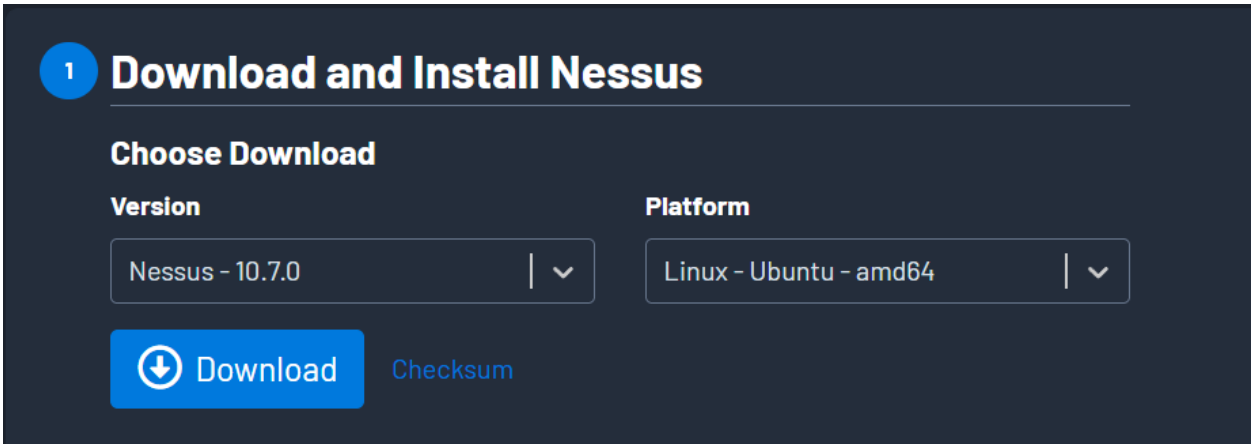
EXERCISE 2 – Vulnerability Scanning and Management

Task 1

From your Host OS, navigate in your browser to
https://www.tenable.com/downloads/nessus?loginAttempted=true

Select Nessus 10.7.0 on the Linux – Ubuntu – amd64 Platform



We will be downloading it to your HOST operating system. Click on DOWNLOAD.

A License Agreement will pop up. We will be clicking on Agree to move forward with the download.

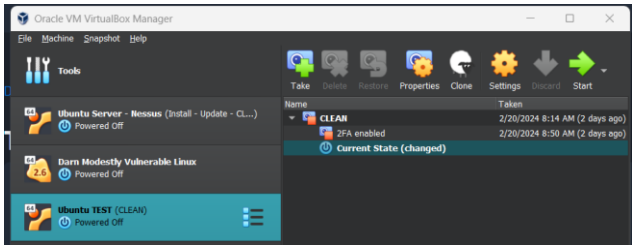
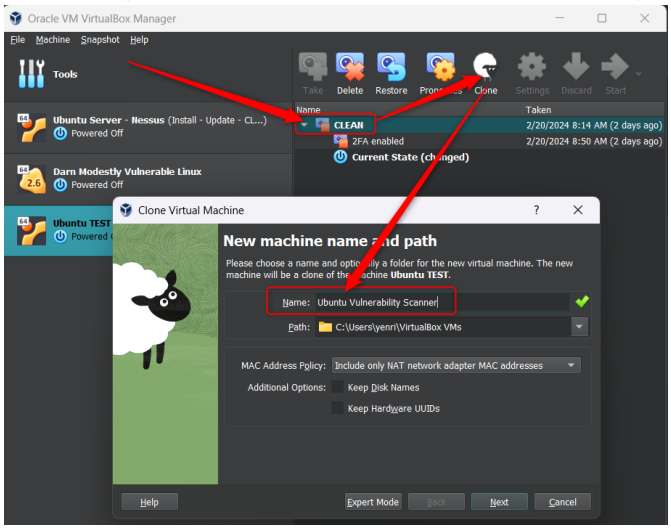
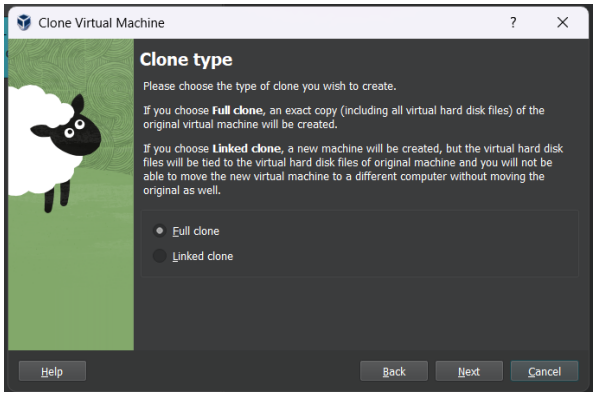
Once the file has been downloaded, we will open the terminal in your HOST operating system and use the CLI to SCP 9 (Secure Copy) to our Linux Guest VM.

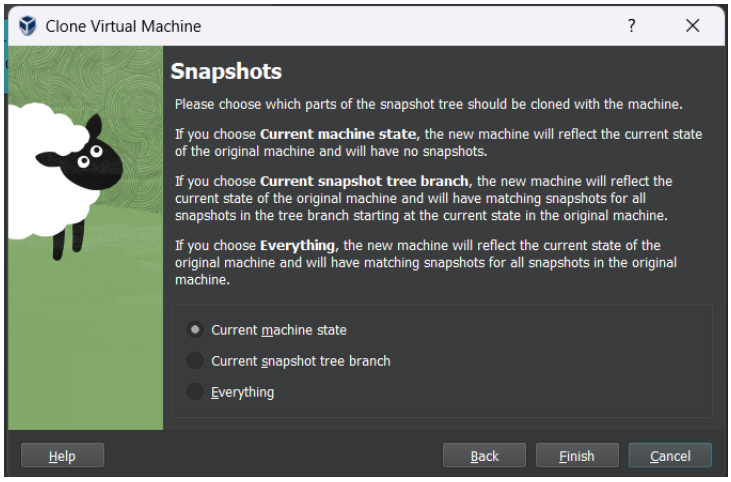
Task 2

First thing we are going to do is CLONE your Ubuntu Server VM from its original CLEAN snapshot.

