CylanceV

Administration Guide

2.8.0.5

Contents

Overview

Note: To download CylanceV, submit a request to BlackBerry support.

Cylance's award winning next generation anti-malware product utilizes the Cylance cloud. The Cylance cloud collects threat data, trains and learns from that threat data, and calculates likely outcomes based on what it sees. The Cylance cloud is constantly getting smarter from environmental feedback and a constant stream of new data from all around the world.

Interaction with the Cylance cloud is done using the CylanceV product and is offered in a cloud and local version, with deployment options ranging from an easy to use GUI, CLI, or custom API integration.

CylanceV gives you the power to apply algorithmic science and machine learning techniques to hunt down threats across your organization or on a single compromised system at the click of a button. CylanceV utilizes REST SSL API integration to the Cylance cloud's intelligent cyber security decision-making platform.

CylanceV enables the enhanced assessment of files and data objects that may be threats to your organization by providing:

- On-demand scanning of directories or drives
- Automated scanning of drives or directories for changed or new files
- Visibility inside archived files, including encrypted archives
- Validation of signed files to assess potential compromise

Note: CylanceV and VCL are threat analysis tools, not protection tools. The purpose of CylanceV is to identify threats. To take action upon those threats, like quarantine, and protect your devices, use BlackBerry Protect Desktop.

CylanceV scan scan for signed files, executable files, and documents. These include:

- PE (portable executable)
- ELF
- OLE
- OOXML
- Mach-O
- PDF
- ZIP

System requirements

Item	Description
Operating systems	Windows desktop • Windows 7 (32-bit and 64-bit) • Windows 8.1 (32-bit and 64-bit) • Windows 10 (32-bit and 64-bit) Windows server • Windows Server 2008 R2 (64-bit) • Winodws Server 2012 and 2012 R2 (64-bit) • Windows Server 2016 (Standard, Datacenter, and Essentials) • Windows Server 2019 (Standard and Datacenter)
RAM	2GB
Available software requirements	• .NET Framework 4.6.2 or later • Internet access to register the product

Download CylanceV

You will receive information for downloading the CylanceV files. This should include a link and password to download the product (archived file). You will also receive a license key to use when installing the product. Extract the archive to a directory onto the system where you want to use CylanceV.

License key

Using CylanceV requires a license key. There are two types of license keys: online and offline.

Item	Description
Online	This license key requires an Internet connection so CylanceV can validate the license. Once the online license key has been validated, the Internet connection is no longer needed so long as the system is not shutdown or restarted.
Offline	This license key does not require an Internet connection for validating the license. This allows CylanceV to be used completely offline. Offline license keys will expire on a set date. After this date, a new offline license key is required. **Note:** For version 2.8.0.5, the offline license keys will expire on December 31, 2022.

Installation files

CylanceV and CylanceVCL are executable applications that can be run from any directory. The CylanceV archived file contains the following files.

Item	Description
CylanceV.exe	The graphical user-interface (GUI)
CylanceVCL.exe	The command-line executable
<ELF model file>.cym	The local model file used for the ELF analysis
<MO model file>.cym	The local model file used for Mach-O analysis; if this file is not available, CylanceV will attempt to connect to Cylance cloud to get the cloud model
<OLE model file>.cym	The local model file used for Microsoft Office Document (OLE) analysis
<OOXML model file>.cym	The local model file used for Office Open XML document analysis
<PE model file>.cym	The local model file used for portable executable (PE) analysis; if this file is not available, CylanceV will attempt to connect to Cylance cloud to get the cloud model
<PDF model file>.cym	The local model file used for PDF analysis

Network settings

CylanceV does not require a fixed IP address. CylanceV can use a local model (DSC file) and not require an Internet connection to run threat analysis (but does require an Internet connection to validate the license). If the local model is not used, then CyalnceV will require an Internet connection to use the Cylance cloud model. Using the cloud model requires either HTTP over port 80 or HTTPS over port 443. The DNS name for the Cylance cloud model is api2.cylance.com.

Note: It is highly recommended to allow HTTPS traffic to the Cylance domain or to *.cylance.com over port 443. Contact support with any questions.

Internet connection

When the CylanceV executable is launched, an Internet connection is required to validate the license. Once the license has been validated, the Internet connection is no longer needed. You could disconnect the device from the Internet or network, if required.

If the CylanceV executable is stopped and then restarted, an Internet connection is required to revalidate the license. Also, if the device running CylanceV is turned off and then back on, an Internet connection is required if you restart the CylanceV executable to revalidate the license.

An authorization failed message displays when launching the CylanceV executable if it cannot connect to the Internet and validate the license. To resolve this issue, ensure the device can connect to the Internet and then re-launch the CylanceV executable.

Using CylanceV

CylanceV does not permanently install on the system. When closing CylanceV, the program will remove any temporary files it may have created while running. You can store the CylanceV files on a USB flash drive and run the application from it.

1. Double-click **CylanceV.exe**. The end-user license agreement (EULA) displays.
2. Click **Accept**. A message displays stating you must enter the authentication key.
3. Click **OK**, enter the authentication key, then click **Save**.
4. Define a path to a folder where you want the scan to start. Either type a path in the Start Folder field or click the browse icon and select a folder.
5. Click the start button (blue arrow) to start the scan. The status bar at the bottom updates with information about safe, abnormal, unsafe, and total analyzed file counts. When the scan completes, the status in the lower-left will display idle.
6. When the scan completes, you can do the following:
   - Clear File List - Clears the list of files analyzed
   - Export to CSV - Exports the list to a .csv file; you can then import the file into another program or view the contents using a spreadsheet application
   - Select Categories - Select the columns visible in the GUI; this also affects what information appears in the .csv file
   - Start - Starts a scan; during a scan, the icon changes to a stop button
   - Start Watching - Scans the files in a folder and then analyzes any files added to the folder
   - Recheck Unscored Files - If any of the analyzed files does not have a Cylance score, clicking Recheck Unscored Files will reanalyze the unscored files
   - Automatically submit unknown samples to Cylance for analysis - Enabled - When enabled, any file that is scanned and has not been analyzed by Cylance cloud will be uploaded for analysis

CylanceV settings

To change the CylanceV settings, go to **Options > Settings**.

