Homeland Security

National Cybersecurity and Communications Integration Center

INCIDENT REPORT

Reference Number: IR-ALERT-MED-17-093-01C **April 27, 2017**

INTRUSIONS AFFECTING MULTIPLE VICTIMS ACROSS MULTIPLE SECTORS

Executive Summary

The National Cybersecurity and Communications Integration Center (NCCIC) has become aware of an emerging sophisticated campaign, occurring since at least May 2016, that uses multiple malware implants. Initial victims have been identified in several sectors, including information technology, energy, healthcare and public health, communications, and critical manufacturing.

According to preliminary analysis, threat actors appear to be leveraging stolen administrative credentials (local and domain) and certificates, along with placing sophisticated malware implants on critical systems. Some of the campaign victims have been IT service providers, where credential compromises could potentially be leveraged to access customer environments. Depending on the defensive mitigations in place, the threat actor could possibly gain full access to networks and data in a way that appears legitimate to existing monitoring tools.

Although this activity is still under investigation, NCCIC is sharing this information to provide organizations information for the detection of potential compromises within their organizations.

NCCIC will update this document as information becomes available.

To report activity related to this Incident Report Alert, please contact NCCIC at NCCICCustomerService@hq.dhs.gov or 1-888-282-0870.

Risk Evaluation

NCCIC Cyber Incident Scoring System (NCISS) Rating Priority Level (Color)
Yellow (Medium)
A medium priority incident may affect public health or safety, national security, economic security, foreign relations, civil liberties, or public confidence.

Details

While NCCIC continues to work with a variety of victims across different sectors, the adversaries in this campaign continue to affect several IT service providers. To achieve operational efficiencies and effectiveness, many IT service providers often leverage common core infrastructure that should be logically isolated to support multiple clients.

Intrusions into these providers create opportunities for the adversary to leverage stolen credentials to access customer environments within the provider network.

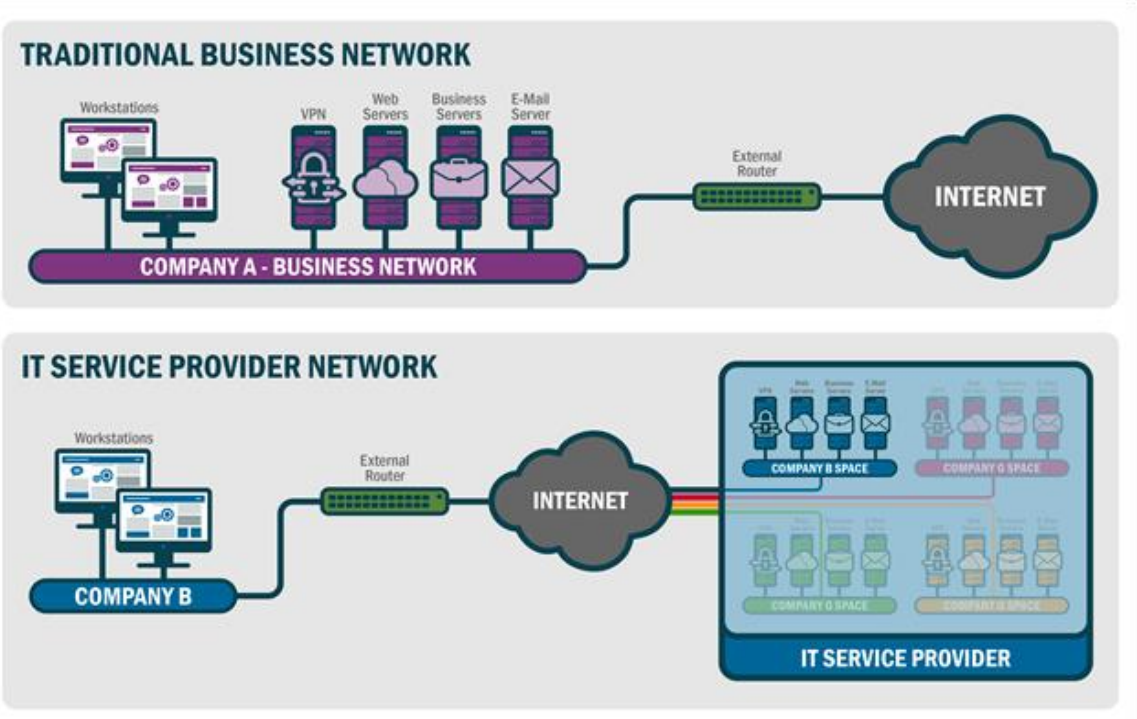

Figure 1: Structure of a traditional business network and an IT service provider network

Technical Analysis

The threat actors in this campaign have been observed employing a variety of tactics, techniques, and procedures (TTPs). The actors use malware implants to acquire legitimate credentials then leverage those credentials to pivot throughout the local environment. NCCIC is aware of several compromises involving the exploitation of system administrators' credentials to access trusted domains as well as the malicious use of certificates. Additionally, the adversary makes heavy use of PowerShell and the open source PowerSploit tool to enable assessment, reconnaissance, and lateral movement.

