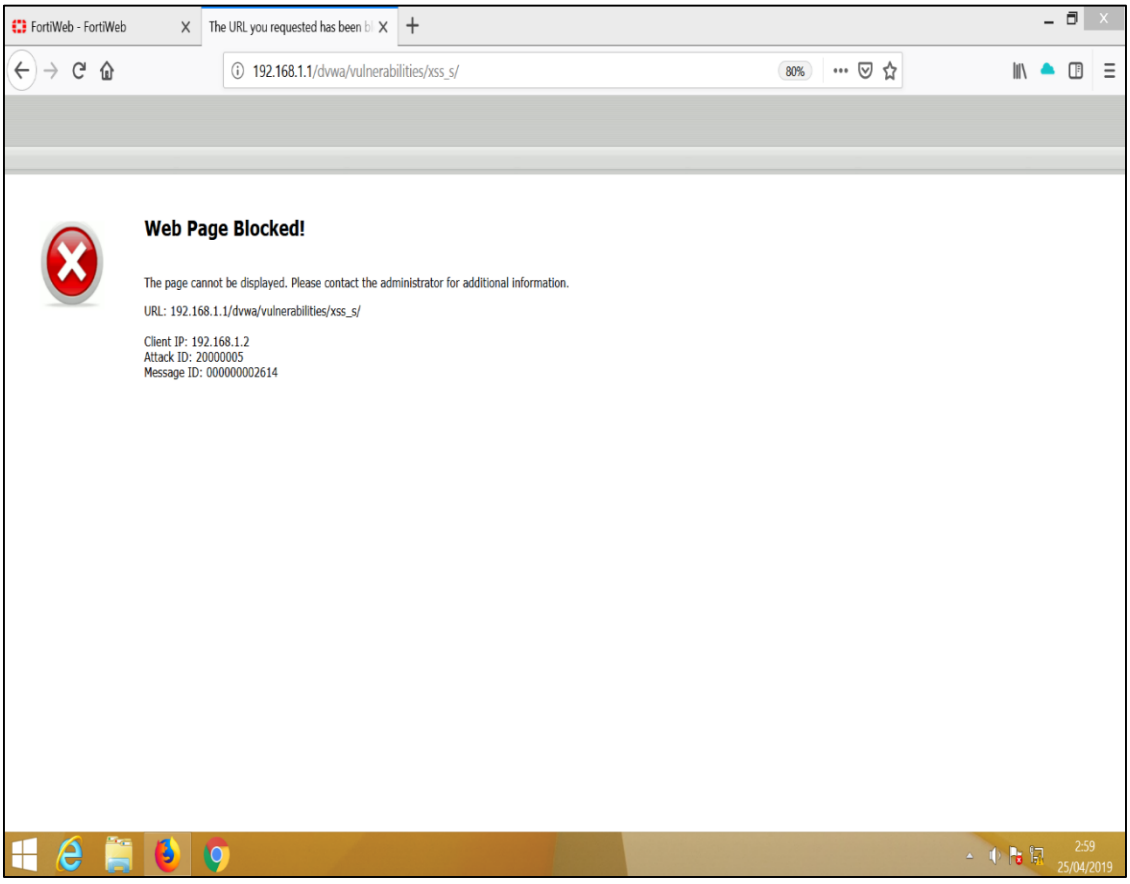

Figura 127. Bloqueo de FortiWeb

En los logs de Fortiweb podemos visualizar como se bloqueó primero por un tema de validación si el atacante lograba pasar la primera validación la firma contra ataques de **XSS** entraría en acción para mitigarlo.

