EPITECH.
L'ECOLE DE L'INNOVATION ET DE L'EXPERTISE INFORMATIQUE

INTRUSION TEST REPORT

Debusschere Antoine
Tridon Quentin
PANDRAUD Nicolas
SCHERPEREEL Clement
CHATEAU Nicolas

Summary

Executive

Agreement with the company

Following the request of the company XXX we will carry out safety tests on 4 of their machines, tests will be carried out in "white box" and "black box".
The company provides the machines as well as the application code and certain identifiers in order to carry out the tests.

context

Intrusion tests will be performed on 4 virtual machines under Vmware containing a web application, a database, an API, and several different services under docker

Methodology

Information -taking (phase 1)

The objective of this phase is to search for information about the machines identified.

The result of this search reveals information about the target machines that could be found by a hacker and that could use it for malicious purposes and break into the company's system.

The purpose of finding this information will allow us to visualize and understand how the target system works and the role of each machine,
but also the gateways to start the search for vulnerability on target machines

Example of information sought:

1. **Os of the machine**
2. **Open port**
3. **Service used**
4. **Machine name**
5. **Version of services**

Vulnerability Analysis and Operations (Phase 2)

The objective of this phase is to look for potential vulnerabilities via the information obtained during Phase 1

This phase is more technical, you have to look for vulnerabilities on the services found via the version and once found you have to exploit the security vulnerabilities.

To prove the success of the intrusion

1. confidential information about the target system must be provided
2. Or show that you can make changes directly to the system or application

This phase does not necessarily lead to a complete control of the target machine, but one can find a gateway into the system and recover information or perform "side movements" (escalation of privileges) in order to take control of the system

Post Exploitation (Phase 3)

The objective of this phase is to identify the flaws that we were able to exploit and in deduce the level of risk and the type of attack (network, system, application, ...)

but also to find a correction solution in order to eliminate the flaws in question and thus make the system safer in order to avoid possible data theft, backout, etc., which could cost the company dearly.

Details of information

Information collected

Scan open ports and services on each machine
Tools used: Nmap, Nessus

Machine 1: IP 192.168.3.128, OS: Centos 7

port	service
21/tcp	ftp
22/tcp	ssh
80/tcp	http (Front application)
8080/tcp	http-proxy

Machine 2: IP 192.168.3.131, OS: Centos 7

port	service
21/tcp	ftp
222/tcp	rsh-spx
80/tcp	http (Gitea)

Machine 3: IP 192.168.3.130, OS: Centos 7

port	service
22/tcp	ssh
80/tcp	http (Nagios web portal)
3306/tcp	mysql (bdd web application)

Machine 4: IP 192.168.3.133, OS: Centos 7

port	service
22/tcp	ssh
8000/tcp	http-alt
9000/tcp	portainer (web portal)

List of vulnerabilities

Vulnerability on Machine 1 (web front)

N° 1 - Type Vulnerability: System

Risk: high

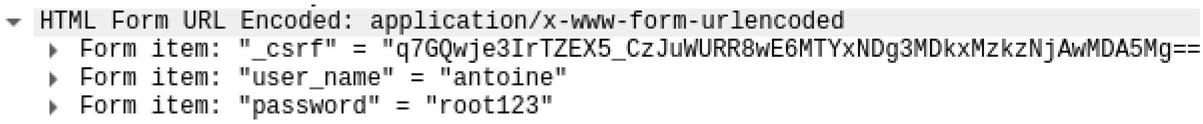
This vulnerability and due to a bad installation and version of the linux grub , the attacker if he can have physical access to the machine can then via the grub reset the user's password "root" and therefore have full access to the machine

N° 2 - Type Vulnerability: Web

Risk: high

Vulnerability to the web login page application

Since the application is not secure in terms of communicating web queries via SSL, an attacker can then intercept the password and username clearly in the network frame of the queries
and thus potentially steal the app's admin account



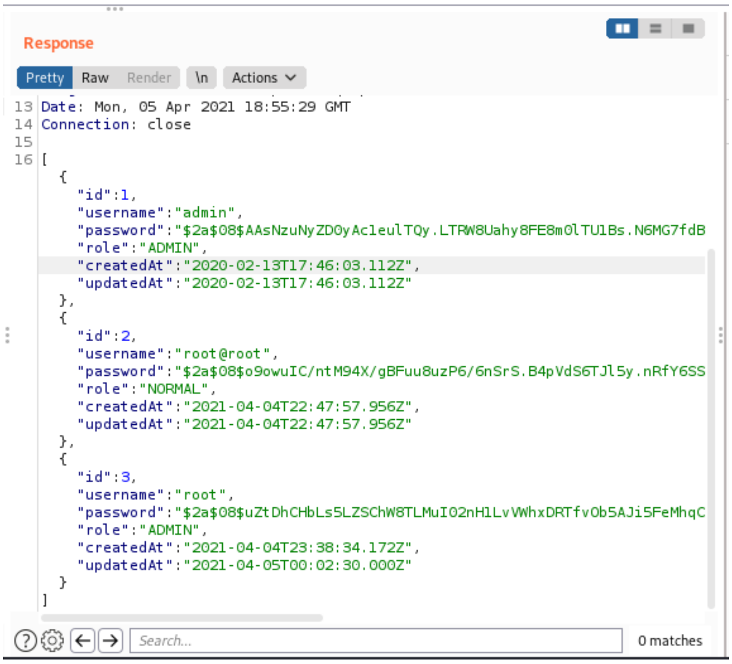
```
▾ HTML Form URL Encoded: application/x-www-form-urlencoded
    ▸ Form item: "_csrf" = "q7GQwje3IrTZEX5_CzJuWURR8wE6MTYxNDg3MDkxMzkzNjAwMDA5Mg==
    ▸ Form item: "user_name" = "antoine"
    ▸ Form item: "password" = "root123"
```

