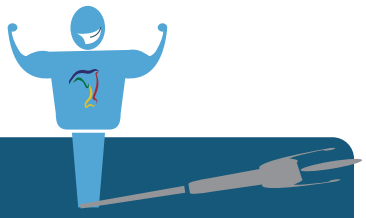
autentia

Avenida de Castilla,1 - Edificio Best Point - Oficina 21B
28830 San Fernando de Henares (Madrid)
tel./fax: +34 91 675 33 06
info@autentia.com - **www.autentia.com**

¿**Qué ofrece** Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
Ese apoyo que siempre quiso tener...

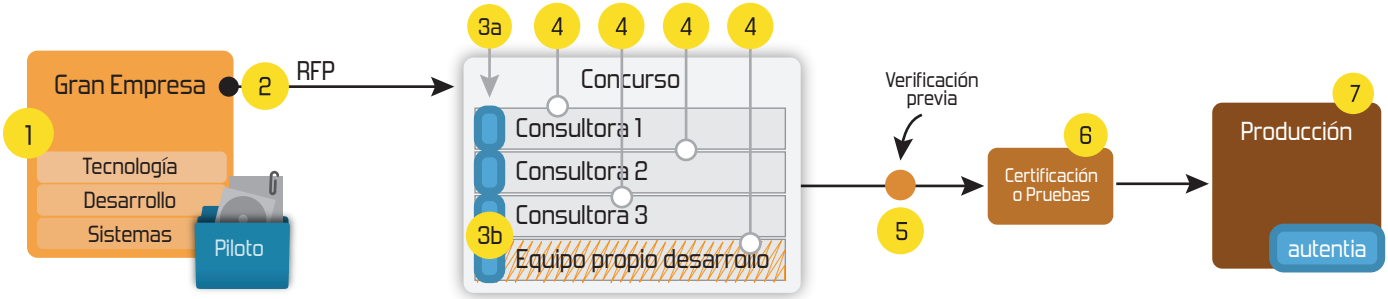
1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

- Spring MVC, JSF-PrimeFaces /RichFaces, HTML5, CSS3, JavaScript-jQuery
- Gestor portales (Liferay)
- Gestor de contenidos (Alfresco)
- Aplicaciones híbridas
- Control de autenticación y acceso (Spring Security)
- UDDI
- Web Services
- Rest Services
- Social SSO
- SSO (Cas)
- JPA-Hibernate, MyBatis
- Motor de búsqueda empresarial (Solr)
- ETL (Talend)
- Dirección de Proyectos Informáticos.
- Metodologías ágiles
- Patrones de diseño
- TDD
- Tareas programadas (Quartz)
- Gestor documental (Alfresco)
- Inversión de control (Spring)
- BPM (jBPM o Bonita)
- Generación de informes (JasperReport)
- ESB (Open ESB)

Compartimos nuestro conociemiento en:
www.adictosaltrabajo.com

Para más información visítenos en:
www.autentia.com

DESARROLLADO POR:
Jaime Carmona Loeches

Ingeniero Informático Superior por la UAM

Fecha de publicación del tutorial: 2009-02-26

JCAPTCHA: ANÁLISIS TÉCNICO EN APLICATIVOS REALES

Jaime Carmona Loeches

Introducción

Este tutorial es continuación del anterior tutorial publicado donde se explicaron los principios básicos de JCaptcha y una configuración inicial en el framework Struts.

En este segundo tutorial, que pretende ser conciso y directo, vamos a ver en qué casos es conveniente utilizar JCaptcha dentro de un aplicativo web y en cuáles puede suponer un coste innecesario, teniendo en cuenta una gráfica bidireccional que valore y estime los parámetros: coste de desarrollo, accesibilidad, y seguridad informática.

Desarrollo

Posibles consecuencias de un ataque automático

Lo primero que debemos tener en cuenta es que JCaptcha es un filtro de ataques automáticos masivos a un ordenador.

La pregunta lógica es, ¿qué consecuencias tendrían estos ataques?

Por orden de gravedad estimada por el autor, las consecuencias pueden ser las siguientes:

-Gravedad media: Saturación del servidor e indisponibilidad por un período de tiempo. Costes indirectos son una bajada de la imagen del aplicativo, pérdida de usuarios, pérdida de credibilidad...

-Gravedad alta: Acceso automático y malicioso a herramientas propias del aplicativo, como Bases de Datos, y posibilidad de perder datos, o bien tener datos erróneos. Si el aplicativo no tiene una política de backup, puede sufrir pérdidas de datos difíciles de recuperar.

Cada aplicativo persigue su propia finalidad, por lo que siempre es bueno medir las consecuencias del error en función del aplicativo concreto.

Análisis en aplicativos reales

Para ello, imaginemos un escenario y dos aplicativos web, dentro de una empresa, para analizar la implementación de JCaptcha en pantallas específicas, con el fin de prevenir ataques externos.

SISTEMA A:

Sistema orientado a la Intranet de una empresa, con un número medio de usuarios.

Las pantallas relativas a formularios, donde el usuario puede lanzar datos y peticiones al servidor, son las siguientes:

-LOGIN: El usuario puede loguearse a la aplicación, que está visible para todos los trabajadores de la empresa.

