Splunk Tool

The Splunk Ecosystem : SIEM and SOAR OVERVIEW

SIEM Capabilities:

1.Centralized security monitoring

2.Data analytics

3.Data correlation

4.Threat detection

5.Dashboard,alerts,reports

SIEM:Detect Malicious Activity

-Centralized data ingestion from a variety of sources

-Data analysis and correlation for threat detection

-User behavior analytics to detect anomalies e.g users signing in from unknown location.

Splunk SOAR (Security Orchestration,Automation and Response)

SOAR Capabilities:

1.Automated incident Response

2.Playbook workflows

3.Add on integrating

SOAR: Stop Malicious Activity

-Get plug-ins,applications, and work flows templates from splunkbase

-Easy procedural consistency through automation

-Dashboard visualization

Automate Detection and Containment

1. Malware incidents (e.g ransomware)
2. Analysis of phishing emails and social media messages
3. Vulnerability detection and remediation
4. Automate generation of IT service tickets

Splunk Overview

Splunk Use Cases

1. Monitore system and application performance
2. Monitor application-specific details such as the number of widgets sold
3. Detect potential security problems
4. Mitigate verified security threats
5. Real-time searching

Splunk Cloud platform

-Based on splunk Enterprise, but the platform is hosted in the cloud as a managed service

-There splunk cloud service level agreement

-No comman line interface as opposed to splunk Enterprise on premises

-On premises forwarders collect data and send it for indeximg to splunk cloud platform. A forwarder is an installation on a device that collects data and send them to other location

Splunk Components

1. The splunk Enterprise runs on Linus,Mac and Windows.

2. Splunk cloud platform
3. Splunk Enterprise console (port 8000)
4. Splunk cloud console (DNS fully qualified Domain Name)

Splunk Forwarders

It is a software that gets deployed on a host.On the forwarder data is collected (**inputs.conf**) and forward to indexer (**outputs.conf**).

There are two types of fprwarder:

Universal forwarder: It collects raw,unfiltered data.

Heavy Forwarder: Data is filtered at source and sent to various destinations.

Splunk Indexer

Data parsing.

-It process input data streams from forwarders

-You configure user defined data transformation e.g filtering out events.

Data indexing:

- Parsed items get written to disk index and can the be searched
- Files **transform.conf** can determine which events are sent to which indexes.

Splunk data ingestion

First we chose the splunk data sources e.g cloud apps,OS,apps,Microsoft active directory, file system and databases,imported files ,TCP/IP port data and script input.

Summary

This brief technical segment covered the configuration of Splunk to receive syslog data from Unix and Linux systems.

Configuration Details

- TCP connection on port 514 (standard syslog port)

- Connection_host parameter set to IP to identify the sending system
- Source type configured as tcp:syslog
- Data directed to a pre-created index called linux_syslog_index

Implementation Steps

- Configure Inputs.conf with the specified parameters
- Restart Splunk on the configured machine
- Different restart procedures may apply depending on installation type (e.g., universal forwarder on Linux)

Next Steps

- Multiple demonstrations of the configuration will be presented later in the session

Notes

Transcript

It's TCP colon slash slash port 514, which is normally used for syslog in Unix and Linux centralized logging. Connection underscore host is equal to IP, which means we want to set the IP address of the host of the system that sends the data to the machine where this is configured.

We're setting the source type to tcp colon syslog and we're directing it to be stored in an index called linux syslog underscore index which we would have previously created. Then you would restart Splunk on the machine where you've configured Inputs.conf. If it's a Linux installation, depending on what you've installed, such as a universal forwarder, you could go under

We'll be doing a bunch of demos a little bit later.

Summary

This session explains how Splunk forwarders work and how to configure data filtering in a Splunk environment.

