Threat Spotlight: Breaking Down FF-Rat Malware

threatvector.cylance.com/en_us/home/breaking-down-ff-rat-malware.html

The BlackBerry Cylance Threat Research Team

RESEARCH & INTELLIGENCE / 06.13.17 / The BlackBerry Cylance Threat Research Team

Introduction

FF-RAT is a family of malware used in a number of targeted attacks over at least the last five years. It is by no means a new threat, but it is still actively used and developed and worthy of a breakdown in an effort to defend against it.

FF-Rat malware has managed to stay under the radar and does not yet have robust, widespread industry coverage. In this post, we're going to look at a recent sample the Threat Guidance team came across.

The Dropper

The sample we'll be analyzing is the main dropper component:

SHA256: *7a4528821e4b26524ce9c33f04506616f57dfc6ef3ee8921da7b0c39ff254e4e*

The first thing the dropper does is identify the architecture of the targeted host. If the host is a 64-bit system, then a 64-bit version of the dropper will be written to disk and executed, as shown below:

Path: *\%WinDir%\Temp\S[8 Byte Hex String].dat*
SHA256: 8ef257058cbb22fbab54837dc0af1bdd93c2a6bae18ca4a26e0a436656e591e1

Otherwise, if the host is a 32-bit system, then the 32-bit dropper will continue to its next phase of execution.

Both droppers (32- and 64-bit) will proceed to decrypt and decompress an embedded DLL named SetupDll.dll. This DLL contains the primary functionality of the dropper, and is executed entirely in memory - never touching the disk. Next, we’ll cover the process of extracting the DLL.

Decoding and Decompression

The recent dropper and different components of FF-RAT make heavy use of a combination of RC4, single byte XOR and LZ compression to protect the payloads and configuration. We have also observed older variants using aPACK instead of LZ compression.

The basic workflow for the decryption and decompression looks like this:

1) Generate the decryption key. The decryption key is generated by taking a hard coded DWORD and formatting it as an 8-byte hex string through a call to snprintf(). For example, given the value 0x12345678, the generated key would be “12345678” *(Figure 1):*

Figure 1: The Decryption Key is Formatted Through a Call to snprintf()

2) Decrypt the payload using RC4 and the key generated from the call to snprintf()

3) Decompress the decrypted payload through a call to RtlDecompressBuffer(). Rather than importing this function, it's called through a wrapper, which obtains the address by calling Loadlibrary followed by GetProcAddress *(Figure 2).*

Figure 2: RTLDecompressBuffer Function

For the payload (SetupDll.dll) mentioned above, the value 0x3D65308E is turned into the hex string “3D65308E” through a call to snprintf(). The payload located at 0x407074 is then decrypted using the key “3D65308E”. The last step of unpacking the payload is accomplished by a call to RTLDecompressBuffer to complete the decompression.

Understanding this process allows a researcher to place a breakpoint on the call to RTLDecompressBuffer to easily extract Setup.dll after it’s been decrypted and decompressed *(Figure 3):*

Figure 3: X64dbg is Used to Dump the Payload By Setting a Breakpoint On the Call to RTLDecompressbuffer

SetupDll.dll

SetupDll.dll contains the core functionality of the dropper and is executed from within the dropper by calling the exported function SetupWork.

SetupWork makes a series of checks for the following:

AV-Related Processes

360rp.exe	avgidsagent.exe	KSafeSvc.exe	RsTray.exe
360rps.exe	avgnsx.exe	KSafeSvc.exe	seccenter.aye
360rps.exe	avgrsx.exe	KSafeTray.exe	vsserv.aye
360Safe.exe	avgrsx.exe	kxescore.exe	ZhuDongFangYu.exe
360sd.exe	avgui.exe	kxetray.exe	
360Tray.exe	avgwdsvc.exe	odscanui.aye	
AvastSvc.exe	avp.exe	QQPCRTP.exe	
AvastUI.exe	BaiduSd.exe	QQPCTray.exe	
AvastUI.exe	BaiduSdSvc.exe	RavMonD.exe	
avgcsrvx.exe	bdagent.aye	RsMgrSvc.exe	

File Objects:

