‹ ALL BLOGS (/blogs)

SECURITY ESSENTIALS (/BLOGS/SECURITY-ESSENTIALS/)

Building a Malware Analysis Lab: Become a Malware Analysis Hunter in 2019

MARCH 1, 2019 | @SUDOSEV (/BLOGS/AUTHOR/SUDOSEV)

As time goes by, criminals are developing more and more complex methods of obscuring how their malware operates, making it increasingly difficult to detect and analyze. The list of tactics used is seemingly endless and can include obfuscation, packers, executing from memory with no file drop, and P2P botnet architecture with frontline command and control servers (C2s) and gateways being compromised websites. Add to these tactics the concerns about Domain Generations Algorithms (DGA), Fast Flux and Dynamic DNS (/blogs/security-essentials/dynamic-dns-security-and-potential-threats), and you complicate the mix even further.

Tracking all of these elements might be difficult, but in all honesty, you don't need 10 years of experience in malware analysis and a bunch of certificates to help you win this battle. You just need to experiment. One great way to learn about malware is to build your own home lab and play with actual malware samples within this environment. This can be a fun and educational project even if you are not an InfoSec pro. If you do happen to be an InfoSec pro, the things you learn in your home lab just might help you do your job more effectively. So how do you set one up? A few simple guidelines will get you started.

What Should Be In Your Malware Analysis Lab?

So what are the essential components of a home lab? There is no right or wrong answer here. You can setup a virtual machine and make that your lab. As long as you sandbox the malware you're analyzing, you should consider your set-up a laboratory environment in my opinion.

However, I would also like to state that just because you are analyzing within a sandbox environment, it does not mean you are completely secure. There are vulnerabilities present in older versions of virtualization software that can allow escapes from virtual machines so you need to ensure that your virtualization environment is up-to-date and patched against such exploits. I would also recommend starting your virtual lab with multiple boxes, which increases security. Using the PFsense firewall is a fantastic start and having an IDS box (Snort/Suricata/Bro) is an additional bonus. I won't go into too much detail about virtual network security because my good friend Tony Robinson (@da_667 (https://twitter.com/da_667)) already has this well documented in a blog: ~~https://blindseeker.com/blahg/?p=375~~ (no longer available).

Personally, I do not have an overly sophisticated lab setup. It is mostly boxes here and there, but each box has a specific purpose. My home lab currently consists of the following components:

- Windows 7 Virtual Machine (home network)
- Windows 10 Virtual Machine (home network)
- Ubuntu 15.10 Virtual Machine (home network)
- Ubuntu 14.04 SSH Honeypot (VPS)
- Windows Server 2012 R2 (VPS)
- Security Onion on an empty server which I'm still configuring
- Various email spam traps for collecting macro malware & other specimens (such as phishing attempts or malware which may be hosted on a link in an email).

I feel as though I cover a lot of ground with this setup. I have collectors to harvest malware automatically, which I can then pull down to my virtual machines on my home network for further analysis. I also have my Windows VPS to accomplish a 'Malware analysis in the cloud' kind of thing. My current setup allows me to analyze pretty much any piece of malware I come across. However, for the purpose of this post, I'll be discussing my toolset and my approach to analyzing malware on Windows systems.

Figure out What You're Looking For

Before we run all of our tools and start clicking on malware links in spam hoping to see some results, we first need to answer some questions:

- What exactly are you are looking for?
- Why are you doing this?

- Once you have analyzed the malware specimen, what do you plan to do with the information extracted from our analysis session?
- What is your end goal?

My end goal is to pass on my results to security and threat intelligence vendors, such as AlienVault, as well as other researchers so that they can add these to their blacklists/rulesets in an attempt to shut down the malware we've discovered. My end goal is to raise awareness and shut things down before the attackers really get the ball rolling. Even if you are only blacklisting newly discovered C2s, sharing your information helps the InfoSec community.

On To the Malware Hunter Toolset!

The tools in this list have become my Swiss army knife for basic malware analysis. You should definitely spend time trying out various tools so that you can build a Swiss army knife to suit your own needs.

- **Wireshark** - Incredibly powerful packet analysis tool which we use for monitoring any additional payloads our malware specimen may be attempting to download. It also highlights post-infection traffic to give us an indication of how our malware specimen is operating.
- **PeStudio** - A great tool for analyzing Portable Executable (PE) files. Provides information such as what functions/APIs are being called, imported libraries, VirusTotal information if it's currently in their database, strings found in the specimen, packer used (if any) and so on.
- **RegShot / TotalCommander** - RegShot is a tool for taking snapshots of the machine's registry allowing you to compare the registry before infection and after infection. I also use TotalCommander for the same thing, which I'll explain more later on.
- **ProcessExplorer** – This is part of the SysInternals Suite from Microsoft. This is essentially a “Task Manager on Steroids” and includes a lot of great features. One of my favorite features is the color coding of processes, which makes it easy to identify rogue processes spawned by malware.
- **ProcessMonitor** - Another tool from the SysInternals Suite. ProcessMonitor allows you to track I/O operations on the machine and view what's happening to your registry, file system, processes. It also provides some basic network information (useful for if your packet capture from Wireshark is HUGE and you need a bit of help).
- **Fakenet / ApateDNS** - Honorable mentions for these tools, which I sometimes use. Both of these allow you to essentially “blackhole” your outbound traffic to an address that you specify (127.0.0.1 maybe? You can then run a malware specimen and capture all DNS requests (and more with Fakenet), in order to identify C2s ready for reporting.