Command and Control (C2) primarily occurs using RC4 cipher communications over port 443 to domains that change IP addresses. Many of these domains spoof legitimate sites and content, with a particular focus on spoofing Windows update sites. Most of the known domains leverage dynamic DNS services, and this pattern adds to the complexity of tracking this activity. Listings of observed domains are found in this document's associated STIX package and .xlsx file. The indicators should be used to observe potential malicious activity on your network.

User impersonation via compromised credentials is the primary mechanism used by the adversary. However, a secondary technique to maintain persistence and provide additional access into the victim network is the use of malware implants left behind on key relay and staging machines. In some instances, the malware has only been found within memory with no on-disk evidence available for examination. To date, the actors have deployed multiple malware families and variants, some of which are currently not detected by anti-virus signatures. The observed malware includes PLUGX/SOGU and REDLEAVES. Although the observed malware is based on existing malware code, the actors have modified it to improve effectiveness and avoid detection by existing signatures.

Both REDLEAVES and PLUGX have been observed being executed on systems via dynamic-link library (DLL) side-loading. The DLL side-loading technique utilized by these malware families typically involves three files: a non-malicious executable, a malicious DLL loader, and an encoded payload file. The malicious DLL is named as one of the DLLs that the executable would normally load and is responsible for decoding and executing the payload into memory.

REDLEAVES Malware

The most unique implant observed in this campaign is the REDLEAVES malware. The REDLEAVES implant consists of three parts: an executable, a loader, and the implant shellcode. The REDLEAVES implant is a remote administration Trojan (RAT) that is built in Visual C++ and makes heavy use of thread generation during its execution. The implant contains a number of functions typical of RATs, including system enumeration and creating a remote shell back to the C2.

Capabilities

System Enumeration. The implant is capable of enumerating the following information about the victim system and passing it back to the C2:

- system name,
- system architecture (x86 or x64),
- operating system major and minor versions,
- amount of available memory,
- processor specifications,
- language of the user,
- privileges of the current process,
- group permissions of the current user,
- system uptime,
- IP address, and
- primary drive storage utilization.

Command Execution. The implant can execute a command directly inside a command shell using native Windows functionality by passing the command to run to cmd.exe with the "/c" option ("cmd.exe /c <command>").

Command Window Generation. The implant can also execute commands via a remote shell that is generated and passed through a named pipe. A command window is piped back to the C2 over the network as a remote shell or alternatively to another process or thread that can communicate with that pipe. The implant uses the mutexRedLeavesCMDSimulatorMutex.

File System Enumeration. The implant has the ability to enumerate data within a specified directory, where it gathers filenames, last file write times, and file sizes.

Network Traffic Compression and Encryption. The implant uses a form of LZO compression to compress data that is sent to its C2. After compression, the data for this implant sample is then RC4-ciphered with the key 0x6A6F686E3132333400 (this corresponds to the string "john1234" with the null byte appended).

Network Communications REDLEAVES connects to the C2 over TCP port 443, but does not use the secure flag when calling the API function InternetOpenUrlW. The data is not encrypted and there is no SSL handshake as would normally occur with port 443 traffic, but rather the data is transmitted in the form that is generated by the RC4 cipher.
Current REDLEAVES samples that have been examined have a hard-coded C2. Inside the implant's configuration block in memory were the strings in Table 1.

Table 1: REDLEAVES Sample Strings Found in C2

QN4869MD – mutex used to determine if the implant is already running (Varies from sample to sample)
2016-5-1-INCO –Unknown
%windir.\system32\svchost.exe - process that the implant was injected into
john1234 (with the null byte afterward) – RC4 Key

While the name of the initial mutex, QN4869MD in this sample, varies among REDLEAVES samples, the RedLeavesCMDSimulatorMutex mutex name appears to be consistent. Table 2 contains a sample of the implant communications to the domain windowsupdates.dnset[.]com over TCP port 443.

Table 2: REDLEAVES Sample Beacon

```
--- BEGIN SAMPLE BEACON ---
00000000 c1 0c 00 00 7a 8d 9b dc 88 00 00 00 ....z... ....
0000000C 14 6f 68 6e 16 6f 68 6e c4 a4 b1 d1 c4 e6 24 eb .ohn.ohn ......$.
0000001C cf 49 81 a7 a1 c7 96 ff 6d 31 b4 48 8b 3e a3 c1 .I...... m1.H.>..
0000002C 92 e2 c3 7c e4 4c cf e9 e1 fa fb 6a fa 66 2c bf ...|.L.. ...j.f,.
0000004C 7b 13 a7 30 17 3d eb fb d3 16 0e 96 83 21 2e 73 {..0.=.. .....!.s
0000005C dc 44 a2 72 fb f4 5e d0 4d b7 85 be 33 cd 13 21 .D.r..^. M...3..!
0000006C 3f e2 63 da da 5b 5e 52 9a 9c 20 36 69 cb cd 79 ?.c..[^R .. 6i..y
0000007C 13 79 7a d4 ed 63 b7 41 5d 38 b4 c2 84 74 98 cd .yz..c.A ]8...t..
0000008C f8 32 49 ef 2d e7 f2 ed .2I.-...
0000003C 5e 4b 72 6a f9 47 86 cd f1 cd 6d b5 24 79 3c 59 ^Krj.G.. ..m.$y
--- END SAMPLE BEACON ---
```

