Justin Strelka

OpenVAS\_kali

6/23/20

**Identifying Vulnerabilities with OpenVAS**

**Abstract:**

This paper begins with a brief description of why it is important to practice ethical hacking procedures within a contained environment. Followed by an outline of the tools used to create a contained environment and the role each tool plays. Next, there is a walk-through for setting up the environment used for this paper on a Windows 10 machine. After the environment has been established, we will deploy OpenVAS and provide an in-depth analysis of the vulnerabilities found during deployment. We will then conclude with ideas for exploiting some of the vulnerabilities found and gain insight for securing our production systems.

**Motivation:**

This paper is intended for individuals wanting to establish a secure and contained environment for testing vulnerabilities. Upon reading this paper you will be exposed to virtual machines that can be used to test against, so we don’t have to target our production machines. You will be able to describe each of the tools used to establish a contained environment and understand the role each tool plays in creating the environment. With an understanding of each of the tools necessary for creating a contained environment, you will be able to design your own environments that expand upon the environment designed in this paper. Lastly, you will be introduced to OpenVAS a vulnerability scanning tool that will help you identify the vulnerabilities that could be exploited on a device. Knowing where a device is vulnerable will help you secure the device and mitigate future attacks.

**Introduction:**

Being able to identify vulnerabilities within our systems will help us deter future exploits that may have catastrophic impacts. Knowing our vulnerabilities will help us be proactive to future attacks that could have otherwise caused reactive responses. Testing our systems in a contained environment using virtual machines (VMs) provides us with a safe and efficient way to identify our vulnerabilities. We can test exploits against VMs without exposing our production systems to harm. Using vulnerability scanning software such as OpenVAS to expose our vulnerabilities, along with testing exploits against known vulnerabilities on VM replicas of our production systems will help us design and implement effective patches to our systems.

**Method/Measurement:**

Before we begin deploying vulnerability scanning software like OpenVAS we must first establish a contained environment using virtual machines that can be exposed to exploits without impacting our production systems. For describing virtual machines, we refer to the physical machine as the host and the virtual machine as the guest. Virtual machines run on hypervisors, which oversee system resources. There are two types of hypervisors, type I hypervisors which act as an operating system and run directly on the system hardware. Type II hypervisors run within a host operating system and are implemented using software. [1] For the purposes of this paper we will be using a Type II hypervisor called VirtualBox to manage our guest VMs. VirtualBox is an opensource software that can be downloaded at the following link: <https://www.virtualbox.org/>.

A virtual machine running on a Type II hypervisor is the emulation of a guest operating system running within the hosts operating system. Virtual machines are compartmentalized within the virtual space of a system and in most cases can be attacked without harming the system they are running on. [2] To boot a virtual machine using VirtualBox we must first obtain an image of a guest operating system of our choice. For the purposes of this paper we will be creating two virtual machines. The first virtual machine we are going to create will be running Kali Linux and will be used to deploy our vulnerability mapping software OpenVAS. For the purposes of this paper we will be downloading a Kali Linux image from the following link: <https://www.offensive-security.com/kali-linux-vm-vmware-virtualbox-image-download/>. Be sure to choose the image for the correct hypervisor software and the 64-bit version of Kali. The second virtual machine we are going to create will be running Metasploitable2 which is a framework designed for testing and attacking against. [2] Metasploitable2 can be downloaded from the following link: <https://sourceforge.net/projects/metasploitable/files/latest/download>.

Once we have both images downloaded, we can begin establishing our virtual test environment. We will begin by creating our Kali Linux virtual machine which will be later used for deploying OpenVAS, a vulnerability scanning utility. Begin by opening the type II hypervisor we downloaded earlier named VirtualBox. We can boot VirtualBox and automate the Kali VM creation by opening the downloaded Kali Linux image from offensive-security.com. The screens in **Figure 1** will appear and as you can see, we have renamed our VM to “kali\_attack”. Once the VM has finished being created we can view and change the configuration settings for any of our VMs in the VirtualBox GUI. There will be one setting we want to pay special attention to, the network adapter. Open the network tab as shown in **Figure 2** and locate the Network adapter attached to the drop-down menu. The default NAT setting will make the virtual machine visible to the global network, however when we conduct our testing, we will not want our vulnerable VM’s (metasploit2) to be visible to the global network. To avoid this, we will later change the network adapter settings to “Host-only Adapter”. [3] For now we will leave the default NAT setting for the kali\_attack VM so we can update the mirrors and download/install OpenVAS.

