Red Leaves implant - overview

Ahmed Zaki David Cannings

March 2017

Contents

1 Handling information

This document was produced by the NCC Group Cyber Defence Operations team. The content of this document should be considered proprietary information. NCC Group has released this report publicly and gives permission to copy it at TLP WHITE. Please see the US CERT website for full details of the traffic light marking system.

2 Introduction

This technical note discusses a relatively undocumented implant used by the APT10 group. This is named "Red Leaves" after strings found in the malware. The sample discussed was found during an incident response engagement in March 2017. The earliest evidence obtained shows it has been in use since at least November 2016.

The name "Red Leaves" is mentioned in an article[1] by Yoshihiro Ishikawa from February 2017. At this time relatively few other samples can be found online.

3 Overview

3.1 Summary of files analysed

SHA256	Name
6bc2558eb8915edc19835d9e734023a2368f876971f5580478782c7444f9581c	aim.exe
02e702af02a6b9a8b31cd470c18e383093ef4ed404811b414d6d131df01f9acd	libcef.dll
79f61eda72c41b5ec526a3d5a1a91f86f0bc0eca470e07ab50d9626231143f11	pastime.dat

In an infected environment these files were also named `recy.exe`, `recyef.dll` and `recyime.dat` whilst being moved between hosts. They appear to have been copied in a ZIP file named `recy.zip`.

3.2 Execution flow

The sample analysed uses AOL Instant Messenger (AIM) with a custom DLL named `libcef.dll`, usually a genuine part of the AIM software. The file `pastime.dat` is XOR encoded on disk and contains stage 1 shellcode, stage 2 shellcode and the Red Leaves implant DLL.

The implant is loaded like so:

- `aim.exe` starts and loads `libcef.dll`.
- AIM calls a modified function named `cef_string_utf16_set`.
- The function `cef_string_utf16_set` loads `pastime.dat` and decodes it.
- The stage 1 shellcode is called. This launches `svchost.exe` and uses process hollowing to copy the stage 2 shellcode.
- The stage 2 shellcode allocates further memory inside `svchost.exe` and loads the Red Leaves implant DLL.
- The DLL is now running and AIM (the parent process) ends.

3.3 Timeline

The earliest known times obtained by NCC Group are shown below. Note that PE timestamps can be trivially altered. However, there is only 61 minutes between the compilation of `libcef.dll` and the first observed use.

[1] https://www.lac.co.jp/lacwatch/people/20170223_001224.html

Time (UTC)	Notes
2016-06-21 05:30:42	Potential compile time of Red Leaves DLL (once deobfuscated)
2016-11-10 08:15:14	Potential compile time of custom AIM DLL `libcef.dll`
2016-11-10 09:16:02	`libcef.dll` first seem in an infected environment

4 Files on disk

4.1 AOL Instant Messenger (AIM)

The sample analysed uses `aim.exe`[2] from AOL Instant Messenger to load. This has a valid authenticode signature from the 6th April 2015. This file is available in an AIM installer[3] from 2015.

There is nothing remarkable about this file except the use of `libcef.dll`, described below.

4.2 Implant loader - `libcef.dll`

The legitimate `libcef.dll` is found in an AOL Instant Messenger installer named `aim_install.exe`a from 2015 and is 23MB. This is the Chromium Embedded Framework[4] and has previously been used by APT10 to load PlugX.

The custom `libcef.dll` is 46KB and was likely compiled on November 10th 2016 at 08:15:14. This file was deployed onto an infected computer 61 minutes later.

The PE export directory for this file suggests it was compiled as `gentee.dll`. The function `cef_string_utf16_set` has been modified to load the Red Leaves stage 1 shellcode from disk and launch it. Unlike some DLL planting attacks all other functions are replaced with a call to `MessageBoxA()`, they are not proxied to the real DLL. Therefore AIM does not work correctly with the modified DLL and will exit shortly after launch.

