Task 12 Sandboxes Day 6: If I can't find a nice malware to use, I'm not going.

16.12.2024

Nikhil Kumar

Learning Objectives

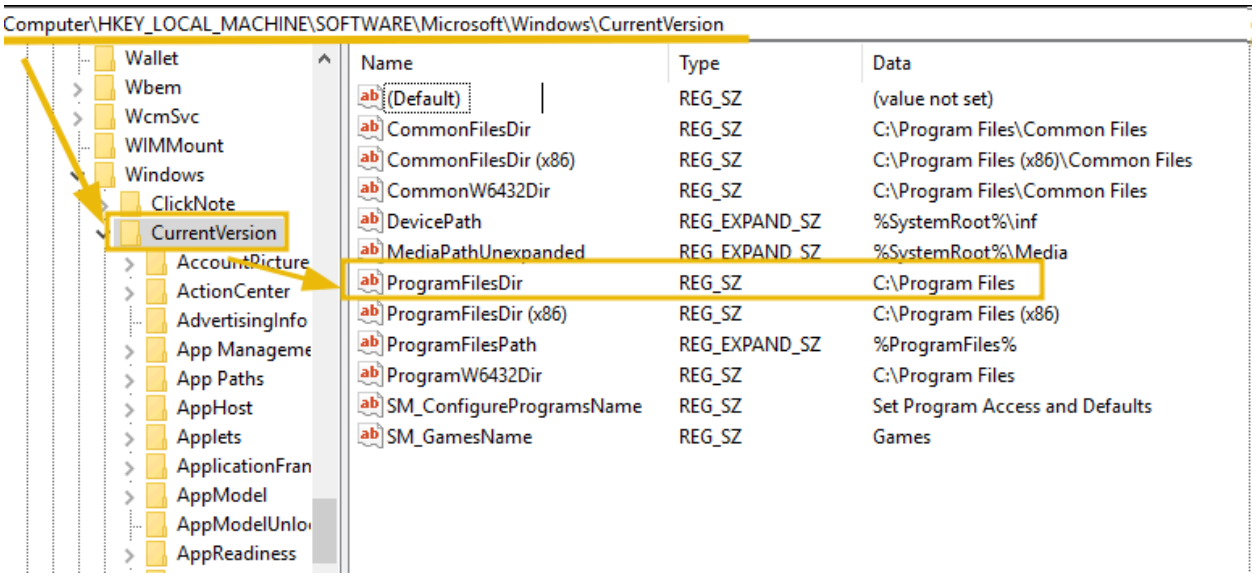
1. ***Analyze malware behavior using sandbox tools.***
2. ***Explore YARA rules to detect malicious patterns.***
3. ***Learn various malware evasion techniques.***
4. ***Implement evasion techniques to bypass YARA rule detection.***

Understanding Sandboxes

- Sandboxes provide an isolated environment to analyze malware behavior without affecting external systems. These environments are equipped with monitoring tools that record and analyze code execution.

Detecting Sandboxes

- Malware often employs detection techniques to identify sandbox environments. For instance, checking the presence of the directory `C:\Program Files` via the registry path `HKLM\Software\Microsoft\Windows\CurrentVersion` can indicate a typical versus virtualized environment.

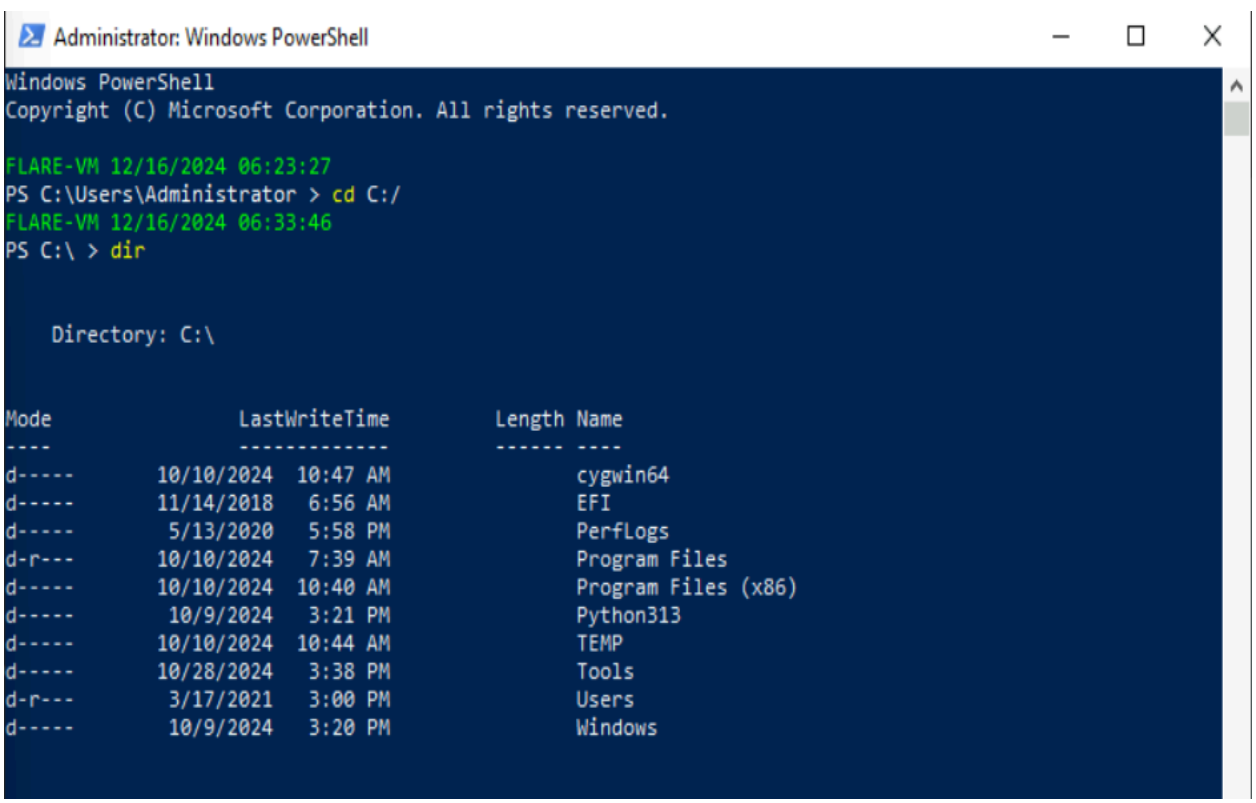


Code Example in C:

```
void registryCheck() {
    const char *registryPath = "HKLM\\Software\\Microsoft\\Windows\\CurrentVersion";
    const char *valueName = "ProgramFilesDir";
```

```
    char command[512];
    snprintf(command, sizeof(command), "reg query \"%s\" /v %s", registryPath, valueName);
    int result = system(command);
    if (result == 0) {
        printf("Registry query executed successfully.\n");
    } else {
        fprintf(stderr, "Failed to execute registry query.\n");
    }
}
int main() {
    registryCheck();
    return 0;
}
```

