

一款针对国内用户的多阶段Rust加载器分析 | CTF导航



ctfiot.com/287200.html

ResolverRAT: AsyncRAT-Affiliated Malware Employing LOLBins and PowerShell for Stealthy System Compromise ResolverRAT：与 AsyncRAT 相关的恶意软件，利用 LOLBins 和 PowerShell 进行隐蔽系统入侵

CREATED 7 MONTHS AGO

创建于 7 个月前 | UPDATED 6 MONTHS AGO by Peef73

6 个月前由 Peef73 修改 | Public | 观众 | TLP: Green

TLP: 绿色的

This analysis examines ResolverRAT, a malware variant affiliated with AsyncRAT, which utilizes a combination of packed executables and straightforward PowerShell scripts to compromise Windows systems. The malware employs Living-off-the-Land Binaries (LOLBins) to evade detection and establish persistence by interacting with legitimate system processes like explorer.exe. Key functionalities include: Execution of encrypted .dat files and manipulation of system configurations based on language setting information and system details via XML documents (rootedsec.xml, Systems.XML) and POST request. Use of UPnP protocols to gather network statistics and potential ARP poisoning activities.

本分析针对 ResolverRAT，一种与 AsyncRAT 关联的恶意软件变种。该恶意软件利用打包的可执行文件和简单的 PowerShell 脚本来入侵 Windows 系统。它使用诸如 skui.exe 和 sppextcomobj.exe 之类的“本地二进制文件”（LOLBins）来逃避检测，并通过与 explorer.exe 等进程交互来建立持久性。其主要功能包括：执行加密的 .dat 文件并根据语言设置操纵系统配置；通过 XML 文档（rootedsec.xml、Systems.XML）和 POST 请求提取广域网信息和系统详细信息；使用 UPnP 协议收集网络统计信息并进行潜在的 ARP 欺骗活动。

REFERENCE: 参考: <https://medium.com/@shubhandrew/analysis-of-resolverrat-exe-and-powershell-payload-e395d035d57a>

TAGS: 标签: resolverrat, command, techniques, wan information, async rat, part, windows, registry, apis, crypto, powershell, asynrat, trojan, explorer, malware, malicious, execution, manipulation, minimal, example, date, analysts, async

resolverrat、命令、技术、WAN 信息、异步 rat、部分、Windows、注册表、API、加密、PowerShell、异步 rat、木马、资源管理器、恶意软件、恶意、执行、操纵、最小、示例、日期、分析师、异步

ADVERSARY: 对手: Analysts 分析师

MALWARE FAMILIES: 恶意软件家族: Windows, ASYNC

Windows, 异步

文章首发地址：<https://xz.aliyun.com/news/90825>文章首发作者：筱一

概述

近日，笔者关注到一款携带国内数字签名的样本，在对其进行初步分析的过程中，笔者发现了一个比较有意思的事情：

随着样本的曝光，此样本的数字签名校验结果由Signed file,valid signature（文件已签名，签名有效）变为了A certificate was explicitly revoked by its issuer.（证书已被其颁发机构明确撤销）

因此，笔者琢磨，此行为到底是CA机构主动撤销该证书，还是所有者主动撤销该证书？

进一步分析，笔者发现，此样本运行后，会释放多阶段木马。运行过程中，会创建以Qihoo字符串开头的计划任务，同时还会检测360安全软件的360tray.exe、zhudongfangyu.exe进程。最后，样本将在内存中释放并加载携带.NET Reactor壳的AsyncRAT远控木马。

hgreiwwm7.exe

文件名称：hgreiwwm7.exe文件大小：13094664 字节文件版本：52.17.70.40MD5：

537aa066a497005e6a0b116a58b91097SHA-1：efddcaf39f7d594fec890fa16ac1bdad861a20edSHA-256：

5834d20cfc80f6fe20514b7a1e3e9764765aa3f548e2874504e9d26345f80560

国内数字签名

通过分析，发现此样本携带了国内数字签名：Wenzhou Feixun Internet Technology Co., Ltd.