N° 3 - Type Vulnerability: Web

Risk: high

App communication vulnerability between front and back

An attacker can by setting up a proxy intercept all queries that pass between the front and back and modify them in order to retrieve information like the user list in the database



```
Date: Mon, 05 Apr 2021 18:55:29 GMT
Connection: close

[
  {
    "id":1,
    "username":"admin",
    "password":"$2a$08$AAsNzuNyZD0yAcleulTQy.LTRW8Uahy8FE8m0lTU1Bs.N6MG7fdB
    "role":"ADMIN",
    "createdAt":"2020-02-13T17:46:03.112Z",
    "updatedAt":"2020-02-13T17:46:03.112Z"
  },
  {
    "id":2,
    "username":"root@root",
    "password":"$2a$08$o9owuIC/ntM94X/gBFuu8uzP6/6nSrS.B4pVdS6TJl5y.nRfY6SS
    "role":"NORMAL",
    "createdAt":"2021-04-04T22:47:57.956Z",
    "updatedAt":"2021-04-04T22:47:57.956Z"
  },
  {
    "id":3,
    "username":"root",
    "password":"$2a$08$uZtDhCHbLs5LZSChW8TLMuI02nH1LvWWhxDRTfv0b5AJi5FeMhqC
    "role":"ADMIN",
    "createdAt":"2021-04-04T23:38:34.172Z",
    "updatedAt":"2021-04-05T00:02:30.000Z"
  }
]
```

N° 4 - Type vulnerability: password

Risk: high ⚠

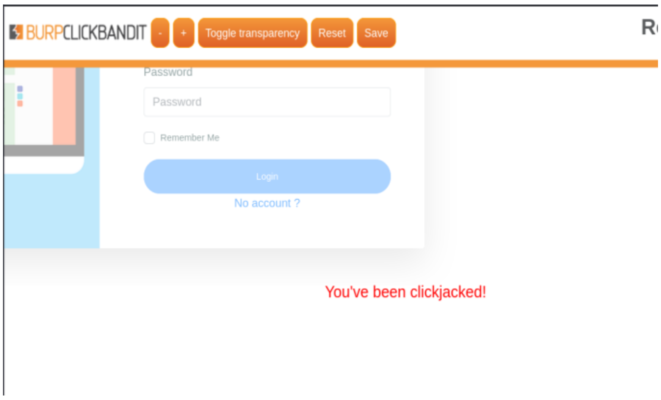
The vulnerability of this vulnerability lies in the complexity of the passwords used by users, because via a brute attackforce an attacker can then find the login of one or more users quite easily

Example for a username "root" and password "root" can be found in a few minutes

N° 5 - Type vulnerability: web

Risk: high ⚠

The vulnerability of this flaw called "Clickjacking" allows an attacker to hijack a user's click in order to redirect it to a malvaating or other site



You've been clickjacked!

Vulnerability on Machine 2 (Gitea)

N° 1 - Type Vulnerability: System

Risk: **high**

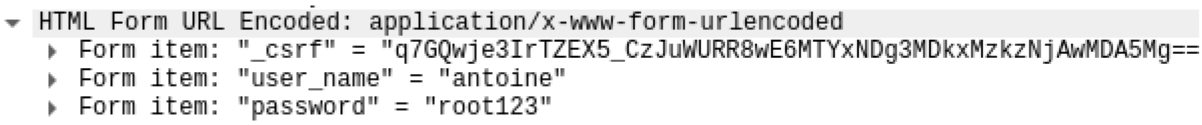
This vulnerability and due to a bad installation and version of the linux grub , the attacker if he can have physical access to the machine can then via the grub reset the user's password "root" and therefore have full access to the machine

N° 2 - Type Vulnerability: Application

Risk: **high**

Vulnerability at Gitea login page

Since the application is not secure in terms of communicating web queries via SSL, an attacker can then intercept the password and username clearly in the network frame of the queries
and thus potentially steal the app's admin account



```
▾ HTML Form URL Encoded: application/x-www-form-urlencoded
  ▸ Form item: "_csrf" = "q7GQwje3IrTZEX5_CzJuWURR8wE6MTYxNDg3MDkxMzkzNjAwMDA5Mg==
  ▸ Form item: "user_name" = "antoine"
  ▸ Form item: "password" = "root123"
```

N° 3 - Type Vulnerability: Application

Risk: **high**

Vulnerability on the rights of the Gitea project
The project share on the Gitea app and accessible in public which leaves the possibility to anyone to recover all the uploader files on it

