Self-Hosted Cloud Infrastructure for Adaptive Offensive Security Research:

Achieving Adaptability, Security, and Accessibility with Proxmox and Tailscale in a Virtualized Penetration Testing Environment

05/27/25

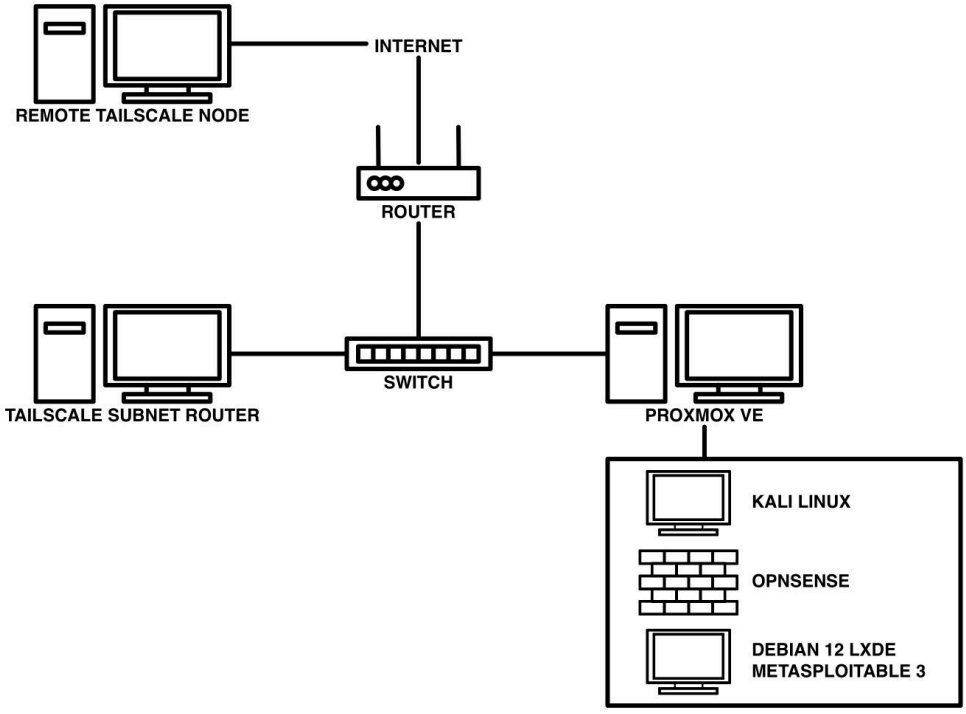
Cuong Huy Nguyen

Abstract

This project investigates the development of a minimal-footprint, self-hosted cloud infrastructure optimized for offensive security research. The lab is designed with three primary objectives: adaptability to evolving research needs, secure remote accessibility for flexible operation, and strong security controls—all while minimizing hardware resource consumption.

The environment leverages a Proxmox-based virtualization stack to efficiently host key components across constrained hardware. A Kali Linux virtual machine provides tooling for offensive operations, including penetration testing, malware analysis, and open-source intelligence (OSINT) gathering. Defensive capabilities are delivered through an OPNsense firewall with integrated IDS/IPS functionality powered by Suricata. Additionally, a Debian 12 LXDE virtual machine is used to host nested Metasploitable 3 instances—Windows Server 2008 and Ubuntu 14.04—emulating vulnerable systems for exploit development and scenario-based testing.

Remote connectivity is securely facilitated via Tailscale, configured as a subnet router, allowing controlled external access without compromising lab isolation. This infrastructure offers a scalable, modular, and secure platform for advanced cybersecurity experimentation and research.



Hardware Overview:

The self-hosted cloud infrastructure consists of two primary hardware systems, each serving a specific role in supporting the virtualized environment and ensuring secure remote access.

Proxmox Machine (Lab Host, Intel N100, 16GB RAM, 512GB SSD):
The Proxmox machine functions as the central virtualization host, running Proxmox VE to manage and orchestrate virtualized environments for penetration testing. Equipped with high-performance hardware, the Proxmox machine provides the necessary computational power for resource-intensive simulations and research tasks, serving as the backbone of the offensive security lab.

Hosting Machine (Windows 11 Pro Desktop, Tailscale Subnet Router):
The hosting machine is a Windows 11 Pro desktop, configured as a Tailscale subnet router to enable secure remote access to the Proxmox environment. Utilizing Tailscale's mesh VPN technology, this machine facilitates encrypted and authenticated connections, ensuring seamless remote access to the Proxmox lab from external networks. It provides a secure gateway, maintaining the isolation of the virtualized lab while enabling flexible connectivity for remote research.

Together, these two machines form the core infrastructure for adaptive offensive security research, with the Proxmox machine hosting the virtualized environments and the Windows 11 Pro desktop ensuring secure, remote access via Tailscale.

Lab Configuration Overview:

The Proxmox-based self-hosted cloud infrastructure is configured with three primary VM templates, each designed to support a range of penetration testing scenarios. These templates are converted into reusable clones, allowing for dynamic, on-demand simulation of various offensive security research environments.

Kali Linux Template:
The Kali Linux VM is pre-configured with a comprehensive suite of penetration testing tools, providing a versatile platform for simulating attack vectors and performing exploitations within the environment.