在关注时，笔者发现了一个比较有意思的事情：

- 笔者刚开始关注此样本时，发现此样本的数字签名在VT上提示：Signed file,valid signature（文件已签名，签名有效）
- 过了一天后，笔者尝试对其进行分析，又发现此样本的数字签名在VT上提示：A certificate was explicitly revoked by its issuer.（证书已被其颁发机构明确撤销。）

数字签名截图如下：

初次关注样本时，VT上提示：

第二次关注样本时，VT上提示：

释放并运行文件

通过分析，发现此样本运行后：

- 将在C:\Users\admin\AppData\Roaming\Microsoft\SystemCertificates目录释放239c4b63_19bc_445e_8761_bf3626e0bc.dll文件
- 使用regsvr32.exe程序调用执行239c4b63_19bc_445e_8761_bf3626e0bc.dll文件，执行参数为：“regsvr32.exe” /s /u 239c4b63_19bc_445e_8761_bf3626e0bc.dll”

相关截图如下：

进一步分析，解析regsvr32.exe程序的运行参数：

- /u：卸载，调用DllUnregisterServer函数
- /s：静默模式

239c4b63_19bc_445e_8761_bf3626e0bc.dll

文件名称：239c4b63_19bc_445e_8761_bf3626e0bc.dll 文件大小：4657008 字节 文件版本：87.64.26.21
MD5：6547fea527d8ac3635ebfbd1d03683c7 SHA-1：4269a9343808e903b93409e0b3d2f24653c4a30c
SHA-256：81e73894d42a66f9d280d4b972d01a3c1ac183d30fd7b045f399abbc924ad916

解密函数

通过分析，发现此样本是一款由rust语言编写的程序。

进一步分析，发现此样本中的字符串均被加密，字符串解密函数代码截图如下：

IDAPYTHON解密脚本

为了能够快速分析样本，笔者经过多轮分析，最终确定此样本的字符串解密算法信息如下：

- 解密算法：chacha20poly1305
- 内置key为：9F80A116F20535761EC9BB91CC406987B981A2AC7E3A709BAE83C8A512F8FAE4
- 加密数据和nonce值均由解密函数传递

相关代码截图如下：

基于此，笔者尝试基于AI + 手动代码调整，快速构建了一个IDAPYTHON脚本，可实现对样本加密字符串的自动化解密，解密效果如下：

此样本中所有解密字符串如下：

```
ntdllRtlCaptureContextntdllNtContinueAdvapi32SystemFunction032kernel32.dllVirtualProtectkernel32.dllWaitForSingleObjectkernel32.dllSetEvent0JXNy.rdata.lib.zip"_BgTaskRegistrationMaintenanceTask{}regsvr32SystemCertificatesPT0Sregsvr32.exePT1MSeDebugPrivilegeUSERNAMEUSERPROFILE.dat360tray.exezhudongfangyu.exe"/systemcertificates//0123456789ABCDEFxxxxxxxx-xxxx-4xxx-yxxx-xxxxxxxxxxxxregsvr32.exe/u/safeinit..dll/safeinit/uIoCompletionntdll.dllZwSetIoCompletiontaskhostw.exeSecurityHealthSystray.exesihost.exesihost.exeexplorer.exeexplorer.exe/LOADSAVEDWINDOWStntdll.dllLdrLoadDllamsi.dll
```

