Vulnerability Assessment Lab

SEIS 720 Semester project, Fall 2018

Esten Rye, University of St Thomas

2018

Contents

[Abstract 2](#_Toc532150776)

[Hypervisor Selection 3](#_Toc532150777)

[Infrastructure as Code Tool Selection 3](#_Toc532150778)

[Operating System Selection 3](#_Toc532150779)

[External Vulnerability Assessment Results 4](#_Toc532150780)

[External Vulnerability Mitigation 5](#_Toc532150781)

[Internal Vulnerability Assessment Results 6](#_Toc532150782)

[Internal Vulnerability Mitigation 7](#_Toc532150783)

[External Mitigation Assessment Results 8](#_Toc532150784)

[Internal Mitigation Assessment Results 10](#_Toc532150785)

[Appendix A – External Vulnerability Assessment 12](#_Toc532150786)

[Appendix B – Internal Vulnerability Assessment 12](#_Toc532150787)

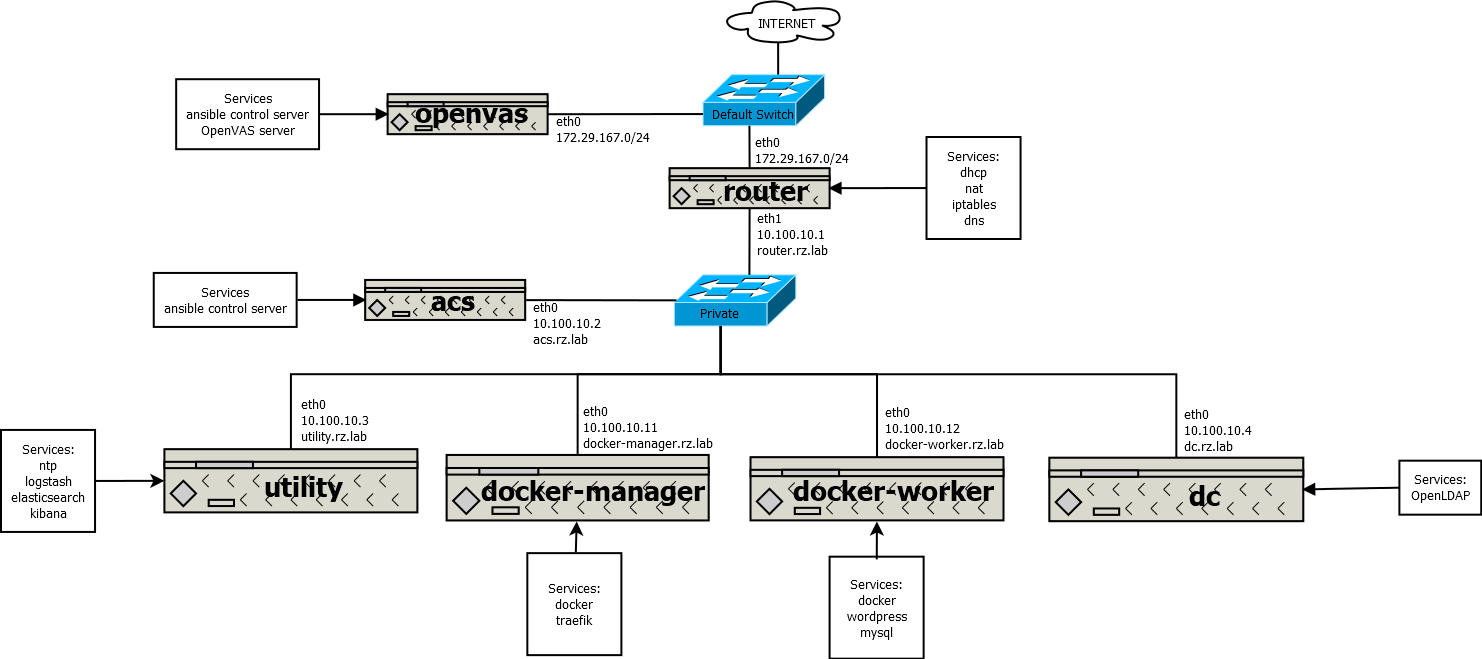
[Appendix C – External Vulnerability Mitigation Assessment 12](#_Toc532150788)

