Pruebas de Penetración contra Aplicaciones Web

Alonso Eduardo Caballero Quezada

Consultor en Hacking Ético e Informática Forense

e-mail: ReYDeS@gmail.com

Sitio Web: www.ReYDeS.com

¿Quién Soy?

Alonso Eduardo Caballero Quezada es Brainbench Certified Network Security (Master), Computer Forensics (U.S.) & Linux Administration (General), IT Masters Certificate of Achievement en Network Security Administrator, Hacking Countermeasures, Cisco CCNA Security, Information Security Incident Handling y Miembro de OWASP.

Ha sido Instructor en el OWASP LATAM Tour Lima, Perú del año 2014, y Conferencista en PERUHACK 2014. Cuenta con más de doce años de experiencia en el área y desde hace ocho años labora como Consultor e Instructor Independiente en las áreas de Hacking Ético & Informática Forense. Perteneció por muchos años al grupo internacional de Seguridad RareGaZz e integra actualmente el Grupo Peruano de Seguridad PeruSEC. Ha dictado cursos en Perú y Ecuador, presentándose también constantemente en exposiciones enfocadas a, Hacking Ético, Informática Forense, GNU/Linux y Software Libre.

@Alonso_ReYDeS www.facebook.com/alonsoreydes

pe.linkedin.com/in/alonsocaballeroquezada/

Temario

- Aplicaciones Web
- Seguridad en Aplicaciones Web
- Pruebas de Penetración a Aplicaciones Web
- Tipos de Pruebas de Penetración
- Métodos de Prueba
- Componentes de una Prueba de Penetración
- Metodología de una Prueba de Penetración a una Aplicación Web
- Reconocimiento
- Mapeo
- Descubrimiento
- Explotación
- Tipos de Fallas

Aplicaciones Web

En la actualidad las Aplicaciones Web son los objetivos más buscados para explotarlos, esto debido a su amplia utilización.

En la actualidad debido al crecimiento y difusión de Internet, la web y por consiguiente las aplicaciones web se integran en la vida diaria de muchas personas. Uno de los principales beneficios de las aplicaciones web es su portabilidad, y su adecuada funcionalidad en un amplio espectro de sistemas operativos.

Las aplicaciones web pueden ser utilizadas en una amplia diversidad de escenarios, como almacenar, gestionar y acceder a datos financieros sensibles o información personal. Los clientes pueden acceder a sus cuentas bancarias. Las empresas pueden utilizarlas para compartir propiedad intelectual, etc.

El alto valor de los datos accedidos mediante la utilización de aplicaciones web incrementa su valor como objetivo de ataque.

Seguridad en Aplicaciones Web

En la actualidad una gran cantidad de organizaciones se centra en realizar pruebas sobre la funcionalidad de las aplicaciones desarrolladas para la web del negocio, pero raramente realizan pruebas de seguridad.

Diariamente se reportan un gran cantidad de vulnerabilidades relacionadas y enfocadas en las aplicaciones web.

Con la creciente utilización de nuevas tecnologías, los sitios web hacen más de lo “perceptible”, lo cual añade nuevos vectores de ataque.

Típicamente el usuario es requerido de hacer clic en un enlace o imagen de un sitio web, la petición va hacia el servidor, y luego se devuelve un resultado conteniendo la página web. Para el caso de Ajax por ejemplo, se interactua con el sitio, se ejecuta código JavaScript para realizar llamadas y recabar datos desde el servidor. De esta manera las páginas son actualizadas dinámicamente con los datos recibidos.

* Open Sourced Vulnerability Database: http://osvdb.org/
* XSS Attack Information: http://xssed.com/

Pruebas de Penetración a Aplicaciones Web

Para realizar pruebas de penetración satisfactorias contra aplicaciones web se necesita tener un buen conocimiento; más allá del nivel de usuario normal; sobre las tecnologías web. Entender como es su funcionamiento desde la perspectiva del desarrollador o administrador web.