N° 4 - Type Vulnerability: Web

Risk: high ⚠

Via The Gitea web portal a flaw is present in the url, a striker can then change the username directly in the url and have access to the account without having to log in

e.g. http://192.168.1.34/n0tth3adm1n?tab=activity

Here the attacker just has to replace "n0tth3adm1n" with the name of another user to gain access to the account

N° 5 Type Vulnerability: Web

Risk: high ⚠

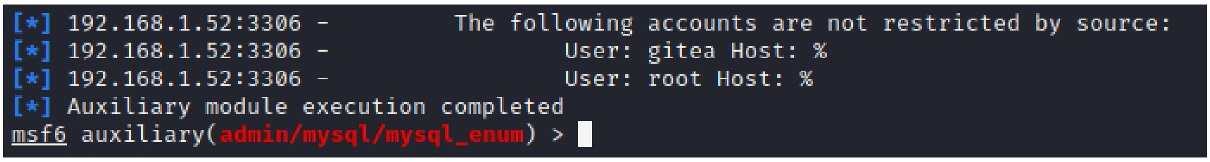
Via the Gitea web portal a flaw is present in the url, an attacker can then list all Gitea users and thus exploit the **number 4** flaw

N° 6 Type Vulnerability: Application

Risk: high ⚠

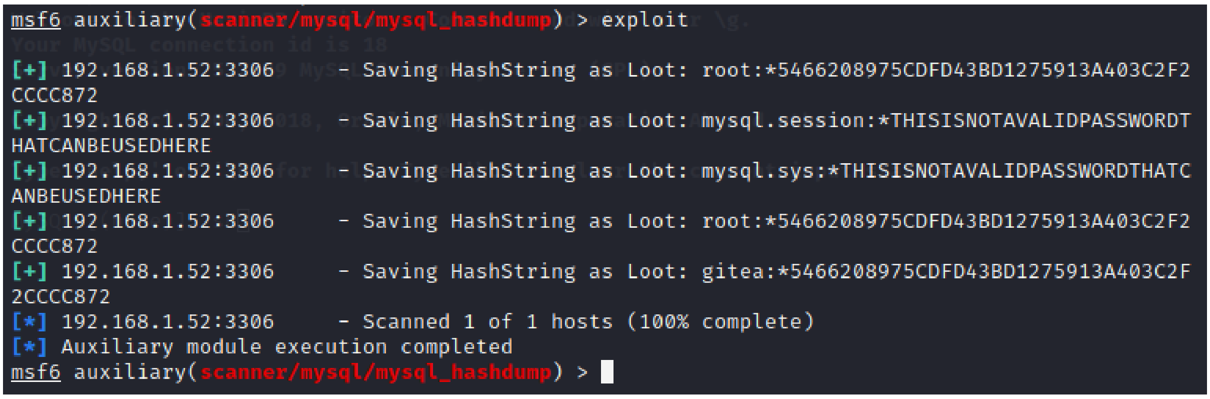
Vulnerability at the database level, an attacker can perform a brute force on the bdd and thus find the root user and therefore use scanner tools in order to collect the list of users in the bdd as well as the corresponding hash that and sha1 thus possibly decipherable depending on the level of complexity of the password

List users in the bdd



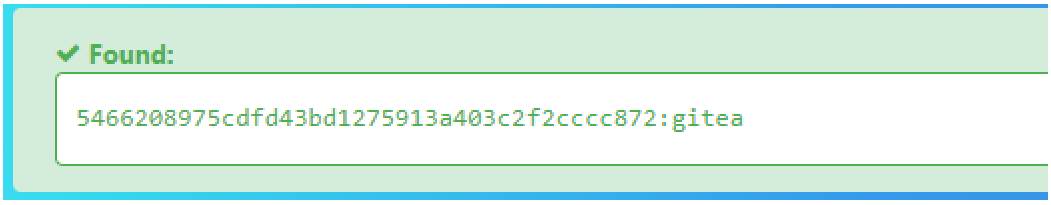
```
[*] 192.168.1.52:3306 -         The following accounts are not restricted by source:
[*] 192.168.1.52:3306 -                 User: gitea Host: %
[*] 192.168.1.52:3306 -                 User: root Host: %
[*] Auxiliary module execution completed
msf6 auxiliary(admin/mysql/mysql_enum) >
```

Show matching hashs



```
msf6 auxiliary(scanner/mysql/mysql_hashdump) > exploit
[+] 192.168.1.52:3306     - Saving HashString as Loot: root:*5466208975CDFD43BD1275913A403C2F2
CCCC872
[+] 192.168.1.52:3306     - Saving HashString as Loot: mysql.session:*THISISNOTAVALIDPASSWORDT
HATCANBEUSEDHERE
[+] 192.168.1.52:3306     - Saving HashString as Loot: mysql.sys:*THISISNOTAVALIDPASSWORDTHATC
ANBEUSEDHERE
[+] 192.168.1.52:3306     - Saving HashString as Loot: root:*5466208975CDFD43BD1275913A403C2F2
CCCC872
[+] 192.168.1.52:3306     - Saving HashString as Loot: gitea:*5466208975CDFD43BD1275913A403C2F
2CCCC872
[*] 192.168.1.52:3306     - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/mysql/mysql_hashdump) >
```

and so decipher the hash and get the password



✓ Found:

```
5466208975cdfd43bd1275913a403c2f2cccc872:gitea
```

N° 7 - Type Vulnerability: System

Risk: **Medium**

Files are present in the machine containing the history of orders placed on the machine and therefore sensitive information

