Microsoft Defender

- Query AV Product and Status

Query AV Product and Status

I frequently run into a mix of antivirus products in various organizations and was looking into a good way to report on which was installed/active. Unfortunately looking at a list of processes or installed applications is not accurate enough. The Windows Security Application shows which AV is on and its health. While this is a start, I dislike any answer that requires me to interact with the system in a manner that interrupts the customer unless I have to.

Doing some quick research I did find a few very useful articles that pointed out that this information can be queried with a WMI Instance.

```
Get-CimInstance -Namespace root/SecurityCenter2 -ClassName AntiVirusProduct


displayName             : Windows Defender
instanceGuid            : {D68DDC3A-831F-4fae-9E44-DA132C1ACF46}
pathToSignedProductExe  : windowsdefender://
pathToSignedReportingExe : %ProgramFiles%\Windows Defender\MsMpeng.exe
productState            : 397568
timestamp               : Sat, 30 Dec 2023 03:30:13 GMT
```

If Defender is the only AV it is returned as the only output. Windows Security is able to query if there are current threats, issues with the protection settings, and if there are out of date updates that are missing. Many of the sites I found did say the productState property had the information but none had any official documentation as to what the mean or only had pieces of decoding it. The sources didn't agree on whether this value needed to be converted to a hex or binary value, both of which can be done easily enough.

```
$AV = Get-CimInstance -Namespace root/SecurityCenter2 -ClassName AntiVirusProduct
[system.convert]::ToString($AV.productState,16) #Convert to Hexadecimal
[system.convert]::ToString($AV.productState,2) #Conver to Binary
```

With enough digging this was the most useful resource I found which will return the ProductName, ProductState, SignatureStatus, and Owner.

How To Tell What AntiVirus Software Installed on a Remote Windows Computer - NEXTOFWINDOWS.COM

Which in turn borrowed from Identifying Antivirus Engine State | IderaBlog.

```
# define bit flags

[Flags()] enum ProductState
{
    Off         = 0x0000
    On          = 0x1000
    Snoozed     = 0x2000
    Expired     = 0x3000
}

[Flags()] enum SignatureStatus
{
    UpToDate    = 0x00
    OutOfDate   = 0x10
}

[Flags()] enum ProductOwner
{
    NonMs       = 0x000
    Windows     = 0x100
}

# define bit masks

[Flags()] enum ProductFlags
{
    SignatureStatus = 0x00F0
    ProductOwner    = 0x0F00
    ProductState    = 0xF000
}

# get bits
$infos = Get-CimInstance -Namespace root/SecurityCenter2 -ClassName AntiVirusProduct -ComputerName
$computer
ForEach ($info in $infos){
   [UInt32]$state = $info.productState

   # decode bit flags by masking the relevant bits, then converting
   [PSCustomObject]@{
        ProductName = $info.DisplayName
```

```
        ProductState = [ProductState]($state -band [ProductFlags]::ProductState)
        SignatureStatus = [SignatureStatus]($state -band [ProductFlags]::SignatureStatus)
        Owner = [ProductOwner]($state -band [ProductFlags]::ProductOwner)
    }
}
```

Of course, if you are only using Microsoft Defender getting the status or configuration it is far simpler and most of what you might be looking of is either in `Get-MpComputerStatus` or `Get-MpPreference`.