De esta manera los profesionales en pruebas de penetración deben pensar de manera maliciosa pero actuando profesionalmente. Deben preguntarse como sería factible pasar las restricciones, analizar cuales podrían ser los errores cometidos por los desarrolladores, administradores, y operadores del sistema objetivo.

Esta es una perspectiva de pensamiento completamente diferente a la de los desarrolladores o administradores. Lo cual permite enfocarse en pasar los controles de la aplicaciones, o encontrar problemas en lógica del negocio.

* OWASP: https://www.owasp.org/

Pruebas de Penetración a App Web (Cont.)

Una buena Prueba de Penetración debe seguir una metodología aprobada. Sin esto se perderían vulnerabilidades y no se completaría el trabajo. Esta metodología deben ser Aprobada, Repetible y Explicable.

Además del uso de una metodología, conocer las herramientas es fundamental. No es necesario recordar mil opciones de las herramientas, es suficiente estar familiarizado con las herramientas existentes para realizar esta labor.

El obtener permiso para realizar las pruebas es la pieza más crítica de información, debido a la existencia de legislaciones sobre hacking. De no tener el permiso para evaluar la seguridad de una aplicación por escrito de alguien con la autoridad, entonces no se evalúa la aplicación.

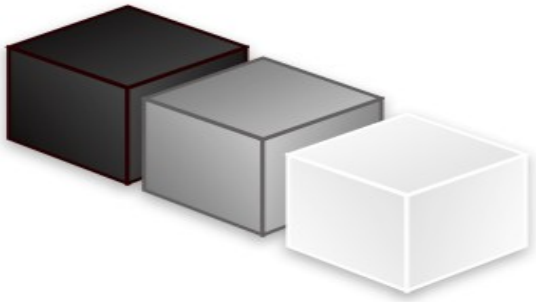
Ejemplo: “Accidentalmente” se descubrió una falla de inyección “SQL” en el sitio web de una página del gobierno.

Tipos de Pruebas de Penetración

Caja Negra: Se proporciona poca o ninguna información sobre el objetivo de evaluación por adelantado. Es más frecuente en pruebas de penetración a redes.

Caja Cristal: Son realizadas generalmente por un equipo interno, pero ahora es más frecuente asignarlo a un equipo externo. El equipo de prueba usualmente es parte del equipo de QA, y como tal se convierte en una parte del ciclo de vida para el desarrollo del software. Se tiene acceso al código fuente para revisarlo y reportar las vulnerabilidades encontradas.

Caja Gris: Es el tipo de prueba más común. Requiere realizar más trabajo para obtener información necesaria. Es crítica la comunicación entre el equipo de pruebas y la empresa en evaluación.



Métodos de Prueba

Manual: Es la primera manera utilizada para comprender un ataque. Aunque es manual se utilizan scripts sencillos y herramientas. Es relativamente lento comparándolo con otros métodos.

Automático: Las herramientas pueden escanear rápidamente un sitio y devolver las vulnerabilidades encontradas. El inconveniente es tener menor control sobre el comportamiento del ataque, por lo cual se es propenso a obtener falsos positivos.

Híbrido: Genera mejores resultados el utilizar métodos manuales y automáticos. Se puede utilizar un escaner de vulnerabilidades para tener un línea base y punto de inicio. Y al mismo tiempo se puede realizar revisiones manuales por problemas en el sitio. Los resultados del escaner deben ser validados, con la intención de utilizarlos luego para expandir el punto de apoyo dentro de la aplicación.

Componentes de una Prueba de Penetración

Existen tres secciones durante una prueba de penetración, todas ellas son obligatorias excepto la presentación.

Preparación

Establecer el Alcance de la Prueba: Propósito, Tipo y Alcance de la Prueba

Obtener Información requerida para la Prueba: Aplicaciones, Nombres de usuario y Contraseñas, Restricciones Tecnológicas, Información Contacto.

Reglas del Contrato: Ambas partes deben estar de Acuerdo.