A screenshot of a cell phone

Description automatically generated

**Figure 1**: Automated Kali VM set-up with name change.

A screenshot of a social media post

Description automatically generated

**Figure 2:** Network adapter setting that hides VM from global network.

Next we will power up our kali\_attack VM to complete the set-up and install OpenVAS. The initial default username and password for the kali\_attack VM is “kali”. Once you have successfully logged in, we will install OpenVAS. Open a terminal and enter the following commands: “sudo su –“, “apt-get update”, “apt-get install openvas”, “openvas-setup”, and “openvasmd –rebuild”. When “openvas-setup” completes it will create a new admin user and provide you with a password. Be sure to save your password for future logins. We can now start OpenVAS by running the command “openvas-start” which will bind to a default address of 127.0.0.1 (localhost) listening on port 9392. Go to this address in a web browser on the “kali\_attack” VM and you will be brought to the OpenVAS GUI where we can deploy and analyze results of vulnerability scans.

Next we will be setting up our vulnerable VM that we can deploy OpenVAS against for identifying vulnerabilities to later be exploited. Begin creating a new VM in VirtualBox by clicking the “New” tab at the top of the screen. You will be prompted to enter some basic information such as a name for the VM which we will call ours “metasploitable2” and metasploitable is a 64-bit Ubuntu Linux flavor. The default will be 1 GB of RAM allocated to the machine, however I would recommend giving 2 GB of RAM and 2 CPU’s to the VM as this will reduce any possible lag when testing. We will want to import a virtual disk image that has already been set up in the metasploitable2 zipped file. Extract the contents of the zipped file to a known location and select the “Use an existing virtual hard disk file” option in VirtualBox. Then search for the extracted metaploitable2 disk image as shown in **Figure 3**.

A screenshot of a social media post

Description automatically generated

**Figure 3:** VirtualBox VM set-up for metasploitable2.

Once the metasploitable2 VM has been created we need to ensure that we change the default network adapter setting from NAT to “Host-only Adapter”. As mentioned previously this will hide metasploitable2 from the global network which is what we want since metasploite2 is a framework designed to have vulnerabilities for testing. Once the network adapter for metasploitable2 has been configured we can now boot the VM. Once booted and logged in with the provided msfadmin username and password we will want to ensure both VMs can communicate. To do this we will obtain the IP address for the metasploitable2 VM by running the ifconfig command. Once the IP address has been obtained, we can now try to ping it from the “kali\_attack” VM. As you can see in **Figure 4** the kali\_attack was indeed able to successfully ping the metasploit2 machine with no dropped packets.

A screenshot of a computer

Description automatically generated

**Figure 4:** Successful communication between the metasploit2 machine and kalit\_attack machine.

At this point we have created our environment consisting of two VMs running on a single host. We have downloaded VirtualBox which is a Type II hypervisor meaning it is running on top of our hosts OS. We downloaded disk images and configured two separate VMs. One for deploying the vulnerability scanning tool OpenVAS which is a Kali Linux distribution OS. The second VM we created is purposed for being scanned by OpenVAS and is a special framework designed to have vulnerabilities for exploiting called metasploit2. We have installed and configured OpenVAS on the Kali Linux machine and tested communication between the two VMs. Our contained testing environment is now complete and we are ready to begin deploying OpenVAS from kali\_attack targeting metasploit2.

On kali\_attack run the command openvas-start to start the OpenVAS service and navigate a browser to 127.0.0.1:9392. Once the OpenVAS browser GUI has appeared enter the credentials noted earlier. After authenticating we will navigate to the Configuration menu and select “Targets”. We can create a new target by clicking the star icon in the upper left corner of the screen. As shown in **Figure 5** you will be prompted to enter information about the target we wish to scan. We will name our target “metasploit2” and provide the IP address obtained from the metasploit2 VM we noted earlier. We will be scanning all IANA registered TCP ports on metasploit2. IANA is short for the Internet Assigned Numbers Authority which is an organization that is responsible for maintaining Domain Names, and other various internet numbering systems such as a port registry. [4] We will create a new credential and name it metasploit2 with username and password “msfadmin” for authenticated vulnerability scanning as shown in **Figure 6.**

