

## **Kaizen Solutions Software**

Date: September 04<sup>th</sup>, 2018.

### Description

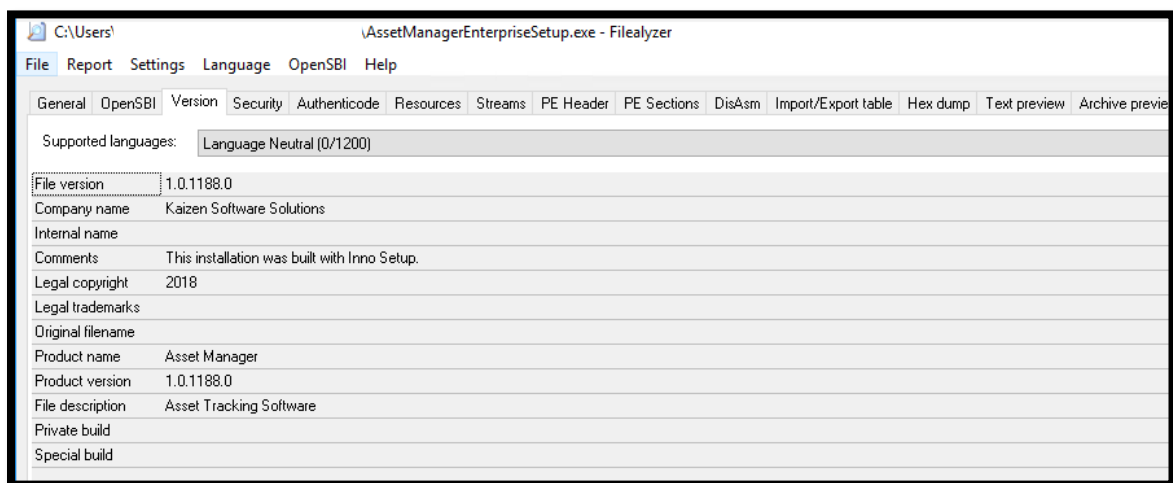
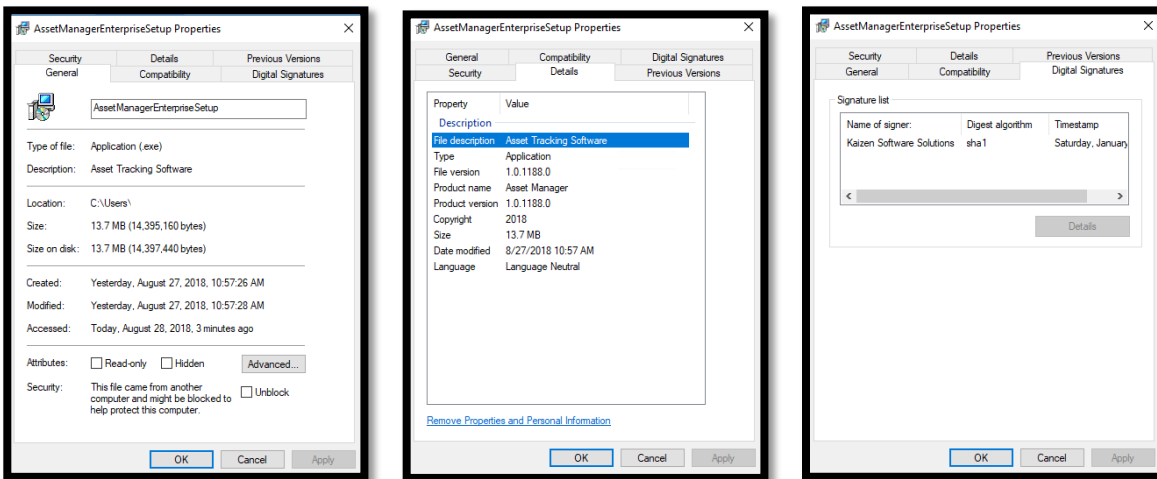
Two solutions were verified in this assessment: 1) the Asset Manager (Enterprise Edition) and 2) the Training Manager (Enterprise Edition). The first is used to control and monitor fixed assets such as computers and equipment. The second is used to track employee training records.

Both software presented some vulnerabilities including remote access and persistence. The main causes of those vulnerabilities were related to specific files such as: “isxdl.dll”, “996E.temp”, “profinder.vshost.exe”, “AssetManager.exe” and “TrainingManager.exe”.

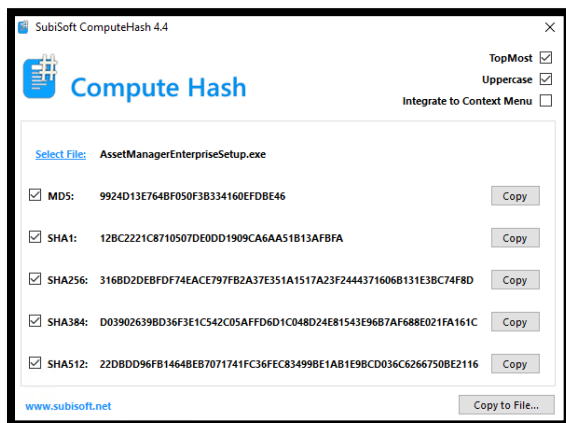
Currently, there are several documented techniques that could be used to exploit those vulnerabilities.

## File Analysis

This is the Asset Manager software:

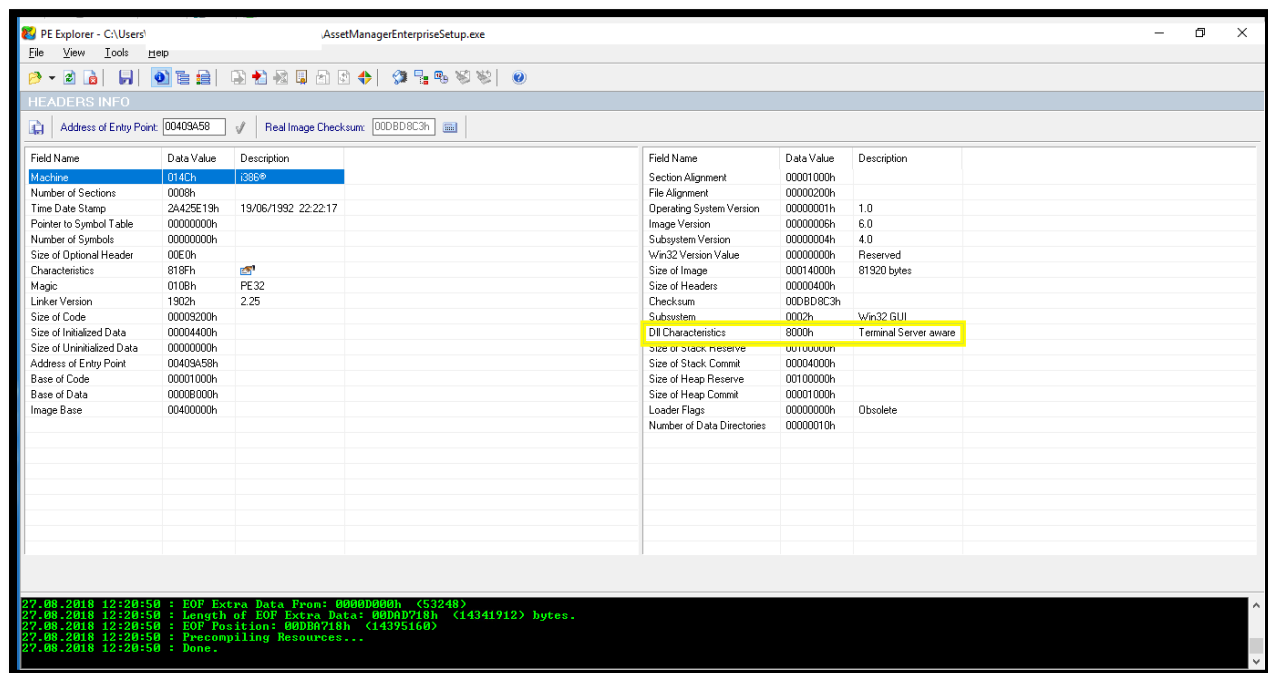


Calculate the Hash:



## Red Flags

We can identify a Terminal Server aware feature:



Terminal Server can create a virtual Windows folder instead of using the system's Windows directory. This gives users access to their own INI files. If an application is Terminal Server aware, it must neither rely on INI files nor write to the HKEY\_CURRENT\_USER registry during setup. In general this characteristics is not valid for drivers, VxDs, or DLLs<sup>1</sup>.

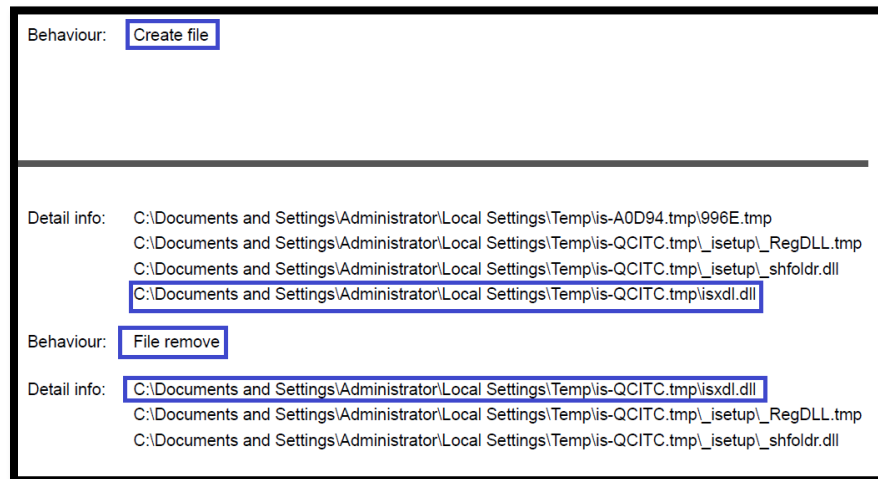
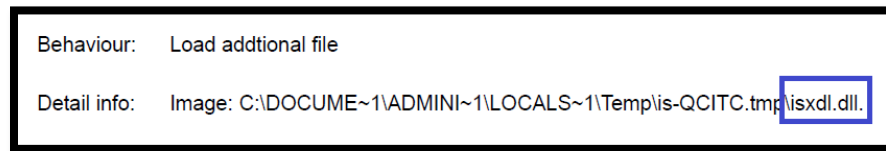
## Persistence

The persistence can be associated with several executable files.

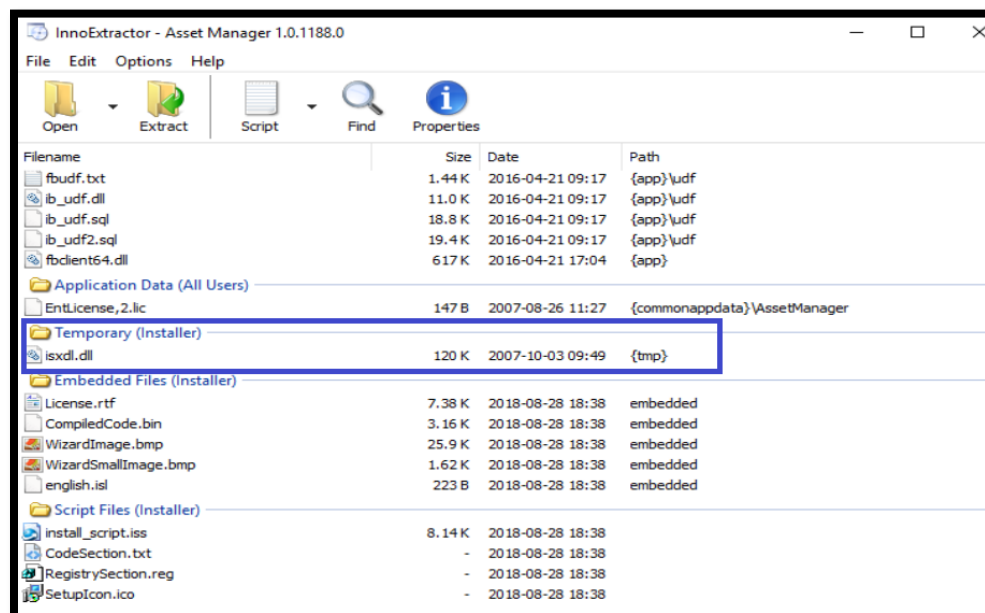
"isxdl.dll" has type "PE32 executable (DLL) (GUI) Intel 80386 for MS Windows"  
"AssetManagerEnterpriseSetup.tmp" has type "PE32 executable (GUI) Intel 80386 for MS Windows"  
"is-SES94.tmp" has type "PE32 executable (GUI) Intel 80386 Mono/.Net assembly for MS Windows"  
"is-LMUDV.tmp" has type "PE32 executable (DLL) (GUI) Intel 80386 for MS Windows"  
"is-H85A9.tmp" has type "PE32 executable (DLL) (console) Intel 80386 Mono/.Net assembly for MS Windows"  
"is-FAIAS.tmp" has type "PE32 executable (console) Intel 80386 for MS Windows"  
"is-CUAEV.tmp" has type "PE32 executable (DLL) (GUI) Intel 80386 for MS Windows"  
"is-ODNS2.tmp" has type "PE32 executable (DLL) (GUI) Intel 80386 for MS Windows"

<sup>1</sup><https://docs.microsoft.com/en-us/cpp/build/reference/tsaware-create-terminal-server-aware-application>

In specific, the file isxdl.dll<sup>2</sup> has the property to create and remove itself.