This will allow us to have a separate server for just this Exercise, it also teaches you how to CLONE VM's in Virtualbox.

Having the ability to CLONE from a CLEAN snapshot makes spinning up new VM's lightning fast. You've already done the hard work of deploying a CLEAN and updated version of Ubuntu Server. Its best we capitalize on that effort.

1. Load up the Virtual Box VirtualBox Manager
2. Power off all VM's in Virtual Box
3. Select the Ubuntu Server VM object on the left hand side
    a. 
    b. Select the button on the right hand side and have the snapshots loaded in the details pane.
4. Select the Post Installation – CLEAN Snapshot. Meaning we will have the cleanest and unused version of your Linux Server available.
5. Click on the sheep "CLONE" button in the ribbon on the top.
    a. 
    b. Rename it Ubuntu Vulnerability Scanner
    c. Click NEXT
6. We will be creating a FULL CLONE
    a. 

7. We will only be cloning the CURRENT MACHINE STATE
   a. 

   b. Click FINISH
8. This will make a complete copy of the CLEAN IMAGE of your server.
9. This is much faster than installing a server from scratch and is very similar to how IT departments deploy new systems.
10. This will not take a lot of time. (Remarkable)
11. Double Check that this server has the right settings.
    a. Processor
    b. RAM
    c. Network Config
12. It should have cloned the exact configuration, but 2x check.
13. You now have a NEW Linux Server to deploy a Vulnerability Server

Task 3

Prerequisites

- Ubuntu Server 22.04 Installed, Updated, Upgraded and running in Virtual Box
- Ubuntu Guest VM Networking settings is set to Bridged Mode in Virtual Box
- The Ubuntu Guest VM is running
- Have an open CLI in your HOST OS, and test that you can SSH into the Guest VM

Power up your new Ubuntu Vulnerability Scanner Server

Open a command prompt / Terminal and SSH into your cloned server from your HOST OS

Open up a 2nd command prompt / Terminal in your HOST OS (Tab or window)

In the 2nd command prompt / Terminal (tab or window) navigate to your downloads folder.

In windows use the directory command, in mac/linux use the ls command

Windows:

dir *.deb

MacOS/Linux

ls *.deb

The output should be something like this:

```
C:\Users\yenri\Downloads>dir *.deb
 Volume in drive C has no label.
 Volume Serial Number is 6CD1-13E7

 Directory of C:\Users\yenri\Downloads

02/22/2024  08:48 AM        68,540,738 Nessus-10.7.0-ubuntu1404_amd64.deb
               1 File(s)     68,540,738 bytes
               0 Dir(s)  1,550,921,539,584 bytes free
```

Whatever the output, it should show the file that you downloaded in your HOST OS.

Using SCP in the HOST OS CLI, copy the file to your Linux Server.

<NO, I'm not going to give you the command, this has already been taught to you, you need to review your notes, remember, or use the following RECAP to figure out how to copy the file from your HOST OS to your GUEST VM.>

Online Tutorial https://linuxize.com/post/how-to-use-scp-command-to-securely-transfer-files/)

The proper successful output should look like this:

```
Nessus-10.7.0-ubuntu1404_amd64.deb                  100%   65MB  28.1MB/s   00:02
```

OK FINE! I'll show you. Let's review SCP.

SCP Command RECAP

The `scp` command is used in Unix-like operating systems (such as Linux and macOS) to securely transfer files between a local and a remote system over a network. It stands for "secure copy protocol." It utilizes SSH (Secure Shell) for data encryption and authentication.

The basic syntax of the `scp` command is:

scp [options] [source] [destination]

- `[options]`: Optional flags that modify the behavior of the command.
- `[source]`: Specifies the path of the file or directory you want to copy.
- `[destination]`: Specifies the destination path where the file or directory will be copied.

For example, to copy a local file to a remote server, you would use:

scp /path/to/local/file username@remote_host:/path/to/destination

PROTIP if you don't provide /path/to/destination, so you just end in : it drops it in your home folder.

To copy a file from a remote server to your local machine:

scp -r /path/to/local/directory username@remote_host:/path/to/destination

The `scp` command prompts for the password if SSH key-based authentication is not set up. Additionally, it supports various options like `-P` for specifying a custom SSH port, `-i` for specifying the identity file, and `-C` for enabling compression during transfer, among others. You can explore more options and details in the `scp` command's manual page by running `man scp` in your terminal.

If you are not using an identity file, you will be merely typing in your password.

Task 4

Now that the file has been successfully copied via SCP to your Linux Server, you'll want to validate that it is in your home folder.

What command could you type in Linux to display the contents of that folder?

<Again, I'm not giving that to you>

Proper output (Your folder contents will vary)

Nessus-10.7.0-ubuntu1404_amd64.deb pride.txt

ajay@server1:~$

Now that file will not allow you to install it because you downloaded it off the internet and Linux is protecting you. You'll need to modify those files permissions.

Once you have changed the permissions for that file so that it can be executed, you'll be able to install the application.

