Project 4 Malware Analyst Report

Group 1

12/15/2023

This report contains sensitive information (privilege or priority information, customer PII, Etc) Disclosing, copying, distributing or taking any action in reliance on the contents of this information is strictly prohibited without prior approval could cause serious harm.
In addition, due to the nature of material being reviewed, potentially offensive material may be present in this report.

Table of Contents

Executive Summary

Type: Trojan

Targeted Platforms: Windows 7, 8, 10

Disguise: Masquerades as legitimate Dependency Walker software

Persistence: Utilizes Busybox for establishing long-term presence

Behavior: Dropped 2 execute files. Operates in stealth mode and deploys Command and Control (C2)

The identified malware has been classified as a trojan, with the initial executable file named depends.exe serving as a dropper responsible for deploying two distinct executable files. One of these files, depends_.exe, is verified as a legitimate component, while the second file, named httpd.exe, is designed to establish persistence on the compromised system. Upon execution of depends.exe, a netsh command is employed to create a firewall configuration, facilitating communication with the malicious server for httpd.exe.

It should be noted that with each instance of depends.exe execution, a new iteration labeled depends__.exe is generated. However, it is noteworthy that the persistence component, httpd.exe, remains singular. This persistent executable is programmed to collect and transmit victim data to the designated malware server upon activation.

The orchestration of this malware underscores a deliberate and methodical approach, utilizing a multi-stage deployment strategy for maintaining a persistent presence on the compromised system, coupled with the exfiltration of sensitive data to the external server.

YARA rules can be found under the YARA section of the report. These rules identify the malware components in this project.

Malware Details

Date	**12/14/2023**
Analyst	**Kiet Hoang** **Breighton Kohl** **Christian Lara** **Tung Nguyen** **Donal Novasky**

depends.exe information

File name	**depends.exe**
File size	**1242726**
File type	**.exe (execute)**
MD5	**9ffd63ce0c331b651386063ce2e541e3**
SHA1	**77a457c61002cd88b07c69555911ebac6766f2bf**
SHA256	**c872c4a53f1d920f969aeaa74418e13206838ddfc79e5017808f387ccb0b7714**
Packer / compiler info	**MinGW**
Compile time	**Thu Nov 30 08:08:53 2023 | UTC**

depends_____.exe infomation

File name	**depends____.exe**
File size	**566272**
File type	**.exe (execute)**
MD5	**fc9015fc4596d90bfe0547ab96cb21b3**
SHA1	**51eb19aba108c41b6febfb6d99133354a2a439fd**
SHA256	**57c483dc985a9757501993e969c2a7043c26517f97fd49a42b33d2d6a4193d8b**
Packer / compiler info	**Visual Studio 2005**
Compile time	**Mon Oct 30 06:49:56 2006 | UTC**

httpd.exe information

File name	**httpd.exe**
File size	**646656**
File type	**.exe (execute)**
MD5	**ec3a2ed0a6e1b6b67ba91b7f90ed73aa**
SHA1	**e6b39fedaab09315462bccefdca73ba6ee5456f6**
SHA256	**137a346a40c0a0facdfc0f10b47ea52e3d4413db2da1e15d1d2093e8ef7f3acb**
Packer / compiler info	**BusyBoxv**

Compile time	Thu Jan 01 00:00:00 1970| UTC

DEPENDS.EXE

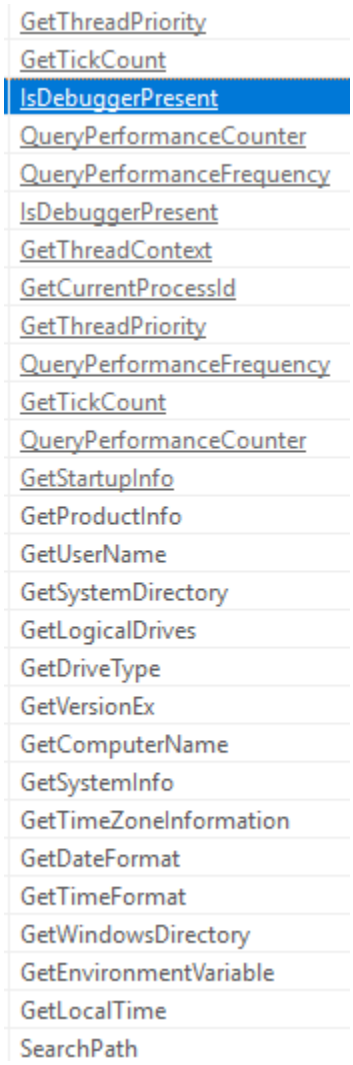
Window API [1]