Identificar Tráfico del “Hacker Ético” y Datos en la Aplicación

Ventana de Tiempo para la Prueba: Acordar Duración y Horarios.

Planificar las Comunicaciones: Contactos diversos y protegidos para la comunicación,

Componentes de una PdP (Cont.)

Reportar

Es probablemente la pieza más importante de la prueba de penetración. Mucha compañías tomarán el reporte y la utilizarán como una directriz para conocer aquello requerido de ser solucionado y asegurar sus aplicaciones.

1. Resumen Ejecutivo: Alto Nivel, 1 página. Hallazgos y Recomendaciones
2. Introducción: Explicar Alcance, Objetivo, Lista del Equipo, 1 a 2 páginas.
3. Metodología: Describirla paso a paso. De 3 a 10 páginas promedio.
4. Hallazgos: Lista categorizada en base al Riesgo. Recomendaciones.
5. Conclusiones: Sección corta, similar al resumen ejecutivo.

También se pueden incluir apéndices, anotaciones importantes, documentos, y otros ítemes obtenidos o creados durante la prueba de penetración.

* OWASP Testing Guide - Reporting: https://www.owasp.org/index.php/Reporting
* Penetration Testing Report: http://www.offensive-security.com/offsec/penetration-test-report-2013/

Componentes de una PdP (Cont.)

Presentación

Es una parte opcional de la prueba, la cual algunas organizaciones obvian. La presentación es básicamente un conjunto de diapositivas describiendo lo evaluado, lo encontrado y las recomendaciones para solucionar los hallazgos.

Aunque no se tenga control sobre quienes asistirán a la presentación, pues esto es definido por el personal del objetivo, se sugiere la asistencia de personal de desarrollo, administración o técnico, gestión y personal de pruebas. Tener en consideración el requerimiento de enfoques diferentes para cada tipo de audiencia.

La presentación no debe incluir información sensible, ejemplo, URLs vulnerables a XSS, SQLi, CSRF, etc. Se debe dar una visión global de los hallazgos y su funcionamiento general. Luego trabajar con el cliente para asegurarse del conocimiento y comprensión por parte de aquellos quienes necesiten entender como ocurre exactamente la explotación.

Metodología de una PdP a una Aplicación Web

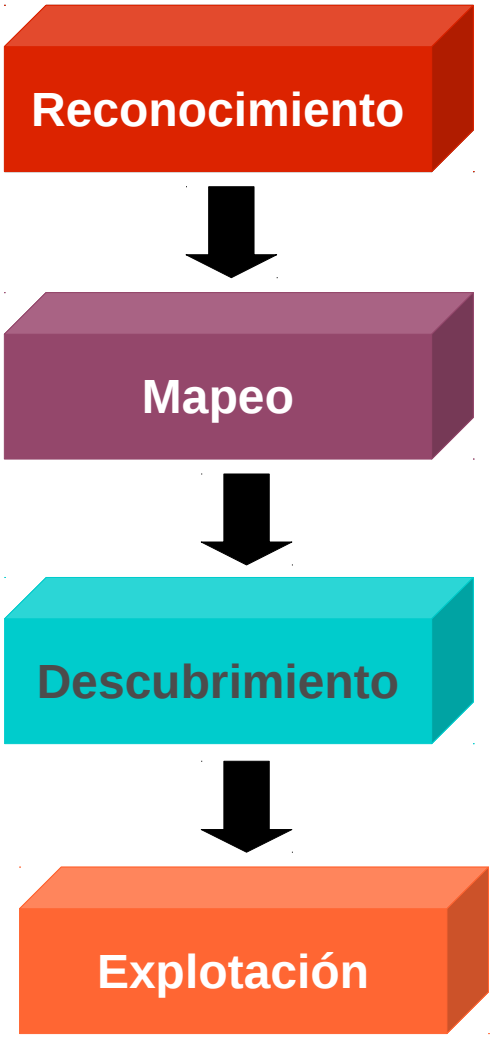
Cada fase se basa en los resultados de las fases previas. El proceso también es cíclico.