- **Hexinator** – Hexinator is an incredibly powerful Hex editor, and while this is only used in a small number of malware investigations, I feel it should still be included within a toolkit like this.
- **Resource Hacker** – Resource hacker allows you to feed in a file (an executable for example) and it will then break that file down into the resources that make up the file. This is useful for extracting malicious files that have been embedded within another file. As an example, I have seen legitimate executables repackaged with a malicious executable. This was identified in resources and extracted with Resource Hacker.

Let's Begin! (Basic Static Analysis)

Considering that this blog is aimed at hobbyists rather than security professionals in a business environment, we can skip a lot of incident response practices, such as taking an image of the disk and creating a memory dump. However, depending on your findings, memory dumps may still be useful for a home user. The beauty of analyzing in a sandbox / virtualized environment is that you can restore to a snapshot and run it again for anything you may have missed. This is especially useful when multiple C2s are involved.

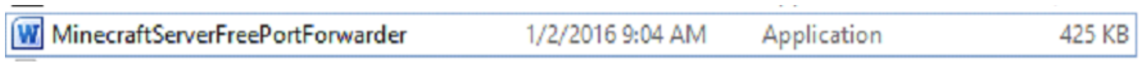
Here's the checklist you should always cover before investigating a malware specimen further (using PeStudio in our case, although other tools exist for this, too):

- Filename
- File creation date
- Compile date
- File size (useful for comparison to other samples later on)
- File appearance (Anything suspicious? .exe with a Word icon?)
- Hash of the file

File creation date/compile date are interesting pieces of data when investigating malware, as they can provide insights. This information can prove to be crucial in investigations and/or for identifying the authors of the malware, especially if an attack is politically-driven. This is especially true when the malware you are looking at is responsible for crippling an Industrial Control System (ICS). For example, @malwarejake (https://twitter.com/MalwareJake) recently conducted some analysis on the Ukrainian power malware in which he discusses the significance of the compile time stamp. You can read more about that here - http://malwarejake.blogspot.co.uk/search?updated-max=2016-01-02T10:07:00-08:00&max-results=1&start=42&by-date=false (http://malwarejake.blogspot.co.uk/search?updated-max=2016-01-02T10:07:00-08:00&max-results=1&start=42&by-date=false)

Step 1: First Appearance

To begin with we will look at a malware sample called "MinecraftServerFreePortForwarder". Mainly due to how easy and obvious this is, it makes for some quick learning.

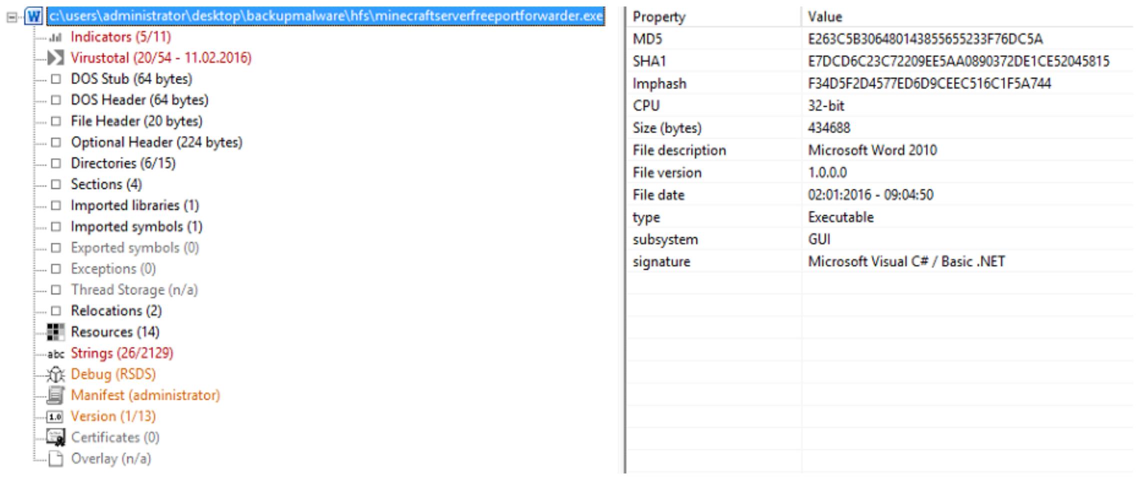


MinecraftServerFreePortForwarder	1/2/2016 9:04 AM	Application	425 KB

Upon first inspection, we can see that it appears to be a Word icon but it's actually an Application (.exe) – our first red flag. The other indicator is that this file is 425KB which is a little smaller than the usual malicious files you will come across, but is a pretty good sign that there is something off about this file.

Note: Most malware I've run into is generally 1MB or smaller.

We will now move this file over to PeStudio so that we can dive deeper to see what is going on.