These functions are employed to modify the memory of another process. Malicious software may utilize them for the following purposes:

- Allocating memory within a target process and subsequently copying the contents of an executable file into that allocated space. This effectively involves "injecting" the file into the memory of the target process.

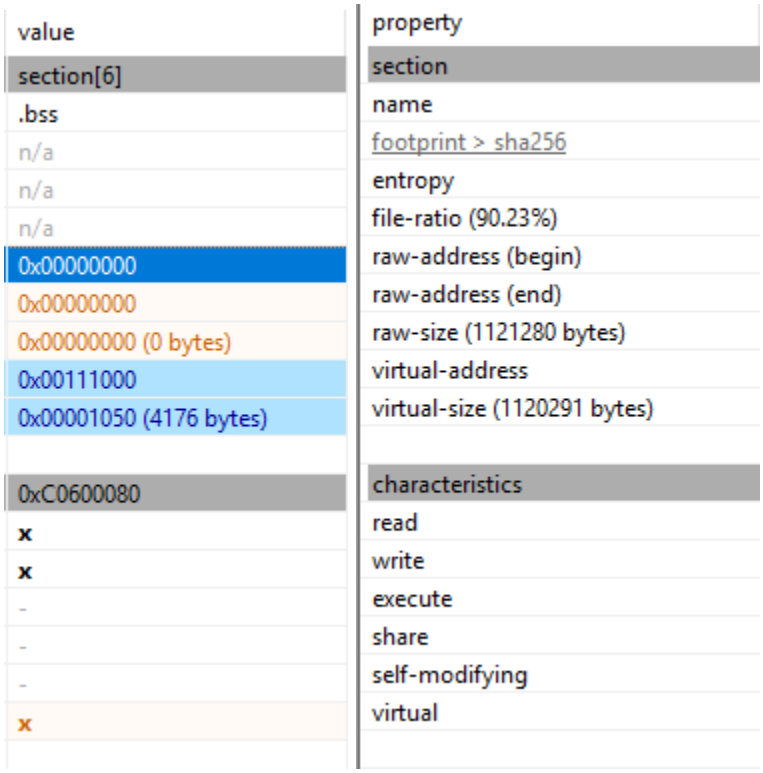
- Initiating a process in a paused or suspended state, removing its existing memory mappings, and substituting them with malicious code. Subsequently, the process is resumed, allowing the malicious code to execute within the context of a genuine and legitimate process.

VirtualAlloc
VirtualFree
VirtualProtect
VirtualQuery
VirtualQuery
VirtualProtect
VirtualAlloc
VirtualFree
VirtualProtect
VirtualAlloc
VirtualFree
VirtualProtect
VirtualQuery
VirtualQuery
VirtualProtect
VirtualAlloc
VirtualFree



- The malware employs the "IsDebuggerPresent" API to ascertain the presence of a debugger, alongside a suite of other APIs aimed at uncovering various details about the computer. These include "GetStartupInfo," "GetUserName," "GetProductInfo," "GetTimeZoneInformation," "GetWindowsDirectory," and "GetLocalTime," among others. This comprehensive approach indicates a systematic effort by the malware to gather extensive information about the victim's computer.

Contain ".bss"