OPNsense Template:
The OPNsense VM serves as a network security appliance, simulating firewall and network configurations. It facilitates the testing of network segmentation, intrusion detection, and firewall bypass techniques, supporting a wide range of security-related scenarios.

Debian 12 LXDE Template with Nested Virtualization (Metasploitable 3):
The Debian 12 LXDE VM hosts two nested virtualized Metasploitable 3 VMs—one running Ubuntu 14.04 and the other running Windows Server 2008. These vulnerable systems serve as targets for testing exploitation and post-exploitation activities. The nested virtualization setup allows for the simulation of multi-tier attack scenarios, where multiple layers of exploitation can be tested and analyzed in isolation.

Each VM template is structured to be cloned and configured as needed, enabling efficient and flexible testing of diverse penetration testing scenarios within the infrastructure.

Prerequisite Knowledge:

Before proceeding with the setup and configuration of the self-hosted cloud infrastructure, it is essential to have a solid understanding of the following concepts and tasks. Familiarity with basic networking principles (IP address, subnet mask, DNS, and DHCP) as well as the following will ensure the successful creation, management, and operation of virtual environments in Proxmox, as well as enabling secure remote access via Tailscale.

- **Proxmox Installation and Initial Configuration:** A fundamental understanding of Proxmox VE installation and initial configuration is necessary to set up the virtualization environment. This includes installing Proxmox on a dedicated machine, configuring network settings, and ensuring proper system resources are allocated for optimal performance (Proxmox, n.d.).
- **Proxmox Nested Virtualization:** For enabling nested virtualization in Proxmox, the official documentation provides detailed instructions for both Intel and AMD processors (Proxmox, n.d.). This setup allows you to run virtual machines inside a VM, which is necessary for environments like penetration testing labs or scenarios involving nested virtualized systems. The guide ensures proper configuration for both Intel and AMD CPUs, allowing you to enable virtualization features and optimize Proxmox for these advanced setups.
- **Creating and Managing VMs in Proxmox:** Understanding how to create and configure virtual machines (VMs) within Proxmox is critical (Proxmox, n.d.). This includes selecting appropriate OS images, configuring hardware settings (e.g., CPU, RAM, storage), and managing VM lifecycle (e.g., starting, stopping, and modifying VMs).
- **Creating and Using VM Templates in Proxmox:** Proficiency in converting a configured VM into a template is essential (Proxmox, n.d.). A template allows for the efficient creation of linked-clones, enabling rapid deployment of multiple identical environments. Understanding how to create, modify, and manage templates will streamline the workflow.
- **Linked Cloning from a VM Template in Proxmox:** Knowledge of linked-clones is necessary to efficiently clone VM templates (Proxmox, n.d.). Linked-cloning allows for quick instantiation of VMs that share the base template, saving disk space while maintaining easy management of multiple similar VMs for testing and research.
- **Kali Linux Installation:** Kali Linux is a widely-used penetration testing and ethical hacking platform. Familiarity with the official installation process of Kali Linux will allow for an efficient setup within the Proxmox environment (Kali Linux, n.d.).
- **OPNsense Installation:** OPNsense is an open-source firewall and routing platform. Understanding the official installation process will enable its deployment as a network security appliance within the lab (OPNsense, n.d.).
- **Debian 12 LXDE Installation:** Debian 12 is the base operating system for one of the primary virtual machines in the lab. Familiarity with Debian's installation process and LXDE desktop environment ensures a consistent setup (Debian, n.d.).
- **Metasploitable 3 Setup:** Metasploitable 3 is an intentionally vulnerable virtual machine used for penetration testing exercises. Proficiency in setting up Metasploitable 3 will ensure that the vulnerable targets are ready for exploitation within the lab environment (rapid7, n.d.).
- **Tailscale Installation and Configuration (Subnet Routing):** Tailscale provides secure, encrypted, and easy-to-set-up VPN connections between devices. Knowledge of Tailscale installation on a Windows machine (Tailscale, n.d.) and configuration as a subnet router (Tailscale, n.d.) is essential for enabling secure remote access to the Proxmox environment. This includes configuring the machine to act as a VPN gateway, facilitating seamless, secure access to the lab from external networks.

References

- Debian. (n.d.). Installation guide for arm64. Debian. https://www.debian.org/releases/stable/arm64/index.en.html
- Kali Linux. (n.d.). Installation. Kali Linux Documentation. https://www.kali.org/docs/installation/
- OPNsense. (n.d.). Installation. OPNsense Documentation. https://docs.opnsense.org/manual/installation.html
- Proxmox. (n.d.). Installation. Proxmox Documentation. https://pve.proxmox.com/wiki/Installation
- Proxmox. (n.d.). Linked clones. In Proxmox VE administration guide. Proxmox Documentation. https://pve.proxmox.com/wiki/Proxmox_VE_Administration_Guide#Linked_Clones
- Proxmox. (n.d.). Nested virtualization. Proxmox Documentation. https://pve.proxmox.com/wiki/Nested_Virtualization
- Proxmox. (n.d.). Templates. In Proxmox VE administration guide. Proxmox Documentation. https://pve.proxmox.com/wiki/Proxmox_VE_Administration_Guide#Templates
- Proxmox. (n.d.). Virtual machines. Proxmox Documentation. https://pve.proxmox.com/wiki/Virtual_Machines
- rapid7. (n.d.). Metasploitable3. GitHub. https://github.com/rapid7/metasploitable3
- Tailscale. (n.d.). Install Tailscale. Tailscale Documentation. https://tailscale.com/kb/1017/install/
- Tailscale. (n.d.). Subnet routers and traffic relay nodes. Tailscale Documentation. https://tailscale.com/kb/1105/subnets/