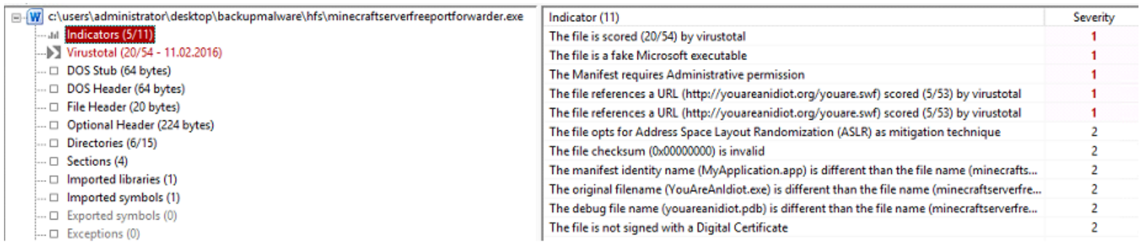


c:\users\administrator\desktop\backupmalware\hfs\minecraftserverfreeportforwarder.exe
- Indicators (5/11)
- Virustotal (20/54 - 11.02.2016)
- DOS Stub (64 bytes)
- DOS Header (64 bytes)
- File Header (20 bytes)
- Optional Header (224 bytes)
- Directories (6/15)
- Sections (4)
- Imported libraries (1)
- Imported symbols (1)
- Exported symbols (0)
- Exceptions (0)
- Thread Storage (n/a)
- Relocations (2)
- Resources (14)
- Strings (26/2129)
- Debug (RSDS)
- Manifest (administrator)
- Version (1/13)
- Certificates (0)
- Overlay (n/a)

Property	Value
MD5	E263C5B306480143855655233F76DC5A
SHA1	E7DCD6C23C72209EE5AA0890372DE1CE52045815
Imphash	F34D5F2D4577ED6D9CEEC516C1F5A744
CPU	32-bit
Size (bytes)	434688
File description	Microsoft Word 2010
File version	1.0.0.0
File date	02:01:2016 - 09:04:50
type	Executable
subsystem	GUI
signature	Microsoft Visual C# / Basic .NET

The introduction screen displays the properties of the file along with the MD5, SHA1 and Import hash (Imphash) of the file which is essentially a unique identifier of the file. Hashes of files are used to verify the integrity of files; if you were to download a certain piece of software where the writers have provided the hash, you can use it to verify if you have the same version or if you have an altered version.

We can also see other useful information here such as file description, which at this point we know is fake. We have already established that this is an executable and the "type" field also verifies that for us. PeStudio will also highlight sections on the left. The coloring is exactly what you think it is. Red means that the file has exceeded the malicious indicator threshold, and orange means that malicious indicators exist but there are not enough of them to conclude whether or not the file is malicious. You can get a summary of this information in the "Indicators" section.



```
c:\users\administrator\desktop\backupmalware\hfs\minecraftserverfreeportforwarder.exe
    Indicators (5/11)
    Virustotal (20/54 - 11.02.2016)
    DOS Stub (64 bytes)
    DOS Header (64 bytes)
    File Header (20 bytes)
    Optional Header (224 bytes)
    Directories (6/15)
    Sections (4)
    Imported libraries (1)
    Imported symbols (1)
    Exported symbols (0)
    Exceptions (0)
```

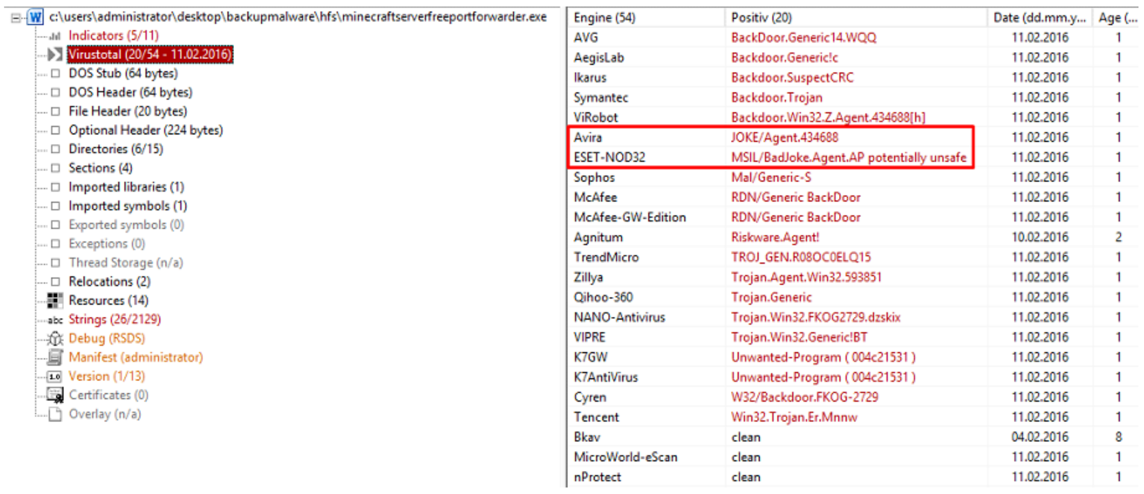
Indicator (11)	Severity
The file is scored (20/54) by virustotal	1
The file is a fake Microsoft executable	1
The Manifest requires Administrative permission	1
The file references a URL (http://youareanidiot.org/youare.swf) scored (5/53) by virustotal	1
The file references a URL (http://youareanidiot.org/youare.swf) scored (5/53) by virustotal	1
The file opts for Address Space Layout Randomization (ASLR) as mitigation technique	2
The file checksum (0x00000000) is invalid	2
The manifest identity name (MyApplication.app) is different than the file name (minecrafts...	2
The original filename (YouAreAnIdiot.exe) is different than the file name (minecraftserverfre...	2
The debug file name (youareanidiot.pdb) is different than the file name (minecraftserverfre...	2
The file is not signed with a Digital Certificate	2

The indicators section is usually what I review before moving any further, since it gives a quick summary of the findings. As you can see from this information, the file we are analyzing has already been submitted to VirusTotal. We can also see that PeStudio has detected that this file is attempting to masquerade as a Microsoft Word file, which was easy to identify in this case, but in some cases will not be so obvious.