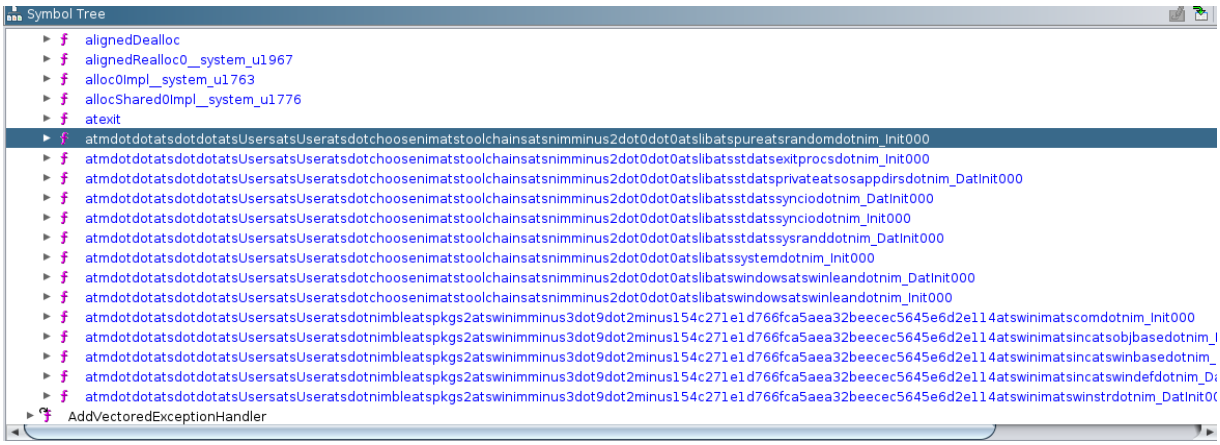


value	property
section[6]	section
.bss	name
n/a	footprint > sha256
n/a	entropy
n/a	file-ratio (90.23%)
0x00000000	raw-address (begin)
0x00000000	raw-address (end)
0x00000000 (0 bytes)	raw-size (1121280 bytes)
0x00111000	virtual-address
0x00001050 (4176 bytes)	virtual-size (1120291 bytes)
0xC0600080	characteristics
x	read
x	write
-	execute
-	share
-	self-modifying
x	virtual

- The "Depend.exe" executable may utilize the ".bss" section for tasks such as process injection, where it prepares code for injection in a zero-initialized space, and for storing obfuscated information, evading detection due to the non-signature-matching nature of the data and lesser scrutiny compared to other sections like ".text" or ".data".

Long and repeated name functions

The presence of excessively lengthy function names within the 'depends.exe' executable may serve as an indication of potentially nefarious activities or efforts at code obfuscation.



Ghidra

Window API [2]

Ghidra

- The “**WriteProcessMemory**” function is integral to modifying the memory of a target process and is often leveraged in malicious exploits to alter a process's functionality.

- The “**CreateRemoteThread**” function, known for initiating a thread in another process's space, raises concerns due to its association with code injection techniques, which are

hallmarks of malware operations, particularly in the creation of Remote Access Trojans (RATs).

Strings

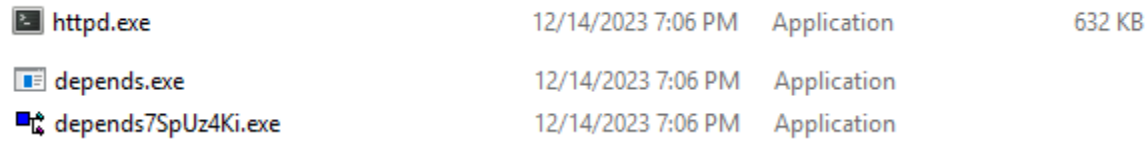
```
140033b87 40 53 68 65 6c 6c 45 78 65 ...    ds    "@ShellExecute"

14005c23f 40 68 74 74 70 64 2e 65 78 ...    ds    "@httpd.exe"

14007cee7 40 64 65 70 65 6e 64 73 2e ...    ds    "@depends.exe"
```

Ghidra

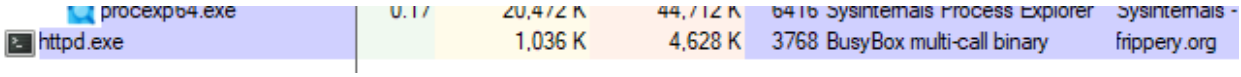
- The executable in question, designated as "depends.exe", appears to be engaging in behavior consistent with the execution of a shell command. Based upon the accumulated evidence, it is posited that this malware attempts to execute a shell process and consequently deploys two executable files. The first bears the identical name "depends.exe", while the second is identified as "httpd.exe". The latter may be implicated in the establishment of a Command and Control (C2) infrastructure, or it may function as a component of a Remote Access Trojan (RAT),

