



LodaRAT Threat Report

Date: 14/10/2020
Krupa Gajjar

A RAT or Remote Access Trojan is a form of malware which allows hackers to control victim machine remotely. It allows hacker for covert surveillance of the victim machine. Hackers can use the compromised machine to perform various activities such as installing additional malwares, deleting programs, webcam hijacking, read the data from keyboard, acquiring login credentials, and clipboard data.

Overview

This RAT known as LodaRAT was first discovered in 2017, it has a variety of capabilities like spying on victim machines, and a newer version of this malware was later discovered in February 2020, by Cisco Talos. This RAT written in AutoIt script, so far targeted countries in South America and Central America. The sample analyzed in this blog is the most recent variant of this RAT discovered in September 2020 by Cisco Talos.

Technical Analysis

The variant discovered in the month of February propagated via malicious Microsoft Word document which downloaded another document which exploited a vulnerability CVE-2017-11882, this exploit payload in turn downloaded an MSI file which contained AutoIt Script.

The method of propagation of the new variant of the LodaRAT is much simple; it is now being distributed via social engineering such as email phishing campaigns attached in the form of a RAR archive. These RAR files have the file extension '.rev' which contains compiled AutoIt Script.

The execution of the binary depends on the user or victim double-clicking the executable file on the machine. Although multiple versions of this RAT were found to be distributed at the same time, but the overall functionality of all the versions are similar with some key differences.

Upon execution the malware tries to detect the version of the OS on the victim machine and then drops its own copy in temp or startup directories of the current user

C:\Users\%User%\AppData\Roaming\Windata\PXQIKA.exe so as to maintain persistence (Fig 1).

```

text:0040E500 VersionInformation= OSVERSIONINFOW ptr -1
text:0040E500 var_44 = byte ptr -44h
text:0040E500 var_42 = byte ptr -42h
text:0040E500 SystemInfo = _SYSTEM_INFO ptr -40h
text:0040E500 hLibModule = dword ptr -1Ch
text:0040E500 var_14 = dword ptr -14h
text:0040E500 var_10 = dword ptr -10h
text:0040E500 var_8 = dword ptr -8
text:0040E500 var_4 = dword ptr -4

text:0040E52A call ds:GetVersionEx
text:0040E530 mov eax, [ebp+VersionInformation.dwBuildNumber]
text:0040E536 mov ecx, [ebp+VersionInformation.dwMajorVersion]
text:0040E53C mov edx, [ebp+VersionInformation.dwMinorVersion]
text:0040E542 mov [edi+8], eax
text:0040E545 lea ebx, [ebp+VersionInformation.szCSDVersion]
text:0040E548 mov eax, esi
text:0040E54D mov [edi], ecx

data:00491840 dd offset aWorkingdir ; "WORKINGDIR"
data:00491844 dd offset aOstype ; "OSTYPE"
data:00491848 dd offset aOsver ; "OSVERSION"
data:0049184C dd offset aOsbuild ; "OSBUILD"
data:00491850 dd offset aOsservicepack ; "OSSERVICEPACK"
data:00491854 dd offset aOslang ; "OSLANG"
data:00491858 dd offset aProcessorarch ; "PROCESSORARCH"
data:0049185C dd offset aOsarch ; "OSARCH"
data:00491860 dd offset aCpuarch ; "CPUARCH"

```

Fig 1

The malware steals multiple sensitive data from the victim machine such as Startup info, Time Zone information. The binary also fetches Clipboard data and password

```

__security_init_cookie();
local_8 = $DAT_0048d0c8;
nStack12 = 0w416460;
GetStartupInfoW(LPSTARTUPINFOW)&local_6);
if ( _DAT_004a972c == 0 ) {
    (*_DAT_00490008)(0,1,0,0);
}

idata:00482380 ; DWORD _stdcall GetTimeZoneInformation(LPTIME_ZONE_INFORMATION lpTimeZoneInformation)
idata:00482380          extrn GetTimeZoneInformation:dword
idata:00482380          ; CODE XREF: __tzset_nolock+129fp
idata:00482380          ; DATA XREF: __tzset_nolock+129fr
idata:00482384 ; BOOL _stdcall FreeEnvironmentStringsW(LPWCH)

text:0046D001          call ds:GetClipboardData
text:0046D007          mov esi, eax
text:0046D009          test esi, esi
text:0046D00B          jnz short loc_46D036
text:0046D00D          call ds:CloseClipboard
text:0046D013          mov ecx, [edi]
text:0046D015          mov eax, [ecx+4]
text:0046D018          push esi

