ORNL/TM-2022/2463

Using Splunk® Enterprise Search Commands for Advanced Analysis of Ivanti Connect Secure© Logs

B. Nance

April 2022

Approved for public release.
Distribution is unlimited.

OAK RIDGE National Laboratory

ORNL IS MANAGED BY UT-BATTELLE LLC FOR THE US DEPARTMENT OF ENERGY

ORNL/TM-2022/2463

Computer Science and Mathematics Division
Performance Engineering Group

Using Splunk® Enterprise Search Commands for Advanced Analysis of Ivanti Connect Secure© Logs

Brad Nance

April 2022

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, TN 37831
managed by
UT-BATTELLE LLC
for the
US DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

CA	Certificate Authority
FQDN	Fully Qualified Domain Name
HTML	HyperText Markup Language
ICS©	Ivanti Connect Secure©
IP	Internet Protocol
IT	Information Technology
LDAP	Lightweight Directory Access Protocol
SSL	Secure Socket Layer
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
URL	Uniform Resource Locator
VPN	Virtual Private Network

ABSTRACT

Analyzing the logs of even the smallest Information Technology (IT) system can be a challenge considering they can generate millions of lines of log data in a very short time. Splunk® Enterprise is an industry leading tool that allows analysis of log data, which can enhance troubleshooting capabilities, improve system performance, and improve the security posture of an IT system.

Ivanti Connect Secure© (ICS) is a market-leading platform powered by the Ivanti Secure Socket Layer Virtual Private Network (SSL VPN) appliance, providing an architecture for secure access to and protection of network resources.

This paper describes an approach for using Splunk Enterprise search capabilities to perform advanced data analysis of ICS logs.

1. INTRODUCTION

Analyzing the logs of even the smallest Information Technology (IT) system can be a challenge, considering that they can generate millions of lines of log data in a very short time. Splunk® Enterprise is an industry leading tool that allows analysis of log data, which can enhance troubleshooting capability, improve system performance, and improve the security posture of an IT system.

Ivanti Connect Secure© (ICS) is a market-leading platform powered by the Ivanti Secure Socket Layer Virtual Private Network (SSL VPN) appliance, providing an architecture for secure access to and protection of network resources.

This paper describes an approach for using Splunk Enterprise search capabilities to perform advanced data analysis of ICS logs. The advanced data analysis includes Splunk search pipelines that provide the following information:

- User session duration
- Concurrent active user sessions
- Application page load duration

2. SPLUNK ENTERPRISE

Splunk Enterprise is a platform for collecting, indexing, and analyzing enterprise data. Splunk can collect data from a host either through the Splunk Universal Forwarder or through the Syslog Configuration, as shown in Figure 1.

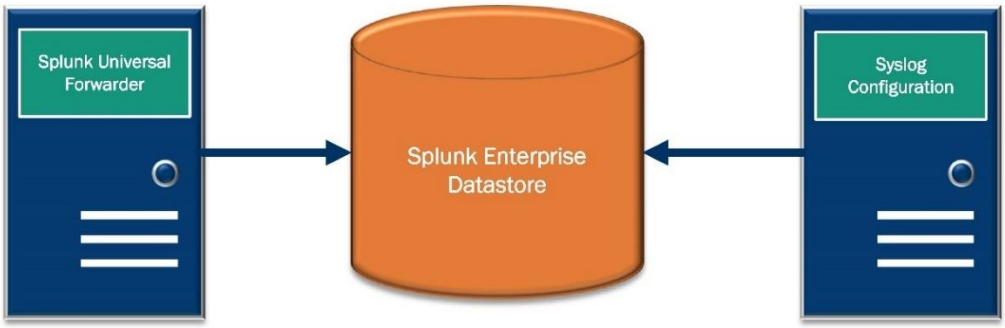

Figure 1. Splunk dataflow diagram.

- The **Splunk Universal Forwarder** is a special version of Splunk Enterprise that runs as a process or service on the host and sends data to a Splunk Enterprise Server. It is the most common means of sending data to a Splunk server.
- **Syslog** is an industry standard for sending and receiving data and was developed as part of the Sendmail project in the 1980s. Splunk Enterprise can serve as a Syslog server, allowing a device or service to send data directly to the Splunk server datastore using User Datagram Protocol (UDP) or Transmission Control Protocol (TCP) on port 514. For the advanced data analysis presented in Section 5, the Syslog method is used to send data from ICS to the Splunk server.