N° 8 - Type Vulnerability: Web

Risk: **high**

Some url paths are accessible directly and make visible configuration files and other sensitive data

```
---- Scanning URL: http://192.168.1.34:80/ ----
http://192.168.1.34:80/admin (CODE:302| SIZE:34)
http://192.168.1.34:80/debug (CODE:200| SIZE:160)
http://192.168.1.34:80/explore (CODE:302| SIZE:37)
http://192.168.1.34:80/issues (CODE:302| SIZE:34)
http://192.168.1.34:80/notifications (CODE:302| SIZE:34)
-----------------
```

Vulnerability on Machine 3 (Nagios and bdd back)

N° 1- Type Vulnerability: System

Risk: **high** ⚠

This vulnerability and due to a bad installation and version of the linux grub , the attacker if he can have physical access to the machine can then via the grub reset the user's password "root" and therefore have full access to the machine

N° 2 - Type Vulnerability: Application

Risk: **Medium** ⚠

Nagios
Opportunity to reset the password nagios directly in the docker nagios

N° 3 - Type Vulnerability: System

Risk: **high** ⚠

nagiosadmin login
password: .-iSp.TUQ0kMm@osE%S

in "/home/admin/m3/nagiosxi-5.5.6/scripts/bootstrap.py"

Vulnerability on Machine 4 (Portainer)

N° 1 - Type Vulnerability: System

Risk: **high** ⚠

This vulnerability and due to a bad installation and version of the linux grub , the attacker if he can have physical access to the machine can then via the grub reset the user's password "root" and therefore have full access to the machine

N° 2 - Type Vulnerability: Application

Risk: **Medium** ⚠

N° 3 - Type Vulnerability: System

Risk: **high** ⚠

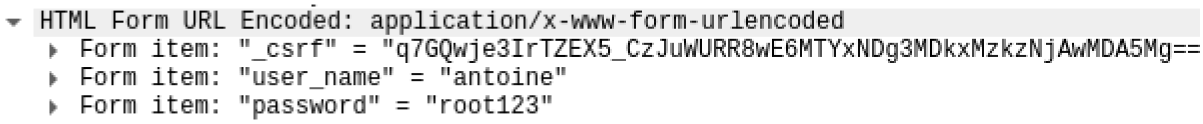
This vulnerability and due to a bad installation and version of the linux grub , the attacker if he can have physical access to the machine can then via the grub reset the user's password "root" and therefore have full access to the machine

N° 4 - Type Vulnerability: Application

Risk: **high** ⚠

Vulnerability at Gitea login page

Since the application is not secure in terms of communicating web queries via SSL, an attacker can then intercept the password and username clearly in the network frame of the queries
and thus potentially steal the app's admin account



```
▾ HTML Form URL Encoded: application/x-www-form-urlencoded
    ▸ Form item: "_csrf" = "q7GQwje3IrTZEX5_CzJuWURR8wE6MTYxNDg3MDkxMzkzNjAwMDA5Mg==
    ▸ Form item: "user_name" = "antoine"
    ▸ Form item: "password" = "root123"
```

N° 5 - Type Vulnerability: Application

Risk: **high** ⚠

Vulnerability on the rights of the Gitea project
The project share on the Gitea app and accessible in public which leaves the possibility to anyone to recover all the uploader files on it

N° 6 - Type Vulnerability: Web

Risk: high

Via The Gitea web portal a flaw is present in the url, a striker can then change the username directly in the url and have access to the account without having to log in

e.g. http://192.168.1.34/n0tth3adm1n?tab=activity

Here the attacker just has to replace "n0tth3adm1n" with the name of another user to gain access to the account

N° 7 - Type Vulnerability: Web

Risk: high

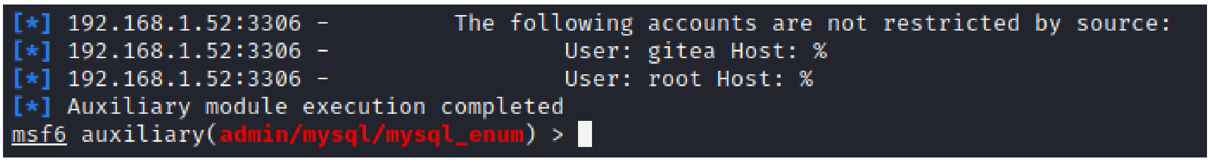
Via the Gitea web portal a flaw is present in the url, an attacker can then list all Gitea users and thus exploit the **number 4** flaw