4688	WriteFile	C:\Users\IEUser\AppData\Local\Temp\depends7SpUz4Ki.exe
4688	WriteFile	C:\Users\IEUser\AppData\Local\Temp\depends7SpUz4Ki.exe
4688	WriteFile	C:\Users\IEUser\AppData\Local\Temp\depends7SpUz4Ki.exe
4688	WriteFile	C:\Users\IEUser\AppData\Local\Temp\depends7SpUz4Ki.exe
4688	CloseFile	C:\Users\IEUser\AppData\Local\Temp\depends7SpUz4Ki.exe
4688	CreateFile	C:\Users\IEUser\AppData\Local\Temp\depends.exe
4688	WriteFile	C:\Users\IEUser\AppData\Local\Temp\depends.exe
4688	WriteFile	C:\Users\IEUser\AppData\Local\Temp\depends.exe
4688	WriteFile	C:\Users\IEUser\AppData\Local\Temp\depends.exe
4688	WriteFile	C:\Users\IEUser\AppData\Local\Temp\depends.exe

Process Monitor



C:\User\IEUser\AppData\Temp



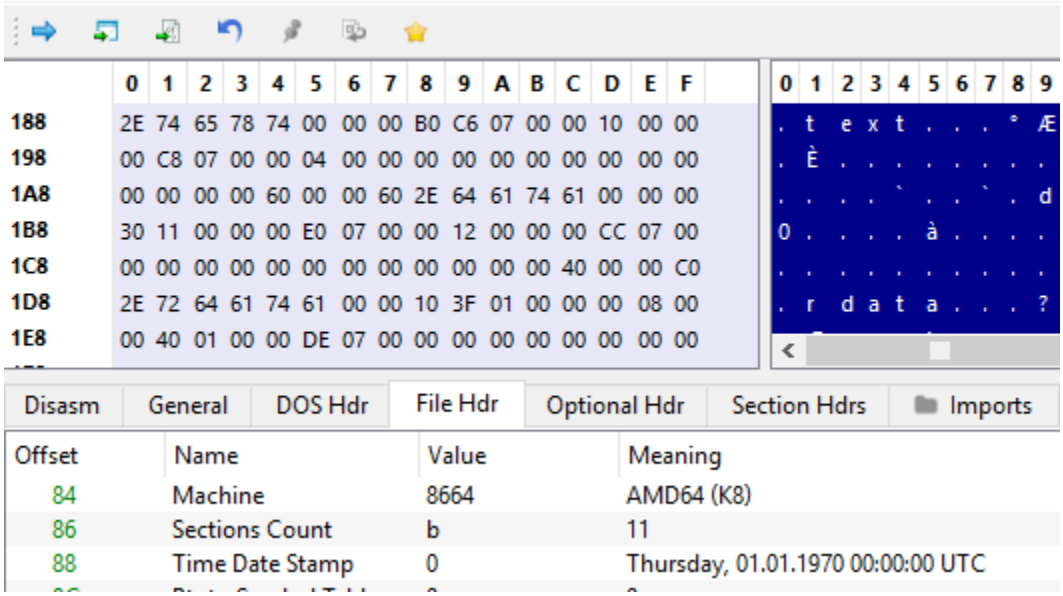
Process Explorer

- Upon initiation, the "depends.exe" file executes a routine that results in the placement of three executable files within the "C:\Users\IEUser\AppData\Temp" directory. Notably, the tool Process Monitor fails to register the creation of "httpd.exe" within this directory, suggesting the possibility that it may have been deployed through a shell execution command. Furthermore, the operation of "httpd.exe" proceeds covertly, evading detection, with its activity only discernible through the use of Process Explorer.

