Review of Prevention Methods in Living off the Land Attacks

Priya Daniel[1], Nutan Sawant[2]

[1]Asst. Prof. of Department of Information Technology, S.I.E.S.(Nerul) College of Arts, Science and Commerce, Maharashtra, India.
[2]Asst. Prof. of Department of Information Technology, S.I.E.S.(Nerul) College of Arts, Science and Commerce, Maharashtra, India.

Abstract - *Living off the Land attacks have become a convenient method to gain control of a machine using existing software or applications. As legitimate applications are used to carry out these attacks, with little or no involvement of malware, any suspicious activity is difficult to detect by traditional anti malware applications. In recent years, LOTL attacks have become very common due to ease of execution, low cost and higher stealth benefit as the applications residing on the target machine itself are used against it. In this paper, the different methods used to prevent these attacks are explored to understand their effectiveness in various LOTL Attack scenarios on Windows machines.*

Key Words: Living off the Land attack, malware, Windows

1. INTRODUCTION

Living off the land attacks are executed by using built in applications and tools present on the target machine which are most often used by system administrators to carry out daily administrative tasks to ensure the smooth functioning of the system and/or associated devices. It may also include writing malicious code directly to memory for execution without writing to disk, in which case it is called a file less malware attack. Sometimes these attacks act as a pre cursor to enable the download, installation and implementation of the real malicious file which will be compromising the system.

Since traditional anti malware tools make use of malware signatures and heuristics from previously known malwares to detect any sort of malicious activity, it is difficult to detect LOTL attacks with these tools as there is no malware file to detect. To understand LOTL attacks, tools and techniques which study behaviors of the built in Windows applications, monitor the network traffic and analyze the data in memory will prove more efficient and also help detect such attacks. This research paper aims to analyse some of these techniques to develop prevention approaches to be used in the event of an LOTL attack.

1.1 LOLBAS project

LOLBAS is the acronym for Living off the Land Binaries and Scripts. The LOLBAS project documents every binary, script and library of the Windows OS that can be used for purposes other than what it was originally created for and thus can be used to carry out Living off the land attacks [3].

1.2 MITRE ATT&CK

This is a useful repository of observed techniques that are implemented to carry out malicious activity on the target system. The techniques are classified into various categories depending on the purpose of the malicious activity like reconnaissance, initial access and execution. For each technique, the details regarding how it is implemented, mitigation steps if any and possible ways of detection are prescribed [4].

2. LITERATURE SURVEY

Survey of techniques used for Living Off the Land attacks.

2.1. Dual Use Tools

The LOLBAS project lists numerous tools which can be alternately used for LOTL attacks apart from their legitimate purposes. For example, AppInstaller.exe is a tool used for installation of AppX/MSIX applications on Windows 10.

```
start ms-appinstaller://?source=<url>
```

AppInstaller.exe is spawned by the default handler for the URI which is called using the start command; it attempts to load/install a package from the URL specified in source attribute. This can allow an attacker to download arbitrary files like malware without being detected[3].

Pcwrun.exe is a Program Compatibility Wizard. Originally created to troubleshoot compatibility issues, it allows users to open and execute arbitrary files having .exe extension even if they are not fully compatible with the Windows OS[3].

The following command opens and executes the target .EXE file with the Program Compatibility Wizard.

```
Pcwrun.exe c:\temp\beacon.exe
```

This can also be used for proxy execution of malicious files.

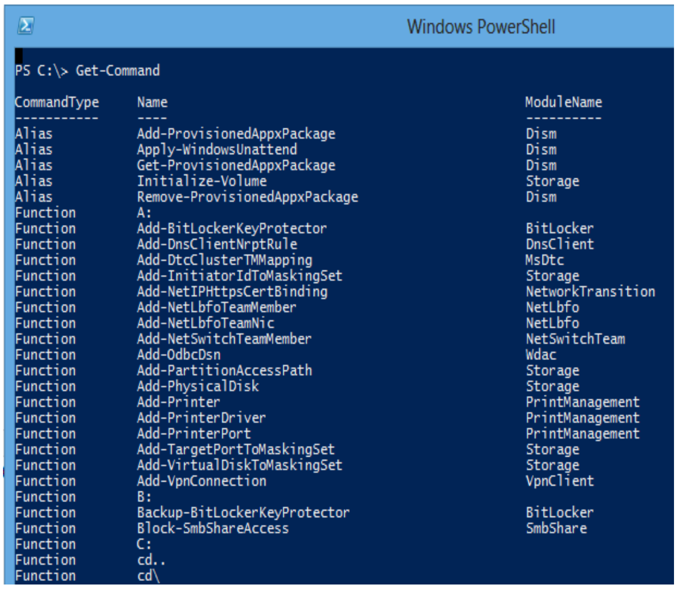
2.1.1 Windows Management Instrumentation