Settings

Item	Description
Authentication key	Enter the authentication key
Available local models	Select the local model • ELF - The local model for ELF analysis • Mach-O - The local model for Mach-O analysis • OLE - The local model for Microsoft Office Document analysis • OOXML - The local model for Office Open XML document analysis • PE - The local model for portable executable analysis • PDF - The local model for PDF analysis

File inspection

Item	Description

Files to inspect	• Executables - Analyzes executable files, for Windows, macOS, and Linux, for malicious payloads; for example, malware contained within the executable file
	• Documents
	  • Office binary - Analyzes Microsoft Office documents for malicious payloads; for example, malware contained within a document
	  • Office XML - Analyzes Microsoft Office Open XML documents for malicious playloads; for example, malware contained within a document
	• Signatures - Performs signature validation when inspecting files

Cylance file analysis	• Max file size to submit to Infinity - This is the maximum file size allowed to upload to Cylance cloud server (in MB). CylanceV will only upload files that the Cylance cloud has not analyzed before. The maximum file size to submit to the Cylance cloud server is 50MB and the minimum is 1MB. The default setting is 50MB. • Max file size to examine - This is the maximum file size that CylanceV will analyze (in MB). Files larger than this will not be examined. The maximum file size to examine localy is 1500MB and the minimum is 1MB. The default setting is 50MB. • Valid file types to submit to Infinity - If everything is unchecked, then no files are uploaded to the Cylance cloud. • Windows executable files - Select to submit Windows executable files to the Cylance cloud for analysis if Cylance has never seen this file before

Alerts

Item	Description
Detection threshold	This sets the range for the Cylance cloud score for abnormal files. Increasing or decreasing the threshold for abnormal files also affects the ranges for safe and unsafe files. Enter a number between 0.00 and 0.95, or use the slider to change the range.
Syslog (Watcher only)	• Send unsafe file detection events to syslog server - This option to configure server settings to have unsafe file detection events sent to a syslog server. • Server - Enter the IP address for the syslog server. Example: 123.45.67.89 • Port - Enter the port number for the syslog server. • Facility - The syslog facility, an information field in the syslog message to provide a general idea of what part of the system the message originated from. • Severity - Select the event to log. Examples: Selecting Emergency (0) results in only severe events being logged, while selecting Debug (7) means all events are logged. The more information added to the log file, the larger the log file could get, depending upon the number of events. Enabling debug logging will result in the largest log files.

Proxy config

If you are using a proxy on your network, you need to configure CylanceV with your proxy information to allow CylanceV to verify the authentication key.

Item	Description
No proxy	This setting means there is no proxy used on your network. This is the default setting.
Auto-detect proxy settings	This setting allows CylanceV to detect the proxy settings.
Manually specify proxy settings	This allows you to configure your proxy settings. The user, password, and domain are not needed if no authorization is selected. • Proxy type - Select no auth, basic, digest, or NTLM • Proxy URI - Enter the proxy URI, which includes the scene; for example, http://<proxy-address> or https://<proxy-address> • User - Enter the user name for the proxy • Password - Enter the password for the proxy user • Domain - Enter the domain name for the proxy server
Ignore certificate validation failure	This setting means CylanceV will ignore any certificate validation failure messages.

Certificates

Occasionally certificates might be compromised. You can configure CylanceV to alert on files signed with a compromised certificate.

Item	Description
Name	The name of the compromised certificate
Thumb print	The thumbprint value for the compromised certificate

Archive files

CylanceV will attempt to analyze archive files. If the archives are protected with a password, enter the password here to have CylanceV analyze those archives. When entering multiple passwords, enter one password per line; use the Enter key to add a new line for another password.

Note: Remember to save your password list.

Watcher

Configure CylanceV to watch for new files that are added to a specific directory. You can also configure what happens to safe, unsafe, and abnormal file types.

Using centroids

Cylance uses the Cylance cloud, which collects threat data, trains and learns from that threat data, and calculates likely outcomes based on what it sees. This same collection of threat data is used by CylanceV as local models. Because these models are local, do not update frequently, and CylanceV does not connect to Cylance cloud, there may be times when users could benefit from adding new samples along with the local models. This can be done using centroids.

"Centroid" means "center of the group." A Cylance centroid is built and based on a representative sampling of the file the centroid is intended to impact. Based on the select grouping of file characteristics, centroids allow or block execution of that grouping. By using a grouping, files that fall within a tolerance range are considered part of the group.

Note: With centroids enabled, CylanceV will check for new centroids on startup. New centroids will be downloaded and may cause a slight delay when CylanceV starts.

1. Make sure you have the centroids.xml file. CylanceV can only use one centroids.xml file at a time. If you do not have the centroids.xml file, you can create an empty file and name it centroids.xml.
2. Open the CylanceV.exe.config file using a text editor. The configuration file should be in the same folder as the CylanceV.exe file.
3. Under CylanceVSettings, in the Analysis section, look for **selectedCentroidFile=""**.
4. Add the absolute path to the centroids.xml file. For example, selectedCentroidFile="C:\CylanceV \centroids.xml".
5. Save the configuration file.
6. Run CylanceV. The centroid.xml file will be updated with the latest centroid information.

CylanceV logging

You can adjust the logging level by selecting **Options > Logging**, then select a log level.

Item	Description
None	Only license authentication and validation information are logged; no other log information is collected
Standard	Only warning messages are logged; no other log information is collected
Debug	Collects information and warning messages; this could be a large log file
All	All log messages are collected; this could be a large log file

File status results

The file status (unsafe, safe, etc.) is displayed in the Results pane. You can select any of the files in the pane and right-click for more options.

Item	Description
Properties	Displays the file properties (the same as right-clicking a file and selecting Properties)
Show in folder	Opens the folder containing the file
View hashes	Displays the hash information for the file, if available; could include MD5, SHA1, and SHA256 hashes
Check with VirusTotal	Opens a web browser and displays the file information on the VirusTotal website
Search Google	Displays submenu options: File, MD5, SHA1, and SHA256; selecting an option opens a web browser and displays search results for the selected file
Remove from list	Removes the file from the results list
Delete files	Deletes the file from the device
Export to csv	Exports the file results to a .csv file; you can select multiple files to export to a .csv file
Export samples	Compresses the selected file to a .zip format and password protects it (password = infected); you can select multiple files to export to a compressed file
Export threat indicators	Exports the file information to a .xml file; you can select multiple files to export to a .xml file

Using CylanceVCL

CylanceVCL provides a command-line option instead of the CylanceV GUI.