IDAPYTHON脚本内容如下：

```
# -*- coding: utf-8 -*-
importidautilsimportidaapiimportidcimportstructimportbinasciifromcryptography.hazmat.primitives.ciphers.aeadimportChaCha20Poly1305# 内置密钥 (32字节) KEY =
bytes.fromhex("9F80A116F20535761EC9BB91CC406987B981A2AC7E3A709BAE83C8A512F8FAE4")defchacha20_decrypt(ciphertext, key, nonce, initial_counter=0): iflen(key) !=32: raiseValueError("Key must be 32 bytes")
iflen(nonce) !=12: raiseValueError("Nonce must be 12 bytes") result = bytearray() pos =0 counter = initial_counter whilepos < len(ciphertext): block = chacha20_block(key, counter, nonce)
counter +=1 remaining = len(ciphertext) - pos take = min(64, remaining) foriinrange(take):
result.append(ciphertext[pos + i] ^ block[i]) pos += take
returnbytes(result)defget_prev_instructions(ea, count=4): """从给定地址 ea 向上获取 count 条有效指令"""
insts = [] cur_ea = ea for_inrange(count): cur_ea = idc.prev_head(cur_ea) ifcur_ea ==
idaapi.BADADDR: break insts.append(cur_ea) returnlist(reversed(insts)) # 保持从上到下的顺序
defextract_immediate_or_offset(ea): """尝试从 mov 指令中提取立即数或 offset 值""" insn = idaapi.insn_t()
ifnotidaapi.decode_insn(insn, ea): returnNone # 检查是否是 mov 指令 ifinsn.itype != idaapi.NN_mov:
returnNone # 右操作数 (src) op = insn.ops[1] ifop.type == idaapi.o_imm: returnop.value elifop.type == idaapi.o_mem: returnop.addr elifop.type == idaapi.o_farorop.type == idaapi.o_near:
returnop.addr else: # 尝试解析 offset unk_XXXX 形式 (可能被识别为 o_imm + name) ifop.type ==
idaapi.o_immandidc.get_name(op.value): returnop.value returnNonedefread_bytes_from_addr(addr,
size): """从指定地址读取指定长度的字节""" data = b"" foriinrange(size): byte_val =
idc.get_wide_byte(addr + i) ifbyte_val == idaapi.BADADDR: print(f"[!] Failed to read byte
at{addr + i:#x}") returnNone data += bytes([byte_val]) returndatadefprocess_xref(xref_ea):
print(f"n[+] Processing call at{xref_ea:#x}") # 获取前4条指令地址 (按执行顺序) prev_insts =
get_prev_instructions(xref_ea,4) iflen(prev_insts) !=4: print(f"[-] Not enough instructions before
call") return # 顺序应为: # [eax+10h] = nonce_len -> inst0 # [eax+0Ch] = nonce_ptr -> inst1 #
[eax+8] = data_len -> inst2 # [eax+4] = data_ptr -> inst3 inst0, inst1, inst2, inst3 =
prev_insts nonce_len = extract_immediate_or_offset(inst0) nonce_ptr =
extract_immediate_or_offset(inst1) data_len = extract_immediate_or_offset(inst2) data_ptr =
extract_immediate_or_offset(inst3) ifNonein(nonce_len, nonce_ptr, data_len, data_ptr): print("[-]
Failed to parse one of the operands") return # 读取数据 ciphertext = read_bytes_from_addr(data_ptr,
data_len) nonce_data = read_bytes_from_addr(nonce_ptr, nonce_len)
ifciphertextisNoneornonce_dataisNone: print("[-] Failed to read ciphertext or nonce") return # 解
密 (注意: ChaCha20 要求 nonce 为 12 字节) ifnonce_len !=12: print(f"[!] Warning: nonce length
is{nonce_len}, but ChaCha20 expects 12. Trying anyway...") chacha = ChaCha20Poly1305(KEY) try:
plaintext = chacha.decrypt(nonce_data, ciphertext, associated_data=None) print(f"[+] Decrypted:
{plaintext.decode('utf-8')}") print(f"[+] Decrypted(hex):{plaintext.hex()}") exceptExceptionase:
print("[-] Decryption failed:", e)defmain(): target_addr =0x72BDB9D0 print(f"[*] Searching xrefs
to{target_addr:#x}...") found =False forxrefinidautils.XrefsTo(target_addr): # 只处理代码引用 (通常是
call 或 jmp) ifidc.is_code(idc.get_full_flags(xref.frm)): found =True
process_xref(xref.frm) ifnotfound: print("[-] No code xrefs found to the target
address.")if__name__ == "__main__": main()
```

创建0f12fc8e.dat

通过分析，发现此样本运行后，将在C:\Users\admin目录创建0f12fc8e.dat文件，相关截图如下：

创建计划任务

通过分析，发现此样本还将创建计划任务，计划任务名为BgTaskRegistrationMaintenanceTask，计划任务启动命令为：

```
regsvr32.exe /s /u  
"%APPDATA%\Microsoft\SystemCertificates\239c4b63" "19bc" "445e" "8761" "bfbe3626e0bc.dll"
```

相关截图如下：

内存加载.NET程序

通过分析，发现此样本运行后，将在内存中释放并加载.NET程序：

- regsvr32.exe是C++程序；
- 239c4b63_19bc_445e_8761_bfbe3626e0bc.dll也是C++代码程序；
- 但239c4b63_19bc_445e_8761_bfbe3626e0bc.dll运行一段时间后，内存空间就变成了.NET程序的内存空间；

相关截图如下：

注入shellcode

基于静态分析，发现此样本存在注入shellcode的功能：

- 调用xor_decrypt函数在内存中解密一段 shellcode；
- 尝试将shellcode注入至taskhostw.exe、SecurityHealthSystray.exe、sihost.exe进程中；
- 若注入进程为 sihost.exe，则将重启资源管理器；