Reconocimiento: Proporciona los fundamentos para un ataque satisfactorio y eficiente. Se trata de identificar el objetivo mediante diversos recursos.

Mapeo: Implica entender como funciona la aplicación y su infraestructura subyacente. Identificar las diversas piezas de la aplicación y sus relaciones.

Descubrimiento: Se inicia la exploración más profunda de la aplicación, y encuentran probables vulnerabilidades e información para el ataque.

Explotación: Se toma toda la información obtenida hasta este punto, y se lo utiliza para explotar la aplicación. Aquí se lanzan los ataques.



Reconocimiento

Este es el primer paso de todo el proceso. Muchos atacantes nóveles lo obvian. El reconocimiento proporciona bases sólidas para un ataque eficiente y satisfactorio. Muchos atacantes lo obvian interpretándolo como una perdida de tiempo. Esto puede ser satisfactorio, pero resulta ineficiente pudiendo alertar al objetivo de evaluación. Al invertir tiempo para encontrar tanto como sea posible sobre el objetivo antes de lanzar los ataques, se tendrá un mejor enfoque para los esfuerzos y un menor riesgo de detección.

Labores típicas en el Reconocimiento

Registros Whois apuntando hacia Servidores de Nombres
Identificación de máquina involucradas en la aplicación (Direcciones IP, Nombres de hosts)
Transferencias de Zona conteniendo información detallada sobre nombres de host.
Buscar en fuentes de información externa como Google u otros motores de búsqueda, Redes Sociales y Listas de Correo, Blogs y diversos sitios webs.

Mapeo

Si no se realiza bien, los pasos posteriores serán difíciles o imposibles.

Se realiza la identificación y enumeración del S.O., Servicios y Aplicación.

El mapeo de la aplicación permite al atacante ver las diversas piezas de la aplicación web en un solo lugar. Ejemplo; páginas de catálogos, de gestión del carro de compras, pagos y entrega, gestión de usuario, etc.

Durante esta fase se intenta entender como encajan todas las piezas de la aplicación, e identificar los lugares donde es factible aprovechar las relaciones entre páginas para un mayor acceso. ¿Cual es el flujo lógico del usuario a través de la aplicación?. Pensar cómo evadir o evitar esta lógica. También se colectan identificadores de sesión, así se tiene un mejor entendimiento sobre como se mantiene el estado de la sesión.

Existen tres pasos principales. Primero se descarga el sitio completo mediante un spidering. Luego se utilizan estos resultados para descifrar como encajan las cosas y entender el flujo de la aplicación. Mientras tanto se capturan IDs y Tokens de sesión. Útil para identificar vulnerabilidades.

Descubrimiento

La fase de descubrimiento es el tercer paso en la metodología. Y el primer paso implicando tráfico potencialmente peligroso. Aquí es donde se empieza a explorar la aplicación y encontrar puntos vulnerables, pero sin explotarlos. Aunque podría ocurrir debido a la naturaleza de la falla.

El descubrimiento en donde realmente se empieza a explorar la aplicación de una manera profunda. Buscando potenciales vulnerabilidades e información útil para planificar el ataque en la siguiente fase.

Labores típicas en el Reconocimiento

Determinar las debilidades, como mensajes de error o problemas en la aplicación.
Cada falla encontrada ayudará a profundizar el mapa de la aplicación, como contenidos perdidos anteriormente en la fase de mapeo.
Recolectar nombres de usuario, capturar todos los IDs posibles de las páginas en una aplicación.
Determinar las herramientas de explotación a utilizarse.

Explotación

En este paso final el atacante toma toda la información capturada hasta este punto y la utiliza para explotar la aplicación. Esto puede implicar realizar un volcado de la base de datos, o hacer un “pivoting” para atacar el resto de la red.