[Appendix D – Internal Vulnerability Mitigation Assessment 12](#_Toc532150789)

[References 13](#_Toc532150790)

# Abstract

This project set out to build a simulated enterprise network which could be used to explore assessment technologies and defense-in-depth strategies in a controlled environment without the typical risks of impacting day-to-day business operations. Much of the project effort centered around creating a set of ansible automation roles that would construct, configure and maintain the simulated network, pictured below:



This network contains many of the common services you might expect to find in the typical enterprise network of an organization progressing down their devOps transformation journey including:

* OpenLDAP for centralized authentication and authorization.
* NTP for centralized network time synchronization.
* The Elastic stack (elasticsearch, kibana, and logstash) for centralized log management and performance monitoring.
* A perimeter gateway device providing DHCP, NAT, DNS and perimeter firewall services.
* Docker Swarms providing a standardized container platform for application delivery
* Web applications and database services.
* Ansible control server providing infrastructure as code services.

Greenbone Security Manager (GSM) Community Edition (CE) version 10 was used to assess the vulnerability of the network that was constructed. GSM CE is an opensource derivative of Greenbone’s GSM ONE product which is targeted at security audit professionals who need a laptop-based solution. Two vulnerability assessments were performed against the simulated network. The first assessment focused on the external vulnerability to attackers outside of the network. The second assessment assessed the internal vulnerability to attackers who had already breached then perimeter firewall.

The paper is concluded with a list of the vulnerabilities that were mitigated, and next steps for future development.

# Hypervisor Selection

The simulated network is built on top of the Microsoft Hyper-V hypervisor that is built into all Windows 10 machines installed with Professional or greater. This hypervisor was selected over Oracle’s Virtual box software for two reasons:

1. Most developers who are developing Microsoft-based applications on Windows machines will likely already have it enabled as a result of installing Docker for Windows.
2. Oracle’s Virtualbox software cannot be installed with Hyper-V enabled. Disabling Hyper-V breaks Docker for Windows.

As a result of selecting Hyper-V as the hypervisor, a couple of key limitations were encountered, and workarounds were developed.

1. Hyper-V does provide an easy interface to create virtual machines with a static IP. As a workaround, the router.rz.lab virtual machine (VM) is provisioned to grant statically assigned IP addresses based on the client’s hardware MAC address via DHCP.
2. Hashicorp’s Vagrant provider for Hyper-V does not support creation of virtual machines with multiple network interface devices. As a workaround, the additional network interface required by the router.rz.lab VM is added manually using native PowerShell commands.
3. The vagrant up command when issued to a VM with multiple network interface devices re-maps all network interface devices with the same network attachment and hardware MAC address. As a workaround, the startup state for router.rz.lab is managed using the native PowerShell commands.

Much of the code developed in this project will likely work the same in Virtualbox with a few tweaks to the Vagrantfile.

# Infrastructure as Code Tool Selection

The simulated network uses Ansible as its infrastructure as code technology. Ansible provides a push-based, centralized automation platform that can be used to script changes to the network once and apply them consistently and repeatably to the virtual machines on the network. Ansible was selected because it is an open-source, agentless solution that does not require centralized server to deploy configuration changes or manage state/host inventory. This reduces the barrier to entry and overall complexity of the configuration management system while minimizing the amount of additional infrastructure that must be stood up to support the system.

