CNA430/530: FIREWALL AND PENETRATION TESTING

ST. CLOUD STATE UNIVERSITY

SPRING 2022

PROJECT-02: Security Root Causes Analysis and Prevention Techniques of Vulnerabilities in MongoDB with Singularity Linux Containers

By Group-01:

Khalid Alghamdi

William Munnich

Project Supervisor:

Dr. Akalanka Bandara Mailewa

Contents

1.Problem Statement

Identify the main causes of at least 15 security flaws detected in Project 1 The next step is to offer a method of preventing each vulnerability that has been discovered. Prioritize vulnerabilities by severity level, such as "Low," "Medium," and "High," before deciding which ones to investigate further. Finally, put some of the preventative strategies into action.

2. Introduction

A vulnerability is a flaw in a system that can be exploited by a hostile individual. It is critical to identify and mitigate any vulnerabilities to prevent them from being exploited by someone with bad intent. The tools we used to detect vulnerabilities in a machine running the Ubuntu operating system will be described in this section. All the stages outlined in the project guide have been completed. At the end of the process, we arrive at all of the conclusions.

3. Procedure & Discussion

In Project 1, we set up the system and get all the installations completed to start the vulnerability assessment of MongoDB. We install the following into our VirtualBox:

1) Ubuntu 18.04
2) Docker Engine
3) Singularity Engine
4) MongoDB images

The tools that we utilized in this project are as follows:

I. OpenVAS/Nessus
II. MongoAudit
III. Nmap

I. OpenVAS

OpenVAS stands for Open Vulnerability Assessment Scanner. It can do both unauthenticated and authenticated testing, as well as a variety of high- and low-level internet and industrial protocols, as well as performance tweaking for large-scale scanning. [4]

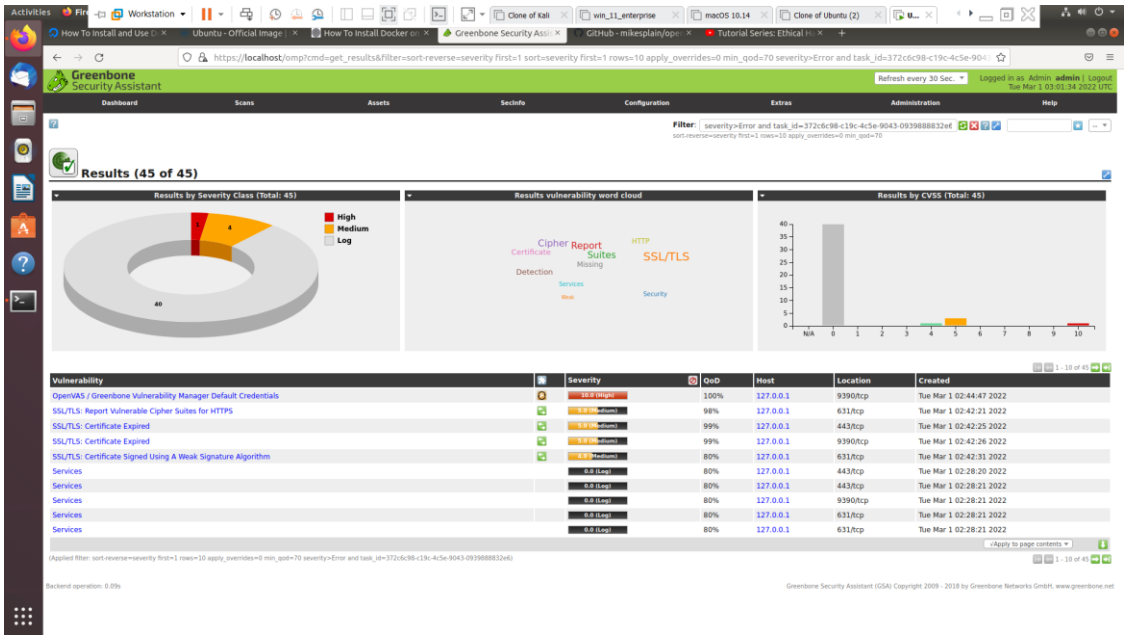

Figure 1 Completed Vulnerabilities Scan by OpenVAS

II. MongoAudit

MongoAudit is a command-line Interface for auditing MongoDB servers, spotting security flaws, and automating penetration testing (Whalen). It is well known that MongoDB's default setup has several security flaws. Because of this, and because of an abundance of many engineers, the media have dubbed it the MongoDB Apocalypse. The among audit tool not only identifies known vulnerabilities and defects but also provides you with instructions on how to repair them. [5]

