

Cybersecurity

---

# Splunk Reports and Alerts

SIEM Day 3



# Class Objectives

By the end of class, you will be able to:

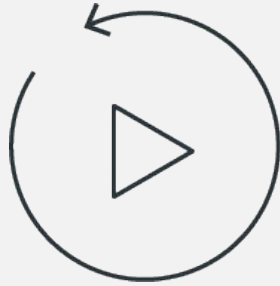
---

1 Use the SPL commands `stats` and `eval` to create new fields in Splunk.

2 Schedule statistical reports in Splunk.

3 Determine baselines of normal activity in order to trigger alerts.

4 Design and schedule alerts to notify if an attack is occurring.



**Let's recap**

Today, we will continue to learn about Splunk's capabilities.

First, let's review what was covered in the last class.



# Splunk Review

Splunk provides software utilities that search, analyze, and monitor big data with a straightforward interface. We can add additional functionality to Splunk with apps and add-ons for specific vendors and industries.



# Splunk Review

Splunk has three primary methods for accessing data:

## Monitoring

Splunk monitors logs from a system, device, or application to which it has direct access.

Businesses commonly use this method to monitor their production environment.

## Forwarding

Install a program called a forwarder on the system from which logs are collected.

Forwarders forward logs from a device into the Splunk system.

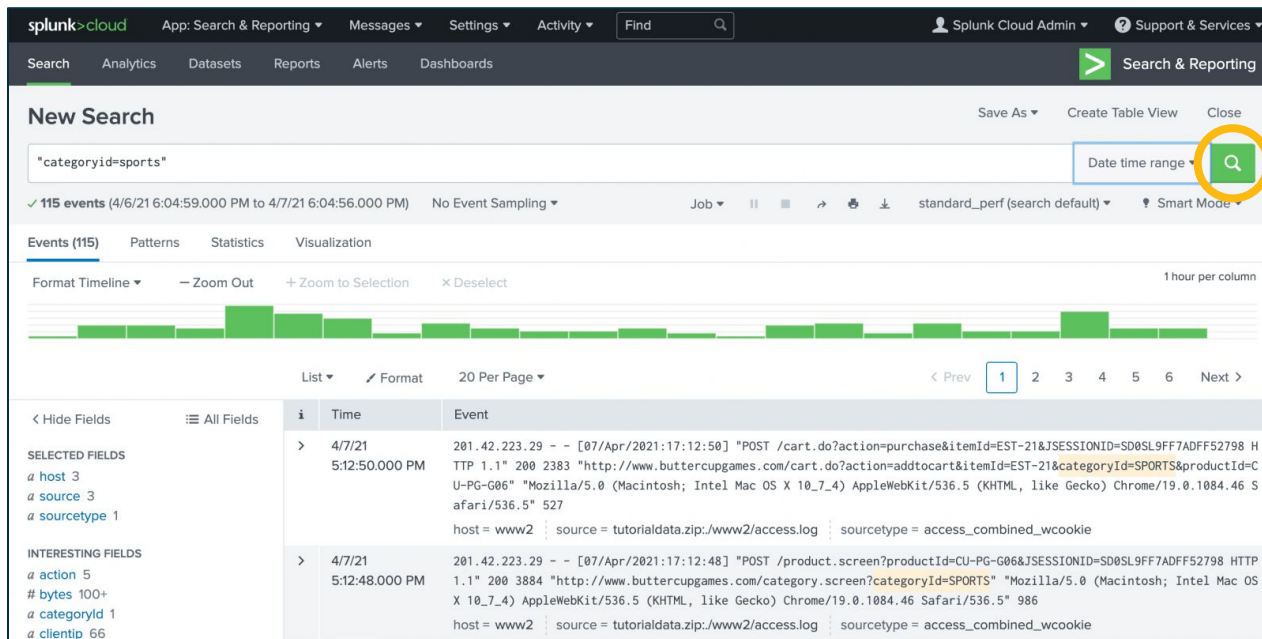
## Uploading

Manually upload logs directly into your Splunk repository.