Environment-Specific Configurations

Enable Nested Hardware-Assisted Virtualization on Proxmox
https://pve.proxmox.com/wiki/Nested_Virtualization

Verify Nested Virtualization:

- Access the Proxmox Web GUI and open the Shell for the host node.
- Check if nested virtualization is enabled (Y/N for Intel and 1/0 for AMD):
    - For Intel:
      ```
      cat /sys/module/kvm_intel/parameters/nested
      ```
    - For AMD:
      ```
      cat /sys/module/kvm_amd/parameters/nested
      ```

Enable Nested Virtualization (if not enabled)

- Run the following commands:
    - For Intel:
      ```
      echo "options kvm-intel nested=Y" >
      /etc/modprobe.d/kvm-intel.conf
      ```
    - For AMD:
      ```
      echo "options kvm-amd nested=1" >
      /etc/modprobe.d/kvm-amd.conf
      ```
- Reload the kernel module:
    - For Intel:
      ```
      modprobe -r kvm_intel
      modprobe kvm_intel
      ```
    - For AMD:
      ```
      modprobe -r kvm_amd
      modprobe kvm_amd
      ```
- Verify again

Debian 12 LXDE Installation

- Create a new VM in Proxmox and select Debian 12 as the OS.
- Name the VM and configure the disk size to 150GB.
- Set the memory to 8192MB (8GB) and allocate 2 CPU cores with the "host" CPU type.
- For networking, choose a bridged network (e.g., vmbr0).
- Mount the Debian 12 LXDE ISO image to the VM.
- Start the VM and proceed with the Debian installation.
- Select LXDE as Desktop Environment

Metasploitable 3 VMs Installation

- Install Virtualbox's dependencies via Debian 12 LXDE's CLI

```
sudo apt update
sudo apt install -y \
  dkms \
  build-essential \
  linux-headers-$(uname -r) \
  qt5-qmake \
  qttools5-dev \
  qttools5-dev-tools \
  libqt5x11extras5 \
  libvncserver1 \
  libx11-dev \
  libsdl1.2-dev \
  libpng-dev \
  libssl-dev \
  libgtk-3-dev \
  libvpx-dev
```

- Navigate to Downloads directory
- Download .deb file from https://www.virtualbox.org/wiki/Linux_Downloads
- Install the .deb file

```
sudo dpkg -i virtualbox-7.0_*.deb
sudo apt --fix-broken install -y
```

- Navigate back to your home directory
- Install Vagrant by following the instructions from https://developer.hashicorp.com/vagrant/install
- Run the following commands to initialize Metasploitable 3 VMs. There are three lines of code (the `curl` command is one line all the way to …`/master/Vagrantfile`).

```
mkdir metasploitable3-workspace
cd metasploitable3-workspace
curl -O
https://raw.githubusercontent.com/rapid7/metasploitable3/master
/Vagrantfile
```

- `vagrant up` and finish setting up/troubleshooting each VM before starting the other. This will help us reduce the scope of resource utilization as well as troubleshooting. There will be some errors initially. That is normal.
  - For Ubuntu Server VM:

```
vagrant up ub1404
```

  - For Windows Server VM:

```
vagrant up win2k8
```

- For ub1404 VM, during your first launch, you will encounter the following error:

```
The IP address configured for the host-only network is not
within the
allowed ranges. Please update the address used to be within the
allowed
ranges and run the command again.
```

- This is normal and caused by a newer security feature governing host-only networking in Virtualbox. You can fix it by:
  - Create `networks.conf` file in `/etc/vbox/` by running:

```
sudo nano /etc/vbox/networks.conf
```

    - Enter the IP range that includes ub1404's IP shown in the CLI, or simply enter all private IP ranges:
        * `192.168.0.0/16`
        * `10.0.0.0/8`
        * `172.16.0.0/12`
- Save the file. `vagrant halt ub1404` to shut it down gracefully. `vagrant up ub1404` again and it should finish configuring on its own.
- win2k8 vm should be a straightforward process. The most likely thing that can happen is by odd luck and bad karma you got a Vagrantfile that didn't specify `win2k8.vm.communicator = "winrm"`. If that's the case, you should `vagrant halt win2k8` to shut it down gracefully. Spend some time talking to your ancestors and then edit the Vagrantfile in the Metasploitable 3 directory to include the following lines:

```
config.vm.define "win2k8" do |win2k8|
  win2k8.vm.box = "rapid7/metasploitable3-win2k8"
  win2k8.vm.hostname = "metasploitable3-win2k8"
  win2k8.winrm.retry_limit = 60
  win2k8.winrm.retry_delay = 10
```

