CS 373 Summer 2019

Week 3 Write Up

Kirsten Wollam

MALWARE DEFENSES

Good malware defense is made up of several stages and goals. First we must be able to determine if there is a threat. If one is found we need to be able to isolate is and classify it, then work on creating a defense against this threat from future attacks. Lastly we need to be able to describe the attack for documentation.

To do this we will need to understand the flow of a malware attack as well as strategies and tools that can be used to analyses malware and defend against future attacks.

Malware Attack Flow

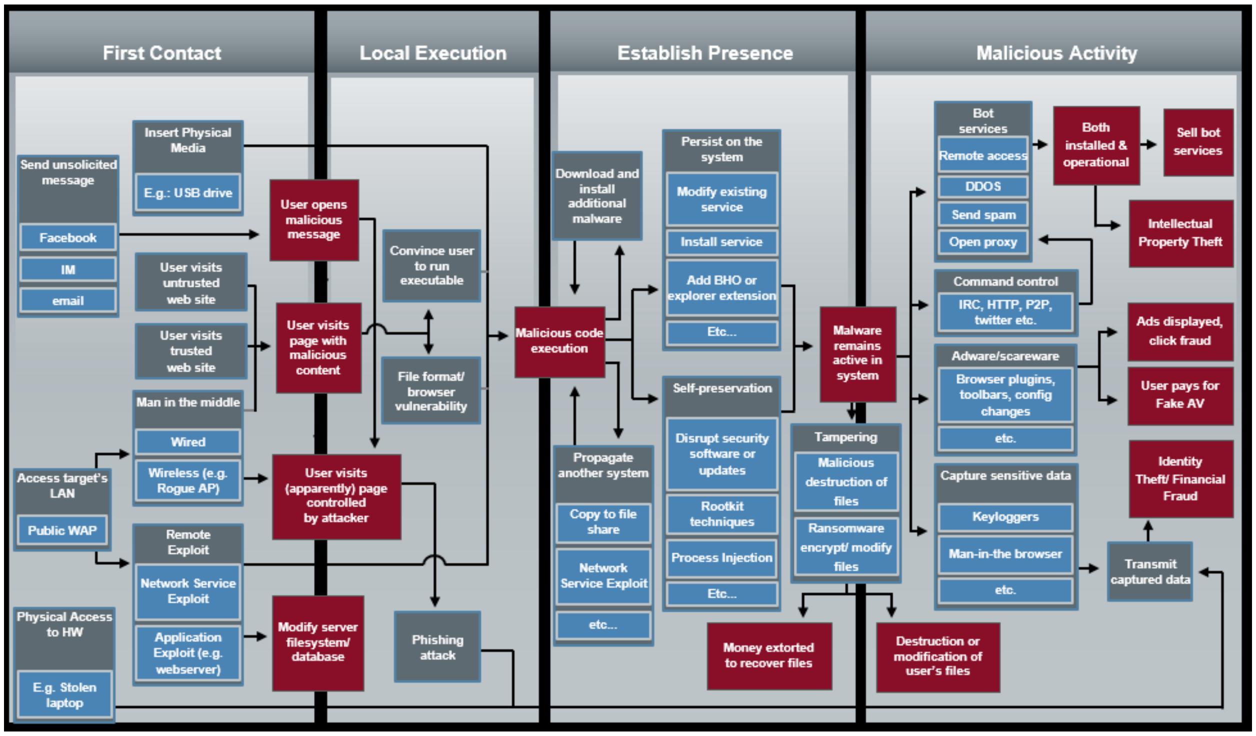

First Contact
Local Execution
Establish Presence
Malicious Activity
Send unsolicited message
Facebook
IM
email
Insert Physical Media
E.g.: USB drive
User visits untrusted web site
User visits trusted web site
Man in the middle
Wired
Wireless (e.g. Rogue AP)
Access target's LAN
Public WAP
Remote Exploit
Network Service Exploit
Application Exploit (e.g. webserver)
Physical Access to HW
E.g. Stolen laptop
User opens malicious message
User visits page with malicious content
User visits (apparently) page controlled by attacker
Modify server filesystem/ database
Convince user to run executable
File format/ browser vulnerability
Phishing attack
Malicious code execution
Download and install additional malware
Propagate another system
Copy to file share
Network Service Exploit
etc...
Persist on the system
Modify existing service
Install service
Add BHO or explorer extension
Etc...
Self-preservation
Disrupt security software or updates
Rootkit techniques
Process Injection
Etc...
Malware remains active in system
Tampering
Malicious destruction of files
Ransomware encrypt/ modify files
Money extorted to recover files
Destruction or modification of user's files
Bot services
Remote access
DDOS
Send spam
Open proxy
Command control
IRC, HTTP, P2P, twitter etc.
Adware/scareware
Browser plugins, toolbars, config changes
etc.
Capture sensitive data
Keyloggers
Man-in-the browser
etc.
Both installed & operational
Sell bot services
Intellectual Property Theft
Ads displayed, click fraud
User pays for Fake AV
Identity Theft/ Financial Fraud
Transmit captured data

