

# Seguridad y Auditoría Informática

Auditoría de Aplicaciones Web

# Realidad Actual

- Las Aplicaciones Web vulnerables son consistentemente uno de los 10 problemas más importantes
  - Las pruebas pueden ser peligrosas
    - Causando problemas inesperados en la aplicación o infraestructura.
  - La tecnología es aún "nueva"
    - HTTP/HTML es maduro
    - Nuevos avances y técnicas continúan evolucionando para apalancar estas tecnologías
- Las aplicaciones externas típicamente comienzan su vida como "hacks" internos

# Funcionamiento Básico y Entrenamiento

- ¿Donde aprenden su oficio los desarrolladores de aplicaciones web?
  - ¿Universidad?
  - ¿Seminarios de Entrenamiento?
- ¿Qué fuentes de información utilizan para capacitarse?









# Código de Ejemplo


- Cada servidor viene con código de ejemplo
  - Printenv, Northwind, etc.
- Cada pieza principal de código de ejemplo *es un ejemplo perfecto de que no hacer*
- ¿Es de extrañar que escribamos aplicaciones web defectuosas?




# Google Ilustra el Problema

- <https://www.exploit-db.com/google-hacking-database>
  - Vulnerabilidades Conocidas
  - Contraseñas
  - Inyección SQL
  - Números de Tarjetas de Crédito
  - Información de Clientes
  - Explotable Remotamente (Puntos de apoyo)
  - ... y muchos más

# Google Hacking Database



**EXPLOIT  
DATABASE**

[GET CERTIFIED](#)

## Google Hacking Database

[Filters](#) [Reset All](#)

[Show](#)

[Quick Search](#)

Date Added	Dork	Category	Author
2020-05-18	<a href="#">inurl:wp-content/plugins/photo-gallery</a>	Advisories and Vulnerabilities	Abhi Chitkara
2020-05-18	<a href="#">intitle:"WEBCAM 7 " -inurl:/admin.html</a>	Various Online Devices	Nisankh Acharjya
2020-05-18	<a href="#">site:*/auth/forgot</a>	Pages Containing Login Portals	Reza Abasi
2020-05-18	<a href="#">inurl:wp-content/plugins/team-members</a>	Advisories and Vulnerabilities	Abhi Chitkara
2020-05-18	<a href="#">inurl:wp-content/plugins/easy-login-woocommerce</a>	Advisories and Vulnerabilities	Abhi Chitkara
2020-05-18	<a href="#">inurl:wp-content/plugins/wp-product-review</a>	Advisories and Vulnerabilities	Abhi Chitkara
2020-05-18	<a href="#">inurl:/emptyView4.html</a>	Pages Containing Login Portals	MiningOmerta
2020-05-18	<a href="#">intitle:"index of" "tinyfilemanager.php"</a>	Sensitive Directories	Abhi Chitkara

# Buenas Prácticas

- ¿Qué acciones simples podemos hacer para crear aplicaciones seguras?
  - Validación y Desinfección de Entradas
  - Control y Manejo de Errores
  - Administración Robusta de Sesiones
  - Mediación Completa de la Aplicación
  - Solución de Múltiples Capas

# Validación de Entradas

- Verificar todo lo que ingresa a la aplicación para comprobar que eso conforma lo que la aplicación está esperando
- No confiar en nada
  - Desde el cliente
  - Desde la base de datos
  - Cualquier cosa externa a la aplicación



# Desinfección

- Identificar todas las formas de entrada que serán necesarias para que la aplicación funcione (letras, números, etc.)
- Todo debe ser desinfectado a través de un filtro estándar
  - Descartar todo lo que no identificamos previamente

# Control de Errores

- Manejar todas las condiciones de error
  - Cosas que se pueden predecir que van a suceder
    - Errores de base de datos
    - Errores de red
  - Obtener y manejar lo inesperado también

# Administración Robusta de Sesiones

- **Definición suficiente de Sesión:** Una sesión es una instancia única de un usuario a través del curso de su interacción con la aplicación.
- Los IDs de Sesión generan trazabilidad.
- Generar IDs de Sesión Fuertes
  - Aleatorios
  - Fuerte asociación con el cliente
  - Considerar que sean a prueba de falsificaciones.

# Mediación Completa

- El principio de mediación completa es que hay solamente una manera de ingresar y salir de una aplicación.
- Considerar un punto único de ingreso
  - "miaplicacion.com/index.php"
  - Todo pasa por aquí.
  - Nada más es accesible en el sitio sin pasar a través de este punto de acceso.
  - Reutilización de Código confiable.