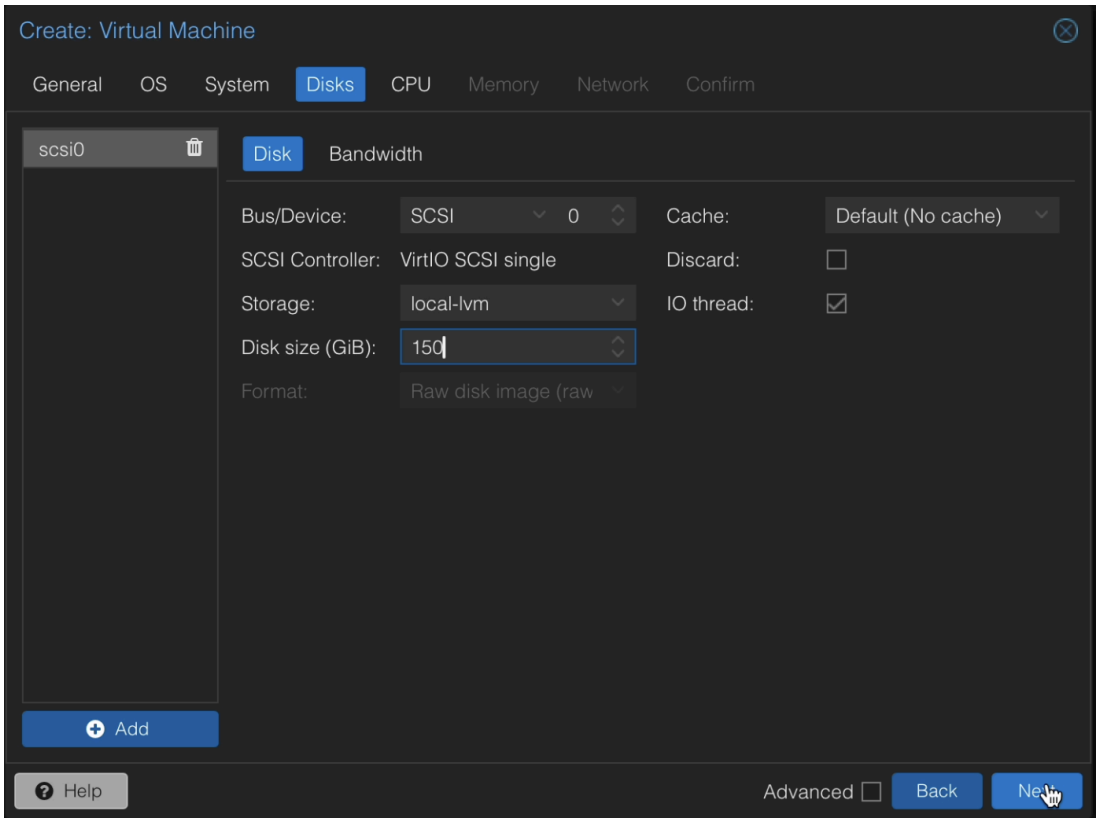
- Now you can thank your ancestors and `vagrant up win2k8` again.
- Alternatively, if you truly want to immerse yourself deeper in compiling and building from source codes, I strongly recommend attempting to build manually. There will be a **LOT** of troubleshooting, but it is a wonderful experience for both technical and mental growth. Here is a good starting base, according to OSINT on software released around Metasploitable 3's initial launch:
    - Debian 10 LXDE
    - Virtualbox 6.x
    - Vagrant 2.3.7
    - Packer 1.8.7

