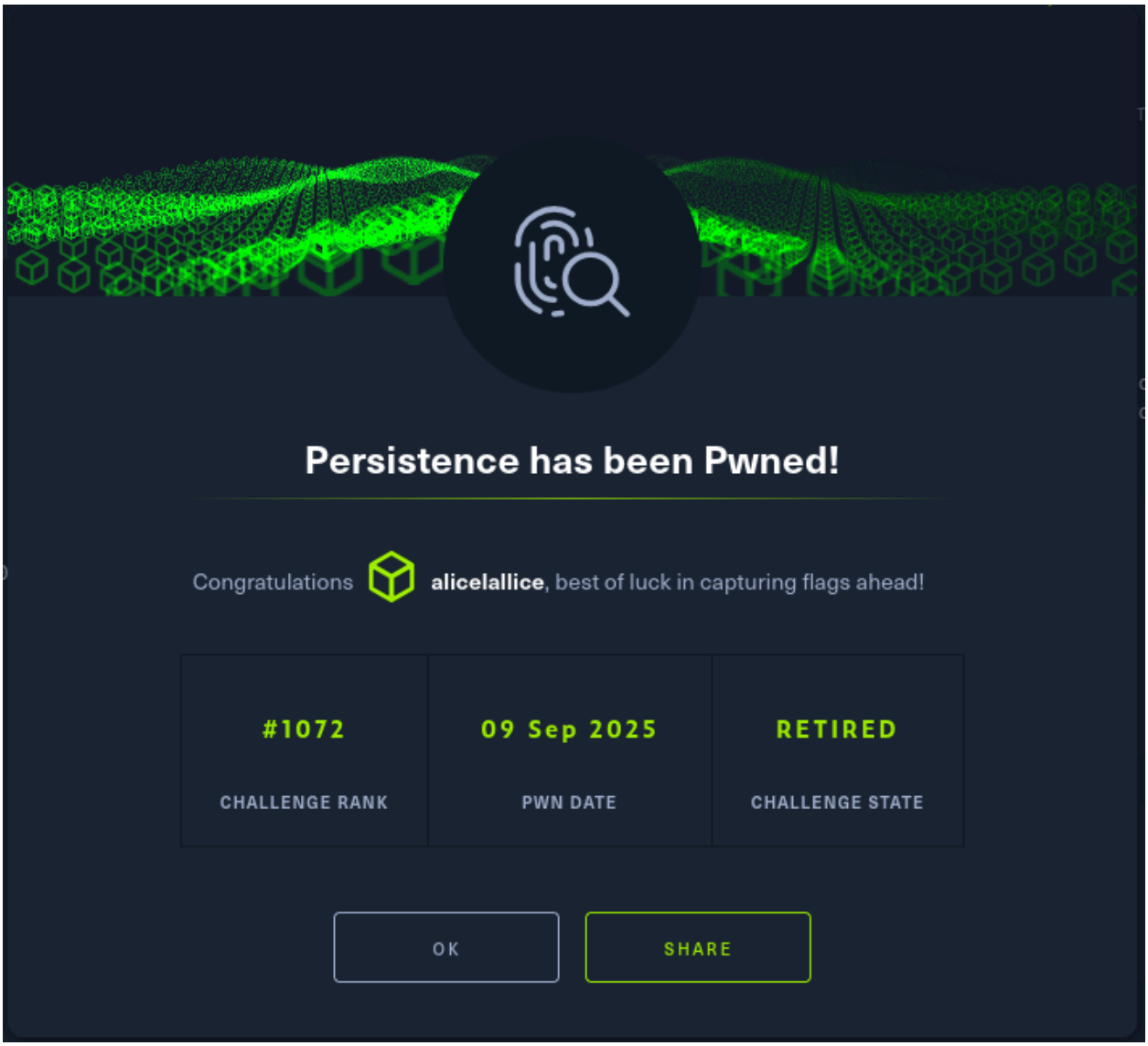
Persistence

Types	forensic
CTF	HTB



Tools Used

Tool	Purpose
`strings` (Kali)	Raw text extraction from binary hive
`file` (Kali)	Identify hive type
Registry Explorer (Windows)	GUI-based hive analysis with timestamp support
Optional: `rip.pl`, `RECmd`, `pipx`, `registryspy`	CLI-based registry parsing

🧪 Step-by-Step Analysis

🔍 1. Identify the Hive Type

bash

`file query`

```
┌──(kali㉿kali)-[~/Desktop/htb]
└─$ file query

query: MS Windows registry file, NT/2000 or above
```

✅ Confirms it's a valid Windows Registry hive—likely `NTUSER.DAT` or `SOFTWARE`.