# Solución de Múltiples Capas

- Para tener seguridad robusta, se recomienda usar tres capas
  - Presentación, Aplicación (o Negocio), Persistencia
- La seguridad es más baja con dos o menos capas
  - La complejidad y el costo puede restringir las implementaciones a dos capas

# Dos Capas vs. Tres Capas

- En una solución de dos capas, la capas de Presentación y Aplicación trabajan juntas
- ¿Se almacena información sensible en la capa de Persistencia?
  - Contraseñas
  - Información del Cliente
- ¿Dónde están las credenciales de base de datos?
- Si se cifran los datos en la base de datos, ¿dónde están las claves de cifrado?

# Conceptos Básicos de Aplicaciones Web

# HTTP

- HTTP es sólo texto
  - HyperText Transfer Protocol
  - Los datos binarios pueden ser envueltos en HTTP
- Método de Comunicación
  - Los clientes solicitan páginas de servidores usando HTTP
  - Los servidores responden con HTML envuelto en encabezados HTTP



# Formularios Web

- La etiqueta <FORM> agrupa entradas
  - El método define cómo los formularios son enviados
    - GET o POST
  - La acción define qué hacer cuando es enviado.
  - Las etiquetas <INPUT> incluyen varios tipos
    - Campos de Texto
    - Contraseñas
    - Áreas de Texto
    - Botones de Envío
  - También cuadros de selección, listas desplegables, etc.

# GET/POST

- HTTP viene en una cantidad de sabores
  - TRACE, HEAD, GET, POST, ...
- GET
  - Límite de 255 caracteres
  - Toda la entrada es incluida en la URL
- POST
  - Sin límite duro (configurable)
  - Toda la entrada es incluida en el cuerpo de la petición HTTP.

# Métodos HTTP y REST (REpresentational State Transfer)

- No dejar que las aplicaciones RESTful engañen
  - Usan CRUD
    - **Create:** POST
    - **Read:** GET
    - **Update:** PUT
    - **Delete:** DELETE
  - Los servidores web no los implementan (en Front-end)
  - Emulan CRUD a través de elementos ocultos
  - Solamente usan GET y POST

# Detalles de POST

- La URL puede aún contener parámetros
- Encabezados HTTP
- Identificador del Navegador
- Referente
- Cookies

# Cookies

- Las Cookies son simplemente texto
  - El servidor envía un nombre de cookie con un valor arbitrario
  - El cliente lo almacena y luego envía la cookie con cada petición
  - Es posible enviar más de una cookie

# Detalles de la Cookie

- Cookies configuradas con el encabezado Set-Cookie
  - Name
  - Domain
  - Path
  - Secure
  - Expires

# Flujo de la Cookie

1. El navegador envía una petición
2. El servidor establece y devuelve una cookie
3. El navegador envía la cookie de vuelta en cada petición

# Consecuencias de la Cookie

- Encabezado “Secure”
  - Los contenidos de la cookie pueden ser rastreados si no está marcada
- Dominio
  - ¿Un equipo específico o un dominio entero?
- Path
  - ¿Es éste un servidor compartido?



# Cambiando Cookies

- Cookies persistentes
  - Comportamiento típico
  - Cookies almacenadas en disco para uso futuro
  - Fácil manipulación
- Cookies no persistentes
  - No pensadas para ser almacenadas en disco
  - Los desarrolladores piensan que no pueden ser cambiadas
  - La manipulación es más difícil, pero no imposible

# SSL/TLS

- Tunnelización para HTTP
  - Solamente cifrado en tránsito
  - Texto plano en los endpoints

# AJAX

- Asynchronous JavaScript and XML
  - Google maps es un ejemplo perfecto
  - Las páginas cargan usando HTTP
  - JavaScript pide información de manera asíncrona, típicamente usando XML como transporte
  - La página se actualiza sin recargar

# CSS

- Cascading Style Sheets
  - Estándar Web
  - Permite un control sobre el diseño
  - Aplica estilos a elementos en páginas
    - Consistencia
    - Permite a los usuarios navegar usando sus propios estilos
- No es importante para analizar vulnerabilidades

# OWASP

- Open Web Application Security Project
  - Guías de Desarrollo
  - Recursos con Mejores Prácticas
  - Base de Datos de Vulnerabilidades Web
  - Herramientas de Aprendizaje
  - Herramienta de Auditoría OWASP ZAP
  - <https://www.owasp.org>

# OWASP ZAP

- Esencialmente un “Hombre en el medio”
  - Ver información históricamente
  - Interceptar y modificar información
  - Soporta SSL
  - Y mucho más

# BURP Suite

- <https://portswigger.net/>
  - Es el mismo concepto que OWASP ZAP
  - Algunas características mejor.
  - Tiene versiones comerciales.

