

APT-C-20利用explorer劫持和LSB隐写等技术实施隐蔽攻击活动分析

 mp.weixin.qq.com/s/TDb_UzNfebMzMxh_bQdMvA

高级威胁研究院

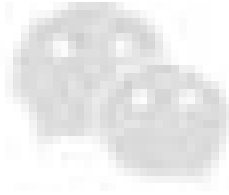
APT-C-20

APT28

APT-C-20 (APT28) 也被称为Fancy Bear，是一支顶尖国家级黑客组织，自2004年起长期针对某国家及全球政府机构实施网络间谍活动；该组织以发起影响国际政治的“入侵与泄露”行动而闻名，技术上擅长利用鱼叉式钓鱼、水坑攻击、零日漏洞等多种手段进行中间人攻击和凭据窃取，是目前全球最具威胁性的网络信息战力量之一。

1. 攻击流程分析

APT-C-20组织用携带VBA宏的恶意文档作为初始入口，通过借助文档Shape对象的坐标操控实施视觉欺骗以规避用户察觉，随后从文档内部剥离并释放恶意组件，然后利用COM劫持实现持久化，并通过隐蔽启动explorer.exe触发侧面加载执行恶意DLL。核心DLL从LSB隐写图片中提取shellcode并执行，再以内存反射加载方式启动后续 C#木马模块，完成最终恶意载荷部署。



此图片来自微信公众平台
未经允许不可引用

2. 载荷投递分析

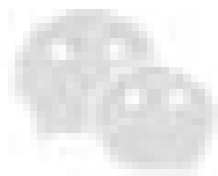
本次攻击行动中，我们捕获到APT-C-20组织多个攻击样本，以其中一个进行分析说明，如下所示：

MD5 77014b3e77529079f041b5b9e73a013b

文件类型 readme.docm

文件大小 469字节

readme.docm是一个携带宏代码的恶意文件，打开文档时显示乱码，提醒用户启用宏以查看完整内容，如下所示。



此图片来自微信公众平台
未经允许不可引用

点击启用宏后显示东欧某国国防部相关诱饵主题内容，如下所示：



此图片来自微信公众平台
未经允许不可引用

为防止宏代码被查看，APT-C-20组织还对宏代码进行了加密。



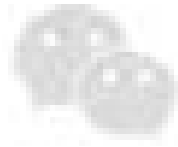
此图片来自微信公众平台
未经允许不可引用

破解宏保护密码后发现其在宏打开阶段主要进行环境检测+文件释放+持久化设置+信息收集+诱饵内容显示，在文档关闭阶段才触发恶意流程和文档还原。



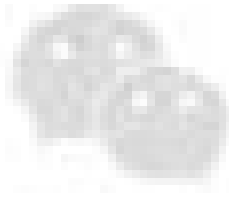
此图片来自微信公众平台
未经允许不可引用

具体来看，在文档打开阶段，首先与dropbox.com进行网络链接进行网络侦察。



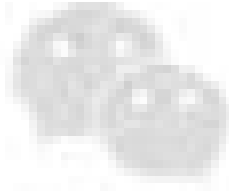
此图片来自微信公众平台
未经允许不可引用

然后将自身改名并复制到temp目录。



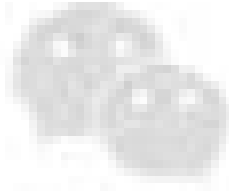
此图片来自微信公众平台
未经允许不可引用

随后从自身中读取数据释放文件%PROGRAMDATA%\Microsoft\DeviceSync\DNX\dnxstore.dll和%PROGRAMDATA%\Microsoft\DeviceSync\EdgeSync\EdgeLogo.png。并设置注册表HKEY_CURRENT_USER\Software\Classes\CLSID\{68DDBB56-9D1D-4FD9-89C5-C0DA2A625392}\InProcServer32指向dnxstore.dll完成com劫持。



此图片来自微信公众平台
未经允许不可引用

接着收集信息包括：判断32/64位环境、Office版本、C:\Program Files目录名（已安装软件列表）并采用单字节0x4A加密写入隐藏文本框。



此图片来自微信公众平台
未经允许不可引用

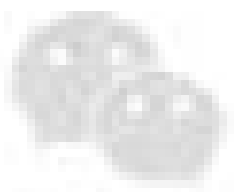
最后显示诱饵文档，切换文档显示状态（显示伪装内容），通过操控文档内特定Shape对象的可见性与坐标属性（如将 Left 属性设为-999996 移出可视区），隐藏打开时的诱饵页面，显示正常内容，降低用户警惕。