# Operating System Selection

Both CentOS 7.5 and Ubuntu 18.04 were evaluated for use as the underlying server operating system to power the network. Ubuntu 18.04 was selected over CentOS 7.5 for the following reasons:

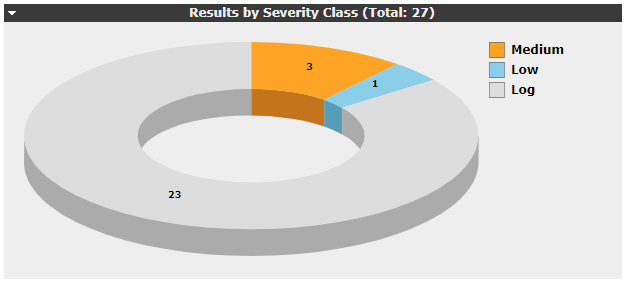
1. Ansible was easier to deploy on the publicly available vagrant boxes.
2. Author’s preference of the apt package repository over the yum package repository.
3. Author’s need to familiarize himself with the single sign-on capabilities of the Ubuntu 18.04 operating system.
4. Ubuntu 18.04 is the long-term support version, supported until April 2023.

# External Vulnerability Assessment Results

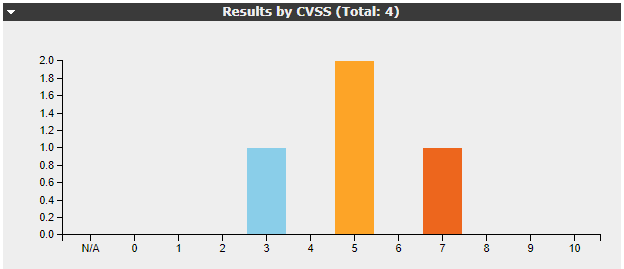
After completing construction of the simulated network, a Greenbone Security Manager Community Edition (GSM CE) virtual machine was downloaded and connected to the same switch as the WAN router interface. GSM CE executed 48,176 individual network vulnerability tests against the WAN interface of the following hosts:

* router.rz.lab

The assessment reported 27 notable test results. Resulting in 3 Medium, 1 Low and 23 Log severity issues.



The test results had the following severity distribution using the Common Vulnerability Scoring System (CVSS). Note that log events all have a severity of 0.0 and have been excluded from the graph to better illustrate the distribution.



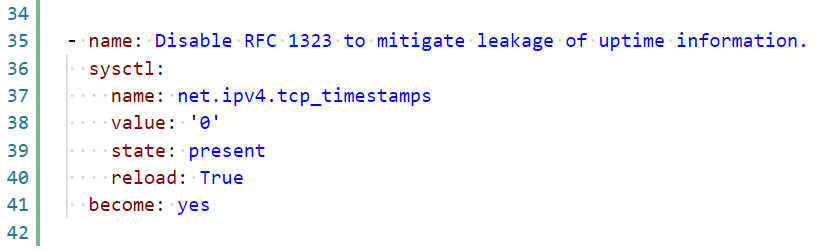
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id | Vulnerability | Solution Type | Severity | Quality of Detection | Location |
| 1 | GZip 'huft\_build()' in 'inflate.c' Input Validation Vulnerability (Linux) | Vendor Fix | 6.8 (Medium) | 30% | general/tcp |
| 2 | OpenSSH 'auth2-gss.c' User Enumeration Vulnerability (Linux) | None Available | 5.0 (Medium) | 30% | 22/tcp |
| 3 | OpenSSH User Enumeration Vulnerability-Aug18 (Linux) | None Available | 5.0 (Medium) | 30% | 22/tcp |
| 4 | TCP timestamps | Mitigation | 2.6 (Low) | 80% | general/tcp |

# External Vulnerability Mitigation

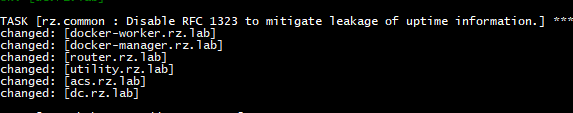
Vulnerability 1 is an input validation vulnerability caused by a flaw in GZip application versions prior to 1.3.13. Manual review of router.rz.lab has verified that GZip version 1.6 has been installed. This risk is marked as a False Positive.

