COLUMBUS STATE UNIVERSITY

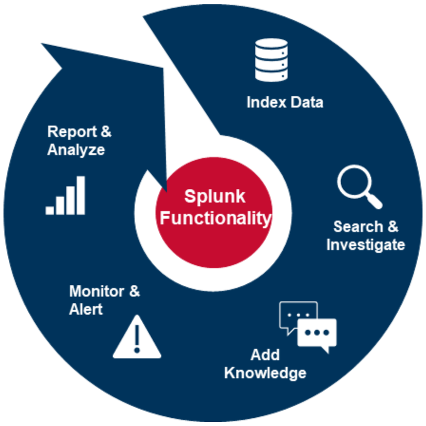
Splunk

Author: Ehab Bedir | bedir_ehab@columbusstate.edu | Spring 2020
Edited: Patrick White | white_patrick2@columbusstate.edu | Summer 2020

Splunk

Splunk Enterprise, with *Splunk User Behavior Analysis*, form an analytical driven SIEM solution system [14]. Like *Security Onion*, it can search, analyze, and visualize the data gathered from the IT infrastructure system (websites, applications, sensors, and devices). However, the difference is that *Splunk* adds all the data to an intelligent searchable index and parses the data stream into individual events that can be retrieved and viewed, in addition to searching the alerts either in historical or real-time. In other words, it structures the unstructured data to allow all sorts of insights into the business and providing reports. The reports may include security issues, human behavior, hardware monitoring, application issues, and custom requirements configured by the user.

For more efficiency, *Splunk's* environment can be extended by adding more apps, which is a collection of *Splunk* configuration files, such as *Splunk Enterprise Security* (ES) or *Splunk User Behavior Analytics* (UBA). Also, apps developed by other vendors can be added, such as *Security Onion* on the Splunk platform. Apps extend *Splunk's* functionality. When *Splunk* is integrated with *Security Onion*, for instance, it correlates events and incorporates field extractions and reports for *Sguil*, *Zeek*, and *OSSEC*.

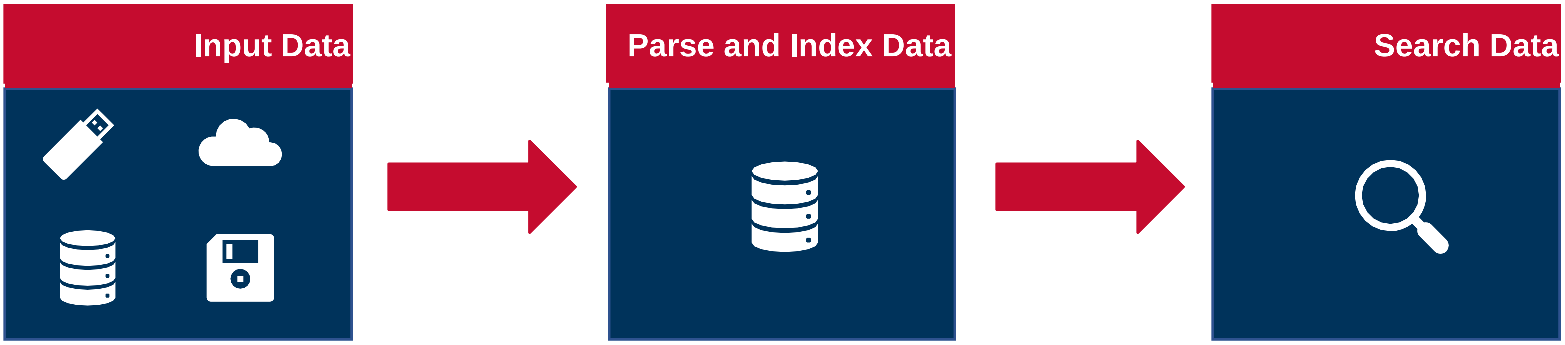


Splunk Processing Components

The *Splunk* platform uses four components to process machine data:

1. **Input** – The input is the source of data. Adding data to *Splunk* can be done with three source options:
   a. By uploading local files from the computer, such as CSV files;
   b. Through monitoring a specified path, HTTP event, network ports, and scripts; and
   c. With the forwarder method, which collects data from remote external instance or third-party system and forwards that data into the indexers for processing. For example, if we are monitoring a web server, we will install a forwarder on this server. The server will then send data to our indexer. There are three types of forwarders universal, heavy, and light. More information about different forwarder types is available in the link below. https://docs.splunk.com/Documentation/Splunk/latest/Data/Usingforwardingagents;

2. **Parsing** – In this step, *Splunk* examines analysis and identifies the timestamps, host, and source type of the data. This process occurs inside indexers or heavy forwarders;
3. **Indexing** – After parsing the data is indexed. The data is stored in the index as an event (single raw of data) in five buckets [15], it then creates several files organized in sets of directories by time. This makes the searches much faster. When searching the data, Splunk will only open the directories that match the time frame of your search; and
4. **Searching** – In the search phase the user interacts with the data using the search head component. The search head allows users to use the Splunk search language to search the index data. The search head also provides users with various tools, such as dashboards, reports, and visualizations.



Input Data

Parse and Index Data

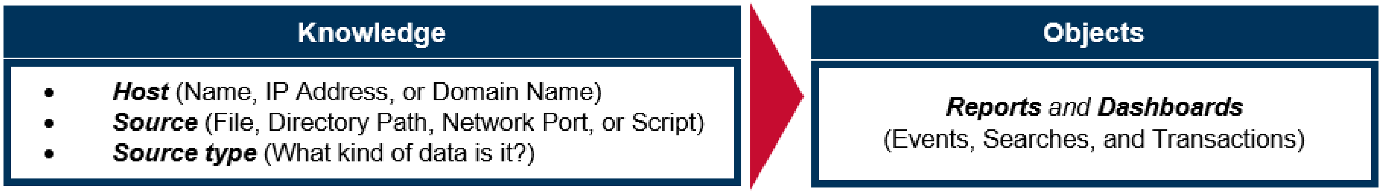
Search Data

Splunk Deployment Types

Splunk can be deployed in four different types of configurations:

1. **Departmental Deployment** – This is when Splunk installed on a Windows or Linux machine and has a single search head/indexer, up to 10 forwarders, and up to 10 users only;
2. **Small Enterprise Deployment** – In this this model a single search head is used with two or three indexers, and up to 200 forwarders;
3. **Medium Enterprise Deployment** – This model utilizes a search head cluster, a group of search heads, governed by a deployer. It has several indexers and up to thousands of forwarders; and
4. **Large Enterprise Deployment** – The largest Splunk deployment employs a search head cluster, an indexer cluster, and thousands of forwarders.