While monitoring and forwarding are important to understand conceptually, we will primarily use the upload process for the remainder of this class.

# Splunk Review

Splunk's primary feature is **searching**, which uses a coding language native to Splunk called SPL.



The screenshot shows the Splunk Search & Reporting interface. At the top, there's a navigation bar with 'splunk>cloud' and various menu items like 'App: Search & Reporting', 'Messages', 'Settings', 'Activity', and a 'Find' search bar. Below this, there's a 'Search & Reporting' section with a 'New Search' button. The search query is 'categoryid=sports'. The results show 115 events from 4/6/21 6:04:59.000 PM to 4/7/21 6:04:56.000 PM. A visualization of the search results is shown as a bar chart. Below the chart, there's a table of the search results. The table has columns for 'Time' and 'Event'. The first event is from 4/7/21 5:12:50.000 PM, and the second is from 4/7/21 5:12:48.000 PM. Both events are HTTP POST requests to 'http://www.buttercupgames.com/cart.do?categoryid=SPORTS&productid=C...'. The table also shows fields like 'host', 'source', 'sourcetype', 'action', 'bytes', 'categoryid', and 'clientip'.

Time	Event
4/7/21 5:12:50.000 PM	201.42.223.29 - - [07/Apr/2021:17:12:50] "POST /cart.do?action=purchase&itemId=EST-21&SESSIONID=SD0SL9FF7ADFF52798 HTTP 1.1" 200 2383 "http://www.buttercupgames.com/cart.do?action=addtocart&itemId=EST-21&categoryid=SPORTS&productid=C... U-PG-G06" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_4) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 S... afari/536.5" 527 host = www2   source = tutorialdata.zip:www2/access.log   sourcetype = access_combined_wcookie
4/7/21 5:12:48.000 PM	201.42.223.29 - - [07/Apr/2021:17:12:48] "POST /product.screen?productId=CU-PG-G06&SESSIONID=SD0SL9FF7ADFF52798 HTTP 1.1" 200 3884 "http://www.buttercupgames.com/category.screen?categoryid=SPORTS" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_4) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 Safari/536.5" 986 host = www2   source = tutorialdata.zip:www2/access.log   sourcetype = access_combined_wcookie


Splunk uses **time-based search**, in which each event or log has a time associated with it.




## Activity:


### Splunk Warm Up

In this activity, you will analyze logs from a Fortinet IPS system, determine the security issue, and provide mitigation strategies.

 Breakout Rooms



The host is inviting you to join Breakout Room:  
**Breakout Room 1**



Join

Later

**Suggested Time:**

15 Minutes





**Time's up!**  
Let's review



**Questions?**





# Splunk Statistics

# Splunk Statistics

Security professionals often need to present Splunk search results to non-technical audiences using simple formats.

## For example:

If we need to illustrate the top 10 IP addresses from a DoS attack, this results page could be confusing to a non-expert.



A screenshot of the Splunk Statistics interface. At the top, it shows '12,310 events' for a specific time range. Below this is a navigation bar with 'Events (12,310)', 'Patterns', 'Statistics', and 'Visualization'. The 'Events' tab is active. Below the navigation bar is a timeline visualization with green bars. Below the timeline is a table of search results. The table has columns for 'Time' and 'Event'. The first row shows a search result for 'a src\_ip 100+'. The table also includes a 'SELECTED FIELDS' section on the left and a 'Show all 21 lines' link. The bottom of the table shows 'INTERESTING FIELDS' with values like 'host = HOST-005', 'source = WinEventLog:Security', and 'sourcetype = WinEventLog'.

# Splunk Statistics

Splunk uses the Statistics feature to display specific data points from search results in an easy-to-read format.



The **stats** command is the most basic Splunk command to create a statistics report.

The screenshot displays the Splunk Statistics interface. On the left, a list of fields is shown under 'SELECTED FIELDS' and 'INTERESTING FIELDS'. The 'Account\_Name' field is highlighted. On the right, a detailed view for 'Account\_Name' is shown, including a table of the top 10 values.

