**MAPS data**

**What is MAPS?**

The Microsoft Active Protection Service is the cloud service that enables:

* Clients to report key telemetry events and suspicious malware queries to the cloud
* Cloud to provide real-time blocking responses back to the client

The MAPS service is available for all Microsoft’s antivirus products and services, including:

* Microsoft Forefront Endpoint Protection
* Microsoft Security Essentials
* System Center Endpoint Protection
* Windows Defender on Windows 8 and later versions

<https://blogs.technet.microsoft.com/mmpc/2015/01/14/maps-in-the-cloud-how-can-it-help-your-enterprise/>

**----------------------------------------------------------------------------------------------------**

A few pitfalls to pay attention to:

* **String fields' format**:  
  Most strings are lowercase, including all hexadecimal fields (e.g. SigSeq, Sha1).  
  Enum-like strings (which have a few possible values) will use lower\_case format - e.g. "stream\_threat".  
  A few fields will remain in the original mixed case format, where their casing is important/useful - e.g. file names, threat names, etc.
* **Obfuscation** – see below

**Legend**:

Obfuscated

**Columns:**

FileNameID,Sha256ID,Md5ID,Sha1ID,IssuerID,SignerID,PublisherID,WebFileIpID,WebFileUrlID,ThreatFamilyID,ThreatNameID,SigSeqID,SignatureNameID,MachineGuidID,RowNumber,PeAttributesString,WebFileUrlDomain,WebFileUrlHostname,Ctph,LsHash,ThreatCategory,ReportTime,SpynetReportGuid,Firewall,Size,Version,Type,SignerHash,IssuerHash,IsValidCertificate,CertInvalidDetails,Checksum,IsRuntimePacked,FileAccess,Screensaver,StartupUserFolder,MarkOfTheWeb,UsnTimestamp,IsInContainer,IsContainer,IsPEFile,IsAutoRunInf,PeTimestamp

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| FileName | string | | | This is the name of the file scanned by the Engine. Note that this only the filename, not the full path. This is at most 513 characters (name, separator '.' and extension). See also [CoreReport/realpath](http://onedoc/wcd/html/CuratedTelemetry.CoreReports%20View.htm#CoreReport/realpath) and [CoreReport/vpath](http://onedoc/wcd/html/CuratedTelemetry.CoreReports%20View.htm#CoreReport/vpath). | |
| SSha256 | string | | | This is the SHA-256 of the file's contents. This is 32 bytes long. For more information on SHA-256, see <http://en.wikipedia.org/wiki/Sha-256> See also: [CoreReport/sha1](http://onedoc/wcd/html/CuratedTelemetry.CoreReports%20View.htm#CoreReport/sha1), [CoreReport/sha512](http://onedoc/wcd/html/CuratedTelemetry.CoreReports%20View.htm#CoreReport/sha512) Example: <corereport ...="" sha256="93A652624F72CD2BEF40897DF7AAA1E11DD1C884C274CFF159B355682DA42D88" ...=""></corereport> | |
| Md5 | string | | | This is the MD5 hash of the file's contents. For more information on MD5, see: <http://en.wikipedia.org/wiki/MD5> Example: <corereport ...="" md5="15064C0C479C26A92DCA7DDB80B3AE49" ...=""></corereport> | |
| Sha1 | string | | | This is the SHA1 hash of the file's contents. This is 20 bytes long. Some notable values include:   * 0xDA39A3EE5E6B4B0D3255BFEF95601890AFD80709 - This is the hash of a file with no contents (0 byte size) * 0x0000000000000000000000000000000000000000 - This denotes that at spynet collection time, the file could not be opened by the Engine. * 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF - This denotes that the file's size (CoreReport/size) exceeds 256 MB, and the hash was not calculated. In this case, the reported size is still correct.   For more information on SHA1, see: <http://en.wikipedia.org/wiki/Sha1> See also: CoreReport/sha256, CoreReport/sha512 Example: <corereport ...="" sha1="B9C264456FF2B4C0EB2B0C57F5304BCD3AB9408A" ...=""></corereport> | |
| Issuer | string | | | For authenticode signed files, this is the name of the issuing authority (the certificate authority). If the signature is considered invalid on the client for any reason, the Signer field starts with "INVALID:". In this case, the [CoreReport/certinvaliddetails](http://onedoc/wcd/html/CuratedTelemetry.CoreReports%20View.htm#CoreReport/certinvaliddetails) field will contain information on why the certificate is considered invalid. For more information on Authenticode, see: <http://msdn.microsoft.com/en-us/library/ie/ms537359(v=vs.85).aspx> Example: <corereport ...="" signer="Microsoft Windows" signerhash="2192383B4786A4D9F48EF81D3E053E6546F14D57" authenticodehash="5D9314D6893066D9DFB03AE97E7B3F98DCCA2BB6" signingtime="130052935930000000" issuer="Microsoft Windows Verification PCA" issuerhash="211C1BA9C43157D5D04293E5569D51F3C240EDC0" ...=""></corereport> | |
| Signer | | string | For authenticode signed files, this is the name of the Authenticode signer. If the signature is considered invalid on the client for any reason, the Signer field starts with "INVALID:". In this case, the [CoreReport/certinvaliddetails](http://onedoc/wcd/html/CuratedTelemetry.CoreReports%20View.htm#CoreReport/certinvaliddetails) field will contain information on why the certificate is considered invalid. For more information on Authenticode, see: <http://msdn.microsoft.com/en-us/library/ie/ms537359(v=vs.85).aspx> Example: <corereport ...="" signer="Microsoft Windows" signerhash="2192383B4786A4D9F48EF81D3E053E6546F14D57" authenticodehash="5D9314D6893066D9DFB03AE97E7B3F98DCCA2BB6" signingtime="130052935930000000" issuer="Microsoft Windows Verification PCA" issuerhash="211C1BA9C43157D5D04293E5569D51F3C240EDC0" ...=""></corereport> | | |
| Publisher | | string | Publisher name from the file's resource strings. | | |
| ThreatFamily | string | | Threat family name (for example: "Nuwar") | |
| ThreatName | string | | Threat name (for example: "TrojanDownloader:JS/Nuwar.A") | |
| SignatureName | string | | The SignatureXml root attribute "Name" (a calculated column) | |
| MachineGuid | Guid | | This is a GUID token that is used to uniquely identify the machine. This value does not change unless Windows is re-installed. Internally this value is read from the regkey HKLM\Software\Microsoft\RemovalTools\MRT\\GUID. It is possible for multiple machines to share a machine guid via replication by imaging or virtual machines.  Example: <spynetreport ...="" machineguid="78bce2d4-a241-5ec1-0092-2678ab30D124" ...=""></spynetreport> | | |
| PeAttributesString | string | | | Boolean attributes for static features evaluated by the engine while scanning PE executable files. This is a comma separated list of attribute identifiers (currently between 0 and 1024). | |