1. Open a command prompt, then navigate to the directory containing the CylanceVCL.exe file.
2. Type CylanceVCL.exe -h, then press Enter to see the command-line options.
3. Type CylanceVCL.exe -k <key> -p c:\test -T 2, then press Enter. You must crate the c:\test folder before you can scan it. You can replace c:\test with any folder you want to scan.
4. When the scan completes, a .csv file is created in the same folder as the CylanceVCL.exe file. The .csv file has the date and time the scan completed in the file name. For example, CylanceV2_2020_11_30T23_30_30Z.csv.

Command-line syntax

The following table contains the command-line syntax for the CylanceVCL executable. Executing CylanceV command-line without any command-line arguments will result in a similar list to help you.

Item	Description
-h, -?, --help	Displays help similar to this table
-k, --key=VALUE	Authentication key; this is required
-p, --path=VALUE	The path to the folder to scan; this is required
-T, --type=VALUE	These are the types of files to analyze; this is required Types are specified as a bit field, which may be combined For example, using -type=7 would be a valid combined type of 1, 2, and 4 Valid types are: • 1 - Check if signed files have valid signatures; this is the same as the -S command • 2 - Portable executable files (PE) • 4 - Portable document format files (PDF) • 8 - Microsoft Office Document files (OLE) • 16 - Office Open XML Document files (OOXML) • 32 - Mach Object files (Mach-O) • 64 - ELF files (Linux)
-f, --file=VALUE	Name of the input file
-A, --avindustry	Get AV industry results
-t, --threatindicators=VALUE	Create a .xml report with threat indicators for all files with Cylance scores less than the given value
-e, --export=VALUE	Create an archive of all files with a Cylance score less than the given value; archived file password = infected

Item	Description
-a, --autoupload	Auto-upload the requested samples to the Cylance cloud
-S, --signature	Check if signed files have valid signatures; this is the same as the -T 1 command **Note:** This parameter is deprecated and included for backward compatibility only; the -T option is preferred.
-x, --examine=VALUE	Maximum size of files to be analyzed (in MB) • Default = 50 • Min = 1 • Max = 1500
-1, --sha1	Enables generation of SHA1 hashes for files
-5, --md5	Enables generation of MD5 hashes for files
-o, --output=VALUE	Output directory path; default '.'
-l, --log=VALUE	Logging level; range is from 0 to 3 (default = 1) • 0 - None • 1 - Standard • 2 - Debug • 3 - All
-m, --model=VALUE	Use the specified local model files instead of auto-selecting the most up-to-date model **Note:** Specify model types with a comma-separated list (no spaces).

Command-line example

To scan all executables on a system, calculate the MD5 hash of each file, and export threat indicators for each file, the command-line would look as follows:

```
CylanceVCL.exe -k <key> -T2 -t -5 -p c:\
```

- -t - Export threat indicators for files convicted as malware
- -5 - Generate MD5 hashes for all files
- -p - Scan the path specified after this switch; in the above example, this is c:\

Threat indicator list

Anomalies

These indicators represent situations where the object has elements that are inconsistent or anomalous in some way. Frequently these are inconsistencies in structural elements in the file.

Item	Description
16bitSubsystem	This object utilizes the 16 bit subsystem. Malware uses this to exist in a less secure and less monitored part of the operating system, and frequently to perform privilege escalation attacks.
Anachronism	This PE appears to be lying about when it was written, which is atypical for professionally written software.
AppendedData	This PE has some extra content appended to it, beyond the normal areas of the file. Appended data can frequently be used to embed malicious code or data and is frequently overlooked by protection systems.
AutoitDbgPrivilege	AutoIt script is capable of performing debug activities.
AutoitManyDllCalls	AutoIt script uses many external DLL calls. AutoIt runtime already has many common functions, therefore using additional functionality from external DLLs may be a sign of maliciousness.
AutoitMutex	AutoIt script creates synchronization objects. This is often used by malware to prevent multiple infection of the same target.
AutoitProcessCarving	AutoIt script is likely performing process carving to run code that appears to come from another process. This is often done to hinder detection.
AutoitProcessInjection	AutoIt script is likely performing process injection to run code in other processes context to possibly stay undetected or steal data.
AutoitRegWrite	AutoIt script writes into the Windows registry.
Base64Alphabet	This object contains evidence of usage of BASE64 Encoding of an alphabet. Malware does this to attempt to avoid common detection, or to attack other programs using BASE64 encoding.

Item	Description
CommandlineArgsImport	This sample imports functions that can be used to read arguments from a command line. Malware uses this to collect information on subsequent runs.
ComplexMultipeFilters	The document contains multiple streams with multiple filters.
ComplexObfuscatedEncoding	The document contains an anomalously high number of obfuscated names.
ComplexUnsupportedVersionEmbeddedFiles	The document uses EmbeddedFiles features from newer versions of the PDF standard than the document declares.
ComplexUnsupportedVersionFlate	The document uses the FlateDecode feature from newer versions of the PDF standard than the document declares.
ComplexUnsupportedVersionJbig2	The document uses the JBIG2Decode feature from newer versions of the PDF standard than the document declares.
ComplexUnsupportedVersionJs	The document uses JavaScript features from newer versions of the PDF standard than the document declares.
ComplexUnsupportedVersionXFA	The document uses XFA features from newer versions of the PDF standard than the document declares.
ComplexUnsupportedVersionXobject	The document uses XOBject features from newer versions of the PDF standard than the document declares.
ContainsFlash	The document contains flash objects.
ContainsPE	Indicates embedded executable files.
ContainsU3D	The document contains U3D objects.
InvalidCodePageUsed	The document uses an invalid or unrecognized locale, possibly to avoid detection.
InvalidData	The document metadata is obviously bogus or corrupt.
InvalidStructure	The document structure is not valid - sizes, metadata, or internal sector allocation table is wrong. May be indicative of an exploit.