\\.\\fstab

Named Events:

Global\1224DC7B-DEE0-4903-ABCA-917778626DD
Global\2BC2426F-C8E6-47a7-9C8A-356A219F83AD
Global\4822063B-F74F-4eb5-B257-1E7BAD1BD8CE

If the check succeeds, it proceeds to write the backdoor to disk:

Path: *%WinDir%\system32\RCoResX64.dat*
SHA256: *b01e5b5ea94a39eb3a80339987c68ae4cb8b90e68f9c794d01d6c3ac1fb8759f*

Followed by the configuration data:

Path: *C:\Windows\Media\WindowsMainSound.wav*
Registry: *HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\{6CD70ECA-9CA1-4862-B00C-BACA47548B1B}\ConfigInfo*

Figure 4. Encrypted Configuration Stored in the Registry

Both the backdoor (RCoResX64.dat) and the configuration (WindowsMainSound.wav) are time stomped using the MAC time of the csrss.exe executable found in the System32 directory.

Persistence is achieved by creating a service using one of the names below:

Irmon
Nwsapagent
NWCWorkstation
Iprip

Service Details:

Our analysis has revealed that, although it wasn't enabled in the configuration for this sample, SetupDll.dll has the ability to infect the MBR with a bootkit. This allows the malware to gain execution early in the boot process, maintain persistence, and can make remediation more difficult. We will be covering this component in more detail in a follow up to this blog.

Figure 5: Infected MBR Extracted From SetupDll.dll

Now that we've discussed the dropper and payload extraction, we can move on to the backdoor.

RCoResX64.dat

Much like the dropper, the core functionality is contained within a compressed and encrypted DLL that is only resident in memory. Just like before, the DLL is extracted through the use of RC4 and RTLDecompressBuffer(). This time, the payload is located at offset 0x10004094 and the value 0x429C20CA is converted to the hex string "429C20CA" and used as the decryption key.

Next, the encrypted configuration is loaded from the registry:

Registry: *HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\{6CD70ECA-9CA1-4862-B00C-BACA47548B1B}\ConfigInfo*

If the backdoor is unsuccessful in loading the configuration from the registry, it is loaded from disk:

Path: *C:\Windows\Media\WindowsMainSound.wav*

Figure 6: Decrypted Configuration Dumped From Memory

At this point, FF-Rat will attempt to connect out to the command and control servers listed in the configuration:

rp.gamepoer7.com:80
dns1-1.verifysign.org:53
login.gamepoer7.com:443

The communication with the command and control server is done over HTTP:

Figure 7. HTTP Post Over Port 53

Something that may be of interest to network defenders is the format of the string used for the URI: *"/%s.php?hdr_ctx=%d_%d"* and the hard coded User-Agent: *"Mozilla/5.0".*

The data exchanged with the command and control server is single byte XOR encoded with the key 0x57. The requests download additional code from the command and control server, which is then stored in the registry and executed in memory.

If the target is behind an authenticated proxy that would block the connection to the command and control server, the backdoor has the ability to authenticate to the proxy as the currently logged in user. These requests use the hard-coded User-Agent: *"Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)"*.

Debug Strings

The dropper and DLL have many occurrences of OutPutDebugStringA() throughout their code. This may be an indication that FF-RAT is still under development. A tool such as DebugView can be used to capture this debug information as the malware executes.

Figure 8: Debug Messages From FF-RAT Execution Captured With Sysinternals DebugView

Conclusion

FF-RAT is an effective, proxy-aware RAT that has been in use for at least the last five years. It has been observed being used in targeted attacks against a number of different industries, including government, aerospace, gaming, IT, and telecommunications. Infecting a system with FF-RAT gives attackers unfettered access to that system and can have a significant impact on an organization.

The malware author goes through a lot of trouble to obfuscate key components and make sure they never touch disk. In our analysis, we covered the Dropper, DLL component and obfuscation techniques. We also provided a number of indicators that can help identify an infection.

If you use our endpoint protection product CylancePROTECT®, you were already protected from this attack. If you don't have CylancePROTECT, contact us to learn how our AI-driven solution can predict and prevent unknown and emerging threats.