|  |  |  |
| --- | --- | --- |
| SigSeq | string | If this is an autolofi, sets the MSB and returns the sigseq Otherwise returns the sigseq itself Returns: The Canonical SigSeq |
| WebFileUrl | string | For IOAV reports (IOfficeAntiVirus, denoted by CoreReport/agent value of 772), this is the szOrigURL parameter, which is the address bar value for Explorer and Internet Explorer. There is no guarantee that this is a well-formed URL. For instance, BitComet has sent www.example.com in the past and is currently sending about:internet. See also: CoreReport/webfileip |
| WebFileIp | string | The is the IP address corresponding to the URL reported in CoreReport/webfileurl. The DNS lookup is done at report generation time to help us get the IP addresses of fast flux domains. The GetAddrInfoW function of Winsock is used to perform DNS resolution. See also: CoreReport/streamip and CoreReport/frameip |

|  |  |  |
| --- | --- | --- |
| Ctph | string | This is the "context triggered piecewise hash" (CTPH) of the given file. This is a locale sensitive hash (conceptually similar to CoreReport/lshash) and can be used to identify highly similar files. For details on the CTPH hash, see <http://www.dfrws.org/2006/proceedings/12-Kornblum.pdf> Example: <corereport ...="" ctph="192:4NXihvU9o+/Bbb8KpB9MTdaWaBWW24:MXA+dVGwWaBWW24" ...=""></corereport> |
| LsHash | string | A locality-sensitive hash (LSHASH) is a hash that is designed to be very similar (many matching bits) between similar files. This provides a way to identify similar files. Note that the lshash algorithm was changed in June 2013 (build 1.1.9700.0) due to an error in the hashing algorithm. For more information, see: <http://sharepoint/sites/mswikis-av/Pages/LSHASH.aspx> Example: <corereport ...="" lshash="7BFEF96FDBBD9D7677AFF967A79995565FA5EA6F5676A5996FAFBF7AA6D76BEF57795FD9E96A5AA79D7A9B99B59DEAA5A5DE5FB76EEFBFFE795B5FD99BD7FAEF" lshashs="E57EEB4AB51CA10083395084B37798F741836E53428B2C89ACD3B6E736839BBF" ...=""></corereport> |
| ThreatCategory | string | Threat category name (for example: "Trojan downloader") |

|  |  |  |
| --- | --- | --- |
| ReportTime | DateTime | The time the report was received in spynet frontend, in UTC Returns: report time |
| SpynetReportGuid | Guid | Identifier of the report |
| Firewall | Boolean? | This attribute is true (1) for Windows 8.1 and above if windows firewall is enabled, as reported by the service. |
| Size | Int64? | This is the size of the file on disk, in bytes. |

