



An Exploration into Heterogeneity

East Tennessee State University

Department of Computing

CSCI 4417/5417

Spring 2016

Jack Ramsey, Lecturer

Introduction

Modern networks are often heterogeneous, meaning that they are a combination of multiple machines often using different operating systems. The converse of heterogeneous is homogeneous, an environment that consists of server/client machines all running under the same operating system. Your lab assignment this week is a little different from those in the past. Your job, using AWS instances, is to use the Windows Server domain controller you created earlier this semester with AD DS and DNS roles and join a Red Hat Linux (RHL) instance to the *lastname.loc* domain you have already created.

Note

In the past, this lab has met with mixed success for students. These instructions have been refined several times in light of the historical fact that Windows and *nix systems don't like to play nice with each other. It is almost universally acknowledged, however, that joining a *nix machine to a Windows-administered domain is far easier than the reverse. So that's what we'll be attempting to do today. However, if you are unsuccessful, after an honest attempt, simply write your report and document the difficulties you encountered thoroughly to receive full credit. The point at which we've encountered difficulties is authentication, so everyone should be able to join the Linux machine to the domain successfully.

Make sure you take screen shots where appropriate as you wade through the instructions.

Purpose

Join a Linux instance to a Windows Active Directory domain and configure it to use Windows AD DS for user authentication.

Materials

- Windows Domain Controller
- Amazon Red Hat Linux instance

Procedure

Launch an Red Hat Linux Instance

1. Begin by starting your Windows Domain Controller
2. While the domain controller is spooling up, launch a new Linux instance
3. For this lab, we will be using an AWS Red Hat Linux instance instead of Ubuntu

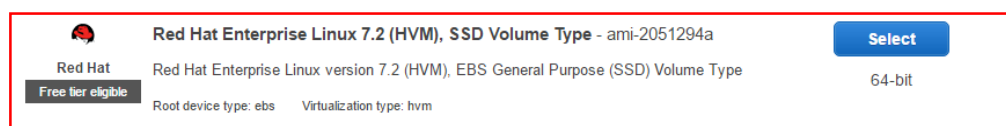


Figure 1: Select the Amazon Red Hat Enterprise Linux image

4. (Step 2) Make sure that 't2.micro' (Free tier) is selected
5. (Step 3) Choose your VPC and the first public subnet (you don't have to assign a static private IP address, if you don't want to, but it makes things easier. In these instructions, I used 173.1.1.27)

6. (Step 4) Click 'Next: Tag Instance'
7. (Step 5) Name your instance *lastnameRHL-ADDS-01*
8. (Step 6) Use the same security group we've been using this semester (*lastnameSG*)

Step 6: Configure Security Group

☒ Select an **existing** security group

Security Group ID	Name	Description	Actions
<input type="checkbox"/> sg-dd9ac4a4	default	default VPC security group	Copy to new
<input type="checkbox"/> sg-01098179	PrivateSG	Security Group for Private Subnet	Copy to new
<input type="checkbox"/> sg-b1f297c9	RamseyNATSG	Security Group for Private NAT-served instances	Copy to new
<input checked="" type="checkbox"/> sg-5cd48a25	RamseySG	Security Group for Ramsey public subnet instances	Copy to new

Inbound rules for sg-5cd48a25 (Selected security groups: sg-5cd48a25)

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ
HTTP	TCP	80	0.0.0.0/0
All traffic	All	All	sg-b1f297c9 (RamseyNATSG)
All traffic	All	All	173.1.0.0/16
SSH	TCP	22	0.0.0.0/0
Custom TCP Rule	TCP	10000	0.0.0.0/0
DNS (TCP)	TCP	53	0.0.0.0/0
RDP	TCP	3389	0.0.0.0/0

Figure 2: Choosing a Security Group

9. Review and Launch the instance

Connect to the Amazon Red Hat Linux Instance

10. Use Remote Desktop to connect to your Windows Domain Controller
11. You should have PuTTY on your domain controller. If not copy and paste it (or download and install it, though with Internet Explorer on AWS, that's a *real* pain) and your key (4417key.ppk) to the domain controller
12. For this lab, we'll connect to the Linux box via the Windows box. So, use PuTTY from your domain controller to connect to the Linux instance using the Linux instance's private IP address
13. The login user name for Amazon Linux instances is "ec2-user" (without the quotes)