REDLEAVES network traffic has two 12-byte fixed-length headers in front of each RC4-encrypted compressed payload. The first header comes in its own packet, with the second header and the payload following in a separate packet within the same TCP stream. The last four bytes of the first header contain the number of the remaining bytes in little-endian format (0x88 in the sample beacon above).
The second header, starting at position 0x0C, is XOR'd with the first four bytes of the key that is used to encrypt the payload. In the case of this sample, those first four bytes would be "john" (or 0x6a6f686e using the ASCII hex codes). After the XOR operation, the bytes in positions 0x0C through 0x0F contain the length of the decrypted and decompressed payload. The bytes in positions 0x10 through 0x13 contain the length of the encrypted and compressed payload.

To demonstrate, in the sample beacon, the second header follows:

0000000C 14 6f 68 6e 16 6f 68 6e c4 a4 b1 d1

The length of the decrypted and decompressed payload is 0x7e000000 in little-endian format (0x146f686e XOR 0x6a6f686e). The length of the encrypted and compressed payload is

0x7c000000 in little-endian (0x166f686e XOR 0x6a6f686e). This is verified by referring back to the sample beacon which had the number of remaining bytes set to 0x88 and subtracting the length of the second header (0x88 – 0xC = 0x7c).

Strings

Note: Use caution when searching based on strings, as common strings may cause a large number of false positives.

Table 3: Strings Appearing in the Analyzed Sample of REDLEAVES

```
[ Unique Ascii strings ] ----------
red_autumnal_leaves_dllmain.dll
windowsupdates.dnset.com windowsupdates.dnset.com
windowsupdates.dnset.com
2016-5-10-INCO
john1234
Feb 04 2015
127.0.0.1 169.254
tcp
https
http
[ Unique Unicode strings ] ----------
RedLeavesCMDSimulatorMutex
QN4869MD
\\\\.\\pipe\\NamePipe_MoreWindows
network.proxy.type
network.proxy.http_port
network.proxy.http network.proxy.autoconfig_url
network.proxy.
a([a-zA-Z0-9])
b([ \\t])
c([a-zA-Z])
d([0-9])
h([0-9a-fA-F])
n(\r|(\r?\n)) q(\"[^\"]*\")|('[^']*')
w([a-zA-Z]+)
z([0-9]+)
```

Malware Execution Analysis

File Name: VeetlePlayer.exe
MD5: 9d0da088d2bb135611b5450554c99672
File Size: 25704 bytes (25.1 KB)
Description: This is the executable that calls the exports located within libvlc.dll

File Name: libvlc.dll
MD5: 9A8C76271210324D97A232974CA0A6A3
File Size: 33792 bytes (33.0 KB)
Description: This is the loader and decoder for mtcReport.ktc, the combined shellcode and implant file.

File Name: mtcReport.ktc
MD5: 3045E77E1E9CF9D9657AEA71AB5E8947
File Size: 231076 bytes (225.7 KB)
Description: This is the encoded shellcode and implant file. When this file is decoded, the shellcode precedes the actual implant, which resides at offset 0x1292 from the beginning of the shellcode in memory. The implant has the MZ and PE flags replaced with the value 0xFF.

All three of these files must be present for execution of the malware to succeed.

When all files are present and the VeetlePlayer.exe file is executed, it will make calls to the following DLL exports within the libvlc.dll file:

- VLC_Version checks to see if its calling file is named “VeetlePlayer.exe”. If the calling file is named something else, execution will terminate and no shellcode will be loaded.
- VLC_Create reads in the contents of the file mtcReport.ktc.
- VLC_Init takes in the offset in which the encoded shellcode/implant file is located and deobfuscates it. After deobfuscation, this export executes the shellcode.
- VLC_Destroy does nothing other than perform a return 0.
- VLC_AddIntf and VLC_CleanUp simply call the export VLC_Destroy, which returns 0.

When the libvlc.dll decodes the shellcode/implant, it calls the shellcode at the beginning of the data blob in memory. The shellcode then activates a new instance of svchost.exe and suspends it. It then makes a call to WriteProcessMemory() and inserts the implant with the damaged MZ and PE headers into its memory space. It then resumes execution of svchost.exe, which runs the implant.

The resulting decoded shellcode with the implant file below it can have a variable MD5 based on how it is dumped from memory. The MD5 checksums of two instances of decoded shellcode are:

1. ba4b4087370780dc988d55cbb9de885d
2. 3d032ba5f73cbc398f1a77af92077cd8

Table 4 contains the implant resulting from the original implant's separation from the shellcode and the repair of its MZ and PE flags.