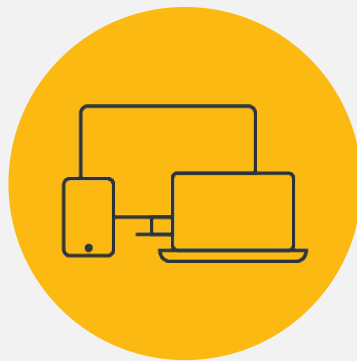
**Account\_Name**  
11 Values, 80% of events  
Selected

**Reports**  
Top values | Top values by time | Rare values  
Events with this field

Top 10 Values	Count	%
user_n	2	16.667%
ADMINISTRATOR ADMINISTRATOR	1	8.333%
BUSDEV-008 user_m	1	8.333%
PROD-POS-003 user_c	1	8.333%
user_a	1	8.333%
user_b user_d	1	8.333%
user_d	1	8.333%
user_d user_g	1	8.333%
user_e user_i	1	8.333%
user_f	1	8.333%



We will use the `stats` command to create a simple statistical report of the top account names (`Account_Name`) being targeted in a brute-force attack.



# Instructor **Demonstration**

Splunk Statistics

# Creating Fields with eval

We can use Splunk to create new fields and add them to a statistical report.

## For example:

Suppose we are analyzing logs for potential brute-force attempts.

Our logs have event codes (`EventCode`) that assign numerical codes to events.

`EventCode 4740`  
indicates a user logout.

Since user lockouts are a potential identifier of a brute-force attack, we can create a new field to identify events that contain this event code.



If the `EventCode` field has a value of `4740`, the field value will be `Potential Brute Force`.



If the field does not have a value of `4740`, the field value will be `Not Brute Force`.



# The eval Command

We can use the `eval` command to create fields.

- 1 The `eval` command calculates an expression (such as `if then`) and places the resulting values into a search field.
- 2 If the search field doesn't exist, it creates a new search field.
- 3 If the search field does exist, it overwrites the field with the new values.

```
source="statsreport.csv" | eval BruteForce = if('EventCode'="4740",  
"Potential Brute Force", "Not Brute Force")
```

# The eval Command

We can use the `eval` command expressions, such as `if`, and place the resulting values into a search field.



Searches through all the results from the `statsreport.csv` file.

```
source="statsreport.csv" | eval BruteForce =  
if('EventCode'="4740", "Potential Brute Force", "Not Brute Force")
```

# The eval Command

We can use the `eval` command expressions, such as `if`, and place the resulting values into a search field.



Creates a new field called `BruteForce`.

```
source="statsreport.csv" | eval BruteForce =  
if('EventCode'="4740", "Potential Brute Force", "Not Brute Force")
```

# The eval Command

We can use the `eval` command expressions, such as `if`, and place the resulting values into a search field.

```
source="statsreport.csv" | eval BruteForce =  
if('EventCode'="4740", "Potential Brute Force", "Not Brute Force")
```



States the following expression: "If the `EventCode` field has a value of `4740`."

# The eval Command

We can use the `eval` command expressions, such as `if`, and place the resulting values into a search field.

```
source="statsreport.csv" | eval BruteForce =  
if('EventCode'="4740", "Potential Brute Force", "Not Brute Force")
```



Continue the statement,  
"If true, name this value  
`Potential Brute Force`."

# The eval Command

We can use the `eval` command expressions, such as `if`, and place the resulting values into a search field.

```
source="statsreport.csv" | eval BruteForce =  
if('EventCode'="4740", "Potential Brute Force", "Not Brute Force")
```



Continues the statement,  
"If false, name this value  
`Not Brute Force`."

# The eval Command

We can use the `eval` command expressions, such as `if`, and place the resulting values into a search field.



Searches through all the results from the `statsreport.csv` file.



Creates a new field called `BruteForce`.

```
source="statsreport.csv" | eval BruteForce =  
if('EventCode'="4740", "Potential Brute Force", "Not Brute Force")
```



States the following expression: "If the `EventCode` field has a value of `4740`."