Item	Description
ManifextMismatch	This object demonstrates an inconsistency in its manifest. Malware does this to avoid detection, but rarely covers its tracks deeply.
NontrivialDLLEP	This PE is a DLL with a nontrivial entry point. This is common among DLLs, but a malicious DLL may use its entry point to take up residence in a process.
NullValuesInStrings	Some strings within the document contain null-characters in the middle.
PDFParserArraysContainsNullCount	The document contains an anomalously high number of Null values in arrays.
PDFParserArraysHeterogeneousCount	The document contains an anomalously high number of arrays containing different types of elements.
PDFParserMailtoURICount	The document contains an anomalously high number of email links (mailto:).
PDFParserMinPageCount	The document has an unusual structure of page objects - a high number of child page objects per node.
PDFParserNamesPoundNameMaxLength	The document may attempt to obfuscate its contents by using long encoded strings.
PDFParserNamesPoundNameMinLength	The document contains an anomalously high minimal length of an escaped name.
PDFParserNamesPoundNameTotalLength	The document may attempt to obfuscate its contents by storing much of its content in encoded strings.
PDFParserNamesPoundNameUpperCount	The document contains an anomalously high number of names escaped with uppercase hexadecimal characters.
PDFParserNamesPoundNameValidCount	The document contains an anomalously high number of valid escaped names.
PDFParserNamesPoundPerNameMaxCount	The document contains an anomalously high max number of escaped characters per single name.
PDFParserNamesPoundUnnecessaryCount	The document contains an anomalously high number of unnecessarily escaped names.
PDFParserNumbersLeadingDigitTallies8	The document contains an anomalously high number of numbers that start with 8 in decimal representation.

Item	Description
PDFParserNumbersPlusCount	The document contains an anomalously high number of numbers with explicit plus sign.
PDFParserNumbersRealMaxRawLength	The document contains an anomalously high max length of a real number.
PDFParserPageCounts	The document contains an anomalously high number of child page objects.
PDFParserPageObjectCount	The document contains an anomalously high number of page objects.
PDFParserSizeEOF	The document contains an anomalously long end of file sequence(s).
PDFParserStringsHexLowerCount	The document contains an anomalously high number of strings escaped with lowercase hexadecimal digits.
PDFParserStringsLiteralStringMaxLength	The document contains an anomalously high max length of a literal string.
PDFParserStringsOctalZeroPaddedCount	The document contains an anomalously high number of octal escaped characters in strings that are unnecessarily zero-padded.
PDFParserTrailerSpread	The document contains an anomalously large spread between trailer objects.
PDFParserWhitespaceCommentMaxLength	The document contains an anomalously high max length of a comment.
PDFParserWhitespaceCommentMinLength	The document contains unusual short comments that are not used by reader software.
PDFParserWhitespaceCommentTotalLength	The document contains an unusually large amount of commented out data.
PDFParserWhitespaceEOL0ACount	The document contains an anomalously high number of short end-of-line characters.
PDFParserWhitespaceWhitespace00Count	The document contains an anomalously high number of zero-bytes used as whitespace.
PDFParserWhitespaceWhitespace09Count	The document contains an anomalously high number of 09 bytes used as whitespace.
PDFParserWhitespaceWhitespaceLongestRun	The document contains an anomalously long whitespace area.

Item	Description
PDFParserWhitespaceWhitespaceTotalLength	The document contains an anomalously high amount of whitespace.
PDFParseru3DObjectsNamesAllNames	The document contains an anomalously high number of u3d objects.
PossibleBAT	This object contains evidence of having a standard windows batch file included. Malware does this to avoid common scanning techniques and to provide persistence.
PossibleDinkumware	This object shows evidence of including some components from DinkumWare. Dinkumware is frequently used in various malware components.
PropertyImpropriety	Reports suspicious OOXML properties.
RaiseExceptionImports	This object imports functions used to raise exceptions within a program. Malware does this to implement tactics that make standard dynamic code analysis difficult to follow.
ReservedFieldsViolation	Document violates specification in terms of reserved fields usage.
ResourceAnomaly	This object contains an anomaly in the resource section. Malware frequently contains malformed or other odd bits in the resource section of a DLL.
RWXSection	This PE may contain modifiable code, which is at best unorthodox and at worst symptomatic of a virus infection. Frequently, this feature implies that the object has been built using something other than a standard compiler, or has been modified after it was originally built.
SectorMalfeasance	Reports structural oddities with OLE sector allocation.
StringInvalid	One of the references to a string in a string table pointed to a negative offset.
StringTableNotTerminated	A string table was not terminated with a null byte. This could cause a fault at runtime due to a string that does not end.
StringTruncated	One of the references to a string in a string table pointed after end of file.

Item	Description
SuspiciousPDataSection	This PE is hiding something in its "pdata" area, and we're not sure what. The pdata section in a PE file is generally used for process runtime structures, but this particular object contains something else.
SuspiciousRelocSection	This PE is hiding something in its "relocations" area, and we're not sure what. The relocations area in a PE file is generally used for relocating particular symbols, but this particular object contains something else.
SuspiciousDirectoryNames	OLE directory names associated with badness.
SuspiciousDirectoryStructure	Reports oddities in the OLE directory structure.
SuspiciousEmbedding	Reports suspicious embedding of OLE.
SuspiciousVBA	Reports suspicious VBA code.
SuspiciousVBALib	Reports suspicious VBA library usage.
SuspiciousVBANames	Reports suspicious names associated with VBA structures.
SuspiciousVBAVersion	Reports suspicious VBA versioning.
SWFOddity	Reports certain usages of embedded SWF.
TooMalformedToProcess	Document is so malformed that it could not be parsed completely.
VersionAnomaly	This object has issues with how it presents its version information. Malware does this to avoid detection.

Collection

These indicators represent situations where the object has elements that indicate capabilities or evidence of collecting data. This can include enumeration of system configuration or collection of specific sensitive information.