Modify /etc/hosts

14. We need to edit the `/etc/hosts` file to make the new instance point to the Windows Active Directory Domain Controller
15. With Amazon Red Hat Linux (RHL), the nano editor isn't installed by default. You can use `vi` or `vim` if you like. Alternatively, you can install nano by entering

```
sudo yum install -y nano # yum is the RHL version of Ubuntu's apt-get
```

(Note: you don't have to enter the '#' or anything after it. It's just a comment)

16. Note the computer name and private IP address of your domain controller:

```

      Hostname : RAMSEYDC01
      Instance ID : i-e24a021b
      Public IP Address : 54.173.33.41
      Private IP Address : 173.1.1.20
      Availability Zone : us-east-1e
      Instance Size : t2.micro
      Architecture : AMD64

```

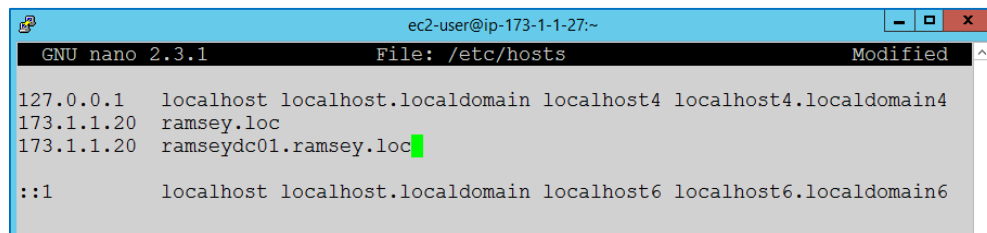
Figure 3: Domain controller's name & IP

17. Enter `sudo nano /etc/hosts`
18. Add the following two lines to `/etc/hosts`, between the localhost (127.0.0.1) line and the IPv6 (:::1 -- anyone know what ':::1' means in IPv6-speak?) lines:

```

173.1.1.20  lastname.loc
173.1.1.20  dc-name.lastname.loc

```



```

ec2-user@ip-173-1-1-27:~
GNU nano 2.3.1      File: /etc/hosts      Modified
127.0.0.1  localhost localhost.localdomain localhost4 localhost4.localdomain4
173.1.1.20  ramsey.loc
173.1.1.20  ramseydc01.ramsey.loc
:::1       localhost localhost.localdomain localhost6 localhost6.localdomain6

```

Figure 4: Modifying the RHL `/etc/hosts` file

Where *lastname* = your last name, and

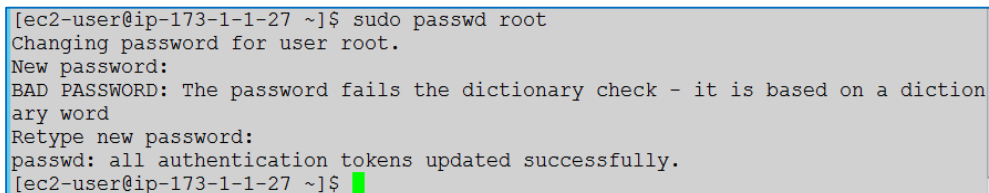
dc-name.lastname = the name of your domain controller - dot - your domain name, i.e., the fully qualified domain name (FQDN)

Note: Case is very important here! Your computer name and domain name should be all lowercase letters!

19. Now, let's give the root user a password (we'll enable password logins momentarily). Enter:

```
sudo passwd root
```

20. When prompted, enter 'Passw0rd!' and confirm (Annoyingly, Linux doesn't provide any feedback regarding how many characters you've typed, which, if your typing skills are less than stellar - as mine are - can sometimes lead to problems. So, type in the passwords carefully!). Linux will very helpfully warn you that your chosen password stinks (and it does). But for purposes of this lab, it's ok



```

[ec2-user@ip-173-1-1-27 ~]$ sudo passwd root
Changing password for user root.
New password:
BAD PASSWORD: The password fails the dictionary check - it is based on a dictionary word
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-173-1-1-27 ~]$