Vulnerabilities 2 and 3 are information leakage vulnerabilities that cannot be mitigated because there is no vendor supported upgrade to OpenSSH 7.9. These risks have been accepted and could be mitigated completely either by not exposing the SSH port (22/tcp) to the WAN or by installing a version 7.9 of OpenSSH from source.

Vulnerability 4 is a low-level information leakage vulnerability. The assessment uncovered that the WAN port implements RFC 1323 which has the unintended side effect of allowing and external entity to potentially calculate uptime of router.rz.lab. To mitigate this vulnerability a modification was made to the rz.common role to allow Ansible to apply the recommended configuration change across the entire network. Below is the code snippet that will apply the mitigation:



Pushing this modification to the acs.rz.lab VM and running “*ansible-playbook -i rz\_lab\_inventory --become playbooks/site.yml”* applies the mitigation to all machines in the network. Successful application of the mitigation appears in the log as shown below:



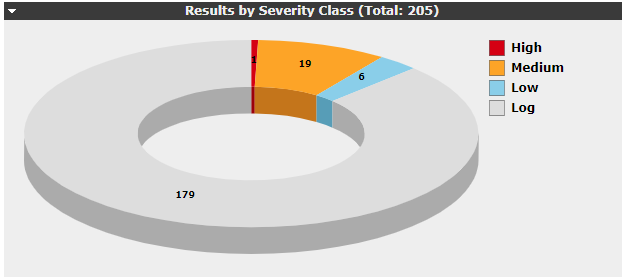
After applying the mitigation to the router.rz.lab VM it was observed that the vulnerability was not mitigated completely. This is due to the port forwarding of port 80 to the docker-manager.rz.lab VM. Because the docker-manager.rz.lab is hosting a vulnerable container that responds on port 80, this result manifests itself on the router.rz.lab VM.

# Internal Vulnerability Assessment Results

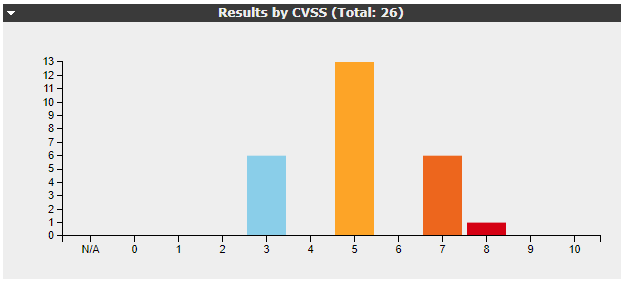
An internal audit of the network was also performed using Greenbone Security Manager Community Edition (GSM CE). The same assessment settings used for the external assessment were also used for the internal assessment. GSM CE executed 48,176 individual network vulnerability tests against the LAN interface of the following 6 hosts:

* router.rz.lab
* acs.rz.lab
* utility.rz.lab
* dc.rz.lab
* docker-manager.rz.lab
* docker-worker.rz.lab.

The assessment reported 205 notable results. Resulting in 1 High, 19 Medium, 6 Low and 179 Log results.



These results had the following severity distribution using the Common Vulnerability Scoring System. Note that log events all have a severity of 0.0 and have been excluded from the graph to better illustrate the distribution.



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id | Vulnerability | Solution Type | Severity | Quality of Detection | Location |
| 1 | ISC BIND 'deny-answer-aliases' Denial of Service Vulnerability | Vendor Fix | 7.8 (High) | 30% | 53/tcp |
| 2-7 | GZip 'huft\_build()' in 'inflate.c' Input Validation Vulnerability (Linux) | Vendor Fix | 6.8 (Medium) | 30% | general/tcp |
| 8-13 | OpenSSH 'auth2-gss.c' User Enumeration Vulnerability (Linux) | None Available | 5.0 (Medium) | 30% | 22/tcp |
| 14-19 | OpenSSH User Enumeration Vulnerability-Aug18 (Linux) | None Available | 5.0 (Medium) | 30% | 22/tcp |
| 20 | ISC BIND 9 Remote Dynamic Update Message Denial of Service Vulnerability | Vendor Fix | 4.3 (Medium) | 30% | 53/tcp |
| 21-26 | TCP timestamps | Mitigation | 2.6 (Low) | 80% | general/tcp |