This code checks the registry for the **`ProgramFilesDir`** path to identify the environment.



```
FLARE-VM 12/16/2024 06:36:30
PS C:\ > cd .\Tools
FLARE-VM 12/16/2024 06:37:04
PS C:\Tools > ls


    Directory: C:\Tools


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        10/8/2024   9:14 AM                Explorer Suite
d-----        10/8/2024   9:15 AM                FLOSS
d-----       10/28/2024  10:48 PM                Malware
d-----       10/11/2024   2:38 PM                MinGW
d-----        10/8/2024   9:17 AM                pestudio
d-----        10/8/2024   9:17 AM                Regshot-x64-Unicode
d-----        10/8/2024   9:18 AM                Situational Awareness BOF
d-----        10/8/2024   8:53 AM                SysinternalsSuite
d-----        10/8/2024   9:19 AM                x64dbg
d-----        10/9/2024  12:25 PM                YARARULES
-a----       10/28/2024   9:21 AM           5817 JingleBells.ps1
-a----       10/28/2024  10:46 PM              0 YaraMatches.txt
```

Administrator: Windows PowerShell

```
PS C:\Tools > get-Content .\JingleBells.ps1
# Define the YARA rule path
$yaraRulePath = "C:\Tools\YARARULES\CheckRegCommand.yar"
# Define the path to the YARA executable
$yaraExecutable = "C:\ProgramData\chocolatey\lib\yara\tools\yara64.exe"
$logFilePath = "C:\Tools\YaraMatches.txt"


# Function to log event data to a file
function Log-EventDataToFile {
    param (
        [string]$commandLine,
        [string]$yaraResult,
        [string]$eventId,
        [string]$eventTimeCreated,
        [string]$eventRecordID
    )

    # Prepare log entry
    $logEntry = "Event Time: $eventTimeCreated`r`nEvent ID: $eventId`r`nEvent Record ID: $eventRecordID`
r`nCommand Line: $commandLine`r`nYARA Result: $yaraResult`r`n"
    $logEntry += "-------------------------------------`r`n"

    # Debugging output to ensure logging function is called
    Write-Host "Logging to file: $logFilePath"
    Write-Host $logEntry

    # Append the log entry to the file
    Add-Content -Path $logFilePath -Value $logEntry
}


# Function to run YARA on the command line and log result only if a match is found
function Run-YaraRule {
    param (
        [string]$commandLine,
        [string]$eventId,
        [string]$eventTimeCreated,
        [string]$eventRecordId

    )

    # Create a temporary file to store the command line for YARA processing
    $tempFile = [System.IO.Path]::GetTempFileName()
    try {
```

Detecting Malware with YARA Rules

YARA rules allow analysts to define patterns for detecting malicious activity. Here's an example rule:

Example YARA Rule:

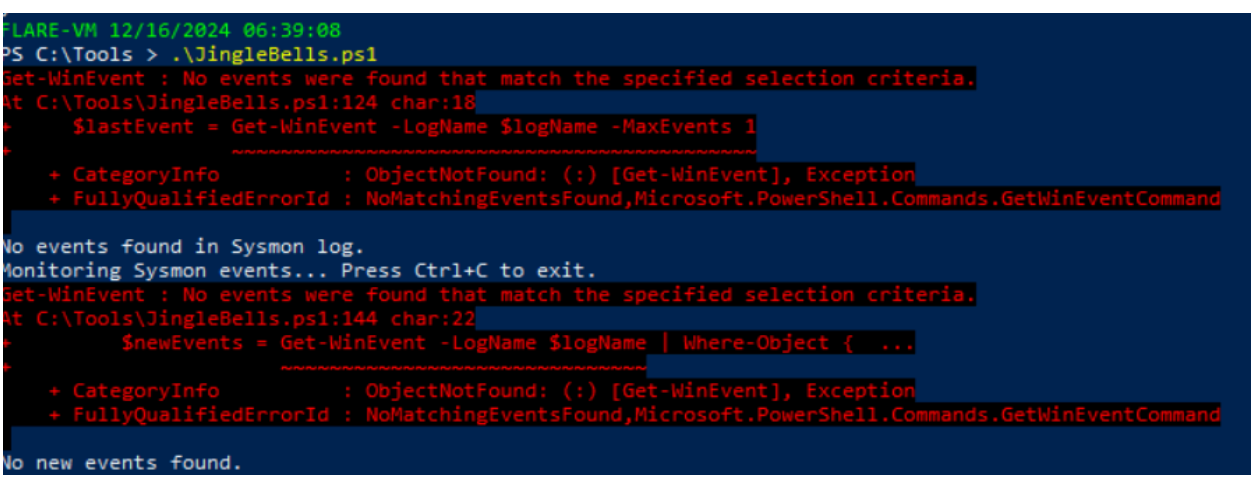
```
rule SANDBOXDETECTED {
  meta:
    description = "Detects the sandbox by querying the registry key for Program
Path"
    author = "TryHackMe"
    date = "2024-10-08"
    version = "1.1"
  strings:
    $cmd = "Software\\Microsoft\\Windows\\CurrentVersion\" /v ProgramFilesDir"
nocase
  condition:
    $cmd
}
```

- **Strings Section**: Defines patterns to search for (e.g., `$cmd`).
- **Condition Section**: Specifies conditions to trigger the rule (e.g., presence of `$cmd`).

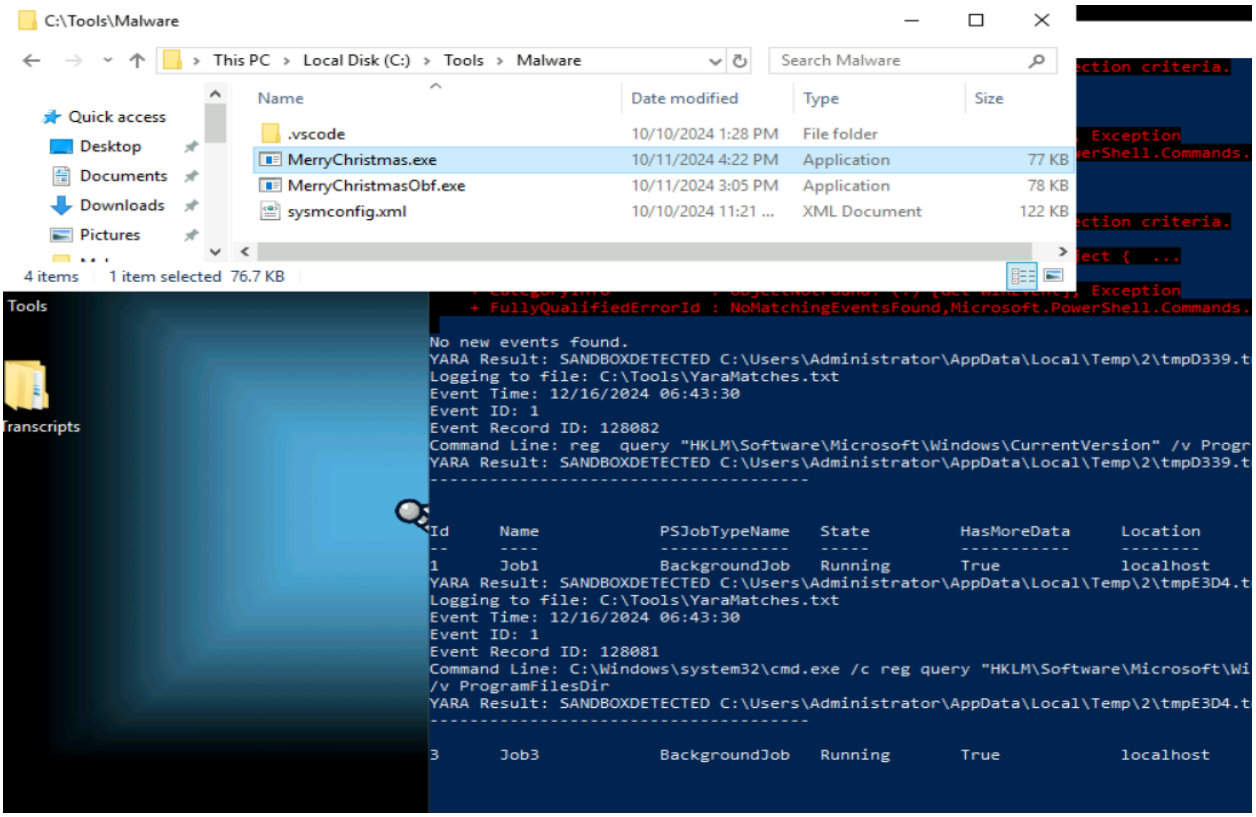
YARA rules can be executed via scripts to monitor logs. Example command:

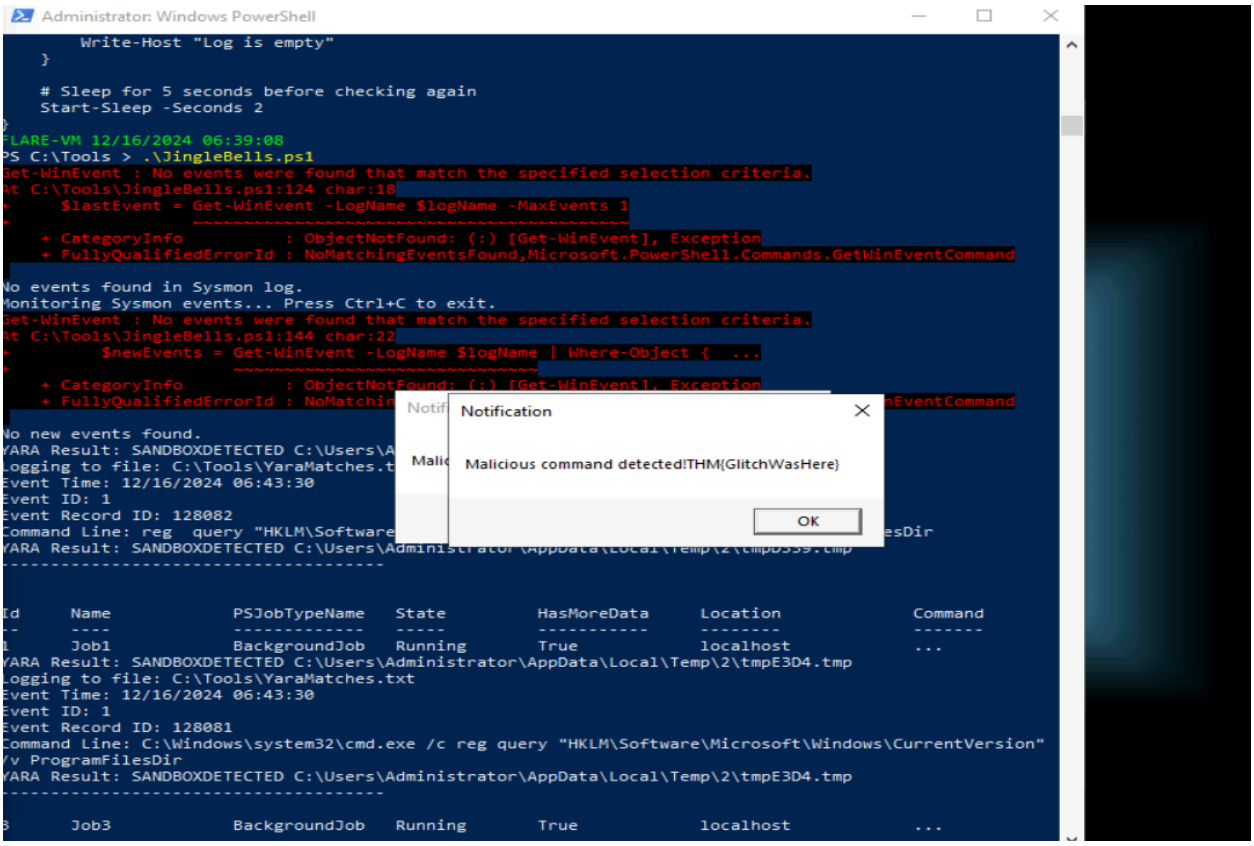
```
PS C:\Tools> .\JingleBells.ps1
```

This script detects registry queries and logs matches in **`YaraMatches.txt`**.



```
FLARE-VM 12/16/2024 06:39:08
PS C:\Tools > .\JingleBells.ps1
Get-WinEvent : No events were found that match the specified selection criteria.
At C:\Tools\JingleBells.ps1:124 char:18
+     $lastEvent = Get-WinEvent -LogName $logName -MaxEvents 1
+                  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : ObjectNotFound: (:) [Get-WinEvent], Exception
    + FullyQualifiedErrorId : NoMatchingEventsFound,Microsoft.PowerShell.Commands.GetWinEventCommand

No events found in Sysmon log.
Monitoring Sysmon events... Press Ctrl+C to exit.
Get-WinEvent : No events were found that match the specified selection criteria.
At C:\Tools\JingleBells.ps1:144 char:22
+         $newEvents = Get-WinEvent -LogName $logName | Where-Object {  ...
+                      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : ObjectNotFound: (:) [Get-WinEvent], Exception
    + FullyQualifiedErrorId : NoMatchingEventsFound,Microsoft.PowerShell.Commands.GetWinEventCommand

No new events found.
```