- Once received by the Splunk server and added to the **Splunk Enterprise Datastore**, the log data is indexed so that it can be analyzed. For the advanced data analysis, regular expressions are used for field extraction, which is the process for giving structure to the unstructured logs sent to Splunk. Splunk searches are run against this indexed, semi-structured data.

3. IVANTI CONNECT SECURE

ICS (formerly Pulse Connect Secure) is a platform that runs on the Ivanti SSL VPN appliance and provides an architecture for secure access to and protection of network resources. Its ability to customize the format of log output, along with the ability to send the logs to a Syslog server, are important features for the advanced analysis capabilities presented.

3.1 ICS ARCHITECTURE

The ICS architecture provides a layered system of access controls, ensuring that internal resources are protected from unauthorized access. Figure 2 shows the primary ICS architectural components, which are discussed in this section.

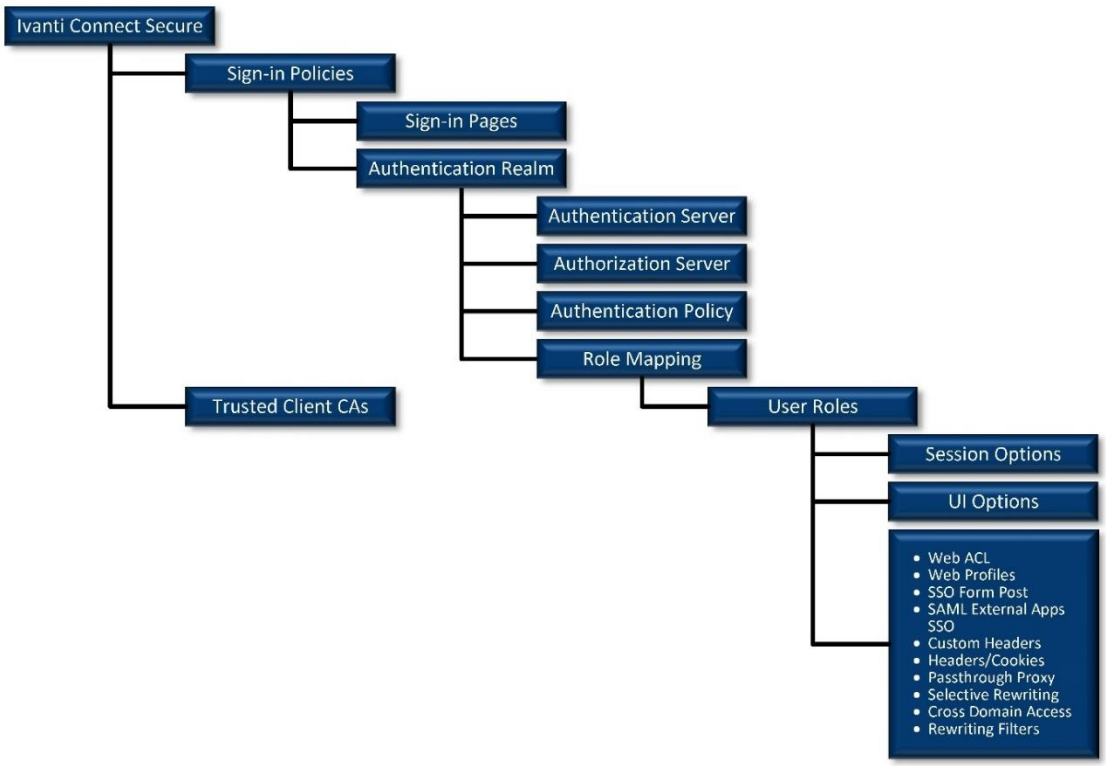

Figure 2. ICS architecture.