Administrative privileges are required, but you could argue that this would be expected for a port forwarder (THE FILE IS STILL MALICIOUS!). In other circumstances, this is a nice indicator. For example, this file could be masquerading as a resume/CV which would not require administrative privileges. The last indicator on this list is also extremely useful. Software without a digital certificate should be looked into and anything masquerading as Microsoft software should be digitally signed by Microsoft. Always remember this as you will run into malware that masquerades as other legitimate software which SHOULD be digitally signed by the company publishing the software.

The most important pieces of information here are the **references to the URL** (youareanidiot.org) and the **original filename** (YouAreAnIdiot.exe). The **URL reference** should immediately be noted and you should be looking for this in Wireshark later on when you run the specimen. Of course, this URL may not be called for, but we have nothing else as of yet, so note it!

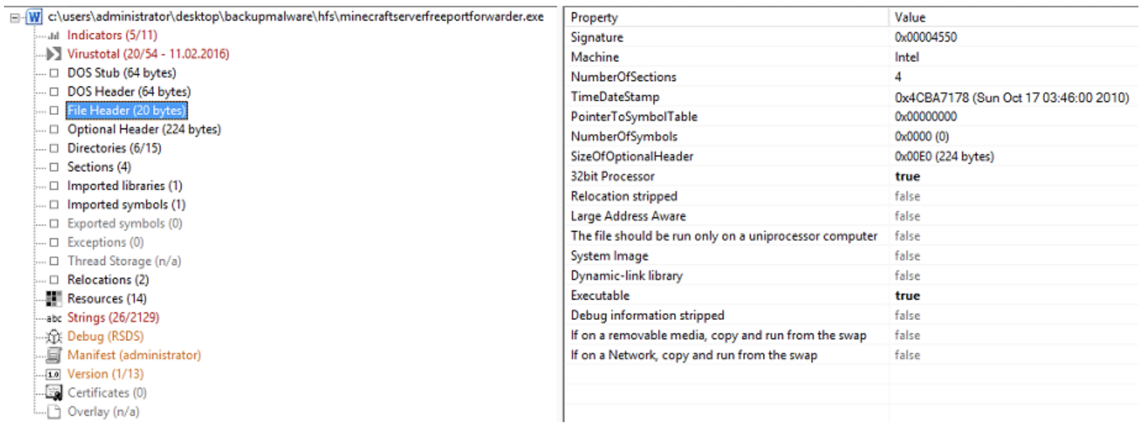
The final thing I see here is more for when you start reverse engineering. The reference to **Address Space Layout Randomization (ASLR)** is important because it’s a **mitigation technique** used to increase the difficulty of analysis.



```
c:\users\administrator\desktop\backupmalware\hfs\minecraftserverfreeportforwarder.exe
  Indicators (5/11)
  Virustotal (20/54 - 11.02.2016)
  DOS Stub (64 bytes)
  DOS Header (64 bytes)
  File Header (20 bytes)
  Optional Header (224 bytes)
  Directories (6/15)
  Sections (4)
  Imported libraries (1)
  Imported symbols (1)
  Exported symbols (0)
  Exceptions (0)
  Thread Storage (n/a)
  Relocations (2)
  Resources (14)
  Strings (26/2129)
  Debug (RSDS)
  Manifest (administrator)
  Version (1/13)
  Certificates (0)
  Overlay (n/a)
```

Engine (54)	Positiv (20)	Date (dd.mm.y...	Age (...
AVG	BackDoor.Generic14.WQQ	11.02.2016	1
AegisLab	Backdoor.Generic!c	11.02.2016	1
Ikarus	Backdoor.SuspectCRC	11.02.2016	1
Symantec	Backdoor.Trojan	11.02.2016	1
ViRobot	Backdoor.Win32.Z.Agent.434688[h]	11.02.2016	1
Avira	JOKE/Agent.434688	11.02.2016	1
ESET-NOD32	MSIL/BadJoke.Agent.AP potentially unsafe	11.02.2016	1
Sophos	Mal/Generic-S	11.02.2016	1
McAfee	RDN/Generic BackDoor	11.02.2016	1
McAfee-GW-Edition	RDN/Generic BackDoor	11.02.2016	1
Agnitum	Riskware.Agent!	10.02.2016	2
TrendMicro	TROJ_GEN.R08OC0ELQ15	11.02.2016	1
Zillya	Trojan.Agent.Win32.593851	11.02.2016	1
Qihoo-360	Trojan.Generic	11.02.2016	1
NANO-Antivirus	Trojan.Win32.FKOG2729.dzskix	11.02.2016	1
VIPRE	Trojan.Win32.Generic!BT	11.02.2016	1
K7GW	Unwanted-Program (004c21531)	11.02.2016	1
K7AntiVirus	Unwanted-Program (004c21531)	11.02.2016	1
Cyren	W32/Backdoor.FKOG-2729	11.02.2016	1
Tencent	Win32.Trojan.Er.Mnnw	11.02.2016	1
Bkav	clean	04.02.2016	8
MicroWorld-eScan	clean	11.02.2016	1
nProtect	clean	11.02.2016	1

The **VirusTotal** tab is incredibly useful, but just to make this clear, do not rely exclusively on this. I've analyzed this malware file before and I've also seen it in action on other systems, it is in fact just a bad joke which will throw smiley faces all over your screen and play audio that laughs at you. Also note that several antivirus vendors do not have a signature for this file. Especially with huge malware threats such as ransomware and banking trojans such as Dridex – we are hunting some evil malware.