Figure 1 - W3L1 - Malware Defenses slide 4

First Contact

The initial step in a malware attack is First Contact. This is how the malware gets to the users system and there are many entry points. A common type of first contact is unsolicited messaging through social media, instant messaging, or email. Another common way the malware can get to the user is though websites, often untrusted or cleverly disguised. Physical media like a USB drive or access to the hardware also allow for malware to be inserted.

Local Execution

Once the malware has made contact the next step is for it to execute.

Social engineering is a common strategy for getting malware executed. In this situation the malware needs to convince the user to install the malware. This is often by getting them to click on a link they believe is trust worthy, or installing a program they do not realize is an infected copycat. The main idea here is that the user does not realize they are actually launching or executing malware.

Another common way malware gets executed is through exploitation of vulnerabilities.   This is the case for browser exploit kits, and they often use vulnerabilities in things like PDF, Java, and plug ins to launch the attack.

Less common but still possible is abusing features like USB auto run or having physical access to the system to execute the malware.

Establish presence

After the malware is executed it has some work to do on the system. It needs to create a presence. It is important that the malware is able to blend in and appear legitimate to avoid users knowing it is there. Malware will often do things like use file names or paths that look correct, change time and date stamps, or have legitimate signatures to avoid notice. Malware also often has tools to hide like bootkits and rootkits.

Malware also needs to set up so that it can persist on the system. This may mean that it runs something on startup of the system, windows, or specific applications or that it schedules tasks. This often makes it so that removing just the visible part of the malware is not sufficient to eliminate the threat.

Malicious Activity

The last step of an attack is to do the actual a bad stuff the malware was set up to do. This could be harvesting information, setting up bot services, installing ad or scare ware or any number of other things depending on the goal. The malware will often also need to set up a way to phone home and relay gathered information or get instructions. Web requests and email are common ways for malware to set up those connections.