# Internal Vulnerability Mitigation

Vulnerability 1 is a denial of service vulnerability that allows remote attackers to cause denial of service through assertion failures in the ‘deny-answer-aliases’ feature. This is patched in version 9.11.4-P1 of ISC BIND but is not yet available as a supported package update for Ubuntu 18.04. This service is not exposed to the WAN, mitigating the risk of an external actor being successful at exploiting this vulnerability. This risk will be accepted as Medium severity and monitored in the event Ubuntu releases a patch that resolves the issue.

Vulnerabilities 2-7 are an input validation vulnerability caused by a flaw in GZip application versions prior to 1.3.13. Manual review of all hosts has verified that GZip version 1.6 has been installed. This risk is marked as a False Positive.

Vulnerabilities 8-13 are an information leakage vulnerability that allows a remote attacker to enumerate valid user accounts as a result of a flaw due to insufficient validation of an authentication request packet. No known solution has been provided by OpenSSH and no vendor supported package update for Ubuntu 18.04 is available. These risks have been accepted and marked as Low severity as the LAN interface is not accessible to a remote attacker, mitigating the risk of this vulnerability. This risk will be monitored and mitigated if or when an update becomes available.

Vulnerabilities 14 through 19 are an information leakage vulnerability that allows a remote attacker to enumerate users due to a short-circuit path in the authentication logic that does not delay bailout until after the packet making the request has been fully parsed. No known solution has been provided by OpenSSH and no vendor supported package update for Ubuntu 18.04 is available. These risks have been accepted and marked as Low severity as the LAN interface is not accessible to a remote attacker, mitigating the risk of this vulnerability. This risk will be monitored and mitigated if or when an update becomes available.

Vulnerability 20 is a denial of service vulnerability that occurs when a remote attacker specially crafts a dynamic update request allowing the attacker to crash the affected DNS server. This vulnerability affects versions of BIND prior to 9.4.3-P3, 9.5.1-P3 and 9.6.1-P1. This host currently has BIND version 9.11.3.1ubuntu1.3 installed and is unaffected by this vulnerability. This risk will be marked as a false positive.

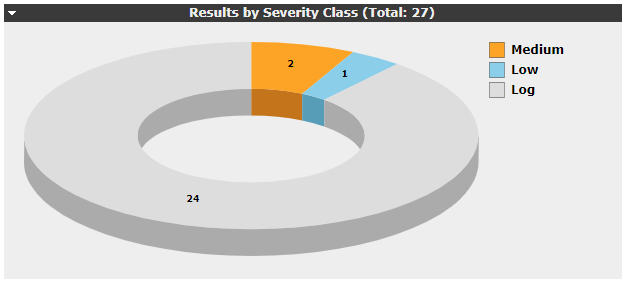
Vulnerabilities 21-26 are an information leakage vulnerability where implementation of RFC 1323 has the unintended side effect of allowing and external entity to potentially calculate uptime of a host. This has been mitigated for all hosts by updating the appropriate configuration setting, however, it was discovered that hosts that run docker containers may still be vulnerable due to the hosted container implementing RFC 1323.

# External Mitigation Assessment Results

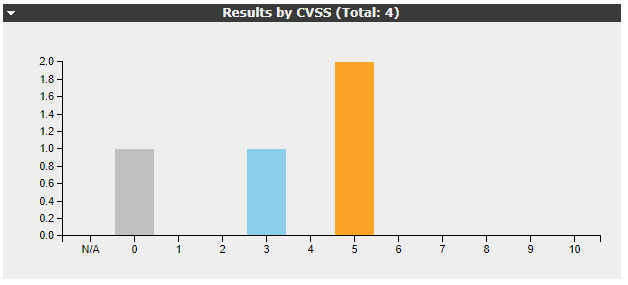
After completing construction of the simulated network, a Greenbone Security Manager Community Edition (GSM CE) virtual machine was downloaded and connected to the same switch as the WAN router interface. GSM CE executed 48,176 individual network vulnerability tests against the WAN interface of the following hosts:

* router.rz.lab

The assessment reported 27 notable test results. Resulting in 2 Medium, 1 Low, 23 Log and 1 False Positive severity issues.