Splunk Forwarder Types

- Splunk Indexers listen for data on port 9997 by default
- Universal forwarder: lightweight, separate download from Splunk Enterprise with minimal filtering capabilities
- Heavy forwarder: full Splunk Enterprise instance configured with detailed filtering options before data transmission

Configuration Files

- Forwarder targets are defined in the outputs.conf file
- Props.conf is used to specify data sources and transformation rules
- Transforms.conf contains stanzas for specific filtering operations
- Example configuration shown for filtering app logs using regex patterns

Data Filtering Considerations

- Heavy forwarders allow filtering before data reaches the indexer
- Example shown of using regex `\\d{6}` to match six-digit product codes
- Filtering options available at three points: forwarder, indexer, or search time
- Best practice is to filter earlier in the pipeline to reduce indexed data volume

License Optimization

- Splunk licensing is based on ingested and indexed data volume
- Filtering data before indexing helps optimize license usage
- Consider environment needs when choosing between heavy and universal forwarders

Notes

Transcript

And then forwarded to a Splunk Indexer, which could be the same host or a different host, but the Splunk Indexer, by default, listens for data to be sent to it on port 9997. Now, the two types of forwarders, of course, are a universal forwarder

and a heavy forwarder. The universal forwarder is a separate download and installation from Splunk Enterprise. It's lightweight, and it's designed to have very...

Minimal filtering options, it just really basically ingests data and sends it off to an indexer. A heavy forwarder is really just a Splunk enterprise instance configured. with an input stock conf and you could have some detailed filtering configured at the heavy forwarder before that data is sent off to an indexer. So the forwarder target or targets are defined on your forwarder machine in the output.

So think about your environment. Do we want to have a universal or a heavy forwarder? Do we want to have a lot of filtering done before data is sent to the indexer? Well, if that's the case, then you would need a heavy forwarder.

We've got the props.conf configuration file. In this case, it would be in your Splunk installation directory location under ETC system local. So what we have is a stanza where we're specifying the source we want to look at. Source, colon, colon, slash, app1, slash, log. This means we're going into the app1 directory and we have a log file called log.

Now, we then use the transforms-set directive with the value of set null comma and set parsing. This means in the transforms.conf file on the same machine, we will have two stanzas, one for set null and one for set parsing to direct how data should be treated.

So, if we go to the transforms.conf example here... In the set null stanza, we have a regular expression, regex, equal to dot. That means match everything. We're setting the destination underscore key to Q. format equal to null queue, which is really just another way of saying that we want to discard that data. However, then we are capturing specific data we want in the set parsing stanza where our regular expression or regex equals backslash d and then curly braces six.

This means that we want to match six digits in a row, so each of those digits would be within 0 through to 9. Let's say it's for a product code that results from our app log.

Now, this is pretty specific in that we're on a heavy forwarder where we can run regular expressions and parse this type of data out before it gets indexed. But, again, in your environment. Where would you filter data? Would you do it directly on the forwarder? Which means you need a heavy forwarder, a Splunk Enterprise

Instance installation. Or would you rather do it on the indexer side? Which is normally a different host on the network.

So these are the things that we want to think about. If you don't filter any data out at all, in this case from our OP1 log, technically, you could also filter it out when you are performing... It is a search after it has been indexed, but remember you want to try to limit the data that gets indexed because your Splunk license count is based on the amount of ingested and indexed data.

Summary

This tutorial demonstrates how to configure user roles, create user accounts, and set up authentication in Splunk.

Understanding Splunk Roles

- Roles are collections of related permissions assigned to users
- Built-in roles include admin, power, user, and splunk system role
- Roles can inherit settings from other roles
- Each role has specific capabilities that can be enabled/disabled
- Capabilities marked as "native" belong specifically to that role (not inherited)
- Custom roles can be created from scratch or by inheriting from existing roles

Role Permissions and Access Control

- Indexes: Control which data indexes users can access
  - Can use wildcard notation to specify groups of indexes
  - Indexes can be marked as "included" or "default"
  - Default indexes are searched when users don't specify an index
- Search restrictions: Limit what search results are returned
  - Search filter generator can help create restrictions based on indexed fields

Creating Users in Splunk

- Access user management via Settings → Users and Authentication → Users

- User properties include:
  - Username (no spaces or slashes allowed)
  - Real name
  - Email address
  - Password
  - Time zone
  - Default app (e.g., Search & Reporting)
  - Assigned role(s)
  - Password change requirements

Authentication Methods

- Direct Splunk authentication: Users created and stored in Splunk

Notes

Transcript

I'm going to begin in the upper right by clicking settings and down in the bottom right of that opened up screen you'll have users and authentication. Let's begin by going into roles because in Splunk a role is a collection of related permissions and we assign one or more roles to users.