Table 4: Resulting Implant from Shellcode Separation

File Name: red_autumnal_leaves_dllmain.dll **MD5**: 3EBBFEEE3A832C92BB60B531F749230E **File Size**: 226304 bytes (221.0 KB) **PE Compile Date**: 10 May 2016 During execution, the file will create two mutexes called RedLeavesCMDSimulatorMutex and QN4869MD. It checks the QN4869MD mutex to see if it is already running. It will then perform initial enumeration of the system to include operating system versions, number of processors, RAM, and CPU information.

PLUGX

PLUGX is a sophisticated Remote Access Tool (RAT) operating since approximately 2012. Although there are now many variants of this RAT in existence today, there are still characteristics common to most variants.

Typically, PLUGX uses three components to install itself.

1. A non-malicious executable
2. A malicious DLL/installer
3. An encoded payload – the PLUGX RAT.

A non-malicious executable with one or more imports is used to start the installation process. The executable will likely exist in a directory not normally associated with its use. In some cases, the actor may use an executable signed with a valid certificate, and rename the DLL and encoded payload with file names that suggest they are related to the trusted file. Importantly, the actor seems to vary the encoding scheme used to protect the encoded payload to stifle techniques used by AV vendors to develop patterns to detect it. The payload is either encoded with a single byte or encrypted and decompressed. Recently, NCCIC has observed a case where the encoded payload contains a decoding stub within itself, beginning at byte zero. The malware simply reads this payload and executes it starting at byte zero. The stub then decodes and executes the rest of itself in memory. Notably, this stub varies in its structure and algorithm, again stifling detection by signature based security software. The PLUGX malware is never stored on disk in an unencrypted or decoded format.

When the initial executable is launched, the imported library, usually a separate DLL, is replaced with a malicious version that in turn decodes and installs the third and final component, which is

the PLUGX rat itself. Typically, the PLUGX component is obfuscated and contains no visible executable code until it is unpacked in memory, protecting it from AV/YARA scans while static. During the evolution of these PLUGX compromises, NCCIC noted an increasing implementation of protections of the actual decoded PLUGX in memory. For example, the most recent version we looked at implements a secure strings method, which hides the majority of the common commands used by PLUGX. This is an additional feature designed to thwart signature based security tools.

Once the PLUGX RAT is installed on the victim, the actors has complete C2 capabilities of the victim system, including the ability to take screenshots and download files from the compromised system. The communications between the RAT (installed on the victim system) and the PLUGX C2 server are encoded to secure the communication and stifle detection by signature based network signature tools.

The advanced capabilities of PLUGX are implemented via a plugin framework. Each plugin operates independently in its own unique thread within the service. The modules may vary based on variants. Table 5 lists the modules and capabilities contained within one sample recently analyzed by NCCIC.

Table 5: Modules and Capabilities of PLUGX

Module Name	Capability
Disk	wide range of system-related capabilities including file / directory / drive enumeration, file / directory creation, create process, and obtain environment variables
Keylog	logs keystrokes and saves data to log file
Nethood	enumerates the host's network resources via the Windows multiple provider router DLL
Netstat	set the state of a TCP connection or obtain the extended TCP or UDP tables (lists of network endpoints available to a process) of each active process on the host
Option	provides the ability to initiate a system shutdown, adjust shutdown-related privileges for a given process, and lock the user's workstation
Portmap	port mapping
Process	process enumeration, termination, and capability to obtain more in-depth information pertaining to each process (e.g. CompanyName, FileDescription, FileVersion of each module loaded by the process)
Regedit	create, read, update & delete registry entries
Screen	capability to capture screenshots of the system
Service	start, stop, remove, configure & query services
Shell	remote shell access
SQL	enumerate SQL databases and available drivers; execute SQL queries

Module Name	Capability
Telnet	provides a telnet interface

The PLUGX operator may dynamically add, remove, or update PLUGX plugins during runtime. This provides the ability to dynamically adjust C2 capabilities based on the requirements of the C2 operator.

Network activity is often seen as POST requests similar to that shown in table 6. Network defenders can look to detect non-SSL HTTP traffic on port 443, which can be indicative of malware traffic. The PLUGX malware is also seen using TCP ports 80, 8080, and 53.

Table 6: Sample PLUGX Beacon

```
POST /D15DB9E25ADA34EC9E559736 HTTP/1.1

Accept:          */*
HX1:    0
HX2:    0
HX3:    61456
HX4:    1
User-Agent:     Mozilla/4.0 (compatible; MSIE 9.0; Windows NT 6.1; SLCC2; .NET CLR 2.0.50727;
.NET4.0C; .NET4.0E)
Host:    sc.weboot.info:443
Content-Length:          0
Cache-Control:  no-cache
```

Even though the beacon went to port 443, which is commonly used for encrypted HTTP communications, this traffic was plaintext HTTP, as is common for this variant of PLUGX.

