MAEC™ 5.0 Specification

Vocabularies

October 9, 2017

The following sections provide object-specific listings for each of the vocabularies referenced in the object description sections. MAEC “open” vocabularies, which have type names ending in '-ov', provide a listing of common and industry accepted terms suggested to the user but do not limit the user to that defined list.

1. Analysis Conclusions

Type Name: `analysis-conclusion-ov`

The Analysis Conclusion open vocabulary is used by the following object/*property*:

- Malware Instance (`malware-instance`)
    - analysis_metadata (list of `analysis-metadata`)
        - *conclusion* (`open-vocab`)

This vocabulary is an enumeration of conclusions resulting from the analysis of a malware instance.

Vocabulary Value	Description
`benign`	As a conclusion of the analysis, the malware instance was determined to be benign.
`malicious`	As a conclusion of the analysis, the malware instance was determined to be malicious.
`suspicious`	As a conclusion of the analysis, the malware instance was determined to be suspicious.
`indeterminate`	The conclusion of the analysis was indeterminate.

2. Analysis Environment Properties

Type Name: `analysis-environment-ov`

The Analysis Environment open vocabulary is used by the following object/*property*:

- Malware Instance (`malware-instance`)
    - analysis_metadata (list of `analysis-metadata`)
        - *analysis_environment* (`open-vocab`)

This vocabulary is an enumeration of properties associated with the environment used in malware analysis.

Vocabulary Value	Description
`operating-system`	The operating system used for the dynamic analysis of the malware instance. This applies to virtualized operating systems as well as those running on bare metal. The corresponding value for this entry **MUST** be of type `object-ref` and the referenced STIX Cyber Observable Object **MUST** be of type `software`.
`host-vm`	The virtual machine used to host the guest operating system (if applicable) used for the the dynamic analysis of the malware instance. If this value is not included in conjunction with `operating-system`, this means that the dynamic analysis was performed on bare metal (i.e., without virtualization). The corresponding value for this entry **MUST** be of type `object-ref` and the referenced STIX Cyber Observable Object **MUST** be of type `software`.

`installed-software`	Any non-standard software installed on the operating system (specified through the `operating-system` value) used for the dynamic analysis of the malware instance. The corresponding value for this entry **MUST** be of type `list` and each STIX Cyber Observable Object(s) referenced in the list **MUST** be of type `software`.

3. Analysis Types

Type Name: `analysis-type-ov`

The Analysis Type open vocabulary is used by the following object/*property*:

- Malware Instance (`malware-instance`)
  - analysis_metadata (list of `analysis-metadata`)
    - *analysis_type* (`open-vocab`)

This vocabulary is an enumeration of malware analysis types.

Vocabulary Value	Description
`static`	Static malware analysis, achieved by inspecting but not executing the malware instance. For example, inspection can be done by studying memory dumps captured after the instance is run.
`dynamic`	Dynamic malware analysis, achieved by executing the malware instance (e.g., in a sandbox) and recording its behavior.
`combination`	A combination of dynamic and static malware analysis, achieved by both inspecting and executing the malware instance.

4. Behaviors

Type Name: `behavior-ov`

The Behavior open vocabulary is used in the following object/*property*:

- Behavior (`behavior`)
  - *name* (`open-vocab`)

This vocabulary is a non-exhaustive enumeration of malware behaviors.

Vocabulary Value	Description
`access-premium-service`	Accesses a premium service, such as a premium SMS service.

`autonomous-remote-infection`	Infects a remote machine autonomously, without the involvement of any end user (e.g., through the exploitation of a remote procedure call vulnerability).
`block-security-websites`	Prevents access from the system on which the malware instance is executing to one or more security vendor or security-related websites.
`capture-camera-input`	Captures data from a system's camera, including from embedded cameras (i.e. on mobile devices) and/or attached webcams.
`capture-file-system-data`	Captures data from a file system.
`capture-gps-data`	Captures GPS data from the system on which the malware instance is executing.
`capture-keyboard-input`	Captures data from the keyboard attached to the system on which the malware instance is running.
`capture-microphone-input`	Capture data from a system's microphone, including from embedded microphones (i.e. on mobile devices) and those that may be attached externally.
`capture-mouse-input`	Captures data from a system's mouse.
`capture-printer-output`	Captures data sent to a system's printer, either locally or remotely.
`capture-system-memory`	Captures data from a system's RAM.
`capture-system-network-traffic`	Captures network traffic from the system on which the malware instance is executing.
`capture-system-screenshot`	Captures images of what is currently being displayed on a system's screen, either locally (i.e. on a display) or remotely via a remote desktop protocol.
`capture-touchscreen-input`	Captures data from a system's touchscreen.
`check-for-payload`	Queries a command and control server to check whether a new payload is available for download.
`check-language`	Checks the language of the host system on which it executes.

`click-fraud`	Simulates legitimate user clicks on website advertisements for the purpose of revenue generation.
`compare-host-fingerprints`	Compares a previously computed host fingerprint to one computed for the current system on which the malware instance is executing, to determine if the malware instance is still executing on the same system.
`compromise-remote-machine`	Gains control of a remote machine through compromise, e.g., by exploiting a particular vulnerability.
`control-local-machine-via-remote-command`	Controls the machine on which the malware instance is executing, through one or more remotely sent commands.
`control-malware-via-remote-command`	Executes commands issued to the malware instance from a remote source such as a command and control server, for the purpose of controlling its behavior.
`crack-passwords`	Consumes system resources for the purpose of password cracking.
`defeat-call-graph-generation`	Defeats accurate call graph generation during disassembly of the malware instance.
`defeat-emulator`	Defeats or prevents the execution of the malware instance in an emulator.
`defeat-flow-oriented-disassembler`	Defeats disassembly of the malware instance in a flow-oriented (recursive traversal) disassembler.
`defeat-linear-disassembler`	Prevents the disassembly of the malware instance in a linear disassembler.
`degrade-security-program`	Degrades one or more security programs running on a system, either by stopping them from executing or by making changes to their code or configuration parameters.
`denial-of-service`	Causes the local machine on which the malware instance is executing and/or a remote network resource to be unavailable.
`destroy-hardware`	Physically destroys a piece of hardware, e.g., by causing it to overheat.
`detect-debugging`	Detects whether the malware instance is being executed inside of a debugger.

`detect-emulator`	Detects whether the malware instance is being executed inside of an emulator.
`detect-installed-analysis-tools`	Indicates that the malware instance attempts to detect whether certain analysis tools are present on the system on which it is executing.
`detect-installed-av-tools`	Indicates that the malware instance attempts to detect whether certain anti-virus tools are present on the system on which it is executing.
`detect-sandbox-environment`	Detects whether the malware instance is being executed in a sandbox environment.
`detect-vm-environment`	Detects whether the malware instance is being executed in a virtual machine (VM).
`determine-host-ip-address`	Determines the IP address of the host system on which the malware instance is executing.
`disable-access-rights-checking`	Bypasses, disables, or modifies access tokens or access control lists, thereby enabling the malware instance to read, write, or execute a file with one or more of these controls set.
`disable-firewall`	Evades or disables the host-based firewall running on the system on which the malware instance is executing.
`disable-kernel-patch-protection`	Bypasses or disables kernel patch protection mechanisms such as Windows' PatchGuard, enabling the malware instance to operate at the same level as the operating system kernel and kernel mode drivers (KMD).
`disable-os-security-alerts`	Disables operating system (OS) security alert messages that could lead to identification and/or notification of the presence of the malware instance.
`disable-privilege-limiting`	Bypasses or disables mechanisms that limit the privileges that can be granted to a user or entity.
`disable-service-pack-patch-installation`	Disables the system's ability to install service packs and/or patches.
`disable-system-file-overwrite-protection`	Disables system file overwrite protection mechanisms such as Windows file protection, thereby enabling system files to be modified or replaced.

