Shell Crew Variants Continue to Fly Under Big AV’s Radar

threatvector.cylance.com/en_us/home/shell-crew-variants-continue-to-fly-under-big-avs-radar.html

The BlackBerry Cylance Threat Research Team

RESEARCH & INTELLIGENCE / 02.09.17 / The BlackBerry Cylance Threat Research Team

Background

Cylance SPEAR™ has identified a newer family of samples deployed by Shell Crew that has flown under AV’s radar for more than a year and a half. Simple programmatic techniques continue to be effective in evading signature-based detection.

Shell Crew, first named by RSA in this paper, has been incredibly proficient over time and breached numerous high-value targets. The backdoor provided an alternative foothold in several observed instances for the group and employed a few tricks like using the Intel SSE extended instruction set to avoid emulation and obscure analysis.

Most of the variants Cylance identified were 64-bit; however, a couple of earlier 32-bit variants were created in May 2015.

Malware Family

Cylance dubbed this family of malware **StreamEx**, based upon a common exported function used across all samples ‘stream’, combined with the dropper functionality to append ‘ex’ to the DLL file name.

The StreamEx family has the ability to access and modify the user’s file system, modify the registry, create system services, enumerate process and system information, enumerate network resources and drive types, scan for security tools such as firewall products and antivirus products, change browser security settings, and remotely execute commands. The malware documented in this post was predominantly 64-bit, however, there are 32-bit versions of the malware in the wild.

A few of the samples were picked up by AV heuristics within the last few months, but newer samples are still coming back with zero detection rates.

Persistence and Initial Execution Setup

The droppers for the backdoor use a semi-random name chosen from the existing service entries under the ‘netsvcs’ registry key on the machine. Once a suitable service name is identified, the dropper appends ‘ex.dll’ to the file path associated with the service DLL. The registry key, which contains available services that belong to the netsvcs group, is defined at:

‘HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Svchost\netsvcs’

Pseudo code to create the service:

Figure 1: Pseudo Code Used to Create the Service

This initial DLL with ‘ex’ appended to the end of the file name is then saved into the system directory. The malware is copied from the resource section of the dropper and set in the registry to auto-start as the newly created service. The malware will temporarily be saved into the ‘temporary path’ location on the computer (found using the ‘GetTempPathA’ API call) and then moved into the final location under the system directory.

“The GetTempPath function checks for the existence of environment variables in the following order and uses the first path found:

1. *The path specified by the TMP environment variable.*
2. *The path specified by the TEMP environment variable.*
3. *The path specified by the USERPROFILE environment variable.*
4. *The Windows directory.*

Ref: https://msdn.microsoft.com/en-us/library/windows/desktop/aa364992%28v=vs.85%29.aspx

The dropper will find and locate the backdoor as either ‘IDR_PEACH_DLL’ or ‘PEACH_DLL’ within the resource section of the dropper.

Dropper – Find resource pseudo code:

save_rsc_to_tmp_path.png

Figure 2: Save RSC to Temp Path

The malware relies upon execution as a ServiceDLL to persist on a victim system and thus will utilize the ServiceMain export by default. During execution of ServiceMain, a new DLL is copied into the system directory with a randomized name, starting with the ascii characters ‘bt’ followed by 6 numeric digits and the extension ‘.dll’ and may display a falsified 'File Modified' date.

Next, rundll32 will be used to call the exported function ‘stream’ from the newly copied ‘bt’ DLL. The name of the associated service will be added after ‘stream’ in the command line argument that calls the DLL. This control flow starts the primary operation of the DLL.

String Obfuscation Techniques

Some commands in the code are obfuscated by a simple technique that utilizes statically programmed fragments of strings when starting the ‘bt’ DLL. The code appends the strings in the proper order and then utilizes them in accordance with the part of execution that is being set up. This technique is fairly common and unsophisticated, but it may possibly help prevent rudimentary analysis by making it harder to read the strings seen in the binary.

An example of this is shown where the code is setting up the command line syntax to start rundll32 with the ‘bt’ file name utilizing the stream export:

Figure 3: Code Snippet Showing String Obfuscation

Ultimately, the code results prints this string (or something similar to it) for use by the malware:
C:\Windows\system32\rundll32.exe “C:\Windows\system32\bt123456.dll”,stream ServiceName.

Malware Configuration and Operation

The malware used a simple one-byte xor against the byte 0x91 to encode its configuration data. Once the configuration information is decoded in the normal execution flow, the malware will attempt to contact the command and control (C&C) server(s) using an HTTP GET request. The following python snippet can be used to find and decode the configuration block from StreamEx samples:

```
def ex_decode(buf):
        offset = buf.find("&^%$#")
        configblock = buf[offset+5:offset+5+0x3D8]
        out = ''
        for byte in configblock:
                out += chr(ord(byte)^0x91)
        return out
```

Figure 4: Python Code Snippet to Find and Decode StreamEx Configuration Block

Interestingly, some of the samples appeared to utilize a log file to record the malware’s network operations. After the connection is made with the C&C server, the malware can send and receive data and accept input from the attackers, allowing them to take full advantage of the backdoor’s functionality. The log file that the malware writes to disk is located here: “%TEMP%\TT_2015.log”. The data in the log is displayed in the following format (this is where the misspelled string ‘start send requset’ is seen on disk):

[processID threadID] [year-month-day hour:minute:second] start send requset
tag:request

The log data can be seen in the screenshot of the log file below:

logfile_example.png

Figure 5: Log Data

Pseudo code for the log file data:

tt_2015log.png

Figure 6: Pseudo Code for the Log File Data

Another simple spelling mistake was also present across all of the identified droppers: ‘error. OpenSCManager faild’. Once the malware successfully makes a connection to one of the statically programmed domains, the attackers had the ability to instruct the malware to conduct various system operations to further their control over the victim’s environment.

