

APT-C-36（盲眼鹰）利用Hijackloader开展攻击活动

ctfiot.com/287745.html

APT-C-36

盲眼鹰

APT-C-36（盲眼鹰）是一个疑似来自南美洲的APT组织，主要目标位于哥伦比亚境内和南美洲的一些国家和地区，如厄瓜多尔、智利和巴拿马等。该组织自2018年被发现以来，针对以上目标地区的政府部门、金融、保险等行业以及大型公司持续发起定向攻击。

一、概述

近期360高级威胁研究院监测发现，盲眼鹰组织在2025年10月份实施了新一轮攻击活动，在本次攻击活动中，攻击者使用了Hijackloader[1][2]加载恶意载荷，并呈现出以下技术特点：

- 1、灵活使用各类API间接调用方法；
- 2、该loader以模块化形式运作，并使用各类注入手段多阶段、反复进行shellcode注入。通过以上技术的综合运用，达到欺骗受害者、迷惑分析人员和逃避各类自动化检测手段的目的。

二、攻击流程分析

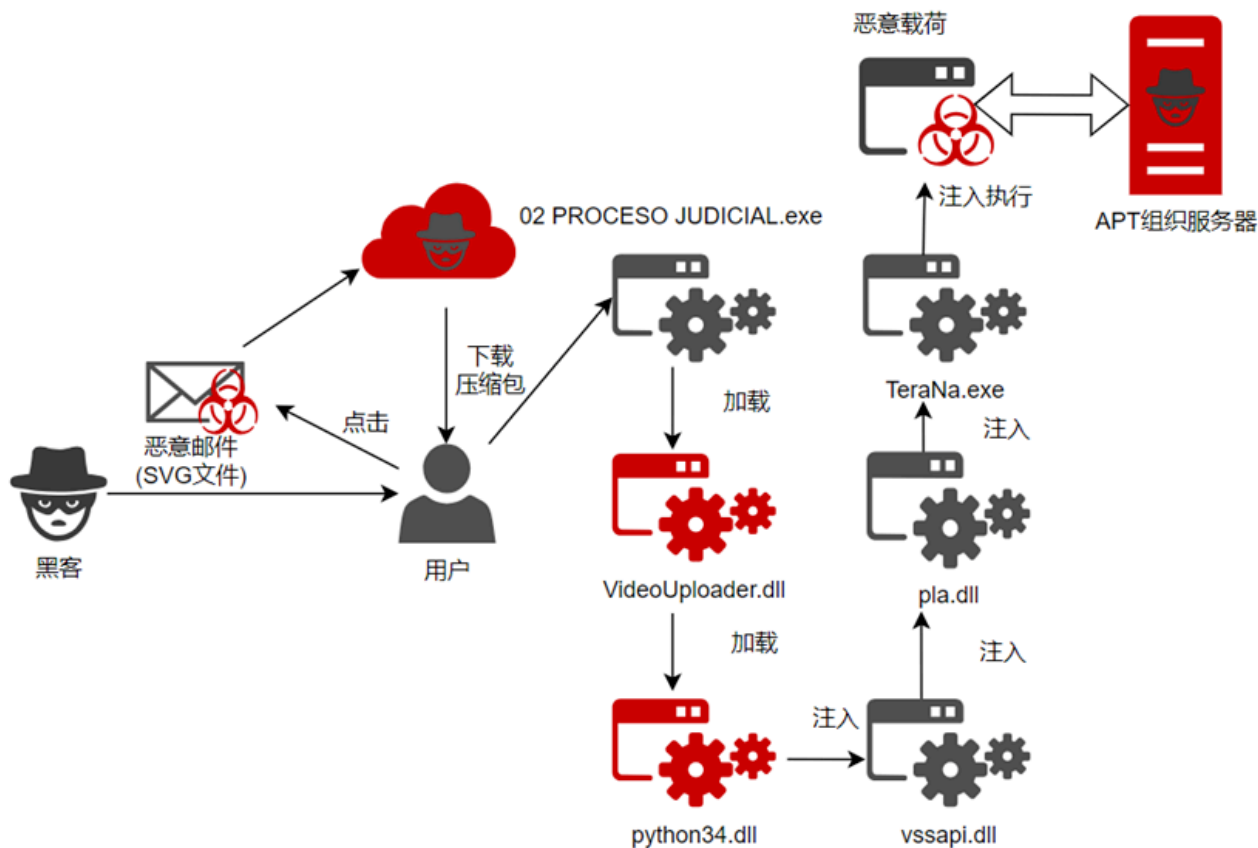


图1 攻击流程示意

攻击者使用“白加黑”的方法依次加载两个恶意dll文件，再将数据文件中的Hijackloader解密和解压缩出来，注入到合法PE文件中并执行，最终加载Pure远程控制木马，实现对受害者计算机的控制。

1. 载荷投递

本次攻击活动中的钓鱼手法与过往相比没有变化，仍然使用SVG文件作为诱饵，向目标发送包含这个附件的钓鱼邮件。

受害者点击链接后下载压缩包Demanda por daños y perjuicios – Juzgado 49，解压后文件夹内包含6个文件，其中的02 PROCESO JUDICIAL.exe和MSVCR100.dll为合法应用程序和动态链接库，python34.dll和VideoUploader.dll为恶意PE文件，通过执行exe来依次加载两个dll；Rookcrung.eh和Vind.wt为加密的恶意载荷文件。