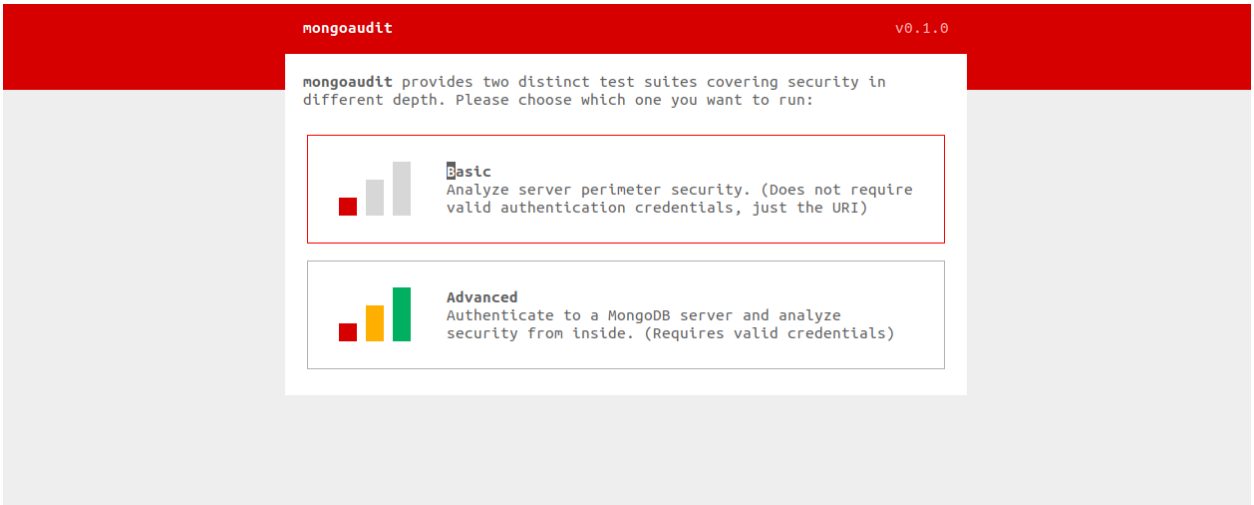

Figure 2 Mongoaudit two test suites

III. Nmap

One of the most popular open-source tools for discovering and auditing networks is called Nmap. [1] Now, system administrators utilize it as one of their primary methods of network mapping. On a network, Nmap looks for hosts and services. [3]

```
osboxes@osboxes:~$ sudo nmap -p 27017 127.0.0.1 -sV -O

Starting Nmap 7.60 ( https://nmap.org ) at 2022-03-01 14:38 EST
Stats: 0:00:10 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 0.00% done
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00010s latency).

PORT      STATE SERVICE VERSION
27017/tcp open  mongodb MongoDB 3.4.4
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.32
OS details: Linux 2.6.32
Network Distance: 0 hops

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 10.19 seconds
```

Figure 9 port scanning of Nmap

The methodology we are following to complete Project 2 is:

- Prioritizing vulnerabilities discovered in Project-01 across all categories, including Application, Image Containers, Host, and Network.
- Choosing at least 15 vulnerabilities from the prioritized list to cover all the categories above and discovering the underlying causes of each prioritized vulnerability in the existing system.
- Find the root cause of each prioritized vulnerability in the current system.
- Propose a prevention technique for each root cause of each priority vulnerability in the current system as a security mechanism.
- To avoid the specific vulnerability, try to incorporate the specified security mechanism/technique into the current system.