|  |  |  |
| --- | --- | --- |
| Version | string | Product version from the file's resource strings. |
| Type | string | If the file is a PE file, this attribute specifies the specific type of PE file. Specifically, this is the dwFileType attribute in the file's VS\_FIXEDFILEINFO header in string format. See <http://msdn.microsoft.com/en-us/library/windows/desktop/ms646997(v=vs.85).aspx> for more information. This field contains the dwFileType without the "VFT\_" prefix, e.g. for a DLL, dwFileType will be "VFT\_DLL" but the value in field will be "DLL". |
| SignerHash | string | For authenticode signed files, this is the SHA1 hash of the signer's certificate. The signer is reported in [CoreReport/signer](http://onedoc/wcd/html/CuratedTelemetry.CoreReports%20View.htm#CoreReport/signer). For more information on Authenticode, see: <http://msdn.microsoft.com/en-us/library/ie/ms537359(v=vs.85).aspx> Example: <corereport ...="" signer="Microsoft Windows" signerhash="2192383B4786A4D9F48EF81D3E053E6546F14D57" authenticodehash="5D9314D6893066D9DFB03AE97E7B3F98DCCA2BB6" signingtime="130052935930000000" issuer="Microsoft Windows Verification PCA" issuerhash="211C1BA9C43157D5D04293E5569D51F3C240EDC0" ...=""></corereport> |
| IssuerHash | string | For authenticode signed files, this is the SHA1 hash of the certificate authority's own certificate. The certificate authority is reported in [CoreReport/issuer](http://onedoc/wcd/html/CuratedTelemetry.CoreReports%20View.htm#CoreReport/issuer). For more information on Authenticode, see: <http://msdn.microsoft.com/en-us/library/ie/ms537359(v=vs.85).aspx> Example: <corereport ...="" signer="Microsoft Windows" signerhash="2192383B4786A4D9F48EF81D3E053E6546F14D57" authenticodehash="5D9314D6893066D9DFB03AE97E7B3F98DCCA2BB6" signingtime="130052935930000000" issuer="Microsoft Windows Verification PCA" issuerhash="211C1BA9C43157D5D04293E5569D51F3C240EDC0" ...=""></corereport> |
| IsValidCertificate | Boolean? | Gets a value indicating whether the certificate is valid Returns: True is returned when the certificate is valid, False is returned when Signer starts with the "INVALID:" prefix. When Signer is null the return value is null. |
| CertInvalidDetails | string | For invalid Authenticode certificates, this field will provide information on why the certificate is considered invalid. The value listed here is the return value of the WinVerifyTrust function (<http://msdn.microsoft.com/en-us/library/windows/desktop/aa388208(v=vs.85).aspx>) Some common values include:   * 800B0101 - "A required certificate is not within its validity period when verifying against the current system clock or the timestamp in the signed file." (e.g. certificate expired) * 80096010 - "The digital signature of the object did not verify." (e.g. invalid signature) * 800B010A - "A certificate chain could not be built to a trusted root authority." (e.g. invalid cert chain) * 800B0109 - "A certificate chain processed, but terminated in a root certificate which is not trusted by the trust provider." (e.g. untrusted root certificate) * 80096005 - "The timestamp signature and/or certificate could not be verified or is malformed." * 80092003 - "An error occurred while reading or writing to a file." (could be indicative of an internal sporadic error) * 80096004 - "The signature of the certificate cannot be verified." * 800B0100 - "No signature was present in the subject." * 800b010d - "The certification path terminates with the test root which is not trusted with the current policy settings" * 800B0110 - "The certificate is not valid for the requested usage." * 80096003 - "One of the counter signatures was invalid. " * 800B0111 - "The certificate was explicitly marked as untrusted by the user."   Other (less common) failure codes can be looked up in Windows error-codes, e.g. online or by running errlook in the Visual Studio command prompt. For more information on Authenticode, see: <http://msdn.microsoft.com/en-us/library/ie/ms537359(v=vs.85).aspx> Example: <corereport ...="" signer="INVALID:TestCertificate" certinvaliddetails="800B0109" ...=""></corereport> |
| Checksum | string | Checksum of the file's PE header. |
| IsRuntimePacked | Int16? | This attribute is true (1) if and only if the file is high entropy (for an arbitrarily set definition of "high entropy"). In information theory, entropy is a calculation of the amount of randomness/uncertainty associated with a particular set of data. The more random the data, the higher the entropy. In practice, files that are high entropy tend to correlate strongly with files that are encryped, obfuscated or compressed. |
| FileAccess | string | The is an enum value that describes how the engine accessed the file. Possible values include:   1. Unknown - An invalid FileAccess value was sent in telemetry. 2. Failed - Failed to access the file for scanning 3. Normal - Regular file-open access with full path 4. Relative Open - File was accessed using a path relative to its handle, for instance hAbs = Open("C:\foo.exe", NO\_ACCESS); hRelative = Open(hAbs, ".", FULL\_ACCESS); 5. File ID - File was accessed using its file ID 6. Raw Read - File was located and ID obtained normally, but could only be read using direct file system parsing (DFSP). See <http://sharepoint/sites/mswikis-av/Pages/Direct%20File%20System%20Parsing.aspx> 7. DFSP - File was accessed using DFSP |
| Screensaver | Boolean? | True (1) if the file is registered as a screen saver, false (0) otherwise. |
| StartupUserFolder | Boolean? | True (1) if the file is in the user's or all users startup folders, false (0) otherwise. |
| MarkOfTheWeb | string | This is the security zone that this file was saved from, according to the file's Mark of the Web. If there is no MotW info in this file or it cannot be read, this attribute is omitted, in which case "unknown" is returned. See the enumerated type documentation at [here.](http://onedoc/wcd/html/T_Microsoft_ProtectionServices_Entities_MarkOfTheWeb.htm) |
| UsnTimestamp | DateTime? | This is the timestamp corresponding to the last USN record available for this file, in 100-nanosecond intervals since January 1, 1601 (FILETIME struct). The timestamp is obtained by querying DeviceIoControl with the relevant codes, such as [FSCTL\_QUERY\_USN\_JOURNAL](http://msdn.microsoft.com/en-us/library/windows/desktop/aa364583(v=vs.85).aspx). For more information about USN journaling, see <http://msdn.microsoft.com/en-us/library/windows/desktop/aa363798(v=vs.85).aspx> |
| IsInContainer | Boolean? | Whenever a file in a container is detected as a threat, telemetry on the container itself is also sent up as a FileReport/CoreReport, inheriting the same threatid and sigseq. The contained file will have this attribute set to true (1) to identify it as the contained file. See also CoreReport/iscontainer |
| IsContainer | Boolean? | Whenever a file in a container is detected as a threat, telemetry on the container itself is also sent up as a FileReport/CoreReport, inheriting the same threatid and sigseq. The container file will have this attribute set to true (1) to identify it as the container file. Note that the absence of this attribute should not be interpreted to mean that this file is not a container. This attribute is only set and present when one of its child files has been detected as a threat. This attribute is redundant starting from Engine August 2015, it is the same as checking CoreReport/reporttype equal to parent\_container\_report. See also CoreReport/isincontainer |
| IsPEFile | Boolean? | Is the reported file a Portable Executable. true indicates it's a PE, false means it's not a PE and null means it cannot be determined based on this telemetry report. Returns: Whether this is a portable executable file. |
| IsAutoRunInf | Boolean? | This value is present and true (1) if and only if the file was detected from an autorun.inf file. |
| PeTimestamp | DateTime? | If the file is a PE file, this attribute specifies the date and time the image was created by the linker, in seconds elapsed since midnight, January 1, 1970 UTC. Specifically, it is the "TimeDateStamp" value in the file's IMAGE\_FILE\_HEADER. See <http://msdn.microsoft.com/en-us/library/windows/desktop/ms680313(v=vs.85).aspx> for more information. |