Defense Technologies and Strategies

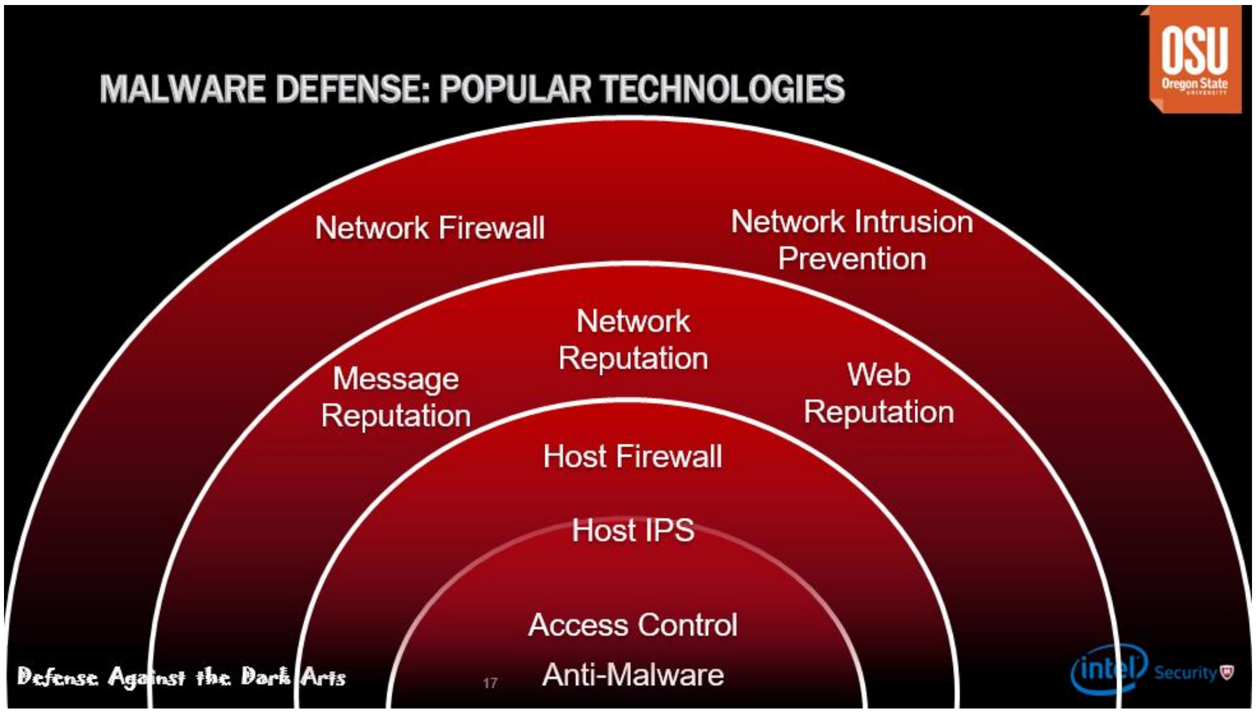


Figure 2 - W3L1 - Malware Defenses Slide 17

Features

Protecting against malware involves scanning and monitoring on many levels. At a base level it involves scanning files, registries, cookies, memory, or the cloud to see if anything fits known malicious profiles. That alone is not enough though. A good defense also requires checking scripts and packaged files which is more difficult.

In addition to scanning it is also important to set up firewalls that can set up a barrier. These can be utilized on several levels such as a network firewall or a host firewall and prevent from unknown or unwanted intrusion. It is important to balance the level of security with the impact to the user. a company may set a much stricter firewall than is necessary on a home computer system because they have much more sensitive information and are less worried about impacting the users.

Reputation also plays a huge role in essentially white or black listing entities. This can happen at a message, network, or web level and essentially screens out blacklisted entities or only allows whitelisted or trusted entities.

YARA

Once a threat has been identified, a way to defend against it is needed. This is often achieved by creating a signature that can be used to scan and identify that threat in the future.

Yara is a open source language that is often used for this purpose. It is a language written specifically to do pattern matching and logic that can be used to scan a file and determine if it should be flagged as dangerous. The language is simple which allows for rapid development and deployment of the signatures. It can be used for scanning files or memory. The magic here is making a good concise rule that will catch bad stuff and not have false positives.

INCLUDE SAMPLE YARA

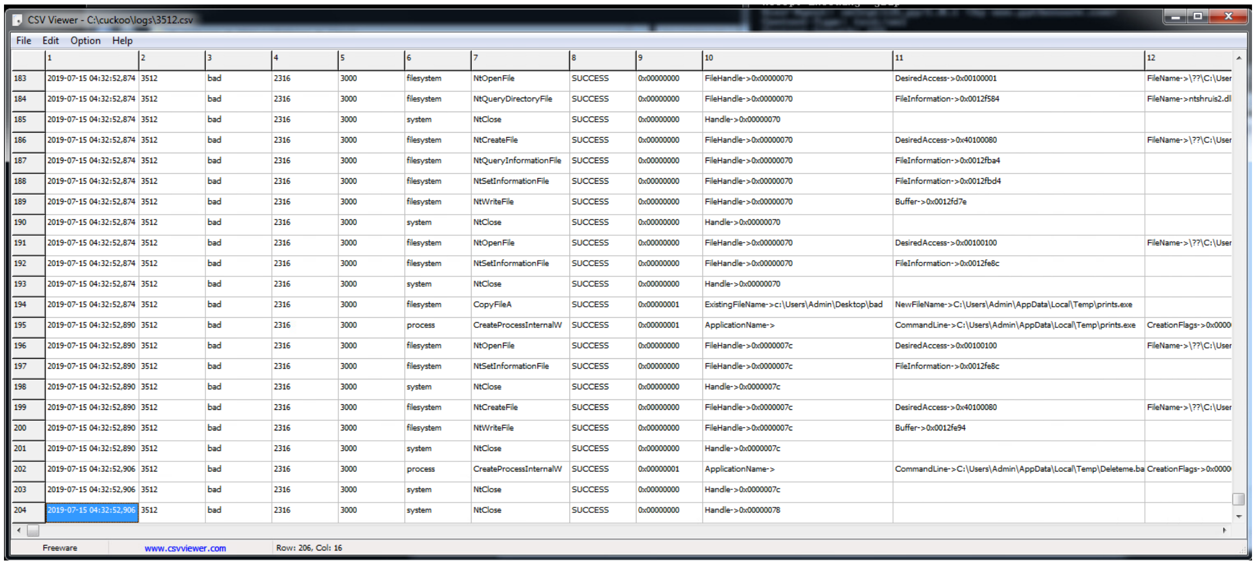
Automation

Writing custom rules and signatures with a tool like Yara or an in house scanner is extremely time consuming for the hundreds of thousands or malicious samples that need to be processed every day. For this reason many threats are analyzed and rules are generated through automation and machine learning. This helps overcome the scale of malware, but can provide consistency in approach.