The ICS architectural components are defined as follows:

- **Sign-in Policies –** The entry point, i.e., the Uniform Resource Locator (URL), used to access an ICS web presence.
- **Sign-in Pages –** The underlying, customizable HyperText Markup Language (HTML) used to render the user interface.
- **Authentication Realm –** A set of configured servers and policies that define the behavior of the secure portal.
- **Authentication Server –** The part of the interface that is responsible for the first challenge to the end user to validate their identity.
- **Authorization Server –** The part of a configuration that provides the identity of the authorized user to the underlying applications.
- **Authentication Policy –** The part of a configuration that limits access to a secure portal based upon parameters such as source Internet Protocol (IP) address, client-side certificate, or browser user agent.
- **Role Mapping –** The process of assigning user roles to an authorized user. The role mapping rules are contained within a user realm configuration.
- **User Roles –** Categorizations of authorized users that are used to determine session options, user interface options, and the application of defined resource policies. These are based upon account attributes, including username and Lightweight Directory Access Protocol (LDAP) group membership.
- **Trusted Client Certificate Authorities (CAs) –** Intermediate and root CAs that define client-side certificates that can be used to authenticate to a certificate-based authentication server.

3.2 CUSTOM LOG FILTERS

ICS utilizes custom filters for formatting log output. This feature is important as it makes it easier to extract fields from the raw log data once it is sent to the Splunk server datastore. Figure 3 shows an example of a custom log filter that is used to format the logs that are generated on the ICS appliance.

```
%date% %time% - %node% - sourceip=[%sourceip%] user=[%user%] user_realm=[%realm%] role=[%role%] -
%msg% - id=%id% - severity=%severity% - useragent=[%userAgent%]
```

Figure 3. Example of a custom log filter.

3.3 SYSLOG SETTINGS

ICS also can send log data directly to a Syslog server. Figure 4 shows a configuration that sends log data to a Splunk server, using a custom filter and the UDP transmission protocol.

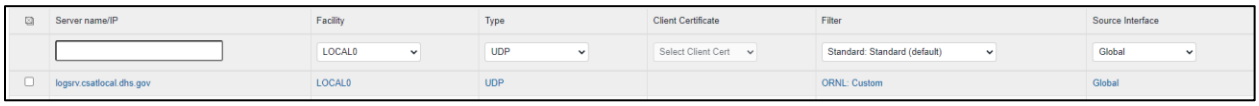

Figure 4. ICS Syslog settings.

The parameters of the Syslog server configuration are as follows:

- **Server Name/IP –** The Fully Qualified Domain Name (FQDN) or IP address of the Syslog (i.e., Splunk) server.
- **Facility –** The name of the output log file on the Syslog server.
- **Type –** The transmission protocol used to send data to the Syslog server.
- **Client Certificate –** The SSL certificate used to encrypt the data sent to the Syslog server. If no client certificate is specified, the log data is sent unencrypted to the Syslog server.
- **Filter –** The custom log filter used to format the data sent to the Syslog server.
- **Source Interface –** The network interface through which the filtered log data is sent.

4. SPLUNK SEARCH

Splunk Enterprise provides a web interface that can be used to search and analyze the data that is collected. The search results are returned in an unmodified (i.e., raw) format. Splunk commands, along with field extractions, can be used to transform the raw data into charts, graphs, and tables. The searches also can be saved as reports, alerts, and dashboards for later use. Figure 5 is an example of a Splunk dashboard that shows the number of daily unauthenticated requests to ICS during the course of an entire month. The green horizontal line indicates the daily average during the month.