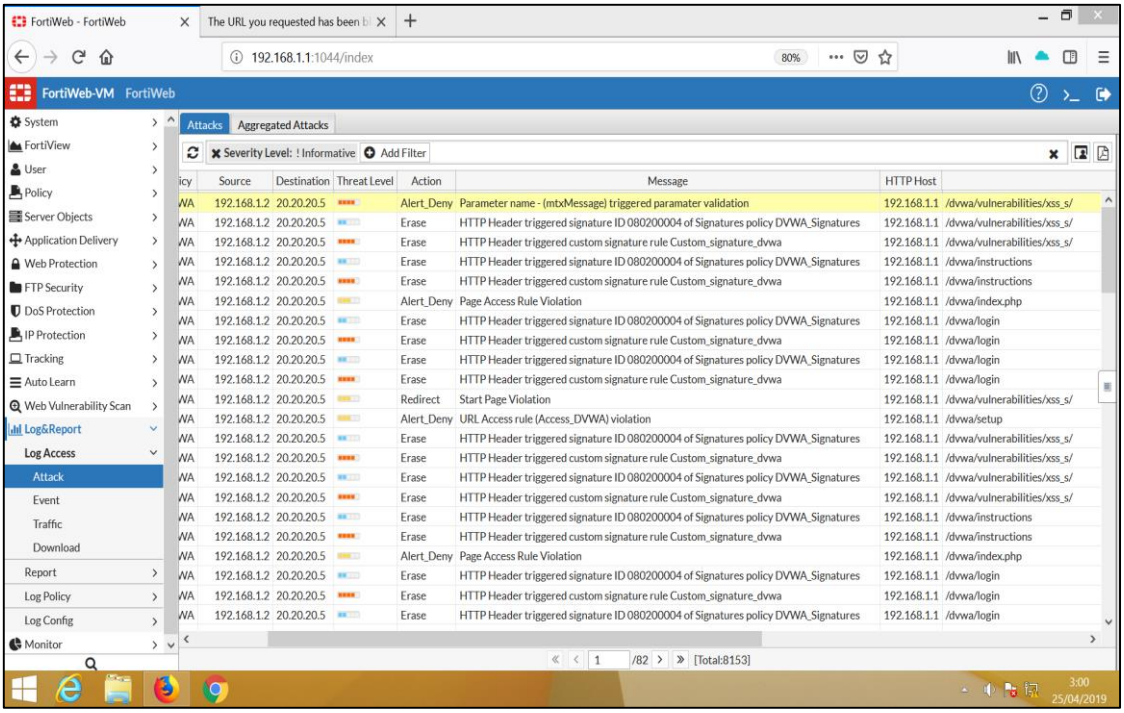

Figura 128. Logs generado por regla de validación

Fortiweb cuenta con diversos tipos de datos ya definido, pero revisaremos los URI que coincide con cualquier URI válido, no solo redirige al mismo sitio. Por lo tanto, aún permitiría redirecciones a otro sitio, potencialmente malicioso. Además, este tipo de datos también podría bloquear los redireccionamientos legítimos del mismo sitio.

Finalmente, esta expresión regular combina las URL como las escribirías en un navegador, no las que tienen codificación URL. (La codificación URL traduce algunos caracteres que no están permitidos en una URL, como espacios, en codificaciones de entidad, como **'% 20'**. De manera predeterminada, FortiWeb solo decodifica 1 capa de codificación URL).

Para bloquear los redireccionamientos a otro sitio potencialmente malicioso o para hacer coincidir las entradas con codificación URL que no se han decodificado por completo, personalizamos el tipo de datos y establecemos una configuración avanzada.

En este ejemplo vemos cómo se modifica la expresión para que coincida solo con el tipo de datos aceptable para la entrada de redireccionamiento, lo que se busca son coincidencias y solo permitir redirecciones a **example.com, www.example.com** o **ftp.example.com**.