For IT Service Providers

All organizations that provide IT services as a commodity for other organizations should evaluate their infrastructure to determine if related activity has taken place. Active monitoring of network traffic for the indicators of compromise (IOCs) provided in this report, as well as behavior analysis for similar activity, should be conducted to identify C2 traffic. In addition, frequency analysis should be conducted at the lowest level possible to determine any unusual fluctuation in bandwidth indicative of a potential data exfiltration. Both management and client systems should be evaluated for host indicators provided. If an intrusion is suspected, please reach out to the NCCIC at the contact information provided at the end of this report.

For Private Organizations and Government Agencies

All organizations should include the IOCs provided in their normal intrusion detection systems for continual analysis. Organizations that determine their risk to be elevated due to alignment to the sectors being targeted, unusual detected activity, or other factors, should conduct a dedicated investigation to identify any related activity. Organizations which leverage external IT service providers should validate with their providers that due diligence is being conducted to validate if there are security concerns with their specific provider. If an intrusion is suspected, please reach out to the NCCIC at the contact information provided at the end of this report.

Detection

NCCIC is providing a compilation of IOCs from a variety of sources to aid in the detection of this malware. The IOCs provided in the associated STIX package and .xlsx file were derived from various government, commercial, and publically available sources. The sources provided does not constitute an exhaustive list and the U.S. Government does not endorse or support any particular product or vendor's information listed in this report. However, NCCIC includes this compilation here to ensure the distribution of the most comprehensive information. This alert will be updated as additional details become available.

Table 7: Sources Referenced

Source	Title
PaloAltoNetworks	"menuPass Returns with New Malware and New Attacks Against Japanese Academics and Organizations"
FireEye	"APT10 (Menupass Team) Renews Operations Focused on Nordic Private Industry; operations Extend to Global Partners". February 23, 2017 10:14:00 AM,17-00001858, Version: 2
CyLance	"The Deception Project: A New Japanese-Centric Threat"
PwC/BAE Systems	"Operation Cloud Hopper: Exposing a systematic hacking operation with an unprecedented web of global victims: April 2017"
JPCERT/CC	"RedLeaves-Malware Based on Open Source Rat" http://blog.jpcert.or.jp/2017/04/redleaves---malware-based-on-open-source-rat.html
NCC Group	"RedLeaves Implant-Overview"
National Cyber Security Centre	"Infrastructure Update Version 1.0" Reference: March 17, 2017"
FireEye	"BUGJUICE Malware Profile". April 05, 2017 11:45:00 AM, 17-00003261, Version: 1
JPCERT/CC	"ChChes- Malware that Communicates with C&C Servers Using Cookie Headers" http://blog.jpcert.or.jp/2017/02/chches-malware--93d6.html

NCCIC recommends monitoring activity to the following domains and IP addresses, and scanning for evidence of the file hashes as potential indicators of infection. Some of the IOCs provided may be associated with legitimate traffic. Nevertheless, closer evaluation is warranted if the IOCs are observed. If these IOCs are found, NCCIC can provide additional assistance in further investigations. A comprehensive listing of IOCs can be found in the associated STIX package and .xlsx file.

Network Signatures

Table 8: REDLEAVES Network Signatures

```
 alert tcp any any -> any any (msg: "REDLEAVES Implant"; content: "|00 00 7a 8d 9b dc|"; offset: 2; depth: 6; content: "|00 00|"; offset: 10; depth: 2; sid: 314;)


alert tcp any -> any any (msg:”Suspicious PLUGX URI String”; content:”POST”; http_method; content:”/update?id=”; http_uri; fast_pattern:only; pcre:”/update\?id=[a-fA-F0-9]{8} HTTP/”; sid:101;)
```

Table 9: REDLEAVES YARA Signatures

```
rule Dropper_DeploysMalwareViaSideLoading {
   meta:
         description = "Detect a dropper used to deploy an implant via side loading. This dropper has specifically been observed deploying REDLEAVES & PlugX"
         author = "USG"
         true_positive = "5262cb9791df50fafcb2fbd5f93226050b51efe400c2924eecba97b7ce437481: drops REDLEAVES. 6392e0701a77ea25354b1f40f5b867a35c0142abde785a66b83c9c8d2c14c0c3: drops plugx. "
   strings:
         $UniqueString = {2e 6c 6e 6b [0-14] 61 76 70 75 69 2e 65 78 65} // ".lnk" near "avpui.exe"
         $PsuedoRandomStringGenerator = {b9 1a [0-6] f7 f9 46 80 c2 41 88 54 35 8b 83 fe 64} // Unique function that generates a 100 character pseudo random string.
   condition:
         any of them
}
```

```
rule REDLEAVES_DroppedFile_ImplantLoader_Starburn {
   meta:
         description = "Detect the DLL responsible for loading and deobfuscating the DAT file containing shellcode and core REDLEAVES RAT"
         author = "USG"
         true_positive = "7f8a867a8302fe58039a6db254d335ae" // StarBurn.dll
   strings:
         $XOR_Loop = {32 0c 3a 83 c2 02 88 0e 83 fa 08 [4-14] 32 0c 3a 83 c2 02 88 0e 83 fa 10} // Deobfuscation loop
   condition:
         any of them
}
```

```
rule REDLEAVES_DroppedFile_ObfuscatedShellcodeAndRAT_handkerchief {
    meta:
        description = "Detect obfuscated .dat file containing shellcode and core REDLEAVES RAT"
        author = "USG"
        true_positive = "fb0c714cd2ebdcc6f33817abe7813c36" // handkerchief.dat
    strings:
        $RedleavesStringObfu = {73 64 65 5e 60 74 75 74 6c 6f 60 6d 5e 6d 64 60 77 64 72 5e 65 6d 6d 6c
60 68 6f 2f 65 6d 6d} // This is 'red_autumnal_leaves_dllmain.dll' XOR'd with 0x01
    condition:
        any of them
}
```

```
rule REDLEAVES_CoreImplant_UniqueStrings {
    meta:
        description = "Strings identifying the core REDLEAVES RAT in its deobfuscated state"
        author = "USG"
    strings:
        $unique2 = "RedLeavesSCMDSimulatorMutex" nocase wide ascii
        $unique4 = "red_autumnal_leaves_dllmain.dll" wide ascii
        $unique7 = "\\NamePipe_MoreWindows" wide ascii
    condition:
      any of them
}
```