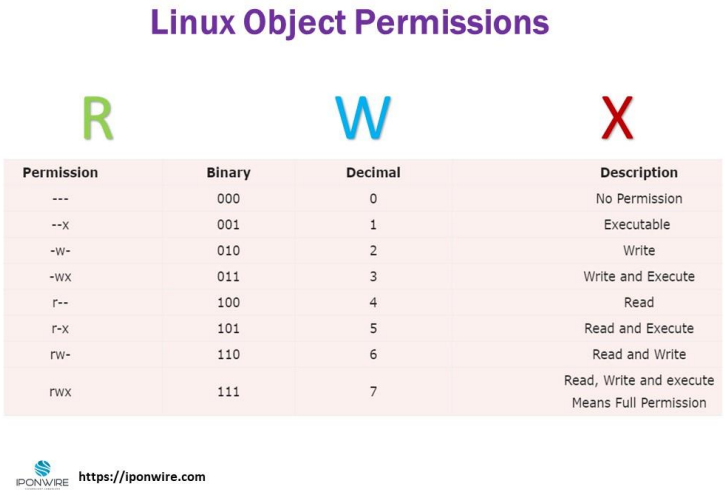
Linux Directory and File Security RECAP

File Permissions:

Linux uses a permission system that defines who can read, write, and execute files or directories. Permissions are represented by three sets of characters: `r` (read), `w` (write), and `x` (execute), corresponding to three types of users: owner, group, and others.

- **Owner**: The user who owns the file or directory.
- **Group**: The group associated with the file or directory.
- **Others**: Any other user on the system.

Each file or directory has three sets of permission bits, represented as `rwx`. These bits indicate the permissions for the owner, group, and others, respectively.



For example:

-rw-r--r-- 1 owner group 1048 Jan 31 10:00 example.txt

In this example, the first set of permissions `-rw-r--r--` indicates that the owner has read and write permissions (`rw-`), the group has read-only permissions (`r--`), and others have read-only permissions (`r--`).

CHMOD

The `chmod` command in Linux is used to change the permissions of files and directories. It can be used in two main ways: the octal method and the symbolic method.

Octal Method:

In the octal method, permissions are represented by three octal digits (0-7), each representing the permission bits for the owner, group, and others, respectively. Each permission is represented by a combination of read (4), write (2), and execute (1) permissions, with the sum of these values representing the desired permissions. Here's how it works:

- **Read (r): 4**
- **Write (w): 2**
- **Execute (x): 1**

To use the octal method with `chmod`, you calculate the desired permission value for each user category (owner, group, and others), and then specify these values as an octal number.

For example, to give read, write, and execute permissions to the owner, and only read permissions to the group and others, you would use:

chmod 744 file.txt

Here, `7` represents `rwx` for the owner, and `4` represents `r--` for both the group and others.

Symbolic Method (+x Method):

In the symbolic method, you can use symbols to specify permissions relative to the current permissions of the file. The symbols used include:

- `+`: Adds the specified permission.
- `-`: Removes the specified permission.
- `=`: Sets the specified permission explicitly, removing any others.

You can combine these symbols with the permission characters (`r`, `w`, `x`) and the user categories (`u` for owner, `g` for group, `o` for others, `a` for all). Here's how it works:

chmod +x file.txt

This command adds execute (`x`) permission to the file `file.txt` for all users. Similarly, you can remove permissions or set permissions explicitly using the same syntax.

Summary:

- **Octal Method**: Directly sets permissions using octal numbers representing permission bits.
  - Example: `chmod 744 file.txt`
- **Symbolic Method (+x Method)**: Adds, removes, or sets permissions relative to the current permissions.
  - Example: `chmod +x file.txt`

Both methods provide flexibility in managing file permissions according to your requirements. Choose the method that best suits your needs and preferences.

Ownership:

Every file and directory in Linux is associated with an owner and a group. The owner is usually the user who created the file, and the group is a set of users defined on the system. Ownership is represented by the username of the owner and the group name associated with the file or directory.

Changing Permissions and Ownership:

You can change file permissions and ownership using the `chmod` and `chown` commands, respectively.

- **chmod**: Used to change file permissions.
  - Example: `chmod 755 file.txt` sets `rwx` for the owner, and `r-x` for group and others.
- **chown**: Used to change file ownership.
  - Example: `chown user:group file.txt` changes the owner to `user` and the group to `group`.

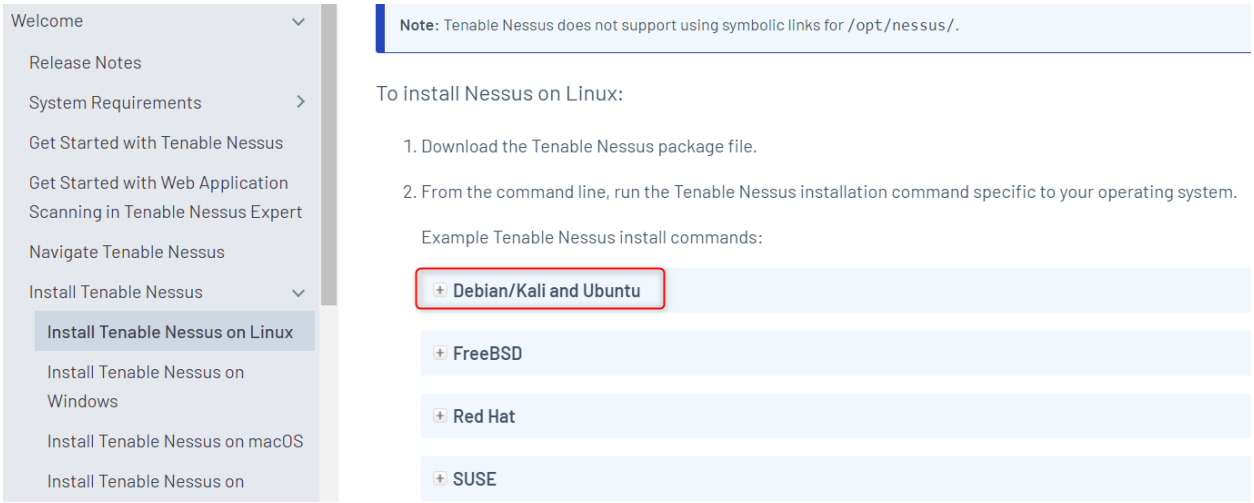
In Linux, file and directory security is managed through permissions and ownership. Understanding and properly setting these attributes is crucial for ensuring the integrity and security of your system. Permissions control who can read, write, and execute files, while ownership determines which users and groups have control over them. Additionally, special permission bits provide further control over file execution and deletion.