N° 8 - Type Vulnerability: Application

Risk: high

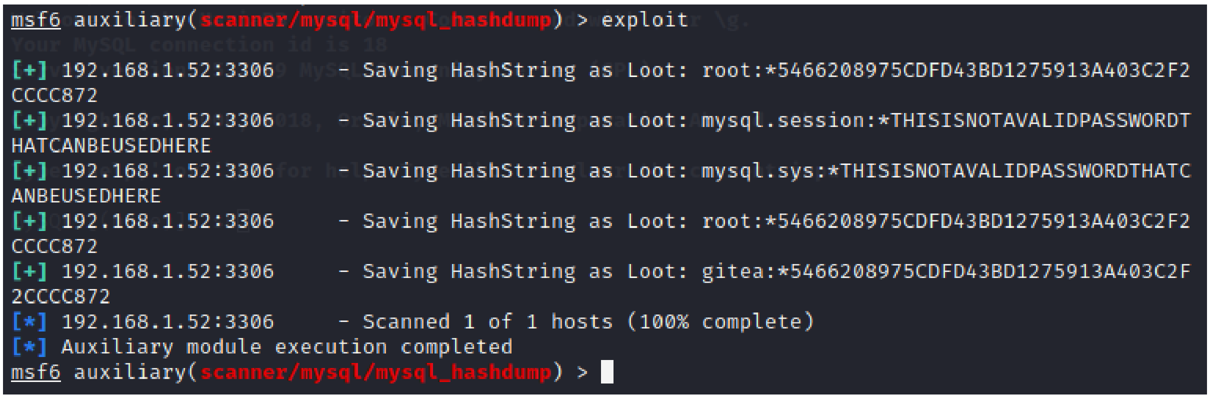
Vulnerability at the database level, an attacker can perform a brute force on the bdd and thus find the root user and therefore use scanner tools in order to collect the list of users in the bdd as well as the corresponding hash that and sha1 thus possibly decipherable depending on the level of complexity of the password

List users in the bdd



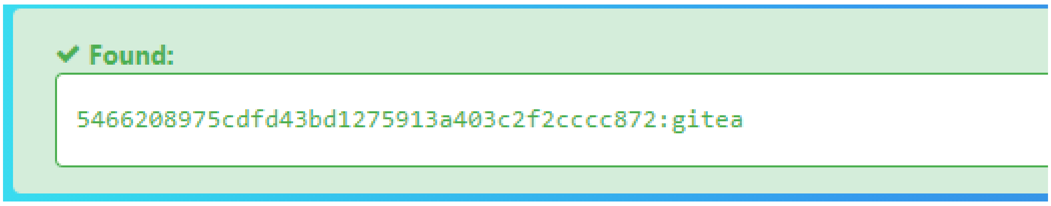
```
[*] 192.168.1.52:3306 -         The following accounts are not restricted by source:
[*] 192.168.1.52:3306 -                 User: gitea Host: %
[*] 192.168.1.52:3306 -                 User: root Host: %
[*] Auxiliary module execution completed
msf6 auxiliary(admin/mysql/mysql_enum) > 
```

Show matching hashs



```
msf6 auxiliary(scanner/mysql/mysql_hashdump) > exploit
[+] 192.168.1.52:3306     - Saving HashString as Loot: root:*5466208975CDFD43BD1275913A403C2F2
CCCC872
[+] 192.168.1.52:3306     - Saving HashString as Loot: mysql.session:*THISISNOTAVALIDPASSWORDT
HATCANBEUSEDHERE
[+] 192.168.1.52:3306     - Saving HashString as Loot: mysql.sys:*THISISNOTAVALIDPASSWORDTHATC
ANBEUSEDHERE
[+] 192.168.1.52:3306     - Saving HashString as Loot: root:*5466208975CDFD43BD1275913A403C2F2
CCCC872
[+] 192.168.1.52:3306     - Saving HashString as Loot: gitea:*5466208975CDFD43BD1275913A403C2F
2CCCC872
[*] 192.168.1.52:3306     - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/mysql/mysql_hashdump) >
```

and so decipher the hash and get the password



✓ Found:

5466208975cdfd43bd1275913a403c2f2cccc872:gitea

N° 9 - Type Vulnerability: System

Risk: Medium

Files are present in the machine containing the history of orders placed on the machine and therefore sensitive information

N° 10 - **Type** Vulnerability: Web

Risk: high

Some url paths are accessible directly and make visible configuration files and other sensitive data

```
---- Scanning URL: http://192.168.1.34:80/ ----
http://192.168.1.34:80/admin (CODE:302| SIZE:34)
http://192.168.1.34:80/debug (CODE:200| SIZE:160)
http://192.168.1.34:80/explore (CODE:302| SIZE:37)
http://192.168.1.34:80/issues (CODE:302| SIZE:34)
http://192.168.1.34:80/notifications (CODE:302| SIZE:34)
-----------------s
```

Vulnerability on Machine 3 (Nagios and bdd back)

N° 1 - Type Vulnerability: System

Risk: high ⚠

This vulnerability and due to a bad installation and version of the linux grub , the attacker if he can have physical access to the machine can then via the grub reset the user's password "root" and therefore have full access to the machine

N° 2 - Type Vulnerability: Application

Risk: Medium ⚠

Nagios
Opportunity to reset the password nagios directly in the docker nagios

N° 3 - Type Vulnerability: System

Risk: high ⚠

nagiosadmin login
password: .-iSp.TUQ0kMm@osE%S

in "/home/admin/m3/nagiosxi-5.5.6/scripts/bootstrap.py"