4. Results

4.1 Vulnerabilities Found

Here, in this project, “Security Root Causes Analysis and Prevention Techniques of Vulnerabilities in MongoDB with Singularity Linux Containers,” we will find root causes of at least 15 vulnerabilities found in Project 1. And we are proposing a prevention technique for each vulnerability.

Number	Vulnerability	Risk
1	OpenVAS / Greenbone Vulnerability Manager Default Credentials	High
2	SSL/TLS: Report Vulnerable Cipher Suites for HTTPS	Medium
3	SSL/TLS: Certificate Signed Using A Weak Signature Algorithm	Medium
4	SSL/TLS: Certificate Expired	Medium

5	CVE-2019-2389 Incorrect scoping of kill operations in MongoDB Server's packaged SysV init scripts	Low
6	MongoDB server accepts connections from unauthorized hosts	Medium
7	MongoDB listens on a non-default port	Low
8	The server only accepts whitelisted hosts/Networks	Medium
9	MongoDB is not exposing its version number	Low
10	TLS/SSL encryption is not enabled	High
11	Authentication is enabled	High
12	CVE-2020-7921 permits a user with valid credentials to bypass IP whitelisting protection mechanisms following administrative action	Low
13	CVE-2021-32039 may be able to access unencrypted user credentials saved by MongoDB Extension for VS Code	Low
14	CVE-2021-32036 Authenticated user may be able to repeatedly invoke the features command where at a high volume may lead to resource depletion, this may result in denial of service and in rare cases could result in id field collisions	High
15	CVE-2020-7925 a specially crafted request can be used to cause a denial-of-service attack	High

Table 1 Vulnerabilities and security risk level

4.2 Vulnerabilities: Classified as High

#10 Vulnerability	**TLS/SSL encryption is not enabled**
Vulnerability Category	Network
Severity (priority)	High
Root Cause	By default, it is disabled
Prevention Technique(s)	To communicate with each other, a TLS/SSL certificate can be set up, and client and server certificates can be signed.
Implementation	sudo nano /etc/mongod.conf *add the following lines* net: ssl: Mode: requireSSL PEMKey: <route-to-cert-file> CAFile: <route-to-ca-file>
Security verification	Check the results of the Mongo Audit security basic scan.

#11 Vulnerability	**Authentication is not enabled**
Vulnerability Category	Application
Severity (priority)	High

Root Cause	Authentication is not enabled, and authentication has not been set up
Prevention Technique(s)	Enable authentication and set up authentication username and password

Implementation	In mongo shell:
	> use admin
	> db.createUser(
	{
	User: “Admin”,
	Pwd: “myNewPassword,
	Roles: [{ role: “userAdminAnyDatabase”, db: “admin” }]
	}
	}
	In linux shell:
	sudo nano /etc/mongod.conf
	Add the following lines
	security:
	authorization: “enabled”

Security verification	Use MongoAudit

#1 Vulnerability	**OpenVAS / Greenbone Vulnerability Manager Default Credentials**
Vulnerability Category	Application
Severity (priority)	High
Root Cause	Basic hardening was not implemented
Prevention Technique(s)	Change default username and password
Implementation	gvmd --user=admin --new-password=new_password
Security verification	Check that default credentials are not used, (OpenVAS)

#14 Vulnerability	**CVE-2021-32036 may result in denial of service and in rare cases could result in id field collisions**
Vulnerability Category	Image Container
Severity (priority)	High
Root Cause	Allocation of Resources Without Limits or Throttling
Prevention Technique(s)	Upgrade to most recent version of MongoDB
Implementation	If version < v5.0.3 ----> sudo apt-get install –y mongodb-org
Security verification	mongod --version

#15 Vulnerability	**CVE-2020-7925 Denial of Service when processing malformed Role names**
Vulnerability Category	Application
Severity (priority)	High
Root Cause	Undefined Behavior for Input to API