Continue the statement, "If true, name this value `Potential Brute Force`."



Continues the statement, "If false, name this value `Not Brute Force`."



## Activity:

### Splunk Statistics

---

In this activity, you will create statistical reports to illustrate details about the DoS attack.

**Suggested Time:**  
15 Minutes







**Time's up!**  
Let's review



# Questions?





# Splunk Reports

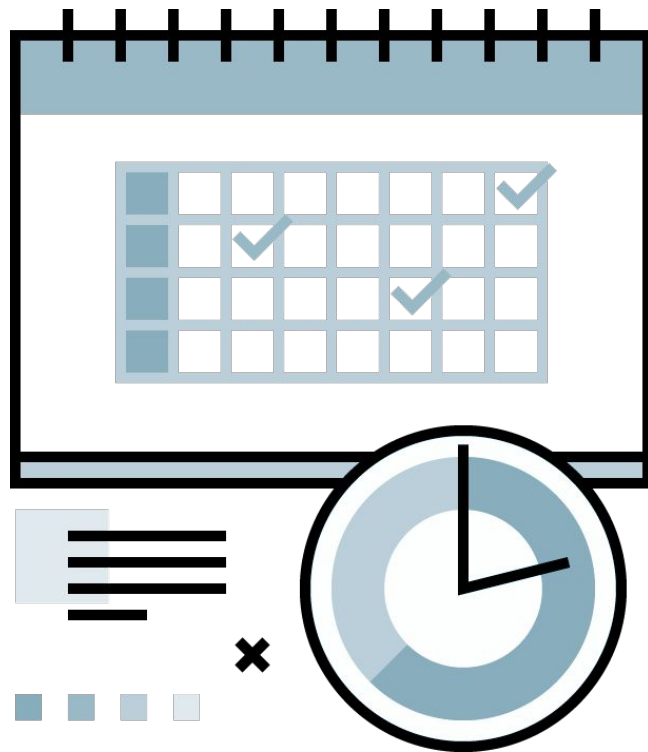
# Splunk Reports

Statistical reports may need to be run at specific or recurring times.

## For example:

If an organization is experiencing suspicious network attacks around 12am daily, it would want to analyze its network traffic every night at that time.

With Splunk, we can create and schedule custom reports to automate this task.



# Splunk Report Demonstration

In the following demonstration, we will create and schedule a report using the continued scenario of monitoring brute-force attacks:

- 1 We were notified that the most recent brute-force attacks happened around 12am.
- 2 Therefore, we will run a report at 1am each night to view activity for the past several hours.
- 3 We'll also automate an email linking to the report after it runs.



# Instructor **Demonstration**

Creating and Scheduling Reports



## Activity:

### Splunk Reports

---

In this activity, you will schedule a statistical report for OMP management so they can review the current state of the attacks against a server.

**Suggested Time:**

15 Minutes



**Time's up!**  
Let's review





# Questions?



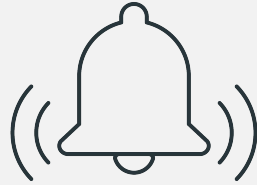


**Break**  
15 mins

Countdown timer

**15:00**

(with alarm)



# Splunk Alerts

# Splunk Alerts

So far we have covered how Splunk statistics and reports can help information security professionals identify security issues.

- This process can be further improved through the use of **alerts**.
- Splunk alerts are designed to automatically notify one or more individuals when a specific condition, known as a **trigger condition**, is met.
- Splunk alerts are **automatic**. Once they are created, Splunk's software checks the trigger condition.

The screenshot shows the 'Save As Alert' dialog box in Splunk. The dialog is divided into several sections: 'Settings', 'Alert type', 'Trigger Conditions', and 'Trigger Actions'. The 'Alert type' section is highlighted with an orange box, showing 'Scheduled' selected over 'Real-time', with a frequency of 'Run every week' and a schedule of 'On Monday at 6:00'. The 'Trigger Conditions' section is also highlighted with an orange box, showing 'Trigger alert when' set to 'Number of Results', which is 'is greater than 0'. The 'Trigger' is set to 'Once' and 'Throttle' is unchecked. The 'Trigger Actions' section has a '+ Add Actions' button. At the bottom right, there are 'Cancel' and 'Save' buttons.