Tier 1 analysts will mainly interact with the search and reporting application interface for searching and analyzing data and creating knowledge objects. Splunk uses knowledge objects (see Figure 3) to give form to the chaos of raw data or machine data. They are how a user can create a multi-dimensional data structure that enables them to infer meaning and actionable insights from a steady stream of raw data.



Knowledge

- ***Host*** (Name, IP Address, or Domain Name)
- ***Source*** (File, Directory Path, Network Port, or Script)
- ***Source type*** (What kind of data is it?)

Objects

Reports** and **Dashboards
(Events, Searches, and Transactions)

Figure 3.Splunk Knowledge Objects

Searching in Splunk

Searching helps users find what they need. This information can be filtered, summarized, and visualized with Splunk. Users can access the search functionality by navigating to the Search & Reporting tab within the app (see Figure 4).

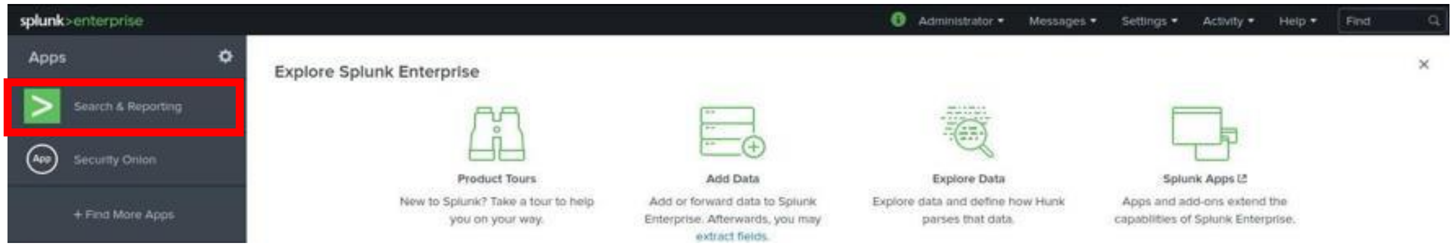
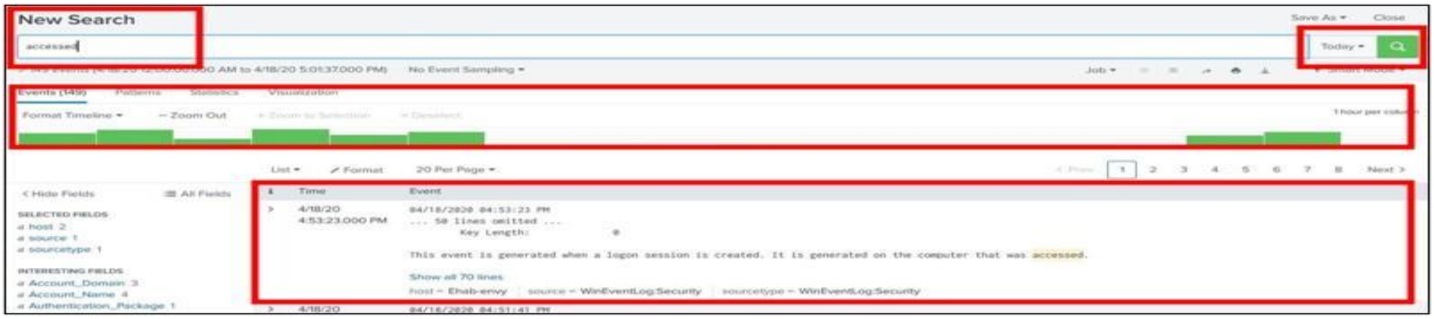

splunk>enterprise
Administrator ▾
Messages ▾
Settings ▾
Activity ▾
Help ▾
Find
Apps
Search & Reporting
App
Security Onion
+ Find More Apps
Explore Splunk Enterprise
Product Tours
New to Splunk? Take a tour to help you on your way.
Add Data
Add or forward data to Splunk Enterprise. Afterwards, you may extract fields.
Explore Data
Explore data and define how Hunk parses that data.
Splunk Apps
Apps and add-ons extend the capabilities of Splunk Enterprise.

Figure 4. Splunk Search & Reporting App

When the user types a search command in the search bar, a new search interface will show up, which contains the event list tab that shows the retrieved search events, the event pattern tab, statistics, visualization tabs, and the time range (see Figure 5).



Search: accessed

149 events (4/18/20 12:00:00.000 AM to 4/18/20 5:01:37.000 PM)

```
> 4/18/20             04/18/2020 04:53:23 PM
  4:53:23.000 PM      ... 50 lines omitted ...
                          Key Length:              0

                      This event is generated when a logon session is created. It is generated on the computer that was accessed.

                      Show all 70 lines
                      host = Ehab-envy | source = WinEventLog:Security | sourcetype = WinEventLog:Security
> 4/18/20             04/18/2020 04:51:41 PM
```

Figure 5. Splunk New Search

Search Anatomy

A search contains a series of commands that are bounded by pipe (|) characters [16]. “The pipe character tells Splunk software to use the output or result of one command (to the left of the pipe) as the input for the next command (to the right of the pipe) [16]”. Also, the first term after the pipe is the name of the search command. In the following example top and fields are the commands while searching for events were a syslog error occurred (see Figure 6).

```
sourcetype=”syslog ERROR” | top user | fields – percent
```

Figure 6.. Example Splunk Search Command

For better understanding on how search commands act on the data, let us imagine that all indexed data is stored as a table. Each search command reshapes the table until only the user’s requested data is left. For demonstration (see Figure 7), we will use the same search command above [16]:

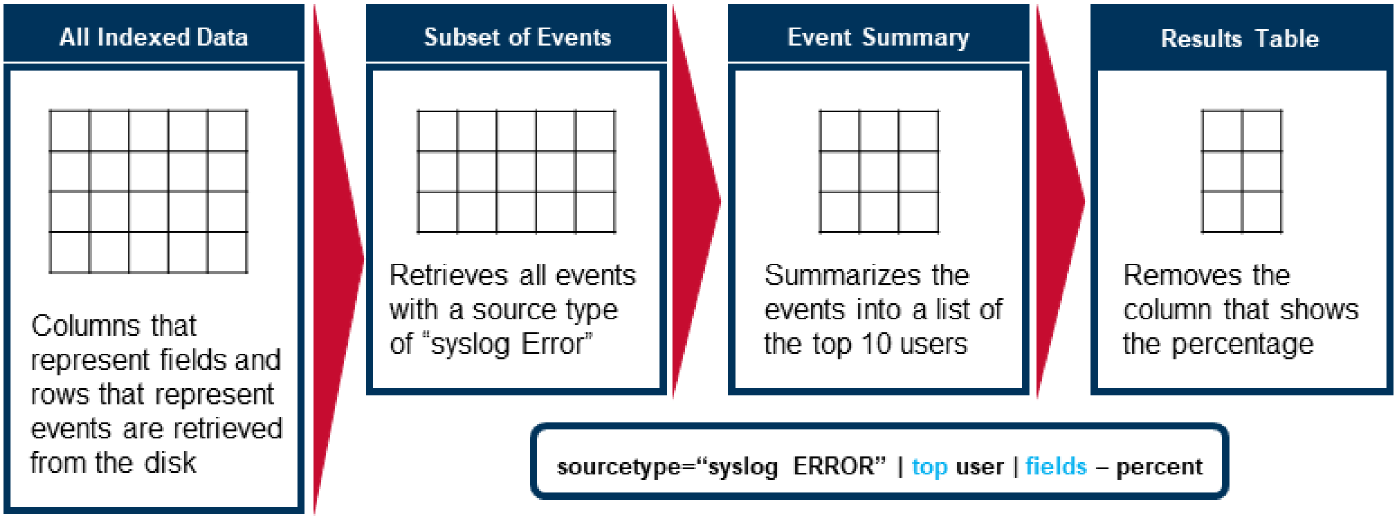

All Indexed Data
Columns that represent fields and rows that represent events are retrieved from the disk
Subset of Events
Retrieves all events with a source type of “syslog Error”
Event Summary
Summarizes the events into a list of the top 10 users
Results Table
Removes the column that shows the percentage
sourcetype=“syslog ERROR” | top user | fields – percent

Figure 7. Search Process

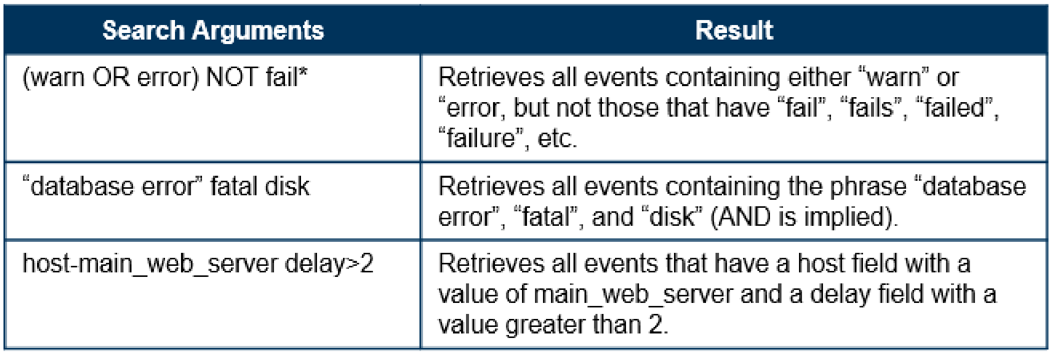
Search Terms & Field Operators

Splunk uses characters in queries to analyze search results. Common functions are:

- **Wildcards** – To match an unlimited number of characters in a string, you can use an asterisk (*) at the end of a term, such as access* or fail*. To avoid searching the entire event, you can indicate a field value pair whenever possible. For instance, status=” fail*” [17]. However, there are some conditions that wildcards should be avoided, such as using it in the middle of a string to match punctuation or using it as a prefix.

- **Booleans** – All Boolean operators are case sensitive and need to be uppercase. The order of evaluation of the three types of Boolean operations is as follows:
    1. NOT
    2. OR
    3. AND

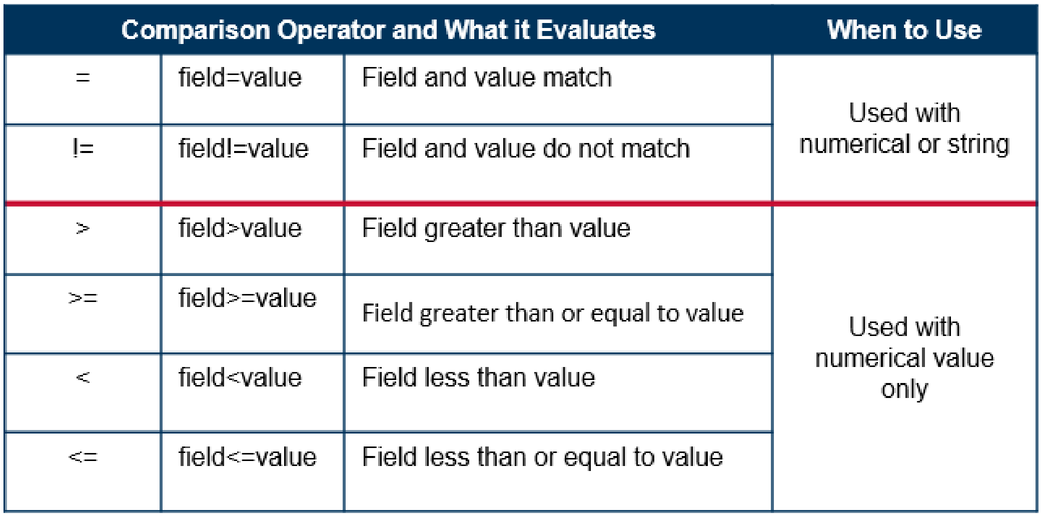
  Parentheses can be used to influence the order of operation. If no Boolean is used in a search, then AND is implied (see Table 1).



Search Arguments	Result
(warn OR error) NOT fail*	Retrieves all events containing either “warn” or “error, but not those that have “fail”, “fails”, “failed”, “failure”, etc.
“database error” fatal disk	Retrieves all events containing the phrase “database error”, “fatal”, and “disk” (AND is implied).
host-main_web_server delay>2	Retrieves all events that have a host field with a value of main_web_server and a delay field with a value greater than 2.

Table 1. Boolean Search Example

- **Escaping Characters and Quotes** – Quotes are needed with phrases and field values that contain breaking characters, such as white spaces, commas, pipes, or brackets. For instance, host=web09 is ok, but if the host value has spaces, you will need quotes with the value, as in host=”webserver #9”. Also, to search for reserved keywords (AND, OR, NOT, etc.), use quotes [18]. To escape quotes characters and pipes, the backslash character (\) is used. For example, to find the phrase—Splunk changed “time management” for me—you would search for: “Splunk changed \”time management\” for me”.
- **Operators and Expressions** – In Splunk comparison operators can be used. The user should be aware that certain expressions can only be used with numerical values (see Table 2).



