Unit 3 Lab – Identity and Access Management

**Required Materials**

Putty or other connection tool Lab Server

Root or sudo command access

STIG Viewer 2.18 (download from <https://public.cyber.mil/stigs/downloads/> )

# EXERCISES (Warmup to quickly run through your system and familiarize yourself)

1. ls -l /etc/pam.d/
	1. What are the permissions and names of files? Can everyone read them?
2. cat /etc/pam.d/sshd
	1. What information do you see in this file?
	2. Does any of it look familiar to you?

# PreLAB

Download the STIG Viewer 2.18 from - <https://public.cyber.mil/stigs/downloads/>



Download the STIG for RHEL 9 and the import it into your STIG viewer



Create a checklist from the opened STIG for RHEL 9



# LAB

This lab is designed to have the engineer practice securing a Linux server or service against a set of configuration standards. These standards are sometimes called benchmarks, checklists, or guidelines. The engineer will be using STIG Viewer 2.18 to complete this lab.

# PAM configuration:

1. Connect to a hammer server
2. Filter by pam and see how many STIGS you have. (Why is it really only 16?)
3. Examine STIG V-257986
	1. What is the problem?
	2. What is the fix?
	3. What type of control is being implemented?
	4. Is it set properly on your system?
		1. grep -i pam /etc/ssh/sshd\_config
		2. Can you remediate this finding?
4. Check and remediate STIG V-258055
	1. What is the problem?
	2. What is the fix?
	3. What type of control is being implemented?
		1. Are there any major implications to think about with this change on your system? Why or why not?
	4. Is it set properly on your system?
	5. How would you go about remediating this on your system?
5. Check and remediate STIG V-258098
	1. What is the problem?
	2. What is the fix?
	3. What type of control is being implemented?
	4. Is it set properly on your system?
6. Filter by “password complexity”



* 1. How many are there?
	2. What are the password complexity rules?
		1. Are there any you haven’t seen before?
1. Filter by sssd
	1. How many STIGS do you see?
	2. What do these STIGS appear to be trying to do? What types of controls are they?

# OpenLDAP Setup

You will likely not build an LDAP server in a real world environment. We are doing it for understanding and ability to complete the lab. In a normal corporate environment this is likely Active Directory.

To simplify some of the typing in this lab, there is a file located at

/lab\_work/identity\_and\_access\_management.tar.gz that you can pull down to your system with the correct .ldif files.

1. Install and configure OpenLDAP
	1. Stop the warewulf client systemctl stop wwclient
	2. Edit your /etc/hosts file #use your server line # Entry for hammer1

192.168.200.151 hammer1 hammer1-default ldap.prolug.lan ldap

* 1. Setup dnf repo

dnf config-manager --set-enabled plus

dnf repolist

dnf -y install openldap-servers openldap-clients openldap

* 1. Start slapd

 systemctl start slapd

ss -ntulp

* 1. A2llow ldap through the firewall

firewall-cmd --add-service={ldap,ldaps} --permanent firewall-cmd --reload

firewall-cmd --list-all

* 1. Generate a password #use testpassword [root@hammer1 ~]# slappasswd

New password:

Re-enter new password:

{SSHA}wpRvODvIC/EPYf2GqHUlQMDdsFIW5yig

* 1. Change the password vi changerootpass.ldif

dn: olcDatabase={0}config,cn=config changetype: modify

replace: olcRootPW

olcRootPW: {SSHA}vKobSZO1HDGxp2OElzli/xfAzY4jSDMZ

[root@hammer1 ~]# ldapadd -Y EXTERNAL -H ldapi:/// -f changerootpass.ldif SASL/EXTERNAL authentication started

SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth SASL SSF: 0

modifying entry "olcDatabase={0}config,cn=config"

* 1. Generate basic schemas

ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/openldap/schema/cosine.ldif ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/openldap/schema/nis.ldif

ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/openldap/schema/inetorgperson.ldif

* 1. Set up the domain #USE THE PASSWORD YOU GENERATED EARLIER

vi setdomain.ldif

dn: olcDatabase={1}monitor,cn=config changetype: modify

replace: olcAccess olcAccess: {0}to \* by

dn.base="gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth" read by dn.base="cn=Manager,dc=prolug,dc=lan" read by \* none

dn: olcDatabase={2}mdb,cn=config changetype: modify

replace: olcSuffix

olcSuffix: dc=prolug,dc=lan

dn: olcDatabase={2}mdb,cn=config changetype: modify

replace: olcRootDN

olcRootDN: cn=Manager,dc=prolug,dc=lan

dn: olcDatabase={2}mdb,cn=config changetype: modify

add: olcRootPW

olcRootPW: {SSHA}Uf13AbVHOcs/aDWJOvDxxfBSl3omExG2

dn: olcDatabase={2}mdb,cn=config changetype: modify

add: olcAccess

olcAccess: {0}to attrs=userPassword,shadowLastChange by dn="cn=Manager,dc=prolug,dc=lan" write by anonymous auth by self write by \* none

olcAccess: {1}to dn.base="" by \* read

olcAccess: {2}to \* by dn="cn=Manager,dc=prolug,dc=lan" write by \* read

###Run it

[root@hammer25 ~]# ldapmodify -Y EXTERNAL -H ldapi:/// -f setdomain.ldif SASL/EXTERNAL authentication started

SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth SASL SSF: 0

modifying entry "olcDatabase={1}monitor,cn=config" modifying entry "olcDatabase={2}mdb,cn=config" modifying entry "olcDatabase={2}mdb,cn=config" modifying entry "olcDatabase={2}mdb,cn=config" modifying entry "olcDatabase={2}mdb,cn=config"

* 1. Search and verify the domain is working.