Here is the guide that the author chose to conduct the installation:

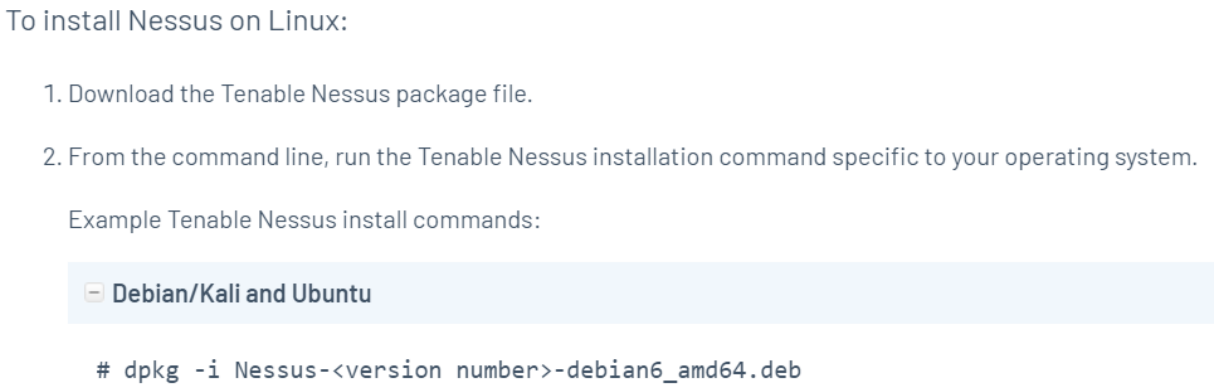
https://docs.tenable.com/nessus/Content/InstallNessusLinux.htm

You'll note that you can install Nessus on MANY TYPES of OSes. It is utterly important to follow the guide to the Server OS you are running.

Since we are using Ubuntu Server 22.04 we'll chose that one.



Click the plus on the left of your target OS and it will expand and show you the commands in which to follow.



It is asking us to use the dpkg command which we previously covered. But nevertheless, let's do a quick RECAP.

DPKG RECAP

`dpkg` is a low-level package management system used by Debian-based Linux distributions, including Ubuntu. It stands for "Debian package" and is used to install, remove, and provide information about `.deb` packages. While `dpkg` itself is a powerful tool, it does not handle dependencies; it will not automatically download or install packages that a package depends on or remove dependencies that are no longer needed. Higher-level tools like `apt` (Advanced Package Tool) and `apt-get` are used for managing packages along with their dependencies, but they, too, rely on `dpkg` at a lower level to manage the actual installation and removal of packages.

Installing a Package: To install a `.deb` package, you can use the command:

sudo dpkg -i package_name.deb

Removing a Package: To remove an installed package without removing its configuration files, use:

sudo dpkg -r package_name

Listing Installed Packages: You can list all packages installed on your system with:

dpkg -l

Listing Installed Packages since the output can be large, is usually paired with pipe and grep to filter the output to JUST WHAT YOU WANT TO SEE.

Example:

dpkg -l | grep package_name

Handling Dependency Issues:

When `dpkg` encounters dependency problems (for example, if a package depends on another package that is not installed), it will refuse to install the package until the dependencies are resolved. In such cases, you can use `apt-get` or `apt` to install the missing dependencies:

sudo apt install -f

The `-f` option stands for "fix-broken." It attempts to correct a system with broken dependencies in place.

While `dpkg` is a powerful tool for managing `.deb` packages directly, for daily package management tasks, especially those involving handling dependencies, it's often more convenient to use `apt` or `apt-get`. These higher-level tools provide a more user-friendly interface and automatically handle many of the complexities of package management, including dependency resolution and automatic updates.

Since we already have downloaded the Nessus Debian file, we'll just be using `dpkg` to install.

Installing Nessus via dpkg

To install Nessus on Linux:

1. Download the Tenable Nessus package file.
2. From the command line, run the Tenable Nessus installation command specific to your operating system.

Example Tenable Nessus install commands:

Debian/Kali and Ubuntu

```
# dpkg -i Nessus-<version number>-debian6_amd64.deb
```

The guides instruction is to use the command `dpkg -i Nessus-<version number>-debian6_amd64.deb` to install the Nessus Debian package.

So go ahead and try. You'll fail, but try.

`dpkg -i Nessus-<version number>-debian6_amd64.deb`

```
ajay@server1:~$ dpkg -i Nessus-10.7.0-ubuntu1404_amd64.deb
dpkg: error: requested operation requires superuser privilege
ajay@server1:~$ mkdir: cannot create directory '/run/needrestart': Permission denied
```

Do you know how to overcome this obstacle?

I hope so, because I won't be telling you. You should know that by this point.

Off we go!

```
Selecting previously unselected package nessus.
(Reading database ... 111612 files and directories currently installed.)
Preparing to unpack Nessus-10.7.0-ubuntu1404_amd64.deb ...
Unpacking nessus (10.7.0) ...
Setting up nessus (10.7.0) ...
HMAC : (Module_Integrity) : Pass
SHA1 : (KAT_Digest) : Pass
SHA2 : (KAT_Digest) : Pass
SHA3 : (KAT_Digest) : Pass
TDES : (KAT_Cipher) : Pass
AES_GCM : (KAT_Cipher) : Pass
AES_ECB_Decrypt : (KAT_Cipher) : Pass
RSA : (KAT_Signature) : RNG : (Continuous_RNG_Test) : Pass
Pass
ECDSA : (PCT_Signature) : Pass
ECDSA : (PCT_Signature) : Pass
DSA : (PCT_Signature) : Pass
TLS13_KDF_EXTRACT : (KAT_KDF) : Pass
TLS13_KDF_EXPAND : (KAT_KDF) : Pass
TLS12_PRF : (KAT_KDF) : Pass
PBKDF2 : (KAT_KDF) : Pass
SSHKDF : (KAT_KDF) : Pass
KBKDF : (KAT_KDF) : Pass
HKDF : (KAT_KDF) : Pass
SSKDF : (KAT_KDF) : Pass
X963KDF : (KAT_KDF) : Pass
X942KDF : (KAT_KDF) : Pass
HASH : (DRBG) : Pass
CTR : (DRBG) : Pass
```