A number of variants of the loader have been observed using both `StarBurn.dll` (part of a CD burning package) and `libcef.dll`. These use various techniques for encoding the implant on disk including:

- Single byte XOR.
- A ten byte XOR key stored at the start of the encoded `.dat` file.
- A ten byte XOR key (as above) with custom key rotation (the key byte used depends on the position in the file, rather than cycling `key[0]..key[9]`).

4.3 Encoded implant - `pastime.dat`

The analysed file is XOR encoded with the single byte `0x3D` and contains:

- The stage 1 shellcode, used by `libcef.dll`.
- The stage 2 shellcode, run inside `svchost.exe`.
- The implant DLL, loaded by stage 2 shellcode into `svchost.exe`.

This shellcode is responsible for launching the Red Leaves implant.

The implant DLL was originally named `red_autumnal_leaves_dllmain.dll` (data from the export directory) and is kept completely in memory, it is never written directly to disk unencoded.

[2] SHA256: 6bc2558eb8915edc19835d9e734023a2368f876971f5580478782c7444f9581c
[3] SHA256: a8cf3b5554fba856b878cfb3558e2092bce18109be0cddbd3e1dd7a95f8fb4e6
[4] https://bitbucket.org/chromiumembedded/cef

4.4 Other files

4.4.1 config.xml

On a number of infected machines a file named `config.xml` was found with the following contents:

```
<?xml version="1.0" encoding="UTF-8"?>
<config check="0">
</config>
```

This appears to relate to AIM and may prevent an automatic update check conducted by the software.

4.4.2 t.vbs

Use of `wmiexec.vbs`[5] was observed on one infected computer, where it was named `t.vbs`. This Visual Basic script uses WMI to run a command or start a shell on a remote system.

5 Malware analysis

5.1 Observed versions

Compile time	Notes
2016-06-21 05:30:42	Earliest known version. Configuration not encoded.
2016-11-24 01:56:27	Configuration still plaintext.
2016-12-12 05:56:56	Compile time from DLL embedded in `goat.dat`, found during IR.
2017-01-17 05:14:52	Compile time from DLL embedded in `handerchief.dat`.
2017-01-17 05:14:52	First observed encoded configuration (single byte XOR, 0x940 bytes long). Despite having an identical compile time this is different from the row above.

5.2 Capability overview

The Red Leaves sample analysed provides typical functionality for a software implant, including:

- Returning host information.
- Downloading a file from a remote server using HTTP.
- Deleting local files.
- File transfer between the infected computer and the C2 server.
- Running arbitrary commands, including an interactive shell.
- Taking a screenshot of the desktop.
- List currently logged in users.
- Modifying the implant configuration.
- File enumeration on the infected computer.
- Get disk information, such as free disk space, on the infected machine.
- Tunneling TCP connections.

Additional functionality is still being analysed at the time of writing.

Very limited information is saved to disk. The implant uses pipes for directing input/output from various threads in the 'RedLeavesCMDSimulator' mode which is message identifier `0x31`. In the sample we analysed the pipe name was

[5] https://github.com/Twi1ight/AD-Pentest-Script/blob/master/wmiexec.vbs

`NamedPipe_MoreWindows`. In that mode the implant uses a dedicated mutex `RedLeavesCmdSimulatorMutex` for synchronization between the multiple threads it creates (the implant creates a separate mutex during initialisation).

Most useful evidence during an IR engagement will come from memory analysis. However, evidence of files copied to or from the infected machine may be present on disk. The implant does not appear to conduct any anti-forensics such as overwriting files or zeroing memory.

5.3 Message identifiers

As more features are added to the implant new message identifiers are used to invoke those features. The earliest version of the implant recognised identifiers values `0x02`, `0x04`, `0x05`, `0x08`, `0x0C`, `0x0E`, `0x12`, `0x13`, `0x14`, `0x15` and `0x24` to `0x3B`. Several of those identifiers would not perform any operation such as `0x02`.