Vulnerability on Machine 4 (Portainer)

N° 1 - Type Vulnerability: System

Risk: high ⚠

This vulnerability and due to a bad installation and version of the linux grub , the attacker if he can have physical access to the machine can then via the grub reset the user's password "root" and therefore have full access to the machine

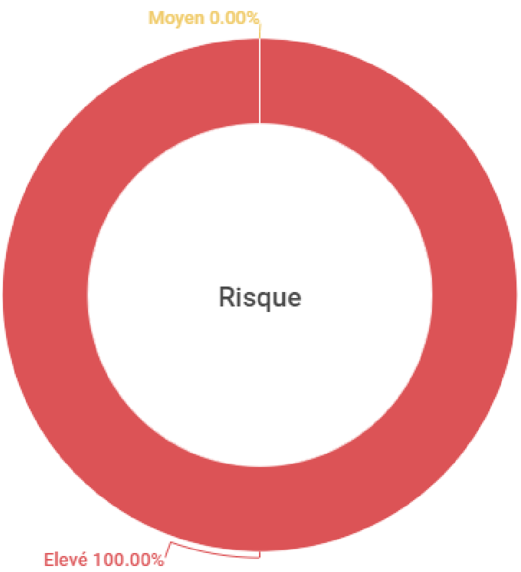
N° 2 - Type Vulnerability: Application

Risk: Medium ⚠

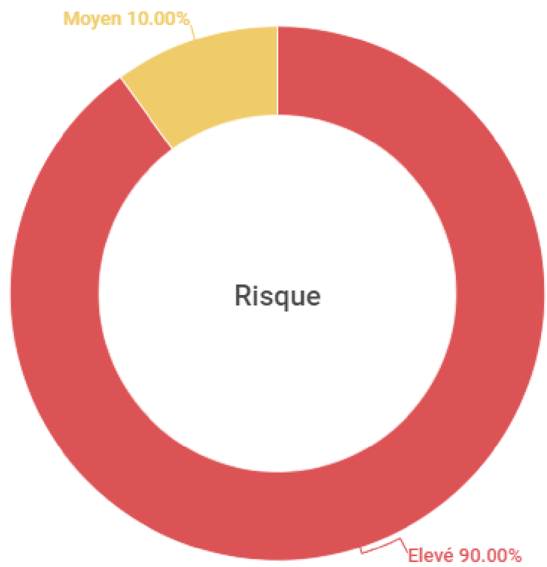
Opportunity to reset the admin password via the DB file of the docker volume containing portainer

Risk level

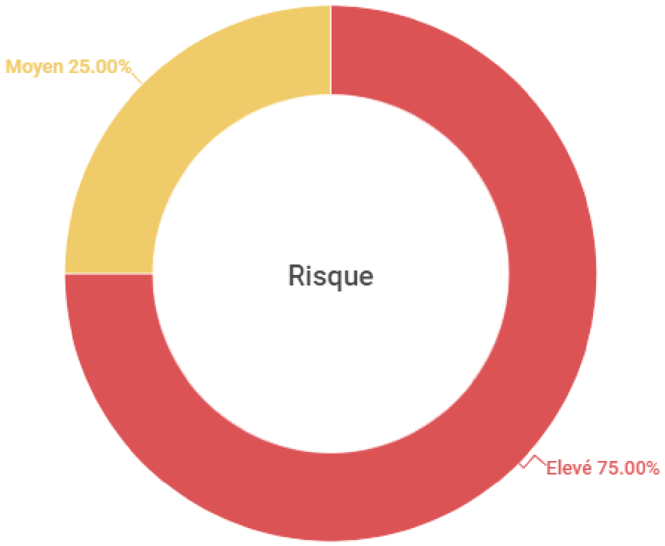
Machine 1 (web front)



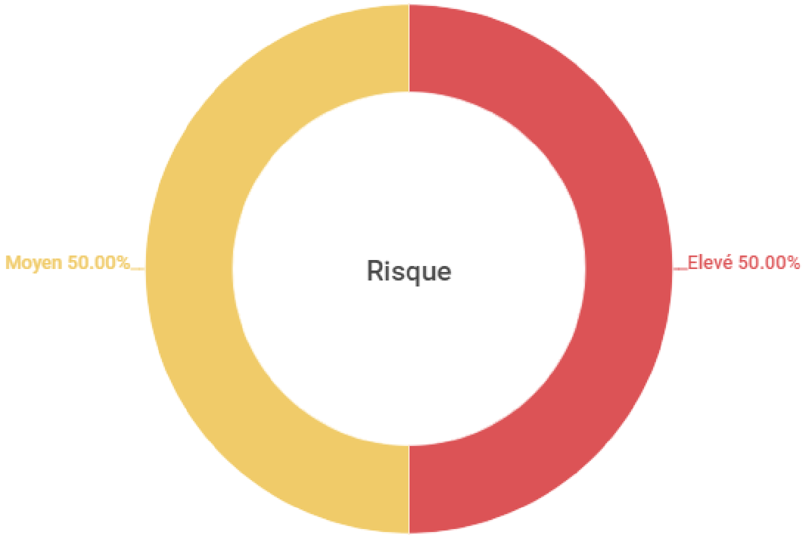
Machine 2 (Gitea)