```

Fig 2

The malware has implemented a Powershell keylogger to read the data from the keyboard and log the information about the keys pressed by the user. The Powershell script used for keylogging has been copied from a short blog.

```

text:0044488E          mov ebx, ds:GetAsyncKeyState
text:00444894          push 0A0h ; ' ' ; vKey
text:00444899          call ebx ; GetAsyncKeyState
text:0044489B          mov ecx, 8000h
text:004448A0          test cx, ax
text:004448A3          jnz short loc_4448BA
text:004448A5          push 0A0h ; ' ' ; nVirtKey
text:004448AA          call ds:GetKeyState
text:004448B0          mov edx, 8000h
text:004448B5          test dx, ax
text:004448B8          jz short loc_4448C3
text:004448BA

text:00444450          call ds:GetKeyboardState
text:00444456          test eax, eax
text:00444458          jz short loc_4444AA
text:0044445A          cmp byte ptr [edi], 0
text:0044445D          mov al, 80h ; 'e'
text:0044445F          jz short loc_44446A
text:00444461          xor [ebp+var_F0], al
text:00444467          xor [ebp+var_60], al
text:0044446A

text:00445ED1          call ebx ; MapVirtualKeyW
text:00445ED3          mov ecx, [esi]
text:00445ED5          shl eax, 10h
text:00445ED8          or eax, 0C0000001h
text:00445EDD          push eax ; lParam
text:00445EDE          push 27h ; ' ' ; wParam
text:00445EE0          push 101h ; Msg
text:00445EE5          push ecx ; hWnd
text:00445EE8

```

Fig 3

After communicating with the CnC server the executing binary logs the output of the key press in a temp directory as a text file. This log file will be displayed once the keylogger script gets aborted by pressing Ctrl+C.

The binary uses HTTP GET request with its custom header and body to connect to the C&C server to get further information and communication. However, the malware abuses legitimate services such as Ngrok.io and portmap.io. The malware downloads ng.txt file in the victim machine while communicating with the CnC server.

```

GET /ng.txt HTTP/1.1
User-Agent: AutoIt
Host: roodan888tools.atwebpages.com
Cache-Control: no-cache

HTTP/1.1 200 OK
Date: Mon, 12 Oct 2020 04:45:00 GMT
Server: Apache
Last-Modified: Tue, 01 Sep 2020 07:23:31 GMT
ETag: "15-5ae3b656adf4e"
Accept-Ranges: bytes
Content-Length: 21
Content-Type: text/plain

0.tcp.ngrok.io:10662

```

Fig 4

Communicating IP addresses: 185[.]176[.]43[.]100
3[.]17[.]7[.]232

External Domain: roodan888tools[.]atwebpages[.]com
0[.]tcp[.]ngrok[.]io

CnC : http://roodan888tools[.]atwebpages[.]com/ng[.]txt

While communicating with the CnC server the victim machine receives several commands such as to get the current window of the user’s machine. It also receives a Screen command, which sends screenshot of the current window of the infected machine back to CnC server and this is sent at regular intervals.

<pre> void UndefinedFunction_00462986(void) { HWND pHVar1; uint uVar2; pHVar1 = GetForegroundWindow(); uVar2 = FUN_00430c09((int)sDAT_004a8630,pHVar1); if (uVar2 != 0xffffffff) { FUN_00456391((int)sDAT_004a8630, (HWND *) (* (HWND *) (DAT_) return; } </pre>	<pre> text:0046FDF2 call ds:GetForegroundWindow text:0046FDF8 mov edx, [esi+2E8h] text:0046FDFE mov ecx, 1 text:0046FE03 cmp edx, ecx text:0046FE05 jl short loc_46FE1D text:0046FE07 mov esi, [esi+2E4h] text:0046FE0D mov edi, edi </pre>
--	---

Fig 5

File Hash: ab5b9dd6a6650476fc891df0834ee6f0

IOCs:

55344011c4ecd81be4ad18b091b8028c
5468137378656f6525ef86b79ab48012
68bd21f2938c88ebecf906e8ec9a9d36
010fe248f5c918cea3051df45a64b420
19a108631331c5d533b0b0788cabafee (RAR archive)
81d2eacd6fa43b2560865e44a87780a2
1515a5c693145dfb8b7a81050d4ad7c1 (RAR archive)
2d5c61935f9d870e940ec8e7c42ae26b (RAR archive)

MITRE Techniques:

T1124 - System Time Discovery	T1204 - User Execution
T1012 - Query Registry	T1190 - Spearphishing Attachment
T1027 - Obfuscated Files or Information	T1546.013 - Event Triggered Execution(Powershell Profile)
T1033 - System Owner/User Discovery	T1060 - Registry Run Keys/ Startup Folder
T1082 - System Information Discovery	T1105 - Remote File Copy
T1071 - Application Layer Protocol	T1112 - Modify Registry
T1064 - Scripting	T1115 - Clipboard Data
T1056 - Input Capture	

CVE:

CVE-2017-11882

Subexsecure Protection

- Subexsecure detects the malware as 'SS_Gen_LodaRAT_PE_A'.

OUR HONEYPOT NETWORK

This report has been prepared from threat intelligence gathered by our honeypot network that is today operational in 62 cities across the world. These cities have at least one of these attributes:

- Are landing centers for submarine cables
- Are internet traffic hotspots
- House multiple IoT projects with a high number of connected endpoints
- House multiple connected critical infrastructure projects
- Have academic and research centers focusing on IoT
- Have the potential to host multiple IoT projects across domains in the future

There are more than 3.5 million attacks registered in a day across this network of individual Honeypot are studied, analyzed, categorized and marked according to a threat rank index, there is a priority assessment framework that we have developed within Subex. The network includes over 4000 physical and virtual devices covering over 400 device architectures and varied connectivity flavors globally. Devices are grouped based on the sectors they belong to for purposes of understanding sectoral attacks. Thus, a layered flow of threat intelligence is made possible.