C:\Tools\Malware
This PC › Local Disk (C:) › Tools › Malware
Search Malware
Quick access
Desktop
Documents
Downloads
Pictures
Name
Date modified
Type
Size
.vscode
10/10/2024 1:28 PM
File folder
MerryChristmas.exe
10/11/2024 4:22 PM
Application
77 KB
MerryChristmasObf.exe
10/11/2024 3:05 PM
Application
78 KB
sysmconfig.xml
10/10/2024 11:21 ...
XML Document
122 KB
4 items
1 item selected 76.7 KB
Tools
Transcripts
No new events found.
YARA Result: SANDBOXDETECTED C:\Users\Administrator\AppData\Local\Temp\2\tmpD339.t
Logging to file: C:\Tools\YaraMatches.txt
Event Time: 12/16/2024 06:43:30
Event ID: 1
Event Record ID: 128082
Command Line: reg query "HKLM\Software\Microsoft\Windows\CurrentVersion" /v Progr
YARA Result: SANDBOXDETECTED C:\Users\Administrator\AppData\Local\Temp\2\tmpD339.t
Id Name PSJobTypeName State HasMoreData Location
1 Job1 BackgroundJob Running True localhost
YARA Result: SANDBOXDETECTED C:\Users\Administrator\AppData\Local\Temp\2\tmpE3D4.t
Logging to file: C:\Tools\YaraMatches.txt
Event Time: 12/16/2024 06:43:30
Event ID: 1
Event Record ID: 128081
Command Line: C:\Windows\system32\cmd.exe /c reg query "HKLM\Software\Microsoft\Wi
/v ProgramFilesDir
YARA Result: SANDBOXDETECTED C:\Users\Administrator\AppData\Local\Temp\2\tmpE3D4.t
3 Job3 BackgroundJob Running True localhost


Administrator: Windows PowerShell
Write-Host "Log is empty"
Sleep for 5 seconds before checking again
Start-Sleep -Seconds 2
FLARE-VM 12/16/2024 06:39:08
PS C:\Tools > .\JingleBells.ps1
Get-WinEvent : No events were found that match the specified selection criteria.
At C:\Tools\JingleBells.ps1:124 char:18
$lastEvent = Get-WinEvent -LogName $logName -MaxEvents 1
+ CategoryInfo : ObjectNotFound: (:) [Get-WinEvent], Exception
+ FullyQualifiedErrorId : NoMatchingEventsFound,Microsoft.PowerShell.Commands.GetWinEventCommand
No events found in Sysmon log.
Monitoring Sysmon events... Press Ctrl+C to exit.
Get-WinEvent : No events were found that match the specified selection criteria.
At C:\Tools\JingleBells.ps1:144 char:22
$newEvents = Get-WinEvent -LogName $logName | Where-Object { ...
+ CategoryInfo : ObjectNotFound: (:) [Get-WinEvent], Exception
Notification
Malicious command detected!THM{GlitchWasHere}
OK
No new events found.
Logging to file: C:\Tools\YaraMatches.t
Event Time: 12/16/2024 06:43:30
Event ID: 1
Event Record ID: 128082
Command Line: reg query "HKLM\Software
Id Name PSJobTypeName State HasMoreData Location Command
1 Job1 BackgroundJob Running True localhost ...
YARA Result: SANDBOXDETECTED C:\Users\Administrator\AppData\Local\Temp\2\tmpE3D4.tmp
Logging to file: C:\Tools\YaraMatches.txt
Event Time: 12/16/2024 06:43:30
Event ID: 1
Event Record ID: 128081
Command Line: C:\Windows\system32\cmd.exe /c reg query "HKLM\Software\Microsoft\Windows\CurrentVersion"
/v ProgramFilesDir
YARA Result: SANDBOXDETECTED C:\Users\Administrator\AppData\Local\Temp\2\tmpE3D4.tmp
3 Job3 BackgroundJob Running True localhost ...

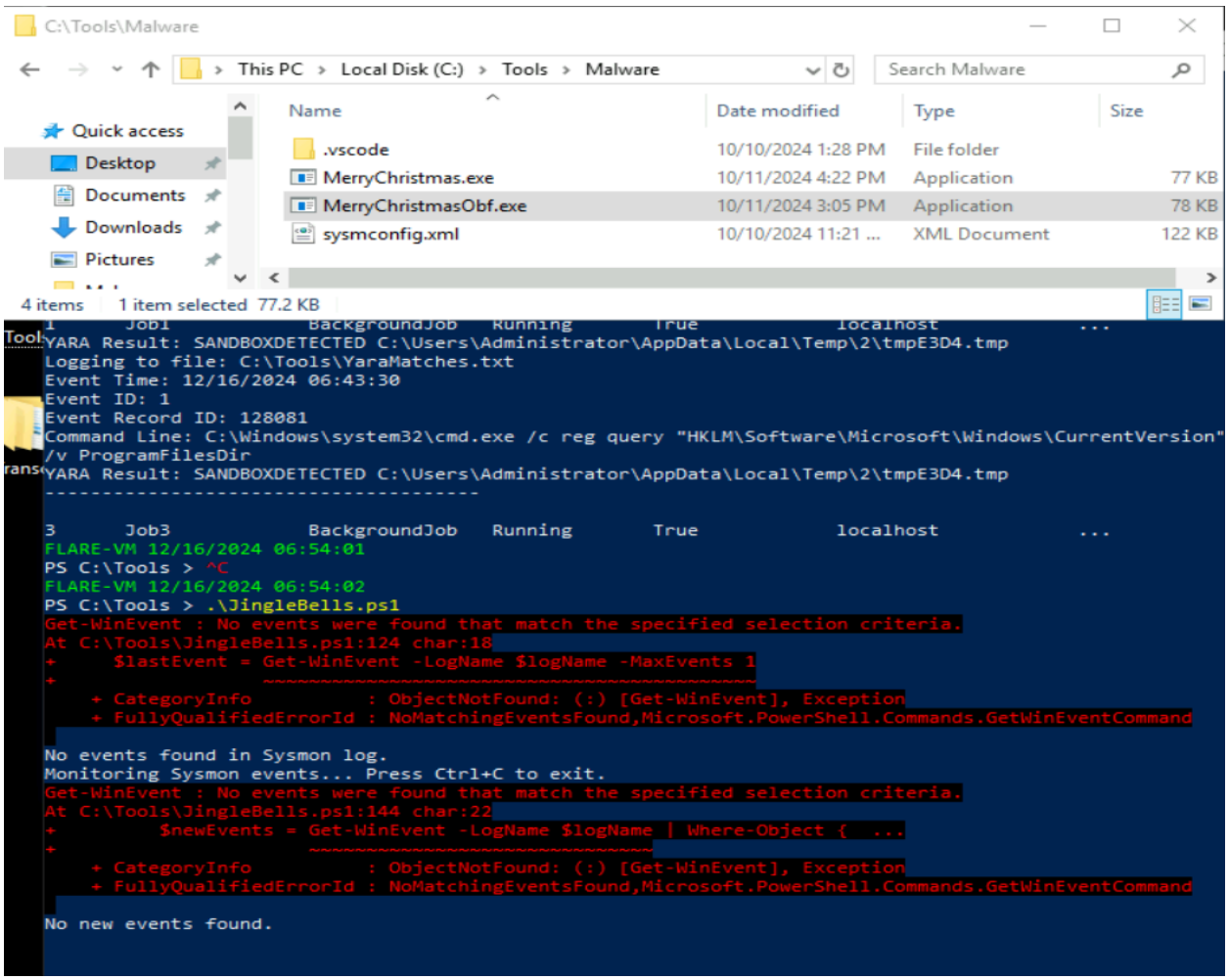
Q1> What is the flag displayed in the popup window after the EDR detects the malware?