There are some downsides of automation. Machines and automation do not understand context well, and may create rules that a human would understand is not useful or miss a rule that would be useful if the machine had understood the context. Many creators of malware are also writing code that can specifically tell things about systems that are used for these techniques. For instance malware is often aware if it is tried to run on a virtual machine, as those are often used in analysis, and will not run.

Because of the downsides, there is still a large need for human interaction in creating signatures, but automation plays an ever increasing role in handling the vast majority of issues.

Cuckoo is an example of a program that can be used for automation. It will launch an executable on a VM and then create a log of what the executable does. This provides a great way to look at what a file is doing and sort the information easily to determine if it is malicious.



	1	2	3	4	5	6	7	8	9	10	11	12
183	2019-07-15 04:32:52,874	3512	bad	2316	3000	filesystem	NtOpenFile	SUCCESS	0x00000000	FileHandle->0x00000070	DesiredAccess->0x00100001	FileName->\??\C:\User
184	2019-07-15 04:32:52,874	3512	bad	2316	3000	filesystem	NtQueryDirectoryFile	SUCCESS	0x00000000	FileHandle->0x00000070	FileInformation->0x0012f584	FileName->ntshruis2.dl
185	2019-07-15 04:32:52,874	3512	bad	2316	3000	system	NtClose	SUCCESS	0x00000000	Handle->0x00000070		
186	2019-07-15 04:32:52,874	3512	bad	2316	3000	filesystem	NtCreateFile	SUCCESS	0x00000000	FileHandle->0x00000070	DesiredAccess->0x40100080	FileName->\??\C:\User
187	2019-07-15 04:32:52,874	3512	bad	2316	3000	filesystem	NtQueryInformationFile	SUCCESS	0x00000000	FileHandle->0x00000070	FileInformation->0x0012fba4	
188	2019-07-15 04:32:52,874	3512	bad	2316	3000	filesystem	NtSetInformationFile	SUCCESS	0x00000000	FileHandle->0x00000070	FileInformation->0x0012fbd4	
189	2019-07-15 04:32:52,874	3512	bad	2316	3000	filesystem	NtWriteFile	SUCCESS	0x00000000	FileHandle->0x00000070	Buffer->0x0012fd7e	
190	2019-07-15 04:32:52,874	3512	bad	2316	3000	system	NtClose	SUCCESS	0x00000000	Handle->0x00000070		
191	2019-07-15 04:32:52,874	3512	bad	2316	3000	filesystem	NtOpenFile	SUCCESS	0x00000000	FileHandle->0x00000070	DesiredAccess->0x00100100	FileName->\??\C:\User
192	2019-07-15 04:32:52,874	3512	bad	2316	3000	filesystem	NtSetInformationFile	SUCCESS	0x00000000	FileHandle->0x00000070	FileInformation->0x0012fe8c	
193	2019-07-15 04:32:52,874	3512	bad	2316	3000	system	NtClose	SUCCESS	0x00000000	Handle->0x00000070		
194	2019-07-15 04:32:52,874	3512	bad	2316	3000	filesystem	CopyFileA	SUCCESS	0x00000001	ExistingFileName->c:\Users\Admin\Desktop\bad	NewFileName->C:\Users\Admin\AppData\Local\Temp\prints.exe	
195	2019-07-15 04:32:52,890	3512	bad	2316	3000	process	CreateProcessInternalW	SUCCESS	0x00000001	ApplicationName->	CommandLine->C:\Users\Admin\AppData\Local\Temp\prints.exe	CreationFlags->0x0000
196	2019-07-15 04:32:52,890	3512	bad	2316	3000	filesystem	NtOpenFile	SUCCESS	0x00000000	FileHandle->0x0000007c	DesiredAccess->0x00100100	FileName->\??\C:\User
197	2019-07-15 04:32:52,890	3512	bad	2316	3000	filesystem	NtSetInformationFile	SUCCESS	0x00000000	FileHandle->0x0000007c	FileInformation->0x0012fe8c	
198	2019-07-15 04:32:52,890	3512	bad	2316	3000	system	NtClose	SUCCESS	0x00000000	Handle->0x0000007c		
199	2019-07-15 04:32:52,890	3512	bad	2316	3000	filesystem	NtCreateFile	SUCCESS	0x00000000	FileHandle->0x0000007c	DesiredAccess->0x40100080	FileName->\??\C:\User
200	2019-07-15 04:32:52,890	3512	bad	2316	3000	filesystem	NtWriteFile	SUCCESS	0x00000000	FileHandle->0x0000007c	Buffer->0x0012fe94	
201	2019-07-15 04:32:52,890	3512	bad	2316	3000	system	NtClose	SUCCESS	0x00000000	Handle->0x0000007c		
202	2019-07-15 04:32:52,906	3512	bad	2316	3000	process	CreateProcessInternalW	SUCCESS	0x00000001	ApplicationName->	CommandLine->C:\Users\Admin\AppData\Local\Temp\Deleteme.ba	CreationFlags->0x0000
203	2019-07-15 04:32:52,906	3512	bad	2316	3000	system	NtClose	SUCCESS	0x00000000	Handle->0x0000007c		
204	2019-07-15 04:32:52,906	3512	bad	2316	3000	system	NtClose	SUCCESS	0x00000000	Handle->0x00000078		