Table 10: PLUGX Network Signatures

```
alert tcp any any -> any any (msg:"Non-Std TCP Client Traffic contains 'HX1|3a|' 'HX2|3a|' 'HX3|3a|'
'HX4|3a|' (PLUGX Variant)"; sid:XX; rev:1; flow:established,to_server; content:"Accept|3a 20 2a 2f 2a|";
nocase; content:"HX1|3a|"; distance:0; within:6; fast_pattern; content:"HX2|3a|"; nocase; distance:0;
content:"HX3|3a|"; nocase; distance:0; content:"HX4|3a|"; nocase; distance:0; classtype:nonstd-tcp;
priority:X;)
```

```
alert tcp any any -> any any (msg:"Non-Std TCP Client Traffic contains 'X-Session|3a|"X-Status|3a|"X-
Size|3a|"X-Sn|3a|'(PLUGX)"; sid:XX; rev:1; flow:established,to_server; content:"X-Session|3a|"; nocase;
fast_pattern; content:"X-Status|3a|"; nocase; distance:0; content:"X-Size|3a|"; nocase; distance:0;
content:"X-Sn|3a|"; nocase; distance:0; classtype:nonstd-tcp; priority:X;)
```

```
alert tcp any any -> any any (msg:"Non-Std TCP Client Traffic contains 'MJ1X|3a|' 'MJ2X|3a|' 'MJ3X|3a|'
'MJ4X|3a|' (PLUGX Variant)"; sid:XX; rev:1; flow:established,to_server; content:"MJ1X|3a|"; nocase;
fast_pattern; content:"MJ2X|3a|"; nocase; distance:0; content:"MJ3X|3a|"; nocase; distance:0;
content:"MJ4X|3a|"; nocase; distance:0; classtype:nonstd-tcp; priority:X;)
```

```
alert tcp any any -> any any (msg:"Non-Std TCP Client Traffic contains 'Cookies|3a|' 'Sym1|2e|'
'|2c|Sym2|2e|' '|2c|Sym3|2e|' '|2c|Sym4|2e|' (Chches Variant)"; sid:XX; rev:1; flow:established,to_server;
content:"Cookies|3a|"; nocase; content:"Sym1|2e|0|3a|"; nocase; distance:0; fast_pattern;
content:"|2c|Sym2|2e|"; nocase; distance:0; content:"|2c|Sym3|2e|"; nocase; distance:0;
content:"|2c|Sym4|2e|"; nocase; distance:0; classtype:nonstd-tcp; priority:X;)
```

Host Signatures

Table 11: PLUGX and REDLEAVES YARA Signatures

```
rule PLUGX_RedLeaves
{
meta:
author = "US-CERT Code Analysis Team"
date = "03042017
incident = "10118538"
date = "2017/04/03"
MD5_1 = "598FF82EA4FB52717ACAFB227C83D474"
MD5_2 = "7D10708A518B26CC8C3CBFBAA224E032"
MD5_3 = "AF406D35C77B1E0DF17F839E36BCE630"
MD5_4 = "6EB9E889B091A5647F6095DCD4DE7C83"
MD5_5 = "566291B277534B63EAFC938CDAAB8A399E41AF7D"
info = "Detects specific RedLeaves and PlugX binaries"
strings:
$s0 = { 80343057403D2FD0010072F433C08BFF80343024403D2FD0010072F4 }
$s1 = "C:\\Users\\user\\Desktop\\my_OK_2014\\bit9\\runsna\\Release\\runsna.pdb"
$s2 = "d:\\work\\plug4.0(shellcode)"
$s3 = "\\shellcode\\shellcode\\XSetting.h"
$s4 = { 42AFF4276A45AA58474D4C4BE03D5B395566BEBCBDEDE9972872C5C4C5498228 }
$s5 = { 8AD32AD002D180C23830140E413BCB7CEF6A006A006A00566A006A00 }
$s6 = { EB055F8BC7EB05E8F6FFFFFF558BEC81ECC8040000535657 }
$s7 = {
8A043233C932043983C10288043283F90A7CF242890D18AA00103BD37CE2891514AA00106A006A00
6A0056 }
$s8 = { 293537675A402A333557B05E04D09CB05EB3ADA4A4A40ED0B7DAB7935F5B5B08 }
$s9 = "RedLeavesCMDSimulatorMutex"
condition:
$s0 or $s1 or $s2 and $s3 or $s4 or $s5 or $s6 or $s7 or $s8 or $s9
}
```

