Bypassing your .htACCESS

Matías Katz
@matiaskatz

Maximiliano Soler
@maxisoler

Agenda

- Presentación
- Conceptos básicos
  - .htaccess - ¿Qué es y para qué sirve?
  - Tipos de autenticación
- Protección de directorios con .htaccess
- DEMO
- Conclusiones
- Recomendaciones

Presentación

Presentación

Matías Katz (**@matiaskatz**) es un Penetration Tester que se especializó en seguridad sobre aplicaciones Web. Disfruta creando herramientas para descubrir y explotar sobre cualquier software o red. *Master* del Super Mario.

Maximiliano Soler (**@maxisoler**) actualmente vive en Buenos Aires y trabaja como Analista de Seguridad en un Banco Internacional. Maxi ha descubierto vulnerabilidades en diferentes aplicaciones Web y productos de Microsoft.

Conceptos básicos

Conceptos básicos

.htaccess *- ¿Qué es y para qué sirve?*

`.htaccess = hypertext access`

Es un archivo de configuración distribuida, permite realizar cambios de configuración en función de cada **directorio** y subdirectorios.

Sin necesidad de editar el archivo de configuración principal de Apache.

Conceptos básicos

.htaccess *– Algunos ejemplos de uso*

Redireccionamiento

Redirección de URLs

Listar directorios

Reescritura de URLs

Forzar diálogos entre cliente y servidor

Respuestas de error personalizadas

Autorización

Autenticación

Conceptos básicos

Tipos de autenticación

Basic Authentication

- Método en el cual un cliente provee usuario y contraseña cuando se realiza la solicitud.

Formatos: Texto plano, Crypt (Unix), SHA1, MD5.

Digest Authentication

- El servidor Web negocia las credenciales con el navegador del cliente. Aplica función de hash a la contraseña antes de enviarlo.

Protección de directorios con .htaccess

Protección de directorios con .htaccess

Ejemplo:

```
AuthUserFile /[FOLDER]/.htpasswd
AuthName "Protected Area"
AuthType Basic

<Limit GET POST>
require valid-user
</Limit>
```

Ruta completa al archivo htpasswd.

Descripción de pantalla de login.

Línea requerida.

Inicia el tag límite en GET y POST.

Setea la restricción a un usuario válido.

Finalizar el tag límite.

DEMO

Presentación HTExploit

Conclusiones

Conclusiones

Conclusiones (**I**)

- El problema reside en cómo se limitan las solicitudes **HTTP** y sobre **QUIÉN**.

- El archivo .htaccess indica que "**`require valid-user`**" sólo aplica para **`GET`** y **`POST`**. Por lo tanto las solicitudes de tipo *loquesea* quedan PERMITIDAS.

Conclusiones

Conclusiones (**II**)

`AuthUserFile /[FOLDER]/.htpasswd`	Ruta completa al archivo htpasswd.
`AuthName "Protected Area"`	Descripción de pantalla de login.
`AuthType Basic`	Línea requerida.
`<Limit GET POST>`	Inicia el tag límite en GET y POST.
`require valid-user`	Setea la restricción a un usuario válido.
`</Limit>`	Finalizar el tag límite.

```
<LimitExcept GET POST>
Order Allow,Deny
Deny from all
</LimitExcept>
```

Restringe los controles de acceso a todos los métodos HTTP excepto a los que se especifiquen.

Conclusiones

Conclusiones (**III**)

Desde el código podemos hacer lo siguiente:

- Verificar sí la variable `$PHP_AUTH_USER` está seteada.

y/o

- Verificar sí `$_SERVER["REQUEST_METHOD"]` lleva **GET** o **POST**, caso contrario arrojar un error.

Recomendaciones

Recomendaciones

- **Informarse** sobre los últimas versiones y Vulnerabilidades (*listas de seguridad?*)
- **Prevenirse** de ataques del tipo DoS (*Denegación de Servicios*).
- Verificar los **permisos** con los cuales se ejecuta el Servidor Web.
- Prestar atención a los **scripts** utilizados y de terceros (contenido dinámico, CGI, etc.).
- Proteger el **sistema de archivos** del Servidor.
- Ver los **Logs**!

Sobre HTExploit

Características

- Abierto y gratuito.
- Programado en Python.
- Modularizado **:)**
  - SQL Injection
  - Local File Inclusion
  - Remote File Inclusion
- **¿Cuándo se publicará?**…*Pronto*! **:)**

Muchas Gracias!

Matías Katz
(@matiaskatz)

Maximiliano Soler
(@maxisoler)