And after a little:

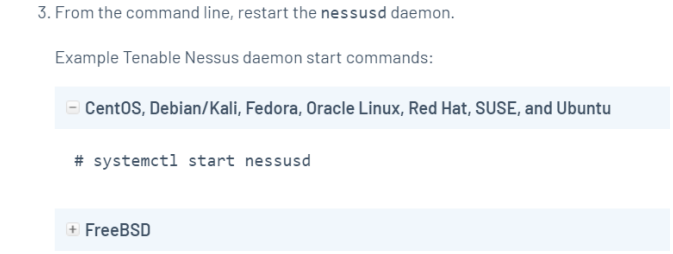
```
SSHKDF : (KAT_KDF) : Pass
KBKDF : (KAT_KDF) : Pass
HKDF : (KAT_KDF) : Pass
SSKDF : (KAT_KDF) : Pass
X963KDF : (KAT_KDF) : Pass
X942KDF : (KAT_KDF) : Pass
HASH : (DRBG) : Pass
CTR : (DRBG) : Pass
HMAC : (DRBG) : Pass
DH : (KAT_KA) : Pass
ECDH : (KAT_KA) : Pass
RSA_Encrypt : (KAT_AsymmetricCipher) : Pass
RSA_Decrypt : (KAT_AsymmetricCipher) : Pass
RSA_Decrypt : (KAT_AsymmetricCipher) : Pass
INSTALL PASSED
Unpacking Nessus Scanner Core Components...
Created symlink /etc/systemd/system/nessusd.service → /lib/systemd/system/nessusd.service.
Created symlink /etc/systemd/system/multi-user.target.wants/nessusd.service → /lib/systemd/system/nessusd.ser

 - You can start Nessus Scanner by typing /bin/systemctl start nessusd.service
 - Then go to https://server1:8834/ to configure your scanner

ajay@server1:~$
```

The installation is complete. However, we're not done yet!

Step 3 of the online Nessus Installation guide says we have to start the service.



Run the command :

```
sudo systemctl start nessusd
```

Hmm, no feedback. Let's check to ensure that its running.

```
systemctl status nessusd
```

IF, and I mean IF everything ran smoothly, this screenshot should be your output.

```
ajay@server1:~$ systemctl status nessusd
● nessusd.service - The Nessus Vulnerability Scanner
     Loaded: loaded (/lib/systemd/system/nessusd.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2024-02-25 17:03:00 UTC; 1min 9s ago
   Main PID: 1623 (nessus-service)
      Tasks: 14 (limit: 9389)
     Memory: 61.3M
        CPU: 29.788s
     CGroup: /system.slice/nessusd.service
             ├─1623 /opt/nessus/sbin/nessus-service -q
             └─1624 nessusd -q

Feb 25 17:03:00 server1 systemd[1]: Started The Nessus Vulnerability Scanner.
Feb 25 17:03:01 server1 nessus-service[1624]: Cached 0 plugin libs in 0msec
Feb 25 17:03:01 server1 nessus-service[1624]: Cached 0 plugin libs in 0msec
```

Step 4

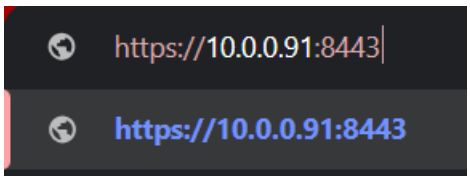
4. Open Tenable Nessus in your browser.

   - To access a remotely installed Tenable Nessus instance, go to https://<remote IP address>:8834 (for example, https://111.49.7.180:8834).
   - To access a locally installed Tenable Nessus instance, go to https://localhost:8834.

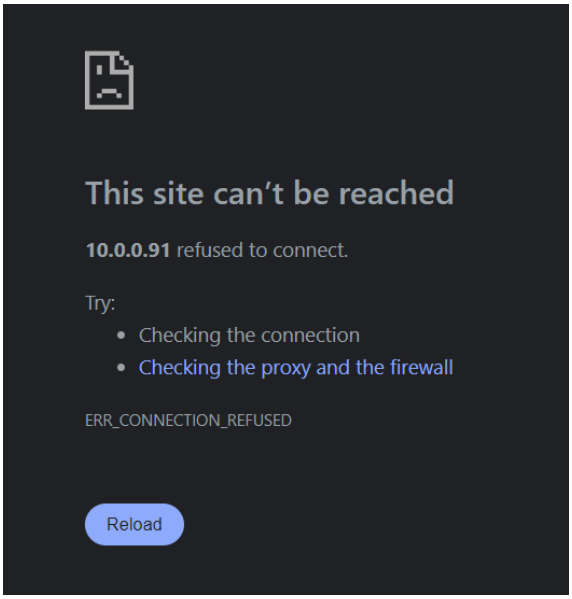
So since we have a locally installed Tenable Nessus instance, we need to go to the local ip of the web interface for the Nessus Server.

PLEASE NOTE that at the end of the URL, they have a colon which means go to another port than the default for HTTPS which is 443. They want us to go to port 8834.