Comparison Operator and What it Evaluates			When to Use
=	field=value	Field and value match	Used with numerical or string
!=	field!=value	Field and value do not match	
>	field>value	Field greater than value	Used with numerical value only
>=	field>=value	Field greater than or equal to value	
<	field<value	Field less than value	
<=	field<=value	Field less than or equal to value	

Table 2. Comparison Operators

Search Language

The Splunk search language consists of five main components that can be stringed together to meet the user's needs (see Figure 8):

- **Search term** – (Explained above)
  - Keywords Example: error login;
  - Quoted phrases Example: "policy-violation";
  - Boolean expressions Example: login NOT (error OR fail);
  - Fields Example: host=so, Alert_Message="attempted-user";
  - Wildcards Example: fail*; and
  - Comparison expressions Example: fieldA!=fieldB.
- **Commands** – Tell Splunk what to do with search results (formatting, statistics, create charts).
  - chart/timechart – Returns results in tabular output for charting;
  - rename – Rename a specific field;
  - sort – Sort results by a specific field;
  - stats – Provides statistics;
  - eval – Calculate an expression;
  - dedup – Removes duplicates; and
  - table – Builds a table with specific fields.

- **Functions** – Explain how we want charts, compute, and results evaluated. These can be count, distinct count, sum, average, min, max, list, and values.
- **Arguments** – The variable that we want to apply to the function.
- **Clauses** – Grouped or defined with by and as.

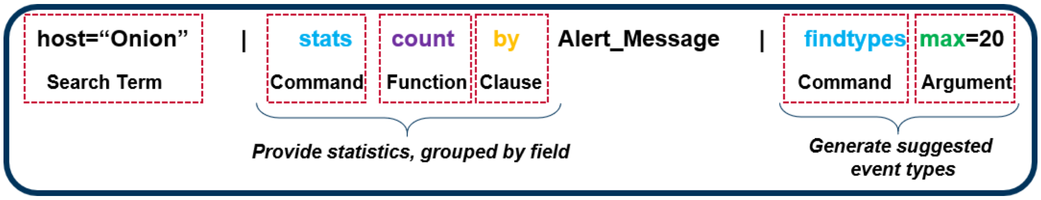
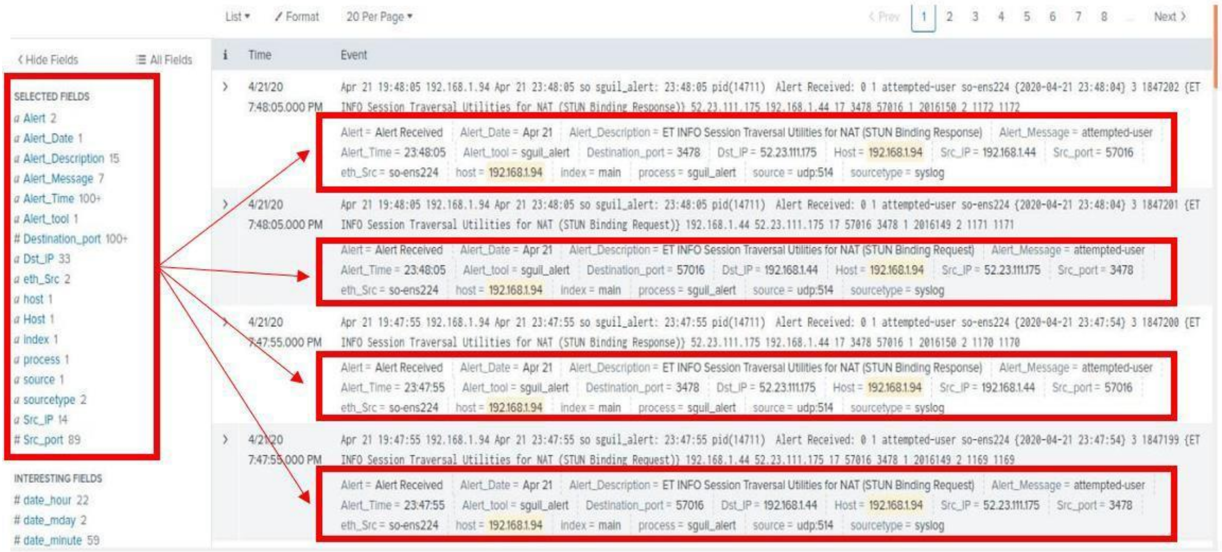


Figure 8. Splunk Search Language

Searching NIDS Alerts

As a Tier 1 analyst, you will receive the forwarded Security Onion or other NIDS tool alerts from the administrator or the power user. This will provide you with host (name, IP address, or domain). Type the host into the search bar under the Search & Reporting App. In the search results sidebar, there are selected fields that appear in the lower part of the events. A lowercase letter A (a) next to the field means the field contains a string value, while the hash (#) means the field value is numerical. Clicking on one of the fields will open a window and show the values, count, and percentage. From the newly opened window you can also launch a visualized or quick statistical report to include top values, top values by time, and rare values (see Figure 9).



SELECTED FIELDS
a Alert 2
a Alert_Date 1
a Alert_Description 15
a Alert_Message 7
a Alert_Time 100+
a Alert_tool 1
Destination_port 100+
a Dst_IP 33
a eth_Src 2
a host 1
a Host 1
a index 1
a process 1
a source 1
a sourcetype 2
a Src_IP 14
Src_port 89