相关代码截图如下：

进一步分析，尝试提取shellcode，发现shellcode中携带了一个PE文件（schedJump.exe），相关截图如下：

hook_ldr_load_dll

基于静态分析，发现此样本存在HOOK内核LdrLoadDll函数的功能。相关代码截图如下：

检测360进程

基于静态分析，发现此样本在创建计划任务时，将检测系统中是否存在正在运行的360安全软件的相关进程：360tray.exe、zhudongfangyu.exe。

相关代码截图如下：

schedJump.exe

文件名称：schedJump.exe文件大小：12800 字节文件版本：0.0.0.0MD5：
E2CC06DCF0D2E13EF5CC2C87D44CF68ESHA1：
2CC941214B4EEA1E57C7C96013601A6FB2FD26C2SHA-256：
3bfb2437d9104775d3c1e47c8021edff51c143c6f4a749b1de8aab2df0c26ff2

解密0f12fc8e.dat

通过分析，发现此样本运行后，将从0f12fc8e.dat文件中解密载荷，并从中提取239c4b63_19bc_445e_8761_bfbc3626e0bc.dll文件路径。相关代码截图如下：

创建计划任务

通过分析，发现此样本将创建计划任务，计划任务名为QihooGetWordSearchFatch，计划任务启动命令为：

```
C:\ProgramData\sihost.exe /s /u
```

```
C:\Users\admin\AppData\Roaming\Microsoft\SystemCertificates\239c4b63""19bc""445e""8761""bfbc3626e0bc.dll
```

备注：「此计划任务名与239c4b63_19bc_445e_8761_bfbc3626e0bc.dll创建的计划任务名不同。」

相关截图如下：

相关代码截图如下：

终止进程

通过分析，发现此样本将终止“wiawow64.exe”，“winrshost.exe”，“wksprt.exe”，“wuapihost.exe”，“raserver.exe”进程。相关代码截图如下：

Kxhxxsutofr.exe

文件名称：Kxhxxsutofr.exe文件大小：573952 字节文件版本：1.0.0.0MD5：

0f57c1f94fad0d039f15750b6b9fa2ceSHA-1：bb4965e5ec5441acaed4cfcb18cc8298db9c18a8SHA-256：
c8dccb63ec5e0ed5343d46bbebcc6f9b85599bb6a99dd962e12f2ea2678a136c

内存解密DLL

通过分析，发现此样本运行后，将在内存中解密并释放DLL文件。相关截图如下：

加载运行DLL

进一步分析，发现样本将调用Assembly.Load函数加载DLL文件，然后调用DLL文件的ClexJqLcrN函数代码。相关代码截图如下：

Apoom.dll

文件名称：Apoom.dll文件大小：1291264 字节文件版本：1.0.9432.23367MD5：

adaec6a75224b3224fa2b00adfc5ae45SHA-1：2147bec03040942dbd3e4daf17b5510b9e3c5bc6SHA-256：
ff66bd39822a1b58e4d9d555f0f90efbff722270999ac1d76327fb9373826d6f