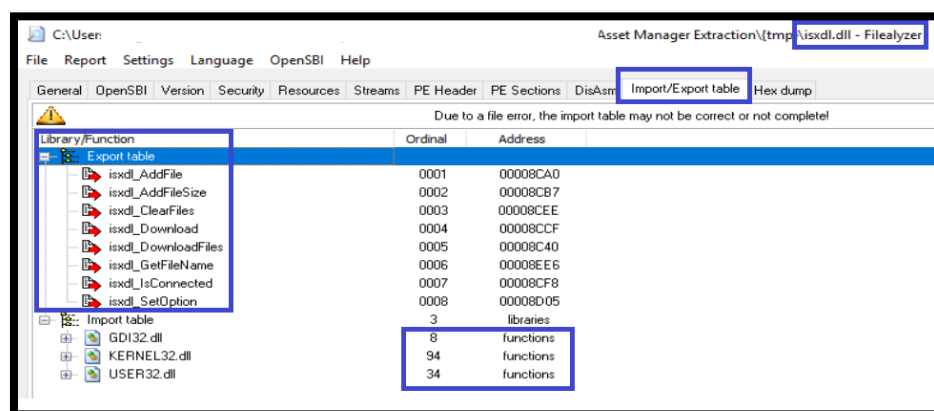


The software has some compressed data. A more detailed information was found after an extraction in the file:

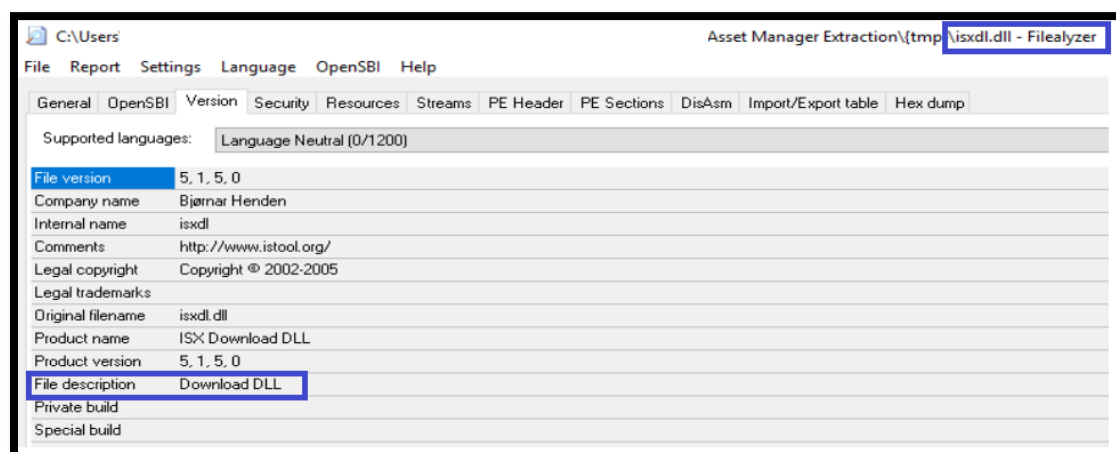


<sup>2</sup><https://www.virustotal.com/#/file/316bd2debfdf74eace797fb2a37e351a1517a23f2444371606b131e3bc74f8dc/behavior>

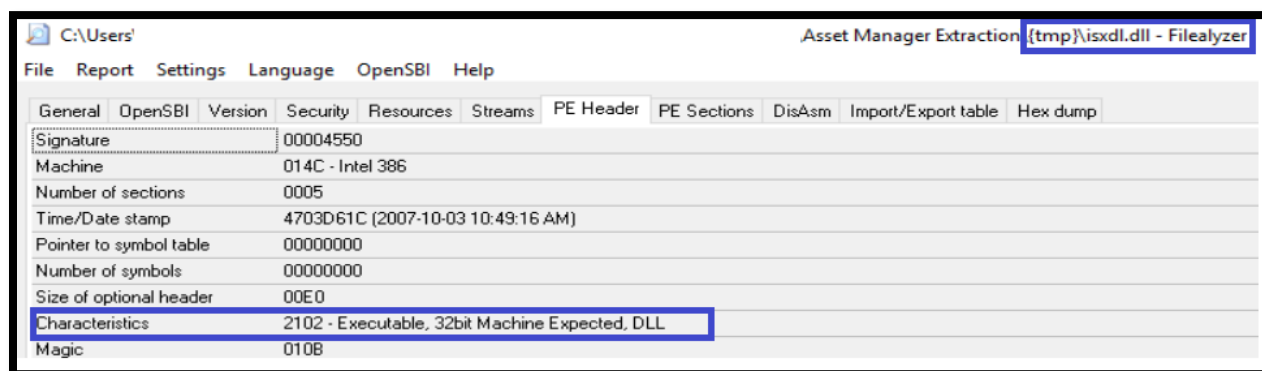
This is the attributes of the file, once the software is installed:



Note the capacity to download other dll:



It is possible to confirm the initial characteristics:



### Suspicious File

The file 996E.tmp seems like a temporary file, but it is in fact an executable file, and possibly not a good one<sup>3</sup>.

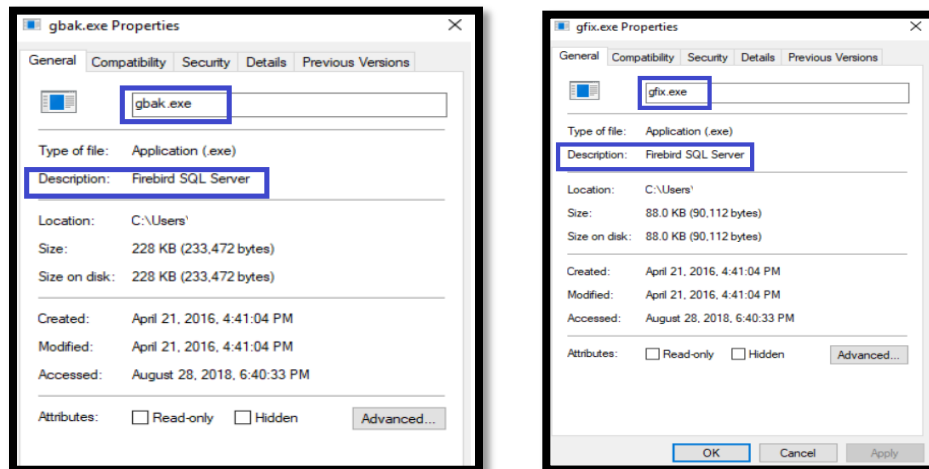
Behaviour: Create executable file

Detail info: C:\Documents and Settings\Administrator\Local Settings\Temp\is-A0D94.tmp\996E.tmp  
 C:\Documents and Settings\Administrator\Local Settings\Temp\is-QCITC.tmp\\_isetup\\_RegDLL.tmp  
 C:\Documents and Settings\Administrator\Local Settings\Temp\is-QCITC.tmp\\_isetup\\_shfoldr.dll  
 C:\Documents and Settings\Administrator\Local Settings\Temp\is-QCITC.tmp\isxdl.dll

Executable files found after the extraction:

AssetManager.exe	2018-01-27 2:12 PM	Application
AssetManager.vshost.exe	2005-09-23 5:56 AM	Application
gbak.exe	2016-04-21 4:41 PM	Application
gfix.exe	2016-04-21 4:41 PM	Application
hh.exe	2009-10-03 1:47 PM	Application

<sup>3</sup> <http://www.exefilesupport.com/easy-guide-to-remove-996e-exe-from-pc>



The “gfix.exe” and “gbak.exe” offers SQL capability and the AssetManager.exe was considered malicious<sup>4</sup>:



## Pointing Directories

Since the software can point resources in to itself through the resource table, it is possible to identify two files with portable executable characteristics.