Item	Description
BrowserInfoTheft	This object contains evidence of an intent to read passwords stored in browser caches. Malware uses this to collect the passwords for exfiltration.
CredentialProvider	This object contains evidence of interaction with a credential provider, or the desire to appear as one. Malware does this as credential providers get access to many types of sensitive data, such as usernames and passwords, and by acting as one they may be able to subvert the authentication integrity.

Item	Description
CurrentUserInfoImports	This object imports functions that are used to gather information about the currently logged in user. Malware uses this to determine paths of action to escalate privileges and to better tailor attacks.
DebugStringImports	This object imports functions that are used to output debug strings. Typically this is disabled in production software, but left on in malware that is being tested.
DiskInfoImports	This object imports functions that can be used to gather details about volumes on the system. Malware uses this in conjunction with listing to determine facts about the volumes in preparation for a further attack.
EnumerateFileImports	This object imports functions that are used to list files. Malware uses this to look for sensitive data, or to find further points of attack.
EnumerateModuleImports	This object imports functions that can be used to list all of the DLLs a running process uses. Malware uses this capability to locate and target specific libraries for loading into a process, and to map out a process it wishes to inject into.
EnumerateNetwork	This object demonstrates evidence of a capability to attempt to numerate connected networks and network adapters. Malware will do this to determine where a target system lies in relation to others, and to look for possible lateral paths.
EnumerateProcessImports	This object imports functions that can be used to list all of the running processes on a system. Malware used this capability to locate processes to inject into or those that it wishes to delete.
EnumerateVolumeImports	This object imports functions that can be used to list the volumes on the system. Malware uses this to find all of the areas it might need to search for data, or to spread an infection.
GinaImports	This object imports functions that are used to access Gina. Malware does this to attempt to breach the secure ctrl-alt-delete password entry system or other network login functions.
HostnameSearchImports	This object imports functions that are used to gather information about hostnames on the network and the hostname of the machine itself. Malware uses this capability to better target further attacks or scan for new targets.
KeystrokeLogImports	This object imports functions that can capture and log keystrokes from the keyboard. Malware uses this to capture and save keystrokes to find sensitive information such as passwords.
OSInfoImports	This object imports functions that are used to gather information about the current operating system. Malware uses this to determine how to better tailor further attacks and to report information back to a controller.
PossibleKeylogger	This object contains evidence of keylogger type activity. Malware uses keyloggers to collect sensitive information from the keyboard.

Item	Description
PossiblePasswords	This object has evidence of including common passwords, or structure that would enable brute forcing common passwords. Malware uses this to attempt to further penetrate a network by accessing other resources via password.
ProcessorInfoWMI	This object imports functions that can be used to determine details about the processor. Malware uses this to tailor attacks and exfiltrate this data to common command and control infrastructure.
RDPUsage	This object shows evidence of interacting with the Remote Desktop Protocol (RDP). Malware frequently uses this to move laterally and to offer direct command and control functionality.
SpyString	Indicates possible spying on clipboard or user actions via accessibility API usage.
SystemDirImports	This object imports functions used to locate the system directory. Malware does this to find where many of the installed system binaries are located, as it frequently hides among them.
UserEnvInfoImports	This object imports functions that are used to gather information about the environment of the current logged in user. Malware uses this to determine details about the logged in user and look for other intelligence that can be gleaned from the environment variables.

Data loss

These indicators represent situations where the object has elements that indicate capabilities or evidence of exfiltration of data. This can include outgoing network connections, evidence of acting as a browser, or other network communications.

Item	Description
AbnormalNetworkActivity	This object implements a non-standard method of networking. Malware does this to avoid detection of more common networking approaches.
BrowserPluginString	Indicates capability to enumerate or install browser plugins.
ContainsBrowserString	This object contains evidence of attempting to create a custom UserAgent string. Malware frequently uses common UserAgent strings to avoid detection in outgoing requests.
DownloadFileImports	This object imports functions that can be used to download files to the system. Malware uses this as both a way to further stage an attack and to exfiltrate data via the outbound URL.
FirewallModifyImports	This object imports functions used to modify the local windows firewall. Malware uses this to open holes and avoid detection.

Item	Description
HTTPCustomHeaders	This object contains evidence of the creation of other custom HTTP headers. Malware does this to facilitate interactions with command and control infrastructures and to avoid detection.
IRCCommands	This object contains evidence of interaction with an IRC server. Malware commonly uses IRC to facilitate a command and control infrastructure.
MemoryExfiltrationImports	This object imports functions that can be used to read memory from a running process. Malware uses this to determine proper places to insert itself, or to extract useful information from a running process's memory, like passwords, credit cards, or other sensitive information.
NetworkOutboundImports	This object imports functions that can be used to send data out to the network or the general internet. Malware uses this as a method for exfiltration of data or as a method for command and control.
PipeUsage	This object imports functions that allow the manipulation of named pipes. Malware uses this as a method of communication, and of data exfiltration.
RPCUsage	This object imports functions that allow it to interact with Remote Procedure Call (RPC) infrastructure. Malware uses this to spread, or to send data to remote systems for exfiltration.

Deception

These indicators represent situations where the object has elements that indicate capabilities or evidence of an object attempting to be deceptive. Deception can come in the form of hidden sections, inclusion of code to avoid detection, or indications that it is labeled improperly in metadata or other sections.

Item	Description
AddedHeader	Document contains an add-obfuscated PE header that may be a hidden malicious payload.
AddedKernel32	Document contains an add-obfuscated reference to kernel32.dll – a library that may be used by malicious payload.
AddedMscoree	Document contains an add-obfuscated reference to mscoree.dll – a library that may be used by malicious payload.
AddedMsvbvm	Document contains an add-obfuscated reference to msvbvm – a library that may be used by malicious payload compiled for VB6.