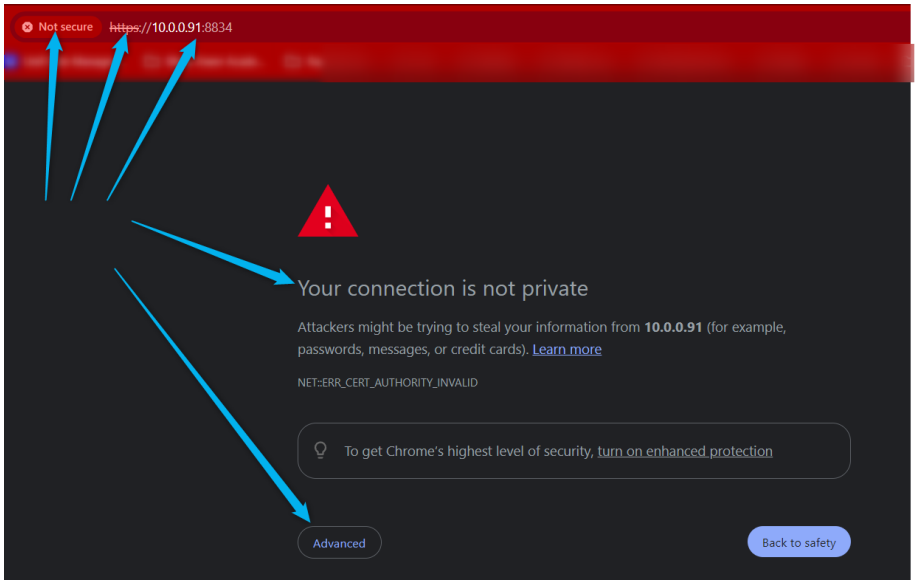
Here is a screenshot within google chrome of what that looks like:



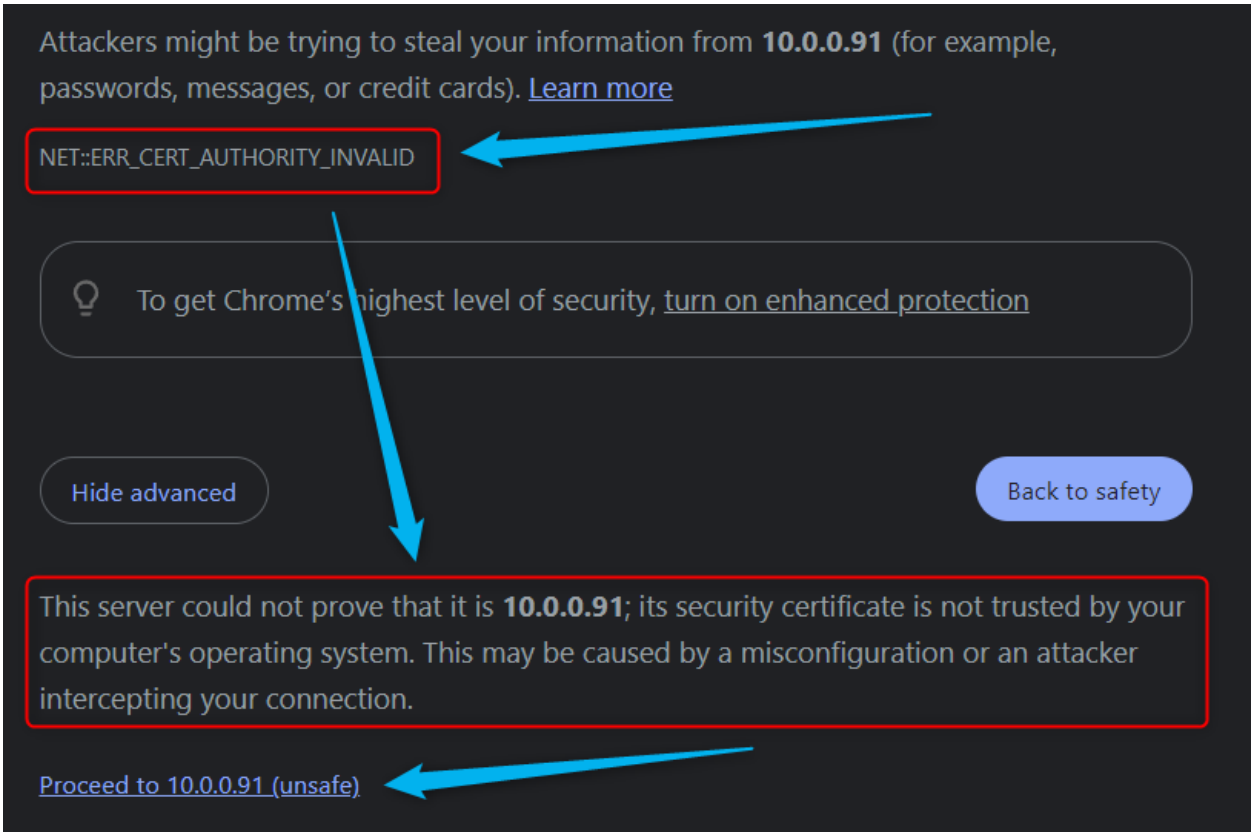
Oops, I made a mistake. I went to the wrong port, I mistyped it. Well, this is what it looks like if you go to the wrong port:



Let's try it again with the correct port:

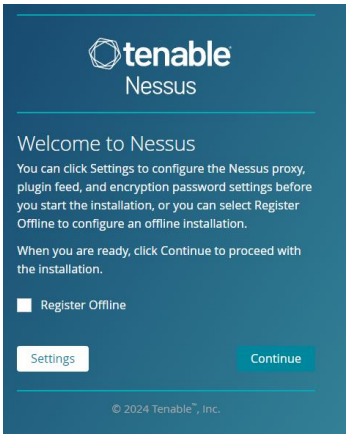


1. This isn't a secure connection even though it was via HTTPS
2. ~~https~~ is struckout
3. it's the right IP and the right port
4. It's letting you know this isn't a PRIVATE connection but it DID connect
5. Let's dive into the advanced tab to see what that provides.