The image is a screenshot of the PE Explorer application showing the 'SECTION HEADERS' for the file 'AssetManagerEnterpriseSetup.exe'. The table below lists the sections and their properties:

Name	Virtual Size	Virtual Address	Size of Raw Data	Pointer to Raw Data	Characteristics	Pointing Directories
CODE	00009174h	00401000h	00009200h	00000400h	60000020h	
DATA	0000024Ch	0040B000h	00000400h	00009600h	C0000040h	
BSS	00000E48h	0040C000h	00000000h	00009A00h	C0000000h	
.idata	00000950h	0040D000h	00000A00h	00009A00h	C0000040h	Import Table
.tls	00000008h	0040E000h	00000000h	0000A400h	C0000000h	
.rdata	00000018h	0040F000h	00000200h	0000A400h	50000040h	TLS Table
.reloc	000008B4h	00410000h	00000000h	0000A600h	50000040h	
.rsrc	00002A00h	00411000h	00002A00h	0000A600h	50000040h	Resource Table

<sup>4</sup> <https://www.hybrid-analysis.com/sample/b64187b4dc9b0ae5cd558a400af82675c93dcd8ea148c6785c668ebd112c0343/5b85d8fe7ca3e16eee36b374>

Behaviour: Find resource in self with type of PE

Detail info: (FindResourceA) hModule = 0x00400000, ResName: REGDLL\_EXE, ResType:  
(FindResourceA) hModule = 0x00400000, ResName: SHFOLDERDLL, ResType:

It is possible to note also the use of a TLS table<sup>5</sup>. Which can be used by the executable code to get the address of the TLS data area for the given program and module.

## Suspicious Behaviours

### Admin privilege

A tentative to identify administrator details.

Behaviour: Find file

Detail info: FileName = C:\DOCUME~1  
FileName = C:\DOCUME~1\ADMINI~1  
FileName = C:\DOCUME~1\ADMINI~1\LOCALS~1  
FileName = C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp  
FileName = C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\is-A0D94.tmp  
FileName = C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\is-A0D94.tmp\996E.tmp  
FileName = C:\Documents and Settings  
FileName = C:\Documents and Settings\Administrator  
FileName = C:\Documents and Settings\Administrator\「开始」菜单  
FileName = C:\Documents and Settings\Administrator\「开始」菜单\程序  
FileName = C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\is-QCITC.tmp\  
FileName = C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\is-QCITC.tmp\\_isetup\\*

Which can be useful if someone need to install new drivers:

Behaviour: Modify process token privilege

Detail info: SE\_LOAD\_DRIVER\_PRIVILEGE

<sup>5</sup> <https://docs.microsoft.com/en-us/windows/desktop/Debug/pe-format#the-tls-section>



The software can create mutex functions to manage the shared resource from simultaneous access by multiple threads or processes<sup>6</sup>:

Behaviour:	Create mutex
Detail info:	CTF.LBES.MutexDefaultS-*
	CTF.Compart.MutexDefaultS-*
	CTF.Asm.MutexDefaultS-*
	CTF.Layouts.MutexDefaultS-*
	CTF.TMD.MutexDefaultS-*
	CTF.TimListCache.FMPDefaultS-*MUTEX.DefaultS-*
	MSCTF.Shared.MUTEX.EBH
	MSCTF.Shared.MUTEX.AFK

It is possible that the shared mutex might be related to data stored in the .rdata, .reloc, .rsrc:

Section	VirtSize	VirtAddr	PhysSize	PhysAddr	Flags	CRC32	MD5	Characteristics
CODE	00009174	00001000	00009200	00000400	60000020	43545943	EA92E1415BC80E2738E334267EBB921	Code, Execute Access, Read Access
DATA	0000024C	0000B000	00000400	00009600	C0000040	6352B82C	F96DA19D2571A42BDF1B9E9BD62EC99	Initialized Data, Read Access, Write Access
BSS	00000E48	0000C000	00000000	00009A00	C0000000			Read Access, Write Access
.idata	00000950	0000D000	00000A00	00009A00	C0000040	AA2C7DA9	B5485BF96B970E5EA01292AF2ACDBA	Initialized Data, Read Access, Write Access
.rdata	00000018	0000F000	00000200	0000A400	50000040	5EC36383	9BA824905BF9C7922B6FC97A38B7436E	Initialized Data, Shared, Read Access
.reloc	000008B4	00010000	00000000	0000A600	50000040			Initialized Data, Shared, Read Access
.rsrc	00002A00	00011000	00002A00	0000A600	50000040	3F4D47F3	7CC2677A179ECCCE0680060F2A7F4DA4	Initialized Data, Shared, Read Access

And the software has anti-debugging functions<sup>7</sup>:

Behaviour:	Open event
Detail info:	HookSwitchHookEnabledEvent
	Global\crypt32LogoffEvent
	_fCanRegisterWithShellService
	CTF.ThreadMIConnectionEvent.00000714.00000000.00000012
	CTF.ThreadMarshalInterfaceEvent.00000714.00000000.00000013
	CTF.ThreadMIConnectionEvent.00000714.00000000.00000013
	MSCTF.SendReceiveConection.Event.EBH.IC
	MSCTF.SendReceive.Event.EBH.IC

<sup>6</sup> <https://docs.microsoft.com/en-us/windows/desktop/sync/using-mutex-objects>

<sup>7</sup> <https://www.veracode.com/blog/2009/02/anti-debugging-series-part-iv>

A possible evasion technique could be in place using a sleep technique:

Behaviour:	Call Sleep function
Detail info:	[1]: MilliSeconds = 50. [2]: MilliSeconds = 50. [3]: MilliSeconds = 50. [4]: MilliSeconds = 50. [5]: MilliSeconds = 50. [6]: MilliSeconds = 50. [7]: MilliSeconds = 50. [8]: MilliSeconds = 50. [9]: MilliSeconds = 50. [10]: MilliSeconds = 50. [2]: MilliSeconds = 250. [3]: MilliSeconds = 250. [4]: MilliSeconds = 250. [5]: MilliSeconds = 250. [6]: MilliSeconds = 250.

## Identified Risks<sup>8</sup>

**AssetManagerEnterpriseSetup.exe** suspicious

This report is generated from a file or URL submitted to this webservice on August 28th 2018 13:58:16 (CEST) and action script **Threat Score: 80/100**  
*Heavy Anti-Evasion* **AV Detection: 2%**

Guest System: Windows 7 32 bit, Home Premium, 6.1 (build 7601), Service Pack 1  
Report generated by Falcon Sandbox v8.10 © Hybrid Analysis

[Link](#) [Twitter](#) [E-Mail](#)

[Overview](#) [Login to Download Sample \(14MiB\)](#) [Downloads](#) [External Reports](#) [Re-analyze](#) [Hash Not Seen Before](#)

[No similar samples](#) [Report Abuse](#)

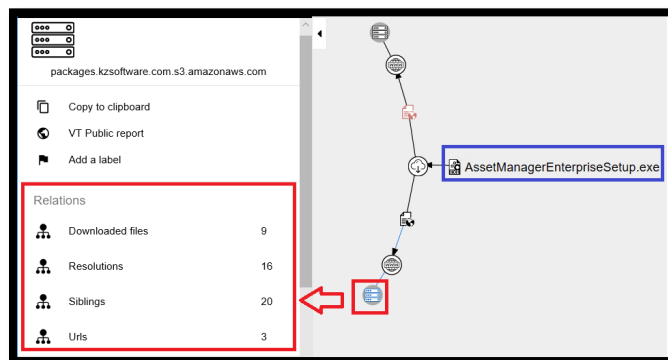
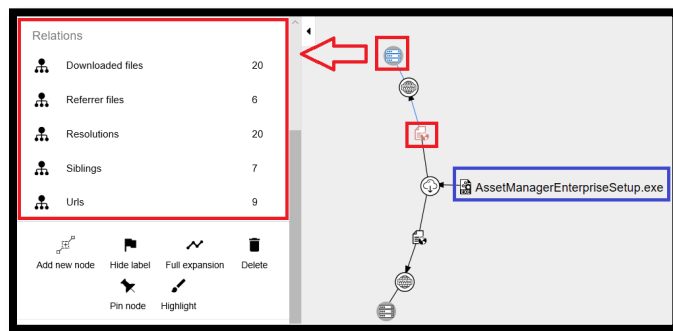
### Incident Response