此图片来自微信公众平台
未经允许不可引用

在文档关闭阶段完成载荷激活与清理。

不直接调用 regsvr32 等高危程序，而是调用 kernel32.dll 的 CreateProcessW 函数，配置 STARTUPINFO 结构体令 wSW = 0 (隐藏窗口)，隐蔽启动 %WINDIR%\system32\explorer.exe。从而侧面加载资源管理器(explorer.exe)，由于资源管理器在启动和初始化过程中会大量请求系统COM组件。此操作直接触发前序被劫持的 CLSID，使得 dnxstore.dll 顺理成章地被系统合法进程加载至内存中运行。



此图片来自微信公众平台
未经允许不可引用

最后再次通过动态修改Shape的Visible与Left属性切换显示内容，将此前隐藏的对象重新置为可见，并隐藏正常诱饵内容。该机制可确保文档在后续再次打开时仍保持初始伪装状态，从而实现诱饵内容恢复、恶意载荷隐藏以及对静态分析与取证行为的规避。

3.攻击组件分析

攻击组件信息如下：

MD5	b077401fe3d9642345d4c3deafa60aa5
-----	----------------------------------

文件类型	dnxstore.dll
------	--------------

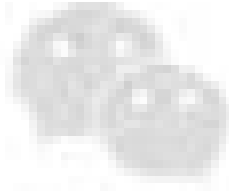
文件大小	227 KB (232,960 字节)
------	---------------------

dnxstore.dll是一个shellcode加载器，首先通过自定义的加解密算法，动态获取导出函数，有效规避静态查杀。通过将dnxstore模块中的DllCanUnloadNow和DllGetClassObject函数转发到系统模块stobject.dll中，避免程序崩溃，增强程序健壮性。

此图片来自微信公众平台
未经允许不可引用

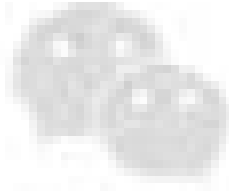
由于{68DDBB56-9D1D-4FD9-89C5-C0DA2A625392}是系统内置的 COM类，为UnexpectedShutdownReason（意外关机原因）对象，经常被explorer进程进行实例化。如果通过COM劫持将正常系统组件劫持到dnxstore.dll模块后，一旦explorer进程实例化该COM类，就会加载dnxstore.dll模块。所以该模块通过当前进程路径是否是explorer.exe来判断程序是否处于调试状态。

如果当前进程路径为regsvr32.exe，这是一个非预期的场景，说明程序可能处于调试环境，则提前退出。如果当前进程是explorer.exe，是一个预期的场景，则执行后续功能。



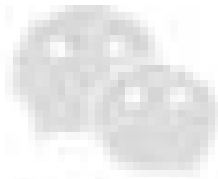
此图片来自微信公众平台
未经允许不可引用

接着，程序通过时间间隔来检测是否处于沙箱环境，分别取两次时间戳，在两次时间戳之间休眠5000ms，如果处于沙箱环境，会劫持Sleep函数，缩短休眠时间，如果两次时间戳的差小于4900ms说明在沙箱环境中。

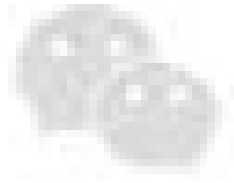


此图片来自微信公众平台
未经允许不可引用

此后获取%programdata%\Microsoft\DeviceSync\EdgeSync\EdgeLogo.png文件路径，该png文件打开正常显示edge图标，以迷惑用户。



此图片来自微信公众平台
未经允许不可引用

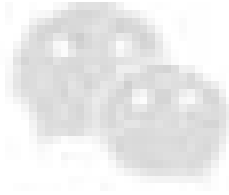


此图片来自微信公众平台
未经允许不可引用

然后基于内置原始密钥种子(“838645f4bab7458fb130c7120a500982”),通过PBKDF2算法派生AES-256密钥,然后加载目标PNG图像并转换为RGBA格式提取像素流。



此图片来自微信公众平台
未经允许不可引用



此图片来自微信公众平台
未经允许不可引用

接着使用LSB模式从像素流中提取Salt和IV参数，之后利用派生的AES密钥解密图像中隐藏的64字节头部数据，解析出核心载荷的提取模式、像素偏移和大小等元信息，最后根据元信息从像素流中提取核心加密载荷并解密，并最终内存执行该shellcode，其功能是内存反射加载自身携带的攻击载荷。



此图片来自微信公众平台
未经允许不可引用

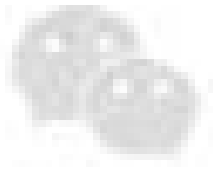
最终载荷信息如下，这是一个C，使用合法的云存储服务(Filen.io)作为C2服务器的远控程序。