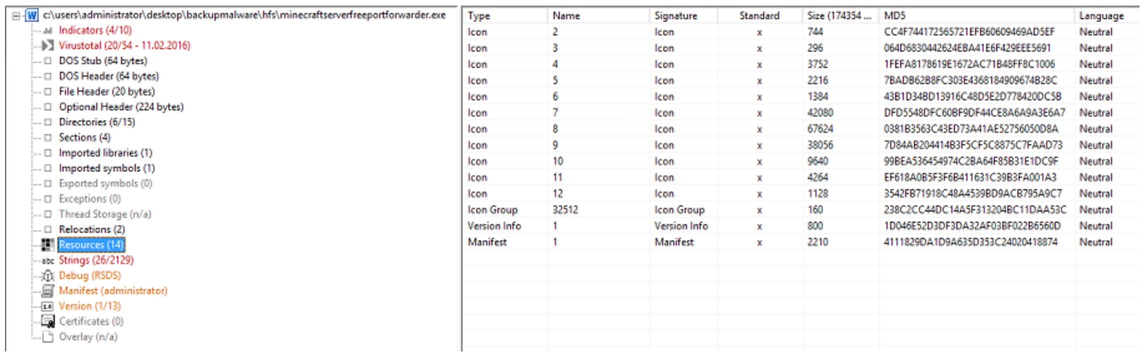


```
c:\users\administrator\desktop\backupmalware\hfs\minecraftserverfreeportforwarder.exe
  Indicators (5/11)
  Virustotal (20/54 - 11.02.2016)
  DOS Stub (64 bytes)
  DOS Header (64 bytes)
  File Header (20 bytes)
  Optional Header (224 bytes)
  Directories (6/15)
  Sections (4)
  Imported libraries (1)
  Imported symbols (1)
  Exported symbols (0)
  Exceptions (0)
  Thread Storage (n/a)
  Relocations (2)
  Resources (14)
  Strings (26/2129)
  Debug (RSDS)
  Manifest (administrator)
  Version (1/13)
  Certificates (0)
  Overlay (n/a)
```

Property	Value
Signature	0x00004550
Machine	Intel
NumberOfSections	4
TimeDateStamp	0x4CBA7178 (Sun Oct 17 03:46:00 2010)
PointerToSymbolTable	0x00000000
NumberOfSymbols	0x0000 (0)
SizeOfOptionalHeader	0x00E0 (224 bytes)
32bit Processor	**true**
Relocation stripped	false
Large Address Aware	false
The file should be run only on a uniprocessor computer	false
System Image	false
Dynamic-link library	false
Executable	**true**
Debug information stripped	false
If on a removable media, copy and run from the swap	false
If on a Network, copy and run from the swap	false

I have skipped **DOS Stub** and **DOS Header** because I don't consider them useful in this exercise. File Header, however, is incredibly useful for building the bigger picture of what this file is or more what it is targeting. **We can see that this piece of malware is targeting 32bit Intel (x86) architecture. You will come across other architectures (machine field), but 32bit Intel is by far the most common that you are likely to see.** This tab also gives us the compile time of the malware, which you can see in the **TimeDateStamp** field.

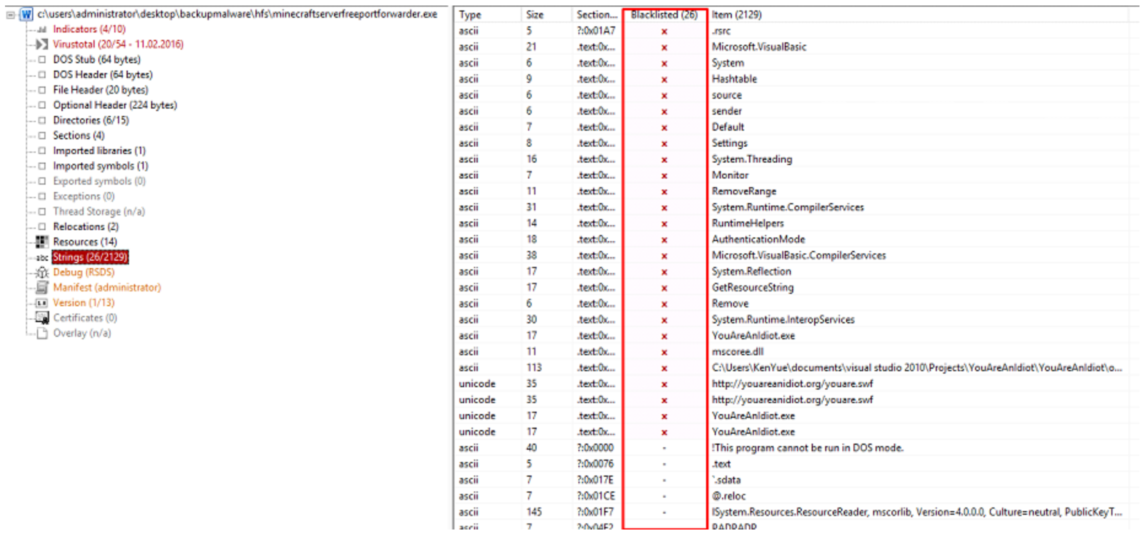
I have seen malware that was supposedly compiled in 1998 and in some cases, in the future! Obscure compile dates is also an indicator that PeStudio will check for when you load your file into the application. In short, **File Header** asks a bunch of Boolean questions which is awesome for quick analyses of a file i.e. can this file do X? Yes or no.



```
c:\users\administrator\desktop\backupmalware\hfs\minecraftserverfreeportforwarder.exe
  Indicators (4/10)
  Virustotal (20/54 - 11.02.2016)
  DOS Stub (64 bytes)
  DOS Header (64 bytes)
  File Header (20 bytes)
  Optional Header (224 bytes)
  Directories (6/15)
  Sections (4)
  Imported libraries (1)
  Imported symbols (1)
  Exported symbols (0)
  Exceptions (0)
  Thread Storage (n/a)
  Relocations (2)
  Resources (14)
  Strings (26/2129)
  Debug (RSDS)
  Manifest (administrator)
  Version (1/13)
  Certificates (0)
  Overlay (n/a)
```