#### Risk Assessment

<b>Remote Access</b>	Reads terminal service related keys (often RDP related)
<b>Persistence</b>	Writes data to a remote process
<b>Fingerprint</b>	Reads the active computer name Reads the cryptographic machine GUID
<b>Spreading</b>	Opens the MountPointManager (often used to detect additional infection locations)

<sup>8</sup> <https://www.hybrid-analysis.com/sample/316bd2debfd74eace797fb2a37e351a1517a23f2444371606b131e3bc74f8dc?environmentId=100>

## Final Map



## Attacks

The types of attacks can be selected from documented techniques<sup>9</sup>.

MITRE ATT&CK™ Techniques Detection

Minimal

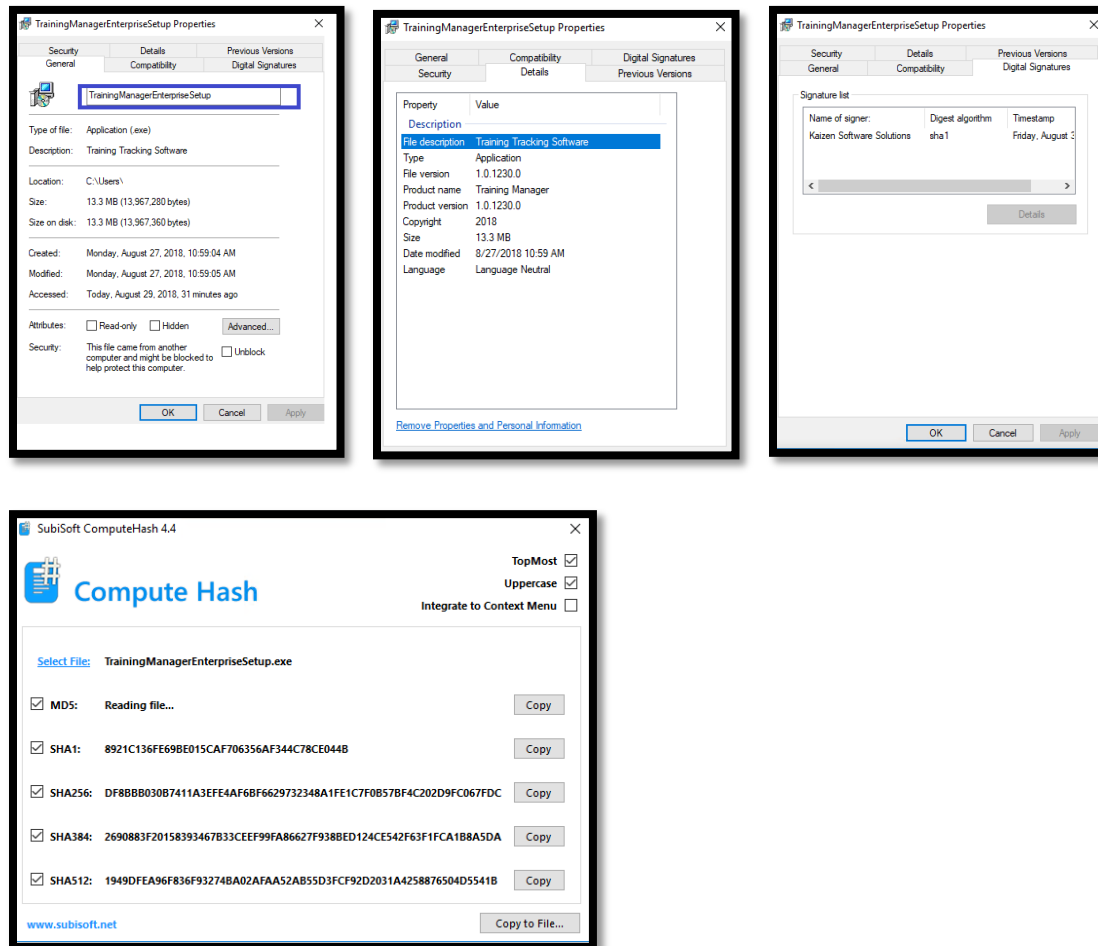
Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Command and Control
		Hooking 1	Hooking 1	Code Signing 1	Hooking 1	Peripheral Device Discovery 2	Remote Desktop Protocol 1	Email Collection 1	Data Compression 1	
		Kernel Modules and Extensions 1	Process Injection 1	File Deletion 2		Process Discovery 1				
				Modify Registry 1		Query Registry 4 1				
				Process Injection 1						

<sup>9</sup><https://www.hybrid-analysis.com/sample/316bd2debdf74eace797fb2a37e351a1517a23f2444371606b131e3bc74f8dc>

## Replicated Approach (Training Manager)

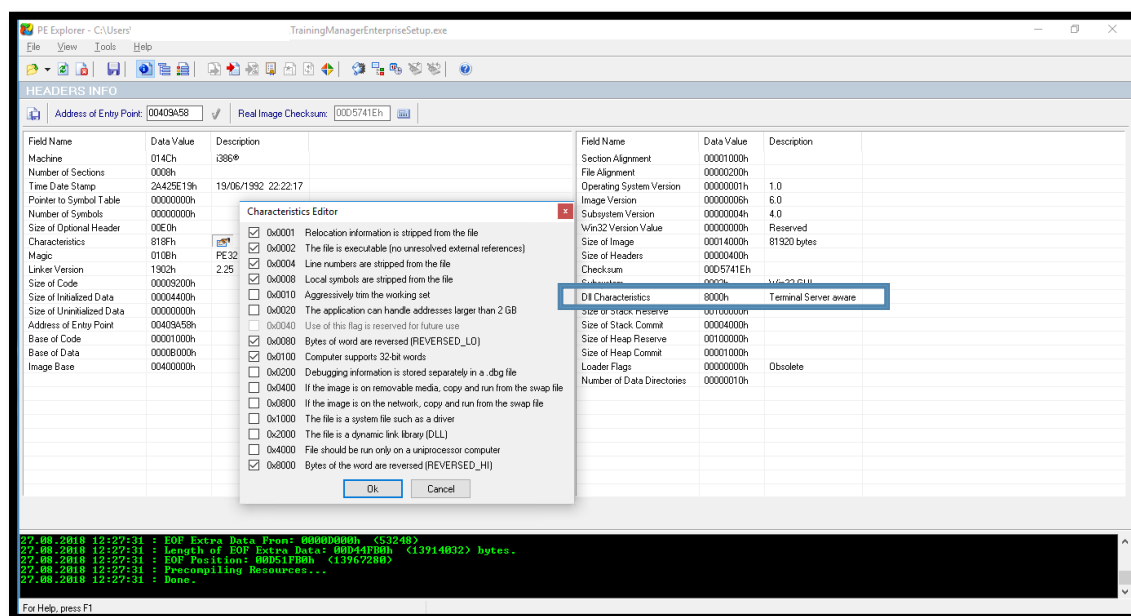
The Training Manager software share similar file structure and similar findings described in the Asset Manager.