`disable-update-services-daemons`	Disables system update services or daemons that may be already be running on the system on which the malware instance is executing.
`disable-user-account-control`	Bypasses or disables Windows' user account control (UAC), enabling the malware instance and/or its component to execute with elevated privileges.
`drop-retrieve-debug-log-file`	Generates and retrieves a log file of errors relating to the execution of the malware instance.
`elevate-privilege`	Elevates the privilege level under which the malware instance is executing.
`encrypt-data`	Encrypts data that will be exfiltrated.
`encrypt-files`	Encrypts one or more files on the system on which the malware instance is executing, to make them unavailable for use by the users of the system.
`encrypt-self`	Encrypts the executing code (in memory) that belongs to the malware instance.
`erase-data`	Destroys data stored on a disk or in memory by erasure.
`evade-static-heuristic`	Evades a static anti-virus heuristic. For example, an heuristic engine can try to figure out if a file are using a dual extension (e.g: invoice.doc.exe) and determine the file as being malicious.
`execute-before-external-to-kernel-hypervisor`	Executes some or all of the malware instance's code before or external to the system's kernel or hypervisor (e.g., through the BIOS).
`execute-non-main-cpu-code`	Executes some or all of the code of the malware instance on a secondary, non-CPU processor (e.g., a GPU).
`execute-stealthy-code`	Executes code in a hidden manner (e.g., by injecting it into a benign process).
`exfiltrate-data-via-covert channel`	Exfiltrates data using a covert channel, such as a DNS tunnel or NTP.
`exfiltrate-data-via--dumpster-dive`	Exfiltrates data via dumpster dive - i.e, encoded data printed by malware is viewed as garbage and thrown away to then be physically picked up.

`exfiltrate-data-via-fax`	Exfiltrates data using a fax system.
`exfiltrate-data-via-network`	Exfiltrates data through the computer network connected to the system on which the malware instance is executing.
`exfiltrate-data-via-physical-media`	Exfiltrates data by writing it to physical media (e.g., to a USB flash drive).
`exfiltrate-data-via-voip-phone`	Exfiltrates data (encoded as audio) using a phone system, such as through voice over IP (VoIP).
`feed-misinformation-during-physical-memory-acquisition`	Reports inaccurate data when the contents of the physical memory of the system on which the malware instance is executing is retrieved.
`file-system-instantiation`	Indicates that the malware instance instantiates itself on the file system of the machine that it is infecting, in one or more locations.
`fingerprint-host`	Creates a unique fingerprint for the system on which the malware instance is executing, e.g., based on the applications that are installed on the system.
`generate-c2-domain-names`	Generates the domain name of the command and control server to which the malware connects to.
`hide-arbitrary-virtual-memory`	Hides arbitrary segments of virtual memory belonging to the malware instance in order to prevent their retrieval.
`hide-data-in-other-formats`	Hides data that will be exfiltrated in other formats (e.g., image files).
`hide-file-system-artifacts`	Hides one or more file system artifacts (e.g., files and/or directories) associated with the malware instance.
`hide-kernel-modules`	Hides the usage of any kernel modules by the malware instance.
`hide-network-traffic`	Hides network traffic associated with the malware instance.
`hide-open-network-ports`	Hides one or more open network ports associated with the malware instance.
`hide-processes`	Hides one or more of the processes in which the malware instance is executing.

`hide-registry-artifacts`	Hides one or more Windows registry artifacts (e.g., keys and/or values) associated with the malware instance.
`hide-services`	Hides any system services that the malware instance creates or injects itself into.
`hide-threads`	Hides one or more threads that belong to the malware instance.
`hide-userspace-libraries`	Hides the usage of userspace libraries by the malware instance.
`identify-file`	Identifies one or more files on a local, removable, and/or network drive for infection.
`identify-os`	Identifies the operating system under which the malware instance is executing.
`identify-target-machines`	Identifies one or more machines to be targeted for infection via some remote means (e.g., via email or the network).
`impersonate-user`	Impersonates another user in order to operate within a different security context.
`install-backdoor`	Installs a backdoor on the system on which the malware instance is executing, capable of providing covert remote access to the system.
`install-legitimate-software`	Installs legitimate (i.e. non-malware) software on the same system on which the malware instance is executing.
`install-secondary-malware`	Installs another, different malware instance on the system on which the malware instance is executing.
`install-secondary-module`	Installs a secondary module (typically related to the malware instance itself) on the same system on which the malware instance is executing.
`intercept-manipulate-network-traffic`	Intercepts and/or manipulates network traffic going to or originating from the system on which the malware instance is executing.

`inventory-security-products`	Creates an inventory of the security products installed or running on a system.
`inventory-system-applications`	Inventories the applications installed on the system on which the malware instance is executing.
`inventory-victims`	Keeps an inventory of the victims that are remotely infected by the malware instance.
`limit-application-type-version`	Limits the type or version of an application that runs on a system in order to ensure that the malware instance is able to continue executing.
`log-activity`	Logs the activity of the malware instance.
`manipulate-file-system-data`	Manipulates data stored on the file system of the system on which the malware instance is executing in order to compromise its integrity.
`map-local-network`	Maps the layout of the local network environment in which the malware instance is executing.
`mine-for-cryptocurrency`	Consumes system resources for cryptocurrency (e.g., Bitcoin, Litecoin, etc.) mining.
`modify-file`	Modifies a file in some other manner than writing code to it, such as packing it (in terms of binary executable packing).
`modify-security-software-configuration`	Modifies the configuration of one or more instances of security software (e.g., anti-virus) running on a system in order to negatively impact their usefulness and ability to detect the malware instance.
`move-data-to-staging-server`	Moves data to be exfiltrated to a particular server, to prepare it for exfiltration.
`obfuscate-artifact-properties`	Hides the properties of one or more artifacts associated with the malware instance (e.g., by altering file system timestamps).
`overload-sandbox`	Overloads a sandbox (e.g., by generating a flood of meaningless behavioral data)
`package-data`	Packages data for exfiltration, e.g., by adding it to an archive file.

`persist-after-hardware-changes`	Continues the execution of the malware instance after hardware changes to the system on which it is executing have been made, such as replacement of the hard drive on which the operating system was residing.
`persist-after-os-changes`	Continues the execution of the malware instance after the operating system under which it is executing is modified, such as being installed or reinstalled.
`persist-after-system-reboot`	Continues the execution of the malware instance after a system reboot.
`prevent-api-unhooking`	Prevents the API hooks installed by the malware instance from being removed.
`prevent-concurrent-execution`	Checks to see if it is already running on a system, in order to prevent multiple instances of the malware running concurrently.
`prevent-debugging`	Prevents the execution of the malware instance in a debugger.
`prevent-file-access`	Prevents access to the file system, including to specific files and/or directories associated with the malware instance.
`prevent-file-deletion`	Prevents files and/or directories associated with the malware instance from being deleted from a system.
`prevent-memory-access`	Prevents access to system memory where the malware instance may be storing code or data.
`prevent-native-api-hooking`	Prevents other software from hooking native system APIs.
`prevent-physical-memory-acquisition`	Prevents the contents of the physical memory of the system on which the malware instance is executing from being retrieved.
`prevent-registry-access`	Prevents access to the Windows registry, including to the entire registry and/or to particular registry keys/values.
`prevent-registry-deletion`	Prevent Windows registry keys and/or values associated with the malware instance from being deleted from a system.
`prevent-security-software--from-executing`	Prevents one or more instances of security software from executing on a system.

`re-instantiate-self`	Re-establishes the malware instance on the system after it is initially detected and partially removed.
`remove-self`	Removes the malware instance from the system on which it is executing.
`remove-sms-warning-messages`	Captures the message body of incoming SMS messages and aborts displaying messages that meets a certain criteria.
`remove-system-artifacts`	Removes artifacts associated with the malware instance (e.g., files, directories, Windows registry keys, etc.) from the system on which it is executing.
`request-email-address-list`	Requests the current list of email addresses, for sending email spam messages to, from the command and control server.
`request-email-template`	Requests the current template, for use in generating email spam messages, from the command and control server.
`search-for-remote-machines`	Searches for one or more remote machines to target.
`send-beacon`	Sends 'beacon' data to a command and control server, indicating that the malware instance is still active on the host system and able to communicate.
`send-email-message`	Sends an email message from the system on which the malware instance is executing to one or more recipients, most commonly for the purpose of spamming.
`send-system-information`	Sends data regarding the system on which it is executing to a command and control server.
`social-engineering-based-remote-infection`	Infects remote machines via some method that involves social engineering (e.g., sending an email with a malicious attachment).
`steal-browser-cache`	Steals a user's browser cache.
`steal-browser-cookies`	Steals one or more browser cookies stored on the system on which the malware instance is executing.
`steal-browser-history`	Steals a user's browser history.
`steal-contact-list-data`	Steals a user's contact list.
`steal-cryptocurrency-data`	Steals cryptocurrency data that may be stored on a system (e.g., Bitcoin wallets).