Figure 3 - Cuckoo Log Example

BLOG

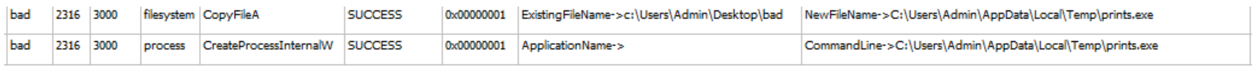
Put Blog Title here

By Kirsten Wollam 7/14/2019 10:15pm

Malware hash: A1874F714F7A15399B9FAE968180B303

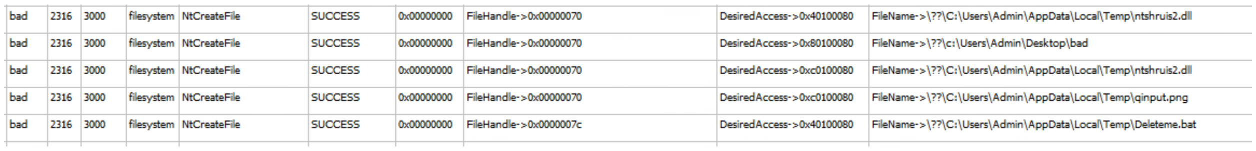
This file has been identified as malicious and clearly dangerous to your system. Immediate action should be taken.

Several key factors indicate that it is malicious. Firstly it creates a copy of its self and calls is prints.exe. There is no reason that a legitimate file would be creating a copy under a name such as this. It is a common technique for malware to use names that are *almost* correct. In this case a user might not think a file named print.exe is unusual so prints.exe blends in.



bad	2316	3000	filesystem	CopyFileA	SUCCESS	0x00000001	ExistingFileName->c:\Users\Admin\Desktop\bad	NewFileName->C:\Users\Admin\AppData\Local\Temp\prints.exe
bad	2316	3000	process	CreateProcessInternalW	SUCCESS	0x00000001	ApplicationName->	CommandLine->C:\Users\Admin\AppData\Local\Temp\prints.exe