HTTPD.EXE

Timestamp

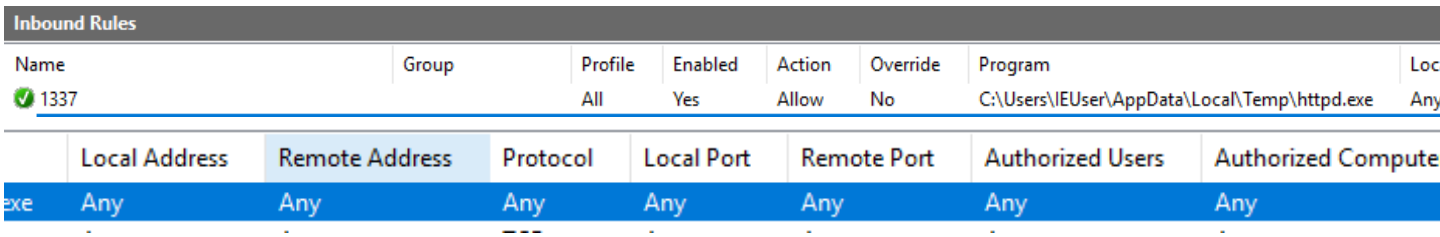
\Temp\httpd.exe]



PEBear

- The timestamp affixed to the header of "httpd.exe" is conspicuously archaic, recorded as Thursday, January 1st, 00:00:00 UTC. This anachronistic timestamp may be a deliberate attempt by the malware to mislead both users and automated security systems into regarding the file as an established, and thus ostensibly innocuous, component of the system's infrastructure.

Firewall Rule



Window Firewall

- Upon execution, the executable "Depends.exe" proceeds to establish a persistent element, "httpd.exe," on the system. Concurrently, it executes a shell command utilizing

netsh to create an inbound firewall rule. This rule, identified as "1337," is configured to authorize "httpd.exe" to operate across any port. Additionally, the configuration of this rule permits its utilization by any user account on the system.

Busybox and RAT

```
LAB_1400137b4:
    pcVar21 = "busybox --install [-s] [-u|DIR]";
    uVar6 = FUN_140010410("busybox --install [-s] [-u|DIR]",ppDVar14,param_3,param_4);
  }
  else {
    ppDVar14 = (DWORD **)(ppbVar24 + DAT_14007e06c);
    unaff_RSI = *ppDVar14;
    uVar6 = uVar5 & 2;
    if (unaff_RSI != (DWORD *)0x0) {
      if ((uVar6 == 0) && (ppDVar14[1] == (DWORD *)0x0)) goto LAB_1400137dc;
      goto LAB_1400137b4;
    }
  }
```

Ghidra

- Upon examination in Ghidra, it has been observed that the httpd.exe executable is utilizing BusyBox utilities. The use of BusyBox in this context may raise concerns regarding the potential for unauthorized activities. This concern is further substantiated by the detection of BusyBox establishing a secure SSL/TLS connection with a remote server. Notably, such connections facilitate encrypted communication between the web server and the client's web browser, remaining transparent to the user.

```
Decompile: UndefinedFunction_1400718c0 - (httpd.exe)
191     pFVar13 = (FILE *)FUN_140076150((short *)(_Memory + 1));
192     pFVar14 = (FILE *)(*(code *)PTR_FUN_14007e0c0)(2);
193     pcVar22 = "Connecting to %s (%s)\n";
194     pcVar24 = (char *)pFVar19;
195     pcVar27 = (char *)pFVar13;
196     FUN_1400729b0(pFVar14,"Connecting to %s (%s)\n",pFVar19,pFVar13);
197     free(pFVar13);
198   }
199 LAB_140071d18:
200   pFVar13 = pFStack_60;
201   *(undefined2 *)((longlong)plVar28 + 100) = 0;
202   if ((bVar3) || (*(char *)&pFStack_60->_ptr != 'f')) {
203     pFVar14 = (FILE *)FUN_140047c10(_Memory,pcVar22,pcVar24,pcVar27);
204     if (pFStack_90 == (FILE *)"https") {
205       uVar7 = _fileno(pFVar14);
206       FUN_140065090((char *)pFVar19,(char *)(ulonglong)uVar7,0,pcVar27);
207     }
208     if (bVar3) {
209       pFVar25 = pFVar13;
210       pcVar27 = (char *)pFStack_58;
211       FUN_1400729b0(pFVar14,"GET %s://%s/%s HTTP/1.1\r\n",pFVar13,pFStack_58);
212     }
224     }
225     if ((*(byte *)(plVar28 + 6) & 2) == 0) {
226       pFVar25 = (FILE *)plVar28[10];
227       FUN_1400729b0(pFVar14,"User-Agent: %s\r\n",pFVar25,pcVar27);
228     }
229     pcVar22 = "Connection: close\r\n";
230     FUN_1400729b0(pFVar14,"Connection: close\r\n",pFVar25,pcVar27);
231     if ((pFStack_68 != (FILE *)0x0) && ((*(byte *)(plVar28 + 6) & 0x10) == 0)) {
232       puVar15 = FUN_14000f130((byte *)pFStack_68);
233       pcVar22 = "Authorization: Basic %s\r\n";
234       FUN_1400729b0(pFVar14,"Authorization: Basic %s\r\n",puVar15,pcVar27);
235     }
236     if (((bVar3) && (pbStack_98 != (byte *)0x0)) && ((*(byte *)(plVar28 + 6) & 0x20) == 0)
237       puVar15 = FUN_14000f130(pbStack_98);
238       pcVar22 = "Proxy-Authorization: Basic %s\r\n";
239       FUN_1400729b0(pFVar14,"Proxy-Authorization: Basic %s\r\n",puVar15,pcVar27);
240     }
```