In the most recent sample, the message identifier `0x05` was not supported any more and as such the ability to alter the C2 protocol was not available. This sample also saw 4 new message identifiers added `0x38`, `0x39`, `0x3A` and `0x3B`. Additional message identifiers are used by the implant in response to certain commands.

The following table provides a summary of the message identifiers analysed to date. We have confirmed their operation through a combination of reverse engineering and creation of a C2 script which interacts with the implant using the custom binary C2 protocol.

Message ID	Notes
`0x04`	*tbc* - the “marks” command
`0x05`	Change C2 protocol configuration (HTTP, HTTPS or TCP) (deprecated in newer samples)
`0x08`	Find files on disk
`0x0C`	Get information about free space on drives
`0x0E`	Get host information
`0x12`	*tbc* - Start a command prompt (not visible to user)
`0x13`	*tbc* - Send input to the command prompt
`0x14`	*tbc* - Contains the result of commands sent to the command prompt
`0x15`	*tbc* - likely terminates the command pipe
`0x24`	Enumerate users (including RDP / terminal services)
`0x28`	Write to a file on the infected computer
`0x29`	Read from a file on the infected computer
`0x2A`	*tbc* - file transfer of some type, similar to `0x28` and `0x29`
`0x2B`	Delete a file from the system
`0x2C`	Run a command using `WinExec()` (interactive, GUI shown)
`0x2D`	Download from a URL to local file
`0x2E`	Start a tunnel by connecting to remote IP and port
`0x2F`	Send data through tunnel, also used by implant for received data
`0x30`	Close tunnel
`0x31`	*tbc* - start “command simulator”
`0x32`	*tbc* - likely sends data to “command simulator”
`0x33`	*tbc* - likely terminates “command simulator”
`0x34`	Takes a screenshot in BMP format
`0x36`	*tbc* - takes params like `strRemoteLanAddress`
`0x37`	*tbc* - likely closes sockets associated with `0x36`
`0x38`	Enhanced session enumeration (in newer samples)
`0x39`	Run a command with user impersonation (in newer samples)
`0x3A`	*tbc* takes ReverseIP and ReversePort paramters, potentially reverse shell (in newer samples)
`0x3B`	*tbc* likely closes sockets associated with `0x3A` (in newer samples)

Commands marked to be confirmed (*tbc*) are still undergoing analysis and testing with our custom C2 server.

5.3.1 0x05 - Change C2 protocol configuration

Switch the implant to a different mode of communication. This happens immediately and all further communication will use the new protocol.

Note that in HTTP mode the implant will use HTTPS if port 443 is configured, regardless of the protocol preference.

Parameter	Notes
protocol	String representing the protocol: http, https, tcp

5.3.2 0x08 - Find files

This command lists files using the FindFirstFile / FindNextFile APIs.

Parameter	Notes
findstr	Search term, for example C:*.*

The response looks like (newlines added for formatting):

```
__msgid=8
__serial=3
clientid=69DD56E8FBAEC546AB7CFDD8B0776770
result=$Recycle.Bin|22|0|130416865566508330:Boot|22|0|130730855383120431:bootmgr|39|
383786|129347834312112940:BOOTSECT.BAK|39|8192|130417187733936310:Documents and Sett
ings|9238|0|128920217365680370:MSOCache|8211|0|131329180725090372:NCC Group Rocks.tx
t|32|0|131347366685552453:pagefile.sys|38|4294434816|131344796806448125:PerfLogs|16|
0|128920152085554264:Program Files|17|0|131329180895302686:Program Files (x86)|17|0|
131329180895146686:ProgramData|8210|0|131329181457995674:Recovery|8214|0|13041686547
5872171:System Volume Information|22|0|131344749704809823:Users|17|0|130416865486480
190:Windows|16|0|131329181415719600:
```

On error the response will contain the following instead of result:

```
Answer=Error!
```