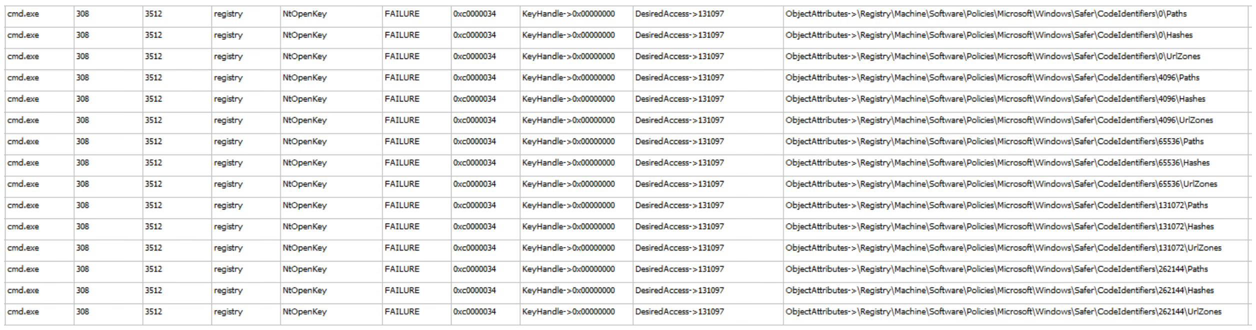
it also creates several other files that are quite suspicious including ntshruis2.dll, qinput.png, and Deleteme.bat. As before with the prints.exe, ntshuis.dll is a valid and normal file to have on your machine as a part of windows and this malware is trying to hide by having a similar name, ntshruis2.dll. input.png is also a plausible file name, so qinput.png might not seem odd.



bad	2316	3000	filesystem	NtCreateFile	SUCCESS	0x00000000	FileHandle->0x00000070	DesiredAccess->0x40100080	FileName->\??\C:\Users\Admin\AppData\Local\Temp\ntshruis2.dll
bad	2316	3000	filesystem	NtCreateFile	SUCCESS	0x00000000	FileHandle->0x00000070	DesiredAccess->0x80100080	FileName->\??\c:\Users\Admin\Desktop\bad
bad	2316	3000	filesystem	NtCreateFile	SUCCESS	0x00000000	FileHandle->0x00000070	DesiredAccess->0xc0100080	FileName->\??\C:\Users\Admin\AppData\Local\Temp\ntshruis2.dll
bad	2316	3000	filesystem	NtCreateFile	SUCCESS	0x00000000	FileHandle->0x00000070	DesiredAccess->0xc0100080	FileName->\??\C:\Users\Admin\AppData\Local\Temp\qinput.png
bad	2316	3000	filesystem	NtCreateFile	SUCCESS	0x00000000	FileHandle->0x0000007c	DesiredAccess->0x40100080	FileName->\??\C:\Users\Admin\AppData\Local\Temp\Deleteme.bat

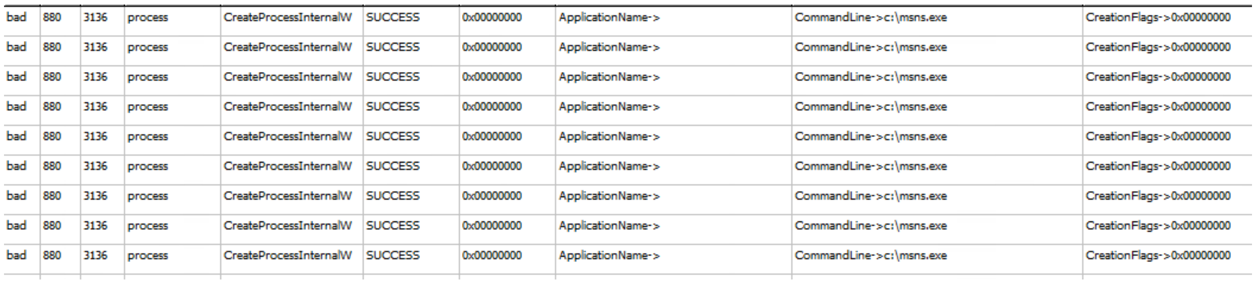
dleteme.bat seems to actually function to remove evidence of the malware on the system by deleting the original file. This helps it to hide and persist.

This malware also accesses many keys in the Safer\CodeIdentifiers files which is a way to try to avoid software restriction policies if they exist.