These results had the following severity distribution using the Common Vulnerability Scoring System. Log events all have a severity of 0.0 and have been excluded from the graph to better illustrate the distribution. False Positive results have been included in the graph to demonstrate the post-assessment vulnerability mitigation.



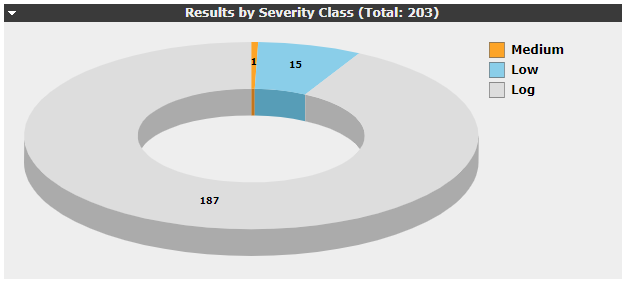
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id | Vulnerability | Solution Type | Severity | Quality of Detection | Location |
| 1 | GZip 'huft\_build()' in 'inflate.c' Input Validation Vulnerability (Linux) | Vendor Fix | 0.0 (False Positive) | 30% | general/tcp |
| 2 | OpenSSH 'auth2-gss.c' User Enumeration Vulnerability (Linux) | None Available | 5.0 (Medium) | 30% | 22/tcp |
| 3 | OpenSSH User Enumeration Vulnerability-Aug18 (Linux) | None Available | 5.0 (Medium) | 30% | 22/tcp |
| 4 | TCP timestamps | Mitigation | 2.6 (Low) | 80% | general/tcp |

# Internal Mitigation Assessment Results

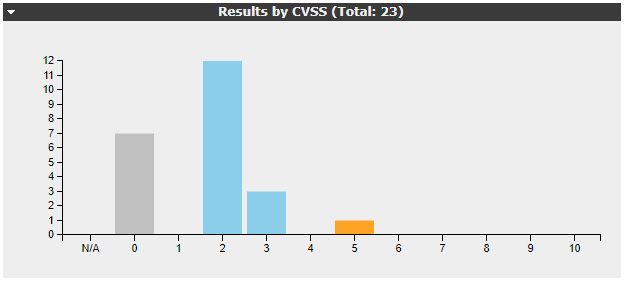
An internal audit of the network was performed with using Greenbone Security Manager Community Edition (GSM CE) after reviewing the vulnerability results. The same assessment settings used for the external assessment were also used for the internal assessment. GSM CE executed 48,176 individual network vulnerability tests against the LAN interface of the following 6 hosts:

* router.rz.lab
* acs.rz.lab
* utility.rz.lab
* dc.rz.lab
* docker-manager.rz.lab
* docker-worker.rz.lab.

The assessment reported 203 notable results. Resulting in 1 Medium, 15 Low and 172 Log and 7 False Positive results.



These results had the following severity distribution using the Common Vulnerability Scoring System. Log events all have a severity of 0.0 and have been excluded from the graph to better illustrate the distribution. False Positive results have been included in the graph to demonstrate the post-assessment vulnerability mitigation.