Item	Description
AntiVM	This object demonstrates features that can be used to determine if the process is running in a virtual machine. Malware does this to avoid running in virtualized sandboxes that are becoming more common.
AutoitDownloadExecute	AutoIt script is capable of downloading and executing files. This is often done to deliver additional malicious payloads.
AutoitObfuscationStringConcat	AutoIt script is likely obfuscated with string concatenation. This is often done to avoid detection of (whole) suspicious commands.
AutoitShellcodeCalling	AutoIt script uses CallWindowProc winapi function that may be indicative of injecting shellcode.
AutoitUseResources	AutoIt script uses data from resources stored alongside with the script. Malware often stores important parts of itself as resource data and unpacks them in runtime – therefore this looks suspicious.
CabinentUsage	This object shows evidence of containing a CAB file. Malware does this to package sensitive components in a way that many detection systems can't see.
ClearKernel32	Document contains reference to kernel32.dll – a library that may be used by malicious payload.
ClearMscoree	Document contains reference to mscoree.dll – a library that may be used by malicious payload.
ClearMsvbvm	Document contains reference to msvbvm – a library that may be used by malicious payload compiled for VB6.
ComplexInvalidVersion	The document declares the wrong PDF version.
ComplexJsStenographySuspected	The document may contain JavaScript code hidden in literal strings.

Item	Description
ContainsEmbeddedDocument	This object contains a document embedded inside the object. Malware can use this to spread an attack to multiple sources, or to otherwise hide its true form.
CryptoKeys	This object contains evidence of having an embedded cryptographic key. Malware does this to avoid detection and perhaps as authentication with remote services.
DebugCheckImports	This object imports functions that would allow it to act like a debugger. Malware uses this capability to read and write from other processes.
EmbeddedPE	This PE has additional PEs within it, which is usually only the case with software installation programs. Frequently malware will embed a PE file that it then drops to disk and executes. This technique is often used to avoid protection scanners by packaging binaries in a format that the underlying scanning technology doesn't understand.
EncodedDosStub1	Document contains an obfuscated PE DOS stub that may belong to a hidden malicious payload.
EncodedDosStub2	Document contains an obfuscated PE DOS stub that may belong to a hidden malicious payload.
EncodedPE	This PE has additional PEs hidden within it, which is extremely suspicious. Similar to the last, but uses an encoding scheme to attempt to further hide the binary inside the object.
ExecuteDLL	This object contains evidence of the capability to execute a DLL using common methods. Malware does this as a method to avoid common detection practices.
FakeMicrosoft	This PE claims to be written by Microsoft, but it doesn't look like a Microsoft PE. Malware commonly masquerades as Microsoft PEs in order to look inconspicuous.

Item	Description
HiddenMachO	Has another MachO executable file within, which is not properly declared. This may be an attempt to hide the payload from being easily detected.
HTTPCustomUserAgent	This object contains evidence of manipulation of the browser UserAgent. Malware does this to facilitate interactions with command and control infrastructures and to avoid detection.
InjectProcessImports	This PE has the ability to inject code into other processes. This capability frequently implies that a process is attempting to be deceptive or hostile in some way.
InvisibleEXE	This PE appears to run invisibly, but it isn't a background service. It might be designed to remain hidden.
JSTokensSuspicious	The document contains unusually suspicious JavaScript.
MSCertStore	This object shows evidence of interacting with the core windows certificate store. Malware does this to collect credentials and insert rogue keys into the stream to facilitate things like man in the middle attacks.
MSCryptoImports	This object imports functions to use the core windows crypto library. Malware will use this to leverage the locally installed cryptography so it doesn’t need to carry its own around.
PDFParserDotDotSlash1URICount	The document may attempt path traversal using relative paths like "../".
PDFParserJSStreamCount	The document contains an unusually high number of JavaScript-related streams.
PDFParserJSTokenCounts0cumulativesum	The document contains an anomalously high number of JavaScript tokens.
PDFParserJavaScriptMagicseval~28	The document may contain obfuscated JavaScript or can run dynamically loaded JavaScript with eval().

Item	Description
PDFParserJavaScriptMagicsunescape~28	The document may contain obfuscated JavaScript.
PDFParserNamesAllNamesSuspicious	The document contains an anomalously high number of suspicious names.
PDFParserNamesObfuscatedNamesSuspicious	The document contains an anomalously high number of obfuscated names.
PDFParserPEDetections	The document contains embedded PE file(s).
PDFParserSwfObjectsxObservationsxSWFObjectsversion	The document contains an SWF object with an unusual version number.
PDFParserSwfObjectsxObservationsxSWFObjectsxZLibcmf	The document contains an SWF object with unusual compression parameters.
PDFParserjsObjectsLength	The document contains an anomalously high number of individual JavaScript scripts.
PDFParserswfObjectsxObservationsxSWFObjectsxZLibflg	The document contains an SWF object with unusual compression flag parameters.
PE_ClearDosStub1	Document contains a DOS stub – indicative of PE file inclusion.
PE_ClearDosStub2	Document contains a DOS stub – indicative of PE file inclusion.
PE_ClearHeader	Document contains PE file header data that does not belong in the document structure.
PEinAppendedSpace	Document contains a PE file that does not belong in the document structure.
PEinFreeSpace	Document contains a PE file that does not belong in the document structure.
ProtectionExamination	This object seems to be looking for common protection systems. Malware does this to initiate an anti-protection action tailored to that installed on the system.

Item	Description
SegmentSuspiciousName	A segment has either an invalid string as a name, or an unusual non-standard name. This may indicate post-compilation tampering or use of packers or obfuscators.
SegmentSuspiciousSize	Segment size is significantly different from size of all content (sections) within. This may indicate usage of unreferenced area, or reservation of space for runtime unpacking of malicious code.
SelfExtraction	This object seems to be a self-extracting archive. Malware frequently uses this tactic to obfuscate their true intentions.
ServiceDLL	This object seems to be a service DLL. Service DLL's are loaded in svchost.exe processes and are a common persistence methodology for malware.
StringJsSplitting	The document contains suspicious JS tokens.
SWFinAppendedSpace	Document contains a Shockwave flash object that does not belong in the document structure.
TempFileImports	This object imports functions used to access and manipulate temporary files. Malware does this as temporary files tend to avoid detection.
UsesCompression	This object seems to have portions of the code that appear to be compressed. Malware uses these techniques to avoid detection.
VirtualProtectImports	This object imports functions that are used to modify the memory of a running process. Malware does this to inject itself into running processes.
XoredHeader	Document contains a xor-obfuscated PE header that may be a hidden malicious payload.
XoredKernel32	Document contains a xor-obfuscated reference to kernel32.dll – a library that may be used by malicious payload.