5.3.3 0x0C - Get free space information

There are no parameters. The response looks like:

```
__msgid=12
__serial=0
clientid=69DD56E8FBAEC546AB7CFDD8B0776770
result=A|2|0|0:C|3|64422408192|37059846144:
```

5.3.4 0x0E - Get host information

There are no parameters. The response looks like:

```
__msgid=14
__serial=0
OnlineTime=1490179751
```

```
address=878946496
clientid=314D599B40253142823341ED0162D54E
cpu=2712
cpuinfo=2
groups=2016-11-10
host=WIN-ANALYSIS
language=1033
marks=
memory=4095
privilege=User
protocol=1
system_platform=1
system_ver=12
system_vercode=(NT 6.1 Build 7601)
```

Observed values for `privilege` include `User` and `SYSTEM`.

5.3.5 `0x2B` - Delete file

This command deletes a file from the local disk.

Parameter	Notes
`clientpath`	Path to the file to delete

There does not appear to be a reply from this command.

5.3.6 `0x2C` - Execute a command (interactively)

Run a command, including interactive commands that use the GUI. For example, if `calc.exe` is run it will appear in the user session. At this time we have not tested what happens if the implant is running with `SYSTEM` privileges and no associated window station.

Parameter	Notes
`clientpath`	Path to the program or command to run

There does not appear to be a reply from this command.

5.3.7 `0x2D` - Download from URL

Downloads a file from a remote server to the infected host, using `UrlDownloadToFile()`.

Parameter	Notes
`url`	Remote URL to download from
`path`	Local path to save to

There does not appear to be a reply from this command.

5.3.8 0x34 - Take a screenshot

Obtains a screenshot of the desktop and returns it to the C2 server. When uncompressed this is a raw bitmap which is typically large, in excess of 10MB for a modern screen.

5.3.9 0x2E - Start tunnel

This command instructs the implant to connect to a remote host and port.

Parameter	Notes
`serverip`	The remote IP to connect to
`serverport`	The remote port to connect to

This command appears to reply with `result=1` regardless of whether the remote socket connected successfully or not.

5.4 Configuration

The configuration block contains:

- 3 domains or IP addresses. These are used in a round-robin fashion. Note that all three configuration slots must be filled. Samples analysed to date duplicate the same C2 domain into multiple slots.
- The port that will be used for communications.
- The communication mode (HTTP, HTTPS or TCP).
- The group identifier.
- A mutex name, used to ensure the implant only runs once.
- The RC4 encryption key.

In the sample analysed this information is unencoded in the Red Leaves DLL itself. A newer version of the implant from January 2017 XOR encodes the configuration.

5.4.1 Known groups

The following group names have been observed in samples or infected machines:

- `2016-11-10` - from IR
- `2016-12-15-NewDomain` - from IR
- `2016-12-16` - from IR
- `2016-12-23-valeo` - from sample found on VirusTotal
- `2017-2-22-ALL` - from sample uploaded to Hybrid Analysis

There does not appear to be any requirement for the group identifier to contain a date, but this is consistent in observed samples so far.

5.4.2 Known encryption keys

The following strings have been used as the RC4 key:

- `Lucky123` - from samples uploaded to VirusTotal and Hybrid Analysis
- `problems` - from IR
- `485621k8\x00` (note: the last character is a null byte, which is part of the key and not simply the string terminator) - from IR

5.5 Comments

Overall the implant appears to be a work in progress. There is limited error handling, some dead code and duplicated code that suggests a poor coding style (for example repeated memory cleanup at the end of functions). The use of C++ makes analysis slightly harder but there is no evidence of anti-debugging or similar techniques.

It is possible to crash threads or the entire implant with an improperly formatted C2 command. For example, sending message `0x2F` with no parameters will terminate `svchost.exe`.

6 C2 protocol

The implant supports HTTP, HTTPS and a custom binary protocol using TCP.