1. The Digital Certificate is not valid. We've done no work to validate this via a PKI.
2. They are providing you feedback about why this error is coming up.
3. We know what we are doing, this is just a test server, were not going to go through the trouble of establishing a fully validated digital certificate.
4. We will proceed past this warning
5. Click on Proceed

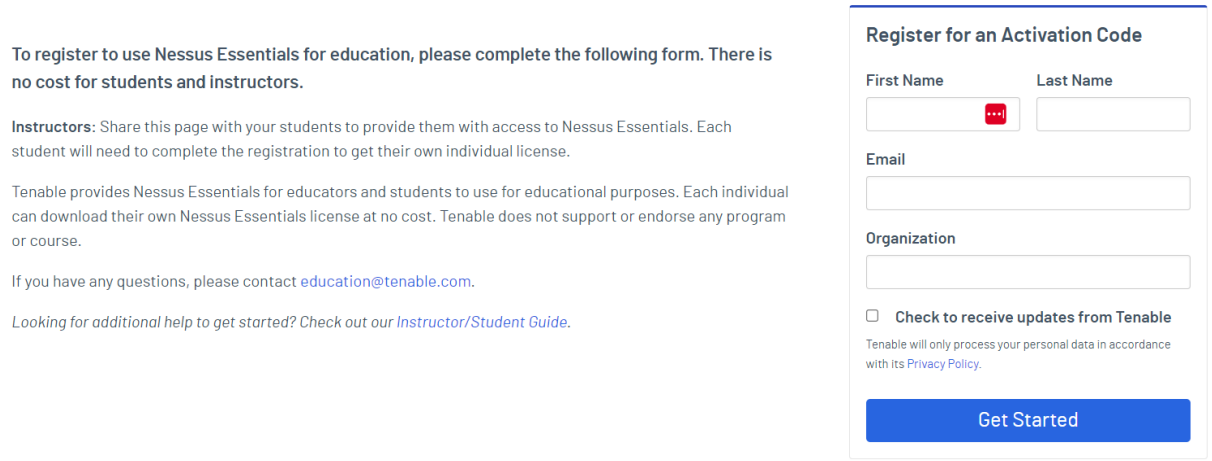
You'll now be presented with the log on splash page for your Vulnerability Scanner which is accessible via a local website.



Before we proceed further, we'll need to register with the Tenable Website to register our "Essentials Edition" which means we can obtain a product key. We cannot use this for commercial purposes, merely educational. That is the restriction of their license. So, use this wisely. If you want to commercially do work, and you don't have the money for a proper commercial license, use the open-source tools.

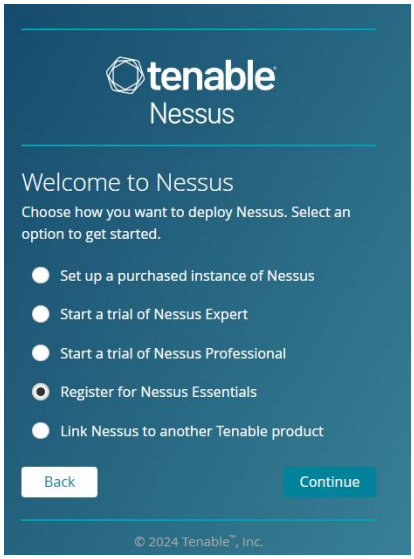
Generate a educational account here: https://www.tenable.com/tenable-for-education/nessus-essentials?edu=true

Fill this out with your Rapid Ascent email alias.



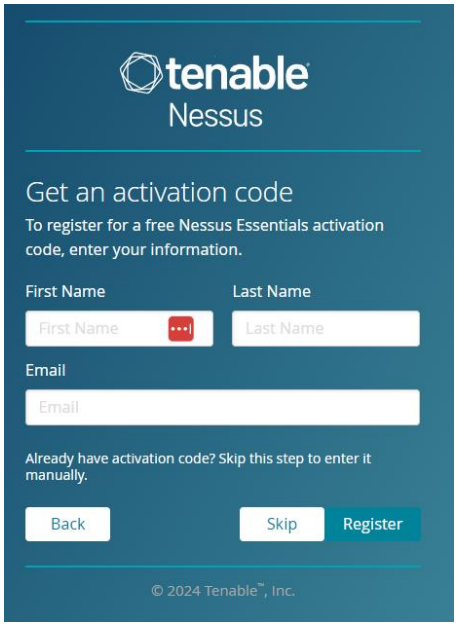
Within 1-2 minutes you should receive the install code.

Click on CONTINUE

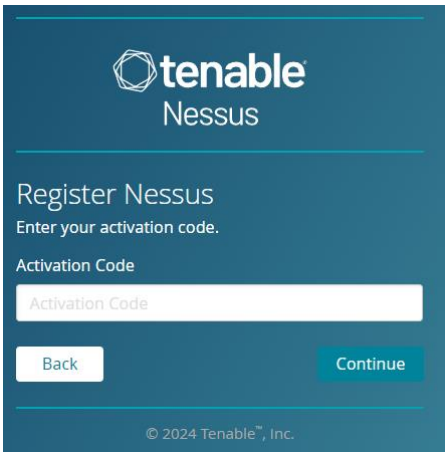


Select the radio button for Nessus Essentials and CONTINUE

If you didn't pre-register, you may do so here, if you already have the Activation CODE, click on SKIP



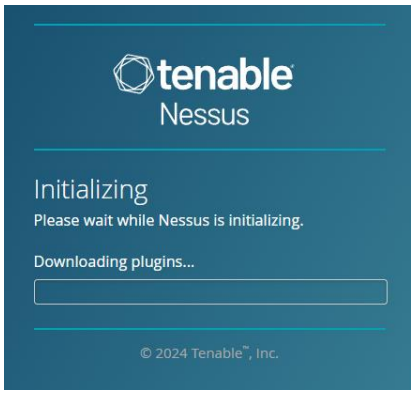
Enter your CODE



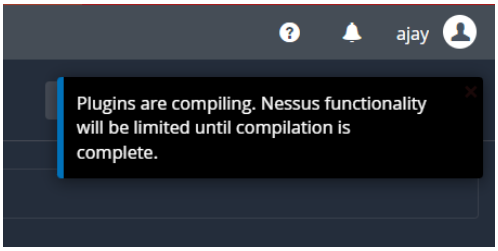
Each account uses a unique activation code, don't try to use anyone else's.