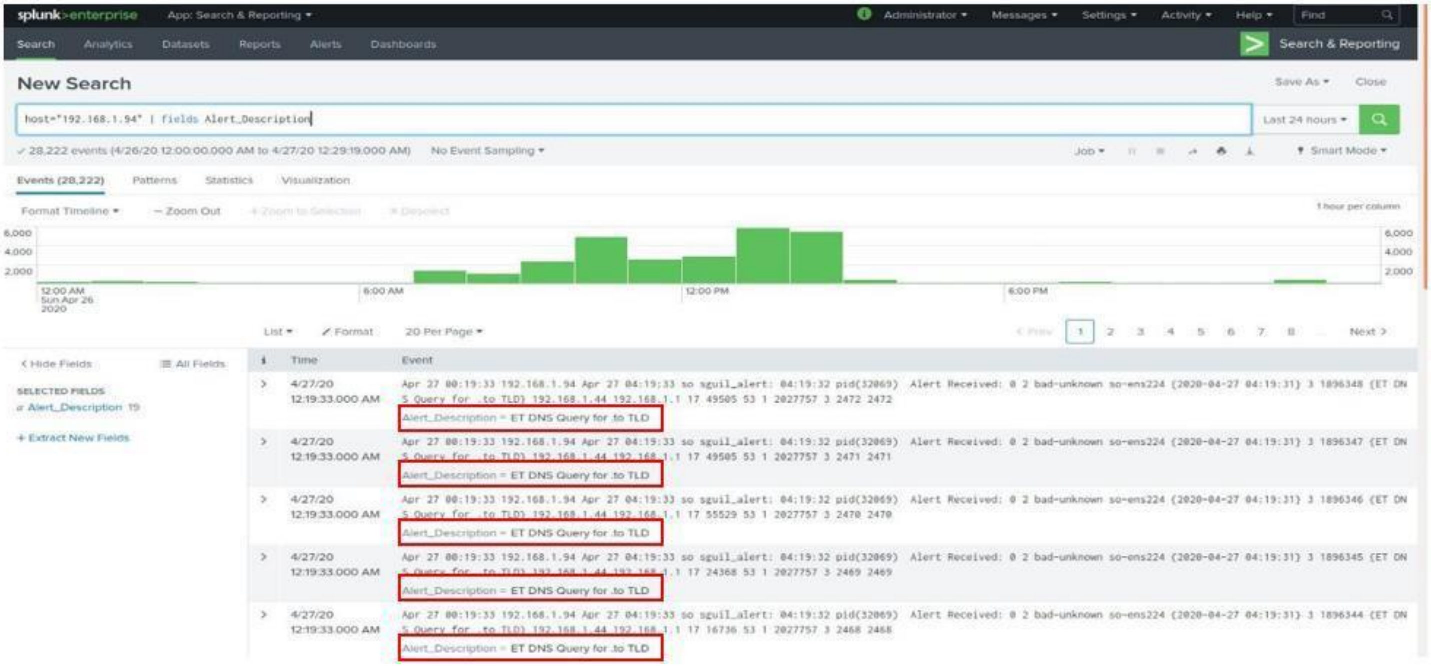
INTERESTING FIELDS
# date_hour 22
# date_mday 2
# date_minute 59

```
> 4/21/20              Apr 21 19:48:05 192.168.1.94 Apr 21 23:48:05 so sguil_alert: 23:48:05 pid(14711)  Alert Received: 0 1 attempted-user so-ens224 {2020-04-21 23:48:04} 3 1847202 {ET
  7:48:05.000 PM       INFO Session Traversal Utilities for NAT (STUN Binding Response)} 52.23.111.175 192.168.1.44 17 3478 57016 1 2016150 2 1172 1172
                       Alert = Alert Received   Alert_Date = Apr 21   Alert_Description = ET INFO Session Traversal Utilities for NAT (STUN Binding Response)   Alert_Message = attempted-user
                       Alert_Time = 23:48:05   Alert_tool = sguil_alert   Destination_port = 3478   Dst_IP = 52.23.111.175   Host = 192.168.1.94   Src_IP = 192.168.1.44   Src_port = 57016
                       eth_Src = so-ens224   host = 192.168.1.94   index = main   process = sguil_alert   source = udp:514   sourcetype = syslog
> 4/21/20              Apr 21 19:48:05 192.168.1.94 Apr 21 23:48:05 so sguil_alert: 23:48:05 pid(14711)  Alert Received: 0 1 attempted-user so-ens224 {2020-04-21 23:48:04} 3 1847201 {ET
  7:48:05.000 PM       INFO Session Traversal Utilities for NAT (STUN Binding Request)} 192.168.1.44 52.23.111.175 17 57016 3478 1 2016149 2 1171 1171
                       Alert = Alert Received   Alert_Date = Apr 21   Alert_Description = ET INFO Session Traversal Utilities for NAT (STUN Binding Request)   Alert_Message = attempted-user
                       Alert_Time = 23:48:05   Alert_tool = sguil_alert   Destination_port = 57016   Dst_IP = 192.168.1.44   Host = 192.168.1.94   Src_IP = 52.23.111.175   Src_port = 3478
                       eth_Src = so-ens224   host = 192.168.1.94   index = main   process = sguil_alert   source = udp:514   sourcetype = syslog
> 4/21/20              Apr 21 19:47:55 192.168.1.94 Apr 21 23:47:55 so sguil_alert: 23:47:55 pid(14711)  Alert Received: 0 1 attempted-user so-ens224 {2020-04-21 23:47:54} 3 1847200 {ET
  7:47:55.000 PM       INFO Session Traversal Utilities for NAT (STUN Binding Response)} 52.23.111.175 192.168.1.44 17 3478 57016 1 2016150 2 1170 1170
                       Alert = Alert Received   Alert_Date = Apr 21   Alert_Description = ET INFO Session Traversal Utilities for NAT (STUN Binding Response)   Alert_Message = attempted-user
                       Alert_Time = 23:47:55   Alert_tool = sguil_alert   Destination_port = 3478   Dst_IP = 52.23.111.175   Host = 192.168.1.94   Src_IP = 192.168.1.44   Src_port = 57016
                       eth_Src = so-ens224   host = 192.168.1.94   index = main   process = sguil_alert   source = udp:514   sourcetype = syslog
> 4/21/20              Apr 21 19:47:55 192.168.1.94 Apr 21 23:47:55 so sguil_alert: 23:47:55 pid(14711)  Alert Received: 0 1 attempted-user so-ens224 {2020-04-21 23:47:54} 3 1847199 {ET
  7:47:55.000 PM       INFO Session Traversal Utilities for NAT (STUN Binding Request)} 192.168.1.44 52.23.111.175 17 57016 3478 1 2016149 2 1169 1169
                       Alert = Alert Received   Alert_Date = Apr 21   Alert_Description = ET INFO Session Traversal Utilities for NAT (STUN Binding Request)   Alert_Message = attempted-user
                       Alert_Time = 23:47:55   Alert_tool = sguil_alert   Destination_port = 57016   Dst_IP = 192.168.1.44   Host = 192.168.1.94   Src_IP = 52.23.111.175   Src_port = 3478
                       eth_Src = so-ens224   host = 192.168.1.94   index = main   process = sguil_alert   source = udp:514   sourcetype = syslog
```

Figure 9. Search Output

If you want to search by field to include or exclude specific fields from search results, click on the value in the field window, or you can type a hostname, then the field name. In the following example a search is conducted for events only with a field value of Alert_Description. This can be accomplished by specifically stating the field name or using a wildcard and searching the term Alert_Description (see Figure 10). The search will return all fields with highlighted Alert_Description field values.

host="192.168.1.94" | fields Alert_Description or **host="192.168.1.66" Alert_Description="*"**



```
host="192.168.1.94" | fields Alert_Description
```

