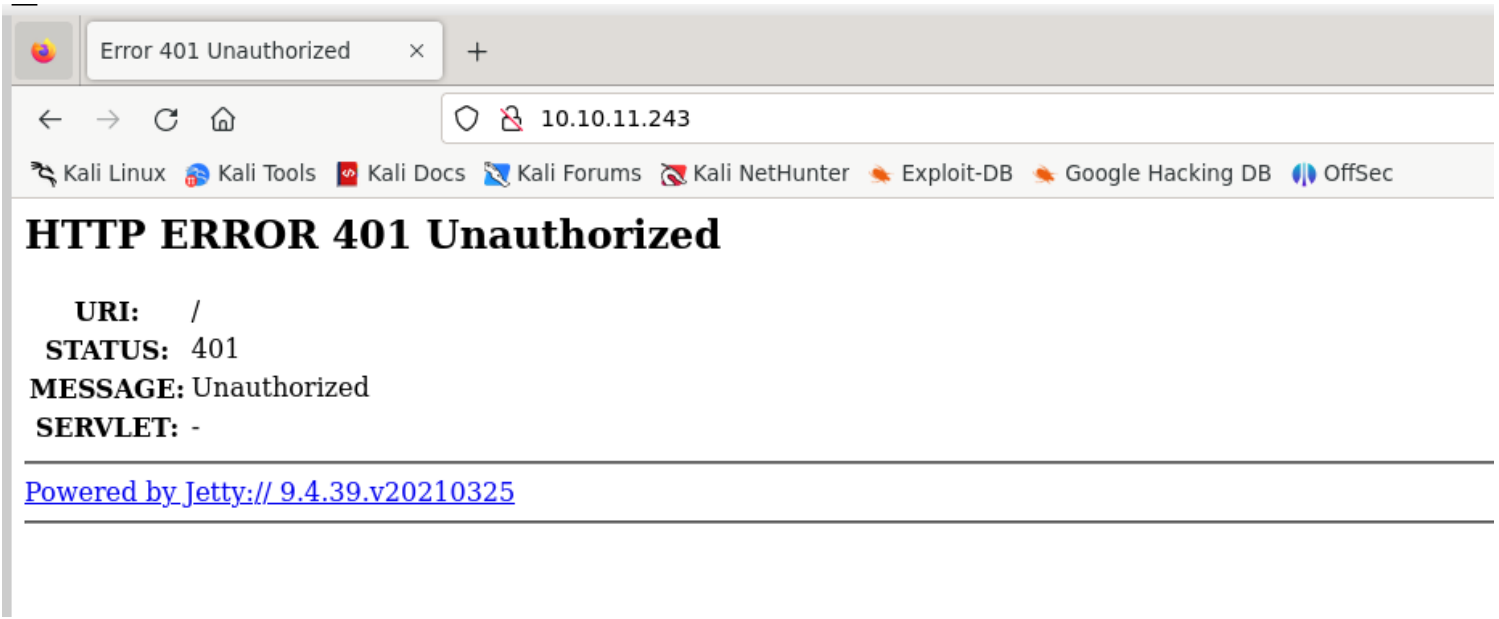
Information Gathering

1) Found open ports

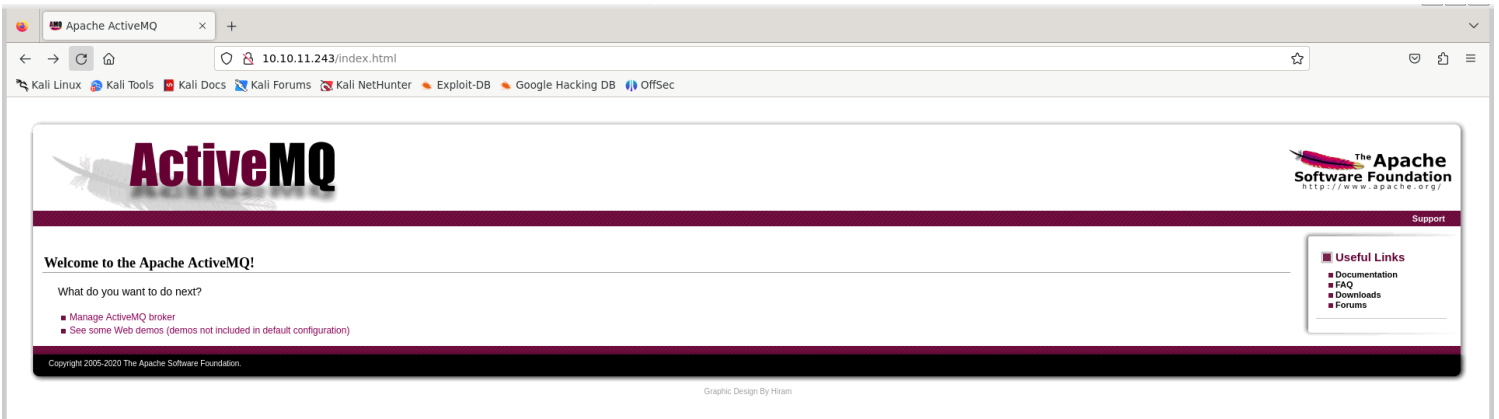
```
┌──(vigneswar㉿VigneswarPC)-[~]
└─$ nmap 10.10.11.243 --open
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-11-17 13:43 IST
Nmap scan report for 10.10.11.243
Host is up (0.24s latency).
Not shown: 509 closed tcp ports (conn-refused), 489 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 110.78 seconds
```

```
┌──(vigneswar㉿VigneswarPC)-[~]
└─$ nmap 10.10.11.243 --open -p61616 -sV -sC
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-11-17 14:28 IST
Nmap scan report for 10.10.11.243
Host is up (0.25s latency).

PORT      STATE SERVICE  VERSION
61616/tcp open  apachemq ActiveMQ OpenWire transport
| fingerprint-strings:
|   NULL:
|     ActiveMQ
|     TcpNoDelayEnabled
|     SizePrefixDisabled
|     CacheSize
|     ProviderName
|     ActiveMQ
|     StackTraceEnabled
|     PlatformDetails
|     Java
|     CacheEnabled
|     TightEncodingEnabled
|     MaxFrameSize
|     MaxInactivityDuration
|     MaxInactivityDurationInitalDelay
|     ProviderVersion
|_    5.15.15
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.c
```

2) Found web page requiring authentication



3) Logged in with admin:admin



Apache ActiveMQ

Computer program

Apache ActiveMQ is an open source message broker written in Java together with a full Java Message Service client. It provides "Enterprise Features" which in this case means fostering the communication from more than one client or server. Wikipedia