Each protocol has a specific value associated with it:

- 1 is TCP
- 2 is HTTP
- 3 is HTTPS
- 4 is TCP and HTTP (in newer samples).

Value 4 is a recognised identifier for the procotol to be used in more recent samples. In this mode the client uses TCP and HTTP for communication with the ability to choose the inital call back to one of 4 ports on the desitnation server 995, 80, 53 or 443.

In the sample analysed the TCP communication takes places on port 443. The custom binary protocol is detailed below. At this time limited analysis has been conducted of the HTTP and HTTPS modes.

6.1 TCP mode

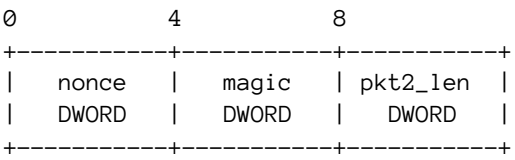
TCP mode uses sockets to obtain commands and report status and results. RC4 encryption is used with MiniLZO for compression of the raw data.

During our analysis a minimally functional C2 server was written to assist debugging. This can execute most commands and successfully retrieved a 16MB (uncompressed) screenshot of the test system.

Once connected in TCP mode it is possible to keep sending commands in the same stream. The implant connects regularly to the configured C2 servers. Each command or response is made up of two packets, described below.

6.1.1 Packet 1

The first packet is always 12 bytes long and contains three values:



The values are:

- `nonce` is a 32-bit value chosen at startup by the implant. This must be present in subsequent replies.
- `magic` is a 32-bit fixed value used to ensure syncronisation (`0xdc9b8d7a` in all analysed samples to date).
- `pkt2_len` is the total length of the second packet, which immediately follows.

6.1.2 Packet 2

The second packet contains the actual C2 data, either a command (from server to victim) or reply (from victim to server). It contains another 12 bytes of header followed by compressed and encrypted data.

```
0           4           8           C
+-----------+-----------+-----------+-----------~
|  raw_len  |  comp_len |   magic   |   data..
|   DWORD   |   DWORD   |   DWORD   |   char[]
+-----------+-----------+-----------+-----------~
```

The values are:

- `raw_len` is the uncompressed length of `data` (after MiniLZO).
- `comp_len` is the compressed length of `data` (as found in this packet).
- `magic` is a 32-bit fixed value used to validate the packet (`0xbfd9cbae` in all analysed samples to date, but see note below about XOR).
- `data` is a buffer containing the encrypted and compressed message.

All three headers in this packet are XOR with the first four bytes of the key. This was `4856` (or `0x34383536`) in the sample analysed.

To obtain raw data it is necessary to:

- Check that `comp_len` == the length of data received.
- RC4 decrypt the data payload with the key, which was `485621k8\x00` in the sample (note the null byte is part of the key).
- Decompress the buffer using MiniLZO. The data does not have an LZO header therefore `raw_len` must be passed.

Note that in some cases the use of MiniLZO makes data larger when compressed than in the raw format.

6.1.3 Key extraction

It is possible to obtain portions of the RC4 key from C2 traffic:

- Three bytes from the `raw_len` and `comp_len` headers in packet 2. The first message is always <255 in length so both `raw_len` and `comp_len` have three null bytes XOR with the key.
- Four bytes from the `magic` header in packet 2. All currently found implants use the same magic bytes, so `magic_hdr ^ magic_known` will give four bytes of the key.

It should be possible to extract the remainder of the key either from a list of previously observed keys or by a simple known-plaintext attack.

6.2 HTTP and HTTPS mode

Due to time constraints the HTTP(S) command has not been fully investigated. The implant can be configured to use HTTP(S) instead of a direct TCP connection using message `0x05` in the older samples. Note that in HTTP mode the implant automatically uses TLS if port 443 is configured.