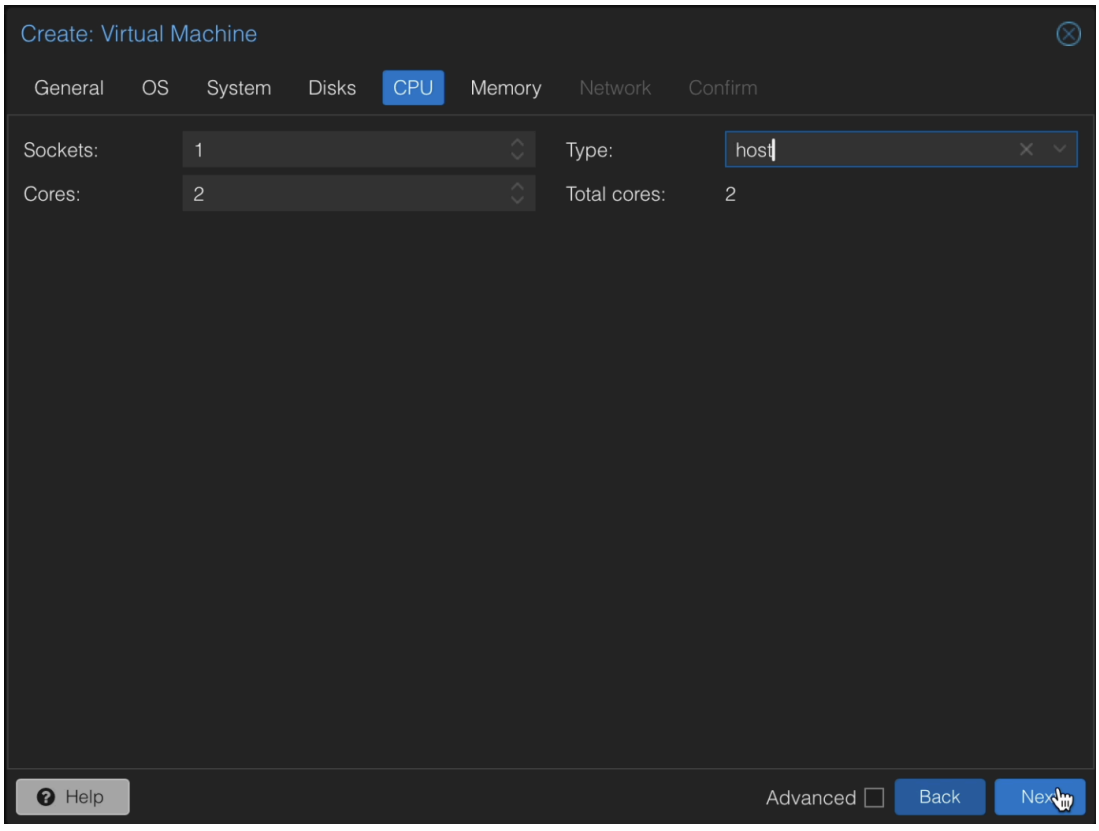
Tailscale Configuration

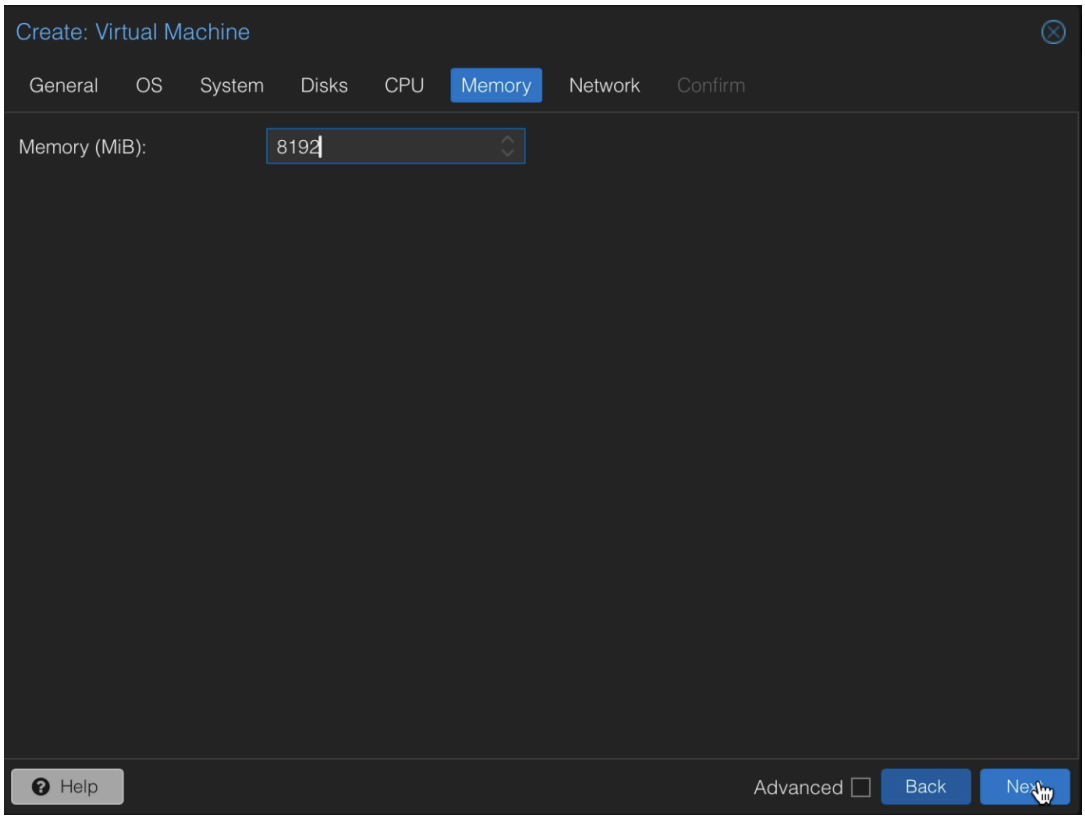
- Tailscale's subnet router feature enables secure remote access to entire local networks using virtually any device capable of processing network traffic. From Raspberry Pis and old laptops to VMs and servers, a wide range of hardware can serve as a subnet router without the need for specialized equipment. This flexibility makes it easy to extend private network access securely and cost-effectively across diverse environments.
- Install Tailscale on your hosting machine following instructions here:
    - https://tailscale.com/kb/1017/install/
- Follow instructions here to setup your subnet router on the hosting machine:
    - https://tailscale.com/kb/1105/subnets/
    - **TL;DR:** find your network IP address, then run the `tailscale set --advertise-routes=xxx.xxx.xxx.xxx/xx` after making OS-specific prerequisite configurations. You should know what this part "`xxx.xxx.xxx.xxx/xx`" means.