# Algunos Puntos para Recordar

- HTTP/HTML son solamente texto
- GET envía los parámetros en la URL
- POST envía los parámetros en el cuerpo del mensaje
  - POST también puede enviar parámetros en la URL
- SSL provee cifrado, no seguridad
- Las cookies son solo piezas de texto arbitrario
- El cliente controla todo lo que es enviado al servidor



# Seguridad de Servidores

# Indexación de Directorios

- Es más divulgación de información que una vulnerabilidad
  - Acceso a un directorio en lugar de a una página web
  - Podría ser intencional
- Configurable en todos los servidores web principales
  - Generalmente por defecto se carga “index.htm” o “index.html”
  - Comprobar si la configuración coincide con la postura de seguridad y el propósito

## Index of /.git/

<a href="#">../</a>	10-Oct-2014 18:22	-
<a href="#">branches/</a>	10-Oct-2014 18:22	-
<a href="#">hooks/</a>	10-Oct-2014 18:22	-
<a href="#">info/</a>	10-Oct-2014 18:22	-
<a href="#">logs/</a>	10-Oct-2014 18:22	-
<a href="#">objects/</a>	10-Oct-2014 18:22	-
<a href="#">refs/</a>	10-Oct-2014 18:22	-
<a href="#">COMMIT_EDITMSG</a>	10-Oct-2014 18:22	15
<a href="#">HEAD</a>	10-Oct-2014 18:22	23
<a href="#">config</a>	10-Oct-2014 18:22	92
<a href="#">description</a>	10-Oct-2014 18:22	73
<a href="#">index</a>	10-Oct-2014 18:22	912

# Encabezados

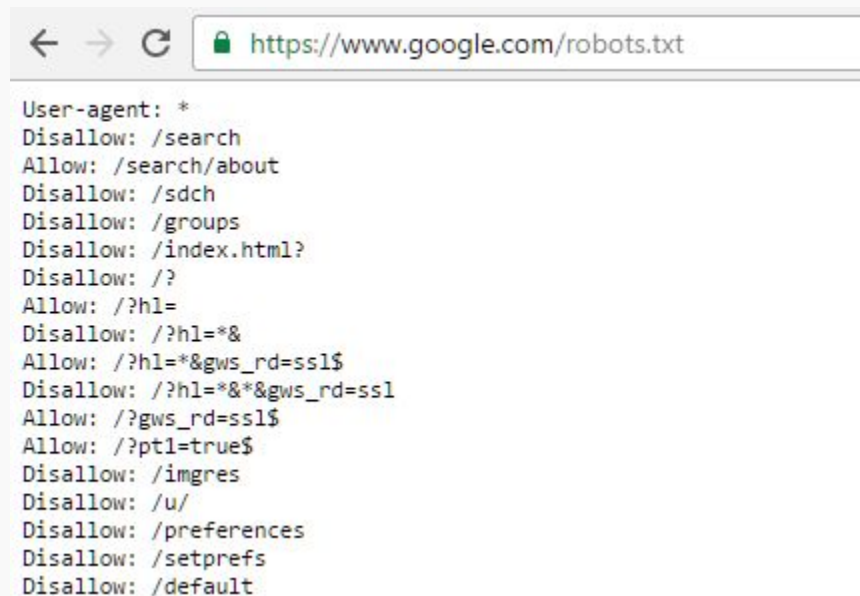
- Siguen siendo una exposición
  - Ideales para identificar exploits
  - Encontrar sitios donde el mantenimiento no es frecuente
  - Encontrar Complementos (plugins)
  - ¿Qué tan probables es que alguien esté mirando mi sitio?

Header	Value
X-Frame-Options	SAMEORIGIN
Date	Fri, 03 Jul 2015 14:52:20 GMT
Content-Length	17343
#status#	HTTP/1.1 200 OK
Content-Encoding	deflate
X-UA-Compatible	IE=9
Set-Cookie	CMSPreferredCulture=en-US; expire...
Set-Cookie	ASP.NET_SessionId=zhmrvaj4gl1uf...
Content-Type	text/html; charset=utf-8
Server	Microsoft-IIS/7.0
Cache-Control	private, no-store, must-revalidate

Headers (11) Attachments (0) SSL Info (2 certs) JMS (0)

# Robots.txt

- Destinado a controlar la indexación
  - Configurar restricciones basadas en rastreadores web
  - Comúnmente regalan información extra