Imaginemos que la empresa tiene trabajadores internos, dentro de los cuales para un grupo de ellos está destinado el aplicativo, y trabajadores externos.

A esta pantalla pueden acceder un número de usuarios bastante elevado que no tienen relación con el aplicativo.

-REGISTRO: El usuario puede enviar datos para registrarse. Después de una primera pantalla, se envía un email al usuario, desde el cual puede acceder a una segunda pantalla de registro.

-PANTALLA PRINCIPAL (requiere loguearse): Una vez logueado dentro del sistema, existen diferentes pantallas con formularios de operaciones de lectura y escritura.

Notas a tener en cuenta:

-Dentro de esta aplicación, que no es visible a priori desde fuera de la empresa, existe un número de trabajadores que pueden hacer un uso malicioso de la aplicación, intentando saturarla, así como los propios usuarios de la misma por razones que pueden ser desconocidas inicialmente.

-En la pantalla de login inicial, el acceso no está controlado, por lo que podríamos sufrir un ataque masivo con facilidad.

Las consecuencias de este ataque estarían relacionadas con:

1) El alto número de peticiones simultáneas (y saturación del servidor)

2) Operaciones de lectura contra la Base de Datos o LDAP interno, relacionados con el logueo del aplicativo

-En la pantalla inicial de registro, el caso es similar. Sin embargo, en la segunda pantalla, el usuario ha especificado un email, por lo que configurar un ataque automático en estas condiciones tiene un nivel mayor de dificultad.

Además, en esta pantalla hay un acceso de escritura contra base de datos, por lo que las consecuencias podrían ser más peligrosas.

-Una vez que el usuario entra dentro del aplicativo, el sistema ya tiene constancia del usuario que se ha introducido, por lo que a priori sería poco adecuado lanzar un ataque automático (por decirlo de otra manera, no sería muy inteligente por el atacante).

Después de este primer análisis, las decisiones que se toman son las siguientes:

-Introducir JCaptcha en la pantalla de Login y la pantalla inicial de Registro.

-Ignorar la introducción de JCaptcha en la segunda pantalla de Registro y en la pantalla Principal.

SISTEMA B:

Sistema orientado publicado en la WWW, accesible para toda persona con conexión a Internet.

Catálogo de servicios Autentia

Últimas Noticias

- XIII Charla Autentia - AOS y TDD
- XII Charla Autentia - LiquiBase - Material
- XI Charla Autentia - Mule - Vídeos y Material
- Reflexiones sobre AOS2010
- Comentando el libro: Nunca comas solo de Keith Ferrazzi y Tahl Raz.

Histórico de NOTICIAS

Últimos Tutoriales

- Zen-coding: una nueva forma de escribir código HTML
- JBoss autenticación basada en certificados cliente
- Rdiff-backup: Herramienta para realizar backups
- Cómo crear un efecto reflejo con Adobe Photoshop
- Trabajando con los Web Services de Liferay

Últimos Tutoriales del Autor

- Configuración de jCaptcha en Struts
- Manual de Javascript

Síguenos a través de:

Últimas ofertas de empleo

2010-10-11
Comercial - Ventas - SEVILLA.

2010-08-30
Otras - Electricidad - BARCELONA.

2010-08-24
Otras Sin catalogar - LUGO.

2010-06-25
T. Información - Analista / Programador - BARCELONA.

Las pantallas específicas son las siguientes:

-LOGIN: El usuario puede loguearse a la aplicación, esta pantalla es visible para cualquier persona con conexión a Internet

-REGISTRO: El usuario puede enviar datos para registrarse. Esta pantalla es visible para cualquier persona con conexión a Internet y, a diferencia del aplicativo anterior, consta de una sola pantalla.

-PANTALLA PRINCIPAL (requiere loguearse): Una vez logueado dentro del sistema, existen diferentes pantallas con formularios de operaciones de lectura y escritura.

Notas a tener en cuenta:

Dentro de esta aplicación, que es visible a un número mayor de usuarios, con la dificultad de rastrear los ataques en internet, el nivel de seguridad a priori debe ser mayor. Para ello, se toman las siguientes deciones:

-En la pantalla de login inicial, el acceso no está controlado, por lo que podríamos sufrir un ataque masivo con facilidad.

-En la pantalla de registro, el caso es similar. Además, en este caso la operación contra la base de datos no es sólo de lectura, si no que es de escritura, por lo que los daños del ataque pueden ser de mayor consideración.

-Una vez que el usuario entra dentro del aplicativo, el sistema ya tiene constancia del usuario que se ha introducido, por lo que a priori sería poco adecuado lanzar un ataque automático (por decirlo de otra manera, no sería muy inteligente por el atacante, o bien requeriría un nivel de habilidad mucho más avanzado).

Conclusiones

JCaptcha nos ayuda a dificultar los ataques automáticos a nuestros aplicativos web, donde es conveniente analizar su uso teniendo factores como los siguientes:

-Usuarios a los que son accesibles a la web.

-Número potencial de los mismos.

-Consecuencias de los ataques, como saturación del servidor y posible corrupción de datos.

-Incremento de la dificultad de la accesibilidad del aplicativo, así como el tiempo de desarrollo.

Comentarios