Prevention Technique(s)	Upgrade to most recent version of MongoDB
Implementation	If version < v4.4.0-rc12 ----> sudo apt-get install –y mongodb-org
Security verification	mongod --version

4.2 Vulnerabilities: Classified as Medium

#3 Vulnerability	**SSL/TLS: Certificate Signed Using A Weak Signature Algorithm**
Vulnerability Category	Server(Application)
Severity (priority)	Medium
Root Cause	weak SHA-1, MD5, MD4 or MD2 hashing algorithm
Prevention Technique(s)	obtain new SHA-2 signed SSL/TLS certificates
Implementation	?
Security verification	Browsers like chrome do not give you a warning next to url

#4 Vulnerability	**SSL/TLS: Certificate Expired**
Vulnerability Category	Host
Severity (priority)	Server (Application)
Root Cause	The certificate of the remote service expired on 2020-08-20 19:18:24
Prevention Technique(s)	Replace the SSL/TLS certificate by a new one.
Implementation	Obtain a domain validation certificate from your domain register
Security verification	Browsers like chrome do not give you a warning next to url

#2 Vulnerability	**SSL/TLS: Report Vulnerable Cipher Suites for HTTPS**
Vulnerability Category	Server (Application)
Severity (priority)	Medium
Root Cause	SSL/TLS cipher suites accepted by a service where attack vectors exist
Prevention Technique(s)	configuration of these services should be changed so that it does not accept the listed cipher suites anymore
Implementation	If ssl <= 3.0 then disable, if tls <= 1.1 then disable Enable tls 1.2 Enable current ssl ciphers
Security verification	OpenVAS

#8 Vulnerability	**The server does NOT only accept whitelisted hosts/Networks**
Vulnerability Category	Application
Severity (priority)	Medium
Root Cause	Whitelist of acceptable hosts is not configured
Prevention Technique(s)	Configure and enable mongodb whitelist
Implementation	Through UI configure a new IP whitelist instance
Security verification	IPs outside of whitelist cannot access

#6 Vulnerability	MongoDB server accepts connections from unauthorized hosts
Vulnerability Category	application
Severity (priority)	Medium
Root Cause	Guest connections are frequently granted by MongoDB.
Prevention Technique(s)	Pay attention to any inbound connections whose source IP address is associated with database-related apps. It guards against denial-of-service assaults on the server.
Implementation	Through UI configure a new user whitelist instance
Security verification	Re-run the application.

4.3 Vulnerabilities: Classified as low

#7 Vulnerability	MongoDB listens on the default port
Vulnerability Category	Application
Severity (priority)	Low
Root Cause	The MongoDB server is currently listening on default port
Prevention Technique(s)	Change the port of the server so make it harder for external player to find
Implementation	> sudo nano /etc/mongod.conf *write the following lines to the file* net: port: 23987
Security verification	> nc –l 23456

#9 Vulnerability	MongoDB is exposing its version number
Vulnerability Category	Nmap/Mongoaudit
Severity (priority)	Low
Root Cause	No firewall or authentication
Prevention Technique(s)	Put mongodb process behind a firewall and enable authentication
Implementation	*use IP tables to configure what is allowed manually* iptables -A INPUT -s <ip-address> -p tcp --destination-port 27017 -m state --state NEW,ESTABLISHED -j ACCEPT iptables -A OUTPUT -d <ip-address> -p tcp --source-port 27017 -m state --state ESTABLISHED -j ACCEPT
Security verification	MongoAudit

#5 Vulnerability	CVE-2019-2389 Incorrect scoping of kill operations in MongoDB Server's packaged SysV init scripts
Vulnerability Category	Application
Severity (priority)	Low
Root Cause	Incorrect scoping of kill operations in MongoDB Server's packaged SysV init scripts
Prevention Technique(s)	Upgrade to most recent version of MongoDB
Implementation	If version < v4.0.11 ----> sudo apt-get install –y mongodb-org