.NET Reactor加壳

通过分析，发现此样本存在.NET Reactor壳，相关截图如下：

配置信息

进一步分析，在内存中可提取此样本的配置信息，配置信息如下：

- 外联地址：118.107.40.167:56001
- Certificate：

```
MIIE5jCCAs6gAwIBAgIQAJpwd6FPpg/xo168eQyHozANBgkqhkiG9w0BAQ0FADAUMRIWEAYDVQQDDA1Mb3pyZnlrcmQwIBcN
MjUwMjE5MDA1NTE4WhgPOTk50TEyMzEyMzU5NTlaMBQxEjAQBGNVBAMMCUxvenJmeWtyZDCCA1IwDQYJKoZIhvcNAQEBBQAD
ggIPADCCAggCggIBAI6vY/px0u2TZ2F6DAyNe3gIACi1UUp4oxGHj+ivr6Be36itKh1yor/oIHbkAre7pc/E4PgOQMR90de6
jgHeVmfa9QdWMEW978JF+GFP8xWGo+9j2Zq1g0sMW6Q2S/6GDJAGTvg2S+2kznUTtP8Tnq97PoL2EiUr3r4IYEDYNRZk7d1F
L6+Wa0yEV1rLrYJI6xS/ARXIXgB/ukoXGXAqZwPa4QA9Q0WRB1da134xAGs8nR0RWYxwBuC6bL3B6/wGppo9bV9L9iIJhGB
IrsVK3l+abZhwpzcGneyG9AoBJ6WbIsapTBxZqF71GvGOWF8e5j9PNv3SBMzq09LvTcnt84SlZ1/n9DA/34oFrretJAN0qpv
ZJZUSpA6fo5xtRHmV4aihoioF70yXVMzpW0FVTneK3GkGkJ+6QYz6tnLf0JoWe3TYWRvom2aEpDYek8lhV+GQ1f288z6sjZ9
ZDS5d4/+FV883AIU6MS2PavVuvOpaaT/lZkGQ2P0ybJUBvtMAVOC7Xrsie4QyeLjZnpGplUdeJU6WuvuR8gzVY+dHnIJ7Hm2
bH6UhleKWWhY93kTS6NctT1JQ/q2EO4Wjnhih68YmkNjRATwvZd0qbVE4saGl02JI+lwbt5Ds3KqyyaLkeZD9X014VrpevsJ
Lbu457X9az2iLRjCN4/xDMiR3u6rAgMBAAGjMjAwMB0GA1UdDgQWBBSYbdVK2VAREbn1BK91rU0DVVAIfjAPBgNVHRMBAf8E
BTADAQH/MA0GCSqGSIb3DQEBAQUAA4ICAQCIQrLzx7D2gq7pbFkYwtrYqUuux12JwQwZ4EtWGU98sbhIyg4DpJ56jArUNRjT
m2hBo1mUrTw0ELm0dICQuzK3MNaEQ0qtfaTfHiC+vZN+3LlhV/H0SNSQneJlLEDNC2y1WZvbFgkTHmbpHweGlnwqYHIRwYUP
CGaR/95aoCIwc2V/viRJ1QAScc5SuamFFyjfRoe3piAE49Z7njqhYQTNhtUUuXALYC4i6F1NWuuK8FTQb6bwZoRA2Y6yFexu
RB05P1M7f43fv0VY+61WqSs+5CPMKIRb8w1S9eCp1Y9ZsIWjXysLnJyL+7zaApkVypHnIsjFaTjZTaufPSVud7djVoTZeXyN
OuIW1Xgi04Yu+QiILXN3vBafd3wJI23zczXMmoUBS/t4FZGVMC0tJkL+Tzr150qB913DMKe1ch61XjGJB5EiR0xv0ddTdpii
dJc92JU/yJmARyRhoeTbd+UyzFqguJ7jI+jGbm68iilGN7Ejinc/rAX03gsBT2uI3NkmMoSfnsKcf4Ecgc0iejJREBVT0h04
SDPDUGBkHC3uCGsiXKakcS9l1Ef0Krv8LrNng5XXhzxpEniRNzaFdXsgZY/+Q8udYRZoYHHKKgn8gp59Bpf55LLWU9ZasJha
RCY7bI8Mg9u6Rinpdofg0KgfA02VaJLkn8IQXcQwmXT9Q==
```

相关截图如下：

Certificate解码信息如下：

背景溯源

为了确定样本家族背景，笔者尝试基于网络信息进行关联，梳理如下：

- 基于Apoom.dll样本的Certificate解码后的CN值Lozrfykrd，确定此样本家族为ResolverRAT；
- 基于微步沙箱结果，确定此样本家族为AsyncRAT；
- 基于网络中分析报告，确定ResolverRAT样本是与AsyncRAT样本有所关联的；

相关截图如下：

IOCs

#hgreiwwm7.exe5834d20cfc80f6fe20514b7a1e3e9764765aa3f548e2874504e9d26345f80560Wenzhou Feixun Internet Technology Co., Ltd. #239c4b63_19bc_445e_8761_bfbc3626e0bc.dll181e73894d42a66f9d280d4b972d01a3c1ac183d30fd7b045f399abbc924ad916#schedJump.exe3bfb2437d9104775d3c1e47c8021eddf51c143c6f4a749b1de8aab2df0c26ff2#Kxhxsutofr.exec8dccb63ec5e0ed5343d46bbebcc6f9b85599bb6a99dd962e12f2ea2678a136c#Apoom.dll1ff66bd39822a1b58e4d9d555f0f90efbfff722270999ac1d76327fb9373826d6f#外联地址118.107.40.167:56001#Novemberbill.exe01f844ca8f535c4fe372d4e6f256ad60d08e795939ce212d519e239e490c74c1Wenzhou Feixun Internet Technology Co., Ltd.

| 本篇文章来源于微信公众号: T0daySeeker