Windows Management Instrumentation or WMI has been gaining popularity amongst attackers for its ability to perform system reconnaissance, Anti-Virus and Virtual Machine detection, code execution while supporting lateral movement and persistence for malicious payload. It was one of the components in the exploits used by the infamous Stuxnet malware [1].

For example, one of the methods used by attackers for performing lateral movement is the static Create method in the Win32_Process class. WMI also provides support for users to register event handlers upon the creation, modification, or deletion of any WMI object instance [1].

2.1.2. PowerShell

PowerShell is a .NET based framework which provides a command shell with its own scripting language to automate and manage administrative tasks [2]. To see the commands available on a Windows PowerShell, use Get-Command at the PowerShell command line.



```
PS C:\> Get-Command

CommandType     Name                                               ModuleName
-----------     ----                                               ----------
Alias           Add-ProvisionedAppxPackage                         Dism
Alias           Apply-WindowsUnattend                              Dism
Alias           Get-ProvisionedAppxPackage                         Dism
Alias           Initialize-Volume                                  Storage
Alias           Remove-ProvisionedAppxPackage                      Dism
Function        A:
Function        Add-BitLockerKeyProtector                          BitLocker
Function        Add-DnsClientNrptRule                              DnsClient
Function        Add-DtcClusterTMMapping                            MsDtc
Function        Add-InitiatorIdToMaskingSet                        Storage
Function        Add-NetIPHttpsCertBinding                          NetworkTransition
Function        Add-NetLbfoTeamMember                              NetLbfo
Function        Add-NetLbfoTeamNic                                 NetLbfo
Function        Add-NetSwitchTeamMember                            NetSwitchTeam
Function        Add-OdbcDsn                                        Wdac
Function        Add-PartitionAccessPath                            Storage
Function        Add-PhysicalDisk                                   Storage
Function        Add-Printer                                        PrintManagement
Function        Add-PrinterDriver                                  PrintManagement
Function        Add-PrinterPort                                    PrintManagement
Function        Add-TargetPortToMaskingSet                         Storage
Function        Add-VirtualDiskToMaskingSet                        Storage
Function        Add-VpnConnection                                  VpnClient
Function        B:
Function        Backup-BitLockerKeyProtector                       BitLocker
Function        Block-SmbShareAccess                               SmbShare
Function        C:
Function        cd..
Function        cd\
```

Fig -1. PowerShell Command Line

It will list thousands of cmdlets as shown in fig. 1.which can be used for carrying out any task on the Windows Machine. The PowerShell provides a rich set of functionalities which can be implemented using the appropriate cmdlets some of which can be used for interacting with the WMI.

Added to this, PowerShell scripts are poorly recorded in the Windows Event logs making it difficult to conduct forensic analysis on malware executed using PowerShell scripts while by-passing whitelisting applications. Also, it allows running malicious code directly from memory making it stealthy. These are a few features which attract adversaries to utilize the PowerShell to orchestrate full blown cyber-attacks[2].

2.2. Memory Based Attacks

In this type of attack, the malicious payload is executed directly from memory for example memory worms like Code Red and SQL Slammer which exploited the vulnerabilities of the Windows Services for execution [5].

If shell code execution is not possible, PowerShell can be used to download and execute the payload using a Web Client. But this method requires power shell to be pre-installed on the target machine and that remote access to it is enabled. For this, insider credentials preferably one with administrator privileges need to be used [5].