Type	Name	Signature	Standard	Size (174354 ...	MD5	Language
Icon	2	Icon	x	744	CC4F744172565721EFB60609469AD5EF	Neutral
Icon	3	Icon	x	296	064D6830442624EBA41E6F429EEE5691	Neutral
Icon	4	Icon	x	3752	1FEFA8178619E1672AC71B48FF8C1006	Neutral
Icon	5	Icon	x	2216	7BADB62B8FC303E4368184909674B28C	Neutral
Icon	6	Icon	x	1384	43B1D34BD13916C48D5E2D778420DC5B	Neutral
Icon	7	Icon	x	42080	DFD5548DFC60BF9DF44CE8A6A9A3E6A7	Neutral
Icon	8	Icon	x	67624	0381B3563C43ED73A41AE52756050D8A	Neutral
Icon	9	Icon	x	38056	7D84AB204414B3F5CF5C8875C7FAAD73	Neutral
Icon	10	Icon	x	9640	99BEA536454974C2BA64F85B31E1DC9F	Neutral
Icon	11	Icon	x	4264	EF618A0B5F3F6B411631C39B3FA001A3	Neutral
Icon	12	Icon	x	1128	3542FB71918C48A4539BD9ACB795A9C7	Neutral
Icon Group	32512	Icon Group	x	160	238C2CC44DC14A5F313204BC11DAA53C	Neutral
Version Info	1	Version Info	x	800	1D046E52D3DF3DA32AF03BF022B6560D	Neutral
Manifest	1	Manifest	x	2210	4111829DA1D9A635D353C24020418874	Neutral

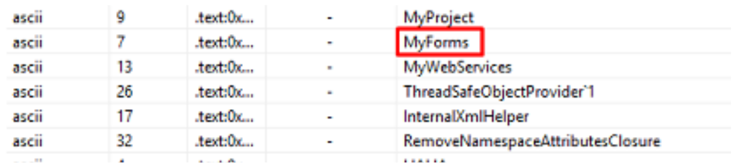
I'll skip past a bunch of the reverse engineering-ish tabs here, since there is a lot more left to cover. The resources tab will tell you what items/objects are present in the file i.e., icons, but most importantly, you can sometimes come across other executable files embedded within this primary file - the resources tab will inform you if that is the case. We'll extract an embedded executable later on when we use some other tools.



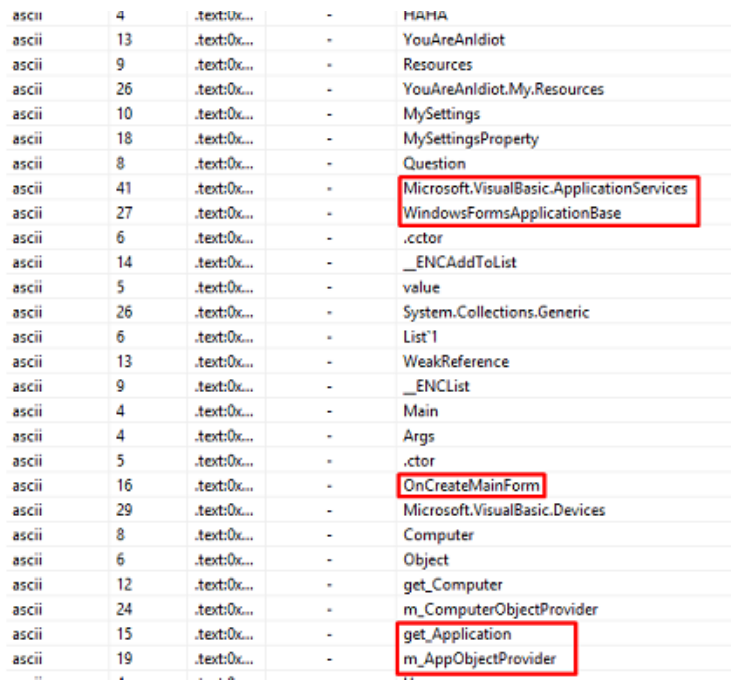
```
c:\users\administrator\desktop\backupmalware\hfs\minecraftserverfreeportforwarder.exe
  Indicators (4/10)
  Virustotal (20/54 - 11.02.2016)
  DOS Stub (64 bytes)
  DOS Header (64 bytes)
  File Header (20 bytes)
  Optional Header (224 bytes)
  Directories (6/15)
  Sections (4)
  Imported libraries (1)
  Imported symbols (1)
  Exported symbols (0)
  Exceptions (0)
  Thread Storage (n/a)
  Relocations (2)
  Resources (14)
  Strings (26/2129)
  Debug (RSDS)
  Manifest (administrator)
  Version (1/13)
  Certificates (0)
  Overlay (n/a)
```