`steal-database-content`	Steals content from a database that the malware instance may be able to access.
`steal-dialed-phone-numbers`	Steals the list of phone numbers that a user has dialed (i.e. on a mobile device).
`steal-digital-certificates`	Steals one or more digital private keys that may be present on the system on which the malware instance is executing, to then use to hijack the corresponding digital certificates, e.g., those used in public-key infrastructure (PKI).
`steal-documents`	Steals document files (e.g., PDF) stored on a system.
`steal-email-data`	Steals a user's email data.
`steal-images`	Seals image files that may be stored on a system.
`steal-password-hashes`	Steals password hashes.
`steal-pki-key`	Steals one or more public key infrastructure (PKI) keys.
`steal-referrer-urls`	Steals HTTP referrer information (URL of the webpage that linked to the resource being requested).
`steal-serial-numbers`	Steals serial numbers stored on a system.
`steal-sms-database`	Steals a user's short message service (SMS) (text messaging) database (i.e. on a mobile device).
`steal-web-network-credential`	Steals usernames, passwords, or other forms of web (e.g., for logging into a website) and/or network credentials.
`stop-execution-of-security-software`	Stops the execution of one or more instances of security software that may already be executing on a system.
`suicide-exit`	Terminates the execution of the malware instance based on some trigger condition or value.
`test-for-firewall`	Tests whether the network environment in which the malware instance is executing contains a hardware or software firewall.
`test-for-internet-connectivity`	Tests whether the network environment in which the malware instance is executing is connected to the internet.

`test-for-network-drives`	Tests for network drives that may be present in the network environment in which the malware instance is executing.
`test-for-proxy`	Tests whether the network environment in which the malware instance is executing contains a hardware or software proxy.
`test-smtp-connection`	Tests whether an outgoing SMTP connection can be made from the system on which the malware instance is executing to some SMTP server, by sending a test SMTP transaction.
`update-configuration`	Updates the configuration of the malware instance using data received from a command and control server.
`validate-data`	Validates the integrity of data received from a command and control server.
`write-code-into-file`	Writes code into one or more files.

5. Capabilities

Type Name: `capability-ov`

The Malware Capability open vocabulary is used in the following object/*property*:

- Capability (`capability`)
  - *name* (`open-vocab`)

Vocabulary Value	Description
`anti-behavioral-analysis`	Indicates that the malware instance or family is able to prevent behavioral analysis or make it more difficult.
`anti-code-analysis`	Indicates that the malware instance or family is able to prevent code analysis or make it more difficult.
`anti-detection`	Indicates that the malware instance or family is able to prevent itself and its components from being detected on a system.
`anti-removal`	Indicates that the malware instance or family is able to prevent itself and its components from being removed from a system.
`availability-violation`	Indicates that the malware instance or family is able to compromise the availability of a system or some aspect of the system.

`collection`	Indicates that the malware instance or family is able to capture information from a system related to user or system activity (e.g., from a system's peripheral devices).
`command-and-control`	Indicates that the malware instance or family is able to receive and/or execute remotely submitted commands.
`data-theft`	Indicates that the malware instance or family is able to steal data from the system on which it executes. This includes data stored in some form, e.g. in a file, as well as data that may be entered into some application such as a web-browser.
`destruction`	Indicates that the malware instance or family is able to destroy some aspect of a system.
`discovery`	Indicates that the malware instance or family is able to probe its host system or network environment; most often this is done to support other Capabilities and their Objectives.
`exfiltration`	Indicates that the malware instance or family is able to exfiltrate stolen data or perform tasks related to the exfiltration of stolen data.
`fraud`	Indicates that the malware instance or family is able to defraud a user or a system.
`infection-propagation`	Indicates that the malware instance or family is able to propagate through the infection of a machine or is able to infect a file after executing on a system. The malware instance may infect actively (e.g., gain access to a machine directly) or passively (e.g., send malicious email). This Capability does not encompass any aspects of the initial infection that is done independently of the malware instance itself.
`integrity-violation`	Indicates that the malware instance or family is able to compromise the integrity of a system.
`machine-access-control`	Indicates that the malware instance or family is able to access or control one or more remote machines and/or the machine on which it is executing.
`persistence`	Indicates that the malware instance or family is able to persist and remain on a system regardless of system events.

`privilege-escalation`	Indicates that the malware instance or family is able to elevate the privileges under which it executes.
`secondary-operation`	Indicates that the malware instance or family is able to achieve secondary objectives in conjunction with or after achieving its primary objectives.
`security-degradation`	Indicates that the malware instance or family is able to bypass or disable security features and/or controls.

6. Common Attributes

Type Name: `common-attribute-ov`

The Common Attribute open vocabulary is used in the following objects/*properties*:

- Capability (`capability`)
  - *attributes* (`dictionary`)
- Behavior (`behavior`)
  - *attributes* (`dictionary`)

This vocabulary is a non-exhaustive enumeration of common attributes associated with Capabilities or Behaviors of a Malware Instance or Malware Family.

Vocabulary Value	Description
`applicable-platform`	Captures the name of a targeted platform.
`archive-type`	Captures the name of a file archive format used.
`autonomy`	Captures the level of autonomy used.
`backdoor-type`	Captures the type of backdoor used.
`cryptocurrency-type`	Captures a cryptocurrency targeted.
`encryption-algorithm`	Captures the encryption algorithm used.
`erasure-scope`	Captures the scope of the erasure performed.
`file-infection-type`	Captures the method that an executable infector uses to infect a file.
`file-modification-type`	Captures how a malware file modifies itself to avoid detection.
`file-type`	Captures the type of file format used for storing data to be exfiltrated.

`frequency`	Captures the frequency with which a C2 Server sends and receives data. It is recommended that the description follow the format of "every x (units)", e.g., "every 5 minutes."
`infection-targeting`	Captures the type of targeting employed, e.g., whether the targeted machines are randomly selected, or chosen from some particular set.
`network-protocol`	Captures the name of the network protocol used in command and control communications. The protocol name **MUST** come from the service names defined in the Service Name column of the IANA Service Name and Port Number Registry. In cases where an there is variance in the name of a network protocol not included in the IANA Registry, content producers should exercise their best judgement, and it is recommended that lowercase names be used for consistency with the IANA registry.
`port-number`	Captures the port number used in command and control communications.
`persistence-scope`	Captures the scope of persistence employed.
`propagation-scope`	Captures the scope of the infection or propagation performed, i.e., whether it infects just the local machine or actively propagates to other machines as well.
`targeted-application`	Captures the names of any targeted applications.
`targeted-file-architecture type`	Captures the types of file architectures targeted.
`targeted-file-type`	Captures the types of files targeted.
`targeted-program`	Captures the names of any targeted programs.
`targeted-sandbox`	Captures the names of any sandboxes targeted.
`targeted-vm`	Captures the names of any virtual machines (VM) targeted.
`targeted-website`	Captures the domain names of any targeted websites.
`technique`	Captures the name of the technique used.
`trigger-type`	Captures the trigger used to wake or terminate the malware instance.
`user-privilege-escalation type`	Captures the type of user privilege escalation employed.

`vulnerability-id-cve`	Captures the Common Vulnerabilities and Exposures (CVE) vulnerability identifier being referenced.
`vulnerability-id-osvdb`	Captures the vulnerability identifier of the Open Source Vulnerability Database (OSVDB) entry being referenced.

7. Delivery Vectors

Type Name: `delivery-vector-ov`

The Delivery Vector open vocabulary is used in the following objects/*properties*:

- Malware Instance (`malware-instance`)
  - field_data (`field-data`)
    - *delivery_vectors* (`list` of type `open-vocab`)
- Malware Family (`malware-family`)
  - field_data (`field-data`)
    - *delivery_vectors* (`list` of type `open-vocab`)

This vocabulary is a non-exhaustive enumeration of vectors used to deliver malware

Vocabulary Value	Description
`active-attacker`	The malware instance or family was delivered via an active attacker.
`auto-executing-media`	The malware instance or family was delivered via media that automatically executes.
`downloader`	The malware instance or family was delivered via downloader.
`dropper`	The malware instance or family was delivered via dropper.
`email-attachment`	The malware instance or family was delivered via email attachment.
`exploit-kit-landing-page`	The malware instance or family was delivered via an exploit kit landing page.
`fake-website`	The malware instance or family was delivered via a fake website.
`janitor-attack`	The malware instance or family was delivered via a janitor attack.
`malicious-iframes`	The malware instance or family was delivered via malicious iframes.
`malvertising`	The malware instance or family was delivered via malvertising.

