ЯEVERSINGLABS

Spectra Analyze – Advanced Search and Pivoting
Cheat Sheet

Keywords and Aliases

Advanced Search and Pivoting on Spectra Analyze supports more than 100 keywords, making it possible to build more than 500 unique search queries using boolean operators and keyword auto-completion.

Start typing into the search box and a dropdown list with all matching keywords will open. Short explanations and examples are provided in the dropdown.

Each expression must contain at least one keyword and one value. The basic formula is:

```
keyword:value OPERATOR keyword2:value
OPERATOR keyword3:[value1, value2,...]
```

Highlight a keyword in the dropdown and press Enter to add it to the search box. **If the keyword supports predefined values, they will be listed in the dropdown.**

Some keywords support **aliases**. A keyword and its aliases can be used interchangeably to get the same results. Aliases are indicated in the dropdown list with an ALIAS label.

Example aliases:

```
av-count = antivirus = positives = p
av-detection = engines
av-vendorname = vendorname
available = in
firstseen = fs
lastseen = ls
itw = uri-source
sampletype = filetype = type
uri-config = c2
```

Lists, Operators

Items in a list should be comma-separated and enclosed in square brackets. Comma means OR, so the search results will include any of the listed items.

```
keyword:[value1, value2, value3]
```

The maximum number of hashes that can be in a list is limited as follows: **SHA1**: 50, **MD5**: 59, **SHA256**: 32

Supported operators are **AND**, **OR**, **NOT**. They are case-insensitive. If an operator is not provided, AND is used as the default.

OR is used to search for multiple values of a single keyword:

```
classification:suspicious OR
classification:malicious
```

AND, OR, NOT are used to combine keywords (parentheses are used to combine different values of the same keyword):

```
type:*binary* AND (av-detection:trojan
OR av-detection:wannacry) NOT
classification:malicious
```

Wildcards

Wildcards can be used for fuzzy searching.

? as a substitute for any single character:

```
av-detection:emo?et
```

This query matches all samples with the threat name "Emotet", but also any other variant where the first letter T is replaced by another (such as Emo**n**et).

* as a substitute for any number of characters:

```
av-detection:*emo*
```

This query matches all samples that have the string "emo" anywhere in their threat name (such as W**emo**sis, R**emo**ra, T**emo**nde).

Find samples with TOR-related network references:

```
uri:[*.tor, *.onion]
```

Find a specific IP range:

```
ipv4:2.?.29:*
```

Find suspicious Linux samples with a risk score of 10:

```
type:*ELF* classification:suspicious
riskscore:10
```

Quotation Marks

Quotation marks are used to escape restricted characters and reserved words, and to search for phrases that contain spaces. **Wildcard characters (* and ?) can be used within quotation marks.**

Restricted characters: ([:])

```
pdb:"C:\Windows*"
```

```
uri:"https://asfr.in*"
```

Reserved words (all case variations): AND OR NOT

```
cert.subject-name:"AND"
```

```
cert-issuer:name:"not"
```

Non-keyword searches containing commas, colons or brackets must be enclosed in quotation marks:

```
"http://evildomain.com/gate.php?12,3586"
```

Searching for phrases with spaces:

```
document-author:"Microsoft Corp"
```

```
threatname:"WIN32.PUA.casino eldorado"
```

Non-Keyword Queries

Search queries can be quickly built without using keywords. Non-keyword searches are available only for a subset of indicators of compromise, such as:

```
SHA1, SHA256, MD5, URLs, IP addresses,
domains, email addresses
```

Non-keyword searches can be performed as standalone queries containing one or more non-keyword values, or be combined with keywords.

When combining non-keyword searches with keywords, consecutive non-keyword values will be enclosed in brackets and the spaces between them will be interpreted as **OR**.

Spaces between non-keyword values and keywords will be interpreted as **AND**. This makes the order of keywords and non-keyword values important.

```
127.0.0.1 "2620:119:35::35" example.com
```

```
NOT *@mockmail.com "https://hope-bd.com/
googledocs.php" AND NOT
0000038704cb5f0e1bd87d6a75e904529af0d6ac
class:MALICIOUS
```

After performing a search, final transformed queries will be returned in the Advanced Search box and added to the Recent Queries list, so that they can be saved as favorites or shared with other users.

Ranges and Comparisons

To search for a range of values, use the formula:

```
keyword:[value1 TO value2]
```

Some keywords that support searching for value ranges:

```
av-count, filecount, firstseen,
lastanalysis, lastseen, size, riskscore
```

Find PDF files between 5 and 10 MB in size:

```
type:*PDF* size:[5000000 TO 10000000]
```

Use wildcards * to create open-ended ranges and search for greater/less-than values:

Greater than:
```
keyword:[value TO *]
```

Less than:
```
keyword:[* TO value]
```

Find suspicious PDF files with at most 5 AV detections:

```
type:*PDF* av-count:[* TO 5]
classification:suspicious
```

Find PDF files with a risk score of 3 and greater:

```
type:*PDF* riskscore:[3 TO *]
```

Search Tips and Examples

The `OR` operator can't be used in lists:

```
av-detection:[emotet OR wannacry]   ❌
```

```
av-detection:[trojan, wannacry]   ✅
```

```
av-detection:trojan OR av-
detection:wannacry   ✅
```

`threatname` looks for full partial malware family names:

```
threatname:Trickbot
```

```
threatname:[trojan, backdoor]
```

```
threatname:CVE-2018-8373
```

The `filename` keyword looks at the full name of the file. It can be used to find files by extension, or to find specific filenames:

```
filename:*exe
```

```
filename:Invoice* OR filename:Order*
sampletype:document
```

`av-detection` looks for the specific string through all detection strings that the sample received when it was scanned by AV engines:

```
av-detection:Emotet
```

The `sampletype` keyword looks at the file type as detected by Spectra Core. Supported file types are listed in the dropdown.

Find email file formats:

```
sampletype:[msg, MIME,hqx,uue]
```

The `pe-original-name` keyword applies only to PE files. It looks for the filename as written in the **Version info > Original File Name** metadata field.

```
pe-original-name:ccleaner.exe AND
classification:malicious
```

The `uri` keyword searches for files containing specific URIs anywhere in the file (found during static and dynamic analysis):

```
uri:/wp-content/
```

```
uri:"https://bankofamerica" NOT
uri:"https://secure.bankofamerica.com"
```

The `uri-source` keyword looks for files that were downloaded from specific URIs:

```
uri-source:[softonic.com,*cnet.com]
```

The `email` keyword finds files containing a specific email address anywhere in the file. The address can also be the file source:

```
email:*@microsoft.com
```

The `tag` keyword can be used to find files based on their capabilities and behaviors as detected during static analysis. For example, it can find files signed with blacklisted certificates, files that execute other files or files that target specific platforms.

Find PDFs with embedded scripts:

```
classification:known sampletype:pdf
tag:capability-scripting
```

Find files signed with valid certificates issued by trusted vendors:

```
cert-issuer-org:[microsoft, google,
apple] tag:cert-signed NOT tag:cert-
invalid NOT tag:cert-expired
```