2.3. Persistence Methods

Once the targeted system is compromised, the next objective of the attacker is to remain undetected as long as possible to continue exploiting the vulnerabilities of the system. There are multiple facilities in Windows OS which can ensure this for example the malware can place an entry in the registry run sub key which points to the malicious code so that it can execute when the systems starts. Or the entire payload can be stored inside Windows Registry as was done by the Poweliks file less malware [5].

An attacker can use WMI to trigger the execution of a malicious script which is usually a PowerShell script. Creating Scheduled Tasks which will execute commands at specific trigger moments on a local or remote system is also a viable option for an attacker [5].

For creating a trigger, the New-ScheduledTaskTrigger cmdlet can be used. The command below creates a trigger to run daily at 7 AM.

```
# Create a new trigger (Daily at 7 AM)

$taskTrigger = New-ScheduledTaskTrigger -Daily -At 7AM

$tasktrigger
```

3. PREVENTION METHODS

As far as LOTL attacks are concerned, traditional anti-virus or anti malware tools do not work in detecting these attacks as they rely heavily on malware signatures or past behaviors of known malware which is also called heuristics.

Static and dynamic malware analysis techniques also prove futile in discovering any useful forensic data as there is no real malware file written to disk which can be analyzed for traces of any malicious activity.

Although, there are certain tools, techniques and best practices that are followed by organizations to safeguard their Windows Systems. Below is an analysis of such common tools and techniques.

3.1. Microsoft Defender Antivirus

Formerly known as Windows Defender, it identifies malicious activities by analyzing the behavior of legitimate built-in applications. If the program behaves in a way which is flagged as malicious by the Microsoft Defender, it will block the action. If a malicious payload execution which involves built in applications can masquerade as being safe or legal, the Microsoft Defender would allow it to execute [6].

3.2. AppLocker

This is an application whitelisting technology which facilitates restriction of applications by defining rules based on file attributes like the application or file name, path of execution, publisher, hash etc. These rules can be applied to specific user groups or individual users while also implementing exceptions if required [6]. But if an attacker devises techniques to execute malicious payload using a whitelisted application within the scope of the rules defined over it by the AppLocker , such action will not be detected by it. Also, applying rules to .dll files which turn out to be malicious in certain cases can hinder the operating system performance.

3.3. Memory Forensics

LOTL attacks hardly leave any traces of malicious activity due to its file less nature. Therefore, analyzing the volatile memory or RAM on real time basis can prove effective in detecting malicious payload execution. This can be achieved by creating memory dumps which are image captures of the data present in RAM at a specific instant.

When carefully analyzed, this data can provide useful insight into the run time system behavior like recently executed commands, network activity, IPC messages etc. to detect suspicious activity. Even encrypted malware has to be decrypted to be able to run in memory. Scrutinizing Memory Dumps at regular intervals can aide timely prevention or at least mitigation of malicious attacks.

But the same Memory Dump can be used by attackers to gain sensitive data like user names and passwords, chat messages, email content, encryption keys if they gain remote access. If the memory dumps are being transferred to other devices over the network for analysis, it is susceptible to eaves dropping or traffic sniffing. Encryption of these files can prevent this to some extent while in transit over the network.

3.4. Monitoring the behavior of Dual Use Tools

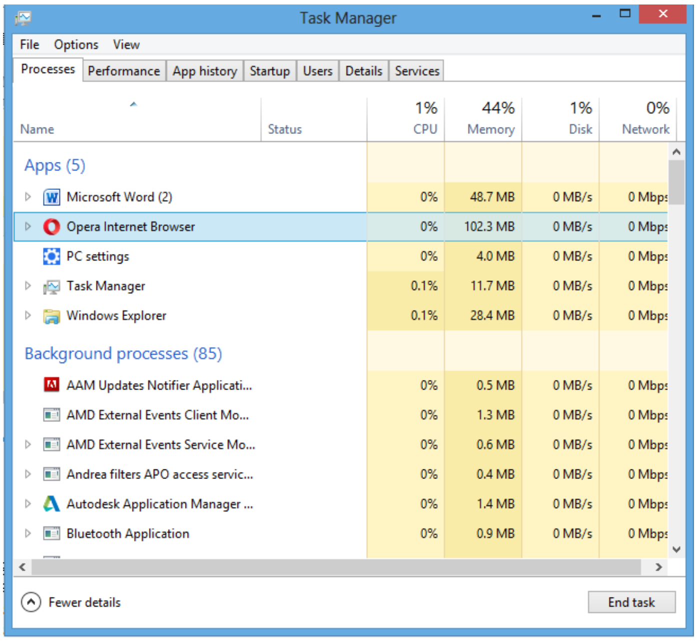
Tools like PowerShell can be disabled if not currently required. If this is not an option, the execution of scripts can be restricted to only Administrator accounts. Apart from this, creating alerts or triggers for every time a PowerShell script is executed can be defined for the Administrator to decide whether to allow execution or not. Enable logging for PowerShell so that in case of an incident, the logs can be examined to discover any sort of suspicious activity.

