TSP相关知识总结

# TSP主要模块及物理机器关系

|  |  |  |  |
| --- | --- | --- | --- |
| 模块名称 | 部署名称 | 所在机器 | 作用 |
| TSPClient.exe | 回连控制端 | 10.0.16.9  111.204.211.204 | 1. 10.0.16.9机器用来生存木马和用于测试用回连；  2. 111.204.211.204机器用于客户生成木马的回连。 |
| TSPLisence.exe | 许可验证服务  TSPLisence.exe | 10.0.0.58  lazycat8049.gicp.net | 1. TSP登陆验证服务 2. 10.0.0.58 提供server20.dat的下载服务 3. lazycat8049.gicp.net 提供插件的下载服务 |
| TSPServer.dll（sbiedll.dll.bak） |  | 被控机 | 对被控机进行控制和信息获取，与TSPClient.exe通讯 |
| TSPLoader.exe  （server20.dat） | 未写入回连地址的木马加载器 | 10.0.0.58 | 木马释放及植入程序 |

# 机器说明

|  |  |  |
| --- | --- | --- |
| 机器 | 运行的进程或服务 | 功能说明 |
| 10.0.16.9 | TSPClient.exe | 1. 木马生成 2. 测试用回连端 |
| 111.204.211.204 | 1.TSPClient.exe  2.Web服务器Tomcat  3.MySQL数据库 | 1. 客户木马回连端 2. 客户用TSP页面Web服务 3. 客户用TSP后台数据库 |
| lazycat8049.gicp.net | TSPLisence.exe  和http 服务 | 1. TSPClient.exe登陆许可验证服务 2. 提供http的插件下载服务 |
| 10.0.0.58 | TSPLisence.exe | 提供10.0.16.9机器生成木马时server20.dat的下载服务 |
| 被控机 | ctfmon.exe | 加载运行我们的木马dll |