### File Analysis

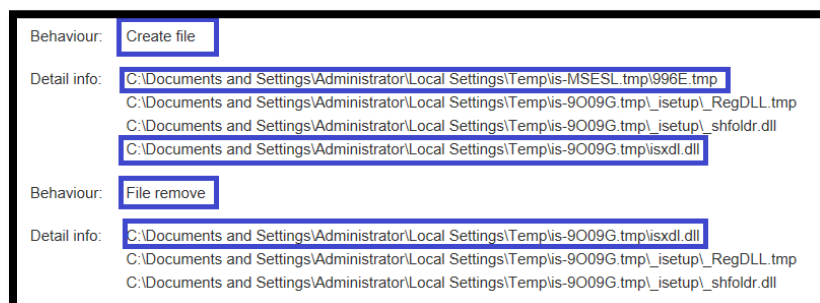
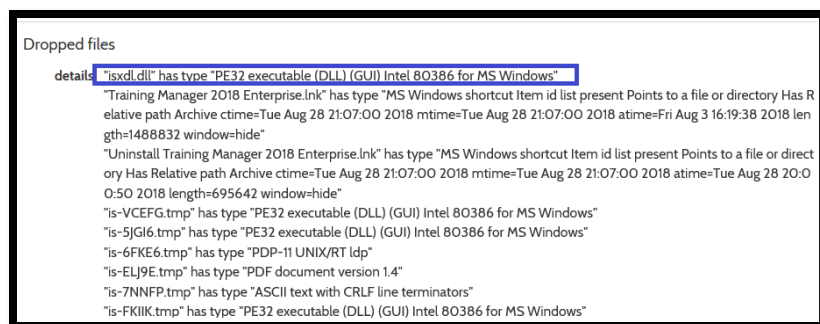


## Red Flags

## Terminal Server



Use the isxdl.dll and the 996E.tmp files<sup>10</sup> previously discussed.



<sup>10</sup><https://www.virustotal.com/#/file/df8bbb030b7411a3efe4af6bf6629732348a1fe1c7f0b57bf4c202d9fc067fdc/behavior>

Behaviour: Load additional file

Detail info: Image: C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\is-9009G.tmp\isx.dll.dll.

#### Process And Service Actions ⓘ

##### Permissions Requested

SE\_LOAD\_DRIVER\_PRIVILEGE

##### Processes Terminated

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\is-MSESL.tmp\996E.tmp  
C:\Documents and Settings\Administrator\Local Settings\Temp\EB93A6\996E.exe  
C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\is-H5JHD.tmp\996E.tmp

As the Asset Manager, the Training Manager also make changes in the Windows Directory

#### Touches files in the Windows directory

details "<Input Sample>" touched file "C:\Windows\AppPatch\sysmain.sdb"  
"AssetManagerEnterpriseSetup.tmp" touched file "C:\Windows\Fonts\StaticCache.dat"  
"AssetManagerEnterpriseSetup.tmp" touched file "C:\Windows\System32\en-US\user32.dll.mui"  
"AssetManagerEnterpriseSetup.tmp" touched file "C:\Windows\Globalization\Sorting\SortDefault.nls"  
"AssetManagerEnterpriseSetup.tmp" touched file "C:\Windows\System32\en-US\msctf.dll.mui"  
"AssetManagerEnterpriseSetup.tmp" touched file "C:\Windows\System32\en-US\KernelBase.dll.mui"  
"AssetManagerEnterpriseSetup.tmp" touched file "C:\Windows\System32\netmsg.dll"  
"AssetManagerEnterpriseSetup.tmp" touched file "C:\Windows\System32\en-US\netmsg.dll.mui"  
"AssetManagerEnterpriseSetup.tmp" touched file "C:\Windows\System32\shfolder.dll"  
"AssetManagerEnterpriseSetup.tmp" touched file "C:\Windows\System32\imageres.dll"  
"AssetManagerEnterpriseSetup.tmp" touched file "C:\Windows\System32\en-US\imageres.dll.mui"  
"AssetManagerEnterpriseSetup.tmp" touched file "%APPDATA%\Microsoft\Windows\Start Menu"  
"AssetManagerEnterpriseSetup.tmp" touched file "C:\Users\%USERNAME%\AppData\Roaming\Microsoft\Windows\Start Menu\Programs"

#### Touches files in the Windows directory

details "<Input Sample>" touched file "C:\Windows\System32\en-US\KernelBase.dll.mui"  
"<Input Sample>" touched file "C:\Windows\System32\netmsg.dll"  
"<Input Sample>" touched file "C:\Windows\System32\en-US\netmsg.dll.mui"  
"<Input Sample>" touched file "C:\Windows\Globalization\Sorting\SortDefault.nls"  
"<Input Sample>" touched file "C:\Windows\AppPatch\sysmain.sdb"  
"TrainingManagerEnterpriseSetup.tmp" touched file "C:\Windows\Fonts\StaticCache.dat"  
"TrainingManagerEnterpriseSetup.tmp" touched file "C:\Windows\System32\en-US\user32.dll.mui"  
"TrainingManagerEnterpriseSetup.tmp" touched file "C:\Windows\Globalization\Sorting\SortDefault.nls"  
"TrainingManagerEnterpriseSetup.tmp" touched file "C:\Windows\System32\en-US\msctf.dll.mui"  
"TrainingManagerEnterpriseSetup.tmp" touched file "C:\Windows\System32\en-US\KernelBase.dll.mui"  
"TrainingManagerEnterpriseSetup.tmp" touched file "C:\Windows\System32\netmsg.dll"  
"TrainingManagerEnterpriseSetup.tmp" touched file "C:\Windows\System32\en-US\netmsg.dll.mui"  
"TrainingManagerEnterpriseSetup.tmp" touched file "C:\Windows\System32\shfolder.dll"

After extraction similar files such as “gfix.exe” and “gbak.exe” were found:

Name	Date modified	Type	Size
doc	8/29/2018 8:47 AM	File folder	
intl	8/29/2018 8:47 AM	File folder	
udf	8/29/2018 8:47 AM	File folder	
gbak,1	8/29/2018 8:47 AM	Application	168 KB
gbak,2	8/29/2018 8:47 AM	Application	228 KB
gfix,1	8/29/2018 8:47 AM	Application	44 KB
gfix,2	8/29/2018 8:47 AM	Application	88 KB
hh	8/29/2018 8:47 AM	Application	11 KB
TrainingManager	8/29/2018 8:47 AM	Application	1,454 KB
TrainingManager.vshost	8/29/2018 8:47 AM	Application	6 KB

The Training Manager was considered malicious<sup>11</sup>.