✓ 28,222 events (4/26/20 12:00:00.000 AM to 4/27/20 12:29:19.000 AM)

```
4/27/20 12:19:33.000 AM
Apr 27 00:19:33 192.168.1.94 Apr 27 04:19:33 so sguil_alert: 04:19:32 pid(32069)  Alert Received: 0 2 bad-unknown so-ens224 {2020-04-27 04:19:31} 3 1896348 {ET DNS Query for .to TLD} 192.168.1.44 192.168.1.1 17 49505 53 1 2027757 3 2472 2472
Alert_Description = ET DNS Query for .to TLD

4/27/20 12:19:33.000 AM
Apr 27 00:19:33 192.168.1.94 Apr 27 04:19:33 so sguil_alert: 04:19:32 pid(32069)  Alert Received: 0 2 bad-unknown so-ens224 {2020-04-27 04:19:31} 3 1896347 {ET DNS Query for .to TLD} 192.168.1.44 192.168.1.1 17 49505 53 1 2027757 3 2471 2471
Alert_Description = ET DNS Query for .to TLD

4/27/20 12:19:33.000 AM
Apr 27 00:19:33 192.168.1.94 Apr 27 04:19:33 so sguil_alert: 04:19:32 pid(32069)  Alert Received: 0 2 bad-unknown so-ens224 {2020-04-27 04:19:31} 3 1896346 {ET DNS Query for .to TLD} 192.168.1.44 192.168.1.1 17 55529 53 1 2027757 3 2470 2470
Alert_Description = ET DNS Query for .to TLD

4/27/20 12:19:33.000 AM
Apr 27 00:19:33 192.168.1.94 Apr 27 04:19:33 so sguil_alert: 04:19:32 pid(32069)  Alert Received: 0 2 bad-unknown so-ens224 {2020-04-27 04:19:31} 3 1896345 {ET DNS Query for .to TLD} 192.168.1.44 192.168.1.1 17 24368 53 1 2027757 3 2469 2469
Alert_Description = ET DNS Query for .to TLD

4/27/20 12:19:33.000 AM
Apr 27 00:19:33 192.168.1.94 Apr 27 04:19:33 so sguil_alert: 04:19:32 pid(32069)  Alert Received: 0 2 bad-unknown so-ens224 {2020-04-27 04:19:31} 3 1896344 {ET DNS Query for .to TLD} 192.168.1.44 192.168.1.1 17 16736 53 1 2027757 3 2468 2468
Alert_Description = ET DNS Query for .to TLD
```

Figure 10. Search for Alert_Description Field

If there is a specific source you want to limit your search to, add the source as a source type to the term. For example, we only want syslog events. Remember that field names are case sensitive, and values are not (see Figure 11).

host=“192.168.1.94” sourcetype=“syslog ERROR” | fields Alert_Description

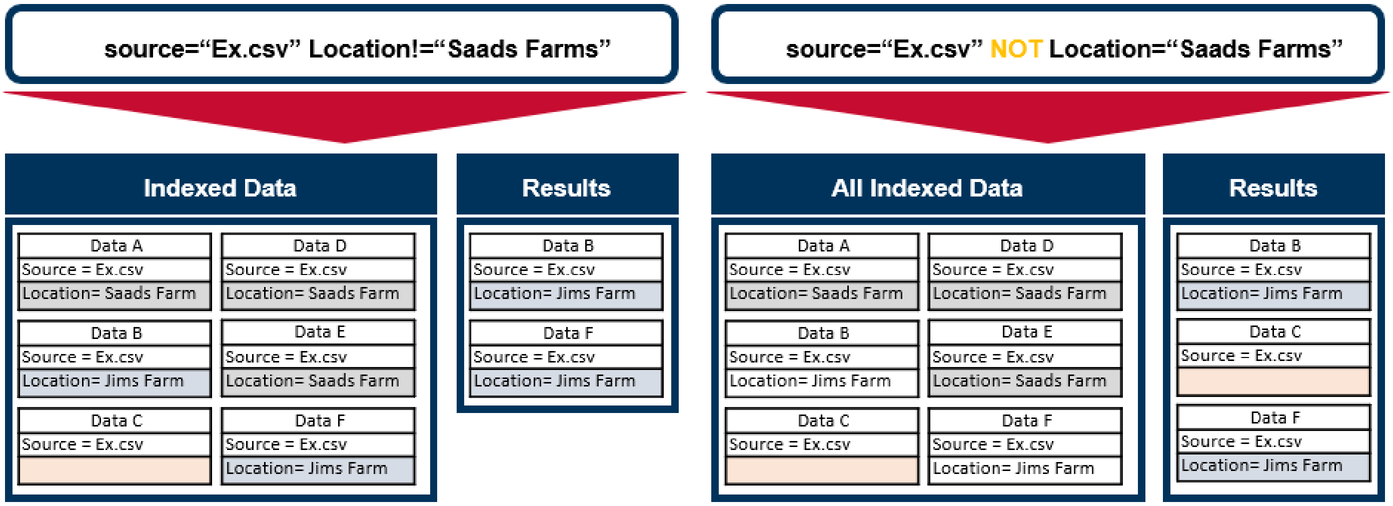
Figure 11. Additional Term

When searching try to use boolean and comparison operators to make the search range smaller and decrease search time. These operators also help refine searches after the initial query. In the following example a search is conducted on events from host 192.168.1.94, that have any alert description filed value, where the alert message field value is attempted-user, and the field value 3478 for the source port is excluded (see Figure 12).

```
host="192.168.1.94" Alert_Description="*" Alert_Message="attempted-user"
Src_por!=3478 Or
host="192.168.1.94" Alert_Description="*" Alert_Message="attempted-user" NOT
Src_por=3478
```

Figure 12. Comparison and Boolean

However, it is essential to understand that the is not operator (!=) and the boolean (NOT) do not return the same results every time. The difference is that with a comparison operator, such as !=, it's indicated that the field exists, but does not have the value specified. So, if the field is not found at all in the event, the search will not match, and other events will not return. On the other hand, using a boolean, such as NOT, will check if the field has the specified value, and if it doesn't, it will match. In the following example (see Figure 13) both operators are used with different results. When the comparison operator is used, the result are events that do not have the search field value. However, events that do not have a Location field value are also ignored. This results in two events returned. The boolean search returns events that do not have the search field value in Location, resulting in three events displayed [19].



source="Ex.csv" Location!="Saads Farms"

Indexed Data

Data A
Source = Ex.csv
Location= Saads Farm

Data D
Source = Ex.csv
Location= Saads Farm

Data B
Source = Ex.csv
Location= Jims Farm

Data E
Source = Ex.csv
Location= Saads Farm

Data C
Source = Ex.csv

Data F
Source = Ex.csv
Location= Jims Farm

Results

Data B
Source = Ex.csv
Location= Jims Farm

Data F
Source = Ex.csv
Location= Jims Farm

source="Ex.csv" NOT Location="Saads Farms"