**A screenshot of a cell phone

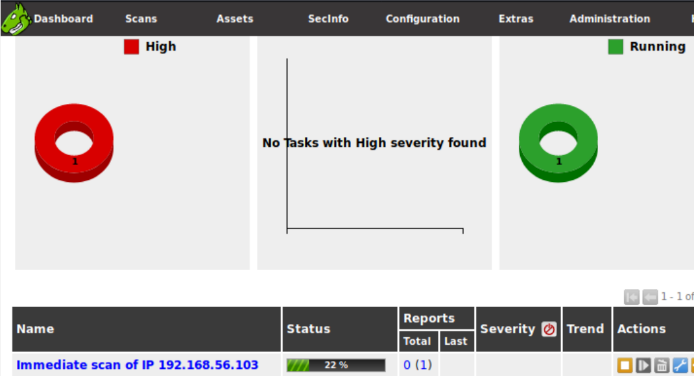
Description automatically generatedA screenshot of a cell phone

Description automatically generated**

**Figure 6:** Creating new credential in OpenVAS browser GUI

**Figure 5:** Creating new target in OpenVAS browser GUI

Next we will navigate to the scans tab to configure a new scan that will target only metasploit2. Click the star icon in the top right corner of the screen to create a new task. We will call the new task “msploit” to keep our targets and tasks separately identified. We will set our scan config setting to “Full and Fast” which can later be changed to a more in-depth scan of the system. **Figure 7** shows the configurations that have been set to conduct our scan on metasploit2. We can then start our scan by clicking the green play arrow located under the actions tab. While a report is running, we can monitor the results and progress of the scan as shown in **Figure 8.** When the report has completed the status of the task will change from a percentage of completion to displaying “Done”.

**A screenshot of a cell phone

Description automatically generated**

**Figure 8:** Monitoring task as it runs in OpenVAS GUI

**Figure 7:** Task settings in OpenVAS GUI

**Results:**

We can now analyze the results of our metasploit2 scan and view the vulnerabilities identified by OpenVAS. Navigate to the “Scans” tab and select “Results” which will display the vulnerabilities found. The “Results” page provides us with information that describes the vulnerabilities identified. The vulnerability column gives a brief description of the vulnerability found and by clicking the hyperlink of the vulnerability we can get detailed information regarding what flagged the vulnerability and possible exploit. The Solution type column offers us known solutions that can be implemented to remove the vulnerability from the target system. The severity column tells us how well a vulnerability matches the defined Common Vulnerability Scoring System (CVSS) that can be adjusted by the user. The Quality of Detection (QoD) column tells us how reliable the results are to minimize the possibility of a false positive vulnerability alert. The host lets us know what machine the vulnerabilities were found on in case we run a multi-host scan. The Location column tells us the location of the vulnerability by protocol type and port number. [5]

The metasploit2 scan identified 139 vulnerabilities on the metasploit2 target. As shown in **Figure 9,** 82 of the vulnerabilities identified were of the log level severity. Alerts that fall within the log level severity generally do not pose immediate threat to the system. However, with a full understanding of an application they can provide information that can be used to help secure a system. Two of the alerts reported were of “Low” severity which is considered to be similar to “Log” severity alerts with slightly more vulnerable information exposed. There are 33 medium severity vulnerabilities and 22 high severity vulnerabilities which both pose immediate threat to the system and should be addressed ASAP.

­A screenshot of a social media post

Description automatically generated

**Figure 9:** Display of identified vulnerability spread.

The first vulnerability we are going to look at is named “MySQL / MariaDB weak password”. As shown in **Figure 10,** if we click on the vulnerability hyperlink to review detailed information about the vulnerability, we can see OpenVAS was able to authenticate as root without a password in the Database Management System (DBMS). This vulnerability could be exploited easily by an internal user obtaining sensitive or confidential data. Or an external user could potentially exploit this vulnerability as well by using the “curl” command to extract or ingest data into the database, compromising sensitive data or corrupting the database. Database security is of extreme importance since the data they hold can be potentially harmful to individuals and corrupting the data could cost the company an extensive amount of time and money in research investments. We can see the recommended solution is to change the password as soon as possible which is easy to do with most modern DBMS.