Type	Size	Section...	Blacklisted (26)	Item (2129)
ascii	5	?:0x01A7	x	.rsrc
ascii	21	.text:0x...	x	Microsoft.VisualBasic
ascii	6	.text:0x...	x	System
ascii	9	.text:0x...	x	Hashtable
ascii	6	.text:0x...	x	source
ascii	6	.text:0x...	x	sender
ascii	7	.text:0x...	x	Default
ascii	8	.text:0x...	x	Settings
ascii	16	.text:0x...	x	System.Threading
ascii	7	.text:0x...	x	Monitor
ascii	11	.text:0x...	x	RemoveRange
ascii	31	.text:0x...	x	System.Runtime.CompilerServices
ascii	14	.text:0x...	x	RuntimeHelpers
ascii	18	.text:0x...	x	AuthenticationMode
ascii	38	.text:0x...	x	Microsoft.VisualBasic.CompilerServices
ascii	17	.text:0x...	x	System.Reflection
ascii	17	.text:0x...	x	GetResourceString
ascii	6	.text:0x...	x	Remove
ascii	30	.text:0x...	x	System.Runtime.InteropServices
ascii	17	.text:0x...	x	YouAreAnIdiot.exe
ascii	11	.text:0x...	x	mscoree.dll
ascii	113	.text:0x...	x	C:\Users\KenYue\documents\visual studio 2010\Projects\YouAreAnIdiot\YouAreAnIdiot\o...
unicode	35	.text:0x...	x	http://youareanidiot.org/youare.swf
unicode	35	.text:0x...	x	http://youareanidiot.org/youare.swf
unicode	17	.text:0x...	x	YouAreAnIdiot.exe
unicode	17	.text:0x...	x	YouAreAnIdiot.exe
ascii	40	?:0x0000	-	!This program cannot be run in DOS mode.
ascii	5	?:0x0076	-	.text
ascii	7	?:0x017E	-	`.sdata
ascii	7	?:0x01CE	-	@.reloc
ascii	145	?:0x01F7	-	lSystem.Resources.ResourceReader, mscorlib, Version=4.0.0.0, Culture=neutral, PublicKeyT...
ascii	7	?:0x04F2		PADPADP

"Strings" is what I consider to be the most interesting tab available. Just be sure to ignore the blacklisted field because there will often be a lot of information hidden in the rest of the strings output as you can see below.



ascii	9	.text:0x...	-	MyProject
ascii	7	.text:0x...	-	MyForms
ascii	13	.text:0x...	-	MyWebServices
ascii	26	.text:0x...	-	ThreadSafeObjectProvider`1
ascii	17	.text:0x...	-	InternalXmlHelper
ascii	32	.text:0x...	-	RemoveNamespaceAttributesClosure



ascii	4	.text:0x...	-	HAHA
ascii	13	.text:0x...	-	YouAreAnIdiot
ascii	9	.text:0x...	-	Resources
ascii	26	.text:0x...	-	YouAreAnIdiot.My.Resources
ascii	10	.text:0x...	-	MySettings
ascii	18	.text:0x...	-	MySettingsProperty
ascii	8	.text:0x...	-	Question
ascii	41	.text:0x...	-	Microsoft.VisualBasic.ApplicationServices
ascii	27	.text:0x...	-	WindowsFormsApplicationBase
ascii	6	.text:0x...	-	.cctor
ascii	14	.text:0x...	-	_ENCAddToList
ascii	5	.text:0x...	-	value
ascii	26	.text:0x...	-	System.Collections.Generic
ascii	6	.text:0x...	-	List`1
ascii	13	.text:0x...	-	WeakReference
ascii	9	.text:0x...	-	_ENCList
ascii	4	.text:0x...	-	Main
ascii	4	.text:0x...	-	Args
ascii	5	.text:0x...	-	.ctor
ascii	16	.text:0x...	-	OnCreateMainForm
ascii	29	.text:0x...	-	Microsoft.VisualBasic.Devices
ascii	8	.text:0x...	-	Computer
ascii	6	.text:0x...	-	Object
ascii	12	.text:0x...	-	get_Computer
ascii	24	.text:0x...	-	m_ComputerObjectProvider
ascii	15	.text:0x...	-	get_Application
ascii	19	.text:0x...	-	m_AppObjectProvider

Certain entries in this strings output suggest that this piece of malware will launch a form or application on our screen. This is quite odd since most malware wishes to remain hidden, and you may have been expecting PowerShell launch commands to hide the window while code executes. This is a perfect example of why analyzing the entire strings output is something you should always do.