All Indexed Data

Data A
Source = Ex.csv
Location= Saads Farm

Data D
Source = Ex.csv
Location= Saads Farm

Data B
Source = Ex.csv
Location= Jims Farm

Data E
Source = Ex.csv
Location= Saads Farm

Data C
Source = Ex.csv

Data F
Source = Ex.csv
Location= Jims Farm

Results

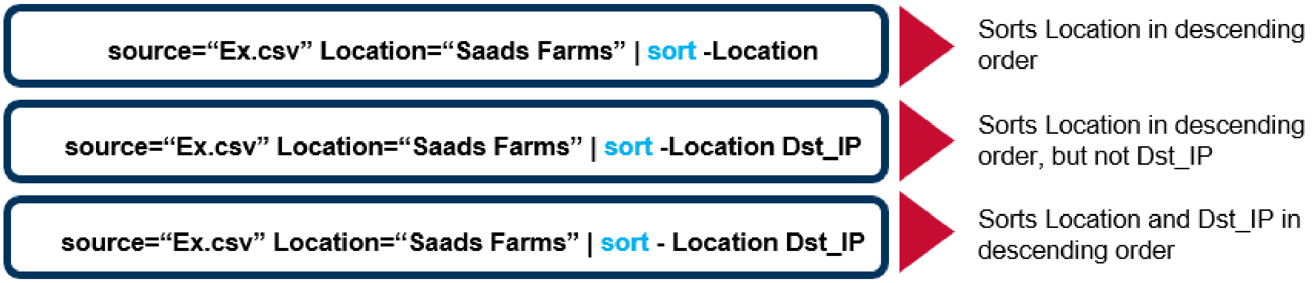
Data B
Source = Ex.csv
Location= Jims Farm

Data C
Source = Ex.csv

Data F
Source = Ex.csv
Location= Jims Farm

Figure 13. Comparison vs. Boolean Search

One of the more useful commands is the sort command. With this command you can display the results in ascending or descending order. The plus sign is used for ascending and the minus sign for descending. The signs are added in front of the field or column to be sorted. Of note, when specifying more than one field, a space is left between the sign and field, else use the command with no space to specifically sort one column or field (see Figure 14.)



source="Ex.csv" Location="Saads Farms" | sort -Location

Sorts Location in descending order

source="Ex.csv" Location="Saads Farms" | sort -Location Dst_IP

Sorts Location in descending order, but not Dst_IP

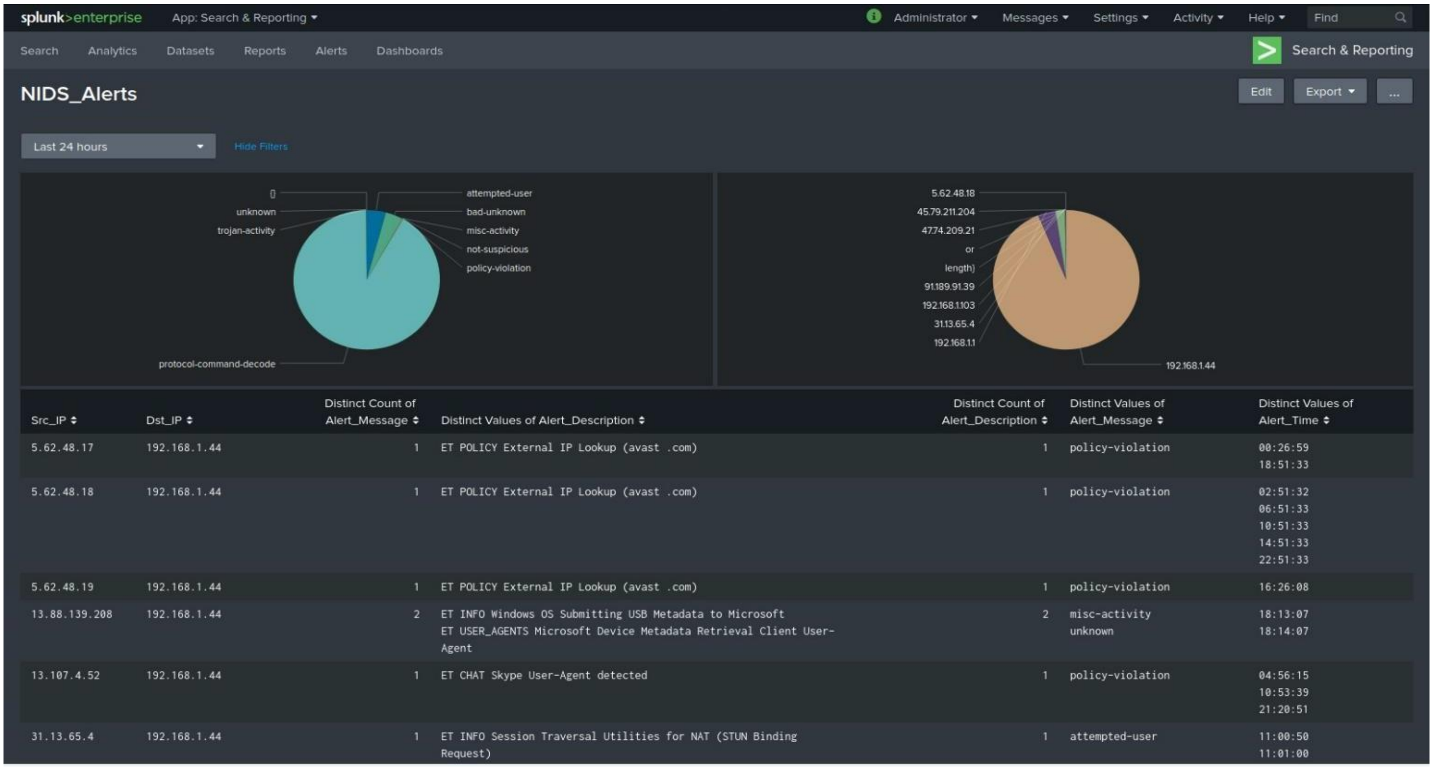
source="Ex.csv" Location="Saads Farms" | sort - Location Dst_IP

Sorts Location and Dst_IP in descending order

Figure 14. Using Sort Command

There are many other commands that can be used. To see a full list and detailed explanation go to the following link: https://docs.splunk.com/Documentation/Splunk/8.0.3/SearchReference/ListOfSearchCommands

Another great tool for visualizing the search results is the option to display them in a dashboard. You can do this by clicking on the dashboard button and choosing NIDS Alerts (see Figure 15).