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id | Vulnerability | Solution Type | Severity | Quality of Detection | Location |
| 1 | ISC BIND 'deny-answer-aliases' Denial of Service Vulnerability | Vendor Fix | 5.0 (Medium) | 30% | 53/tcp |
| 2-7 | GZip 'huft\_build()' in 'inflate.c' Input Validation Vulnerability (Linux) | Vendor Fix | 0.0 (False Positive) | 30% | general/tcp |
| 8-13 | OpenSSH 'auth2-gss.c' User Enumeration Vulnerability (Linux) | None Available | 2.0 (Low) | 30% | 22/tcp |
| 14-19 | OpenSSH User Enumeration Vulnerability-Aug18 (Linux) | None Available | 2.0 (Low) | 30% | 22/tcp |
| 20 | ISC BIND 9 Remote Dynamic Update Message Denial of Service Vulnerability | Vendor Fix | 0.0 (False Positive) | 30% | 53/tcp |
| 21-23 | TCP timestamps | Mitigation | 2.6 (Low) | 80% | general/tcp |

# Appendix A – External Vulnerability Assessment

<https://github.com/estenrye/ansible-rancher-lab/raw/master/.docs/SEIS%20720%20-%20Semester%20Project%20Report/Project%20-%20Appendix%20A%20-%20External%20Vulnerability%20Assessment.pdf>

# Appendix B – Internal Vulnerability Assessment

<https://github.com/estenrye/ansible-rancher-lab/raw/master/.docs/SEIS%20720%20-%20Semester%20Project%20Report/Project%20-%20Appendix%20B%20-%20Internal%20Vulnerability%20Assessment.pdf>

# Appendix C – External Vulnerability Mitigation Assessment

<https://github.com/estenrye/ansible-rancher-lab/raw/master/.docs/SEIS%20720%20-%20Semester%20Project%20Report/Project%20-%20Appendix%20C%20-%20External%20Vulnerability%20Mitigation%20Assessment.pdf>

# Appendix D – Internal Vulnerability Mitigation Assessment

<https://github.com/estenrye/ansible-rancher-lab/raw/master/.docs/SEIS%20720%20-%20Semester%20Project%20Report/Project%20-%20Appendix%20D%20-%20Internal%20Vulnerability%20Mitigation%20Assessment.pdf>

# References

!robot. (2018). Configure static IP addresses on ubuntu 18.04 LTS server. Retrieved from <https://websiteforstudents.com/configure-static-ip-addresses-on-ubuntu-18-04-beta/>

Anicas, M. (2015a). Iptables essentials: Common firewall rules and commands. Retrieved from <https://www.digitalocean.com/community/tutorials/iptables-essentials-common-firewall-rules-and-commands>

Anicas, M. (2015b). Iptables essentials: Common firewall rules and commands. Retrieved from <https://www.digitalocean.com/community/tutorials/iptables-essentials-common-firewall-rules-and-commands>

Bilal, A. (2018). Fixing DNS issues in ubuntu 18.04 LTS. Retrieved from <https://medium.com/@ahmadb/fixing-dns-issues-in-ubuntu-18-04-lts-bd4f9ca56620>

Brailovsky, N. (2013). Setting up a linux gateway/router, a guide for non network admins. Retrieved from <https://monoinfinito.wordpress.com/series/setting-up-a-linux-gatewayrouter-a-guide-for-non-network-admins/>

chmduquesne. (2018). Persistent iptables handlers. Retrieved from <https://github.com/chmduquesne/ansible-iptables-persistent/blob/master/handlers/main.yml>

Doug. (2018). Ubuntu 18.04 as a router. Retrieved from <https://askubuntu.com/questions/1050816/ubuntu-18-04-as-a-router>

Ducea, M. (2006). How to enable IP forwarding in linux. Retrieved from <http://www.ducea.com/2006/08/01/how-to-enable-ip-forwarding-in-linux/>

Ellingwood, J. (2015). How to forward ports through a linux gateway with iptables. Retrieved from <https://www.digitalocean.com/community/tutorials/how-to-forward-ports-through-a-linux-gateway-with-iptables>

Halfgarr. (2012). Configure OpenLDAP with TLS=required. Retrieved from <https://serverfault.com/questions/459718/configure-openldap-with-tls-required>