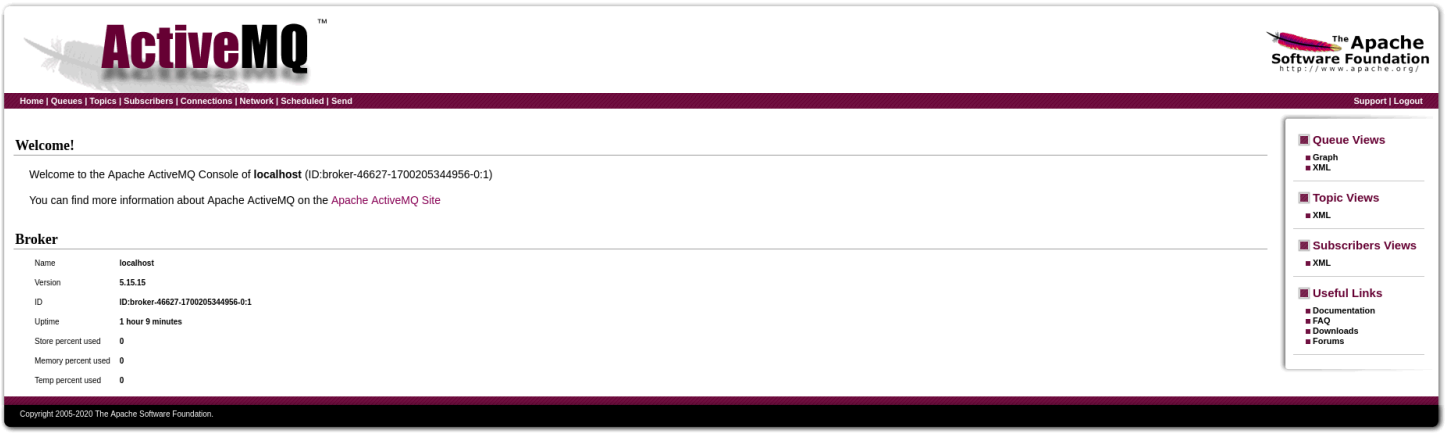
Programming language: Java

Developer: Apache Software Foundation

License: Apache License 2.0

Stable release: 6.0.0 / 14 November 2023; 3 days ago

4) Found more info

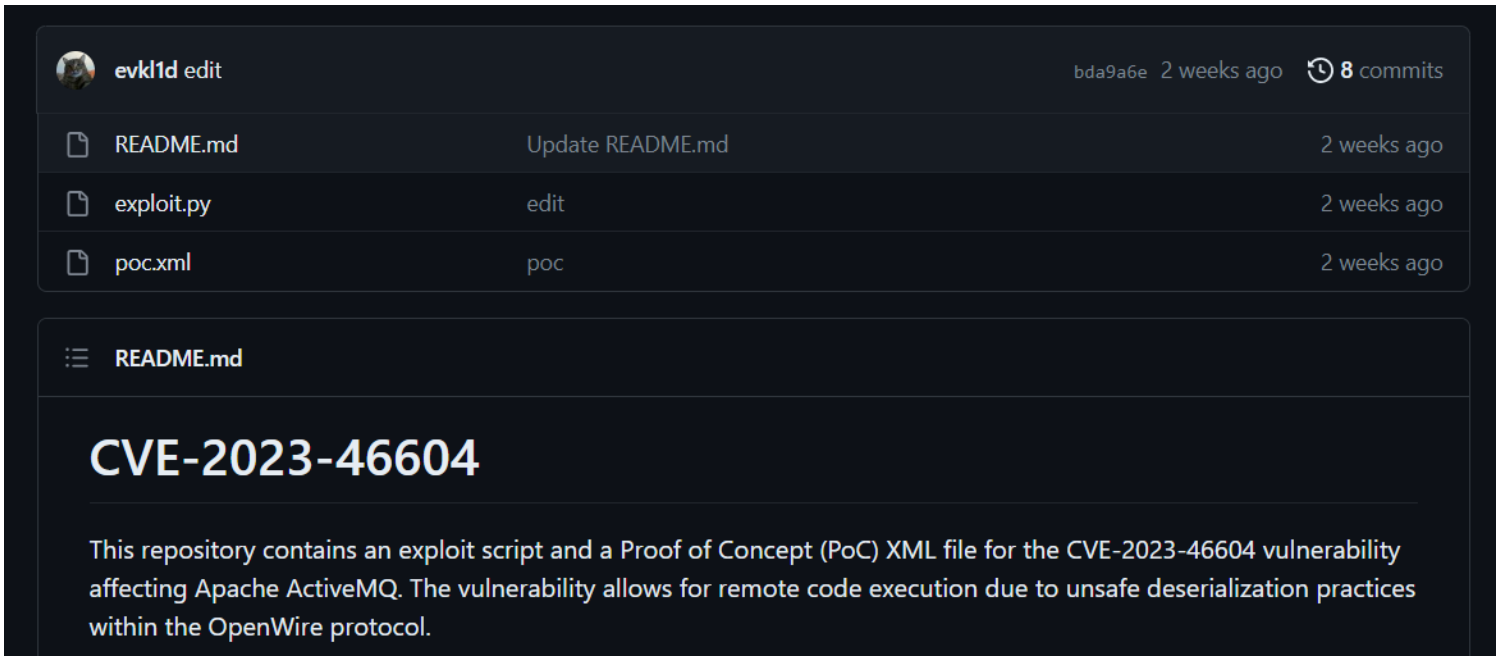


Vulnerability Assessment

1) The version of activeMQ is vulnerable

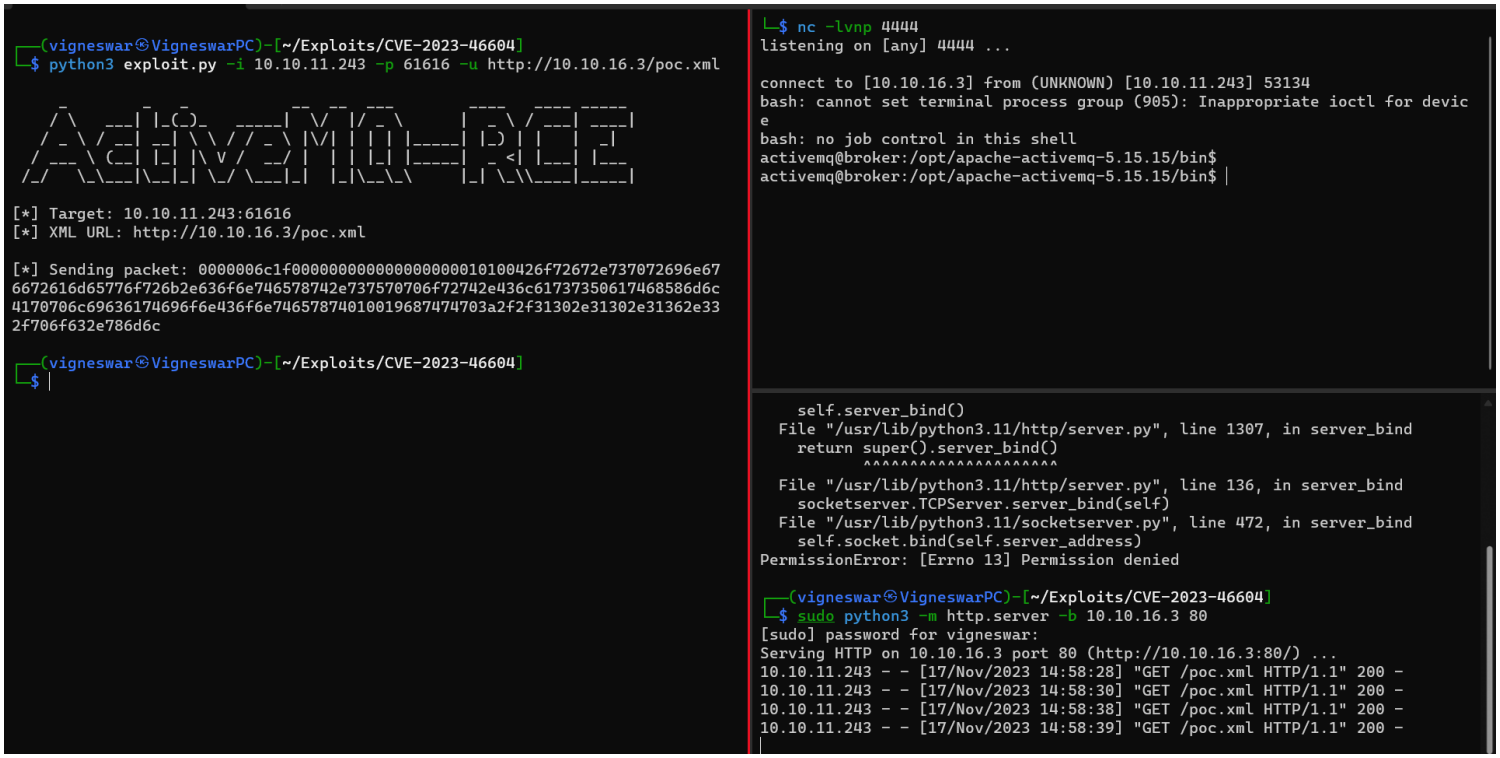
```
Affected versions:

- Apache ActiveMQ 5.18.0 before 5.18.3
- Apache ActiveMQ 5.17.0 before 5.17.6
- Apache ActiveMQ 5.16.0 before 5.16.7
- Apache ActiveMQ before 5.15.16
- Apache ActiveMQ Legacy OpenWire Module 5.18.0 before 5.18.3
- Apache ActiveMQ Legacy OpenWire Module 5.17.0 before 5.17.6
- Apache ActiveMQ Legacy OpenWire Module 5.16.0 before 5.16.7
- Apache ActiveMQ Legacy OpenWire Module 5.8.0 before 5.15.16

Description:

The Java OpenWire protocol marshaller is vulnerable to Remote Code Execution. This vulnerability may allow a remote attacker with network access to either a Java-based OpenWire broker or client to run arbitrary
shell commands by manipulating serialized class types in the OpenWire protocol to cause either the client or the broker (respectively) to instantiate any class on the classpath.

Users are recommended to upgrade both brokers and clients to version 5.15.16, 5.16.7, 5.17.6, or 5.18.3 which fixes this issue.

This issue is being tracked as AMQ-9370

References:

https://activemq.apache.org/security-advisories.data/CVE-2023-46604
https://activemq.apache.org/
https://www.cve.org/CVERecord?id=CVE-2023-46604
https://issues.apache.org/jira/browse/AMQ-9370
```