We have an example of here such as admin, can delete, power, splunk system role, and user. If I have the user built-in role, then I can determine if it inherits its settings from another role. That's what this number one inheritance part is about that we are looking at. So nothing is checked, so it's not inheriting from another role. Next, roles have capabilities.

And the checkmark here will either be turned on or off. If it's on, obviously that capability is available, like accelerated search, changing of one's own password. And in the next column, if it says native, it means that capability is specifically a part of this role. It's not been inherited.

We already know that Inheritance hasn't been enabled for this particular role. As I go further down, we have a list of all of the other different types of permissions

that are available with this specific role named User. Now you can also create your own custom roles. There's a new role button in the upper right here where we can do this all completely from scratch. And you can even choose to inherit from an existing role to speed things up, but you don't have to.

Now, we got into inheritance and capabilities. Indexes. This is important. When it comes to searching through Splunk data, you can determine which indexes that a user with this role assigned to them would be able to access. You can even use wildcard notation to determine which group of indexes a user with this role would have access to.

Down below we have all of our indexes where we can determine whether it should be included or set as a default. The default means you would be limiting users to the indexes that you put a checkmark on for this column, whereas default means if a user performs a search and doesn't specify the index, then it is assumed to search through the indexes flagged as default.

So we can do that within a rule. We can also specify search restrictions for this rule. So this takes it a step beyond limiting which indexes are searchable by people with this role, because we can also determine what the search results are that are returned from searches. Now over on the left, you can use the search filter generator

to specify indexed fields and values. Now, because I've never done any searching, there's nothing for me to select there. What that will do is generate the search filter syntax and place it for you over here on the right. So you can do that within a role.

You don't have to. And finally, we have resources, such as determining what the default app is when a user signs in, such as search. That's interesting. We can also specify some other things like search job limits and so on. I'm going to cancel out of this. So our goal here is that we need to create a user, let's say that can perform searches. So to do that, let's go back into settings and under users and authentication in the bottom right.

I'll click users. Now my existing users are shown here. I've got a user called Splunk Admin. That is the user that I specified when I installed Splunk Enterprise. I'm going to click new user and I'm I'm going to call them Splunk Admin 1. You can't have spaces or slashes in the name of a user account. And maybe the real name of that person is John Smith.

and other details like email address. I'm going to set and confer a password for that account. I could set the default time zone for this account, the default app, so I can do that here. Let's say I set the default app here to search for search and reporting as soon as they sign in to the web console.

Here are the roles that we were looking at. These are the built-in roles. Of course, we know that we can create custom roles. So let's say I'm going to select the user.

Now it'll show up as being listed on the right once it's been selected, and you can add multiple roles if you so choose. However, I'm going to remove power, I'm just going to go back to the original where user... is the selected role. Down below, do we require a password change on first login? Well, that normally makes a lot of sense so the user can set their own passwords. I'm gonna leave that on.

And that's it. I'm going to go ahead and click save. We now have a user called Splunk Admin shown here in the list. Now it's important to understand as well that what we've just done is created a user directly in Splunk. You can reuse existing user accounts in other identity providers. LDAP compliant providers, for instance, like Microsoft Active Directory. I would set that up. I would link Splunk to Active Directory by going into Settings and down under Users and Authentication.

If I click Authentication Methods, I can choose LDAP, I can choose Configure Splunk to use LDAP, choose New LDAP, and fill out the blanks. What that means is specifying the DNS name for domain controller host, let's say, in Active Directory. The listening port number, like 636 for LDAP over SSL. The bind distinguish name, that would be the name of an account that can read users in Active Directory and the password for that account.

We can also specify a base distinguish name or a base DN within our Active Directory. In our case, all we have done is created a user directly in Splunk.