Key Visual References

Important Proxmox VM Settings for Debian 12 LXDE and Metasploitable 3



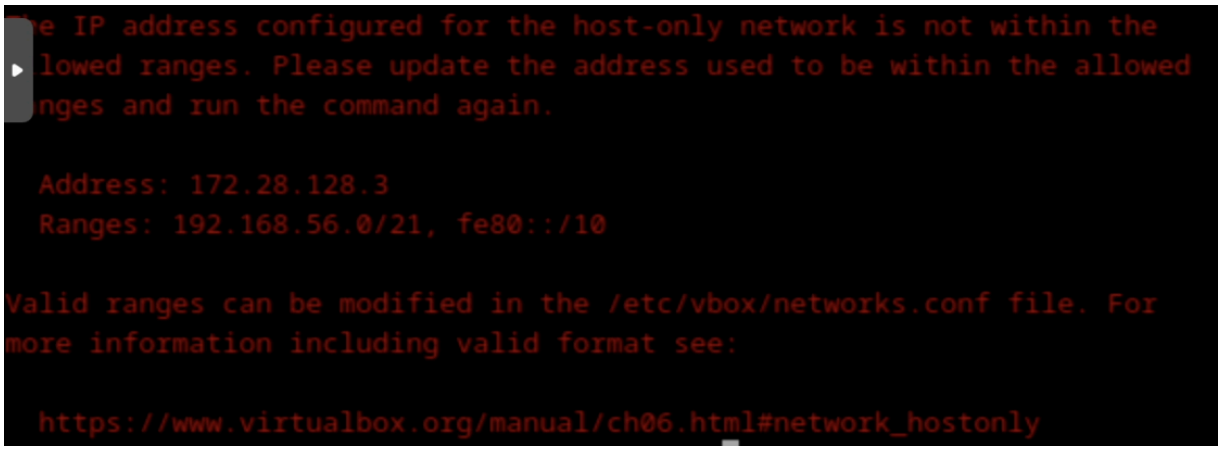



Create: Virtual Machine
General
OS
System
Disks
CPU
Memory
Network
Confirm
Memory (MiB):
8192
Help
Advanced
Back
Next

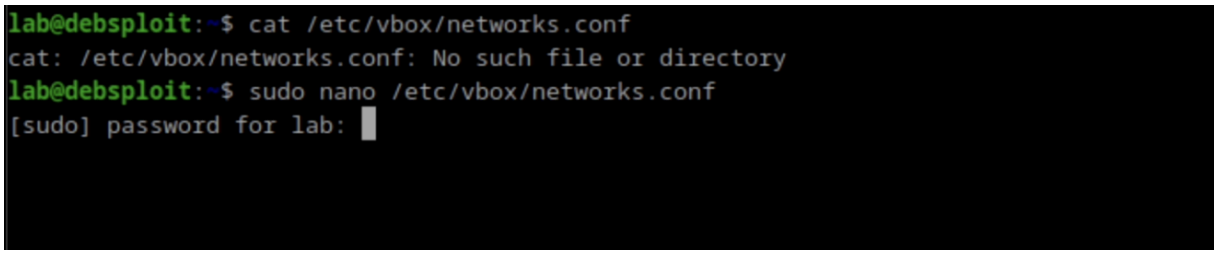
Vagrantfile From Metasploitable3 GitHub

```
lab@debsploit:~/metasploitable3-workspace$ cat Vagrantfile
# -*- mode: ruby -*-
# vi: set ft=ruby :

Vagrant.configure("2") do |config|
  config.vm.synced_folder '.', '/vagrant', disabled: true
  config.vm.define "ub1404" do |ub1404|
    ub1404.vm.box = "rapid7/metasploitable3-ub1404"
    ub1404.vm.hostname = "metasploitable3-ub1404"
    config.ssh.username = 'vagrant'
    config.ssh.password = 'vagrant'

    ub1404.vm.network "private_network", ip: '172.28.128.3'

    ub1404.vm.provider "virtualbox" do |v|
      v.name = "Metasploitable3-ub1404"
      v.memory = 2048
    end
  end

  config.vm.define "win2k8" do |win2k8|
    # Base configuration for the VM and provisioner
    win2k8.vm.box = "rapid7/metasploitable3-win2k8"
    win2k8.vm.hostname = "metasploitable3-win2k8"
    win2k8.vm.communicator = "winrm"
    win2k8.winrm.retry_limit = 60
    win2k8.winrm.retry_delay = 10

    win2k8.vm.network "private_network", type: "dhcp"

    win2k8.vm.provider "libvirt" do |v|
      v.memory = 4096
      v.cpus = 2
      v.video_type = 'qxl'
      v.input :type => "tablet", :bus => "usb"
      v.channel :type => 'unix', :target_name => 'org.qemu.guest_agent.0', :target_type => 'virtio'
      v.channel :type => 'spicevmc', :target_name => 'com.redhat.spice.0', :target_type => 'virtio'
      v.graphics_type = "spice"

      # Enable Hyper-V enlightenments: https://blog.wikichoon.com/2014/07/enabling-hyper-v-enlightenments-with-kvm.html
      v.hyperv_feature :name => 'stimer',  :state => 'on'
      v.hyperv_feature :name => 'relaxed', :state => 'on'
      v.hyperv_feature :name => 'vapic',   :state => 'on'
      v.hyperv_feature :name => 'synic',   :state => 'on'
    end

    # Configure Firewall to open up vulnerable services
    case ENV['MS3_DIFFICULTY']
      when 'easy'
        win2k8.vm.provision :shell, inline: "C:\\startup\\disable_firewall.bat"
      else
        win2k8.vm.provision :shell, inline: "C:\\startup\\enable_firewall.bat"
        win2k8.vm.provision :shell, inline: "C:\\startup\\configure_firewall.bat"
    end

    # Insecure share from the Linux machine
    win2k8.vm.provision :shell, inline: "C:\\startup\\install_share_autorun.bat"
    win2k8.vm.provision :shell, inline: "C:\\startup\\setup_linux_share.bat"
    win2k8.vm.provision :shell, inline: "rm C:\\startup\\*" # Cleanup startup scripts
  end
end
```