```

Figure 5: Adding a root user password

Modify /etc/resolv.conf

21. Now we need to modify RHL's /etc/resolv.conf file. The resolv.conf file is a computer file used by many Linux systems to configure the system's Domain Name System (DNS) resolver. Recent versions of Ubuntu prohibit modifying this file by hand; it is automatically generated by the Network Manager
22. Enter `sudo nano /etc/resolv.conf`
23. Delete the 'search ec2.internal' and 'nameserver 173.1.1.2' lines replace them with:

```
search ramsey.loc
nameserver 173.1.1.20
```

This instructs the system to use our domain as its search space and the domain controller as the authoritative domain name server (remember what that is?). If we had multiple DNS servers, as we would in a production environment, we could list multiple servers on the 'nameserver' line, e.g.,

```
nameserver 173.1.1.20 173.1.1.21 173.1.1.22
```

But we currently have just one, so stick with

```
nameserver 173.1.1.20
```

(Make sure you're using the private IP address of your domain controller! If, for some reason, you assigned a different IP address to your domain controller back when we did the Active Directory lab, be sure to use that IP address in place of x.x.x.20)

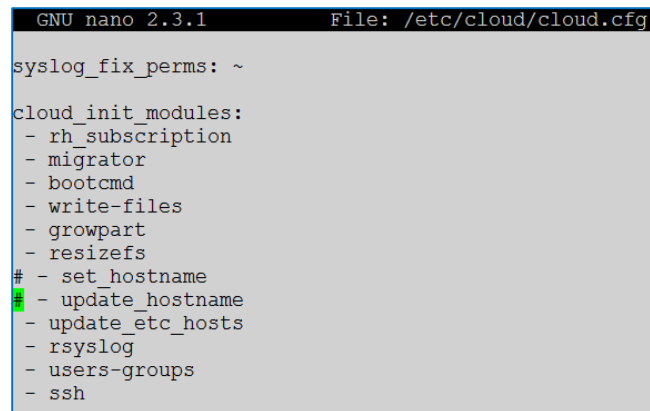
24. Save your changes and exit the file (Ctrl-x, 'y', Enter)
25. Another AWS idiosyncrasy: AWS RHL instances, by default, automatically rewrite the /etc/resolv.conf file when the machine boots up. This is not what we want. A quick fix is

```
sudo chattr +i /etc/resolv.conf
```

This will prevent anyone from overwriting the resolv.conf file (even root!)

26. Let's also change the instance's host name to make it easier to identify in the domain controller's Active Directory Users and Computers application. Amazon, naturally, makes this a tad bit difficult to do
27. First, edit the /etc/cloud/cloud.cfg file. Comment out '- set_hostname' and '- update_hostname' under 'cloud_init_modules:'

```
sudo nano /etc/cloud/cloud.cfg
```

A terminal window showing the contents of the file /etc/cloud/cloud.cfg. The window title is 'GNU nano 2.3.1' and the file path is '/etc/cloud/cloud.cfg'. The content of the file is as follows:

```
syslog_fix_perms: ~
cloud_init_modules:
- rh_subscription
- migrator
- bootcmd
- write-files
- growpart
- resizefs
# - set_hostname
- update_hostname
- update_etc_hosts
- rsyslog
- users-groups
- ssh
```

Figure 6: Modify /etc/cloud/cloud.cfg

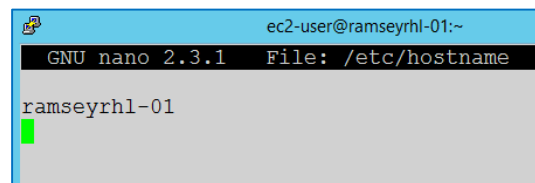
Save and exit. Then enter

```
sudo nano /etc/hostname
```

Delete the text (i.e., ip-173-1-1-27) and replace it with

```
lastnamerhl-01
```

Note: that's your last name, followed by the letters 'r,' 'h,' and 'l,' not the number one '1'. The last two digits are zero ('0') and one ('1')

A terminal window showing the contents of the file /etc/hostname. The window title is 'ec2-user@ramseyrhl-01:~' and the file path is '/etc/hostname'. The content of the file is as follows:

```
ramseyrhl-01
```

Figure 7: Changing the hostname

Save, and exit

28. For the host name change to take effect, you have to reboot the server. Enter

```
sudo reboot
```

29. It isn't necessary to restart PuTTY. Simply wait a few minutes. Then right-click on the terminal's menu bar and select 'Restart Session'. Once the instance is back up and running, you can log back in as 'ec2-user' (You may have to restart the session a couple of times, if you're impatient and do it too soon. I did).