# TSP登录数据流程图



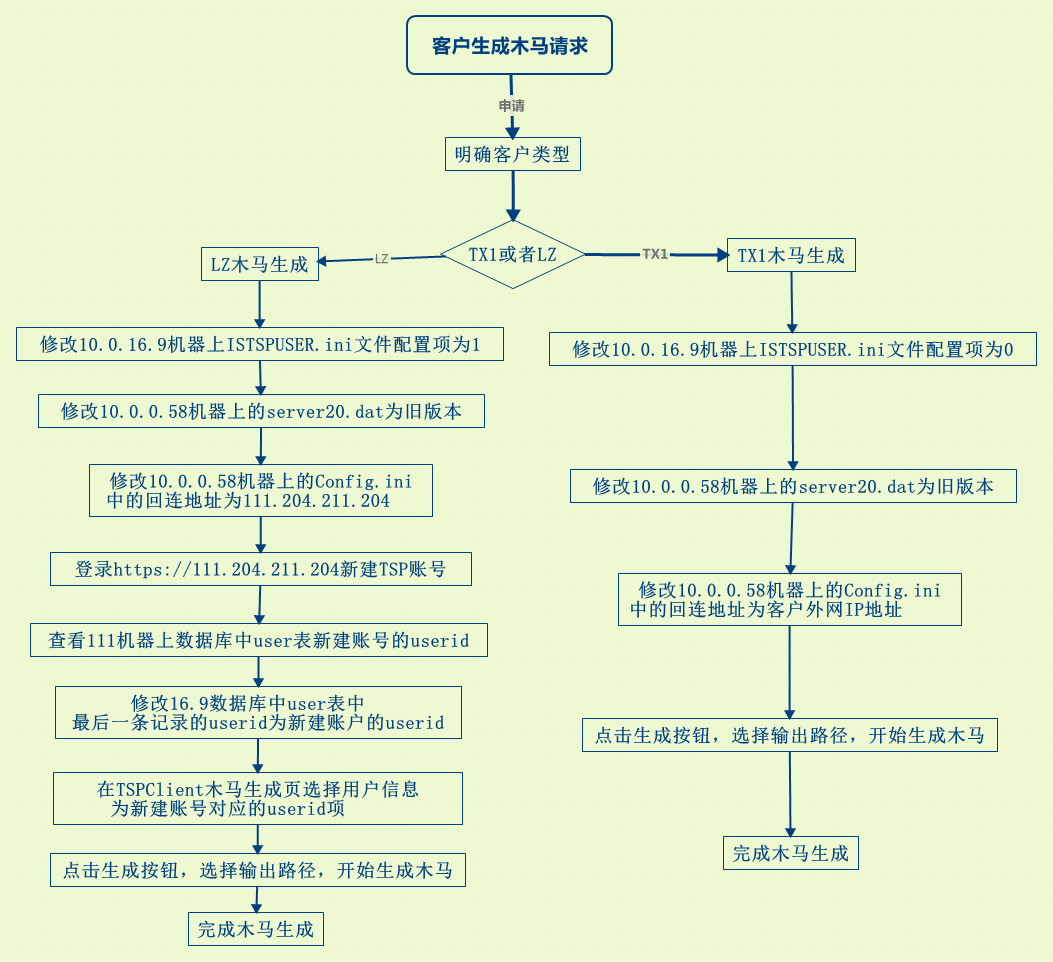
# TSP木马生成数据流程图



# TSP LZ木马运行回连结构图



# TSP木马生成过程



由于在10.0.16.9机器上，C:\Windows\System32\drivers\etc\host文件中，添加了如下地址转换：10.0.0.58 lazycat8049.gicp.net，即在该机器上对lazycat8049.gicp.net的访问将被定向到10.0.0.58机器上。

1. 修改10.0.16.9机器上TSP安装目录D:\run\ISTSPUSER.ini 中的配置项ISTSPUSER的值，生成LZ木马，则修改为1，生成TX1木马，则修改为0，不需要重启客户端，因为每次生成木马的时候，都会直接从配置文件读取该参数。
2. 确定回连地址：

修改10.0.0.58 机器上D:\run\TSPLicense\Config.ini中开头的adminpass区段下的dns中ip地址，我们测试用的时候，写10.0.16.9:8000,给客户生成木马，回连地址写111.204.211.204:8000，如果客户需要木马回连到他们自己安装的TSP客户端，则该地址应该填写客户的ip地址。

1. 确定server20.dat的版本：

在10.0.0.58机器上确保启动TSPLicense进程，且443端口启动正常，回连10.0.16.9,则需要使用新版本的server20.dat,该版本的为600多K，回连111.204.211.204则需要使用老版本的server20.dat,该版本的为400度K，名字必须为server20.dat,由于10.0.16.9和111.204.211.204上安装的TSP客户端分别为新旧版本，如果server20.dat版本不一致，将导致不能回连。

1. 生成TSP账号（LZ木马）

操作流程如下图，详情见文字说明。



1. 当回连10.0.16.9机器时，在浏览器登录<https://10.0.16.9/>,使用admin账号登陆tsp，然后添加新用户，新建账户用于管理控制被控机。
2. 当回连111.204.211.204机器时，在浏览器登录<https://111.204.211.204/>,使用admin账号登陆tsp，然后添加新用户，新建账户用于管理控制被控机。
3. 如果是回连111.204.211.204机器，则需要在该机器上查看数据库user表中新建账号的userid，并在10.0.16.9机器上修改user表中最后一条记录的userid为该id，目的是在生成木马的时候，可以选择该id的用户信息选项。
4. 生成木马

在TSPCLient木马生成tab页，如果是生成LZ木马，则需要选择刚才新建的userid对应的选项，如果是生成TX1木马，则不需要选择。然后点击生成即可生成木马了。如果提示下载失败，请确保host文件中香港服务器到10.0.0.58机器的映射是否存在，确保58机器上TSPLicense运行正常，如果不正常，请确保进程正常后重新生成。

# TSP木马验证

1. 准备工作：

将生成的木马拷贝到一台未植入过木马的干净机器，确保该机器与生成木马时设置的回连机器通讯正常，回连客户端TSPClient.exe运行正常。

1. 植入测试：

在被控机器上以管理员身份执行生成的木马文件，然后到C:\windows\meed木马下，查看释放出的文件，包括ctfmon.exe、sbieDLL.dll、sbiedll.dll.bak,其中sbiedll.dll.bak就是我们的TSPServer.dll，只不过后面加入了回连地址和案件线索等信息。过几分钟后，会在meed木马下生成cfg.ini文件，打开里面是一团乱码，该文件是从回连端请求的关于插件的配置信息加密后的结果，如果该文件下载成功，则说明回连成功。

1. 配置文件cfg.ini 下载失败问题说明：

也可以从TSPClient上查看被控机的上线信息，如果TSPClient上看到了上线信息，但是被控机器上却没有生成cfg.ini，则可能是TSPClient下没有找到PluginUpdate.ini,因为TSPClient在生成加密cfg.ini配置文件的时候，需要从该文件读取配置信息，然后加密处理，然后发送到被控机器。