Machine 3 (Nagios and bdd back)



Machine 4 (Portainer)



Tools used

System used: Kali Linux

1. **Nmap** (Scan of Ports and Services)
2. **Metasploit** (Exploit Search and Exploitation)
3. **Cewl** (Creating Dictionaries for Bruteforce)
4. **ZAPROXY** (Website Analysi)
5. **Wireshark** (Analyze of the network frames)
6. **Nessus** (System Analysis)
7. **DIRB** (Analyze accessible paths from website and web portals)
8. **Nikto** (Analyze of the accessible paths of the website and web portals)
9. **Hydra** (For the bruteforce of the application)
10. **Burpsuite** (Proxy to intercept queries between the front and back)
11. **Burpsuite Clickbandit** (Clickjacking Fault Operation)

Recommendations

Here are our recommendations to fix the security flaws found on the machines in your environment

Fixes on Machine 1 (web front)

No. 1

System update to Centos 8 or reinstallation of centos 7 with another version of GRUB

No. 2

Setting up the 443 SSL protocol to encrypt queries and make them unreadable

No. **3**

Setting up the 443 SSL protocol to encrypt queries and make them unreadable

No. **4**

Increased level of password complexity to make brute force almost impossible

No. **5**

Setting up a server like nginx for the operation of the application with a setting

```
add_header X-Frame-Options "SAMEORIGIN";
```

This prevents the browser from running code

Fixes on Machine 2 (Gitea)

No. **1**

System update to Centos 8 or reinstallation of centos 7 with another version of GRUB

No. **2**

For 2 to 5 cracks for Gitea

Gitea is not sure level security is has a lot of flaws for this it must replace it with a Gitlab which is much safer and more often updated against security flaws

No. **3**

Increased level of password complexity to make brute force almost impossible

No. **4**

Delete files with the history of orders placed on the machine

No. 5

Setting up a robot.txt that hides pages from search engines, but setting up Gitlab corrects the problem

No. 6

Same solution as the 2.

No. 7

Same solution as the 2.

No. 8

Same solution as the 2.

Fixes on Machine 3 (Nagios and bdd back)

No. **1**

System update to Centos 8 or reinstallation of centos 7 with another version of GRUB

No. **2**

Make docker commands executable only by root or admin user

No. **3**

Removing the configuration file boostrap.py featuring the admin password or variabilising the password and placing it in a .near file at the root of the user admin of the machine

Fixes on Machine 4 (Portainer)

No. **1**

System update to Centos 8 or reinstallation of centos 7 with another version of GRUB

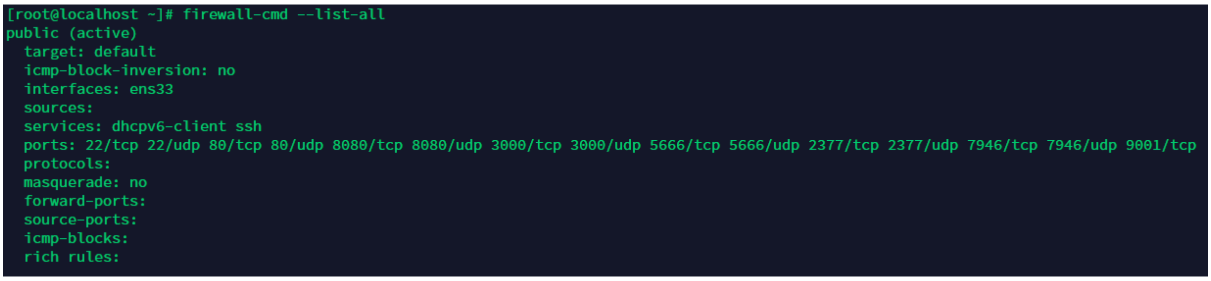
No. **2**

Hide the DB file in the root so that no user can come and edit it

Firewall Configuration

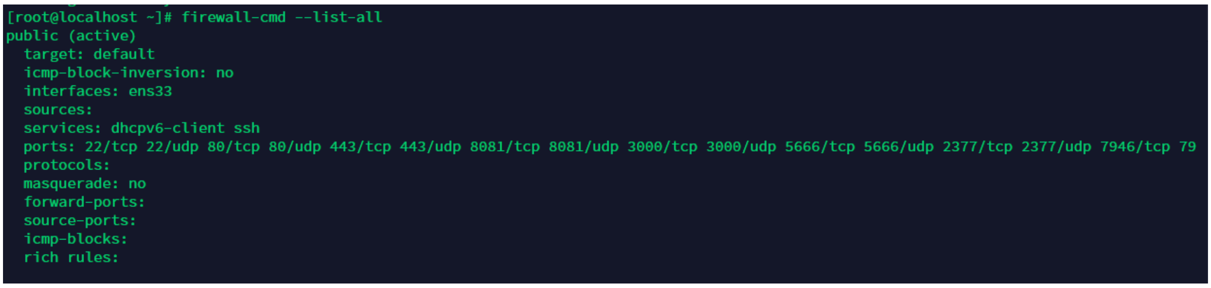
To avoid the maximum of risks, we decide to configure firewalls on each VM to protect them against attacks.
With this protection, all the useless ports are closed. And we can avoid potential attacks on open ports for nothing.