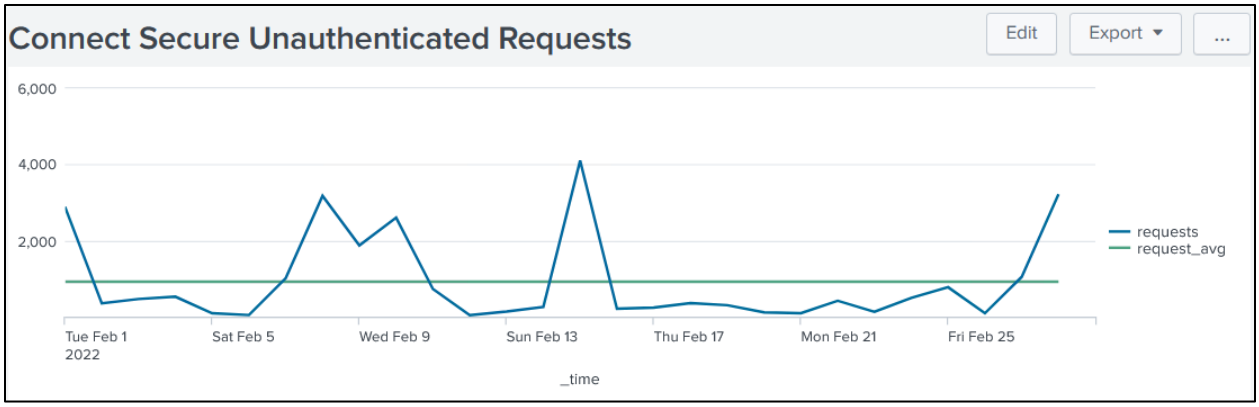

Figure 5. Example of a Splunk dashboard.

Sections 4.1 through 4.6 provide more detail on the following aspects of the Splunk search capability:

- Search pipeline
- Custom field extractions
- Search operators
- Search commands used for the advanced data analysis
- Other relevant search commands
- Basic search examples

4.1 SEARCH PIPELINE

A Splunk search pipeline is a search that consists of several individual search commands that are connected using the "|" (pipe) character. For example, consider the search pipeline contained in Figure 6, where A, B, and C represent individual search commands.

```
A | B | C
```

Figure 6. Example of a search pipeline.

The search pipeline is evaluated left to right. The search command C is applied against the results of B after B has been applied to the results of A. Search commands can add, remove, and transform data. More detail regarding the effects of a particular search command, including individual search commands and the search pipelines used for the advanced data analysis, are discussed in detail in Section 4.4.

4.2 FIELD EXTRACTIONS

As data is sent to the Splunk server datastore, and then indexed, it is preprocessed to extract field-level information from the raw log data. For example, if a log entry contains user=[john.doe], then the Splunk indexer will automatically create a field called user and generate metadata to extract the same information from similar log entries. In some instances, the data analyst might need to modify the default field extraction regular expression or add new ones. Figure 7 shows the regular expression used to extract application_id, which is required for advanced data analysis presented in Section 4.4.

```
^[^\?\n]*f\?p=(?P<application_id>[^:\s\%]+)
```

Figure 7. Regular expression to extract *application_id*.

4.3 RELEVANT SEARCH OPERATORS

The following search operators are relevant to the advanced data analysis:

- AND
- OR
- NOT

In a search command, these operators are case sensitive. Figure 8 shows an example of a search command that includes operators. Parentheses are used to control the order of evaluation.

```
"Unauthenticated request" NOT (ua_url="/dana*" OR ua_url="/")
```

Figure 8. Example of a search operator.

4.4 SEARCH COMMANDS FOR ADVANCED DATA ANALYSIS

The search commands relevant for completing advanced data analysis are detailed in this section.

4.4.1 Transaction

The transaction command is used to group raw log entries into events, based upon certain common content. The transaction command adds the fields duration and eventcount to the search results. In one of the advanced data analysis scenarios, this command is used to group logged entries into a single application page load event. The beginning of the event is found using the startswith option. The end of the event is found using the endswith option. Figure 9 shows the transaction command used to group log entries into application page load events. This particular search command is a building block for determining the duration of an application page load event.

```
transaction maxspan=5m user, application_id, application_page_id startswith=("Request: GET /apex/f?p" AND
apex_session_id="*") endswith=("WebRequest completed" OR result=404)
```

Figure 9. Transaction command for application page loads.

4.4.2 Concurrency

The concurrency command is used to determine the number of events occurring at the same time an event started including itself (the concurrency value). Figure 10 shows the concurrency command used to determine overlaps with the duration window returned by the transaction command.

```
concurrency duration=duration
```

Figure 10. Concurrency command for application page loads.

Figure 11 is an illustration of the concept of concurrency. The blue columns represent events on a vertical time scale. The number just to the right of each column indicates the *concurrency value*. The value for concurrency is determined at the beginning of each event.