`media-baiting`	The malware instance or family was delivered via media baiting.
`pharming`	The malware instance or family was delivered via pharming.
`phishing`	The malware instance or family was delivered via phishing.
`trojanized-link`	The malware instance or family was delivered via a trojanized link.
`trojanized-software`	The malware instance or family was delivered via trojanized software.
`usb-cable-syncing`	The malware instance or family was delivered via usb cable syncing.
`watering-hole`	The malware instance or family was delivered via a watering hole.

8. Entity Associations

Type Name: `entity-association-ov`

The Entity Association open vocabulary is used in the following object/*property*:

- Collection (`collection`)
    - *association_type* (`open-vocab`)

This vocabulary is an non-exhaustive enumeration of entity association types relevant to MAEC entities and STIX Cyber Observables.

Vocabulary Value	Description
`file-system-entities`	The collection contains Cyber Observable Objects that correspond to file system entities.
`network-entities`	The collection contains Cyber Observable Objects that correspond to network entities.
`process-entities`	The collection contains Cyber Observable Objects that correspond to process entities.
`memory-entities`	The collection contains Cyber Observable Objects that correspond to memory entities.
`ipc-entities`	The collection contains Cyber Observable Objects that correspond to inter-process communication (IPC) entities.
`device-entities`	The collection contains Cyber Observable Objects that correspond to device entities.

`registry-entities`	The collection contains Cyber Observable Objects that correspond to registry entities.
`service-entities`	The collection contains Cyber Observable Objects that correspond to service entities.
`potential-indicators`	The collection contains entities that serve as potential indicators.
`same-malware-family`	The collection contains that Malware Instances are from the same malware family.
`clustered-together`	The collection contains Malware Instances that have been clustered together by some method such as a scoring algorithm.
`observed-together`	The collection contains Malware Instances that have been observed together.
`same-malware-toolkit`	The collection contains Malware Instances that are derived from the same malware toolkit.

9. Malware Actions

Type Name: `malware-action-ov`

The Malware Action open vocabulary is used in the following object/*property*:

- Malware Action (`malware-action`)
  - *name* (`open-vocab`)

This vocabulary is a non-exhaustive enumeration of Actions that may be performed by a Malware Instance.

Vocabulary Value	Description
`accept-socket-connection`	The action of accepting a socket connection.
`add-connection-to-network-share`	The action of adding a connection to a network share.
`add-network-share`	The action of adding a new network share on a server.
`add-scheduled-task`	The action of adding a scheduled task to a system.
`add-system-call-hook`	The action of adding a new system call hook.
`add-user-to-group`	The action of adding an existing user to a group.
`add-user`	The action of adding a new user.

`add-windows-hook`	The action of adding a new Windows application-defined hook procedure.
`allocate-process-virtual-memory`	The action of allocating a virtual memory region in an existing process.
`bind-address-to-socket`	The action of binding a socket address to a socket.
`call-library-function`	The action of calling a function exported by a library.
`change-password`	The action of changing a user's password.
`check-for-kernel-debugger`	The action of checking for the presence of a kernel debugger.
`check-for-remote-debugger`	The action of checking for the presence of a remote debugger.
`close-file`	The action of closing an existing file previously opened for reading or writing.
`close-port`	The action of closing a network port.
`close-registry-key`	The action of closing a handle to an existing registry key.
`close-socket`	The action of closing a socket.
`connect-to-ftp-server`	The action of connecting to an ftp server.
`connect-to-ip-address`	The action of connecting to an IP address.
`connect-to-irc-server`	The action of connecting to an IRC server.
`connect-to-named-pipe`	The action of connecting to a named pipe.
`connect-to-network-share`	The action of connecting to a network share.
`connect-to-socket-address`	The action of connecting to a socket address.
`connect-to-socket`	The action of connecting to a socket, which consists of an IP address and port number.
`connect-to-url`	The action of connecting to a URL.
`copy-file`	The action of copying a file from one location to another.
`create-critical-section`	The action of creating a new critical section.
`create-dialog-box`	The action of creating a new dialog box.

`create-directory`	The action of creating a new directory.
`create-event`	The action of creating a new event.
`create-file-alternate-data-stream`	The action of creating a new file alternate data stream in a file.
`create-file-mapping`	The action of creating a new file mapping object.
`create-file-symbolic-link`	The action of creating a new symbolic link to a file.
`create-file`	The action of creating a new file.
`create-mailslot`	The action of creating a new mailslot.
`create-mutex`	The action of creating a new mutex.
`create-named-pipe`	The action of creating a new named pipe.
`create-process-as-user`	The action of creating a new process as a particular user.
`create-process`	The action of creating a new process.
`create-registry-key-value`	The action of creating a new registry key value.
`create-registry-key`	The action of creating a new registry key.
`create-remote-thread-in-process`	The action of creating a new thread that runs in the virtual address space of another process.
`create-semaphore`	The action of creating a new named semaphore.
`create-service`	The action of creating a new service.
`create-socket`	The action of creating a new socket.
`create-thread`	The action of creating a new thread.
`create-window`	The action of creating a new window.
`delete-critical-section`	The action of deleting a critical section object.
`delete-directory`	The action of deleting a directory on a filesystem.
`delete-event`	The action of deleting a named event object.
`delete-file`	The action of deleting a file.
`delete-mutex`	The action of deleting a named mutex.
`delete-named-pipe`	The action of deleting a named pipe.

`delete-network-share`	The action of deleting a network share on a server.
`delete-registry-key-value`	The action of deleting a named value under an existing registry key.
`delete-registry-key`	The action of deleting a registry key.
`delete-semaphore`	The action of deleting a named semaphore.
`delete-service`	The action of deleting a service.
`delete-user`	The action of deleting a user.
`disconnect-from-ftp-server`	The action of disconnecting from a FTP server.
`disconnect-from-ip`	The action of disconnecting from a previously established connection to an IP address.
`disconnect-from-irc-server`	The action of disconnecting from an IRC server.
`disconnect-from-named-pipe`	The action of disconnecting from a named pipe.
`disconnect-from-network-share`	The action of disconnecting from a network share.
`disconnect-from-socket`	The action of disconnecting from a socket.
`download-file`	The action of downloading a file from a remote location.
`enumerate-libraries`	The action of enumerating the libraries used by a process.
`enumerate-network-shares`	The action of enumerating the available shared resources on a server.
`enumerate-processes`	The action of enumerating all of the running processes on a system.
`enumerate-registry-key-subkeys`	The action of enumerating the registry key subkeys under a registry key.
`enumerate-registry-key-values`	The action of enumerating the named values under a registry key.
`enumerate-services`	The action of enumerating a specific set of services on a system.
`enumerate-system-handles`	The action of enumerating all open handles on a system.
`enumerate-threads`	The action of enumerating all threads in the calling process.

`enumerate-users`	The action of enumerating all users.
`enumerate-windows`	The action of enumerating all open windows.
`execute-file`	The action of executing a file.
`find-file`	The action of searching for a file.
`find-window`	The action of searching for a window.
`flush-process-instruction-cache`	The action of flushing the instruction cache of a process.
`free-library`	The action of freeing a library previously loaded into the address space of the calling process.
`free-process-virtual-memory`	The action of freeing virtual memory regions from a process.
`get-disk-attributes`	The action of querying the attributes of a disk, such as the amount of available free space.
`get-disk-type`	The action of getting the disk type.
`get-elapsed-system-up-time`	The action of getting the elapsed up-time for a system.
`get-file-or-directory-attributes`	The action of getting the attributes of a file or directory.
`get-function-address`	The action of getting the address of an exported function or variable from a library.
`get-host-by-address`	The action of getting information on a host from a local or remote host database by its IP address.
`get-host-by-name`	The action of getting information on a host from a local or remote host database by its name.
`get-netbios-name`	The action of getting the NetBIOS name of a system.
`get-process-current-directory`	The action of getting the current directory of a process.
`get-process-environment-variable`	The action of setting an environment variable used by a process.
`get-process-startupinfo`	The action of getting the STARTUPINFO struct associated with a process.
`get-registry-key-attributes`	The action of getting the attributes of a registry key.
`get-system-global-flags`	The action of getting the enabled global flags on a system.