Machine 1



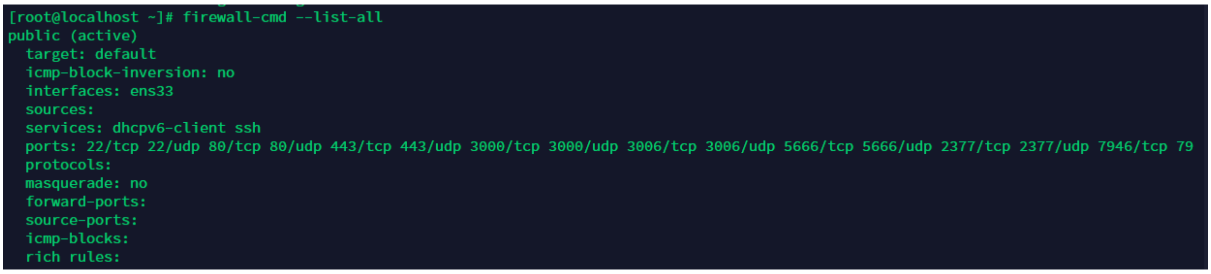
```
[root@localhost ~]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens33
  sources:
  services: dhcpv6-client ssh
  ports: 22/tcp 22/udp 80/tcp 80/udp 8080/tcp 8080/udp 3000/tcp 3000/udp 5666/tcp 5666/udp 2377/tcp 2377/udp 7946/tcp 7946/udp 9001/tcp
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Machine 2



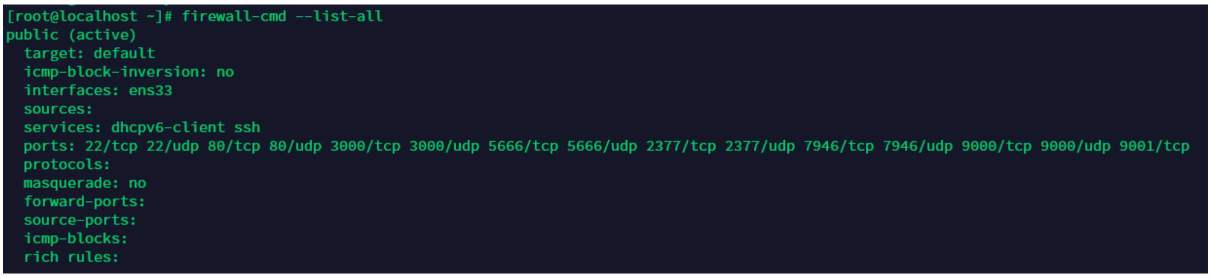
```
[root@localhost ~]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens33
  sources:
  services: dhcpv6-client ssh
  ports: 22/tcp 22/udp 80/tcp 80/udp 443/tcp 443/udp 8081/tcp 8081/udp 3000/tcp 3000/udp 5666/tcp 5666/udp 2377/tcp 2377/udp 7946/tcp 79
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Machine 3



```
[root@localhost ~]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens33
  sources:
  services: dhcpv6-client ssh
  ports: 22/tcp 22/udp 80/tcp 80/udp 443/tcp 443/udp 3000/tcp 3000/udp 3006/tcp 3006/udp 5666/tcp 5666/udp 2377/tcp 2377/udp 7946/tcp 79
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Machine 4



```
[root@localhost ~]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens33
  sources:
  services: dhcpv6-client ssh
  ports: 22/tcp 22/udp 80/tcp 80/udp 3000/tcp 3000/udp 5666/tcp 5666/udp 2377/tcp 2377/udp 7946/tcp 7946/udp 9000/tcp 9000/udp 9001/tcp
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Fail2Ban

We put in place a Fail2ban on each VM, it’s a prevention framework against intrusion. We wrote a script to deploy it easily on VM’s.

```
#!/bin/bash
sudo yum install -y epel-release
sudo yum install -y fail2ban
sudo systemctl enable fail2ban

touch /etc/fail2ban/jail.local

echo [DEFAULT] > /etc/fail2ban/jail.local
echo   >> /etc/fail2ban/jail.local
echo bantime = 3600 >> /etc/fail2ban/jail.local
echo   >> /etc/fail2ban/jail.local
echo findtime = 600 >> /etc/fail2ban/jail.local
echo maxretry = 3 >> /etc/fail2ban/jail.local
echo   >> /etc/fail2ban/jail.local
echo ignoreip = 127.0.0.1/8 192.168.0.1/24 >> /etc/fail2ban/jail.local
echo   >> /etc/fail2ban/jail.local
echo action = '%(action_mwl)s' >> /etc/fail2ban/jail.local
echo   >> /etc/fail2ban/jail.local
echo [sshd] >> /etc/fail2ban/jail.local
echo enabled = true >> /etc/fail2ban/jail.local

sudo systemctl restart fail2ban
sudo fail2ban-client status
sudo fail2ban-client status sshd
```