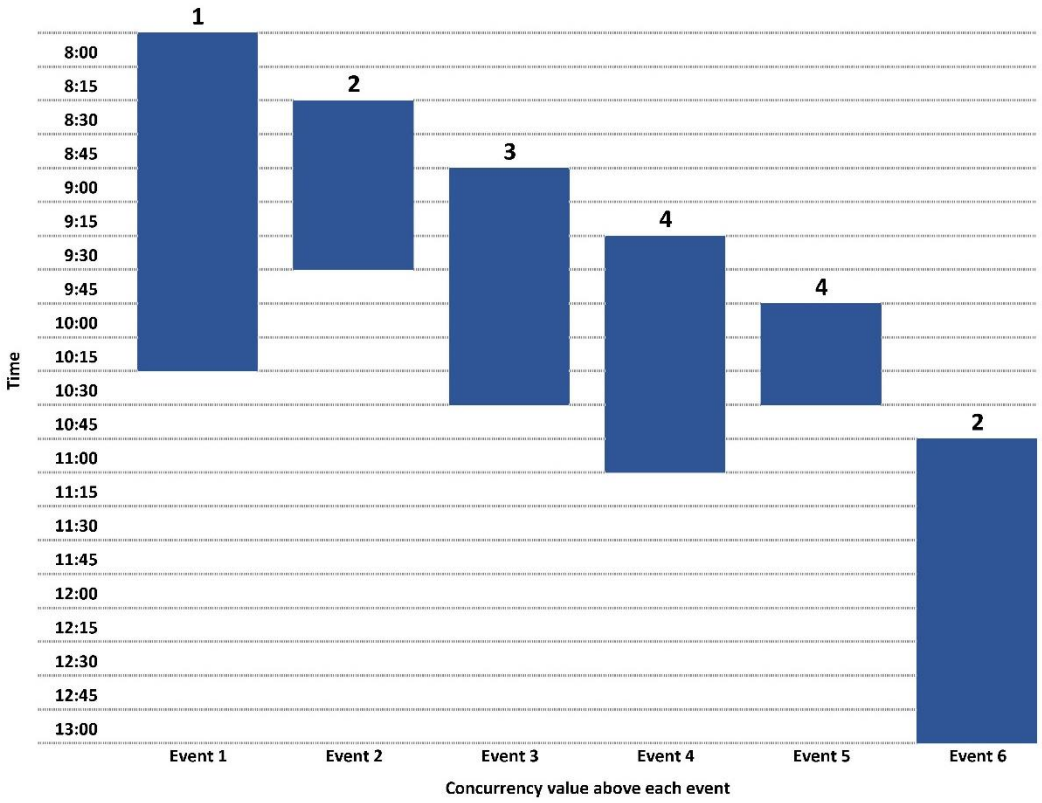

Figure 11. Illustration of concurrency.

4.4.3 Eval

The eval command performs a calculation or data transformation. In the advanced data analysis, it is used along with the tostring() function to convert the duration value from seconds to HH:MM:SS format. Figure 12 shows the command to perform the data transformation of the duration field.

```
eval duration=tostring(duration,"duration")
```

Figure 12. Eval command for converting duration seconds to HH:MM:SS.

4.4.4 Table

The table command returns a table with the specified columns. Figure 13 shows the command used to generate a table that will be used in the advanced data analysis.

```
table duration, user, user_realm, connect_secure_session, _time
```

Figure 13. Table command example.

4.4.5 Sort

The sort command sorts the data at that point in the pipeline based upon the specified fields in the specified direction (i.e., either ascending or descending). Figure 14 shows the command used to sort the results in descending order, based upon time.

```
sort _time desc
```

Figure 14. Sort command example.

4.4.6 Search

The search command is used to apply a search command to the results at that point in the pipeline. The command used to remove records from the pipeline that have a user_realm value, starting with "session" is shown in Figure 15.

```
search NOT user_realm="session*"
```

Figure 15. Search command example.

4.5 OTHER RELEVANT SEARCH COMMANDS

Other search commands used in the advanced search include the following:

- **Stats –** The stats command calculates aggregate statistics for a result set.
- **Timechart –** The timechart command returns a statistical aggregation, with time as the X-axis, that can be used to produce a chart.
- **Top –** The top command finds the most common values for the fields in a list.
- **Head –** The head command returns the first N number of search results, based upon the sorted order.

4.6 BASIC SEARCH EXAMPLES

Table 1 contains some basic search examples.