Install Additional Software

30. There are a number of additional software packages that need to be installed. The first is called 'realmd.' Enter

```
sudo yum install -y realmd
```

```
[ec2-user@ip-173-1-1-27 ~]$ sudo yum install -y realmd
Loaded plugins: amazon-id, rhui-lb, search-disabled-repos
Resolving Dependencies
--> Running transaction check
--> Package realmd.x86_64 0:0.16.1-5.el7 will be installed
--> Processing Dependency: oddjob-mkhomedir for package: realmd-0.16.1-5.el7.x86_64
--> Running transaction check
--> Package oddjob-mkhomedir.x86_64 0:0.31.5-4.el7 will be installed
--> Processing Dependency: oddjob = 0.31.5-4.el7 for package: oddjob-mkhomedir-0.31.5-4.el7.x86_64
```

Figure 8: Installing realmd

31. Let's test what we have so far. We should now be able to ping our domain controller. Enter

```
ping -c4 lastname.loc
```

(the '-c4' switch instructs the system to ping four times, rather than continuously, which is the default. This makes it behave like Windows' ping command)

32. If the ping command doesn't work, let me know

33. The packages that need to be installed now require some updated packages (dependencies). So enter

```
sudo yum -y upgrade
```

This will take a little while...

34. Once the upgrade has completed, test realmd. You should be able to use realmd to 'discover' your domain. Enter

```
realm discover lastname.loc
```

```
[ec2-user@ip-173-1-1-27 ~]$ realm discover ramsey.loc
ramsey.loc
type: kerberos
realm-name: RAMSEY.LOC
domain-name: ramsey.loc
configured: no
server-software: active-directory
client-software: sssd
required-package: oddjob
required-package: oddjob-mkhomedir
required-package: sssd
required-package: adcli
required-package: samba-common
[ec2-user@ip-173-1-1-27 ~]$
```

Figure 9: Output from `realm discover ramsey.loc`

Notice from the output above, we're provided several pieces of information:

- The authentication type is 'kerberos'
- The realm name and domain name are ramsey.loc
- This server isn't currently configured to use the realm
- The domain controller is using Active Directory

- And we're going to have to install several packages to make everything work -- oddjob, oddjob-mkhomedir, sssd, adcli, and samba-common

35. We can install all of these with a single command:

```
sudo yum -y install oddjob oddjob-mkhomedir sssd adcli samba-common
```

Join RHL Instance to the Realm

36. Once all of the packages are installed, it is time to join the RHL server to the *lastname.loc* domain:

```
sudo realm join ramsey.loc
```

You will be prompted to enter the domain Administrator's password (which should be 'Passw0rd!')

```
[ec2-user@ip-173-1-1-27 ~]$ sudo realm join ramsey.loc
Password for Administrator:
[ec2-user@ip-173-1-1-27 ~]$
```

Figure 10: Joining the realm. Pretty anti-climactic, huh?

37. You can confirm that the join was successful by running another realm discover:

```
realm discover ramsey.loc
```

```
[ec2-user@ip-173-1-1-27 ~]$ realm discover ramsey.loc
ramsey.loc
type: kerberos
realm-name: RAMSEY.LOC
domain-name: ramsey.loc
configured: kerberos-member
server-software: active-directory
client-software: sssd
required-package: oddjob
required-package: oddjob-mkhomedir
required-package: sssd
required-package: adcli
required-package: samba-common
login-formats: %U@ramsey.loc
login-policy: allow-realm-logins
[ec2-user@ip-173-1-1-27 ~]$
```