Ans: **THM{GlitchWasHere}**

- Now let us test the malware ***MerryChristmas.exe*** .
- We can see the logs as we run the **.exe file**.

Enhancing Malware Evasion Techniques

- To evade detection, malware can obfuscate its operations. For example, using ***Base64*** encoding in registry queries:



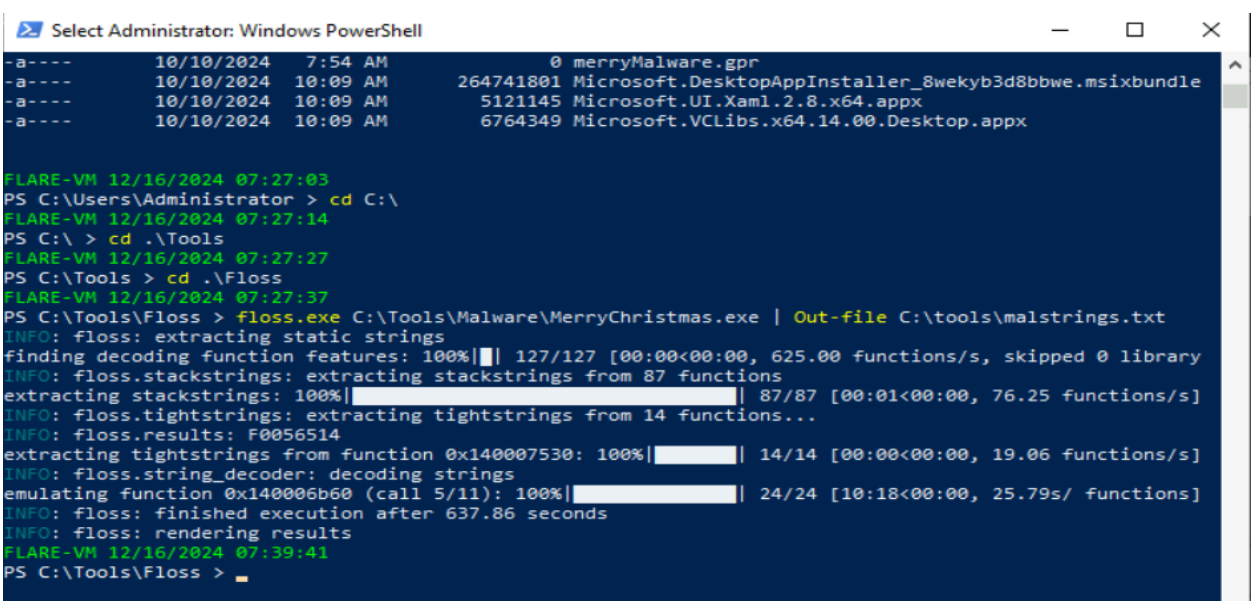
- **Obfuscated Code Example:**

```
void registryCheck() {
const char *encodedCommand =
"RwBlAHQALQBJAHQAZQBtAFAAcgBvAHAAZQByAHQAeQAgAC0AUABhAHQAaAAgA
CIASABLAEwATQA6AFwAUwBvAGYAdAB3AGEAcgBlAFwATQBpAGMAcgBvAHMAbwB
mAHQAXABXAGkAbgBkAG8AdwBzAFwAQwB1AHIAcgBlAG4AdABWAGUAcgBzAGkAb
wBuACIAIAAtAE4AYQBtAGUAIABQAHIAbwBnAHIAYQBtAEYAaQBsAGUAcwBEAGkAcg
A=";
   char command[512];
   snprintf(command, sizeof(command), "powershell -EncodedCommand %s",
encodedCommand);
   int result = system(command);
   if (result == 0) {
      printf("Registry query executed successfully.\n");
   } else {
      fprintf(stderr, "Failed to execute registry query.\n");
   }
}
```

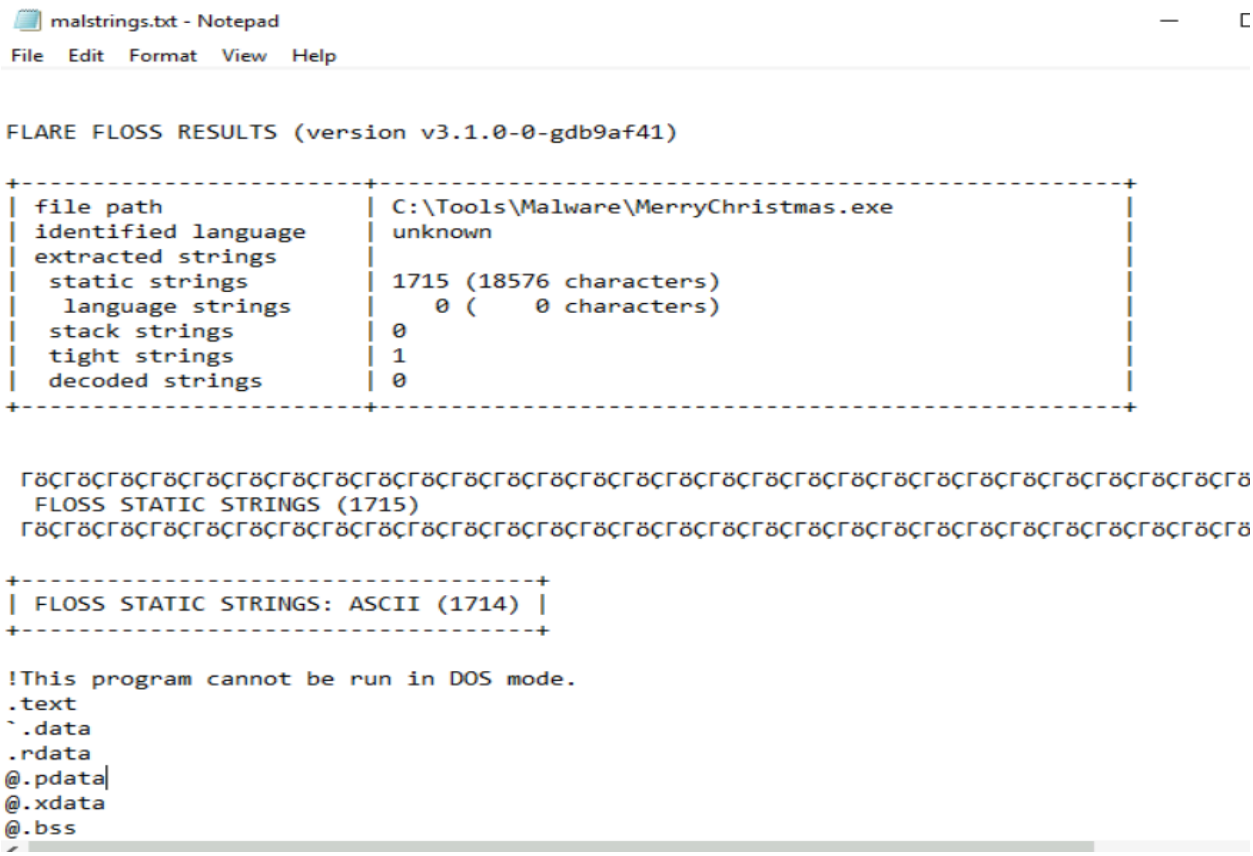
This obfuscation complicates detection, but tools like **Floss** can reveal concealed details.