Hucculak, M. (2018). How to change the priority order of network adapters on windows 10. Retrieved from <https://www.windowscentral.com/how-change-priority-order-network-adapters-windows-10>

Joy, B. (2015). Ansible role openldap server. Retrieved from <https://github.com/bennojoy/openldap_server>

Kennedy, B. (2013). My first 5 minutes on A server; or, essential security for linux servers. Retrieved from <https://plusbryan.com/my-first-5-minutes-on-a-server-or-essential-security-for-linux-servers>

killtacknine. (2017a). Building an ubuntu 16.04 router part 1: Network interfaces. Retrieved from <https://killtacknine.com/building-an-ubuntu-16-04-router-part-1-network-interfaces/>

killtacknine. (2017b). Building an ubuntu 16.04 router part 2: DHCP. Retrieved from <https://killtacknine.com/building-an-ubuntu-16-04-router-part-2-dhcp/>

killtacknine. (2017c). Building an ubuntu 16.04 router part 3: Firewalls. Retrieved from <https://killtacknine.com/building-an-ubuntu-16-04-router-part-3-firewalls/>

killtacknine. (2017d). Building an ubuntu 16.04 router part 5: DNS. Retrieved from <https://killtacknine.com/building-an-ubuntu-16-04-router-part-5-dns/>

Langasek, S., & Canonical Ltd. (2008). Pam-auth-update.8. Retrieved from <http://manpages.ubuntu.com/manpages/bionic/man8/pam-auth-update.8.html>

Lapenna, A. (2018). Elastic stack (ELK) on docker. Retrieved from <https://github.com/deviantony/docker-elk>

Laurenson, T. (2018). Building a linux gateway using ubuntu server 18.04 Retrieved from <https://www.thomaslaurenson.com/blog/2018/07/05/building-a-ubuntu-linux-gateway/>

Ma, E. Z.Port forwarding using iptables. Retrieved from <https://www.systutorials.com/816/port-forwarding-using-iptables/>

Mallett, A., & Pluralsight. (2018). Linux: Network client management (LPIC-2). Retrieved from <https://app.pluralsight.com/library/courses/linux-network-client-management-lpic-2/table-of-contents>

Miller, T. C. (2018). Sudoers.ldap. Retrieved from <http://manpages.ubuntu.com/manpages/bionic/man5/sudoers.ldap.5.html>

Salter, J. (2014). Allowing traceroutes to succeed with iptables. Retrieved from <http://jrs-s.net/2014/08/22/allowing-traceroutes-to-succeed-with-iptables/>

Schroder, C. (2017a). Building IPv6 firewalls: IPv6 security myths. Retrieved from <https://www.linux.com/learn/intro-to-linux/2017/7/building-ipv6-firewalls-ipv6-security-myths>

Schroder, C. (2017b). How to write iptables rules for IPv6. Retrieved from <https://www.linux.com/learn/intro-to-linux/2017/8/iptables-rules-ipv6>

SSSD developers. (2017a). Debugging and troubleshooting SSSD. Retrieved from <https://docs.pagure.org/SSSD.sssd/users/troubleshooting.html>

SSSD developers. (2017b). Troubleshooting SUDO. Retrieved from <https://docs.pagure.org/SSSD.sssd/users/sudo_troubleshooting.html>

Stepan, gkop & Cakemox.How to disable SSH login with password for some users? Retrieved from <https://serverfault.com/questions/285800/how-to-disable-ssh-login-with-password-for-some-users>

Stoik, J. (2017). Fix linux DNS issues caused by systemd-resolved. Retrieved from <https://blobfolio.com/2017/05/fix-linux-dns-issues-caused-by-systemd-resolved/>

Taheri, H., & Carvalho, E. (2013). How to make IP forwarding permanent? Retrieved from <https://askubuntu.com/questions/311053/how-to-make-ip-forwarding-permanent>

The SSSD upstream. (2018). Sssd-ldap(5) - linux man page. Retrieved from <https://linux.die.net/man/5/sssd-ldap>