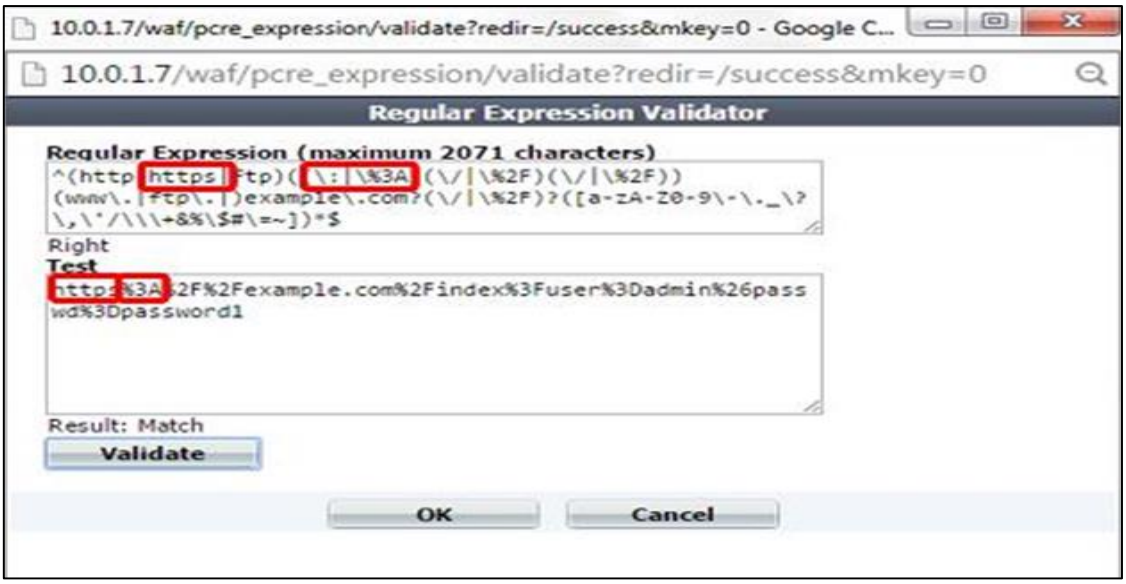

Figura 129. Regla de expresión para url del sistema

Para evitar que los atacantes evadan la detección codificando la URL dos o más veces, alternativamente, puede habilitar la opción para deshacerlo. Esta es una configuración global, por lo que puede afectar el rendimiento de escanear cada entrada, en cada política.