ub1404 Host-Only Network Error



`networks.conf` file in `/etc/vbox/`



Successful Configuration of Metasploitable 3 Prebuilt Vagrant Boxes

```
  https://www.virtualbox.org/manual/ch06.html#network_hostonly
lab@debsploit:~/metasploitable3-workspace$ vagrant halt ub1404
lab@debsploit:~/metasploitable3-workspace$ vagrant up ub1404
Bringing machine 'ub1404' up with 'virtualbox' provider...
==> ub1404: Checking if box 'rapid7/metasploitable3-ub1404' version '0.1.12-weekly' is up to date...
==> ub1404: Clearing any previously set network interfaces...
==> ub1404: Preparing network interfaces based on configuration...
    ub1404: Adapter 1: nat
    ub1404: Adapter 2: hostonly
==> ub1404: Forwarding ports...
    ub1404: 22 (guest) => 2222 (host) (adapter 1)
==> ub1404: Running 'pre-boot' VM customizations...
==> ub1404: Booting VM...
==> ub1404: Waiting for machine to boot. This may take a few minutes...
    ub1404: SSH address: 127.0.0.1:2222
    ub1404: SSH username: vagrant
    ub1404: SSH auth method: password
    ub1404: Warning: Connection reset. Retrying...
    ub1404:
    ub1404: Inserting generated public key within guest...
    ub1404: Removing insecure key from the guest if it's present...
    ub1404: Key inserted! Disconnecting and reconnecting using new SSH key...
==> ub1404: Machine booted and ready!
==> ub1404: Checking for guest additions in VM...
    ub1404: No guest additions were detected on the base box for this VM! Guest
    ub1404: additions are required for forwarded ports, shared folders, host only
    ub1404: networking, and more. If SSH fails on this machine, please install
    ub1404: the guest additions and repackage the box to continue.
    ub1404:
    ub1404: This is not an error message; everything may continue to work properly,
    ub1404: in which case you may ignore this message.
==> ub1404: Setting hostname...
==> ub1404: Configuring and enabling network interfaces...
lab@debsploit:~/metasploitable3-workspace$ vagrant halt ub1404
==> ub1404: Attempting graceful shutdown of VM...
lab@debsploit:~/metasploitable3-workspace$ clear
```

```
==> win2k8: Running provisioner: shell...
    win2k8: Running: inline PowerShell script
    win2k8:
    win2k8: C:\Windows\system32>copy C:\vagrant\scripts\installs\setup_linux_share.bat C:
    win2k8: The system cannot find the path specified.
    win2k8:
    win2k8: C:\Windows\system32>reg add HKLM\Software\Microsoft\Windows\CurrentVersion\Ru
    win2k8: The operation completed successfully.
==> win2k8: Running provisioner: shell...
    win2k8: Running: inline PowerShell script
    win2k8: "Linux host not available."
==> win2k8: Running provisioner: shell...
    win2k8: Running: inline PowerShell script
lab@debsploit:~/metasploitable3-workspace$ vagrant halt win2k8
==> win2k8: Attempting graceful shutdown of VM...
lab@debsploit:~/metasploitable3-workspace$
```

Use Cases

Windows Server 2008 Attack Scenarios

EternalBlue – SMB Remote Code Execution (MS17-010)

One of the most infamous vulnerabilities in Windows Server 2008 is EternalBlue, which targets the Server Message Block version 1 (SMBv1) protocol. An attacker can exploit this vulnerability using tools such as Nmap to identify open port 445 and confirm that SMBv1 is active. Using Metasploit's ms17_010_eternalblue module, the attacker sends a specially crafted packet to the vulnerable machine, resulting in remote code execution and a reverse shell session (Microsoft, 2017).

SMB Null Session Enumeration and Credential Abuse

Null sessions, a legacy flaw in SMB services, allow unauthenticated users to enumerate usernames and shared resources. Tools like enum4linux can be used to extract this information. Following enumeration, attackers use brute-force tools such as Hydra or CrackMapExec to obtain credentials. With valid credentials, Metasploit's psexec module can be deployed to upload and execute a service payload, granting SYSTEM-level access (Brenton & Hunt, 2001).

RDP Brute-force or Weak Credential Access

Remote Desktop Protocol (RDP), typically found on port 3389, is another vector for unauthorized access. Attackers may use Nmap for detection, followed by brute-force attempts using Hydra or Ncrack. If credentials are weak or default, they can successfully log in via rdesktop or xfreerdp to gain full graphical user interface (GUI) control with administrative rights (Skoudis & Liston, 2006).