Ghidra

- Corroborating our initial findings, we successfully identified the function tasked with establishing a server connection to handle incoming requests, labeled as “UndefinedFunction_1400718c0.” Within this function, there are discernible references to host details, User-Agent strings, and Proxy-Authorization parameters. This evidence strongly suggests that the malware is configured to set up a connection to a remote

server, likely for the purposes of establishing a Remote Access Trojan (RAT) operation, thereby enabling unauthorized access and control over the infected system.

httpd.exe	6784	TCP	Listen	0.0.0.0	80
System	4	TCP	Listen	0.0.0.0	445
System	4	TCP	Listen	0.0.0.0	5985
svchost.exe	2372	TCP	Listen	0.0.0.0	7680
System	4	TCP	Listen	0.0.0.0	47001
httpd.exe	6784	TCPv6	Listen	::	80

TCPView

- Upon preliminary examination, it has been observed that upon execution, the malware initiates a web server that binds to port 80 on the host system. This action suggests that the malware possesses the functionality to act as a rudimentary web server, utilizing port 80, which is traditionally reserved for HTTP traffic. Notwithstanding the activation of this local server, there is no evidence of subsequent external network communications.

- Httpd appears to be a legitimate program. The executable is a daemon , this is when it runs on a web server to handle incoming HTTP requests from clients, such as web browsers, and serves web content in response

Data Collection

```
00007FFC0E54F6B6   48:8B03          mov rax,qword ptr ds:[rbx]
00007FFC0E54F6B9   48:85C0          test rax,rax
00007FFC0E54F6BC   74 3B            je msvcrt.7FFC0E54F6F9
00007FFC0E54F6BE   48:83C9 FF       or rcx,FFFFFFFFFFFFFFFF
00007FFC0E54F6C2   48:FFC1          inc rcx
00007FFC0E54F6C5   803C08 00        cmp byte ptr ds:[rax+rcx],0
00007FFC0E54F6C9   75 F7            jne msvcrt.7FFC0E54F6C2
00007FFC0E54F6CB   48:3BCF          cmp rcx,rdi
00007FFC0E54F6CE   76 18            jbe msvcrt.7FFC0E54F6E8
00007FFC0E54F6D0   803C38 3D        cmp byte ptr ds:[rax+rdi],3D
00007FFC0E54F6D4   75 12            jne msvcrt.7FFC0E54F6E8
00007FFC0E54F6D6   4C:8BC7          mov r8,rdi
00007FFC0E54F6D9   48:8BD6          mov rdx,rsi
00007FFC0E54F6DC   48:8BC8          mov rcx,rax
00007FFC0E54F6DF   E8 4C57FFFF      call <msvcrt._mbsnbicoll>
00007FFC0E54F6E4   85C0             test eax,eax
00007FFC0E54F6E6   74 06            je msvcrt.7FFC0E54F6EE
00007FFC0E54F6E8   48:83C3 08       add rbx,8
00007FFC0E54F6EC   EB C8            jmp msvcrt.7FFC0E54F6B6
```

x64dbg

- Upon execution of "httpd.exe" within the x64dbg environment, this particular assembly code block is observed to be operational. It is postulated that this block constitutes a segment of a function designed to retrieve an array of data from the target computer. This is inferred from the function's traversal across critical file paths and configurations, potentially leveraging environment variables. Such activity appears to be integral to the data acquisition process, specifically focusing on the C drive.