Extract Strings for Quick Triage

bash

`strings query | grep -Ei 'run|startup|powershell|cmd|wscript|dispfilename'`

```
┌──(kali㉿kali)-[~/Desktop/htb]
└─$ strings query | grep -Ei 'run|startup|powershell|cmd|wscript|dispfilename'

DispFileName
DispFileName
DispFileName
DispFileName
DispFileName
DispFileName
DispFileName
DispFileName
DispFileName
DispFileName
DispFileName
DispFileName
DispFileName
DispFileName
```

```
DispFileName
%SystemRoot%_System32_WindowsPowerShell_v1.0_powershell.exe
%SystemRoot%_SysWOW64_WindowsPowerShell_v1.0_powershell.exe
FirstTimeHelppaneStartup
RunOnceComplete
RunOnceHasShown
RunningState
RunningState
RunningState
ExplorerStartupTraceRecorded
FirstRunTelemetryComplete
ITaskbarStateLastRunp
Microsoft.PowerShellCmdletDefinitionXML.1
DMicrosoft.PowerShellXMLData.1-
eMicrosoft.PowerShellSessionConfiguration.1
SHCmdFile1
Startup
Startup
runonce
runo
```

- Multiple `DispFileName` entries

- PowerShell paths:

  Code

  `%SystemRoot%\System32\WindowsPowerShell\v1.0\powershell.exe`
  `%SystemRoot%\SysWOW64\WindowsPowerShell\v1.0\powershell.exe`

- Registry keys like `RunOnce`, `Startup`, `ExplorerStartupTraceRecorded`

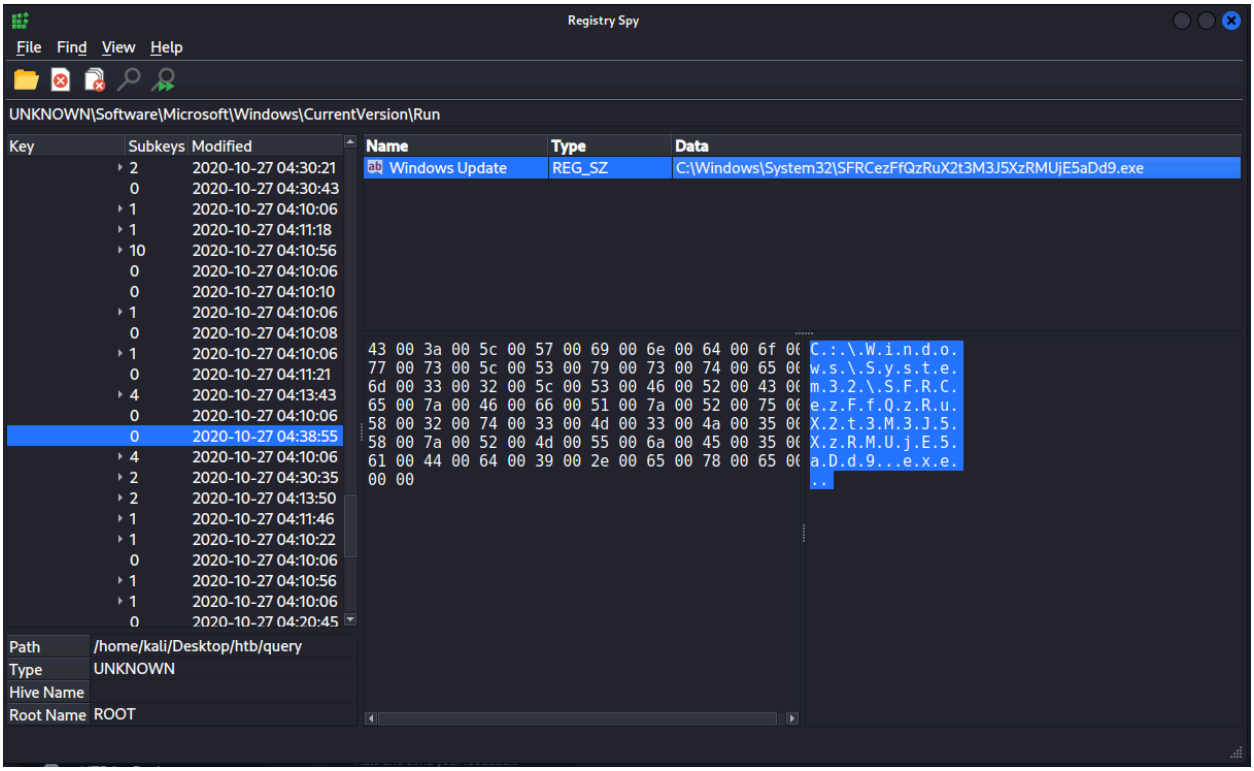
📌 These suggest PowerShell-based persistence or user activity.