Figure 11: Confirming successful join

Note that the output now says 'configured: kerberos-member'. Also make note of the 'login-formats:' line. This tells you that logins will be permitted using the format 'user@ramsey.loc', e.g., Administrator@ramsey.loc

38. Now we need to permit realm logins from domain users. Enter

```
sudo realm permit --realm lastname.loc --all
```

39. From the Windows domain controller, launch the Server Manager

40. Click on 'Tools' (in the upper right corner) and select 'Active Directory Users and Computers'

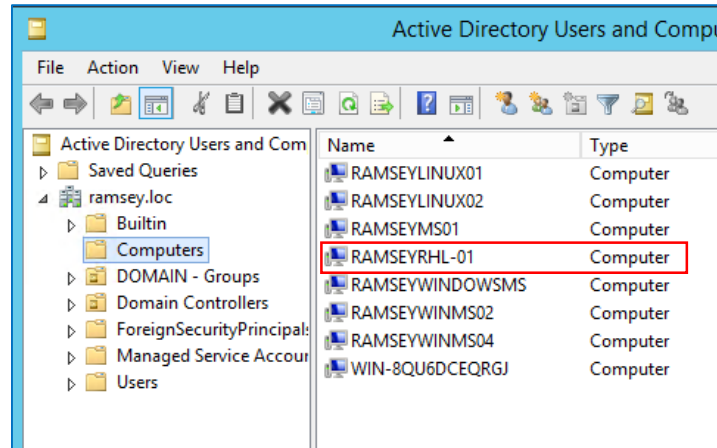


Figure 12: RAMSEYRHL-01 displayed in AD Users and Computers

You should see your RHL instance displayed under Computers

41. Now, we need to modify the RHL's ssh configuration file to allow password authentication, Kerberos authentication, and GSSAPI authentication (The Generic Security Service Application Program Interface (GSSAPI, also GSS-API) is an application programming interface for programs to access security services. The GSSAPI is an IETF standard that addresses the problem of many similar but incompatible security services in use today). Enter:

```
sudo nano /etc/ssh/sshd_config
```

Down-arrow (scroll) to the line that reads, 'PasswordAuthentication no' and change 'no' to 'yes'. Scroll down further to the section that starts with '# Kerberos options.' Make the following changes:

- KerberosAuthentication yes (be sure to uncomment the line also)
- uncomment KerberosOrLocalPasswd yes (allows authentication against either the local /etc/shadow file or via the domain controller's Kerberos server)
- uncomment KerberosTicketCleanup yes
- Leave KerberosGetAFSToken and KerberosUseKuserok commented out

Under the GSSAPI options:

- GSSAPIAuthentication yes
- GSSAPICleanupCredentials yes

```
# Kerberos options
KerberosAuthentication yes
KerberosOrLocalPasswd yes
KerberosTicketCleanup yes
#KerberosGetAFSToken no
#KerberosUseKuserok yes

# GSSAPI options
GSSAPIAuthentication yes
GSSAPICleanupCredentials yes
#GSSAPIStrictAcceptorCheck yes
#GSSAPIKeyExchange no
#GSSAPIEnablek5users no
```

Figure 13: Modifying /etc/ssh/sshd_config

Make careful note in the screen shot above which lines are still commented versus which are not. If the line is commented out, it doesn't matter whether the value is 'no' or 'yes'

42. Save the changes and exit. Having made changes to the configuration file, it is necessary to restart the sshd service:

```
sudo systemctl restart sshd
```

43. Now, just for giggles, let's add the ramsey.loc Domain Admins group to the RHL's sudoers group. Enter:

```
sudo nano /etc/sudoers
```

(Note: **visudo** is the 'preferred' way to modify the **sudoers** file. But for our purposes - unless you just want to wrestle with **vi** - **nano** will do the trick). Scroll to the end of the file. After the last entry, add

```
# Add Domain Admins to sudoers
%Domain\ Admins@lastname.loc ALL=(ALL) ALL
```

```
# Add Domain admins to sudoers
%Domain\ Admins@ramsey.loc ALL=(ALL) ALL
```

Figure 14: Add domain administrators to sudoers

'ALL=(ALL) ALL' instructs the system to grant all privileges to the group in question, in this case, Domain Admins