# server20.dat来龙去脉

1. 来龙：server20.dat是由TSPLoader3编译得到的TSPLoader3.exe文件重命名得到的文件。
2. 去脉：TSPCLient在生成木马的时候从58机器上下载server20.dat，下载的时候由TSPLicense将server20.dat的数据读入内容，并在结尾追加回连地址信息，然后将数据流发送给TSPCLient；TSPCLient根据是ISTSPUSER.ini配置文件决定是否写入线索ID，并由TSPCLient将需要写入的线索ID写入生成的输出文件，该文件生存的时候，先输出到临时目录下的HACKFANSSC.exe，最后将文件移动并重命名为选择的输出文件名，默认为server.exe。

# TSPLoader详解

1. 相关文件的说明

TSPlLoader为木马加载器，植入被控机的文件就是该文件重命名后的文件，该文件在执行的时候，会释放ctfmon.exe、sbieDLL.dll、sbiedll.dll.bak，释放出的这几个都是以十六进制数组的形式，放在TSPlLoader的源码中的，其中ctfmon.exe、sbieDLL.dll这两个文件我们没有源码，所以代码中对两个文件的数据数组不能修改, sbiedll.dll.bak对应TSPServer.dll。

ctfmon.exe 对应ctfmon.h中的数组 unsigned char szctfmon[8290]

sbieDLL.dll对应sbieDLL.h中的数组 unsigned char szbiedll[2371]

sbiedll.dll.bak对应server.h中的数组unsigned char data[456704]

以上三个数组的数据分别是ctfmon.exe、sbieDLL.dll、TSPServer.dll数组经过微软的MakeCAB工具压缩后的数据流，通过WinHex查看复制成C代码的形式粘贴过来的。

MakeCAB 压缩数据时，命令大致如下：

Makecab ctfmon.exe out.data

1. 释放文件

TSPLoader在被控机器上执行的时候，会依次释放每个数组的数据到临时文件，然后将使用微软的Expand工具解压还原得到原始文件。命令格式如下：Expand aa.tmp cfmon.exe。

1. 首次植入时木马的启动执行：

由TSPLoader 执行命令ctfmon.exe –mi启动，ctfmon将会加载执行sbiedll.dll.bak中的代码，sbiedll.dll.bak会在注册表中写入启动项，下次开机时，自动执行ctfmon.exe。

# 插件的执行、更新和卸载

1. 在sbiedll.dll.bak被执行时，会调用rundll32.exe 执行TSPPlug.dll中的代码，TSPPlug.dll部署在香港服务器上，名字为zh.dll，该动态库负责加载所有的插件，所以在被控机器上，只有一个rundll32.exe进程执行我们的插件。
2. TSPPlug.dll（zh.dll）通过共享内存的方式，通过事件Event的方式控制与sbiedll.dll.bak(TSPServer.dll)进行通信，从而完成插件的更新、下载、删除命令。
3. TSPPlug.dll（zh.dll）在调用其他插件时，是通过调用动态加载dll的方式，通过找到每个插件中的入口函数start，然后通过函数指针进行调用，停止某个插件，则是调用插件中的stop函数。

# 本地调试方法

1. 需要安装VS2008，下载源代码。
2. 修改TSPClient中的登录验证服务器地址：

修改TSPClientPublic.h中的的对香港服务器的宏定义，如下：

#define LICSERVER\_IPADDR "lazycat8049.gicp.net"

#define LICSERVER\_LISPORT 443

修改为：

#define LICSERVER\_IPADDR "127.0.0.1"

#define LICSERVER\_LISPORT 444

1. 编译TSPClient.exe、TSPLisence.exe、TSPServer.dll
2. 在本地启动TSPLisence服务，并将server20.dat放到TSPLisence.exe所在目录。
3. 修改Config.ini中adminpass回连地址为本机IP地址。
4. 修改DBConfig.ini中数据库地址为：10.0.16.9，也可以在本机安装MySQL，并导入TSPCenter数据库。
5. 通过以上步骤，就可以在本地生成木马了、并进行植入回连测试了。

# 通信数据的加密和解密

1. 在发送数据时，对数据包进行了加密操作，加密传输的过程如下:
2. 对要发送的原始数据lpData，使用zlib进行zip压缩，得到pDest，长度为destLen
3. 对压缩后的数据pDest逐字节与0x2a进行异或操作
4. 对压缩后数据的长度整数进行变换（加密）,公式如下：

LONG nBufLen = ((destLen + HDR\_SIZE)^0xF4) + 0xFA97;

HDR\_SIZE 宏定义在新旧版本中的TSP中不同，老版本的为30，新版本的为18，该值很关键，在TSPServer和TSPClient中都有定义，且必须保持一致，否则导致某些功能不能正常使用。