Table 1. Basic Search Command Examples

Search Command	Description
sourcetype=syslog	Returns all entries where the sourcetype is syslog.
error	Returns all entries containing the word “error.”
host="webserver*" status=500	Returns all entries on host system named “webserver” that resulted from a 500 error.
"Request: GET /survey*" AND "success.jsp*"| top limit=200 user	Returns a list of the top 200 users by most surveys submitted.
"ORA-00028: your session has been killed"	Returns Oracle Database sessions that have been killed.
"../../../data*"	Returns attempted exploitations of an ICS vulnerability.
"Unauthenticated request" NOT (ua_url="/dana*" OR ua_url="/") user="System" | top limit=20 ua_url	Returns the URL of the top 20 unauthenticated requests to ICS.

5. ADVANCED DATA ANALYSIS

With a basic understanding of Splunk Enterprise and ICS, the advanced analysis of the ICS log data can readily be performed. Sections 5.1 through 5.3 detail search pipelines that can be used to complete the following advanced data analysis scenarios:

- **User Session Duration –** The length of an ICS user session from time of logon until the session ends by either logout, idle timeout, or maximum session length timeout.
- **Concurrent Active User Sessions –** The number of active user sessions each time a new user session is created.
- **Application Page Load Duration –** The amount of time it takes for an application page to load.

5.1 USER SESSION DURATION

Table 2 shows the pipeline components for a search that determines the duration of an ICS user session.

Table 2. Splunk Search Pipeline for User Session Duration

Splunk Search Command	Explanation
sourcetype=syslog	Returns all ICS logs. This search assumes that ICS is the only device sending logs to Splunk using Syslog.
transaction maxpause=60m user, user_realm startswith="login succeeded"	Groups log entries into events based upon the user and user realm. A log entry that contains the phrase “login succeeded” is the indicator for the beginning (startswith) of an event. The maxpause of “60m” assumes an ICS idle session timeout of 1 hour. No endswith setting is required. The command assumes existing field extractions for user and user_realm
eval duration=tostring(duration,"duration")	The duration (added by the transaction command) is converted from seconds to an HH:MM:SS format.
table duration, user, user_realm, connect_secure_session, _time	The results are presented as a table containing columns for user, user_realm, connect_secure_session, and _time. The command assumes the additional field extraction for connect_secure_session. The “time” for the transaction is the most recent activity for the event.
search NOT user_realm="session*"	The user_realm field has a value that starts with the string “session” when a user logs out. These results are removed from the pipeline results at that point in the pipeline.
sort _time desc	Results are sorted by _time in descending order.

Table 3 shows an example of the results of this search pipeline.

Table 3. Example User Session Duration Search Results

sequence	duration (HH:MM:SS)	user	user_realm	connect_secure_session	_time
1	00:00:06	User1	User Realm 1	sid1884d450d3175a7d9c6cb6cc5cbf2b134748451162dd158e	2022-03-14 08:39:53
2	00:00:04	User2	User Realm 1	sid5f5ae27fcba3c38c86fa4c308920161dc6673c5c541136d0	2022-03-14 08:39:44
3	00:00:03	User3	User Realm 1	sid389fbee3145a2b6830dec118a325d8ae55d8bff6474f7799	2022-03-14 08:39:39
4	00:00:02	User4	User Realm 1	sid9564ce2db093c4fb2755836943b1f9480ee37d66c0ac9f86	2022-03-14 08:39:31
5	00:00:04	User5	User Realm 1	sid4c7ab6d45bc3c96f8e5b9cd023a024524da65f50dca46ee9	2022-03-14 08:39:04
6	00:00:02	User6	User Realm 1	sid21c7ab2b899d70c57ef6399fde808b0818982ba385e6bfb1	2022-03-14 08:39:04
7	00:00:02	User7	User Realm 1	sida816b32b8e82eadbe3b7020578b8acb5cba38204268ea6e8	2022-03-14 08:38:56
8	00:01:27	User8	User Realm 2	sid9bbb1e938bb9aab8c938975628c7cfc229d422d737381853	2022-03-14 08:37:36
9	00:00:34	User9	User Realm 3	sidc24ecea51efce3ce682c5e660ec61f39ffda1ec2ec2f7e55	2022-03-14 08:37:32
10	00:01:41	User10	User Realm 4	sidf1cca001901aaecfe9325ba4088bbbeec0eeeb101a753a42	2022-03-14 08:37:26

5.2 CONCURRENT ACTIVE USER SESSIONS

With a slight modification of the User Session Duration pipeline, a graph showing the number of concurrent user sessions can be generated. Table 4 shows the added commands.