Indicators of Compromise

SHA-256 Hashes:

FF-RAT 64-bit Droppers:
6E262EDE79284EB4111ABAE6A6DCFE713DB94184F87C6904EC6729E477FB11BA
8EF257058CBB22FBAB54837DC0AF1BDD93C2A6BAE18CA4A26E0A436656E591E1
9CDAAD7554B1B39FDAF0E5F0AD41E7006D36E0F9791DC9C1CF3D50B73F6CA907
9DE5EE57D9CA1800A442D3F53E43B22807B411FF1839C1A242E21254C3B40A49
AECAAD397351C6466E0B5D16CAEB318BF3AFD2946BC8C5FA21BDFCE02924C74E
C9FEEDC43D4D2DE56A819D7056A24B71C74368B055DDEDAA10A4AAC22B9C1CCE

FF-RAT 32-bit Droppers:
7A4528821E4B26524CE9C33F04506616F57DFC6EF3EE8921DA7B0C39FF254E4E
039E9036DEA6A7609BE87EB83CF0738137A8ED3CFB46A611A9CB4B06BEC14775
0E0B579501ABC8F7D2E41B14C76188267F1CECBBCBC2C78B845C5AA6D328731B
306A5298793EEF46C53FC1CC27AA5851120E186E9891445C309FC8410E1A1B24
3E1E11C9551B9C26FD9E7E379206A506172FCCC73DADF60F930F3CA1D1BA1077
53147EB4709DB10E835A9CEA62DC52276EBA14D54F7C26709C4948734ACA19FD
58116F5C0DACFD7D70A9E57E6328E7105667BB14032DEE6F905C271560767BEB
5918335629A3AFBA3D8A384B59D574327F0F583998AC2ECE4AB84A98B65D6233
610D80BF2F1F335A539684C329F87721EF5B7362A22E263709BBE3F18494095A
62ACDC9DBB35C16C770F97C1CD3D65BC1848E60FAD8E9828758C12FDC0BC8A64
691BC271B1724C5DC8C6DDE185B49A465E73EC18380EF900732EA93637ADD24B

7ED4FCB7733620B7D3FE0BCE2351907723FDEF373F053A865D12AEBA3FBE0722
842E7D030221E10804AF926B783FA5C75EEE009AC74CC22C6D1E6507C53AD453
87D5F1E504D02D31741A4D175699FD82F88AB7441D9908DD4F2EEBD28B1B36EB
8C44625E027DB0A1D8CFAD60DA9102E092F7EC69C638DC0BF5FF97665E449FD1
903AA33253FD8CEECB6FE8D7A9076A650F318433939480D8BD44F2BA240977F1
97DDBF427BF887237B1A9C7C0DD85C8F64390F4EBE2CA0D1FC0A292FB4FCC71A
AE6C390FF56A6E83442E0758E7FB15E6A64B96BC022DE6E56D2CFD44E7094667
E1F564C466E60DDBA8FA437241EE109A2FB012C929A56D7FEEF65B67AF4B407E
EAD6378FCF5FD35A15D9DFA0089834EDF636DE9EED73E66FF37CA8F42F1C5F2C
F194B96317B38512F71BC3CBD070FCD19DBA49BE92EACF430376C54BFD8FE15C
FEE2749D2F88CADB77FAEDE6DA6FABCF23D01E6C39AE1B74BD29AC02CCEAD1CC
FF68BBC1F0EB49B75B940E873BF9F4710B9F566B34FA0543238F9D2A739FD27C
FF96D09E3FE618A296DC5B4425224831DBB49877BE054276DA5BAEFCC52E0F53

FF-RAT 64-bit DLLs:

0CB3B3F5408FE40C7F3DD323272BE662335C4B979FBB766BE4AA6FC2C84CC6F2
0E804464F1669674B83E6605D8C4617D8D2B6EFB36532C71B654B61E5C71B8F9
21961087CC10A4666A263BA3841BA571837181B0288DE533FE9F114E8269E7B9
24D7C59076DF6B6E710E80E708513F0D95A23869B5EA43772B5AF9DB92786B51
5B3A0FCCFD1F652BBF71B9F7757A38E5DB0D0ED5A377A821E5E5BF886461E924
632519CA40720D180205BB8405A1BC3888F69899F59DEC53A2EAF06F08A3D86E
908CFF61E49A89443C11F56BB822FB0139967031052E1F456AA3BA80F2E9612C
98CED0CBE7FDB09810D9B2DED5D0B73EC9659AFE179C1D911EDAB373AE630ECE
9E8578E0EA406F987F0E227810408BEC29864A237C0A745D374971618B35AFFE
D4E80E1208BA43272F368D0ECA38F0467D70745A42ABA4D4AC7E333A64201790
EA0062BA2D26D6C3948E93A01C12ED413327E1E428F25495844B14DFF3DE7C9C
FEC88F4BAAD17942EDF29C1F0A6036D1F30BD7435380247BDCD55F2B7E163A1A

FF-RAT 32-bit DLLs:

0358C0461792A8F15811C57C9FB870CCE00DCF8C5BE8BF590BDC2DDE2DDCB4A2
06FB73DEB589E0DA55786AC83410AF3444355A653FEC34D0BF0B17203446B1D5
112531BF280B8354B3A41F1F0EDC2AFA5FE51F65429B813EC536D744B4B67AE5
16312D26C39965CE0CBC8567F11ADD5D5FBDCC11A8A4364FEA9B4F7E3416B0E4
39F488E65D8BDBE04A87A19452F8291A9870DE54C2850FFE8F4140E7C0F00475
41249D078F11EE3D5E07809A50689F29B784B1484681D519AD703AF7B7F25584
4CA190D05C0F4A729A3E370453E2A00FC9CA7282539FAEB794AF358DB5F62046
618B6782809B9ABA05FB8F99568BF6F89CC9EF8F9A5F8A86F1CB76670E215405
8A0D4B1421B91471C3DC65187D77707AB20FD19185DA57FD4CF568ED4BAB6951
8A37114B3290A1A34101AC4877BEDEC6E57EB0C4642CD1CE4CDFE71BDE23B426
8BF4086470F233FE040A017AC5DF4913A2BF38B8C55916E20A2379DC60163003
919407D7394D59E1E45F936A4D9EC76F8B75560E53BA25BF4ACFFE8FB401B7F6
99B43B190B62C5D997288FBFF7C7AE2B224BD2007A40F44558460B280D5C74F7
9C1F358F4500D605B25A6DF2A20AB7EF05FFBC0474C626F54DBF0F0073FE539C
A84929A9BE9AE8C65D8B09C38BA3F73A63CA4F6BE1A7E7AD84F4407E847D842B

B73F67A1DD39F943BF447D5399DD6577A05DB3C1F0BF91E01FAEE4BF38975AEE
BE1A753A8DAA380797743F67BDD3DFB8FE348401A68AAFFF9B97695C8929F140
CEB3AFB539AB43E04EA27E9B378505483E6B03A8DF5D7C9786E1EFB948201C80
CF0E852A828E8BDBB9C77A7DF32E31DDDD1F6B3B7890C2BD80C3C02B5587B42B
D524BEDFB8514DC76B1AA778D865CAEBC76E27BE3773ED3D7DF8DE9C44A1E22B
DF32A0D6156A94C2EEEDF8F6072BAF75F92CCCCFF4A6D1519B07B906EAA3C9B2
E3D867439D08DB7E622A99DC55BB33018B40D18C7BA6D322F4C0E010B62D4706
F6739B7A2E48DCD505E017F53F3AE85B535F4839B7363929097EAF0937799843
B01E5B5EA94A39EB3A80339987C68AE4CB8B90E68F9C794D01D6C3AC1FB8759F

File-Based:
%WINDIR%\System32\curl.dat
%WINDIR%\System32\frtest.dat
%WINDIR%\System32\frmonk.dat
%WINDIR%\System32\trtest.dat
%WINDIR%\System32\prfi0814.dat
%WINDIR%\System32\RCoResX64.dat
%WINDIR%\Media\Windows Config.wav
%WINDIR%\Media\WindowsMainSound.wav