If I were to examine the file system, so I've installed Splunk Enterprise here, so if I go to Drive C, the default installation location is C Program Files, Splunk. I want to go into the ETC directory and take a peek at the password file spelled PASSWD. That will be familiar if you've managed Unix or Linux environments in the past. If I open that up, let's say in WordPad, what we're going to see is...

What we're really looking at is a file that contains user accounts that we've created in Splunk, with a hash of their password and other details like, in this case,

the real name John Smith, force password change, and so on. So at this point, why don't we test logging in as that Splunk user that we've just created.

In another browser session, I've gone to the IP of my enterprise server, port 8000. Now this time I'm going to specify the username of SplunkAdmin1, and I'll specify the password.

When I created this user account, of course it should prompt me to change it. So it says, for security reasons, the admin of this account has requested that you change your password. Yes, of course. I'm going to go ahead and fill in a new password and confirm it.

All right and then I'll click save password. Okay so now we're signed in as user John Smith. I'm just going to skip all of these detailed tour information items. Notice that I'm automatically placed in the search app according to our configuration for this user account.

And if I were to go into settings, notice that we have far fewer options available than we did when we were signed in as the main Splunk admin. And that's because of the role. The role that we've assigned to this user has basic capabilities, primarily the ability to conduct and save searches.

🟦 Splunk SPL Cheat Sheet

🔍 1. Basic Search Syntax

Syntax	Description
`index=main`	Search in the "main" index
`sourcetype=access_combined`	Limit to a specific log type (e.g., Apache logs)
`status=404`	Filter events with status 404
`host=web01`	Filter by host
`"error"`	Search for the keyword "error"
`source="/var/log/access.log"`	Specify data source

Syntax	Description
`earliest=-1h latest=now`	Time range for last 1 hour

🛠️ 2. Transforming & Aggregation Commands

Command	Description	Example
`stats`	Aggregations (count, avg, etc.)	`stats count by status`
`timechart`	Time-based trends	`timechart span=1h count`
`chart`	Grouped chart metrics	`chart avg(response_time) by uri_path`
`top`	Most common values	`top uri_path`
`rare`	Least common values	`rare status`

📊 3. Display & Formatting

Command	Description	Example
`table`	Table view	`table _time uri_path status`
`fields`	Include/exclude fields	`fields + status uri_path`
`sort`	Sort by field	`sort - count`
`rename`	Rename field	`rename uri_path as URL`
`dedup`	Remove duplicates	`dedup uri_path`
`head`	First N results	`head 10`
`tail`	Last N results	`tail 5`

✨ 4. Field & Value Creation

Command	Description	Example
`eval`	Create/modify fields	`eval duration_sec=duration/1000`
`where`	Logical filtering	`where status=500 AND response_time>2000`
`replace`	Replace values	`replace 200 with "OK" in status`
`bin`	Group numeric/time fields	`bin span=1h _time`

🔍 5. Field Extraction & Parsing

Command	Description	Example
`rex`	Regex-based field extraction	`rex "page=(?<page_name>[^&]+)"`
`spath`	Extract from JSON/XML	`spath input=payload output=code path=response.code`

🔗 6. Joining and Correlation

Command	Description	Example
`append`	Merge search results	`append [search index=backup]`
`join`	Join on common field	`join user_id [search index=logins]`
`transaction`	Group related events	`transaction clientip startswith="login" endswith="logout"`

📈 7. Visualization-Friendly SPL

Use Case	SPL	Recommended Chart
Error trend over time	`timechart count(eval(status>=400)) as errors`	Line
Top error pages	`index=main status>=400	stats count by uri_path`	Column
Avg response time by page	`stats avg(response_time) by uri_path`	Column
Request traffic	`timechart count`	Line
Status breakdown	`stats count by status`	Pie

🧪 8. Dashboard Tokens

Use	Example
Time picker	`earliest=$timepicker.earliest$ latest=$timepicker.latest$`
Dropdown	`status=$status_token$`
Dynamic SPL	`index=main status=$status_token$`

🧠 Pro Tip: Start with These

- `search`
- `table`
- `stats`
- `eval`
- `where`
- `top`
- `rex`

📦 Sample SPL Query

```
index=main status>=400
| stats count by uri_path
| sort -count
| head 5
```