Ubuntu 14.04 Attack Scenarios

VSFTPD 2.3.4 Backdoor Access

VSFTPD version 2.3.4 was compromised with a backdoor that activates when a username ending in :) is used during login. Upon triggering, it opens a shell on port 6200. Attackers can exploit this using Metasploit's vsftpd_234_backdoor module to gain shell access to the Ubuntu system (CVE-2011-2523, 2011).

Apache Tomcat Manager WAR File Upload

When Apache Tomcat Manager is exposed (commonly on port 8080) with default credentials, attackers can access the interface and upload a malicious .war file. Using Metasploit’s tomcat_mgr_upload module, a reverse shell payload can be deployed, granting remote access to the attacker (The Apache Software Foundation, 2009).

Unrestricted File Upload in Web Application

Many legacy web applications fail to validate file types on upload. Attackers can use tools like Nikto and Burp Suite to identify these vulnerabilities and then upload a reverse shell disguised as an image. When accessed via a browser, the server executes the script, providing shell access (OWASP, 2013).

Burner Nodes and Proxy Chains Using Tailscale

Burner nodes are temporary, disposable systems used to hide an operator’s identity at the beginning of a cyber operation (Clark & Landau, 2010). These nodes can be enrolled into a Tailscale mesh VPN, which uses the WireGuard protocol to create encrypted, peer-to-peer connections without exposing

public IP addresses (Tailscale Inc., 2023). Once inside the network, attackers establish a proxy or pivot chain across other connected devices, such as cloud VMs or compromised hosts. This allows for stealthy movement through infrastructure while maintaining encryption and bypassing traditional firewalls. The technique combines strong OPSEC practices with modern networking to improve anonymity, control, and detection evasion (Hoglund & McGraw, 2005). While powerful, it still requires careful operational hygiene to avoid traceability.

Kali Nmap Scanning vs. OPNsense Suricata IDS/IPS Study Case

Kali Linux's Nmap is commonly used to perform active network reconnaissance by scanning open ports and services on target networks. In a defensive environment, an OPNsense firewall integrated with Suricata IDS/IPS monitors this traffic in real time. When Kali's Nmap scans initiate SYN and other probe packets, Suricata analyzes the patterns against its signature database and behavioral rules. Suricata can detect aggressive scanning behavior, generating alerts or blocking the source IP based on predefined thresholds. This interaction reflects real-world events such as the reconnaissance phase of the 2017 Equifax breach, where network scanning activity was detected by IDS/IPS tools, highlighting the critical role of layered defenses in identifying and mitigating early-stage intrusions (Smith & Doe, 2018; OPNsense Community, 2022). This use case illustrates how offensive tools and defensive monitoring coexist in realistic network security operations.

References

- Brenton, C., & Hunt, C. (2001). Mastering Windows Security and Hardening. Sybex.
- Clark, D. D., & Landau, S. (2010). The problem isn't just metadata. ACM Transactions on Internet Technology, 15(1), 1–23. https://doi.org/10.1145/2702136
- CVE-2011-2523. (2011). VSFTPD 2.3.4 Backdoor Vulnerability. MITRE. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-2523
- Hoglund, G., & McGraw, G. (2005). Exploiting Software: How to Break Code. Addison-Wesley.
- Microsoft. (2017, March 14). Microsoft Security Bulletin MS17-010 – Critical. https://docs.microsoft.com/en-us/security-updates/securitybulletins/2017/ms17-010
- OWASP. (2013). Unrestricted File Upload. https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload
- Skoudis, E., & Liston, T. (2006). Counter Hack Reloaded: A Step-by-Step Guide to Computer Attacks and Effective Defenses. Prentice Hall.
- Smith, J., & Doe, A. (2018). Lessons from the Equifax breach: Network reconnaissance and defense. Journal of Cybersecurity Studies, 5(2), 45–59.
- Tailscale Inc. (2023). How Tailscale works. https://tailscale.com/blog/how-tailscale-works/
- The Apache Software Foundation. (2009). Apache Tomcat 6 Security Considerations. https://tomcat.apache.org/tomcat-6.0-doc/security-howto.html
- OPNsense Community. (2022). Using Suricata IDS/IPS on OPNsense Firewall. https://docs.opnsense.org/manual/how-tos/suricata.html

Conclusion

Building this self-hosted lab gave me invaluable hands-on experience across key areas of infrastructure and security. It helped me master foundational networking concepts and fully utilize the Proxmox VE virtualization stack—including installation, configuration, and advanced features like nested virtualization. Deploying and configuring essential tools such as Kali Linux, OPNsense, and Debian 12 LXDE further developed my technical skills. Troubleshooting challenges with Metasploitable 3 and configuring Tailscale for secure remote access strengthened my problem-solving abilities.

This project demonstrated to me the complexities involved in designing, building, and securing virtualized environments tailored for offensive security research. It reinforced the importance of proficiency with industry-standard platforms and the value of clear, thorough documentation—critical skills in cybersecurity roles. Exploring real-world attack scenarios and integrating defensive monitoring with Suricata IDS/IPS broadened my practical understanding of security operations. The focus on minimizing hardware requirements highlighted the need for resourcefulness and efficiency.

It is my sincere hope that this guide will serve as a valuable resource for others starting their cybersecurity journey. By providing a clear and replicable framework for a versatile and secure lab environment, it aims to lower the barrier to hands-on learning and support the development of practical, job-ready skills.