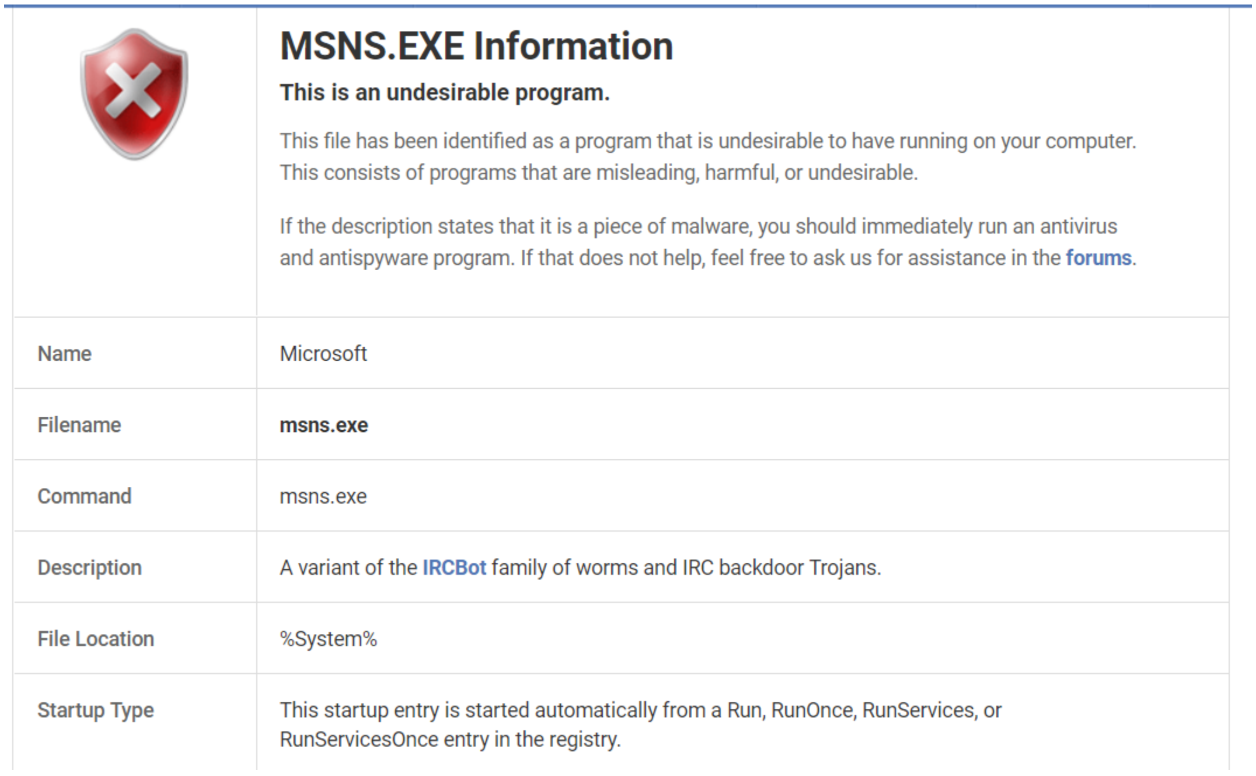
cmd.exe	308	3512	registry	NtOpenKey	FAILURE	0xc0000034	KeyHandle->0x00000000	DesiredAccess->131097	ObjectAttributes->\Registry\Machine\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers\0\Paths
cmd.exe	308	3512	registry	NtOpenKey	FAILURE	0xc0000034	KeyHandle->0x00000000	DesiredAccess->131097	ObjectAttributes->\Registry\Machine\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers\0\Hashes
cmd.exe	308	3512	registry	NtOpenKey	FAILURE	0xc0000034	KeyHandle->0x00000000	DesiredAccess->131097	ObjectAttributes->\Registry\Machine\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers\0\UrlZones
cmd.exe	308	3512	registry	NtOpenKey	FAILURE	0xc0000034	KeyHandle->0x00000000	DesiredAccess->131097	ObjectAttributes->\Registry\Machine\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers\4096\Paths
cmd.exe	308	3512	registry	NtOpenKey	FAILURE	0xc0000034	KeyHandle->0x00000000	DesiredAccess->131097	ObjectAttributes->\Registry\Machine\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers\4096\Hashes
cmd.exe	308	3512	registry	NtOpenKey	FAILURE	0xc0000034	KeyHandle->0x00000000	DesiredAccess->131097	ObjectAttributes->\Registry\Machine\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers\4096\UrlZones
cmd.exe	308	3512	registry	NtOpenKey	FAILURE	0xc0000034	KeyHandle->0x00000000	DesiredAccess->131097	ObjectAttributes->\Registry\Machine\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers\65536\Paths
cmd.exe	308	3512	registry	NtOpenKey	FAILURE	0xc0000034	KeyHandle->0x00000000	DesiredAccess->131097	ObjectAttributes->\Registry\Machine\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers\65536\Hashes
cmd.exe	308	3512	registry	NtOpenKey	FAILURE	0xc0000034	KeyHandle->0x00000000	DesiredAccess->131097	ObjectAttributes->\Registry\Machine\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers\65536\UrlZones
cmd.exe	308	3512	registry	NtOpenKey	FAILURE	0xc0000034	KeyHandle->0x00000000	DesiredAccess->131097	ObjectAttributes->\Registry\Machine\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers\131072\Paths
cmd.exe	308	3512	registry	NtOpenKey	FAILURE	0xc0000034	KeyHandle->0x00000000	DesiredAccess->131097	ObjectAttributes->\Registry\Machine\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers\131072\Hashes
cmd.exe	308	3512	registry	NtOpenKey	FAILURE	0xc0000034	KeyHandle->0x00000000	DesiredAccess->131097	ObjectAttributes->\Registry\Machine\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers\131072\UrlZones
cmd.exe	308	3512	registry	NtOpenKey	FAILURE	0xc0000034	KeyHandle->0x00000000	DesiredAccess->131097	ObjectAttributes->\Registry\Machine\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers\262144\Paths
cmd.exe	308	3512	registry	NtOpenKey	FAILURE	0xc0000034	KeyHandle->0x00000000	DesiredAccess->131097	ObjectAttributes->\Registry\Machine\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers\262144\Hashes
cmd.exe	308	3512	registry	NtOpenKey	FAILURE	0xc0000034	KeyHandle->0x00000000	DesiredAccess->131097	ObjectAttributes->\Registry\Machine\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers\262144\UrlZones

