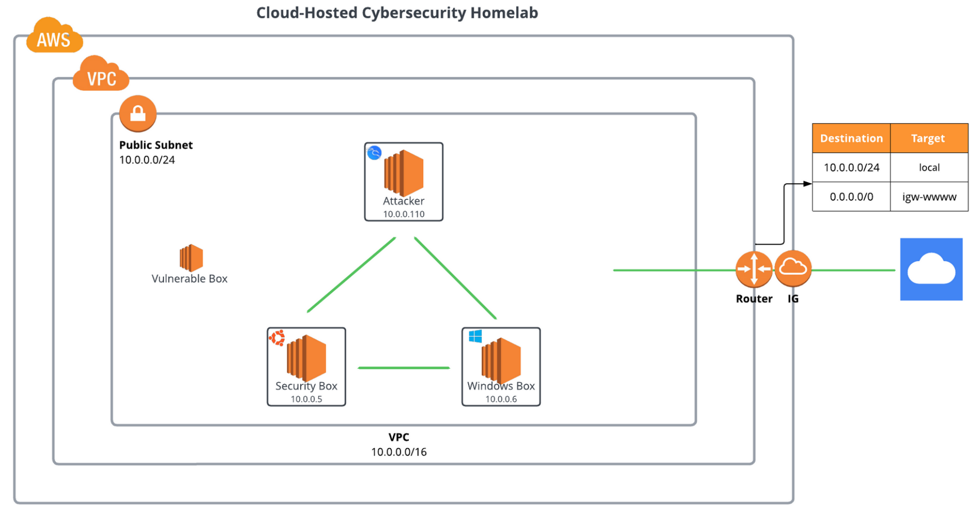
Guide

Network Topology



Destination	Target
10.0.0.0/24	local
0.0.0.0/0	igw-wwww

AWS

Use us-east-2 region.

Create VPC

Create VPC, with one public subnet, route table, and IGW.

- VPC IP Address Block: 10.0.0.0/16 (default)
- Public Subnet - Cybersecurity Homelab: 10.0.0.0/24
- Route Table - Cybersecurity Homelab: Default
- IGW: Default

Key Pair

Create a new key pair in the `.pem` format. Make sure you are in the right region.

Store in the `~/.ssh` directory within Ubuntu 22.04 LTS on Windows Desktop.

Change key's permissions: `chmod 600 {mykeyname.pem}`.

Security Groups

Cybersecurity Homelab SG:

- All ICMP - IPv4 - 0.0.0.0/0
- RDP - Your IP Address
- SSH - Your IP Address

Security Tools - Netspectrum SG:

- Use default provided by the AMI.
- Custom TCP - 9997 - 0.0.0.0/0

Kali Linux VM

-&BuvN4fApX(8%IsMr0LQRUK=GTXdEPj

AWS Configuration

AMI: Kali Linux AMI by Offensive Security.
Instance Type: `t2.micro`.
Volume Size: `EBS` with 12 GIB.

Deploy in Public Subnet Cybersecurity Homelab. Use the Cybersecurity Homelab SG.

XRDP

Resource: https://www.kali.org/docs/general-use/xfce-with-rdp/

The configuration file (for changing the default port) is in `/etc/xrdp/xrdp.ini`.

Create Script

```
#!/bin/sh
echo "[i] Updating and upgrading Kali (this will take a while)"
apt-get update
apt-get full-upgrade -y

echo "[i] Installing Xfce4 & xrdp (this will take a while as well)"
apt-get install -y kali-desktop-xfce xorg xrdp

echo "[i] Configuring xrdp to listen to port 3389 (but not starting the service)"
sed -i 's/port=3389/port=3389/g' /etc/xrdp/xrdp.ini
```

Install

Run `./remote.sh`

Change Password

```
echo kali:kali | sudo chpasswd
```

Start XRDP

```
sudo systemctl enable xrdp --now
```

```
systemctl status xrdp
```

Windows (Vulnerable Client)

AWS Configuration

AMI: Windows Server 2022 Base.
Instance Type: `t2.micro`.
Volume Size: `EBS` with 30 GIB.

Login

Go to the Connect option in AWS, choose RDP client, go to "Get Password", upload the private key associated with the EC2 instance, the private key should be in `.pem` format. Take note of the password, copy to Notepad for notes.

💡 Turn off Windows Defender Firewall to have Kali access Windows.

Security Tools (Ubuntu Desktop)

AWS Configuration

AMI: Use the AMI provided by Netspectrum.
Instance Type: `t3.large`
Volume Size: `EBS` with 30 GIB.

Default Username/Password on Netspectrum

Username: `ubuntu`
Password: EC2 Instance ID

Go to public IP address of the instance in browser.

Change VNC password using the command line shortcut on Desktop.

Change SSH Password

`sudo nano /etc/ssh/sshd_config`.

Change `PasswordAuthentication` to `yes`.

Change the `ubuntu` user's password with `passwd`, the default password is the EC2 instance ID.