The following user agent string is present in the malware however it does not appear to be used during HTTP C2 activity.

```
Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2;
.NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; .NET4.0C; .NET4.0E)
```

The analysed sample will POST to `/index.php`. The content of data in HTTP(S) mode is the same as the second packet, described above. Because HTTP has a defined structure and includes a `Content-Length` header there is no need for the first packet, which is used in raw TCP mode for synchronisation and provides the data length.

7 Known C2 domains

The following domains and IP addresses have been found inside the configuration block of Red Leaves samples:

- `owlmedia.mefound.com` - the domain configured in the analysed sample.
- `67.205.132.17`
- `144.168.45.116`
- `ctldl.windowsupdate.fartit.com`
- `ipv4.windowsupdate.fartit.com`
- `fgipv6.download.windowsupdate.com.mwcname.com` - note this appears to be a Chinese CDN
- `interpreter.shenajou.com`
- `center.shenajou.com`
- `commissioner.shenajou.com`

The current IP address for `owlmedia.mefound.com` is 151.236.20.16 which has previously been reported[6].

7.1 Interesting domains

The following domains were found during incident response investigations:

- `outlook.sindeali.com`
- `ultimedia.vmmini.com`
- `usiness.vmmini.com`

The attacker pinged the domain `outlook.sindeali.com` from an infected machine using an interactive shell launched by the Red Leaves implant. At this time this domain points to 151.236.20.16, the same as `owlmedia.mefound.com`.

The strings `ultimedia.vmmini.com` and `usiness.vmmini.com` were found in memory on an infected machine. The domain `vmmini.com` is known to be used by APT10. It was not possible to determine the context from the memory dump, however both point to 95.47.156.86. Note the subdomains `business` and `multimedia`, which are likely completions of these words, currently resolve to the loopback address 127.0.0.1.

8 Identifying C2 in memory

The implant does not appear to clear memory pages after use, therefore decoded command and control messages can be found in a memory dump. In most cases it is not possible to identify the time of the command, though some C2 messages do include epoch timestamps in the `OnlineTime` or `__serial` fields.

Below are example of two C2 command fragments found in memory. In these cases it is likely the `__serial` value is an epoch timestamp. Note that most data in memory is UTF-16 encoded, the exception being the output of pipes used for shell commands.

These fragments can be found in memory to form a timeline of how the Red Leaves malware has been used on an infected machine.

The attacker attempts to find files in the recycle bin:

```
__serial=1486349282
findstr=C:\RECYCLER\*.*
```

The attacker attempts to start the service `VirusKiller` which did not exist on the server:

```
__serial=1486354173
input=sc start VirusKiller
rn=20
```

[6] https://www.cylance.com/en_us/blog/the-deception-project-a-new-japanese-centric-threat.html

9 Detection

9.1 Generic detection

A number of generic detection mechanisms will assist in identifying the Red Leaves implant:

- Examining any `svchost.exe` processes that do not have `services.exe` as a parent.
- Examining any memory pages inside `svchost.exe` that are mapped as read-write-executable (RWX).
- Using the Volatility `malfind` plugin.

9.2 Yara rules

Full Yara rules are available in a separate document. Note, newlines have been added to the below for formatting.

```
rule malware_red_leaves_generic {
  meta:
    author = "David Cannings"
    description = "Red Leaves malware, related to APT10"

    // This hash from VT retrohunt, original sample was a memory dump
    sha256 = "2e1f902de32b999642bb09e995082c37a024f320c683848edadaf2db8e322c3c"

  strings:
    // MiniLZO release date
    $ = "Feb 04 2015"
    $ = "I can not start %s"
    $ = "dwConnectPort" fullword
    $ = "dwRemoteLanPort" fullword
    $ = "strRemoteLanAddress" fullword
    $ = "strLocalConnectIp" fullword
    $ = "\\\\.\\pipe\\NamePipe_MoreWindows" wide
    $ = "RedLeavesCMDSimulatorMutex" wide
    $ = "(NT %d.%d Build %d)" wide
    $ = "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0;
      SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; .NET4.0C;
      .NET4.0E)" wide
    $ = "red_autumnal_leaves_dllmain.dll" wide ascii
    $ = "__data" wide
    $ = "__serial" wide
    $ = "__upt" wide
    $ = "__msgid" wide

  condition:
    7 of them
}

rule malware_red_leaves_memory {
  meta:
    author = "David Cannings"
    description = "Red Leaves C&C left in memory, use with Volatility / Rekall"

  strings:
    $ = "__msgid=" wide ascii
    $ = "__serial=" wide ascii
```

```
    $ = "OnlineTime=" wide

    // Indicates a file transfer
    $ = "clientpath=" wide ascii
    $ = "serverpath=" wide ascii

  condition:
    3 of them
}
```