Using Floss:

PS C:\Tools\FLOSS> **floss.exe C:\Tools\Malware\MerryChristmas.exe | Out-file C:\tools\malstrings.txt**



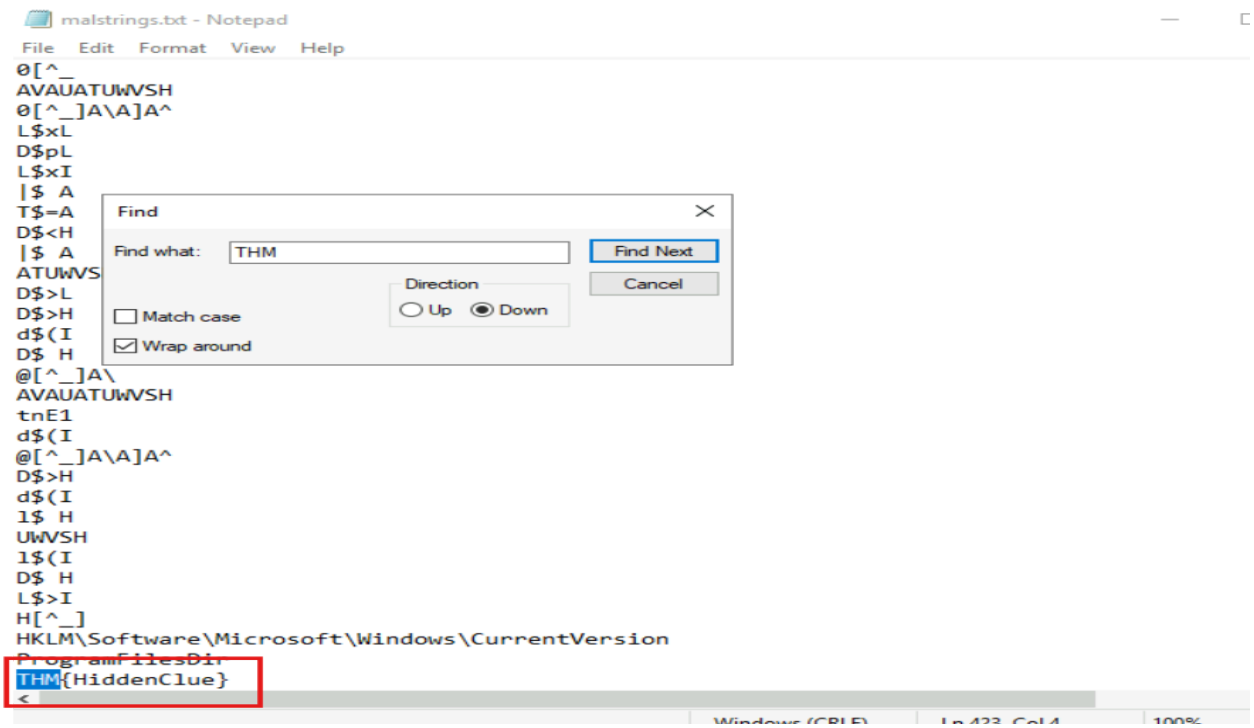
- **Command Breakdown:**
  - `floss.exe`: Scans for strings in the binary.
  - `|`: Redirects output.
  - `Out-file`: Saves results to `malstrings.txt`.

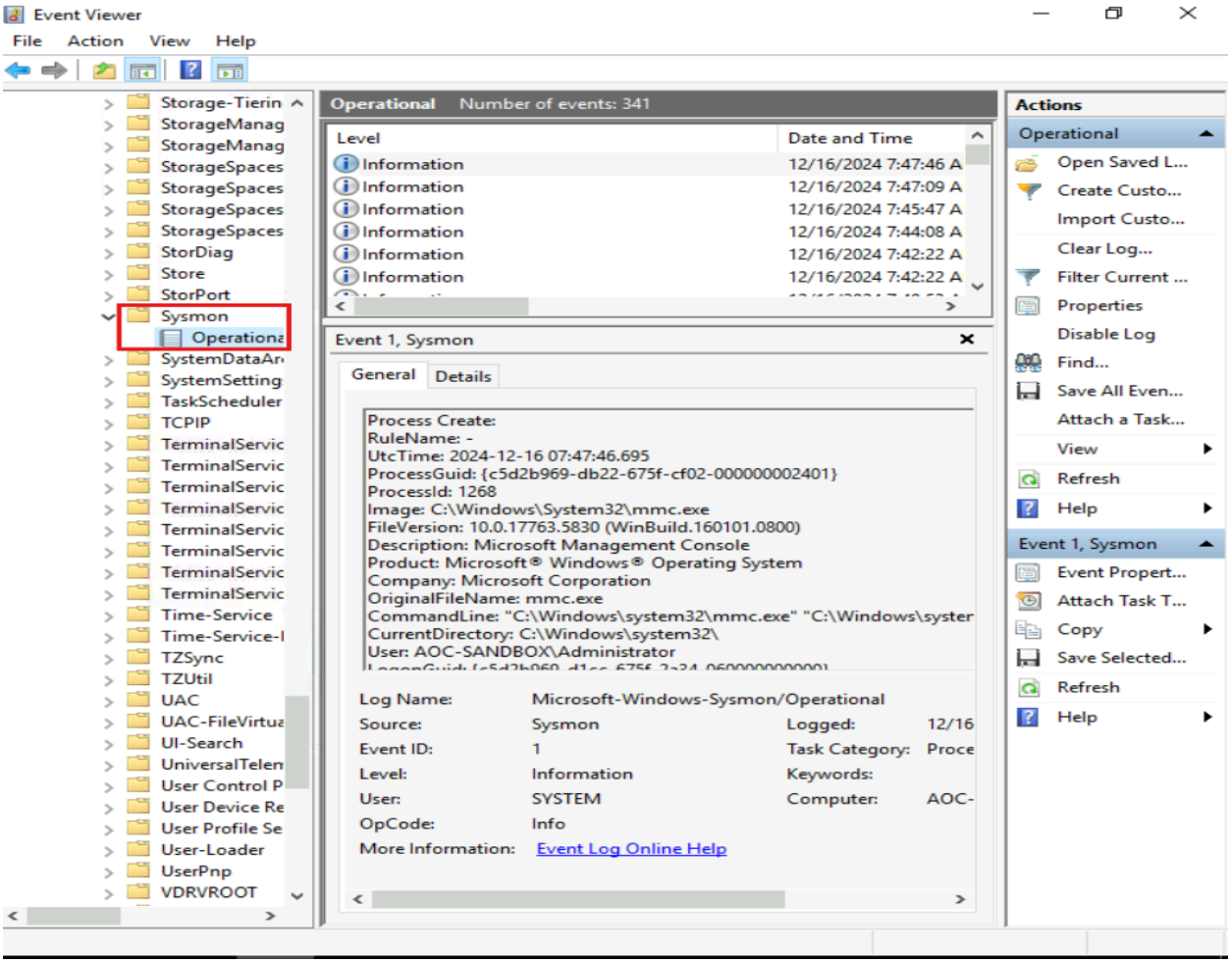


```
FLARE FLOSS RESULTS (version v3.1.0-0-gdb9af41)

+------------------------+-----------------------------------------------------+
| file path              | C:\Tools\Malware\MerryChristmas.exe                 |
| identified language    | unknown                                             |
| extracted strings      |                                                     |
|  static strings        | 1715 (18576 characters)                             |
|   language strings     |    0 (    0 characters)                             |
|  stack strings         | 0                                                   |
|  tight strings         | 1                                                   |
|  decoded strings       | 0                                                   |
+------------------------+-----------------------------------------------------+


 ΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓö
  FLOSS STATIC STRINGS (1715)
 ΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓöÇΓö

+------------------------------------+
| FLOSS STATIC STRINGS: ASCII (1714) |
+------------------------------------+

!This program cannot be run in DOS mode.
.text
`.data
.rdata
@.pdata
@.xdata
@.bss
```