`get-system-host-name`	The action of getting the hostname of a system.
`get-system-local-time`	The action of getting the local time of a system.
`get-system-time`	The action of getting the system time of a system, represented in Coordinated Universal Time (UTC).
`get-thread-context`	The action of getting the context structure (containing process-specific register data) of a thread.
`get-thread-username`	The action of getting the name or ID of the user associated with a thread.
`get-user-attributes`	The action of getting the attributes of a user.
`get-username`	The action of getting the username of the currently logged in user of a system.
`get-windows-directory`	The action of getting the Windows installation directory on a system.
`get-windows-system-directory`	The action of getting the Windows \System or \System32 directory on a system.
`get-temporary-files-directory`	The action of getting the temporary file directory on a system.
`impersonate-process`	The action of a thread in the calling process impersonating the security context of another process.
`invoke-user-privilege`	The action of invoking a privilege given to an existing user.
`join-irc-channel`	The action of joining a channel on an IRC server.
`kill-process`	The action of killing a process.
`kill-thread`	The action of killing a thread in the virtual address space of the calling process.
`kill-window`	The action of killing a window.
`leave-irc-channel`	The action of leaving a channel on an IRC server.
`enumerate-disks`	The action of listing all disks available on a system.
`listen-on-port`	The action of listening on a specific port.
`listen-on-socket`	The action of listening on a socket.

`load-and-call-driver`	The action of loading a driver into a system and then calling the loaded driver.
`load-driver`	The action of loading a driver into a system.
`load-library`	The action of loading a library into the address space of the calling process.
`lock-file`	The action of locking a file.
`logon-as-user`	The action of logging on as a specific user.
`map-file-into-process`	The action of mapping a file into the address space of the calling process.
`map-library-into-process`	The action of mapping a library into the address space of the calling process.
`modify-process-virtual-memory-protection`	The action of modifying the protection on a memory region in the virtual address space of a process.
`modify-registry-key-value`	The action of modifying a named value of a registry key.
`modify-registry-key`	The action of modifying a registry key.
`modify-service-configuration`	The action of modifying the configuration parameters of a service.
`monitor-directory`	The action of monitoring a directory on the filesystem for changes.
`monitor-disk`	The action of monitoring a disk for changes.
`monitor-registry-key`	The action of monitoring a registry key for changes.
`mount-disk`	The action of mounting a file system to a mounting point.
`move-file`	The action of moving a file from one location to another.
`open-critical-section`	The action of opening a critical section object.
`open-event`	The action of opening a named event object.
`open-file-mapping`	The action of opening a file mapping object.
`open-directory`	The action of opening a directory.
`open-file`	The action of opening a file for reading or writing.
`open-mutex`	The action of opening a named mutex.

`open-port`	The action of opening a network port.
`open-process`	The action of opening a process.
`open-registry-key`	The action of opening a registry key.
`open-semaphore`	The action of opening a named semaphore.
`open-service`	The action of opening a service.
`queue-apc-in-thread`	The action of queuing a new Asynchronized Procedure Call (APC) in the context of a thread.
`read-from-file`	The action of reading from a file.
`read-from-mailslot`	The action of reading some data from a named mailslot.
`read-from-named-pipe`	The action of reading data from a named pipe.
`read-from-process-memory`	The action of reading from a memory region of a process.
`read-registry-key-value`	The action of reading a named value of a registry key.
`receive-data-on-socket`	The action of receiving data on a socket.
`receive-http-response`	The action of receiving an HTTP server response for a prior HTTP request.
`receive-irc-private-message`	The action of receiving a private message from another user on an IRC server.
`receive-network-packet`	The action of receiving a packet on a network.
`release-critical-section`	The action of releasing a critical section object.
`release-mutex`	The action of releasing ownership of a named mutex.
`release-semaphore`	The action of releasing ownership of a named semaphore.
`remove-user-from-group`	The action of removing a user from a group.
`rename-file`	The action of renaming a file on a file system.
`reset-event`	The action of resetting a named event object to the non-signaled state.
`revert-thread-to-self`	The action of reverting a thread to its own security context.

`send-control-code-to-file`	The action of sending a control code to a file.
`send-control-code-to-service`	The action of sending a control code to a service.
`send-data-on-socket`	The action of sending data on a connected socket.
`send-data-to-address-on-socket`	The action of sending data to a specified IP address on an unconnected socket.
`send-dns-query`	The action of sending a DNS query.
`send-email-message`	The action of sending an email message.
`send-ftp-command`	The action of of sending a command on an FTP server connection.
`send-http-connect-request`	The action of sending an HTTP CONNECT client request to a server.
`send-http-delete-request`	The action of sending an HTTP DELETE client request.
`send-http-get-request`	The action of sending an HTTP GET client request.
`send-http-head-request`	The action of sending an HTTP HEAD client request.
`send-http-options-request`	The action of sending an HTTP OPTIONS client request.
`send-http-patch-request`	The action of sending an HTTP PATCH client request.
`send-http-post-request`	The action of sending an HTTP HEAD client request.
`send-http-put-request`	The action of sending an HTTP PUT client request.
`send-http-trace-request`	The action of sending an HTTP TRACE client request.
`send-icmp-request`	The action of sending an ICMP request.
`send-irc-private-message`	The action of sending a private message to another user on an IRC server.
`send-network-packet`	The action of sending a packet on a network.
`send-reverse-dns-lookup`	The action of sending a reverse DNS lookup.
`set-file-or-directory-attributes`	The action of setting some attributes on a file or directory.
`set-irc-nickname`	The action of setting an IRC nickname on an IRC server.
`set-netbios-name`	The action of setting the NetBIOS name of a system.

`set-process-current-directory`	The action of setting the current directory of a process.
`set-process-environment-variable`	The action of setting an environment variable used by a process.
`set-system-global-flags`	The action of setting a system's global flags.
`set-system-host-name`	The action of setting a system's hostname.
`set-system-local-time`	The action of setting a system's local time.
`set-system-time`	The action of setting a system's time, represented in UTC.
`set-thread-context`	The action of setting the context structure (containing process-specific register data) for a thread.
`show-window`	The action of showing a window.
`shutdown-system`	The action of shutting down a system.
`sleep-process`	The action of sleeping a process for some period of time.
`sleep-system`	The action of sleeping a system for some period of time.
`start-service`	The action of starting a service.
`stop-service`	The action of stopping a service.
`unload-driver`	The action of unloading a driver from a system.
`unlock-file`	The action of unlocking a file.
`unmap-file-from-process`	The action of unmapping a file from the address space of the calling process.
`unmount-disk`	The action of unmounting a file system from a mounting point.
`upload-file`	The action of uploading a file to a remote location.
`write-to-file`	The action of writing data to a file.
`write-to-mailslot`	The action of writing data to a named mailslot.
`write-to-named-pipe`	The action of writing data to a named pipe.
`write-to-process-memory`	The action of writing to a memory region of an existing process.

10. Malware Configuration Parameters

Type Name: `malware-configuration-parameter-ov`

The Malware Configuration Parameters open vocabulary is used in the following object/*property*:

- Malware Instance (`malware-instance`)
  - static_features (`static-features`)
    - *configuration_parameters* (`dictionary`)

This vocabulary is a non-exhaustive enumeration of malware configuration parameter names.

Vocabulary Value	Description
`filename`	Captures the name of a file (e.g., a binary downloaded, embedded binary).
`group-id`	Captures an identifier of a collection of malware instances.
`id`	Captures an identifier of a malware instance.
`installation-path`	Captures a location on disk where the malware instance is installed, copied, or moved.
`magic-number`	Captures a file signature used to identify or validate content of the malware instance.
`mutex`	Captures a unique mutex value associated with the malware instance.
`c2-ip-address`	Captures an IP address used by the malware instance for command and control.
`c2-domain`	Captures a domain name used by the malware instance for command and control.
`c2-url`	Captures a URL used by the malware instance for command and control.
`directory`	Captures the name of a directory used by the malware instance.
`filepath`	Captures a file path (directory + file name) used by the malware instance.
`injection-process`	Captures a process into which malware instance is injected. Usually this is a process name but it may take other forms such as a filename of the executable.
`interval`	Captures the time malware instance waits between beacons or other activity, in seconds.

`key`	Captures an encryption, encoding, or obfuscation key used by the malware instance. By convention, when these represent binary data, they should be bare hex encoded with no other markup. Base64 or similar custom dictionaries are stored as is.
`password`	Captures a password used by the malware instance.
`useragent`	Captures a software identifier used by the malware instance.
`version`	Captures the version of the malware instance. To the degree possible this should be based directly on artifacts from the malware.