Additionally the Cuckoo log shows that over 400 processes were created on msns.exe.



bad	880	3136	process	CreateProcessInternalW	SUCCESS	0x00000000	ApplicationName->	CommandLine->c:\msns.exe	CreationFlags->0x00000000
bad	880	3136	process	CreateProcessInternalW	SUCCESS	0x00000000	ApplicationName->	CommandLine->c:\msns.exe	CreationFlags->0x00000000
bad	880	3136	process	CreateProcessInternalW	SUCCESS	0x00000000	ApplicationName->	CommandLine->c:\msns.exe	CreationFlags->0x00000000
bad	880	3136	process	CreateProcessInternalW	SUCCESS	0x00000000	ApplicationName->	CommandLine->c:\msns.exe	CreationFlags->0x00000000
bad	880	3136	process	CreateProcessInternalW	SUCCESS	0x00000000	ApplicationName->	CommandLine->c:\msns.exe	CreationFlags->0x00000000
bad	880	3136	process	CreateProcessInternalW	SUCCESS	0x00000000	ApplicationName->	CommandLine->c:\msns.exe	CreationFlags->0x00000000
bad	880	3136	process	CreateProcessInternalW	SUCCESS	0x00000000	ApplicationName->	CommandLine->c:\msns.exe	CreationFlags->0x00000000
bad	880	3136	process	CreateProcessInternalW	SUCCESS	0x00000000	ApplicationName->	CommandLine->c:\msns.exe	CreationFlags->0x00000000
bad	880	3136	process	CreateProcessInternalW	SUCCESS	0x00000000	ApplicationName->	CommandLine->c:\msns.exe	CreationFlags->0x00000000

This is a known undesirable program, associated with the IRCBot family of worms.



MSNS.EXE Information

This is an undesirable program.

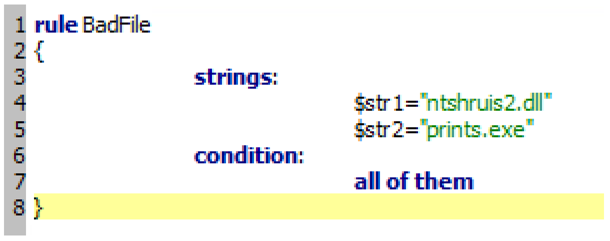
This file has been identified as a program that is undesirable to have running on your computer. This consists of programs that are misleading, harmful, or undesirable.

If the description states that it is a piece of malware, you should immediately run an antivirus and antispyware program. If that does not help, feel free to ask us for assistance in the **forums**.

Name	Microsoft
Filename	**msns.exe**
Command	msns.exe
Description	A variant of the **IRCBot** family of worms and IRC backdoor Trojans.
File Location	%System%
Startup Type	This startup entry is started automatically from a Run, RunOnce, RunServices, or RunServicesOnce entry in the registry.

For all of these reasons we believe that you should remove this file and the ramifications from your system.

The following Yara Signature should capture it in a scan and can be immediately implemented:



```
rule BadFile
{
                strings:
                                $str1="ntshruis2.dll"
                                $str2="prints.exe"
                condition:
                                all of them
}
```