Src_IP ⬍	Dst_IP ⬍	Distinct Count of Alert_Message ⬍	Distinct Values of Alert_Description ⬍	Distinct Count of Alert_Description ⬍	Distinct Values of Alert_Message ⬍	Distinct Values of Alert_Time ⬍
5.62.48.17	192.168.1.44	1	ET POLICY External IP Lookup (avast .com)	1	policy-violation	00:26:59 18:51:33
5.62.48.18	192.168.1.44	1	ET POLICY External IP Lookup (avast .com)	1	policy-violation	02:51:32 06:51:33 10:51:33 14:51:33 22:51:33
5.62.48.19	192.168.1.44	1	ET POLICY External IP Lookup (avast .com)	1	policy-violation	16:26:08
13.88.139.208	192.168.1.44	2	ET INFO Windows OS Submitting USB Metadata to Microsoft ET USER_AGENTS Microsoft Device Metadata Retrieval Client User-Agent	2	misc-activity unknown	18:13:07 18:14:07
13.107.4.52	192.168.1.44	1	ET CHAT Skype User-Agent detected	1	policy-violation	04:56:15 10:53:39 21:20:51
31.13.65.4	192.168.1.44	1	ET INFO Session Traversal Utilities for NAT (STUN Binding Request)	1	attempted-user	11:00:50 11:01:00

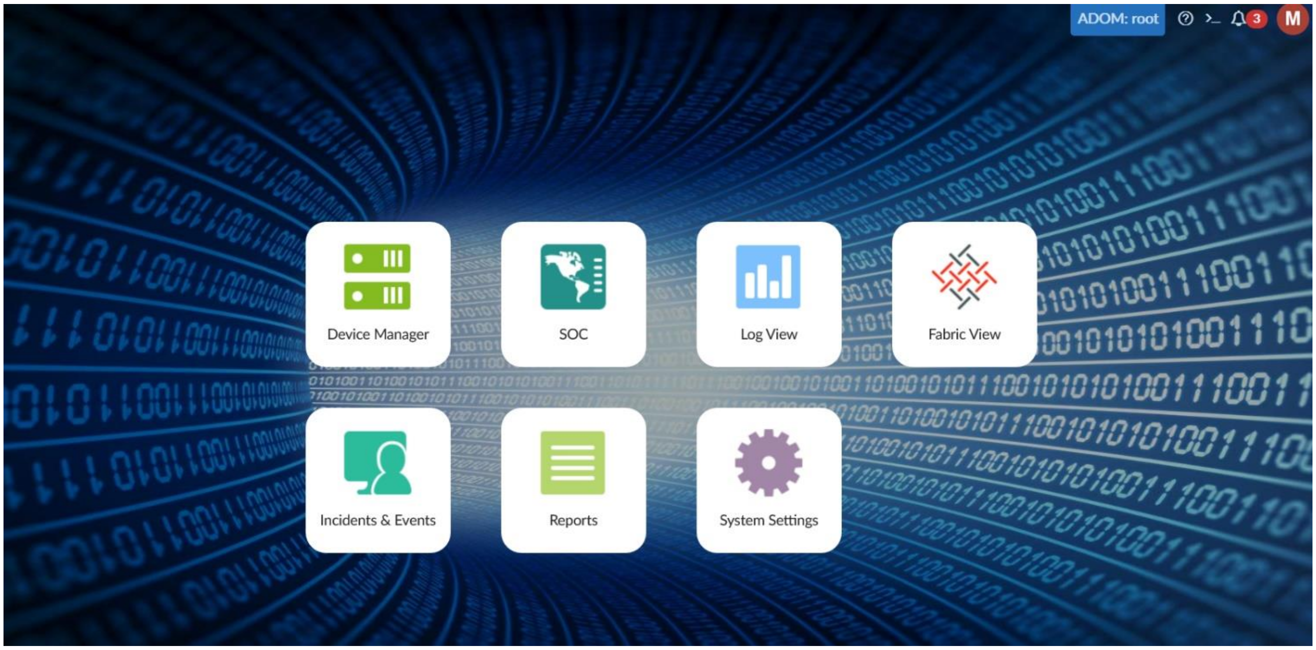
Figure 15. NIDS Alerts Dashboard

Fortianalyzer

FortiAnalyzer. Security-Driven Analytics and Log Management. FortiAnalyzer provides deep insights into advanced threats through Single-Pane Orchestration, Automation, and Response for your entire attack surface to reduce risks and improve your organization's overall security.

Fortianalyzer Documentation

After connecting to the Fortianalyzer, select Fortigate VDOM
You will have read-only access to LogView, Fortiview, SOC, Incidents, and Reports.



References

[1] Splunkbase, "Security Onion App for Splunk software," [Online]. Available: https://splunkbase.splunk.com/app/972/.

[2] Iknow, "Splunk," [Online]. Available: https://www.iknow.us/about-iknow/splunk.

[3] Wikipedia, "DDoS mitigation," [Online]. Available: https://en.wikipedia.org/wiki/DDoS_mitigation.

[4] University System of Georgia , "Peachnet," [Online]. Available: https://www.usg.edu/peachnet/.

[5] Wikipedia, "OSI model," [Online]. Available: https://en.wikipedia.org/wiki/OSI_model#:~:text=The%20Open%20Systems%20Interconnection%20model,underlyin g%20internal%20structure%20and%20technology..

[6] Forbes, "F5 Networks," [Online]. Available: https://www.forbes.com/companies/f5-networks/#4324d9ef602c.

[7] C. Abbott, "What Is BIG-IP?," Dev/Central, [Online]. Available: https://devcentral.f5.com/s/articles/what-is-big-ip-24596.

[8] Snort, "Snort FAQ," [Online]. Available: https://www.snort.org/faq/what-is-snort.

[9] Suricata, "Suricata," [Online]. Available: https://suricata-ids.org/.

[10] Zeek, [Online]. Available: https://zeek.org/.

[11] OSSEC, [Online]. Available: https://www.ossec.net/.

[12] netsniff-ng, "netsniff-ng toolkit," [Online]. Available: http://netsniff-ng.org/.

[13] Infosec. URL: https://resources.infosecinstitute.com/peeling-the-onion-security-onion-os/#gref

[14] Splunk. URL: https://www.splunk.com/blog/2019/06/12/reimagine-security-operations-usingsplunk.html

[15] https://docs.splunk.com/Documentation/Splunk/8.0.2/Indexer/Bucketsandclusters

[16] https://docs.splunk.com/Documentation/Splunk/8.0.3/Search/Aboutsearchlanguagesyntax

[17] https://docs.splunk.com/Documentation/SplunkCloud/8.0.2001/Search/Wildcards

[18] https://intellipaat.com/blog/tutorial/splunk-tutorial/searching-with-splunk/

[19] https://docs.splunk.com/Documentation/Splunk/8.0.3/Search/NOTexpression

s securityonion. URL: https://securityonion.readthedocs.io/en/latest/index.html

Virustotal. URL.https://www.virustotal

Splunk. URL: https://www.splunk.com/blog/2019/06/12/reimagine-security-operations-usingsplunk.html

Infosec. URL: https://resources.infosecinstitute.com/peeling-the-onion-security-onion-os/#gref

Quora. URL: https://www.quora.com/What-are-IDS-IPS-DLP-SIEM-NBAD-systems

Microsoft. URL: https://www.microsoft.com/security/blog/2019/07/02/microsofts-threat-vulnerabilitymanagement-now-helps-thousands-of-customers-to-discover-prioritize-and-remediate-vulnerabilities-inreal-time/

Networkstraining. URL: https://www.networkstraining.com/compare-and-contrast-networktopologies/