11. Malware Labels

Type Name: `malware-label-ov`

The Malware Label open vocabulary is used in the following objects/properties:

- Malware Instance (`malware-instance`)
  - *labels (*`list` of type `open-vocab`)
- Malware Family (`malware-instance`)
  - *labels (*`list` of type `open-vocab`)

This vocabulary is a non-exhaustive enumeration of common malware labels.

Vocabulary Value	Description
`adware`	Any software that is funded by advertising. Adware may also gather sensitive user information from a system.
`appender`	File-infecting malware that places its code at the end of the files it infects, adjusting the file's entry point to cause its code to be executed before that in the original file.
`backdoor`	Malware which, once running on a system, opens a communication vector to the outside so the computer can be accessed remotely by an attacker.
`boot-sector-virus`	Malware that infects the master boot record of a storage device.
`bot`	Malware that resides on an infected system, communicating with and forming part of a botnet. The bot may be implanted by a worm or trojan, which opens a backdoor. The bot then monitors the backdoor for further instructions.
`cavity-filler`	A type of file-infecting virus that seeks unused space within the files it infects, inserting its code into these gaps to avoid changing the size

	of the file and thus not alerting integrity-checking software to its presence.
`clicker`	A trojan that makes a system visit a specific web page, often very frequently and usually with the aim of increasing the traffic recorded by the site and thus increasing revenue from advertising. Clickers may also be used to carry out DDoS attacks.
`companion-virus`	A virus that takes the place of a particular file on a system instead of injecting code into it.
`data-diddler`	A type of malware that makes small, random changes to data, such as data in a spreadsheet, to render the data contained in a document inaccurate and in some cases worthless.
`ddos`	A tool used to perform a distributed denial of service attack.
`downloader`	Malware programmed to download and execute other files, usually more complex malware.
`dropper`	A type of Trojan that deposits an enclosed payload onto a destination host computer by loading itself into memory, extracting the malicious payload, and then writing it to the file system.
`exploit-kit`	A software toolkit to target common vulnerabilities.
`file-infector-virus`	A virus that infects a system by inserting itself somewhere in existing files; this is the "classic" form of virus.
`file-less`	Malware that is file-less, i.e., executes through some other mechanism such as Powershell.
`fork-bomb`	A simple form of malware, a type of rabbit which launches more copies of itself. Once a fork bomb is executed, it will attempt to run several identical processes, which will do the same, the number growing exponentially until the system resources are overwhelmed by the number of identical processes running, which may in some cases bring the system down and cause a denial of service.
`greyware`	Software that, while not definitely malicious, has a suspicious or potentially unwanted aspect.
`implant`	Code inserted into an existing program using a code patcher or other tool.
`keylogger`	A type of program implanted on a system to monitor the keys pressed and thus record any sensitive data, such as passwords, entered by the user.

`kleptographic-worm`	A worm that encrypts information assets on compromised systems so they can only be decrypted by the worm's author, also known as information-stealing worm.
`macro-virus`	A virus that uses a macro language, for example in Microsoft Office documents.
`malware-as-a-service`	Malware that is sold or produced as a service.
`mass-mailer`	A worm that uses email to propagate across the internet.
`metamorphic-virus`	A virus that changes its own code with each infection.
`mid-infector`	A type of file-infecting virus which places its code in the middle of files it infects. It may move a section of the original code to the end of the file, or simply push the code aside to make space for its own code.
`mobile-code`	Either code received from remote, possibly untrusted systems, but executed on a local system; or software transferred between systems (e.g across a network) and executed on a local system without explicit installation or execution by the recipient.
`multipartite-virus`	Malware that infects boot records, boot sectors, and files.
`parental-control`	A program that monitors or limits machine usage. Such programs can run undetected and can transmit monitoring information to another machine.
`password-stealer`	A type of trojan designed to steal passwords, personal data and details, or other sensitive information from an infected system.
`polymorphic-virus`	A type of virus that encrypts its code differently with each infection (or with each generation of infections).
`premium-dialer-smser`	A type of malware whose primary aim is to dial (or send SMS messages to) premium rate numbers.
`prepender`	A file-infecting virus that inserts code at the beginning of the files it infects.
`ransomware`	Malware that encrypts files on a victim's system, demanding payment of ransom in return for the access codes required to unlock files.
`remote-access-trojan`	A remote access trojan program (or RAT), is a trojan horse capable of controlling a machine through commands issued by a remote attacker.
`resource-exploiter`	A type of malware that steals a system's resources (e.g., CPU cycles), such as a bitcoin miner.

`rogue-security-software`	A fake security product that demands money to clean phony infections.
`rootkit`	A method of hiding files or processes from normal methods of monitoring; often used by malware to conceal its presence and activities.
`scareware`	A program that reports false or significantly misleading information on the presence of security risks, threats, or system issues on the target computer.
`screen-capture`	A type of malware used to capture images from the target systems screen, used for exfiltration and command and control.
`security-assessment-tool`	A program that can be used to gather information for unauthorized access to computer systems.
`shellcode`	Either a small piece of code that activates a command-line interface to a system that can be used to disable security measures, open a backdoor, or download further malicious code; or a small piece of code that opens a system up for exploitation, sometimes by not necessarily involving a command-line shell.
`spyware`	Software that gathers information and passes it to a third-party without adequate permission from the owner of the data. It may also refer to software that makes changes to a system or any of its component software, or which makes use of system resources without the full understanding and consent of the system owner.
`trackware`	Malware that traces a user's path on the Internet and sends information to third parties. Compare to spyware, which monitors system activity to capture confidential information such as passwords.
`trojan`	Malware disguised as something inert or benign.
`virus`	Self-replicating malware that requires human interaction to spread; also, self-replicating malware that runs and spreads by modifying and inserting itself into other programs or files.
`web-bug`	Code embedded in a web page or email that checks whether a user has accessed the content (e.g., a tiny, transparent GIF image).
`wiper`	Malware that delete files or entire disks on a machine.
`worm`	Self-replicating malware that propagates across a network either with or without human interaction.

12. Operating System Features

Type Name: `os-features-ov`

The Operating System Features open vocabulary is used by the following object/*property*:

- Malware Instance (`malware-instance`)
  - *os_features* (`list` of type `open-vocab`)

This vocabulary is a non-exhaustive enumeration of operating system features that may be used by malware. Each feature is specific to a particular operating system, unless otherwise specified by "multi-OS" in its description, in which case it can apply to a range of operating systems.

Vocabulary Value	Description
`login-items`	Indicates use of MacOS/OS X login items.
`plist-files`	Indicates use of MacOS/OS X plist files.
`applescript`	Indicates use of MacOS/OS X AppleScript.
`launch-agent`	Indicates use of a MacOS/OS X launch agent.
`launch-daemons`	Indicates use of MacOS/OS X launch daemons.
`kext`	Indicates the use of MacOS/OS X kernel extensions (kexts).
`login-logout-hooks`	Indicates the use of MacOS/OS X login/logout hooks.
`named-pipes`	Indicates use of named pipes (multi-OS).
`berkeley-sockets`	Indicates use of Berkeley sockets (multi-OS).
`cron`	Indicates use of the Linux or MacOS/OS X cron jobs.
`mutexes`	Indicates use of mutual exclusion objects (mutexes) (multi-OS).
`registry-keys`	Indicates use of Windows registry keys.
`services`	Indicates use of Windows services.
`powershell`	Indicates use of Windows powershell.
`ntfs-extended-attributes`	Indicates use of Windows NTFS extended attributes.
`network-shares`	Indicates use of Windows network shares.
`hooks`	Indicates use of Windows hooks.
`wmi`	Indicates use of Windows Management Instrumentation (WMI).
`task-scheduler`	Indicates use of Windows task scheduler (scheduled tasks).

`critical-sections`	Indicates of Windows critical sections.
`device-drivers`	Indicates use of device drivers (multi-OS).
`admin-network-shares`	Indicates use of Windows administrator (ADMIN$) network shares.

13. Operating Systems

Type Name: `operating-system-ov`

The Operating System open vocabulary is used by the following objects/*properties*:

- Behavior (`behavior`)
  - *attributes* (`dictionary`)
- Capability (`capability`)
  - *attributes* (`dictionary`)

This vocabulary is a non-exhaustive enumeration of operating systems.