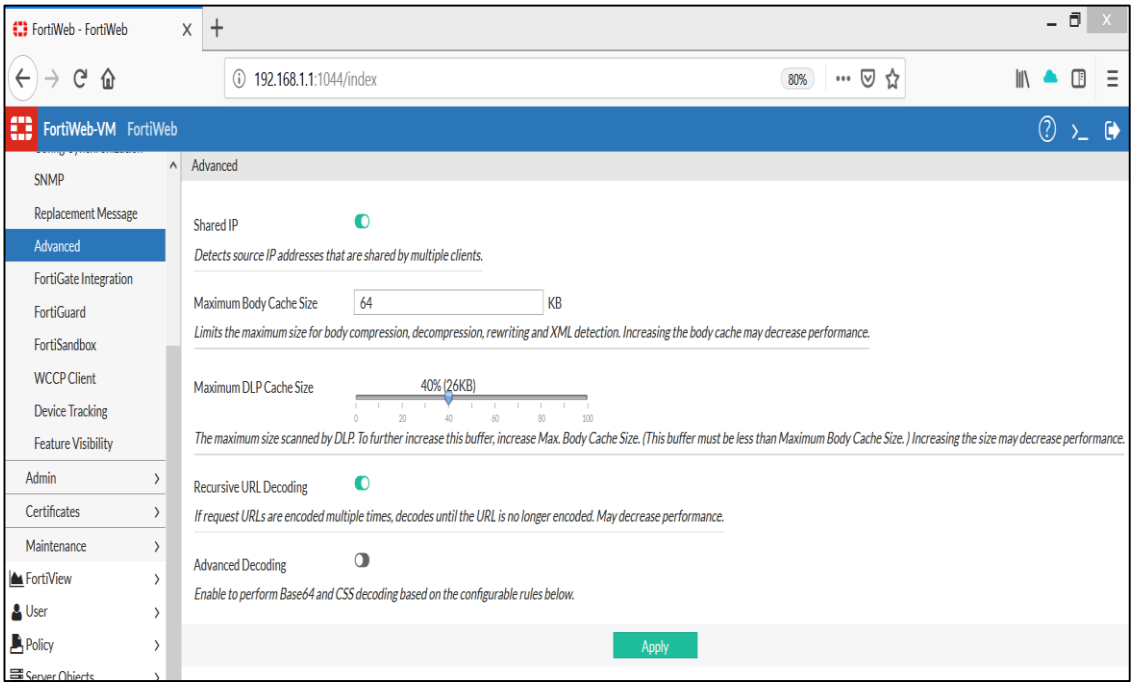

Figura 130. Activando la decodificación de url

17. Conclusiones

En el presente trabajo después de realizar diversos ataques maliciosos contra nuestro sistema web vulnerable, confirmamos como FortiWeb nos brindó seguridad de manera eficiente, además puede reducir de forma muy significativa los costes de despliegue e implementación de las políticas de seguridad a través de la consolidación de las funcionalidades de Firewall de Aplicaciones, Firewall XML, aceleración del tráfico web y balanceo del tráfico de las aplicaciones en un solo dispositivo, que además no se licencia por número de usuarios o servidores. Reduce por tanto drásticamente el tiempo necesario para realizar el análisis, diseño e implementación de una política de protección completa y eficaz de los recursos y aplicaciones web de cualquier organización o empresa.

Mediante el uso de avanzadas técnicas para ofrecer protección bidireccional contra ataques complejos y sofisticados como por ejemplo “SQL Injection” y “Cross-site scripting”, las plataformas FortiWeb ayudan a prevenir robos de identidad, fraudes económicos y espionaje corporativo. Los dispositivos FortiWeb proporcionan la tecnología imprescindible para monitorizar y aplicar las distintas normativas regulatorias, tanto gubernamentales como de iniciativa privada, asegurar la implantación de las prácticas de seguridad recomendadas y las adecuadas políticas internas.

Ya sea que se trate de sencillamente cumplir con los estándares o de proteger las aplicaciones críticas, los Web Application Firewall de FortiWeb proporcionan características avanzadas para defender las aplicaciones web de amenazas conocidas y de día cero. Al utilizar un enfoque multicapa avanzado y correlacionado, FortiWeb proporciona seguridad completa para sus aplicaciones externas e internas basadas en la web frente a las amenazas del OWASP Top 10 y muchas otras. En el núcleo de FortiWeb se encuentran sus motores de detección basados en auto aprendizaje de doble capa que detectan amenazas de manera inteligente sin casi ninguna detección de falsos positivos.

18. Bibliografía

18.1 Específica

[1] Fortiweb 5.6 material de estudio de Certificación, 2018.

[2] Fortiweb 6.1 material de estudio de Certificación, 2019.

[3] Enrique Rando, Amador Aparicio, Pablo Gonzáles, Ricardo Martín, Chema Alonso, Hacking web Technologies, 2016.

[4] Ron Lepofsky, The manager's guide to web application security, 2014.

[5] Joel Scambray, Web application security secrets and solutions third edition, 2010.

[6] Chris Mcnab, Network security assessment, 2016.

[7] Paco Hope, Ben Walther, Web security testing cookbook, 2008.

[8] Victor Marak, Windows malware analysis essentials, 2015.

[9] Monika Agarwal, Abhinav Singh, Metasploit penetration testing cookbook second edition, 2013.

[10] Monika Agarwal, Abhinav Singh, Metasploit penetration testing cookbook second edition, 2013.

[11] Cisco ccna security versión 2.0, 2018.

[12] Andrew Lockhart, Network security hacks second edition, 2006.

[13] Patrick Engebretson, The basics of hacking and penetration testing, 2011.

[14] William Stallings, Cryptography and network security fourth edition, 2005.

[15] Evan Gilman, Doug Barth, Zero trust networks first edition, 2017.

[16] Teri Bidwell, Michael Cross, Ryan Russell, Hack proofing your identity in the information age, 2002.

[17] Jeremiah Grossman, Robert Hansen, Anton Rager, Seth Fogie, D. Petkov, XSS Attacks, 2007.

[18] Stephen Lax, Access denied in the information age, 2001.

[19] Craig A. Schiller, Jim Binkley, David Harley, Gadi Evron, Tony Bradley, Tony Bradley, Carsten Willems, Michael Cross, Botnet the killer web app, 2007.

[20] Dennis Hansen, Social Vulnerability and Assessment Framework, 2017.

[21] Jay Kreps, I heart logs, 2014.