**TrainingManager.exe**

This report is generated from a file or URL submitted to this webservice on August 29th 2018 17:49:38 (CEST) and action script *Heavy Anti-Evasion*

Guest System: Windows 7 32 bit, Home Premium, 6.1 (build 7601), Service Pack 1

Report generated by Falcon Sandbox v8.10 @ Hybrid Analysis

malicious

Threat Score: 65/100

AV Detection: 2%

[Link](#)
[Twitter](#)
[E-Mail](#)

[Overview](#)
[Login to Download Sample \(603KB\)](#)
[Downloads](#)
[External Reports](#)
[Re-analyze](#)
[Hash Seen Before](#)
[No similar samples](#)
[Report Abuse](#)

## Similar Pointing Directories

SECTION HEADERS						
Name	Virtual Size	Virtual Address	Size of Raw Data	Pointer to Raw Data	Characteristics	Pointing Directories
<input checked="" type="checkbox"/> CODE	00009174h	00401000h	00009200h	00000400h	60000020h	
<input checked="" type="checkbox"/> DATA	0000024Ch	0040B000h	00000400h	00009600h	C0000040h	
<input checked="" type="checkbox"/> BSS	00000E48h	0040C000h	00000000h	00009A00h	C0000000h	
<input checked="" type="checkbox"/> .idata	00000950h	0040D000h	00000A00h	00009A00h	C0000040h	Import Table
<input checked="" type="checkbox"/> .tls	00000008h	0040E000h	00000000h	0000A400h	C0000000h	
<input checked="" type="checkbox"/> .rdata	00000018h	0040F000h	00000200h	0000A400h	50000040h	TLS Table
<input checked="" type="checkbox"/> .reloc	000008B4h	00410000h	00000000h	0000A600h	50000040h	
<input checked="" type="checkbox"/> .rsrc	00002A00h	00411000h	00002A00h	0000A600h	50000040h	Resource Table

Similar resources as executable files:

<b>Behaviour:</b>	Find resource in self with type of PE
<b>Detail info:</b>	(FindResourceA) hModule = 0x00400000, ResName: REGDLL_EXE, ResType: (FindResourceA) hModule = 0x00400000, ResName: SHFOLDERDLL, ResType:

<sup>11</sup> <https://www.hybrid-analysis.com/sample/c9a5607f8b33472ef60f1e3647d7863bcd6a28ad299f693bd44cea092e5bf176>

## Similar identification of admin privileges:

```
Behaviour: Find file

Detail info: FileName = C:\DOCUME~1
             FileName = C:\DOCUME~1\ADMINI~1
             FileName = C:\DOCUME~1\ADMINI~1\LOCALS~1
             FileName = C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp
             FileName = C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\is-MSESL.tmp
             FileName = C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\is-MSESL.tmp\996E.tmp
             FileName = C:\Documents and Settings
             FileName = C:\Documents and Settings\Administrator
             FileName = C:\Documents and Settings\Administrator\「开始」菜单
             FileName = C:\Documents and Settings\Administrator\「开始」菜单\程序
             FileName = C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\is-9O09G.tmp\*
             FileName = C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\is-9O09G.tmp\_isetup\*
```

## Use of Mutex

```
Behaviour: Create mutex

Detail info: CTF.LBES.MutexDefaultS-*
             CTF.Compart.MutexDefaultS-*
             CTF.Asm.MutexDefaultS-*
             CTF.Layouts.MutexDefaultS-*
             CTF.TMD.MutexDefaultS-*
             CTF.TimListCache.FMPDefaultS-*MUTEX.DefaultS-*
             MSCTF.Shared.MUTEX.EBH
             MSCTF.Shared.MUTEX.MEK
```

## These are similar characteristics observed in each section:

C:\Users\TrainingManagerEnterpriseSetup.exe Filealyzer								
File Report Settings Language OpenSBI Help								
General OpenSBI Version Security Authenticode Resources Streams PE Header PE Sections DisAsm Import/Export table Hex dump Text preview Archive preview								
Size:	52224							
CRC-32:	C0C47DA5							
MD5:	078E7C20333F6F50E92DB38DFAD99275							
Section	VirtSize	VirtAddr	PhysSize	PhysAddr	Flags	CRC32	MD5	Characteristics
CODE	00009174	00001000	00009200	00000400	60000020	43545943	EA92E1415BC80E2738E334267EBB921	Code, Execute Access, Read Access
DATA	0000024C	0000B000	00000400	00009600	C0000040	6352B82C	F96DA19D2571A42BDF1B9E8BD62EC99	Initialized Data, Read Access, Write Access
BSS	00000E48	0000C000	00000000	00009A00	C0000000			Read Access, Write Access
.idata	00000950	0000D000	00000A00	00009A00	C0000040	AA2C7DA9	BB5485BF968B970E5EA81292AF2ACDBA	Initialized Data, Read Access, Write Access
.tls	00000008	0000E000	00000000	0000A400	C0000000			Read Access, Write Access
.rdata	00000018	0000F000	00000200	0000A400	50000040	5EC36383	9BA824905BF9C7922B6FC87A38B74366	Initialized Data, Shared, Read Access
.reloc	000008B4	00010000	00000000	0000A600	50000040			Initialized Data, Shared, Read Access
.rsrc	00002A00	00011000	00002A00	0000A600	50000040	D51DDE51	9A54223C306392BFE879A13D0B26F2C8	Initialized Data, Shared, Read Access



## Aspects of anti-debugging functions:

Behaviour: Open event

Detail info: HookSwitchHookEnabledEvent  
GlobalCrypt32LogoffEvent  
\_fCanRegisterWithShellService  
CTF.ThreadMIConnectionEvent.00000714.00000000.00000013  
CTF.ThreadMarshallInterfaceEvent.00000714.00000000.00000014  
CTF.ThreadMIConnectionEvent.00000714.00000000.00000014  
MSCTF.SendReceiveConection.Event.EBH.IC  
MSCTF.SendReceive.Event.EBH.IC

## Use of sleep function:

Behaviour: Call Sleep function

Detail info: [1]: Milliseconds = 50.  
[2]: Milliseconds = 50.  
[3]: Milliseconds = 50.  
[4]: Milliseconds = 50.  
[5]: Milliseconds = 50.  
[6]: Milliseconds = 50.  
[7]: Milliseconds = 50.  
[8]: Milliseconds = 50.  
[9]: Milliseconds = 50.  
[10]: Milliseconds = 50.  
[2]: Milliseconds = 250.  
[3]: Milliseconds = 250.  
[4]: Milliseconds = 250.  
[5]: Milliseconds = 250.  
[6]: Milliseconds = 250.

## Identified Risks<sup>12</sup>

HYBRID ANALYSIS

Home Submissions Resources Jobs Contact

IP, Domain, Hash...