Q2> What is the flag found in the malstrings.txt document after running floss.exe, and opening the file in a text editor?

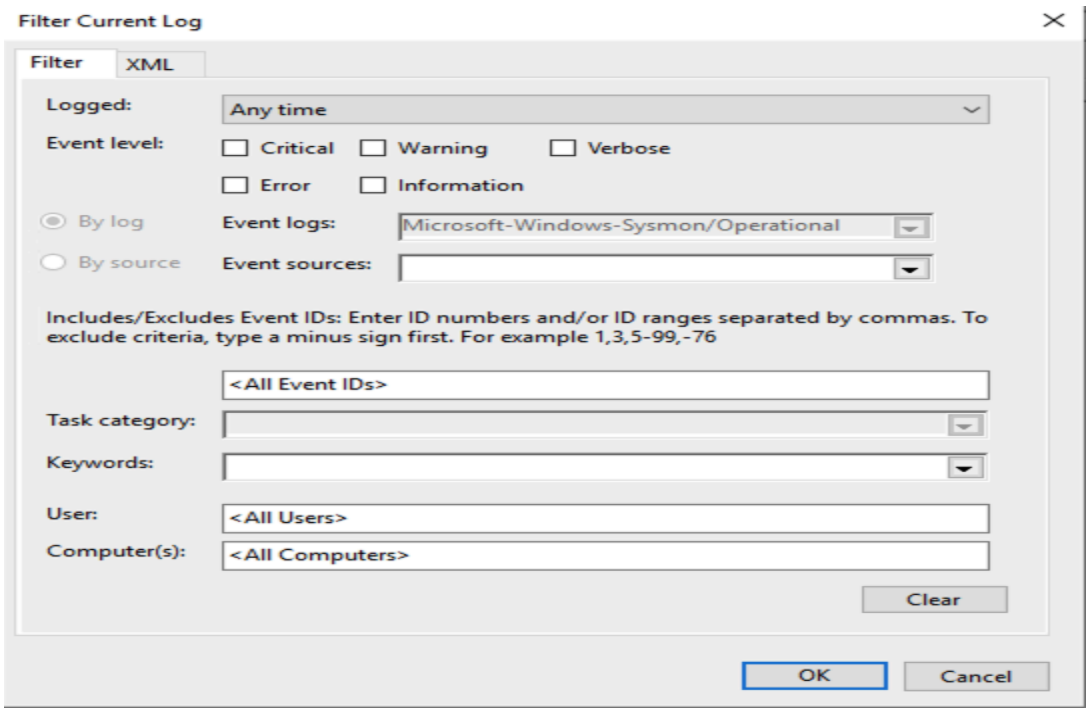
Ans: **THM{HiddenClue}**

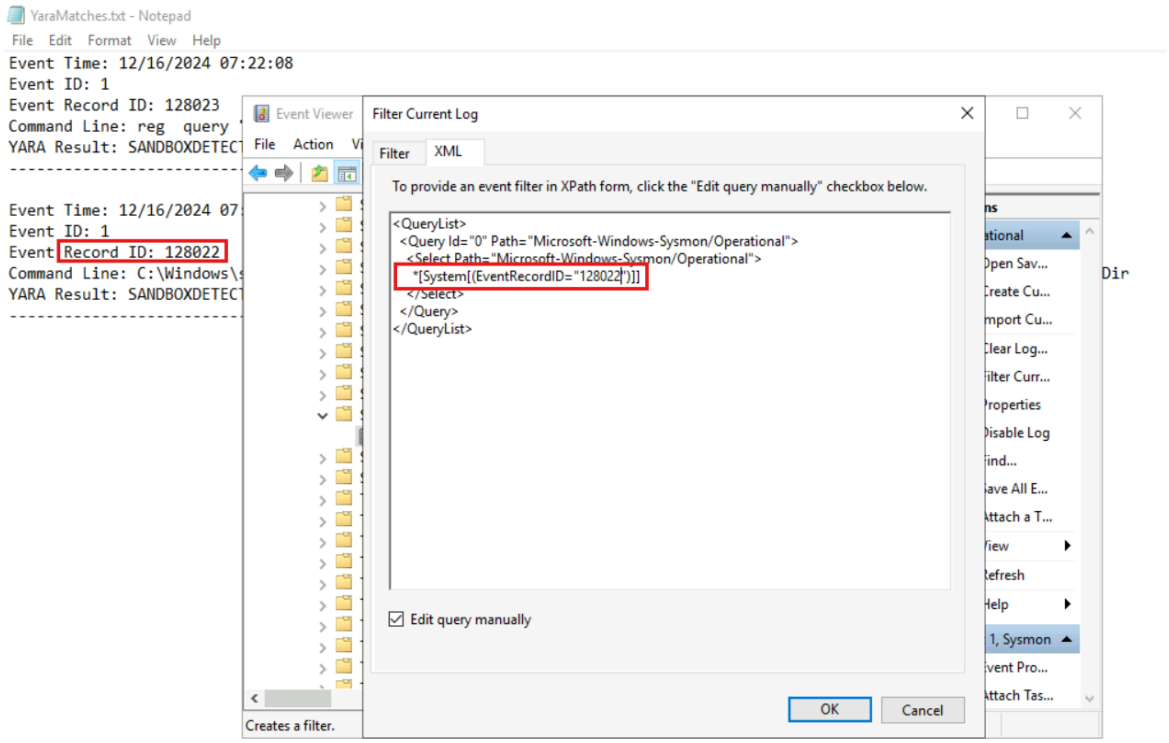

malstrings.txt - Notepad
File Edit Format View Help
0[^_
AVAUATUWVSH
0[^_]A\A]A^
L$xL
D$pL
L$xI
|$ A
T$=A
D$<H
|$ A
ATUWVS
D$>L
D$>H
d$(I
D$ H
@[^_]A\
AVAUATUWVSH
tnE1
d$(I
@[^_]A\A]A^
D$>H
d$(I
l$ H
UWVSH
l$(I
D$ H
L$>I
H[^_]
HKLM\Software\Microsoft\Windows\CurrentVersion
THM{HiddenClue}
Find
Find what: THM
Find Next
Cancel
Direction
Up
Down
Match case
Wrap around