Distribution and Associated Malware

Cylance identified several legitimate compromised Korean websites that were used to distribute StreamEx samples over the course of 2016. One of the most recent samples SPEAR found was served from the website ‘www(dot)aceactor.co(dot)kr’ and contained a configuration block dated October 16, 2016. A number of unique PlugX samples as well as another custom RAT were also served from the same website; they commonly used simple easy-to-remember names such as ‘a.exe’ or ‘32.exe’.

At the end of 2016, the group also took care to use private registration when reregistering domains that were originally purchased from a bulk reseller.

Mitigation

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.

cylance_score.png

Figure 7: CylancePROTECT Console, Showing the Detection of Shell Crew Samples

File Hashes (SHA256):

StreamEx 64-Bit Backdoors:
04f69ebca26ee0ab2fc896f803102fdbb0700726074048755c55c891a9243423
37a2ede8de56fe85b4baf4220046dd2923d66ea7d906a5c009751f9f630aec0b
434df165b56c70ff5479ebd3f8d65c1585076c16a19e20bdee750c9f0119e836
50712f13f0ed2cabc264ec62581857468b2670e3a4226d76369c9367648b9ff0
5747de930d6f2dd456765aada5f31b4c2149388625399ae8d0c025cc8509880b
82a7f8c488cf287908f8f80b458bf19410f16ee0df0d8f2eb9f923efc3e0a2fa
a20d81fcbdcfe6183eaaba489219c44942da3e5fc86ce383568b63b22e6981dc
d26f914eb9f58f9efeba3ae5362cf605a371f881183da201a8528f9c9b65b5ad
e5590c6eca821160d02c75025bf9ee30de418269471ae21bff422933fbb46720

StreamEx 32-bit Backdoors:
369dc64903c52f052ebe547511977f5d677614855da31c416fe13d8eb8ed1015
8269c8183fb5e50acf08dea65d8a3d99f406f7febd61dc361622f21b58570396
bfe4da21398a2ac19b04174a7754acc1c2d1725dac7e0651544ff46df9f9005d
fd0c9c28781de60ed70f32b9e138ab7d95201a5f08a4bc0230b24493597022d7

StreamEx Droppers:
0f1623511432bac0d8f2a87169952df0b341d90ea1e4218a851b8cdb2b691e2d
60599a679efb167cc43746e5d58bb8f74b6fe57cb028950fde79bd9fd0e6b48b
6c80e57f4957d17c80c0fc5e5809e72ac157a70339163579b7e2f3c0d631dd6b
8171f3ca246c56d85bdac23ab09ffdaea09410165bf32ed72ef279d2ddaf745b

Domains:
www (dot) aceactor (dot) co.kr - *Compromised website*
backup.microsoftappstore(dot)com
dataserver.cmonkey3(dot)com
google-helps(dot)com
kpupdate.amz80(dot)com
mail-help(dot)com
mail-issue(dot)top
microsoftupdating(dot)org
microsoftwww(dot)com
ns1.ccccc(dot)work
ns1.superman0x58(dot)com
ns1.xssr(dot)org
ns2.ccccc.work
ns2.superman0x58(dot)com
ns2.xssr(dot)org
qr1.3jd90dsj3df(dot)website
r4.microsoftupdating(dot)org
rouji.xssr(dot)org
t2z0n9.microsoftappstore(dot)com
temp.mail-issue(dot)top
time-service(dot)org
update.microsoftwww(dot)com
updatecz.mykorean(dot)net
uriupdate.newsbs(dot)net
wwgooglewww(dot)com
www.microsoftwww(dot)com
wwwgooglewww(dot)com
zy.xssr(dot)org

Suspected:
seo777.f3322(dot)net
sexy.f3322(dot)org
allmnz(dot)com
incsteelkor(dot)com

IP Addresses:
103.214.143.44
104.148.71.127
106.185.52.7
107.151.218.149
107.161.80.22
118.193.153.5
119.57.196.30
122.10.9.154
158.69.34.129
167.160.16.242
173.231.49.141
174.139.57.26
174.139.57.27
174.139.57.30
211.58.38.100
220.73.222.120
220.73.222.86
221.139.50.134
31.210.102.210
43.249.81.209
43.249.81.210
50.115.138.215
88.208.228.56
92.242.144.2

PDB Filepath:
D:\pdb\ht_d6.pdb

Yara Rule:

```
rule StreamEx
{
strings:
$a = "0r+8DQY97XGB5iZ4Vf3KsEt61HLoTOuIqJPp2AlncRCgSxUWyebhMdmzvFjNwka="
$b = {34 ?? 88 04 11 48 63 C3 48 FF C1 48 3D D8 03 00 00}
$bb = {81 86 ?? ?? 00 10 34 ?? 88 86 ?? ?? 00 10 46 81 FE D8 03 00 00}
$c = "greendll"
```

```
$d = "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/39.0.2171.95 Safari/537.36" wide
$f = {26 5E 25 24 23 91 91 91 91}
$g = "D:\\pdb\\ht_d6.pdb"

condition:
$a or $b or $bb or ($c and $d) or $f or $g
```

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