图2 解压缩后的文件夹内的文件

受害者点击exe文件开始执行后，首先加载videouploader.dll，再通过LoadLibrary加载python34.dll，而后进入python34.dll继续执行。

图3 从videouploader.dll中加载python34.dll

在python34.dll中，首先从kernel32.dll中导入多个函数地址，以备后续操作中使用。然后读取文件RookCrung.eh，调用VirtualProtect更改vssapi.dll读/写/执行属性后，挖空并注入从文件中解密出的恶意载荷。

图4 向vssapi.dll中注入解密后的RookCrung.eh

在vssapi.dll中，首先将Vind.wt文件中的内容解密和解压，得到Hijackloader。解密该文件时使用了如下方法：检索加密文件中的“????IDAT”字符模式进行匹配（这也是该loader又被称作IDAT loader的原因），找到该模式后检查紧随其后的四字节内容是否为C6 A5 79 EA，若是则对其后的数据进行异或运算解密。接下来的八字节为密钥和数据长度。

图5 查找解压位置并解密

图6 加密数据的结构

图7 载荷解密过程

解压时间接调用了ntdll.RtlDecompressBuffer，使用LZNT1解压算法实现。

图8 解密和解压后的文件包含了下一步注入的目录、文件名及其可执行文件，以及各功能模块名及其配置

图9 解密和解压缩Vind.wt

图10 通过计算和比对哈希查找所需要的模块，将其注入pla.dll的.text节中开始执行

2. 载荷加载

该loader包含了多达35个功能模块。这些模块包含了配置信息（待注入目录、文件名、杀毒软件名称等）、后续注入的完整的PE文件或shellcode等。

图11 提取出的35个功能模块

该loader依次执行如下模块：

Ti（主模块）-> SM->TinyCallProxy-> TinyCallProxy-> (创建进程C-R.exe)-> modWriteFile-> SM-> AVDATA-> rshell-> ESAL-> CUSTOMINJECT->CUSTOMINJECTPATH-> TinyCallProxy-> modCreateProcess->(创建进程Terana.exe)-> TinyCallProxy-> modCreateProcess->（创建进程chime.exe）

Ti模块中实现了以下功能：

1、通过哈希校验动态导入kernel32、kernelbase和ntdll模块及其大量函数，保存在一个表中，在后续的调用中直接查表调用，避免静态特征被分析检测，Hijackloader中大量API都通过此方法调用。

图12 建立API间接调用表

图13 模块查找：遍历PEB->ldr->InLoadOrderModuleList将模块名的CRC32哈希值与目标模块的CRC32 哈希值比对

图14 收集名称以Zw开头的系统调用

2、该模块中使用了一种“栈欺骗”技术来掩护受监控的敏感系统调用或内核级 API调用来源，使之看起来像是来自合法的系统调用。具体逻辑如下：

- 1) 通过跟踪EBP寄存器来回溯栈帧；
- 2) 获取返回地址（EBP+4），如果返回地址不是位于ntdll或kernelbase的的text节区内，则记录该返回地址；
- 3) 将返回地址替换为一个合法dll（d3d9.dll）的text节区内随机地址；
- 4) 重复以上过程直到栈底限制或遇到连续三个返回地址都在ntdll或kernelbase的栈帧，说明上层调用已经回到系统代码，可以停止查找和替换。

以上逻辑的具体实现如下：

图15 栈欺骗实现过程

3、此外，该模块还使用了Heaven's Gate即所谓“天堂之门”技术调用64位API。

图16 Heaven's Gate实现

该技术原理为将 64位CS寄存器selector值0x33 压栈，获取 EIP 并压栈，调整返回地址指向后面的64位stub，最后使用far return 使CS被切换到0x33，进入64位模式。

图17 通过天堂之门实现一次64位模式系统调用的过程

4、模块中还实现了unhook功能：将当前加载的ntdll与纯净的ntdll的text节进行比较，并对call和jmp指令的目标地址进行检查比对，确定是否被hook，如果发现存在hook行为，则将当前加载的ntdll修改为纯净的ntdll内容。

图18 unhook ntdll的具体步骤

图19 CUSTOMINJECT模块内是一个待注入的合法PE文件，用以将最终载荷注入

图20 CUSTOMINJECTPATH模块内的配置信息则指定了这个PE的路径。

图21 SM模块的内容相当简单，只有一个未来用作宿主程序的d3d9.dll的名字

图22 TynycallProxy模块用以实现对特定系统调用函数的调用的包装，可变长参数中包含了调用号

AVDATA模块主要包含了与杀毒软件进程名哈希值、执行流程标志位等配置数据，构成一个28字节的结构体。样本将AVDATA模块中的杀毒软件进程名的crc32哈希值与调用NtQuerySystemInformation 检索到的本机进程名哈希值进行比对，若发现一致，则改继续检查它的持久化或注入方法标志位，选择相应的执行流程。