1. 然后开始组织要发送的数据包，依次写入如下的数据：
2. 数据包标示：PacketFlag[10],为10个字节的字符数组，内容为ABCDEUVWXY
3. 写入压缩后数据长度变换后的数值nBufLen，长度为4个字节
4. 写入原始未压缩的数据的长度，长度为4个字节
5. 写入压缩后的数据流，长度为压缩后的长度destLen。
6. 使用socket发送第4步中组织的数据包，发送过程中会分成多次发送。
7. 在接收数据时，对数据包进行解密操作，解密操作的过程如下：
8. 从接收数据包的开头读取前10个字节，得到PacketFlag数组
9. 比较PacketFlag,决定以何种方式还原压缩后的数据长度。
10. 从第10个字节后面读取一个整数，得到压缩后数据长度的加密值
11. 根据PacketFlag内容，对上步得到的整数还原，得到压缩后的数据的长度。
12. 读取未压缩的原始数据的长度
13. 读取压缩后的数据流
14. 使用zlib的uncompress方法，对压缩数据解压，得到发送的原始数据流。

# 回连地址的写入及解析

## TSPLoader回连地址写入空间预留

1. 回连地址预留空间如下:

TCHAR szConnectIp[50]="CHAONIMADEXXX"; 该变量是一个全局变量。

TSPCLient生成木马时会查找该字符串，并在该位置处写入回连地址。

1. 案件线索ID预留空间如下：

TCHAR tchStrCaseAndClueID[50] = "123456789\_abcdefghi" ;

该变量是一个全局变量。

TSPCLient生成木马时会查找该字符串，并在该位置处写入案件线索ID。

## TSPLisence对回连地址的处理

1. TSPLisence在下载server20.dat时对回连地址的操作如下：
2. 配置文件中的地址: 10.0.144.223:8000
3. 异或操作后: VWIWIVSSIUUT]\_WWW
4. Encode加密: S0pgSmBLRkZgTExNVFJKSkoZ
5. 拼接在WCCCRX后面: WCCCRXS0pgSmBLRkZgTExNVFJKSkoZ
6. 写入到发送的server20.dat数据结尾

## TSPClient对回连地址的处理

1. TSPClient在收到server20.dat修改后的数据后，执行的操作：
2. memfind查找CHAONIMADEXXX标示位置，将该地址转换为数组stu\_vpn\_param的指针
3. memfind查找WCCCRX标示位置: WCCCRXS0pgSmBLRkZgTExNVFJKSkoZ
4. 根据TSPClient.ini中是否使用VPN的配置项做不同处理，如下:
5. 如果不使用VPN，则直接将WCCCRXS0pgSmBLRkZgTExNVFJKSkoZ写入到stu\_vpn\_param结构体的szConnectIp处
6. 如果使用VPN，则将配置文件的VPN地址做以下操作：
7. m\_VpnTspIP做异或操作，然后再做encode加密,最后写入到stu\_vpn\_param结构体的szConnectIp处。

## TSPLoader释放文件时对回连地址的处理

回连地址结构体定义如下，与TSPClient中的形式一样。

struct VPNINFO

{

TCHAR szConnectIp[64];

TCHAR szVPNServerIp[64];

TCHAR szVPNUser[32] ;

TCHAR szPassd[32] ;

};

typedef struct VPNINFO stu\_vpn\_param;

采用如下的方式，拼接回连地址及案件线索信息:

sprintf(tchIPAndID,"%s{%s{%s{%s}%s",stu->szConnectIp,stu->szVPNServerIp,stu->szVPNUser,stu->szPassd,tchStrCaseAndClueID);

然后将tchIPAndID字符串写入到释放处理的sbiedll.dll.bak文件结尾。

下面是一个回连地址为10.0.144.223时，在释放sbiedll.dll.bak文件时写入其结尾的回连地址案件线索信息：

WCCCRXS0pgSmBLRkZgTExNVFJKSkoZ{MTwoMjIvMDY5Iz83PgQ={{}123456789\_abcdefghi

## TSPServer对回连地址的解析

TSPServer.dll(sbiedll.dll.bak)，执行对回连地址的解析操作：

1. 找到WCCCRX标示位置，对其后面的数做Dencode解密，然后做异或操作，还原得到回连地址:10.0.144.223:8000

# 其他知识

1. TSP控制页面上面被控机不显示插件问题：该问题是数据库表中的plug\_info对应的机器id相关插件参数错误导致的，其中pid为9、10、11的三项值必须为1，如果不为1，则插件列表可能就显示不出来。9 对应通用库，10对应密取文件回传，11对应插件管理，这三个插件是其他插件运行的基础保证，所以一旦这几个值出现为0的1情况，在TSP页面上就不能显示任何插件了。
2. 给TX1客户生成木马是回连地址需要写客户TSP所在机器的外网IP地址。