Splunk

Download Splunk Enterprise

Resource: https://www.inmotionhosting.com/support/security/install-splunk/#setup

Sign-up / Login into Splunk. Navigate to URL: https://www.splunk.com/en_us/download/splunk-enterprise

Download the `deb` version of Splunk URL for the Ubuntu instance.

Download: `sudo dpkg -i splunk-deb`

Navigate to splunk directory: `/opt/splunk/bin`

Start splunk and create a user. `sudo ./splunk start`.

Create an Splunk admin and username.

Log into dashboard.

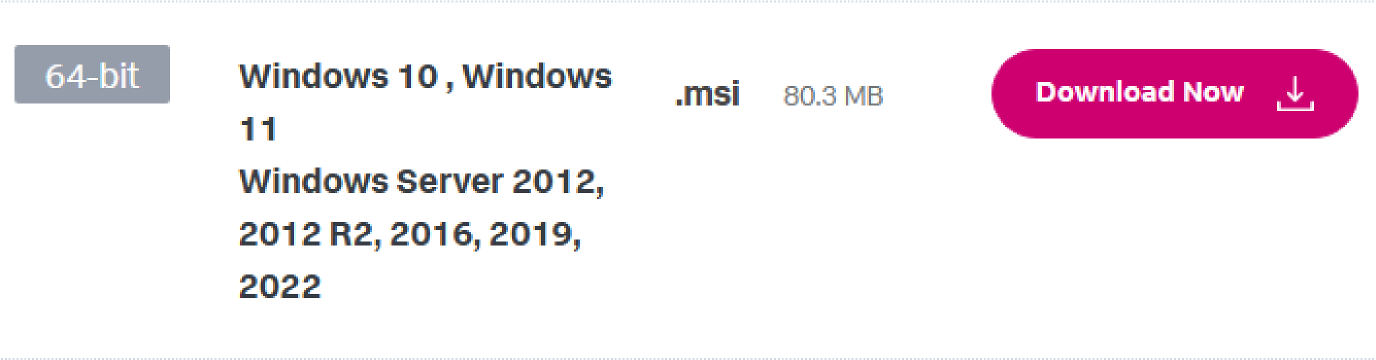
Download Universal Forwarder On Windows

Resource: https://geek-university.com/install-a-splunk-forwarder-on-windows/

Go to Splunk Enterprise -> Settings -> Forwarding and receiving -> Configure receiving -> Ensure "Listen on this port" has 9997. If not, add 9997 as a listener.

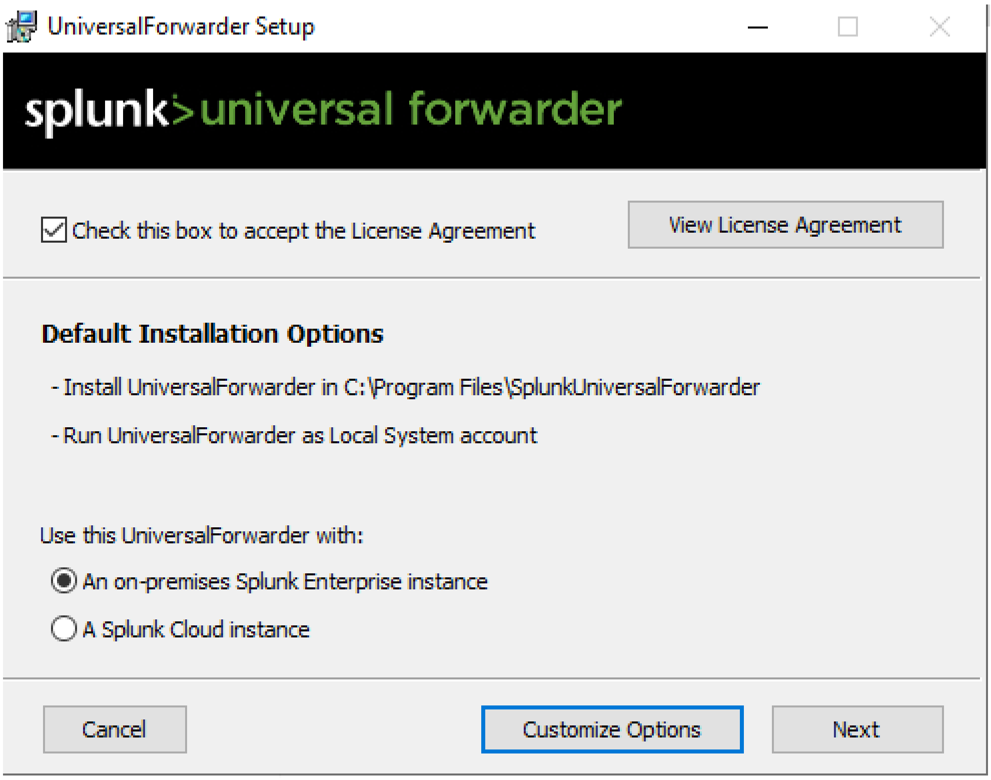
Go to Microsoft Edge, type in the following URL: https://www.splunk.com/en_us/download/universal-forwarder/

Log into Splunk. Navigate to the Windows Server download.