2) Found a PoC



Exploitation

1) Exploited the vulnerability and got the shell



2) Got user flag

```
activemq@broker:~$ cat user.txt
9133986fee2fe376137e0731e049977b
activemq@broker:~$
```

Privilege Escalation

1) Checked sudo permissions

```
activemq@broker:~$ sudo -l
Matching Defaults entries for activemq on broker:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin,
    use_pty

User activemq may run the following commands on broker:
    (ALL : ALL) NOPASSWD: /usr/sbin/nginx
activemq@broker:~$
```

2) Researched nginx configs

```
activemq@broker:~$ nginx -h
nginx version: nginx/1.18.0 (Ubuntu)
Usage: nginx [-?hvVtTq] [-s signal] [-c filename] [-p prefix] [-g directives]

Options:
  -?,-h         : this help
  -v            : show version and exit
  -V            : show version and configure options then exit
  -t            : test configuration and exit
  -T            : test configuration, dump it and exit
  -q            : suppress non-error messages during configuration testing
  -s signal     : send signal to a master process: stop, quit, reopen, reload
  -p prefix     : set prefix path (default: /usr/share/nginx/)
  -c filename   : set configuration file (default: /etc/nginx/nginx.conf)
  -g directives : set global directives out of configuration file
```

nginx consists of modules which are controlled by directives specified in the configuration file. Directives are divided into simple directives and block directives. A simple directive consists of the name and parameters separated by spaces and ends with a semicolon (`;`). A block directive has the same structure as a simple directive, but instead of the semicolon it ends with a set of additional instructions surrounded by braces (`{` and `}`). If a block directive can have other directives inside braces, it is called a context (examples: events, http, server, and location).

- ngx_http_core_module
- ngx_http_access_module
- ngx_http_addition_module
- ngx_http_api_module
- ngx_http_auth_basic_module
- ngx_http_auth_jwt_module
- ngx_http_auth_request_module
- ngx_http_autoindex_module
- ngx_http_browser_module
- ngx_http_charset_module
- ngx_http_dav_module
- ngx_http_empty_gif_module
- ngx_http_f4f_module
- ngx_http_fastcgi_module
- ngx_http_flv_module
- ngx_http_geo_module
- ngx_http_geoip_module
- ngx_http_grpc_module
- ngx_http_gunzip_module
- ngx_http_gzip_module
- ngx_http_gzip_static_module
- ngx_http_headers_module
- ngx_http_hls_module
- ngx_http_image_filter_module
- ngx_http_index_module
- ngx_http_internal_redirect_module
- ngx_http_js_module

The `ngx_http_dav_module` module is intended for file management automation via the WebDAV protocol. The module processes HTTP and WebDAV methods PUT, DELETE, MKCOL, COPY, and MOVE.

This module is not built by default, it should be enabled with the `--with-http_dav_module` configuration parameter.

> WebDAV clients that require additional WebDAV methods to operate will not work with this module.

Example Configuration

```
location / {
    root                  /data/www;

    client_body_temp_path /data/client_temp;

    dav_methods PUT DELETE MKCOL COPY MOVE;

    create_full_put_path  on;
    dav_access            group:rw  all:r;

    limit_except GET {
        allow 192.168.1.0/32;
        deny  all;
    }
}
```

3) Used it to upload passwd file to remove password for root