**Obfuscation procedure:**

For each of the obfuscated columns we do the following:

Build a list containing all (distinct) values seen in the data set(all of it).

For each value in the list give a distinct id (without any special rule, just sort them somehow, and appoint a distinct id to each).

Replace any occurrence of the value with the id.

For example, say that our data contains 3 machines with MachineGuids Guid1,Guid2,Guid3, each can be seen in any given day, with a different amount of times.

Example of the raw data:

Day 1:

Guid1, <some columns>

Guid3, <some columns>

Guid1, <some columns>

Guid2, <some columns>

Day 2:

Guid1, <some columns>

Guid2, <some columns>

So we take the three Guids, and set an id for each:

Value: Guid1 Id: 1

Value: Guid2 Id: 2

Value: Guid3 Id: 3

The output is:

Day 1:

1, <some columns>

3, <some columns>

1, <some columns>

2, <some columns>

Day 2:

1, <some columns>

2, <some columns>

**FileName –** if the file was seen on less than 100 machines, we rename it to “anonymous”. We keep the extension.

**PeAttributesString**

This string is made of human readable “flags” (For example - "isDll,isPacked” ).

Appoint to each of these flags an id, and replace the values of the flags with the ids.

In our example:

Value:"isDll" Id:1

Value:"isPacked”" Id:2

so the new PeAttributesString is : "1,2"

This way we maintain the consistency of data across the machines and reports and there is no way to reverse this without our dictionary.