A screenshot of a web browser window displaying the Google robots.txt file. The address bar shows the URL 'https://www.google.com/robots.txt'. The main content area displays the text of the robots.txt file, which includes various 'User-agent', 'Disallow', and 'Allow' directives for controlling search engine indexing.

```
User-agent: *
Disallow: /search
Allow: /search/about
Disallow: /sdch
Disallow: /groups
Disallow: /index.html?
Disallow: /?
Allow: /?hl=
Disallow: /?hl=*%
Allow: /?hl=*%gws_rd=ssl$
Disallow: /?hl=*%*%gws_rd=ssl$
Allow: /?gws_rd=ssl$
Allow: /?ptl=true$
Disallow: /imgres
Disallow: /u/
Disallow: /preferences
Disallow: /setprefs
Disallow: /default
```

# Reuniendo Elementos

- Elementos de la Lista de Verificación:
  - Contenido por defecto (ejemplos de código)
  - Indexación de Directorios
  - Configuración base segura
    - Examinar contenido oculto
  - Sistema Operativo seguro
  - Despliegue de Red seguro

# Pruebas de Configuración

# ¿Cómo funcionan los Analizadores?

- La estrategia típica es identificar primero el servidor
  - Huella dactilar (Fingerprint)
    - del Sistema Operativo
    - del Servicio Web
    - de los complementos
- Prueba por problemas del Servidor basado en los resultados
  - Esto acelera las cosas
    - Versiones viejas de IIS: Ataque Unicode
    - Versiones viejas de Apache: Ataques de Codificación fragmentada

# Posible Agujero en las Pruebas

- ¿Qué pasa si el defensor oculta la verdad?
  - Ocultar información para IIS
    - <https://forums.iis.net/t/1239190.aspx?Suppress+IIS+information>
  - Reconfigurar Apache
    - `#define SERVER_BASEVENDOR "Microsoft"`
    - `#define SERVER_BASEPRODUCT "IIS"`
    - `#define SERVER_BASEREVISION "6.0"`
- Comprobar que la herramienta pueda probar todo independientemente de los banners



# ¿Analizadores de Propósito General?

- Nessus y similares aún son útiles
  - Buenos para comprobar la seguridad en general
  - Encontrarán material básico por defecto, etc.
  - ¿Por qué usar un martillo para un tornillo?
- Los analizadores web generalmente realizan muchas más pruebas con mayor rigor

# Advertencia Automatizada

- El análisis automatizado es muy bueno
  - Rápido y Fácil
  - Muchas herramientas para apuntar y hacer click
  - Reportes fáciles de leer
- Sin embargo...
  - Pueden encontrarse solamente vulnerabilidades conocidas
  - Pueden usarse solamente técnicas conocidas y buscar por patrones conocidos
  - ¿Qué pasa si uno escribe su propia aplicación web?
- Aún no hay mejor tester de aplicaciones web que una persona bien entrenada

# Fuzzers

- Fuzzing es una táctica antigua con nuevas herramientas
  - Encuentra todos los orificios de entrada que tiene una aplicación
  - Automáticamente pega todo tipo de basura en todos ellos al mismo tiempo
- Algunas herramientas hacen ésto
  - Nessus
  - OWASP ZAP
- Aún no hay nada como un ser humano para algunas tareas

# Introducción a la Autenticación

- Objetivos de Auditoría
  - Determinar si el mecanismo de autenticación es seguro
- Controles
  - Implementación adecuada
  - Cifrado
  - Credenciales fuertes
- Métodos Comunes de Autenticación
  - Autenticación Básica HTTP
  - Autenticación basada en Formularios
  - Certificados de Cliente

# Métodos - Autenticación Básica HTTP

- Fortalezas
  - Fácil de implementar
- Debilidades
  - No fácilmente borrrable desde el navegador
  - No tiene cifrado
  - Fácil para fuerza bruta
  - Actúa como ID de sesión
    - Confunde secreto de largo plazo con secreto de corto plazo

- Mejores Prácticas
  - Cifrar todo el tráfico durante y luego de la autenticación

## Authentication Required

http://awesomesite.app requires a username and password.