La explotación es donde se lanzan los ataques. Muchos profesionales se enfocan en esta etapa en detrimento de las pruebas, pues la explotación se construye sobre toda la información recolectada y las tareas completadas hasta este punto. Sin los primeros tres pasos de la metodología descrita la explotación generalmente falla.

Es un Ataque Cíclico

Cuando se explota satisfactoriamente una vulnerabilidad usualmente se abren nuevos caminos, por lo tanto se inicia el proceso nuevamente, aprovechándose del nuevo acceso o información. A diferencia del atacante malicioso, el trabajo de un profesional es encontrar todas las vulnerabilidades posibles, no solamente una.

Tipos de Fallas

Fallas de Fuga de Información: Permite al atacante descubrir información sobre la configuración y el estado de la aplicación. La cual puede ser utilizada para lanzar ataques más dañinos.

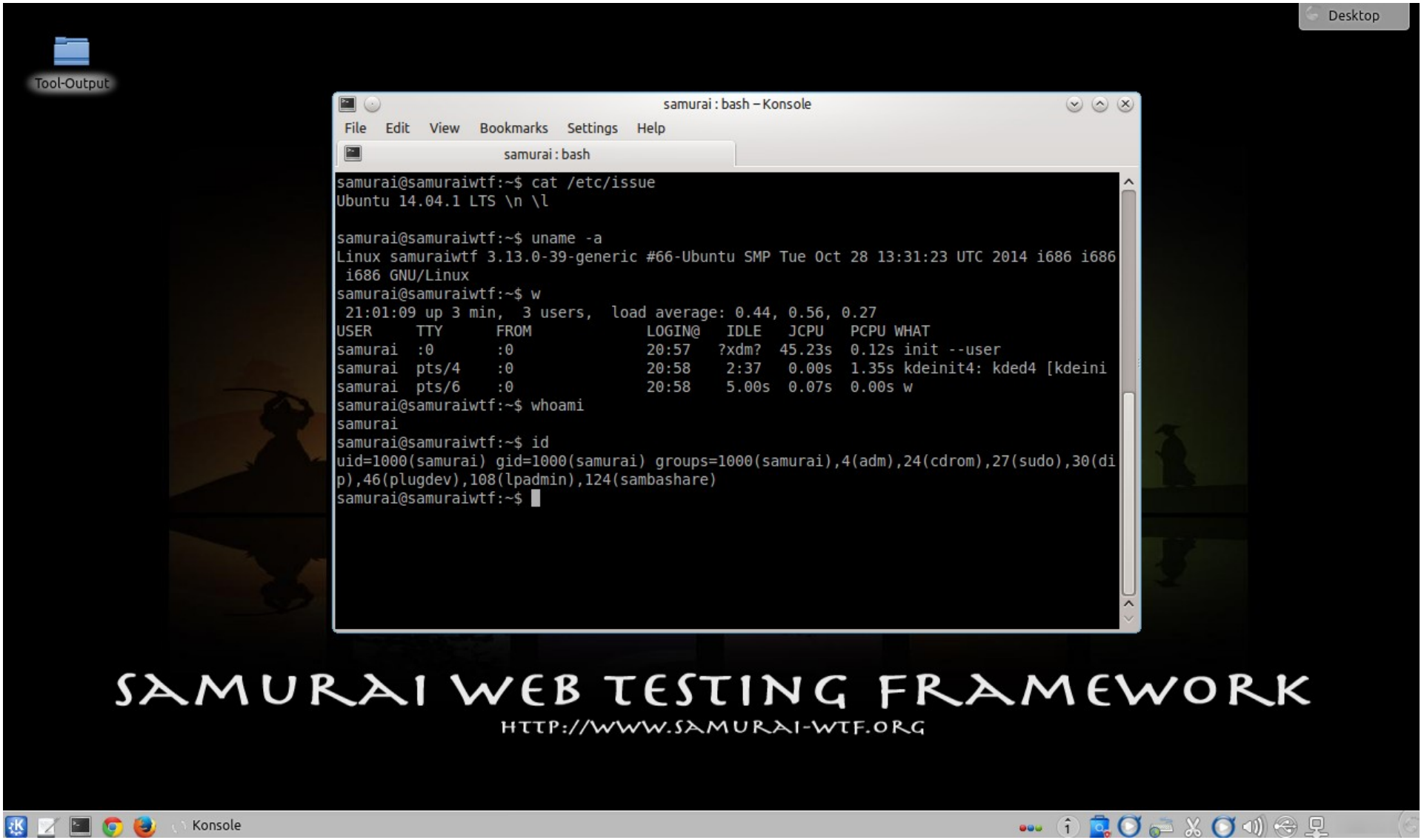
Fallas de Configuración: Temas de configuración con el servidor web o el sistema operativo puede permitir al atacante explotar el sistema sin importar cuan segura sea el diseño de la aplicación web.