A screenshot of a cell phone

Description automatically generated

**Figure 10:** Display of MySQL weak/missing password vulnerability.

A screenshot of a social media post

Description automatically generatedThe next vulnerability we will look at is named “Anonymous FTP Login Reporting”. As shown in **Figure 11** it was possible to login to the targets remote FTP service with standard credentials. If a malicious individual were to connect to the target through the FTP remote service, they could extract sensitive files. The extraction of sensitive files can cause harm to individuals or the company. Malicious users could also upload any file they wanted as a decoy to be executed by a user who could unknowingly compromise an entire system. To rectify this issue, we could disable anonymous logins to the FTP remote service since verified users will have assigned credentials to use the service as intended.

**Figure 11:** Anonymous FTP login display in OpenVAS browser GUI.

A screenshot of a social media post

Description automatically generatedThe final vulnerability we are going to review is displayed in **Figure 12** and is named “phpinfo() output Reporting”. This vulnerability provides malicious users the ability to access a sensitive file called phpinfo.php through HTTP. The phpinfo.php file can contain information such as the username of the user running PHP processes and if the user is a sudo user. The phpinfo.php file can also contain the address of the host, the host OS, root directory of the web server and web server version. By obtaining all this information in a single unprotected file a malicious user could potentially compromise the host through known vulnerabilities to the web server version and possibly gain sudo access by knowing the username of the PHP process user. The phpinfo.php file can also contain internal IP addresses allowing the malicious use to potentially map out the systems architecture to formulate a more sophisticated plan of attack.

**Figure 12:** phpinfo() output reporting vulnerability displayed in OpenVAS GUI.

**Conclusion:**

We have outlined how to set up a contained testing environment using VM’s. The use of VM’s allows us to set-up our environments, break them, and throw them away or quickly replicate them again. We can do all of this without putting our production machines at risk for being corrupted. We have seen that there are discovery tools available for identifying vulnerabilities on our systems such as OpenVAS. These tools are useful for hardening our own systems and we must realize that malicious individuals also have access to such tools. Because of this we must be diligent in hardening our systems, identifying potential attacks and understanding the implications that vulnerabilities may have on our systems.

**Bibliography**

[1] geeksforgeeks.org. Retrieved From <https://www.geeksforgeeks.org/hypervisor/>

[2] Microsoft. 2019. Retrieved From <https://docs.microsoft.com/en-us/azure/lab-services/class-type-ethical-hacking>

[3] virtualbox.org. Retrieved From <https://www.virtualbox.org/manual/ch06.html>

[4] Internet Assigned Numbers Authority. Retrieved From <https://www.iana.org/about>

[5] greenbone.net. 2019. Retrieved From <https://docs.greenbone.net/GSM-Manual/gos-4/en/reports.html>

**Reflection:**

This paper opened my eyes to how easily it is for a malicious individual to identify and understand system vulnerabilities. I was unaware there were opensource vulnerability scanning tools available. I feel they are necessary for cybersecurity professionals, yet they provide malicious users a simple way to discover a system. During the set-up phase of my contained lab I had lots of issues getting everything to run together. A day or two before I began this paper I rolled my system up to the new Windows 10 build 2004. I had also installed Docker the morning I started writing. Docker runs on Hyper-V and apparently for VirtualBox to run correctly Hyper-V should be disabled. After about 10 hours of beating around I finally decided to roll my windows update back to the previous build where VirtualBox had ran fine in the past. Once I had done so Docker was also uninstalled and my VMs ran seamlessly. Throughout my semi-frantic and anger entourage I was able to learn some of the basics about how Docker, virtualization through Hyper-V, and a solid understanding of how mirrors work within the sources.list file. Once my machine was set up and I was able to conduct a successful scan I realized how intricate various exploits can be. There were some exploits listed for applications I have never heard of before and all they required was a single unencrypted piece of data that could compromise an entire system. Moving forward I am going to scan my physical box for vulnerabilities as over the years I have had some careless moments trying to get applications to run. Any one of these moments could have exposed a vulnerability that may or may not have been exploited at this time. However, by using NMAP and OpenVAS together I believe I will be able to re-instate my confidence in the Confidentiality and Integrity of my PC.