Other Detection Methods

Examine Port/Protocol Mismatches: Examine network traffic where the network port and protocol do not match, such as plaintext HTTP over port 443.

Administrative Share Mapping: When a malicious actor tries to move laterally on a network, one of the techniques is to mount administrative shares to perform operations like uploading and downloading resources or executing commands. In addition, tools like System Internals PSEXEC will mount the shares automatically for the user. Since administrators may map

administrative shares legitimately while managing components of the network, this must be taken into account.

- Filter network traffic for SMB mapping events and group the events by source IP, destination IP, the mounted path (providing a count of total mounts to that path), the first map time, and the last map time
- Collect Windows Event Logs – Event ID 5140 (network share object was accessed) can be used to track C$ and ADMIN$ mounts by searching the Share Name field

VPN User authentication mismatch: A VPN user authentication match occurs when a user account authenticates to an IP address but once connected the internal IP address requests authentication tokens for other users. This may create false positives for legitimate network administrators but if this is detected, organizations should verify that the administrative accounts were legitimately used.

VPN activity from VPS providers: While this may also produce false positives, VPN logins from Virtual Private Server (VPS) providers may be an indicator of VPN users attempting to hide their source IP and should be investigated.

Mitigations

Properly implemented defensive techniques and programs make it more difficult for an adversary to gain access to a network and remain persistent yet undetected. When an effective defensive program is in place, actors should encounter complex defensive barriers. Actor activity should also trigger detection and prevention mechanisms that enable organizations to contain and respond to the intrusion more rapidly. There is no single or set of defensive techniques or programs that will completely avert all malicious activities. Multiple defensive techniques and programs should be adopted and implemented in a layered approach to provide a complex barrier to entry, increase the likelihood of detection, and decrease the likelihood of a successful compromise. This layered mitigation approach is known as defense-in-depth.

NCCIC mitigations and recommendations are based on observations made during the hunt, analysis, and network monitoring for threat actor activity, combined with client interaction.

Whitelisting

- Enable application directory whitelisting through Microsoft Software Restriction Policy (SRP) or AppLocker;
- Use directory whitelisting rather than trying to list every possible permutation of applications in an environment. Safe defaults allow applications to run from PROGRAMFILES, PROGRAMFILES(X86), and SYSTEM32. All other locations

should be disallowed unless an exception is granted.
- Prevent the execution of unauthorized software by using application whitelisting as part of the security hardening of operating systems insulating.
- Enable application directory whitelisting via the Microsoft SRP or AppLocker.

Account Control

- Decrease a threat actor's ability to access key network resources by implementing the principle of least privilege.
- Limit the ability of a local administrator account to login from a local interactive session (e.g., "Deny access to this computer from the network") and prevent access via a Remote Desktop Protocol session.
- Remove unnecessary accounts, groups, and restrict root access.
- Control and limit local administration.
- Make use of the Protected Users Active Directory group in Windows Domains to further secure privileged user accounts against pass-the-hash compromises.

Workstation Management

- Create a secure system baseline image and deploy to all workstations.
- Mitigate potential exploitation by threat actors by following a normal patching cycle for all operating systems, applications, software, and all third-party software.
- Apply asset and patch management processes.
- Reduce the number of cached credentials to one if a laptop, or zero if a desktop or fixed asset.

Host Based Intrusion Detection

- Configure and monitor system logs through host-based intrusion detection system (HIDS) and firewall.
- Deploy an anti-malware solution to prevent spyware, adware, and malware as part of the operating system security baseline.
- Monitor antivirus scan results on a regular basis.

Server Management

- Create a secure system baseline image, and deploy to all servers.
- Upgrade or decommission end-of-life non Windows servers.
- Upgrade or decommission servers running Windows Server 2003 and older versions.
- Implement asset and patch management processes.
- Audit for and disable unnecessary services.

Server Configuration and Logging

- Establish remote server logging and retention.
- Reduce the number of cached credentials to zero.
- Configure and monitor system logs via a centralized security information and event management (SIEM) appliance.
- Add an explicit DENY for "%USERPROFILE%".
- Restrict egress web traffic from servers.
- In Windows environments, utilize Restricted Admin mode or remote credential guard to further secure remote desktop sessions against pass-the-hash compromises.
- Restrict anonymous shares.
- Limit remote access by only using jump servers for such access.