Item	Description
XoredMscoree	Document contains a xor-obfuscated reference to mscoree.dll – a library that may be used by malicious payload.
XoredMsvbvm	Document contains a xor-obfuscated reference to msvbvm – a library that may be used by malicious payload compiled for VB6.

Destruction

These indicators represent situations where the object has elements that indicate capabilities or evidence of destruction. Destructive capabilities include the ability to delete system resources like files or directories.

Item	Description
action_writeByte	VBA script within the document is likely writing bytes to a file – which is an unusual action for legitimate documents.
action_hexToBin	VBA script within the document is likely using hexadecimal to binary conversion that may indicate decoding a hidden malicious payload.
appended_URI	The document contains a link that does not belong in the document structure.
appended_exploit	The document contains suspicious data outside of document structure that may be indicative of an exploit.
appended_macro	The document contains a macro script that does not belong in the document structure.
appended_90_nopsled	The document contains a nop-sled that does not belong in the document structure – this is almost certainly there to facilitate exploitation.
AutorunsPersistence	This object attempts to interact with common methods of persistence (startup scripts, etc). Malware commonly uses these tactics to attain persistence.
DestructionString	Has capabilities to kill processes or shutdown the machine via shell commands.
FileDirDeleteImports	This PE imports functions that can be used to delete Files or Directories. Malware uses this to break systems and cover its tracks.
JsHeapSpray	The document likely contains heap spray code.
PossibleLocker	This object demonstrates evidence of a desire to lock out common tools by policy. Malware will do this to retain persistence and make detection and cleanup more difficult.

Item	Description
RegistryManipulation	This object imports functions that are used to manipulate the windows registry. Malware does this to attain persistence, avoid detection, and for many other reasons.
SeBackupPrivilege	This PE might attempt to read files to which it has not been granted access. SeBackup privilege allows access to files without honoring access controls. It is frequently used by programs that handle backups, and is frequently limited to administrative users, but can be used maliciously to get access to specific elements that might otherwise be difficult.
SeDebugPrivilege	This PE might attempt to tamper with system processes. SeDebug Privilege is used to access processes other than your own and is frequently limited to administrative users. It is often paired with reading and writing to other processes.
SeRestorePrivilege	This PE might attempt to change or delete files to which it has not been granted access. The pair to SeBackup, SeRestore privilege allows writing without consideration of access control.
ServiceControlImports	This object imports functions that can control windows services on the current system. Malware uses this to either launch itself into the background via installing as a service, or to disable other services that may have a protective function.
SkylinedHeapSpray	The document contains an unmodified version of skylined heap spray code.
SpawnProcessImports	This PE imports functions that can be used to spawn another process. Malware uses this to launch subsequent phases of an infection, typically downloaded from the Internet.
StringJsExploit	The document contains JavaScript code that is likely capable of exploitation.
StringJsObfuscation	The document contains JavaScript obfuscation tokens.
TerminateProcessImports	This object imports functions that can be used to stop a running process. Malware uses this to attempt to remove protection systems, or to cause damage to a running system.
trigger_AutoExec	VBA script within the document is likely trying to execute automatically.
trigger_AutoOpen	VBA script within the document is likely trying to execute as soon as the document is opened.
trigger_Document_Open	VBA script within the document is likely trying to execute as soon as the document is opened.

Item	Description
trigger_DocumentOpen	VBA script within the document is likely trying to execute as soon as the document is opened.
trigger_AutoExit	VBA script within the document is likely trying to execute automatically when the document is closing.
trigger_AutoClose	VBA script within the document is likely trying to execute automatically when the document is closing.
trigger_Document_Close	VBA script within the document is likely trying to execute automatically when the document is closing.
trigger_DocumentBeforeClose	VBA script within the document is likely trying to execute automatically just before the document closes.
trigger_DocumentChange	VBA script within the document is likely trying to execute automatically when the document is being changed.
trigger_AutoNew	VBA script within the document is likely trying to execute automatically when a new document is being created.
trigger_Document_New	VBA script within the document is likely trying to execute automatically when a new document is being created.
trigger_NewDocument	VBA script within the document is likely trying to execute automatically when a new document is being created.
trigger_Auto_Open	VBA script within the document is likely trying to execute as soon as the document is opened.
trigger_Workbook_Open	VBA script within the document is likely trying to execute as soon as an Excel workbook is opened.
trigger_Auto_Close	VBA script within the document is likely trying to execute automatically when the document is closing.
trigger_Workbook_Close	VBA script within the document is likely trying to execute automatically when an Excel workbook is closing.
UserManagementImports	This object imports functions that can be used to change users on the local system. It can add, delete, or change key user details. Malware can use this capability to achieve persistence or cause harm to the local system.
VirtualAllocImports	This object imports functions that are used to create memory in a running process. Malware does this to inject itself into a running process.

Shellcodes

Indicates a small piece of code used as the payload in the exploitation of a software vulnerability. It is called "shellcode" because it typically starts a command shell from which the attacker can control the compromised machine, but any piece of code that performs a similar task can be called shellcode.

Item	Description
ApiHashing	Document contains a byte sequence that looks like shellcode that tries to stealthily find library APIs loaded in memory.
BlackholeV2	The document looks like it might have come from the Blackhole exploit kit.
ComplexGotoEmbed	The document may be able to force a browser to go to an address or perform an action.
ComplexSuspiciousHeader	PDF header located at non-zero offset which may indicate an attempt to prevent this document from being recognized as a PDF document.
EmbeddedTiff	The document may contain a crafted tiff image with nopsled to facilitate exploitation.
EmbeddedXDP	The document likely contains another PDF as an xml (XDP).
FindKernel32Base1	The document contains a byte sequence that looks like a shellcode that tries to locate kernel32.dll in memory.
FindKernel32Base2	The document contains a byte sequence that looks like a shellcode that tries to locate kernel32.dll in memory.
FindKernel32Base3	The document contains a byte sequence that looks like a shellcode that tries to locate kernel32.dll in memory.
FunctionPrologSig	The document contains a byte sequence that is a typical function prolog - likely contains shellcode.
GetEIP1	The document contains a byte sequence that looks like a shellcode that resolves its own address to locate other things in memory and facilitate exploitation.
GetEIP4	The document contains a byte sequence that looks like a shellcode that resolves its own address to locate other things in memory and facilitate exploitation.
IndirectFnCall1	The document contains a byte sequence that looks like an indirect function call – likely shellcode.
IndirectFnCall2	The document contains a byte sequence that looks like an indirect function call – likely shellcode.
IndirectFnCall3	The document contains a byte sequence that looks like an indirect function call – likely shellcode.
SehSig	The document contains a byte sequence that is typical for Structured Exception Handling – likely contains shellcode.