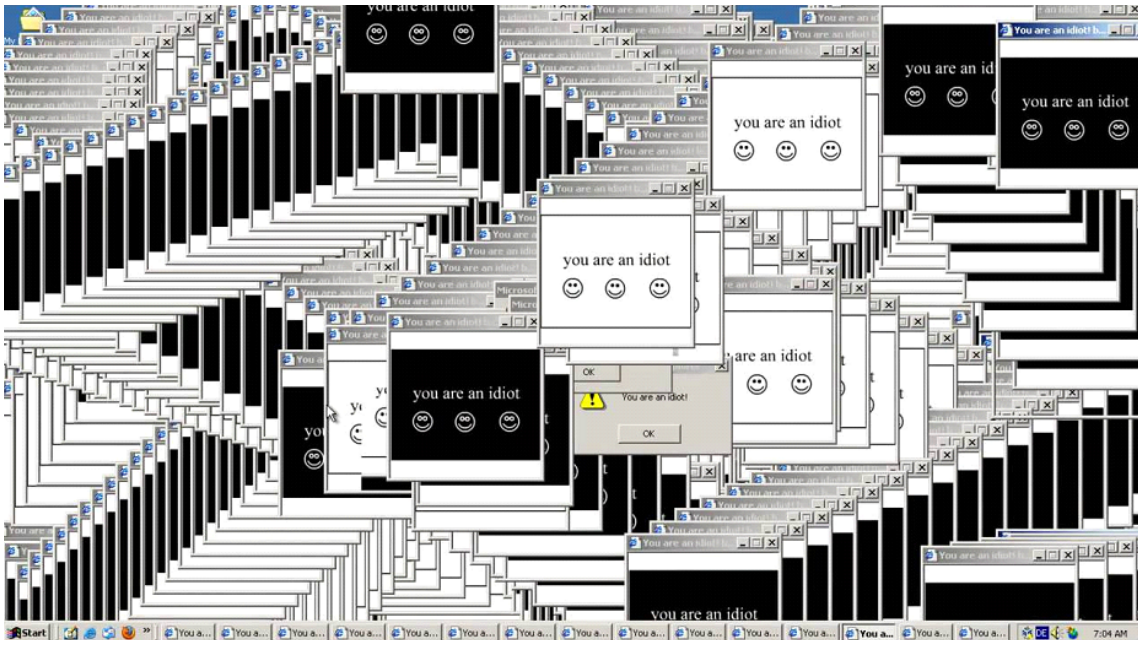
Step 2: Make some realistic predictions

In the previous section, we skipped **Imported Libraries and Imported Symbols**. This is due to the fact that this specimen has one entry for each. Those two tabs are where you should be spending most of your time and should conclude your PeStudio investigation. It's there that you will find the dynamic-link libraries and the functions/API calls being requested by the malware.

While it's incredibly difficult to guess what the malware can possibly do from a list of functions, it can sometimes give you an indication. Certain dynamic-link libraries will eliminate some possibilities and bring others into the spotlight. By process of elimination, you can make a good guess at what you think the file is going to do before you run it.

You can then run the specimen and do your checks while cross-referencing the results of the test with the functions listed in PeStudio. You can do this multiple times, but in order to get the perfect picture, you will have to reverse engineer the malware specimen (which is what I consider the end goal of malware analysis).

Since our little Minecraft port forwarder is nothing spectacular, I'll give you the result of what it does right here so that we can move onto something juicier.



Findings from our basic static analysis match exactly what we see here, but remember, the goal of most malware is to install silently and hide. This one is quite the opposite.

In part 2 of this blog post, we will use all of the tools listed above, we will analyze various malware samples and I will provide in-depth detail of how I use these tools and what indicators I look for when investigating suspected malware. For now, I will leave you with a short reading list that may be useful if malware analysis is something you feel you may enjoy.

- Just starting out with malware analysis? - *Practical Malware Analysis*
- Deep DFIR/Malware forensics - *Malware Forensics Field Guide for (Windows/Linux) Systems*
- Advanced malware analysis/heavy RE - *Malware Analysts Cookbook & DVD*

Aside from these books, it is always good to read reports about current malware threats. Reading through research will give you ideas on how to improve your own analysis methodology, and it'll also teach you some of the tips and tricks that malware authors tend to use in their campaigns.

I'm happy for anyone to reach out if help is required or if you just want a friendly conversation. You can reach me on Twitter @sudosev (https://twitter.com/sudosev) or via email sevaara@protonmail.ch (mailto:sevaara@protonmail.ch).

About the Author

Sev is a Computer & Network Security student, Intern IT Security Analyst and Malware hunter/analyst hobbyist.

(https://www.alienvault.com/blogs/author/sudosev)
About the Author: *@sudosev, Guest blogger*

Computer & Network Security student, Intern IT Security Analyst and Malware hunter/analyst hobbyist.

Read more posts from @sudosev › (/blogs/author/sudosev)

TAGS: malware hunting (/blogs/tag/malware+hunting), home malware lab (/blogs/tag/home+malware+lab)

(/blogs/tag/home+malware+lab)
(/blogs/tag/home+malware+lab)**‹ BACK TO ALL BLOGS (https://www.alienvault.com/blogs/security-essentials)**