IP Addresses:
103.27.108.121
211.55.29.55
59.188.16.147
68.68.43.149

Domains:
aunetdns(dot)com
capstone.homeftp(dot)net
cxman.wicp(dot)net
dns.gogogogoogle(dot)com
dns-1.verifysign(dot)org
dns1-1.verifysign(dot)org
fan001.yahoolive(dot)us
ftpseck.ftp21(dot)net
game.googlecustomservice(dot)com
game.googlesoftservice(dot)net
gifa.cechire(dot)com
hehe000002.3322(dot)org
hookyouxx.blog.163(dot)com
huangxiaoxian.3utilities(dot)com
info.playdr2(dot)com
latecoere.blogdns(dot)com

linuxdns.sytes(dot)net
login.gamepoer7(dot)com
luotuozhizhu.blog.163(dot)com
pcal2.dwy(dot)cc
pcal2.yahoolive(dot)us
pf.playdr2(dot)com
pplove.bounceme(dot)net
qemail.gotdns(dot)com
rp.gamepoer7(dot)com
svhost(dot)org
tk.u2xu2(dot)com
update.gogogogoogle(dot)com
welcome.dnsd(dot)info
welcometohome.strangled(dot)net
wucy08.eicp(dot)net
wuzhiting.3322(dot)org
www.rooter(dot)tk
www.tibetonline(dot)info
www.vxea(dot)com
zz.alltosec(dot)com

Certificates:
Issued to: Beijing Wintone Science & Technology Corporation Ltd.
Thumbprint: 5b90748fdac1631de2c5286544919983d0716156
Serial Number: 7a c3 9e 72 df d5 85 d9 01 9f 91 80 02 68 f3 ef

Issued to: Binzhou XinPin Technology Co.
Thumbprint: c1fdd5f5cb4b69be78c7c8c946890d5726fc4d42
Serial Number: 39 1e 36 3e c8 2a d7 61 3d b4 78 c1 78 18 0e 8b

Issued to: Yijiajian (Amoy) Jiankan Tech Co.
Thumbprint: 4bb350bea954ac8ceae09ff6859989c1029b9cdb
Serial Number: 65 4b 40 6d e3 88 ec 2a ec 25 3f f2 ba 4c 4b bd
Issued to: SHENZHEN HONGQI ELECTRONICS CO.
Thumbprint: e8d1a4f5d4a903241f6de259ff00e7305423bdf7
Serial Number: 5b 84 5f 7e 7c 6e 68 04 6b f4 34 89 5d e4 86 51

Issued to: Xuzhou Chenji Technology Co.
Thumbprint: c184466ce5685d208cd1d38880a1fb8763322447
Serial Number: 19 ce 16 72 10 71 45 e0 6f dc 45 fa 2b 75 3f 0b

Issued to: Hangzhou Degou Information Technology Co.
Thumbprint: 96d5e6925aa3c9928f393d23e91abd0efadd16ac
Serial Number: 64 47 7c 85 f2 6c 2c a6 7d 76 46 84 34 26 3e 0e

Issued to: Shenzhen Jinxian Technology Co.
Thumbprint: 6e165fe061bc4eef38bfbda414da61a0c42491b2
Serial Number: 54 87 1d 9a b3 8d 19 02 c3 21 15 f4 c0 79 7e a2

Issued to: Zhengzhoushi Tiekelian Information Technology Co.
Thumbprint: 572973c5c86134276404fe524ab35cf3829ebacc
Serial Number: 55 ab 71 a3 f9 dd e3 ef 20 c7 88 dd 1d 5f f6 c3

Issued to: Zhengzhou Wanxiang Gaoke Information Technology Co.
Thumbprint: 9a2ea03faf2219962345fc4ffbc348507b725c81
Serial Number: 2e 7c 84 da 11 c9 80 26 63 45 24 b1 e1 8b 0f bd

** Please note that some of the domains listed have been sink holed by security researchers*

The BlackBerry Cylance Threat Research Team

About The BlackBerry Cylance Threat Research Team

The BlackBerry Cylance Threat Research team examines malware and suspected malware to better identify its abilities, function and attack vectors. Threat Research is on the frontline of information security and often deeply examines malicious software, which puts us in a unique position to discuss never-seen-before threats.

Back