Test Authentication

44. Now, for the moment of truth. From your Windows domain controller, launch another PuTTY window. Enter the private IP address of your RHL instance (do NOT load your key file)

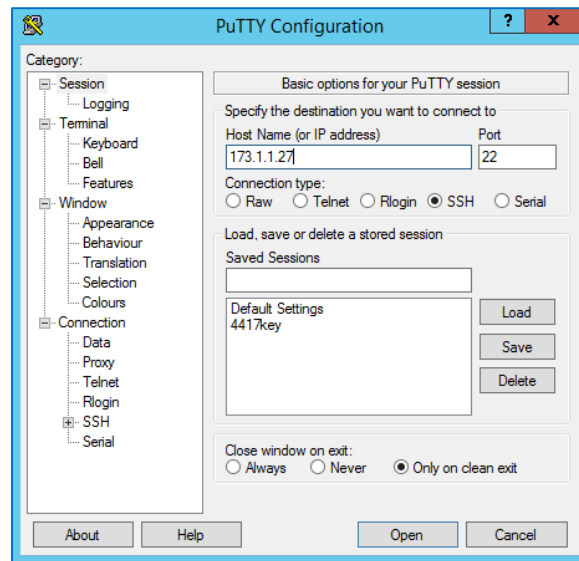


Figure 15: Logging in to the RHL instance with PuTTY

45. Click 'Open'. At the 'login as:' prompt, enter Administrator@lastname.loc
46. Enter your (domain) Administrator password (Passw0rd!)
47. If all goes well, you should be authenticated successfully!
48. Enter `11 /home`. Note that the system created a home directory for 'administrator@ramsey.loc'
49. Test that the Administrator is a member of the sudoers group

sudo su

After entering the Administrator's (domain) password, you should see this:

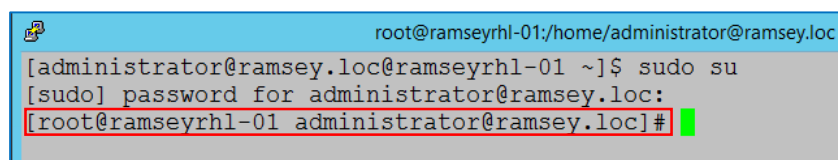
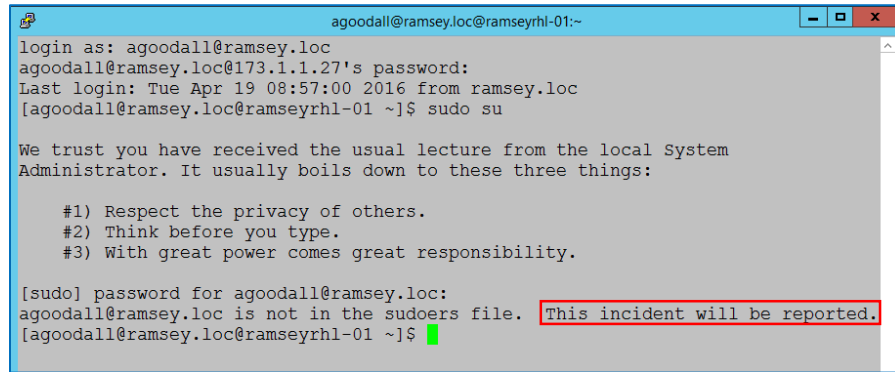


Figure 16: Confirming Administrator is also a sudoer

50. Now, open a new PuTTY window (right-click on the menu bar and select 'Duplicate Session'). Connect to the RHL instance and log in as user 'agoodall' (Remember, Alice is not a member of the Domain Admins group, but she can still log in to the instance because of the realm permit command in Step 38, above)
51. As user 'agoodall,' enter