9.3 Suricata rules

The following rule detects the magic value in the first packet in TCP mode. This is consistent in all samples identified to date and should cause low false positives.

Signatures for HTTP C2 traffic are currently in testing.

```
alert tcp $HOME_NET any -> $EXTERNAL_NET any (msg: "NCC Group - Trojan - Red Leaves
magic packet detected (APT10 implant)"; flow:established,to_server;  dsize:12;
content:"|7a 8d 9b dc|"; offset: 4; depth: 4; flowbits:set,ncc.apt10.beacon_send;
threshold:type limit, track by_src, count 1, seconds 600; classtype:trojan-activity;
priority:1; sid:1; rev:1;)

alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg: "NCC Group - Trojan - Red Leaves
magic packet response detected (APT10 implant)"; flowbits:isset,ncc.apt10.beacon_send;
flow:established,to_client;  dsize:12; content:"|7a 8d 9b dc|"; offset: 4; depth: 4;
threshold:type limit, track by_dst, count 1, seconds 600; classtype:trojan-activity;
sid:2; rev:1;)
```

10 Similar files

10.1 Red Leaves implant

The following samples of Red Leaves are available online and are substantially similar to the version analysed in this technical note. Three files are the final implant taken from memory and appear to have been created by analysts.

SHA256	First upload
`2e1f902de32b999642bb09e995082c37a024f320c683848edadaf2db8e322c3c`	23 Feb 2017 (#1)
`fb4e516e1e2a369d1cdfb208ee885cb4848bed707a0514367f464c8e7519cb50`	09 Mar 2017 (#2)
`af9dde68c73d69ea535103e963f09587b6aa020081bbce06347de05fa469c257`	15 Mar 2017 (#3)
`ff0b79ed5ca3a5e1a9dabf8e47b15366c1d0783d0396af2cbba8e253020dbb34`	25 Jan 2017 (#4)

Note 1 - this was named `mem`.

Note 2 - this was named `ninjastuff.dat` and uploaded to Hybrid Analysis sandbox. It is not available for download however the XOR encoded configuration is displayed in the strings section and can be decoded.

Note 3 - this was named `dump.bin`.

Note 4 - this was named `authority.dat` and is an encoded version of the Red Leaves implant.

10.2 Red Leaves dropper

One file was found relating to the Red Leaves implant uploaded on 23rd February. A comment on the implant links to the 3.7MiB file below which is a dropper.

The executable has a fake Word document icon and the filename suggests targeting of a specific company.

SHA256	First upload
`5262cb9791df50fafcb2fbd5f93226050b51efe400c2924eecba97b7ce437481`	23 Feb 2017

11 Changes

Version	Changes
0.1	Internal draft.
0.2	Added details of more message IDs.
0.3	Clarified `libcef.dll`. Added additional domains and samples. Added extra message IDs.
1.0	First public release.

12 Contact details

To contact the authors with questions, suggestions or corrections please use the email address david.cannings@nccgroup.trust (GPG key `0x06211f5797f3b650`).

For all other queries about NCC Group please email response@nccgroup.trust who will direct your query appropriately.