检索的杀毒软件进程名包括zhudongfangyu.exe (360)、360tray.exe (360)、avp.exe (卡巴斯基)、avastsvc.exe (avast)、vkise.exe (Comodo)、cis.exe (Comodo)、mbam.exe (Malwarebytes) 等。

图23 AVDATA模块中的数据

图24 检查进程是否为杀毒软件进程

ESAL模块的作用为将已注入的shellcode清除，并执行最终载荷。

COPYLIST模块内包含了用以拷贝进临时目录中的文件名，通过它们实现持久化。

modCreateprocess模块的作用为创建指定的进程，该进程执行的是shellcode或最终载荷注入的目标 (%userprofile\AppDataRoaming\systemdockerChime.exe等)，其中的关键调用通过TynycallProxy模块实现。例如CreateProcessW、NtDelayExecution等。

rshell模块功能为执行最终载荷。最终载荷以4字节异或的方式被解密出来，然后检查该载荷的PE信息，以确定后续注入方式。

图25 解密最终载荷

检查预设注入类型后实施注入操作，此处类型为3。

图26 向TeraNa.exe注入rshell模块

根据标志位的设置，可选择6种不同的注入方式。在本例中标志位按位与4为true，同时按位与0x80为false，故采用了进程分身[3][4]的方式实现了对TeraNa.exe的注入操作，该方法是一种利用事务性NTFS（TxF）写入+节映射来在不落地文件的情况下创建并执行恶意进程的隐蔽技术。主要步骤包括：

- 1) TxF：创建一个文件系统事务（Transaction），并将当前线程绑定到该事务；
- 2) PE 结构修改：在内存中构造目标PE 文件的“dat”节区；
- 3) 使用modCreateProcess创建进程，再通过NtOpenProcess调用，打开目标进程；
- 4) 调用NtGetContextThread、NtSetContextThread获取和更改目标线程的寄存器状态；
- 5) 调用NtDelayExecution暂停一段时间，以逃避基于时间检测的沙箱；
- 6) 将rshell模块写入进程入口；
- 7) 调用NtResumeThread运行该进程。

3.最终载荷分析

最终载荷被注入到一个白文件，该恶意载荷是属于Pure家族的远程控制软件，它由地下网络犯罪开发者团体PureCoder开发，具有收集主机信息、虚拟环境监测、加载各类插件等功能。

该载荷如下图所示，已使用.NET Reactor进行了重度混淆保护，具有防止反编译的功能，不具备可读字符。

图28 在dnSpy内查看提取到的RAT反编译结果

图29 提取到的RAT文件信息

该RAT的配置信息经Protobuf序列化、Gzip压缩、Base64编码三步处理后存储在载荷中。

图30 加密的配置信息

经解码和解压缩，可以看到它的配置信息如下：

图31 解密和解压缩后的配置信息

各字段分别代表C2服务器域名、端口、TLS证书、ID、环境变量路径、互斥量等信息。

三、归属研判

通过对本次攻击活动的分析，我们发现攻击方式与APT-C-36（盲眼鹰）组织的技术特征较为符合，具体总结如下：

- (1) 钓鱼诱饵以司法类文件为主题，面向哥伦比亚境内用户，与过往活动完全一致[5]。

(2) 多阶段加载dll、注入手法、注入目标（pla.dll等系统dll）、恶意载荷数据文件命名（无意义的拉丁字母和扩展名）等活动特点与以往类似。

值得注意的是，在此次活动中，该组织开始大规模使用近年来在网络犯罪界开始流行的Hijackloader，使用它的多个功能模块来实现恶意行为，技术手段趋于复杂化。这说明该组织及时跟进了恶意软件的发展潮流，不断丰富自身的工具储备，以提高攻击成功率，规避分析和检测。

附录 IOC

MD5

5894fc151abb8d58439214304a4354b0

e8b6d147ed7712a7db9ab1ec3cfe734b

4b917db1f7ddc803821b573f1bc2a91c

3f4699c17966857c625426fbbda039b3

C2

loque2025.mysynology[.]net

参考

[1]<https://www.trellix.com/blogs/research/analysis-of-hijackloader-and-its-infection-chain/>

[2]<https://www.zscaler.com/blogs/security-research/technical-analysis-hijackloader#second-stage-loading>

[3]<https://attack.mitre.org/techniques/T1055/013/>

[4]<https://blackhat.com/docs/eu-17/materials/eu-17-Liberman-Lost-In-Transaction-Process-Doppelganging.pdf>

[5]<https://research.checkpoint.com/2025/blind-eagle-and-justice-for-all/>

团队介绍

TEAM INTRODUCTION

360高级威胁研究院

360高级威胁研究院是360数字安全集团的核心能力支持部门，由360资深安全专家组成，专注于高级威胁的发现、防御、处置和研究，曾在全球范围内率先捕获双杀、双星、噩梦公式等多起业界知名的0day在野攻击，独家披露多个国家级APT组织的高级行动，赢得业内外广泛认可，为360保障国家网络安全提供有力支撑。

| 本篇文章来源于微信公众号: 360威胁情报中心