sudo su

Alice will be prompted for her password, but then receives a rude response:



```

agoodall@ramsey.loc@ramseyrhl-01:~
login as: agoodall@ramsey.loc
agoodall@ramsey.loc@173.1.1.27's password:
Last login: Tue Apr 19 08:57:00 2016 from ramsey.loc
[agoodall@ramsey.loc@ramseyrhl-01 ~]$ sudo su

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for agoodall@ramsey.loc:
agoodall@ramsey.loc is not in the sudoers file. This incident will be reported.
[agoodall@ramsey.loc@ramseyrhl-01 ~]$

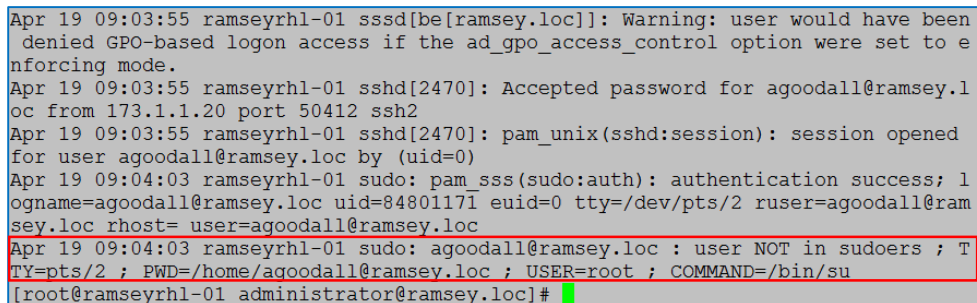
```

Figure 17: Alice is not a Domain Admin, therefore she is not a sudoer either

52. If you're wondering about the 'This incident will be reported.' comment, enter

sudo tail /var/log/secure

from the administrator's terminal window (Alice can't access log files, either)



```

Apr 19 09:03:55 ramseyrhl-01 sssd[be[ramsey.loc]]: Warning: user would have been
denied GPO-based logon access if the ad_gpo_access_control option were set to e
nforcing mode.
Apr 19 09:03:55 ramseyrhl-01 sshd[2470]: Accepted password for agoodall@ramsey.l
oc from 173.1.1.20 port 50412 ssh2
Apr 19 09:03:55 ramseyrhl-01 sshd[2470]: pam_unix(sshd:session): session opened
for user agoodall@ramsey.loc by (uid=0)
Apr 19 09:04:03 ramseyrhl-01 sudo: pam_sss(sudo:auth): authentication success; l
ogname=agoodall@ramsey.loc uid=84801171 euid=0 tty=/dev/pts/2 ruser=agoodall@ram
sey.loc rhost= user=agoodall@ramsey.loc
Apr 19 09:04:03 ramseyrhl-01 sudo: agoodall@ramsey.loc : user NOT in sudoers ; T
TY=pts/2 ; PWD=/home/agoodall@ramsey.loc ; USER=root ; COMMAND=/bin/su
[root@ramseyrhl-01 administrator@ramsey.loc]#

```

Figure 18: 'This incident will be reported'

Note: The file name for authentication information on a RHL system is different than that of an Ubuntu/Debian system. For RHL, it is /var/log/secure; Ubuntu/Debian, /var/log/auth.log (that's the log file I've been showing you this semester to illustrate all the attempts to hack my web servers)

53. Feel free to play around with your new configuration. For example, launch a new PuTTY terminal window and log in as user [bsmith@lastname.loc](#). Note that Bob is a Domain Admin and can therefore switch user (**su**) to root. Have Bob run **sudo yum -i update**, and so forth

When you're done, Stop all of your instances. Since this is our last lab of the semester, you may want to terminate your instances, particularly if AWS is still, in spite of all the emails back and forth, charging you for usage (I'm still working on that, awaiting their response to my latest ticket). But make sure you have all that screen shots you need for your report first! In fact, you may want to delay terminating your instances until you have completed your report, just to be sure.

Observations

Complete your lab report in accordance with the Lab Guide and examples posted on D2L. As part of your observations, reflect on the relevance of being able to configure a heterogeneous environment. What are some other operating systems that may need to be integrated into a production environment (think: BYOD)? Research and present a brief explanation of how Kerberos authentication works. Would this lab have worked if we had parked our RHL instance on the private subnet we set up earlier this semester (without further modifications to the subnet or security groups)? Why or why not? Feel free to experiment for yourself.

Make sure to cite your sources -- use any style you like, e.g., MLA or APA. There should be at least three sources listed.