Vocabulary Value	Description
`android-1.0.x`	Indicates the Android 1.0.x operating system.
`android-1.1.x`	Indicates the Android 1.1.x operating system.
`android-1.5.x`	Indicates the Android 1.5.x operating system.
`android-1.6.x`	Indicates the Android 1.6.x operating system.
`android-2.0.x`	Indicates the Android 2.0.x operating system.
`android-2.1.x`	Indicates the Android 2.1.x operating system.
`android-2.2.x`	Indicates the Android 2.2.x operating system.
`android-2.3.x`	Indicates the Android 2.3.x operating system.
`android-3.0.x`	Indicates the Android 3.0.x operating system.
`android-3.1.x`	Indicates the Android 3.1.x operating system.
`android-3.2.x`	Indicates the Android 3.2.x operating system.
`android-4.0.x`	Indicates the Android 4.0.x operating system.
`android-4.1.x`	Indicates the Android 4.1.x operating system.
`android-4.2.x`	Indicates the Android 4.2.x operating system.

`android-4.3.x`	Indicates the Android 4.3.x operating system.
`android-4.4.x`	Indicates the Android 4.4.x operating system.
`android-5.0.x`	Indicates the Android 5.0.x operating system.
`android-5.1.x`	Indicates the Android 5.1.x operating system.
`android-unknown-version`	Indicates an unknown version of the Android operating system.
`ios-1.0.x`	Indicates the iOS 1.0.x operating system.
`ios-1.1.x`	Indicates the iOS 1.1.x operating system.
`ios-2.0.x`	Indicates the iOS 2.0.x operating system.
`ios-2.1.x`	Indicates the iOS 2.1.x operating system.
`ios-2.2.x`	Indicates the iOS 2.2.x operating system.
`ios-3.0.x`	Indicates the iOS 3.0.x operating system.
`ios-3.1.x`	Indicates the iOS 3.1.x operating system.
`ios-3.2.x`	Indicates the iOS 3.2.x operating system.
`ios-4.0.x`	Indicates the iOS 4.0.x operating system.
`ios-4.1.x`	Indicates the iOS 4.1.x operating system.
`ios-4.2.x`	Indicates the iOS 4.2.x operating system.
`ios-4.3.x`	Indicates the iOS 4.3.x operating system.
`ios-5.0.x`	Indicates the iOS 5.0.x operating system.
`ios-5.1.x`	Indicates the iOS 5.1.x operating system.
`ios-6.0.x`	Indicates the iOS 6.0.x operating system.
`ios-6.1.x`	Indicates the iOS 6.1.x operating system.
`ios-7.0.x`	Indicates the iOS 7.0.x operating system.
`ios-7.1.x`	Indicates the iOS 7.1.x operating system.
`ios-8.0.x`	Indicates the iOS 8.0.x operating system.
`ios-8.1.x`	Indicates the iOS 8.1.x operating system.
`ios-8.2.x`	Indicates the iOS 8.2.x operating system.

`ios-8.3.x`	Indicates the iOS 8.3.x operating system.
`ios-8.4.x`	Indicates the iOS 8.4.x operating system.
`ios-9.0.x`	Indicates the iOS 9.0.x operating system.
`ios-9.1.x`	Indicates the iOS 9.1.x operating system.
`ios-9.2.x`	Indicates the iOS 9.2.x operating system.
`ios-9.3.x`	Indicates the iOS 9.3.x operating system.
`ios-10.0.x`	Indicates the iOS 10.0.x operating system.
`ios-10.1.x`	Indicates the iOS 10.1.x operating system.
`ios-unknown-version`	Indicates an unknown version of the iOS operating system.
`linux-kernel-2.4.x`	Indicates version 2.4.x of the linux kernel.
`linux-kernel-2.6.x`	Indicates version 2.6.x of the linux kernel.
`linux-kernel-3.0.x`	Indicates version 3.0.x of the linux kernel.
`linux-kernel-3.1.x`	Indicates version 3.1.x of the linux kernel.
`linux-kernel-3.2.x`	Indicates version 3.2.x of the linux kernel.
`linux-kernel-3.3.x`	Indicates version 3.3.x of the linux kernel.
`linux-kernel-3.4.x`	Indicates version 3.4.x of the linux kernel.
`linux-kernel-3.5.x`	Indicates version 3.5.x of the linux kernel.
`linux-kernel-3.6.x`	Indicates version 3.6.x of the linux kernel.
`linux-kernel-3.7.x`	Indicates version 3.7.x of the linux kernel.
`linux-kernel-3.8.x`	Indicates version 3.8.x of the linux kernel.
`linux-kernel-3.9.x`	Indicates version 3.9.x of the linux kernel.
`linux-kernel-3.10.x`	Indicates version 3.10.x of the linux kernel.
`linux-kernel-3.11.x`	Indicates version 3.11.x of the linux kernel.
`linux-kernel-3.12.x`	Indicates version 3.12.x of the linux kernel.
`linux-kernel-3.13.x`	Indicates version 3.13.x of the linux kernel.
`linux-kernel-3.14.x`	Indicates version 3.14.x of the linux kernel.

`linux-kernel-3.15.x`	Indicates version 3.15.x of the linux kernel.
`linux-kernel-3.16.x`	Indicates version 3.16.x of the linux kernel.
`linux-kernel-3.17.x`	Indicates version 3.17.x of the linux kernel.
`linux-kernel-3.18.x`	Indicates version 3.18.x of the linux kernel.
`linux-kernel-3.19.x`	Indicates version 3.19.x of the linux kernel.
`linux-kernel-4.0.x`	Indicates version 4.0.x of the linux kernel.
`linux-kernel-4.1.x`	Indicates version 4.1.x of the linux kernel.
`linux-unknown-version`	Indicates an unknown version of the linux kernel.
`mac-os-x-10.0.x`	Indicates the Mac OS X 10.0.x operating system.
`mac-os-x-10.1.x`	Indicates the Mac OS X 10.1.x operating system.
`mac-os-x-10.2.x`	Indicates the Mac OS X 10.2.x operating system.
`mac-os-x-10.3.x`	Indicates the Mac OS X 10.3.x operating system.
`mac-os-x-10.4.x`	Indicates the Mac OS X 10.4.x operating system.
`mac-os-x-10.5.x`	Indicates the Mac OS X 10.5.x operating system.
`mac-os-x-10.6.x`	Indicates the Mac OS X 10.6.x operating system.
`mac-os-x-10.7.x`	Indicates the Mac OS X 10.7.x operating system.
`mac-os-x-10.8.x`	Indicates the Mac OS X 10.8.x operating system.
`mac-os-x-10.9.x`	Indicates the Mac OS X 10.9.x operating system.
`mac-os-x-10.10.x`	Indicates the Mac OS X 10.10.x operating system.
`mac-os-x-10.11.x`	Indicates the Mac OS X 10.11.x operating system.
`mac-os-x-unknown-version`	Indicates an unknown version of the Mac OS X operating system.
`windows-7`	Indicates the Windows 7 operating system.
`windows-7-sp1`	Indicates the Windows 7 SP1 operating system.
`windows-8`	Indicates the Windows 8 operating system.
`windows-8.1`	Indicates the Windows 8.1 operating system.

`windows-10`	Indicates the Windows 10 operating system.
`windows-server-2003`	Indicates the Windows Server 2003 operating system.
`windows-server-2003-sp1`	Indicates the Windows Server 2003 SP1 operating system.
`windows-server-2003-sp2`	Indicates the Windows Server 2003 SP2 operating system.
`windows-server-2008`	Indicates the Windows Server 2008 operating system.
`windows-server-2008-sp1`	Indicates the Windows Server 2008 SP1 operating system.
`windows-server-2008-sp2`	Indicates the Windows Server 2008 SP2 operating system.
`windows-server-2008-r2`	Indicates the Windows Server 2008 r2 operating system.
`windows-server-2008-r2-sp1`	Indicates the Windows Server 2008 R2 SP1 operating system.
`windows-server-2012`	Indicates the Windows Server 2012 operating system.
`windows-server-2012-r2`	Indicates the Windows Server 2012 r2 operating system.
`windows-vista`	Indicates the Windows Vista operating system.
`windows-vista-sp1`	Indicates the Windows Vista SP1 operating system.
`windows-vista-sp2`	Indicates the Windows Vista SP2 operating system.
`windows-xp`	Indicates the Windows XP operating system.
`windows-xp-sp1`	Indicates the Windows XP SP1 operating system.
`windows-xp-sp2`	Indicates the Windows XP SP2 operating system.
`windows-xp-sp3`	Indicates the Windows XP SP3 operating system.
`windows-unknown-version`	Indicates an unknown version of the Windows operating system.