Mitre ATT&CK

Tatic	ID	Technique	Procedure
Execution	T1106	Native API	Imports suspicious APIs
Execution	T1059.003	Windows Command Shell	Add Firewall rule via command shell
Execution	T1129	Shared Modules	PE header
Privilege Escalation	T1055.003	Thread Execution Hijacking	Creates a thread in a remote process
Defense Evasion	T1027.002	Software Packing	“.bss” zero size
Defense Evasion,Discovery	T1622	Debugger Evasion	Contains ability to check debugger is running
Command and Control	T1105	Ingress Tool Transfer	Drops executable files
Discovery	T1083	File and Directory Discovery	Read many critical files in the victim’s computer

Regshot

Keys added:

HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason

HKLM\SOFTWARE\Policies\Microsoft\Windows\IPSec\Policy\Local

HKLM\SOFTWARE\WOW6432Node\Policies\Microsoft\Windows\IPSec\Policy\Local

HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\SessionInfo\1\ApplicationViewManagement\W32:0000000000050526

HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\SessionInfo\1\ApplicationViewManagement\W32:0000000000160586

HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Dependency Walker

HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Dependency Walker\External Viewer

HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Dependency Walker\Settings

HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\System\CurrentControlSet\Control\NetTrace

HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\System\CurrentControlSet\Control\NetTrace\Session

Values deleted: 1

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\BITS\PerfMMFileName: "Global\MMF_BITS81c0f284-ed18-4fa1-8f93-6e16ce341481"

Values added: 15 (trimmed for brevity)

HKLM\SYSTEM\ControlSet001\Services\SharedAccess\Parameters\FirewallPolicy\FirewallRules\{AB47BA67-73D1-4B99-B1DC-642FCF805A81}: "v2.29|Action=Allow|Active=TRUE|Dir=In|App=C:\Users\IEUser\AppData\Local\Temp\httpd.exe|Name=1337|"

Values modified: 39 (trimmed for brevity)

HKLM\SYSTEM\ControlSet001\Services\Tcpip\Parameters\Interfaces\{4aa86136-917b-45d2-be98-087b589b8ca0}\LeaseObtainedTime: 0x00000011

- Further Tcpip value modifications follow entailing lease obtainment and termination, suggesting a briefly established connection interface

Files added: 5

C:\Windows\Prefetch\DEPENDS.EXE-2DB612B4.pf

C:\Windows\Prefetch\DEPENDS.EXE-7D286406.pf

C:\Windows\Prefetch\DEPENDSJ6JEW2YH.EXE-2F02BE81.pf

C:\Windows\Prefetch\HTTPD.EXE-1B19E1A9.pf - Windows Apache HTTP Server file

C:\Windows\Prefetch\NETSH.EXE-3DD790C5.pf - Microsoft network shell file for allowing local or remote network device configuration

Analysis

Creation of firewall rule named “1337” towards application httpd.exe and allowing all connections over it is highly suspect, as an enabled means of connection into the host. Various other connection interfaces and protocols including IPSec, NetTrace, and Netsh.exe are modified or established.

Yara

A. Project 4 Yara rules to identify malware components

```
import "hash"
import "pe"

rule proj4 {

    strings:
     $exe_0x = { 4D 5A } // Standard exe initializer
     $httpd_string = "@httpd.exe" //Suspect executable call
     $1337_0x = { 31 33 33 37 } //Firewall rule catch
     $shell_execute = "ProfileSimulateShellExecute" //Not present in legit version (checked with HxD)

    condition:
     $exe_0x at 0 and
     hash.md5(0, filesize) == "9ffd63ce0c331b651386063ce2e541e3" and //MD5 File hash
     $httpd_string and
     $1337_0x and
     $shell_execute and
     pe.timestamp != 1162190996 // Compiler timestamp of the official program (same version)


}

// netsh advfirewall firewall add rule name=1337 dir=in action=allow program=\"C:\\Users\\IEUser\\AppData\\Local\\Temp\\httpd.exe\" enable=yes
```