Fallas de Bypass: Permite al atacante esquivar controles. Puede permitir al atacante ganar más acceso dentro de la aplicación o permitirle acceder y modificar archivos sobre el servidor.

Fallas de Inyección: Son muy comunes como los bien conocidos ataques de Inyección SQL y Cross Site Scripting (XSS). Implica demasiada confianza en el cliente, y aceptar entradas con filtros inadecuados.

Demostraciones

.



Más Contenidos

Videos de 24 Webinars Gratuitos sobre Hacking Ético, Hacking Aplicaciones Web e Informática Forense.

http://www.reydes.com/d/?q=videos

Diapositivas utilizadas en los Webinars Gratuitos.

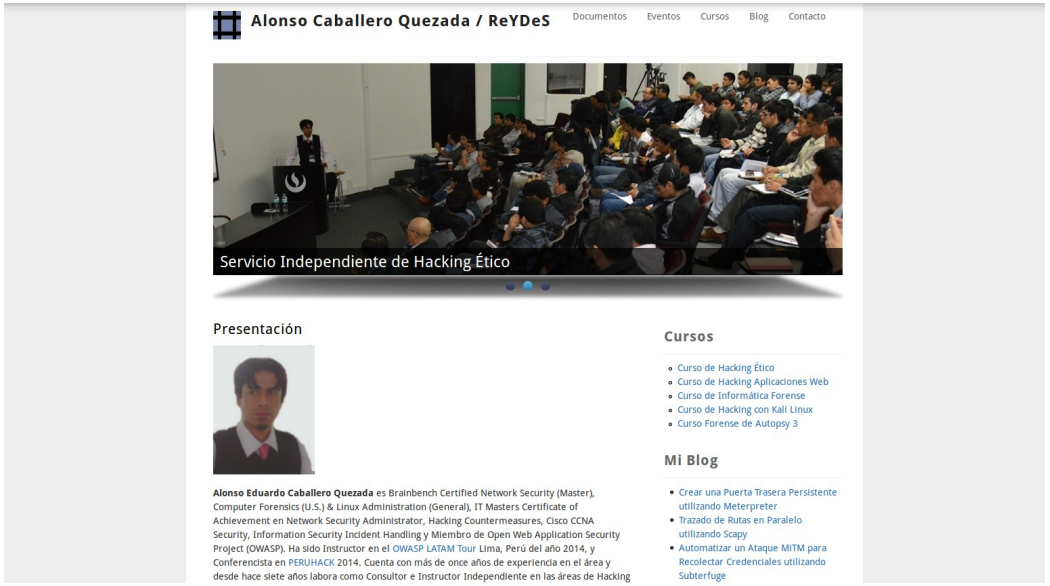
http://www.reydes.com/d/?q=node/3

Artículos y documentos publicados

http://www.reydes.com/d/?q=node/2

Blog sobre temas de mi interés.

http://www.reydes.com/d/?q=blog/1



¡Muchas Gracias!

Alonso Eduardo Caballero Quezada

Consultor en Hacking Ético e Informática Forense

e-mail: ReYDeS@gmail.com

Sitio Web: www.ReYDeS.com