**TrainingManagerEnterpriseSetup.exe**

This report is generated from a file or URL submitted to this webservice on August 28th 2018 13:59:18 (CEST) and action script *Heavy Anti-Evasion*  
Guest System: Windows 7 32 bit, Home Premium, 6.1 (build 7601), Service Pack 1  
Report generated by Falcon Sandbox v8.10 @ Hybrid Analysis

Overview Login to Download Sample (13MiB) Downloads External Reports Re-analyze Hash Not Seen Before

No similar samples Report Abuse

**suspicious**  
Threat Score: 85/100  
AV Detection: 1%

Link Twitter E-Mail

### Incident Response

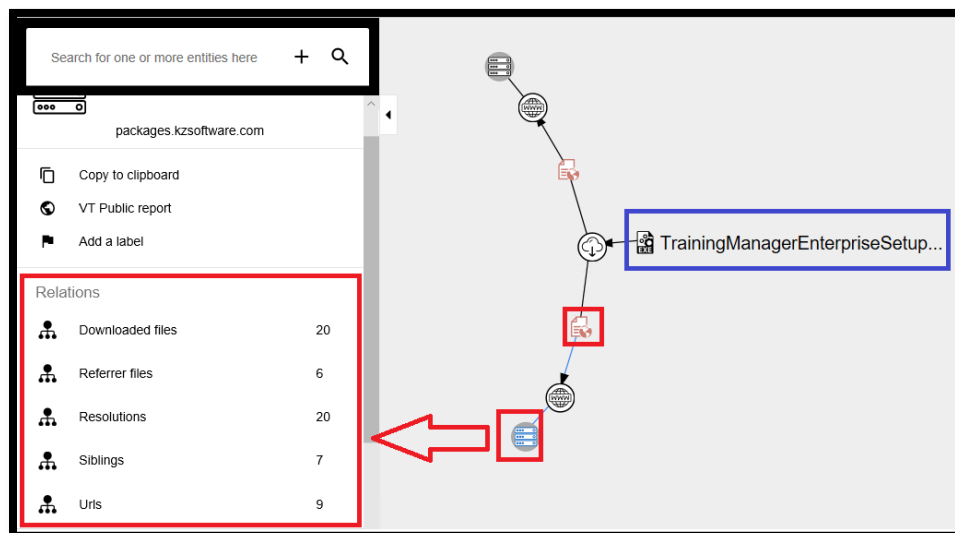
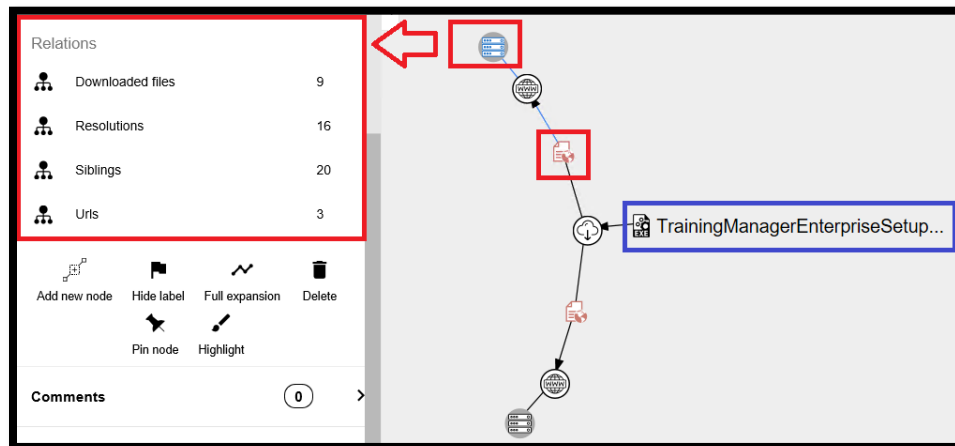
#### Risk Assessment

Remote Access	Reads terminal service related keys (often RDP related)
Persistence	Writes data to a remote process
Fingerprint	Reads the active computer name Reads the cryptographic machine GUID
Spreading	Opens the MountPointManager (often used to detect additional infection locations)

Activate V  
Go to Setting

<sup>12</sup> <https://www.hybrid-analysis.com/sample/df8bbb030b7411a3efe4af6bf6629732348a1fe1c7f0b57bf4c202d9fc067fdc?environmentId=100>

## Final Map



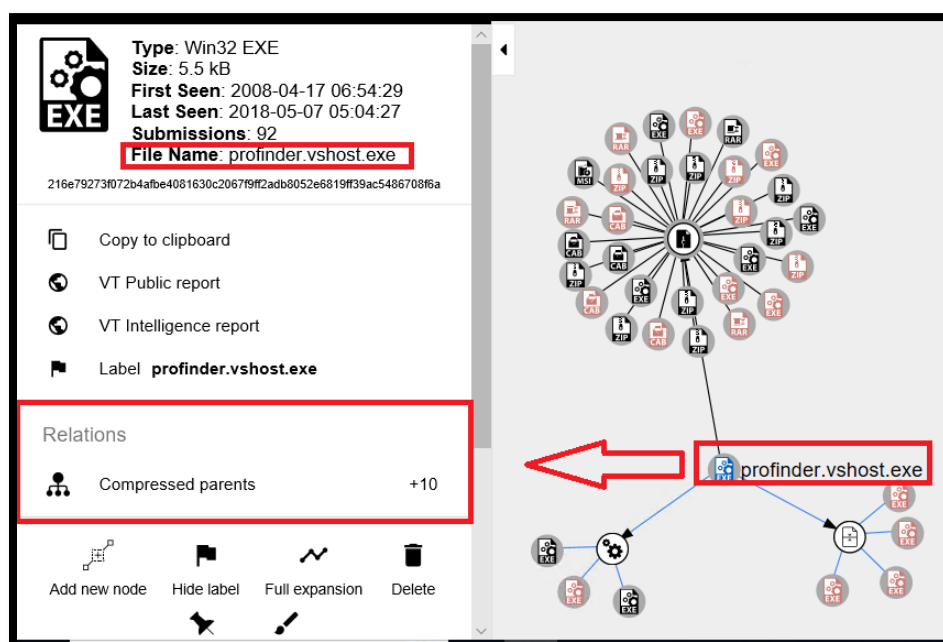
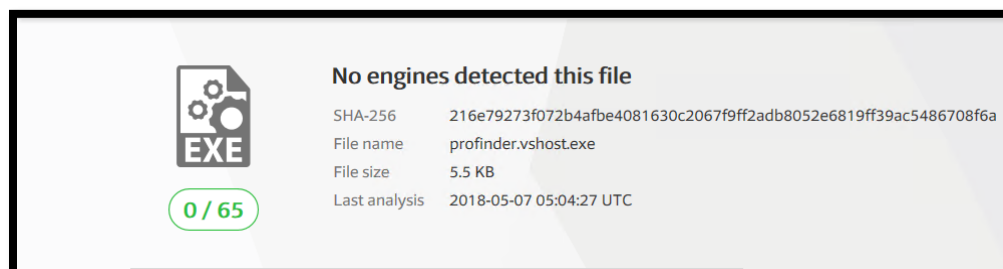
## Possible Attacks

MITRE ATT&CK™ Techniques Detection

Minimal

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Command and Control
		Hooking 1	Hooking 1	Code Signing 1	Hooking 1	Network Service Scanning 1	Remote Desktop Protocol 1		Data Compressed 1	
		Kernel Modules and Extensions 1	Process Injection 1	File Deletion 2		Peripheral Device Discovery 2				
				Modify Registry 1		Process Discovery 1				
				Process Injection 1		Query Registry 4 1				

Finally, an additional suspicious file was found in both software under the names of AssetManager.vshost and TrainingManager.vshost called “profinder.vshost.exe”. In general, hosting process files (vshost.exe) are to be used by Microsoft Visual Studio and should not be run directly or deployed with an application<sup>13</sup>.



## Attack Techniques

Overall the vulnerabilities in both software are similar. A few differences were found in the Discovery section that in the Training Manager an attacker could use a Network Service Scanning technique and in the Asset Manager an Email Collection technique would be an additional attack alternative.

<sup>13</sup> <https://msdn.microsoft.com/en-us/library/ms185331.aspx>