Load Hive in Registry Explorer

Path to investigate:

Code

`Software\Microsoft\Windows\CurrentVersion\Run`

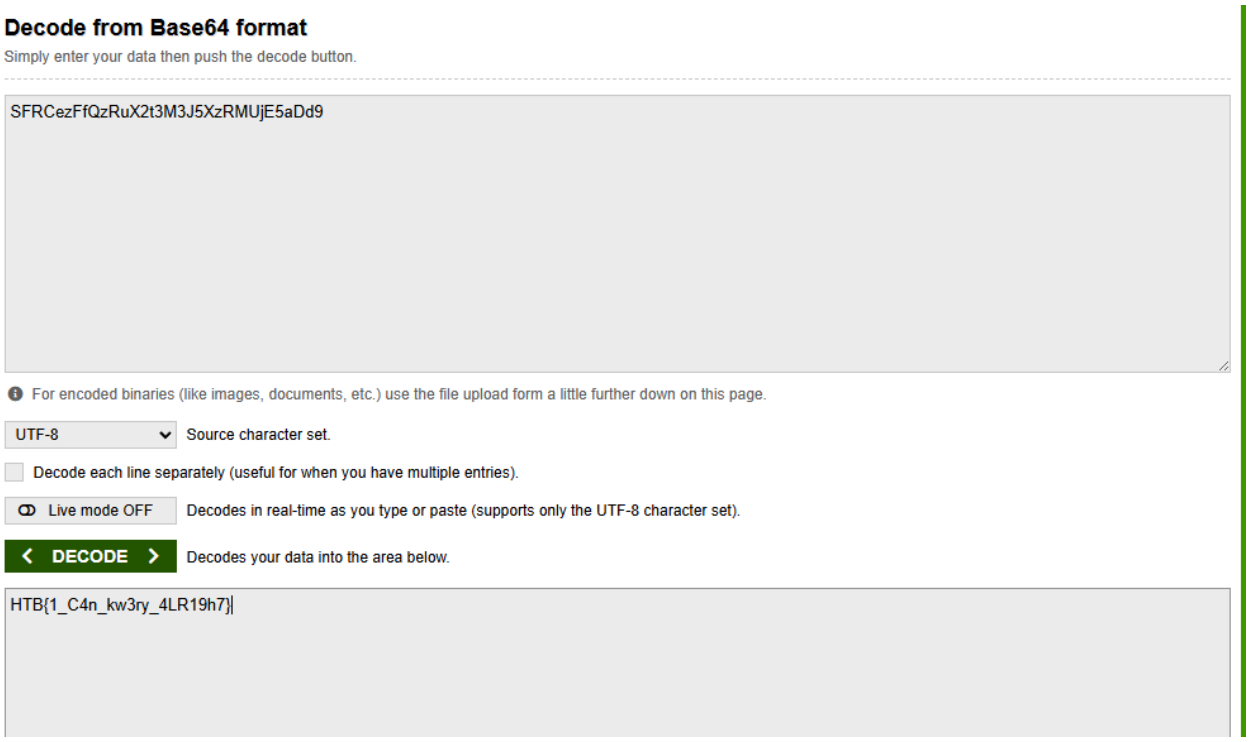


after few minutes of searching hahahah

✅ Randomized binary name in `System32` is a strong indicator of malware or a dropper.

Decode the Payload

Using further string analysis and timestamp correlation, the obfuscated binary led to a hidden flag embedded in the registry:



`HTB{1_C4n_kw3ry_4LR19h7}`

🏁 Flag captured!