Splunk® Enterprise Security
Administer Splunk Enterprise Security 7.0.1

Supported types of threat intelligence in Splunk Enterprise Security

Generated: 5/06/2022 9:11 am

Supported types of threat intelligence in Splunk Enterprise Security

Splunk Enterprise Security supports several types of threat intelligence. The supported types of threat intelligence correspond to the KV Store collections in which the threat intelligence is stored.

The threatlist modular input parses downloaded and uploaded files and adds indicators to these collections. Files can contain any combination of indicators.

Threat collection in KV Store	Supported IOC data types	Local lookup file	Required headers in lookup file with no spaces after commas
certificate_intel	X509 Certificates	Local Certificate Intel	`certificate_issuer,certificate_subject,certificate_issuer_organization,certificat _organization,certificate_serial,certificate_issuer_unit,certificate_subject_unit`
email_intel	Email	Local Email Intel	`description,src_user,subject,weight`
file_intel	File names or hashes	Local File Intel	`description,file_hash,file_name,weight`
http_intel	URLs	Local HTTP Intel	`description,http_referrer,http_user_agent,url,weight`
ip_intel	IP addresses	Local IP Intel	`description,ip,weight`
	domains	Local Domain Intel	`description,domain,weight`
process_intel	Processes	Local Process Intel	`description,process,process_file_name,weight`
registry_intel	Registry entries	Local Registry Intel	`description,registry_path,registry_value_name,registry_value_text,weight`
service_intel	Services	Local Service Intel	`description,service,service_file_hash,service_dll_file_hash,weight`
user_intel	Users	Local User Intel	`description,user,weight`

The `collections.conf` file in the `DA-ESS-ThreatIntelligence` subdirectory lists these KV Store collections.

The `inputs.conf.spec` file in the `SA-ThreatIntelligence` subdirectory lists the specifications for settings used by the threatlist modular input, such as weight:

```
weight = <integer>
* [Required]
* The weight assigned to the intelligence.
* Between 1 and 100.
* A higher weight will result in higher risk scores for corresponding intelligence matches.
* Defaults to 60.
```

Example of observable values and primary keys

Using the `http_intel` collection as an example, consider a threat document called my_threat_intel.csv. An observable value in the file is any value in the â â `http_referrer`, `http_user_agent`, and `url` fields for matching against threat values in your raw data. A row is added to the `http_intel` threat collection for each observable found in my_threat_intel.csv. The last value is used to construct the primary key if duplicate observables exist. If observable values are missing from the CSV file, the first non-empty value in the CSV file is used to construct the primary key. If you don't want to overwrite data, make sure not to use any words such as "null", "N/A", "blank", or "none" throughout the CSV file when data is unavailable, just leave those fields empty.

Consider a source file with duplicates in the `http_user_agent` fields, such as the following:

```
description,http_referrer,http_user_agent,url,weight
ThreatA,,UseragentA,https://urlA,3
ThreatB,,UseragentA,https://urlB,3
```

A search for `|inputlookup http_intel` returns the following results:

Description	Http_user_agent	Threat_key	Time	URL
ThreatA	UseragentA	my_threat_intel	1626387748	https://urlA
ThreatB	UseragentA	my_threat_intel	1626387748	https://urlB
ThreatB	UseragentA	my_threat_intel	1626387748	https://urlB

Based on the two rows in the CSV file, three observable values are discovered: the url for ThreatA, the url for ThreatB, and the http_user_agent for ThreatB. Notice that http_user_agent for ThreatA is overwritten by ThreatB because the name UseragentA is a duplicate observable value. The primary key in the threat intel collection looks as follows:

```
my_threat_intel|https://urlA
my_threat_intel|https://urlB
my_threat_intel|UseragentA
```

Consider a source file without duplicates in the http_user_agent fields, such as the following:

```
description,http_referrer,http_user_agent,url,weight
ThreatA,,UseragentA,https://urlA,3
ThreatB,,UseragentB,https://urlB,3
```

A search for |inputlookup http_intel returns the following results:

Description	Http_user_agent	Threat_key	Time	URL
ThreatA	UseragentA	my_threat_intel	1626387748	https://urlA
ThreatA	UseragentA	my_threat_intel	1626387748	https://urlA
ThreatB	UseragentB	my_threat_intel	1626387748	https://urlB
ThreatB	UseragentB	my_threat_intel	1626387748	https://urlB

Based on the two rows in the CSV, four observable values are discovered: the url for ThreatA, the http_user_agent for ThreatA, the url for ThreatB, and the http_user_agent for ThreatB. There are no duplicates, so every value is displayed. The primary key in the threat intel collection looks as follows:

```
my_threat_intel|https://urlA
my_threat_intel|UseragentA
my_threat_intel|https://urlB
my_threat_intel|UseragentB
```