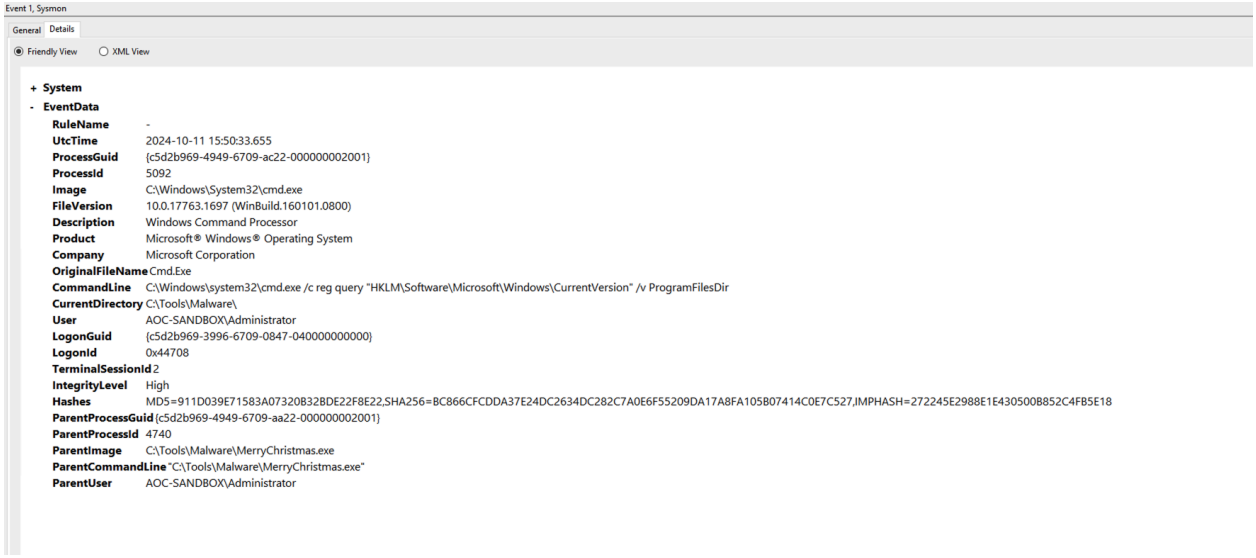

Event Viewer
File Action View Help
Storage-Tierin
StorageManag
StorageManag
StorageSpaces
StorageSpaces
StorageSpaces
StorageSpaces
StorDiag
Store
StorPort
Sysmon
Operationa
SystemDataAr
SystemSetting
TaskScheduler
TCPIP
TerminalServic
Time-Service
Time-Service-I
TZSync
TZUtil
UAC
UAC-FileVirtua
UI-Search
UniversalTelen
User Control P
User Device Re
User Profile Se
User-Loader
UserPnp
VDRVROOT
Operational Number of events: 341
Level
Date and Time
Information 12/16/2024 7:47:46 A
Information 12/16/2024 7:47:09 A
Information 12/16/2024 7:45:47 A
Information 12/16/2024 7:44:08 A
Information 12/16/2024 7:42:22 A
Information 12/16/2024 7:42:22 A
Event 1, Sysmon
General Details
Process Create:
RuleName: -
UtcTime: 2024-12-16 07:47:46.695
ProcessGuid: {c5d2b969-db22-675f-cf02-000000002401}
ProcessId: 1268
Image: C:\Windows\System32\mmc.exe
FileVersion: 10.0.17763.5830 (WinBuild.160101.0800)
Description: Microsoft Management Console
Product: Microsoft® Windows® Operating System
Company: Microsoft Corporation
OriginalFileName: mmc.exe
CommandLine: "C:\Windows\system32\mmc.exe" "C:\Windows\syster
CurrentDirectory: C:\Windows\system32\
User: AOC-SANDBOX\Administrator
Log Name: Microsoft-Windows-Sysmon/Operational
Source: Sysmon
Logged: 12/16
Event ID: 1
Task Category: Proce
Level: Information
Keywords:
User: SYSTEM
Computer: AOC-
OpCode: Info
More Information: Event Log Online Help
Actions
Operational
Open Saved L...
Create Custo...
Import Custo...
Clear Log...
Filter Current ...
Properties
Disable Log
Find...
Save All Even...
Attach a Task...
View
Refresh
Help
Event 1, Sysmon
Event Propert...
Attach Task T...
Copy
Save Selected...
Refresh
Help

Sysmon and YARA Rules

- Sysmon logs provide detailed event data for malware analysis.
- YARA rules can be applied to these logs to detect malicious behavior.







Event 1, Sysmon

General | Details

Friendly View | XML View

```
+ System
- EventData
    RuleName          -
    UtcTime           2024-10-11 15:50:33.655
    ProcessGuid       {c5d2b969-4949-6709-ac22-000000002001}
    ProcessId         5092
    Image             C:\Windows\System32\cmd.exe
    FileVersion       10.0.17763.1697 (WinBuild.160101.0800)
    Description       Windows Command Processor
    Product           Microsoft® Windows® Operating System
    Company           Microsoft Corporation
    OriginalFileName  Cmd.Exe
    CommandLine       C:\Windows\system32\cmd.exe /c reg query "HKLM\Software\Microsoft\Windows\CurrentVersion" /v ProgramFilesDir
    CurrentDirectory  C:\Tools\Malware\
    User              AOC-SANDBOX\Administrator
    LogonGuid         {c5d2b969-3996-6709-0847-040000000000}
    LogonId           0x44708
    TerminalSessionId 2
    IntegrityLevel    High
    Hashes            MD5=911D039E71583A07320B32BDE22F8E22,SHA256=BC866CFCDDA37E24DC2634DC282C7A0E6F55209DA17A8FA105B07414C0E7C527,IMPHASH=272245E2988E1E430500B852C4FB5E18
    ParentProcessGuid {c5d2b969-4949-6709-aa22-000000002001}
    ParentProcessId   4740
    ParentImage       C:\Tools\Malware\MerryChristmas.exe
    ParentCommandLine "C:\Tools\Malware\MerryChristmas.exe"
    ParentUser        AOC-SANDBOX\Administrator
```

Filtering Logs in Event Viewer:

1. Open **Event Viewer**.
2. Navigate to `Applications and Services Logs -> Microsoft -> Windows -> Sysmon -> Operational`.
3. Use the **Filter Current Log** option to narrow results by `Event Record ID`.

Cybersecurity Tip

Always use multiple layers of defense, such as firewalls, antivirus solutions, and intrusion detection systems, to ensure maximum security against malware attacks. Regularly update and patch your systems to minimize vulnerabilities.