Item	Description
StringLaunchActionBrowser	The document may be able to force a browser to go to an address or perform an action.
StringLaunchActionShell	The document may be able to execute shell actions.
StringSingExploit	The document might contain an exploit.

Misc

All indicators that do not fit into the aforementioned categories.

Item	Description
AutoitFileOperations	AutoIt script is capable of performing multiple actions on files. This may be used for information gathering, persistence, or destruction.
AutorunString	Indicates capability to achieve persistence by using autorun mechanism(s).
CodepageLookupImports	This object imports functions used to look up the codepage (location) of a running system. Malware uses this to differentiate which country/region a system is running in to better target particular groups.
MutexImports	This object imports functions to create and manipulate Mutex objects. Malware frequently uses mutexes to avoid infecting a system multiple times.
OpenSSLStatic	This object contains a version of openSSL compiled to appear stealthy. Malware will do this to include cryptography functionality without leaving strong evidence of it.
PListString	Indicates capability to interact with property lists that are used by the OS. This may be used to achieve persistence or subvert various processes.
PrivEscalationCryptBase	This object shows evidence of attempting to use a particular privilege escalation using CryptBase. Malware uses this to gain more privileges on the affected system.
ShellCommandString	Indicates capability to use sensitive shell commands for reconnaissance, elevation of privilege, or data destruction.
SystemCallSuspicious	Indicates capability to monitor and/or control system and other processes, perform debug-like actions.

Legal notice

WERE FORESEEN OR UNFORESEEN, AND EVEN IF BLACKBERRY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW IN YOUR JURISDICTION, BLACKBERRY SHALL HAVE NO OTHER OBLIGATION, DUTY, OR LIABILITY WHATSOEVER IN CONTRACT, TORT, OR OTHERWISE TO YOU INCLUDING ANY LIABILITY FOR NEGLIGENCE OR STRICT LIABILITY.

THE LIMITATIONS, EXCLUSIONS, AND DISCLAIMERS HEREIN SHALL APPLY: (A) IRRESPECTIVE OF THE NATURE OF THE CAUSE OF ACTION, DEMAND, OR ACTION BY YOU INCLUDING BUT NOT LIMITED TO BREACH OF CONTRACT, NEGLIGENCE, TORT, STRICT LIABILITY OR ANY OTHER LEGAL THEORY AND SHALL SURVIVE A FUNDAMENTAL BREACH OR BREACHES OR THE FAILURE OF THE ESSENTIAL PURPOSE OF THIS AGREEMENT OR OF ANY REMEDY CONTAINED HEREIN; AND (B) TO BLACKBERRY AND ITS AFFILIATED COMPANIES, THEIR SUCCESSORS, ASSIGNS, AGENTS, SUPPLIERS (INCLUDING AIRTIME SERVICE PROVIDERS), AUTHORIZED BLACKBERRY DISTRIBUTORS (ALSO INCLUDING AIRTIME SERVICE PROVIDERS) AND THEIR RESPECTIVE DIRECTORS, EMPLOYEES, AND INDEPENDENT CONTRACTORS.

IN ADDITION TO THE LIMITATIONS AND EXCLUSIONS SET OUT ABOVE, IN NO EVENT SHALL ANY DIRECTOR, EMPLOYEE, AGENT, DISTRIBUTOR, SUPPLIER, INDEPENDENT CONTRACTOR OF BLACKBERRY OR ANY AFFILIATES OF BLACKBERRY HAVE ANY LIABILITY ARISING FROM OR RELATED TO THE DOCUMENTATION.

Prior to subscribing for, installing, or using any Third Party Products and Services, it is your responsibility to ensure that your airtime service provider has agreed to support all of their features. Some airtime service providers might not offer Internet browsing functionality with a subscription to the BlackBerry® Internet Service. Check with your service provider for availability, roaming arrangements, service plans and features. Installation or use of Third Party Products and Services with BlackBerry's products and services may require one or more patent, trademark, copyright, or other licenses in order to avoid infringement or violation of third party rights. You are solely responsible for determining whether to use Third Party Products and Services and if any third party licenses are required to do so. If required you are responsible for acquiring them. You should not install or use Third Party Products and Services until all necessary licenses have been acquired. Any Third Party Products and Services that are provided with BlackBerry's products and services are provided as a convenience to you and are provided "AS IS" with no express or implied conditions, endorsements, guarantees, representations, or warranties of any kind by BlackBerry and BlackBerry assumes no liability whatsoever, in relation thereto. Your use of Third Party Products and Services shall be governed by and subject to you agreeing to the terms of separate licenses and other agreements applicable thereto with third parties, except to the extent expressly covered by a license or other agreement with BlackBerry.

The terms of use of any BlackBerry product or service are set out in a separate license or other agreement with BlackBerry applicable thereto. NOTHING IN THIS DOCUMENTATION IS INTENDED TO SUPERSEDE ANY EXPRESS WRITTEN AGREEMENTS OR WARRANTIES PROVIDED BY BLACKBERRY FOR PORTIONS OF ANY BLACKBERRY PRODUCT OR SERVICE OTHER THAN THIS DOCUMENTATION.

BlackBerry Enterprise Software incorporates certain third-party software. The license and copyright information associated with this software is available at http://worldwide.blackberry.com/legal/thirdpartysoftware.jsp.

BlackBerry Limited
2200 University Avenue East
Waterloo, Ontario
Canada N2K 0A7

BlackBerry UK Limited
Ground Floor, The Pearce Building, West Street,
Maidenhead, Berkshire SL6 1RL
United Kingdom

Published in Canada