[root@hammer25 ~]# ldapsearch -H ldap:// -x -s base -b "" -LLL "namingContexts"

dn:

namingContexts: dc=prolug,dc=lan

* 1. Add the base group and organization. vi addou.ldif

dn: dc=prolug,dc=lan objectClass: top objectClass: dcObject objectclass: organization o: My prolug Organisation dc: prolug

dn: cn=Manager,dc=prolug,dc=lan objectClass: organizationalRole cn: Manager

description: OpenLDAP Manager

dn: ou=People,dc=prolug,dc=lan objectClass: organizationalUnit ou: People

dn: ou=Group,dc=prolug,dc=lan objectClass: organizationalUnit ou: Group

ldapadd -x -D cn=Manager,dc=prolug,dc=lan -W -f addou.ldif

* 1. Verifying

ldapsearch -H ldap:// -x -s base -b "" -LLL "+" ldapsearch -x -b "dc=prolug,dc=lan" ou

* 1. Add a user

Generate a password

slappasswd #use testuser1234

vi adduser.ldif

dn: uid=testuser,ou=People,dc=prolug,dc=lan objectClass: inetOrgPerson

objectClass: posixAccount

objectClass: shadowAccount cn: testuser

sn: temp

userPassword: {SSHA}dk/Lks9078gfZQJ31ABvPpvKv3sHhr29 loginShell: /bin/bash

uidNumber: 15000

gidNumber: 15000 homeDirectory: /home/testuser shadowLastChange: 0

shadowMax: 0

shadowWarning: 0

dn: cn=testuser,ou=Group,dc=prolug,dc=lan objectClass: posixGroup

cn: testuser gidNumber: 15000 memberUid: testuser

ldapadd -x -D cn=Manager,dc=prolug,dc=lan -W -f adduser.ldif

* 1. Verify that your user is in the system. ldapsearch -x -b "ou=People,dc=prolug,dc=lan"
	2. Secure the system with TLS #accept all defaults

openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout

/etc/pki/tls/ldapserver.key -out /etc/pki/tls/ldapserver.crt chown ldap:ldap /etc/pki/tls/{ldapserver.crt,ldapserver.key}

[root@hammer25 tls]# ls -l /etc/pki/tls/ldap\*

-rw-r--r--. 1 ldap ldap 1224 Apr 12 18:23 /etc/pki/tls/ldapserver.crt

-rw -. 1 ldap ldap 1704 Apr 12 18:22 /etc/pki/tls/ldapserver.key

vi tls.ldif

dn: cn=config changetype: modify

add: olcTLSCACertificateFile olcTLSCACertificateFile: /etc/pki/tls/ldapserver.crt

add: olcTLSCertificateKeyFile olcTLSCertificateKeyFile: /etc/pki/tls/ldapserver.key

add: olcTLSCertificateFile

olcTLSCertificateFile: /etc/pki/tls/ldapserver.crt [root@hammer25 ~]# ldapadd -Y EXTERNAL -H ldapi:/// -f tls.ldif

* 1. Fix the /etc/openldap/ldap.conf to allow for certs vi /etc/openldap/ldap.conf

#

# LDAP Defaults #

# See ldap.conf(5) for details

# This file should be world readable but not world writable.

#BASE dc=example,dc=com

#URI ldap://ldap.example.com ldap://ldap-master.example.com:666

#SIZELIMIT 12

#TIMELIMIT 15

#DEREF never

# When no CA certificates are specified the Shared System Certificates

# are in use. In order to have these available along with the ones specified # by TLS\_CACERTDIR one has to include them explicitly:

TLS\_CACERT /etc/pki/tls/ldapserver.crt TLS\_REQCERT never

# System-wide Crypto Policies provide up to date cipher suite which should # be used unless one needs a finer grinded selection of ciphers. Hence, the # PROFILE=SYSTEM value represents the default behavior which is in place # when no explicit setting is used. (see openssl-ciphers(1) for more info) #TLS\_CIPHER\_SUITE PROFILE=SYSTEM

# Turning this off breaks GSSAPI used with krb5 when rdns = false SASL\_NOCANON on

* 1. systemctl restart slapd

# SSSD Configuration and Realmd join to LDAP

SSSD can connect a server to a trusted LDAP system and authenticate users for access to local resources. You will likely do this during your career and it is a valuable skill to work with.

1. Install sssd, configure, and validate that the user is seen by the system
	1. dnf install openldap-clients sssd sssd-ldap oddjob-mkhomedir authselect
	2. authselect select sssd with-mkhomedir --force
	3. systemctl enable --now oddjobd.service
	4. systemctl status oddjobd.service
	5. Uncomment and fix the lines in /etc/openldap/ldap.conf BASE dc=prolug,dc=lan

URI ldap://ldap.ldap.lan/

* 1. vi /etc/sssd/sssd.conf [domain/default] id\_provider = ldap autofs\_provider = ldap auth\_provider = ldap chpass\_provider = ldap

ldap\_uri = ldap://ldap.prolug.lan/ ldap\_search\_base = dc=prolug,dc=lan #ldap\_id\_use\_start\_tls = True #ldap\_tls\_cacertdir = /etc/openldap/certs cache\_credentials = True #ldap\_tls\_reqcert = allow

[sssd]

services = nss, pam, autofs domains = default

[nss]

homedir\_substring = /home

* 1. chmod 0600 /etc/sssd/sssd.conf
	2. systemctl start sssd
	3. systemctl status sssd
	4. validate that the user can be seen id testuser

uid=15000(testuser) gid=15000 groups=15000