But if the attacker uses PowerShell Modules which are whitelisted, they can easily evade any detection by anti-virus tools. Also, if the attacker gains administrator level access privileges, they will be able to create their own execution policies that they deem fit to be able to compromise the target machine.

Also, execution of tools like NetCat, Mimikatz which are used by security professionals should be monitored as these can be used by adversaries to skim sensitive information. This can be done by analyzing memory dumps or scanning Windows Event Logs under the Sysmon log. The drawback here is that, these tools are used by security admins and criminals in the same manner which makes detection of any suspicious action extremely difficult.

3.5. Task Manager

At the most basic level, a user can open the Task Manager to view information about running processes, network activity, memory usage, computer performance, services etc. Monitoring the task manager can help find suspicious payload execution to some extent. For example, if a legitimate process or application is using more memory than usual or, if any application seems to be executing on its own without user action or starts running at startup without permission, it is an indicator for a malicious payload execution. This again depends on the expertise of the user to decide whether the excess memory usage or program execution is justified. If established as suspicious, the user can choose to end the process by clicking “End Task” after selecting the process from the list of running applications as shown in Fig -2.



Name	Status	1% CPU	44% Memory	1% Disk	0% Network
Apps (5)					
Microsoft Word (2)		0%	48.7 MB	0 MB/s	0 Mbps
Opera Internet Browser		0%	102.3 MB	0 MB/s	0 Mbps
PC settings		0%	4.0 MB	0 MB/s	0 Mbps
Task Manager		0.1%	11.7 MB	0 MB/s	0 Mbps
Windows Explorer		0.1%	28.4 MB	0 MB/s	0 Mbps
Background processes (85)					
AAM Updates Notifier Applicati...		0%	0.5 MB	0 MB/s	0 Mbps
AMD External Events Client Mo...		0%	1.3 MB	0 MB/s	0 Mbps
AMD External Events Service Mo...		0%	0.6 MB	0 MB/s	0 Mbps
Andrea filters APO access servic...		0%	0.4 MB	0 MB/s	0 Mbps
Autodesk Application Manager ...		0%	1.4 MB	0 MB/s	0 Mbps
Bluetooth Application		0%	0.9 MB	0 MB/s	0 Mbps

Fig -2. Task Manager

4. CONCLUSIONS

LOTL attacks are fast becoming popular among cyber attackers due to ease of implementation, high stealth and the lack of need to create malware files. Since, these attacks are able to evade detection by traditional anti malware technologies, methods that rely on behavioral analysis of built in programs need to be devised. This kind of analysis has to be extended to files even if they are whitelisted or seem legitimate. Memory forensics is useful for identifying key information related to any malicious incident and can help understand behaviors of recently executed processes as part of the attack. Endpoint solutions like SIEM can provide the necessary logging functionality to accumulate data for behavioral analysis while also managing any security threats.

REFERENCES

[1] Matt Graeber, Black Hat 2015, Abusing Windows Management Instrumentation (WMI) to Build a Persistent, Asyncronous, and Fileless Backdoor.

[2] Symantec, Increased use of powershell in attacks,2016.

[3] The LOLBAS project, Website URL https://lolbas-project.github.io

[4] MITRE ATT&CK . Website: https://attack.mitre.org.

[5] Himanshu Anand, Candid Wueest, Symantec, ISTR, Living off the land and fileless attack techniques,2017.

[6] David Brown, Preventing Living off the Land Attacks, 2020.