Your connection to this site is not secure

User Name:

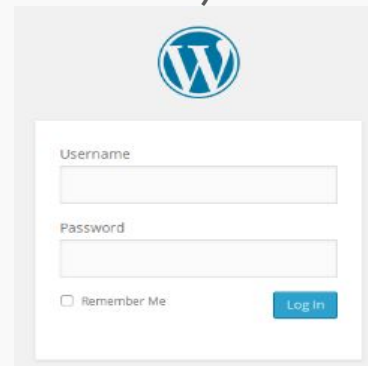
Password:

Cancel

Log In

# Métodos - Autenticación Basada en Formularios

- Fortalezas
  - Fácil de Implementar
  - Balance razonable entre seguridad y conveniencia para los usuarios
- Debilidades
  - Potencial de fuerza bruta
  - Mala configuración conduce a exponer credenciales
  - Mismas debilidades que nombre de usuario/contraseña
  - No cifrado por defecto
- Mejores Prácticas
  - Usar HTTP POST para enviar credenciales de usuario
  - Envío de credenciales de usuario vía cifrado (ej. SSL)
  - Campos de contraseña usar TYPE=PASSWORD
  - Considerar el uso de Tokens (ej. Google Authenticator)

A screenshot of a WordPress login form. At the top is the WordPress logo (a blue 'W' inside a circle). Below it are two input fields: 'Username' and 'Password'. Under the 'Password' field is a checkbox labeled 'Remember Me'. To the right of the checkbox is a blue 'Log In' button.

# Métodos - Certificados del lado del Cliente

- Fortalezas
  - Altamente seguro
  - Permite el “no repudio”
  - Confidencialidad
  - Autenticación mutua
- Debilidades
  - Movilidad e interoperabilidad limitada
  - Administración (ej: Anulación de Certificados)
- Mejores Prácticas
  - Mejor forma de autenticación para B2B y necesidades de alta seguridad
  - Usar token de hardware (ej: tarjeta inteligente) para incrementar movilidad

# Autenticación NTLM

- Solución de Single Sign-On (SSO) para aplicaciones ASP.NET en IIS
  - Solamente útil para aplicaciones de Intranet
  - Requiere el uso de IIS
  - No es un estándar
    - Aprovecha la autenticación basada en hashes
  - Es segura cuando el ID de sesión está protegido



# Carteles de Advertencia



[acpssw.gsfc.nasa.gov](http://acpssw.gsfc.nasa.gov)

---

## NASA IT Security Warning Banner

By accessing and using this information system, you acknowledge and consent to the following: You are accessing a U.S. Government information system, which includes: (1) this computer; (2) this computer network; (3) all computers connected to this network including end user systems; (4) all devices and storage media attached to this network or to any computer on this network; and (5) cloud and remote information services. This information system is provided for U.S. Government-authorized use only. You have no reasonable expectation of privacy regarding any communication transmitted through or data stored on this information system. At any time, and for any lawful purpose, the U.S. Government may monitor, intercept, search, and seize any communication or data transiting, stored on, or traveling to or from this information system. You are NOT authorized to process classified information on this information system. Unauthorized or improper use of this system may result in suspension or loss of access privileges, disciplinary action, and civil and/or criminal penalties.

---

# Recolección de Nombres de Usuario

- **Amenaza:** Un tercero malicioso podría recolectar nombres de usuario válidos
  - A veces se revela demasiada información
    - Login fallido
    - Creación de cuenta (Gmail)
    - Opciones de restablecimiento de contraseña
- **Recomendación:** Indicar a través de un solo mensaje que la autenticación es incorrecta.

# Técnica de Auditoría para Recolección de Nombres de Usuario

- Prueba
  - Fallar intencionalmente intentos de inicio de sesión, cubriendo cada escenario posible
  - Ser cuidadoso de no bloquear cuentas, eso ocurre después
  - Si es posible, probar cada lenguaje posible (ej: español e inglés).
- Escenarios
  - Usuario Válido - PIN Inválido - Respuesta A
  - Usuario Inválido - PIN N/A - Respuesta B
- Aún el tiempo de respuesta para retornar el error puede ser un indicador

# Ataques de Contraseña por Fuerza Bruta

- **Amenaza:** Algunos sitios permiten un número ilimitado de intentos fallidos de inicio de sesión (no bloquean la cuenta).
- **Impacto:** [Autorización] Cuentas comprometidas a través de ataques de fuerza bruta contra la contraseña.
- **Recomendación:** Bloquear las cuentas (cuidadosamente).

# DoS por Fuerza Bruta - Bloqueos de Cuentas

- Algunos sitios hacen cumplir el bloqueo de cuentas luego de un número específico de intentos de inicio de sesión fallidos
- **Amenaza:** Un gran número de cuentas de usuario pueden ser bloqueadas usando herramientas automatizadas
  - La autenticación básica HTTP y la basada en formularios son fácilmente atacadas (aún con SSL)
- **Impacto:** Disponibilidad / DoS
- **Recomendación:** Usar bloqueos de velocidad

# Herramienta - THC Hydra

- <https://github.com/vanhauser-thc/thc-hydra>
- Multiplataforma
- **Propósito:** Ataque de Fuerza Bruta a Autenticación Web

# ¿Preguntas?

¡Muchas Gracias!