Next you'll need to configure a local Nessus Server User Account. This is what you'll use to log into Nessus, please don't lose these credentials.

Next it must complete the install and download all the things, so this part might take some time, be patient. Find something else to do.

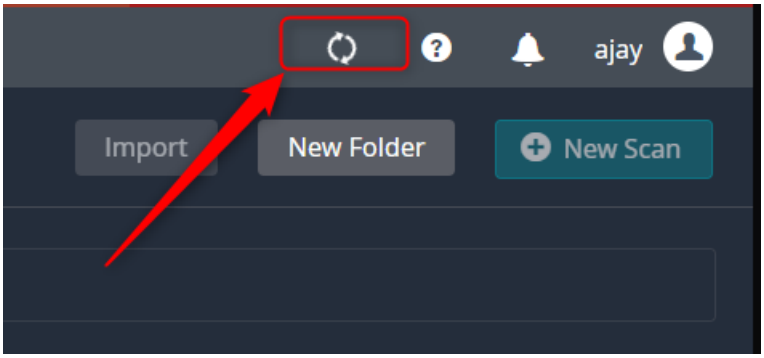


You'll be redirected to the home page, but it is STILL working in the background.

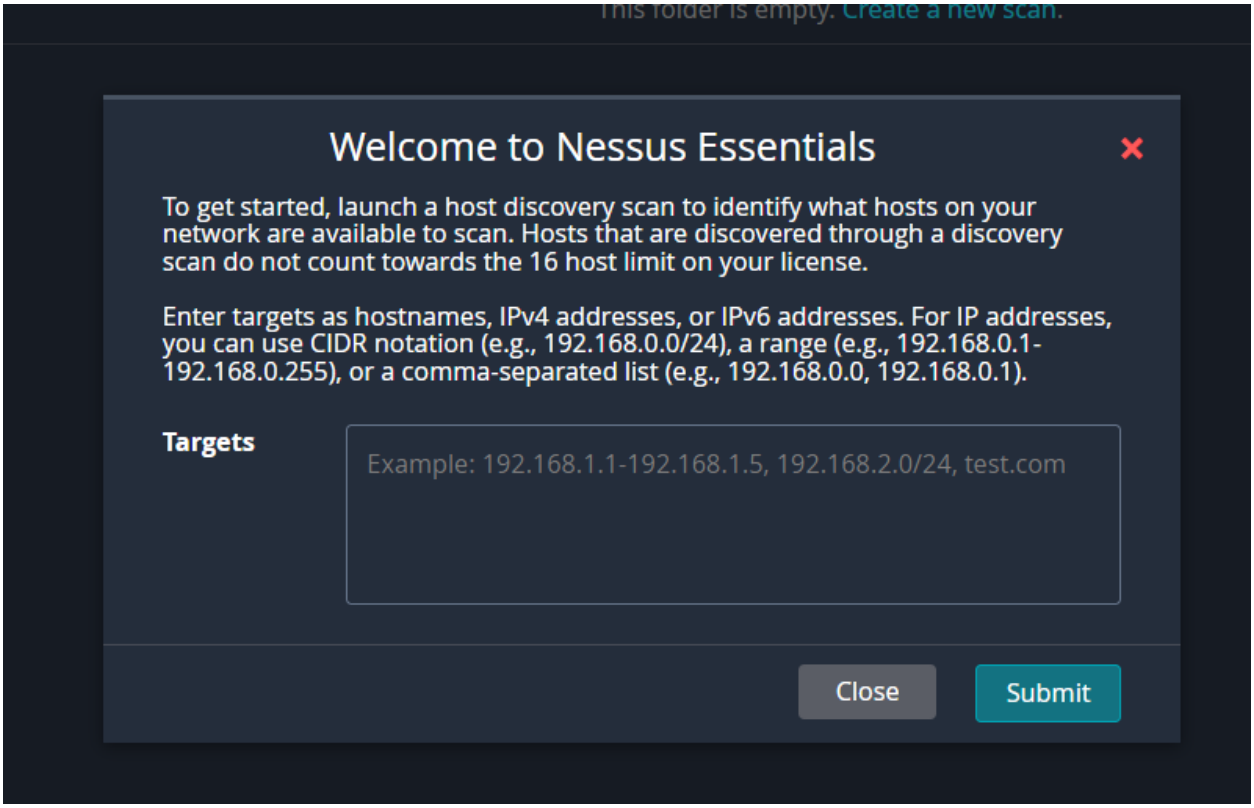


You'll see the little thing spinning, so that means it's downloading all the CVE definitions, which is MASSIVE, this might take 20-30-60 minutes. Be patient. Find something else to do.

Because initializing the definition databases takes so long, most enterprises ALWAYS leave their Vuln Scanners running. That isn't realistic for our little "home lab" situation, so just let it do its thing.



When you get this pop up, you're Nessus Tenable Vulnerability Scanner is installed.



CONGRATULATIONS, you've fully deployed your first CyberSecurity tool on a Linux Server. CLI only!

APPENDIX

Educational Resources

Tenable Nessus Education “Fundamentals Course” $250 (Not required, just FYI)

https://www.tenable.com/education/courses/nessus-fundamentals?ttrp104365=ttrp080188

Tenable Nessus Documentation “Get Started” (Free)

https://docs.tenable.com/nessus/10_7/Content/GetStarted.htm

Tenable Nessus Video “Introduction to Nessus” (Free)

https://www.tenable.com/blog/an-introduction-to-nessus-the-video

Tenable Nessus Architecture

https://www.tenable.com/blog/choosing-the-right-architecture-for-your-nessus-agent-deployment

Tenable Nessus Web App Testing

https://www.tenable.com/blog/tips-for-using-nessus-in-web-application-testing