Change Control

- Create a change control process for all implemented changes.

Network Security

- An Intrusion Detection System (IDS) should:
  - Implement continuous monitoring.
  - Send alerts to a SIEM tool.
  - Monitor internal activity (this tool may use the same tap points as the netflow generation tools).
- Netflow Capture should:
  - Set a minimum retention period of 180 days.
  - Capture netflow on all ingress and egress points of network segments, not just at the Managed Trusted Internet Protocol Services (MTIPS) or Trusted Internet Connections (TIC) locations.
- Network Packet Capture (PCAP):
  - Retain PCAP data for a minimum of 24 hours.
  - Capture traffic on all ingress and egress points of the network.
- Use a virtual private network (VPN):
  - Maintain site-to-site VPN with customers.
  - Authenticate users utilizing site-to-site VPNs through adaptive security appliance (ASA).
- Use authentication, authorization, and accounting (AAA) for controlling network access.
  - Require Personal Identity Verification (PIV) authentication to an HTTPS page on the ASA in order to control access. Authentication should also require explicit

rostering of PIV distinguished names (DNs) that are permitted to enhance the security posture on both networks participating in the site-to-site VPN.
    - Establish appropriate secure tunneling protocol and encryption.
    - Strengthen router configuration (e.g., avoid enabling remote management over the Internet and using default IP ranges; automatically logout after configuring routers; use encryption).
    - Turn off Wi-Fi protected setup (WPS), enforce the use of strong passwords, keep router firmware up-to-date; and
    - Improve firewall security (e.g., enable auto updates, revise firewall rules as appropriate, implement whitelists, establish packet filtering, enforce the use of strong passwords, and encrypt networks).
- Conduct regular vulnerability scans of the internal and external networks and hosted content to identify and mitigate vulnerabilities.
- Define areas within the network that should be segmented to increase visibility of lateral movement by an adversary and increase the defense in-depth posture.
- Develop a process to block traffic to IP addresses and domain names that have been identified as being used to aid previous malicious activities.

Network Infrastructure Recommendations

- Remove unnecessary OS files from the IOS/ASA devices. This will limit the possible targets of persistence (i.e., files to embed malicious code) if the device is compromised, and will align with National Security Agency (NSA) network device integrity (NDI) best practices.
- Remove vulnerable IOS/ASA operating system files (older iterations) from the device's boot variable (i.e., show boot or show bootvar).
- Update to the latest available operating system for Cisco IOS and Cisco ASA devices.
- On ASA devices, update Cisco Adaptive Security Device Manager to version 7.6.2 or later to reduce vulnerabilities and maintain consistent software versions on firewalls throughout the organization.
- For ASA devices with the SSL VPN enabled, routinely verify customized web objects against the organization's known good files for such VPNs, to ensure the ASA devices remain free of unauthorized modification.

Host Recommendations

- Implement policies to block workstations-to-workstation remote desktop protocol (RDP) connections through group policy object (GPO) on Windows, or a similar mechanism.
- Store system logs of mission critical systems for at least one year within a SIEM.
- Review the configuration of application logs to verify fields being recorded will

contribute to an incident response investigation.

Users Management

- Immediately set the password policy to require complex passwords for all users (minimum of 15 characters); this new requirement should be enforced as user passwords expire.
- Reduce the number of domain and enterprise administrator accounts.
- Create non-privileged accounts for privileged users and ensure they use the non-privileged account for all non-privileged access (e.g., web browsing, email access);
- If possible, use technical methods to detect or prevent browsing by privileged accounts (authentication to web proxies would enable blocking of domain administrators).
- Use two-factor authentication (e.g., security tokens for remote access and to any sensitive data repositories);
- If soft tokens are used, they should not exist on the same device that is requesting remote access (laptop), and instead should be on a telephone or other out-of-band device.
- Create privileged role tracking;
  - Create a change control process to all privilege escalations and role changes on user accounts;
  - Enable alerts on privilege escalations and role changes; and
  - Log privileged user changes in the environment and alert on unusual events.
- Establish least privilege controls; and
- Implement a security-awareness training program.

Best Practices

- Implement a vulnerability assessment and remediation program.
- Encrypt all sensitive data in transit and at rest.
- Create an insider threat program.
- Assign additional personnel to review logging and alerting data.
- Complete independent security (not compliance) audit.
- Create an information sharing program.
- Complete and maintain network and system documentation to aid in timely incident response, including:
  - network diagrams,
  - asset owners,
  - type of asset, and
  - an up-to-date incident response plan.

Contact Us

For questions related to this report, contact NCCIC 24 hours, 7 days a week. Include the report reference number in the subject line of all email correspondence.

Toll Free: 1-888-282-0870

International: 703-235-8832

Email: NCCICCustomerService@us-cert.gov

Feedback

NCCIC continuously strives to improve its products and services. You can help by answering a few short questions about this product at the following URL: https://www.us-cert.gov/forms/feedback.