Table 4. Splunk Search Pipeline for Concurrent Active User Sessions

Splunk Search Command	Explanation
concurrency duration=duration	Added in the pipeline between the transaction and eval commands in the User Session Duration pipeline.
stats max(concurrency) as concurrent_sessions by _time	Added to the end of the User Session Duration pipeline. The aggregate function max can be used since there is only one result per user and user_realm.

Figure 16 shows an example of a graph created using the search pipeline results.

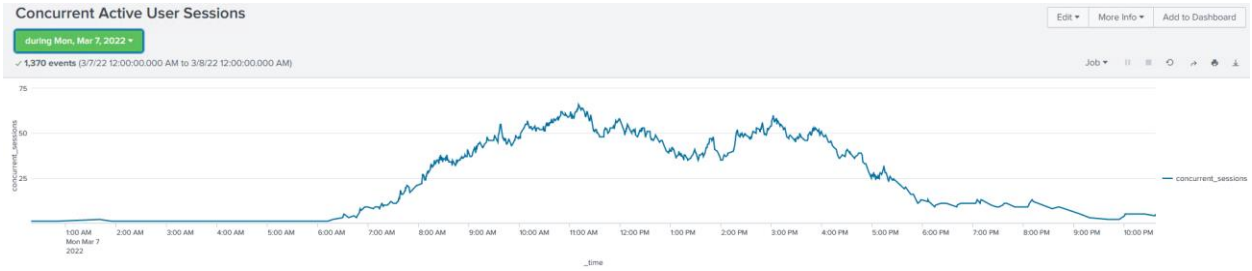

Figure 16. Example graph for concurrent active user sessions.

5.3 APPLICATION PAGE LOAD DURATION

Table 5 shows the pipeline components for a search that determines the duration of application page loads.

Table 5. Splunk Search Pipeline for Application Page Load Duration

Splunk Search Command	Explanation
user_realm="Application Portal" (application_id="*" OR application_page_id="*" OR result=404) NOT user_realm="session*"	Returns all ICS logs for the Application Portal user realm that have either an application_id or an application_page_id or that returns a 404 (page not found) error. The command assumes existing field extractions for user_realm, application_id, and application_page_id.
transaction maxspan=5m user, application_id, application_page_id startswith=("Request: GET /apex/f?p" AND appplication_session_id="*") endswith=("WebRequest completed" OR result=404)	Groups log entries into events based upon the user, application_id, and application_page_id. The indicator for the beginning (startswith) of an event is a GET to an application page that includes an application_session_id. The end of an event is indicated by either the completion of the web request or a 404 error. The maximum length of an event is five minutes. The command assumes an existing field extraction for user.
eval duration=tostring(duration,"duration")	The duration (added by the transaction command) is converted from seconds to an HH:MM:SS format.
table duration, user, application_id, application_page_id	The results are presented as a table containing columns for duration, user, application_id, and application_page_id.
sort duration desc	Results are sorted by duration in descending order.
head 10	Returns the first 10 results from the results that are sorted by duration in descending order.

Figure 16 shows an example of the results of this search pipeline.

sequence	duration (HH:MM:SS)	user	application_id	application_page_id
1	0:01:10	user1	522	2
2	0:00:39	user1	522	2
3	0:00:16	user1	14723	1
4	0:00:13	user2	522	2
5	0:00:13	user1	522	2
6	0:00:10	user3	13002	500
7	0:00:09	user4	13004	602
8	0:00:09	user5	13002	500
9	0:00:08	user6	14723	50
10	0:00:08	user6	14723	50

Figure 17. Example application page load duration search results.

6. CONCLUSION

Splunk Enterprise is a powerful tool for analyzing log data. With a little effort to understand its functions and capabilities, Splunk Enterprise can be used to perform advanced data analysis of ICS log data, resulting in enhanced troubleshooting capabilities, improved performance, and an improved security posture for an IT system.