14. Obfuscation Methods

Type Name: `obfuscation-method-ov`

The Obfuscation Method open vocabulary is used in the following object/*property*:

- Malware Instance (`malware-instance`)
  - static_features (`static-features`)
    - obfuscation_methods (`list` of type `binary-obfuscation`)
      - *method* (`open-vocab`)

This vocabulary is a non-exhaustive enumeration of obfuscation methods used in obfuscating a binary associated with a Malware Instance.

Vocabulary Value	Description
`packing`	Packing of the malware instance code, not including encryption (the `code-encryption` value defined below should be used for encrypted code).
`code-encryption`	Encryption of the malware instance code.
`dead-code-insertion`	Dead code inserted in the malware instance.
`entry-point-obfuscation`	Obfuscation of the malware instance entry point.
`import-address-table-obfuscation`	Obfuscation of the malware instance import address table.
`interleaving-code`	Code interleaving in the malware instance (code is split into sections that are rearranged and connected by unconditional jumps).
`symbolic-obfuscation`	Obfuscation of the malware instance symbols.
`string-obfuscation`	Obfuscation of the malware instance strings.
`subroutine-reordering`	Reordering of subroutines in the malware instance.
`code-transposition`	Reordering of instructions in the malware instance.
`instruction-substitution`	Substitution of malware instance instructions with semantic equivalents.
`register-reassignment`	Replacement of unused registers with those containing malicious code.

15. Processor Architectures

Vocabulary Name: `processor-architecture-ov`

The Process Architecture open vocabulary is used in the following object/*property*:

- Malware Instance (`malware-instance`)
  - *architecture_execution_envs* (`open-vocab`)

Vocabulary Value	Description
`x86`	The 32-bit x86 architecture.

`x86-64`	The 64-bit x86 architecture.
`ia-64`	The 64-bit IA (Itanium) architecture.
`powerpc`	The PowerPC architecture.
`arm`	The ARM architecture.
`alpha`	The Alpha architecture.
`sparc`	The SPARC architecture.
`mips`	The MIPS architecture.

16. Refined Capabilities

Type Name: `refined-capability-ov`

The Refined Capability open vocabulary is used in the following object/*property*:

- Capability (`capability`)
  - *name* (`open-vocab`)

Vocabulary Value	Description
`access-control-degradation`	Indicates that the malware instance or family is able to bypass or disable access control mechanisms designed to prevent unauthorized or unprivileged use or execution of applications or files.
`anti-debugging`	Indicates that the malware instance or family is able to prevent itself from being debugged and/or from being run in a debugger or is able to make debugging more difficult.
`anti-disassembly`	Indicates that the malware instance or family is able to prevent itself from being disassembled or make disassembly more difficult.
`anti-emulation`	Indicates that the malware instance or family is able to prevent its execution inside of an emulator or is able to make emulation more difficult.
`anti-memory-forensics`	Indicates that the malware instance or family is able to prevent or make memory forensics more difficult.

`anti-sandbox`	Indicates that the malware instance or family is able to prevent sandbox-based behavioral analysis or make it more difficult.
`anti-virus-evasion`	Indicates that the malware instance or family is able to evade detection by anti-virus tools.
`anti-vm`	Indicates that the malware instance or family is able to prevent virtual machine (VM) based behavioral analysis or make it more difficult.
`authentication-credentials-theft`	Indicates that the malware instance is able to steal authentication credentials.
`clean-traces-of-infection`	Indicates that the malware instance or family is able to clean traces of its infection (e.g., file system artifacts) from a system.
`communicate-with-c2-server`	Indicates that the malware instance or family is able to communicate (i.e., send or receive data) with a command and control (C2) server.
`compromise-data-availability`	Indicates that the malware instance or family is able to compromise the availability of data on the local system on which it is executing and/or one or more remote systems.
`compromise-system-availability`	Indicates that the malware instance or family is able to compromise the availability of the local system on which it is executing and/or one or more remote systems.
`consume-system-resources`	Indicates that the malware instance or family is able to consume system resources for its own purposes, such as password cracking.
`continuous-execution`	Indicates that the malware instance or family is able to continue to execute on a system after significant system events, such as a system reboot.
`data-integrity-violation`	Indicates that the malware instance or family is able to compromise the integrity of some data that resides on (e.g., in the case of files) or is received/transmitted (e.g., in the case of network traffic) by the system on which it is executing.
`data-obfuscation`	Indicates that the malware instance or family is able to obfuscate data that will be exfiltrated.

`data-staging`	Indicates that the malware instance or family is able to gather, prepare, and stage data for exfiltration.
`determine-c2-server`	Indicates that the malware instance or family is able to identify one or more command and control (C2) servers with which to communicate.
`email-spam`	Indicates that the malware instance or family is able to send spam email messages.
`ensure-compatibility`	Indicates that the malware instance or family is able to manipulate or modify the system on which it executes to ensure that it is able to continue executing.
`environment-awareness`	Indicates that the malware instance or family can fingerprint or otherwise identify the environment in which it is executing, for the purpose of altering its behavior based on this environment.
`file-infection`	Indicates that the malware instance or family is able to infect one or more files on the system on which it executes.
`hide-artifacts`	Indicates that the malware instance or family is able to hide its artifacts, such as files and open ports.
`hide-executing-code`	Indicates that the malware instance or family is able to hide its executing code.
`hide-non-executing-code`	Indicates that the malware instance or family is able to hide its non-executing code.
`host-configuration-probing`	Indicates that the malware instance or family is able to probe the configuration of the host system on which it executes.
`information-gathering-for-improvement`	Indicates that the malware instance or family is able to gather information from its environment to make itself less likely to be detected.
`input-peripheral-capture`	Indicates that the malware instance or family is able to capture data from a system's input peripheral devices, such as a keyboard or mouse.

`install-other-components`	Indicates that the malware instance or family is able to install additional components. This encompasses the dropping/downloading of other malicious components such as libraries, other malware, and tools.
`local-machine-control`	Indicates that the malware instance or family is able to control the machine on which it is executing.
`network-environment-probing`	Indicates that the malware instance or family is able to probe the properties of its network environment, e.g. to determine whether it funnels traffic through a proxy.
`os-security-feature-degradation`	Indicates that the malware instance or family is able to bypass or disable operating system (OS) security mechanisms.
`output-peripheral-capture`	Indicates that the malware instance or family captures data sent to a system's output peripherals, such as a display.
`physical-entity-destruction`	Indicates that the malware instance or family is able to destroy physical entities.
`prevent-artifact-access`	Indicates that the malware instance or family is able to prevent its artifacts (e.g., files, registry keys, etc.) from being accessed.
`prevent-artifact-deletion`	Indicates that the malware instance or family is able to prevent its artifacts (e.g., files, registry keys, etc.) from being deleted.
`remote-machine-access`	Indicates that the malware instance or family is able to access one or more remote machines.
`remote-machine-infection`	Indicates that the malware instance or family is able to self-propagate to a remote machine or infect a machine with malware that is different than itself.
`security-software-degradation`	Indicates that the malware instance or family is able to bypass or disable security programs running on a system, either by stopping them from executing or by making changes to their code or configuration parameters.
`security-software-evasion`	Indicates that the malware instance or family is able to evade security software (e.g., anti-virus tools).

`self-modification`	Indicates that the malware instance or family is able to modify itself.
`service-provider-security-feature-degradation`	Indicates that the malware instance or family is able to bypass or disable mobile device service provider security features that would otherwise identify or notify users of its presence.
`stored-information-theft`	Indicates that the malware instance or family is able to steal information stored on a system (e.g., files).
`system-interface-data-capture`	Indicates that the malware instance or family is able to capture data from a system's logical or physical interfaces, such as from a network interface.
`system-operational-integrity-violation`	Indicates that the malware instance or family is able to compromise the operational integrity of the system on which it is executing and/or one or more remote systems, e.g., by causing them to operate beyond their set of specified operational parameters.
`system-re-infection`	Indicates that the malware instance or family is able to re-infect a system after one or more of its components have been removed.
`system-state-data-capture`	Indicates that the malware instance or family is able to capture information about a system's state (e.g., data currently in its RAM).
`system-update-degradation`	Indicates that the malware instance or family is able to disable the downloading and installation of system updates and patches.
`user-data-theft`	Indicates that the malware instance or family is able to steal data associated with one or more users (e.g., browser history).
`virtual-entity-destruction`	Indicates that the malware instance or family is able to destroy a virtual entity.