MD5 d416eca59188474efa3408827578588b

文件类型 Publish.exe

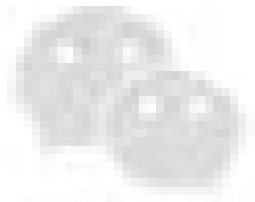
文件大小 68.0 KB (69,633 字节)

Publish.exe经过高度的函数名和变量名混淆，当程序执行后，首先会初始化AES上下文，然后构造上线数据报文，数据报文格式如下：



此图片来自微信公众平台
未经允许不可引用

将"GUID"（基于DomainName和Username生成）,"Type","Meta","IV","EncryptedMessage","HMAC"这些数据进行json格式化后，调用Y9TXNKJ58A.WBX20AQDGG函数进行XOR+BASE64加密。



此图片来自微信公众平台
未经允许不可引用

初始化Filen.io云存储服务API，该程序内置了多个网关节点
(gateway.filen.io,gateway.filen.net)，确保单节点即使出现失效、限流、宕机这些情况，恶意软件仍能通过其他节点保持通信。



此图片来自微信公众平台
未经允许不可引用

将加密之后的上线报文发送给服务端。



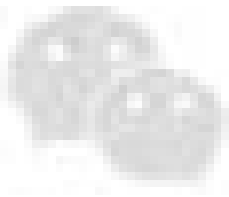
此图片来自微信公众平台
未经允许不可引用

轮询读取服务端返回的消息，



此图片来自微信公众平台
未经允许不可引用

然后经过密钥协商，二次握手确认之后，接收最终需要执行的载荷，并通过内存反射加载后续的攻击载荷。



此图片来自微信公众平台
未经允许不可引用

三、归属研判

通过对本次攻击事件的详细分析，综合初始载荷的执行入口设计、关键战术要素及完整攻击链条特征判断，该样本在整体作战思路与工程实现层面与APT-C-20历史活动高度一致，具体表现如下：

- 1.宏代码在攻击思路与核心技术与以前方式^[1]上高度一致，均通过VBA宏释放恶意DLL，并利用COM劫持实现持久化。两者都会修改注册表项，使Explorer或相关COM组件加载恶意DLL。此外，两者均具备环境检测（win32、win64）、字符串混淆、隐藏文件写入以及Shape对象隐藏数据等特征。
- 2.Dll被加载起来时通过检测宿主禁止是否为explorer.exe，以达到进程伪装目的。同时读取经过隐写术的png图片文件，并从中解密出shellcode执行。整个流程与文章^[2]中提到的PixyNetLoader加载方式一致。
- 3.最终在内存中执行的载荷都属于Covenant Grunt类型，并与File API通信，这类攻击方式在之前报告^[1]中也相同。

4.结合样本上传地址为东欧某国，并且伪装内容是东欧某国相关，符合攻击目标，因此将本次行动归属到APT-C-20(APT28)组织。

总结

本次攻击行动中APT-C-20组织采用了一套高度工业化的攻击框架：该方案以加密宏文档为入口，通过COM劫持实现隐蔽持久化；核心机制利用合法进程explorer.exe触发恶意流程攻击链，并在内存中直接执行基于Filen.io云服务构建的C。这种从载荷释放到执行的攻击方式，极大地规避了传统安防监测，体现了其高精度的隐蔽渗透能力。

此外本文披露的样本只是该组织攻击过程中使用的部分载荷，通过对样本进行的深入分析，希望帮助用户了解此类攻击链条及防范手段。同时也提醒用户提高安全意识，在日常工作中谨慎处理未知压缩包、邮件附件和超链接，避免因轻信而在毫无防护的情况下遭受入侵，造成敏感信息和重要数据的泄露。

附录 IOC

MD5:

77014b3e77529079f041b5b9e73a013b

b077401fe3d9642345d4c3deafa60aa5

1cbffddcb8e1e839737bbf6e50a80aaf

f10d1676b570aa4d97edb844fbb51287

c517d1f4385e0d6e8fa5932d17873eb8

d416eca59188474efa3408827578588b

参考链接

[1]<https://blog.sekoia.io/apt28-operation-phantom-net-voxel/>

[2]<https://www.zscaler.com/blogs/security-research/apt28-leverages-cve-2026-21509-operation-neusplloit>

APT · 目录

上一篇APT-C-36近期针对哥伦比亚境内的活动分析下一篇APT-C-36使用新加载器开展攻击活动