Security verification	mongod --version

#12 Vulnerability	**CVE-2020-7921 permits a user with valid credentials to bypass IP whitelisting protection mechanisms following administrative action**
Vulnerability Category	Application
Severity (priority)	Low
Root Cause	Improper serialization of internal state in the authorization subsystem in MongoDB Server's authorization subsystem
Prevention Technique(s)	Upgrade to most recent version of MongoDB
Implementation	If version < v4.2.3 ----> sudo apt-get install –y mongodb-org
Security verification	mongod --version

#13 Vulnerability	**CVE-2021-32039 credentials may be used by malicious attackers to perform unauthorized actions**
Vulnerability Category	Application
Severity (priority)	Low
Root Cause	may be able to access unencrypted user credentials saved by MongoDB Extension for VS Code
Prevention Technique(s)	Upgrade to most recent version of MongoDB
Implementation	If version < v0.7.0 ----> sudo apt-get install –y mongodb-org
Security verification	mongod --version

5. Conclusion

Data security is defined as maintaining confidentiality, integrity, and availability. Many threats and problems may be exploited against the MongoDB-based data analysis frameworks housed in Singularity Linux containers. Containerized applications/systems are only secure as the system that they are running on. A possible solution to insecure testbed systems is to not have users deal with them at all. This project is focused on docker/singularity platform deployable tools. For the platform's vulnerabilities, we used Mongoaudit, Nmap , and OpenVAS. It assists us in identifying flaws in current systems and suggesting potential remedies. We have categorized seventeen vulnerabilities into three categories: high, medium, and low severity. We can now identify the most important tools, as well as the fundamental problem and potential solutions. We hope that the solutions described in this project will assist users in securing their systems and making them vulnerable-free.

6. Future Work

The testbed OpenVAS is only able to scan that which is underlying the application (in this case Ubuntu underneath MongoDB). Linking services together is something that I heard Prof Akalanka talk about briefly, which I was not able to implement personally. The solution where we use a sterile testbed is not realistic to set up a general user, those intricacies could be managed by a third party in the cloud. Furthermore, we can try to develop and implement a new or existing security mechanism to avoid a project vulnerability. We can work either on improving authentication with Mongooseas or by encrypting data in transit.

References

1. Bogert, J. (n.d.). *Nmap basics: What is Nmap & how is it used?* Linux Security. Retrieved March 1, 2022, from https://linuxsecurity.com/features/nmap-basics-what-is-nmap-how-is-it-used
2. Mailewa Dissanayaka, A. B., Mengel, S., Gittner, L. A., & Khan, H. (n.d.). *The security assurance of MongoDB in singularity lxcs: An elastic and convenient testbed using Linux containers to explore vulnerabilities*. Texas Tech University Scholars. Retrieved March 2, 2022, from https://scholars.ttu.edu/en/publications/security-assurance-of-mongodb-in-singularity-lxcs-an-elastic-and--2
3. Nmap. (n.d.). Retrieved March 1, 2022, from https://nmap.org/

4. *Open vulnerability assessment scanner*. OpenVAS. (n.d.). Retrieved March 1, 2022, from https://openvas.org/
5. Whalen, I. (n.d.). *Mongoaudit*. MongoDB Tools. Retrieved March 1, 2022, from http://mongodb-tools.com/tool/mongoaudit/
6. https://stack.watch/product/mongodb/mongodb/
7. https://www.cvedetails.com/vulnerability-list.php?vendor_id=12752&product_id=25450&version_id=0&page=1&hasexp=0&opdos=0&opec=0&opov=0&opcsrf=0&opgpriv=0&opsqli=0&opxss=0&opdirt=0&opmemc=0&ophttprs=0&opbyp=0&opfileinc=0&opginf=0&cvssscoremin=0&cvssscoremax=0&year=0&cweid=0&order=1&trc=18&sha=41ff5e176c3da50d346d76bab6095e55a1b8aa8e
8. https://www.digicert.com/faq/sha2/transitioning-to-sha-2.htm
9. https://www.acunetix.com/blog/articles/tls-ssl-cipher-hardening/
10.

Khalid Alghamdi – 100%

William Munnich – 100%