64-bit	Windows 10 , Windows 11 Windows Server 2012, 2012 R2, 2016, 2019, 2022	.msi	80.3 MB	Download Now

Navigate to the Downloads Folder. Click on the `.msi` file and follow Wizard options.



UniversalForwarder Setup

splunk>universal forwarder

☑ Check this box to accept the License Agreement

Default Installation Options

- Install UniversalForwarder in C:\Program Files\SplunkUniversalForwarder
- Run UniversalForwarder as Local System account

Use this UniversalForwarder with:

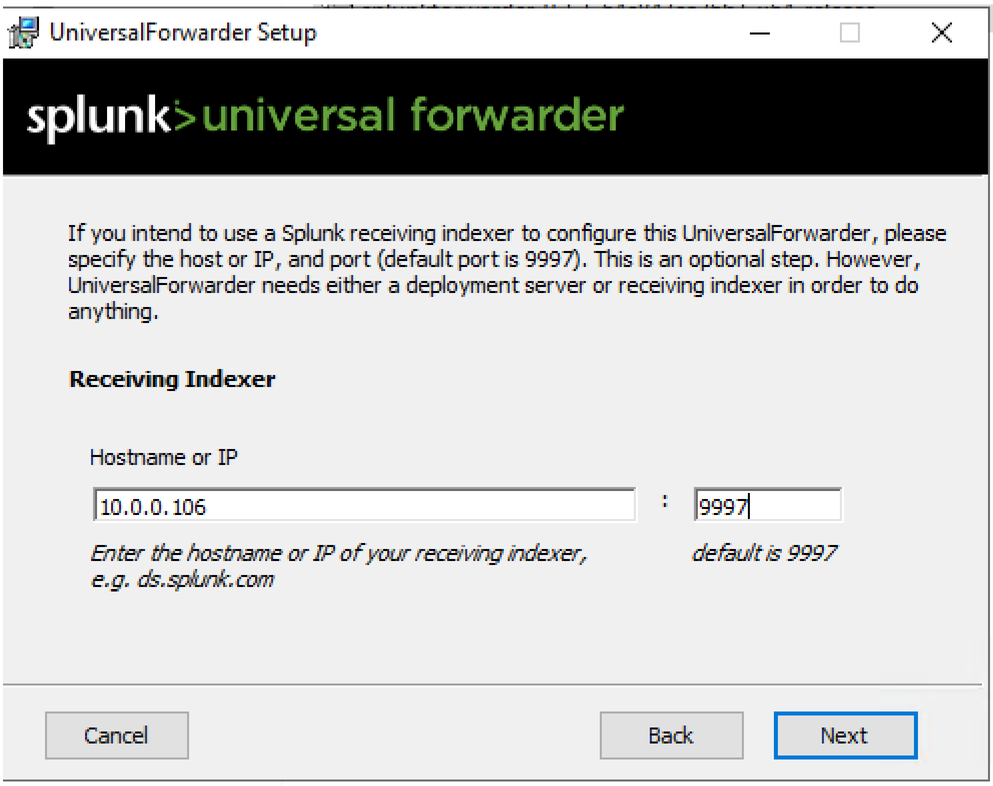
● An on-premises Splunk Enterprise instance

○ A Splunk Cloud instance

Run through default options until getting to the "Windows Events Logs" view. Select "Application", "Security", and "System" log types.

Create an administrator account. You can use the same credentials for the Splunk Enterprise Dashboard.

Skip "Deployment Server" view.



UniversalForwarder Setup

splunk>universal forwarder

If you intend to use a Splunk receiving indexer to configure this UniversalForwarder, please specify the host or IP, and port (default port is 9997). This is an optional step. However, UniversalForwarder needs either a deployment server or receiving indexer in order to do anything.

Receiving Indexer

Hostname or IP

10.0.0.106 : 9997

Enter the hostname or IP of your receiving indexer, e.g. ds.splunk.com

default is 9997

Add the private IP address of Ubuntu Security Tools and use the default port of 9997.

Select "Install".

Add Security Event Logs to win-security Index

Go to Splunk Enterprise in the Security Tools box. Navigate to Settings -> Indexes -> New Index -> Name Index "win-security" -> Use default options.

Go to Windows VM. Navigate to `C:\Program Files\SplunkUniversalForwarder\etc\system\local`.

Create a new file (you can copy the `outputs.conf` file to keep the same format type). Rename to `inputs.conf`.

Add the following to the `inputs.conf`.

```
[WinEventLog://Security]
index = win-security
disabled = 0
```

Open CMD. Navigate to `cd C:\Program Files\SplunkUniversalForwarder\bin`. Type `splunk.exe restart` to restart Splunk service with new settings. Wait until the CMD output state "Done!".

Go to Splunk Enterprise -> Search & Reporting -> Add `index="win-security"`.

Tenable Nessus

Download Tenable Nessus version Ubuntu - amd64: https://www.tenable.com/downloads/nessus?loginAttempted=true

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

Start Nessus: `sudo systemctl start nessusd.service`

Navigate to the URL: `http://localhost:8834`. Choose Nessus Essentials -> Register with a new account -> Set a username/password for the console -> Wait for plugins to setup.