**Save As Alert**

**Settings**

Title: Alert File Size

Description: Email Alert when the file size report is run

Permissions: Private | Shared in App

**Alert type**: Scheduled | Real-time

Run every week ▼

On: Monday ▼ at: 6:00 ▼

**Trigger Conditions**

Trigger alert when: Number of Results ▼

is greater than ▼ 0

Trigger: Once | For each result

Throttle ? ☐

**Trigger Actions**

+ Add Actions ▼

Cancel Save

# Splunk Alerts

A Splunk user selects a trigger condition based on the security event they are trying to monitor. Trigger conditions contain the following:

01

## Search/report results

indicate which criteria to check.

### For example:

300 logins have been attempted.

02

## Time parameters

indicate the time period to check.

### For example:

Within last 24 hours

03

## Schedule

determines the frequency with which these criteria are checked.

### For example:

Every day at 12pm

# Splunk Alerts

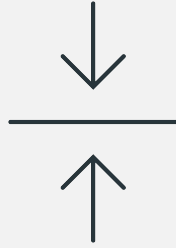
When the condition is met, a **trigger action** is executed to alert the Splunk user.

**For example:**

```
"Send an email to soc_manager@acme.com."
```

In summary, the complete alert would be:

```
Every day at 12pm, check if at least 300 login attempts have  
occurred within the last 24 hours. If this condition is met,  
send an email to soc_manager@acme.com.
```



# Baselining

# Designing Strong Alerts

A required skill for designing strong alerts is avoiding **false positives** and **false negatives**.

	False positive	False negative
What occurred	Regular login activity	Brute-force attack
Alerts	Yes	No
Outcome	Alerts went off but security professionals identified a non-issue.	No alerts went off, a brute-force attack occurred, and several accounts were breached.



# False Positives

**False positives** occur when conditions are met and an alert is triggered, but the security situation did not actually occur.

- 1 For example, an alert is created to detect suspicious login activity on our Linux server.
- 2 The chosen criteria checks activity every hour and creates an alert when 10 login attempts occur within an hour.
- 3 Several alerts were triggered per these conditions, but further research determined the alerts were set off by normal user activity.
- 4 The SOC realizes that 10 login attempts within an hour is not very suspicious.

# False Negatives

**False negatives** occur when the condition is met and an alert is not triggered, meaning the security situation occurred undetected.

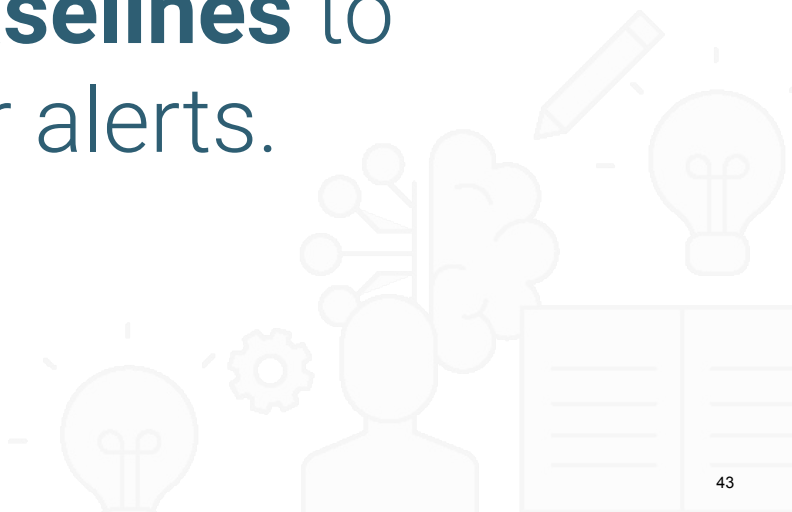
1 For example, an alert was created to detect suspicious login activity on our Linux server.

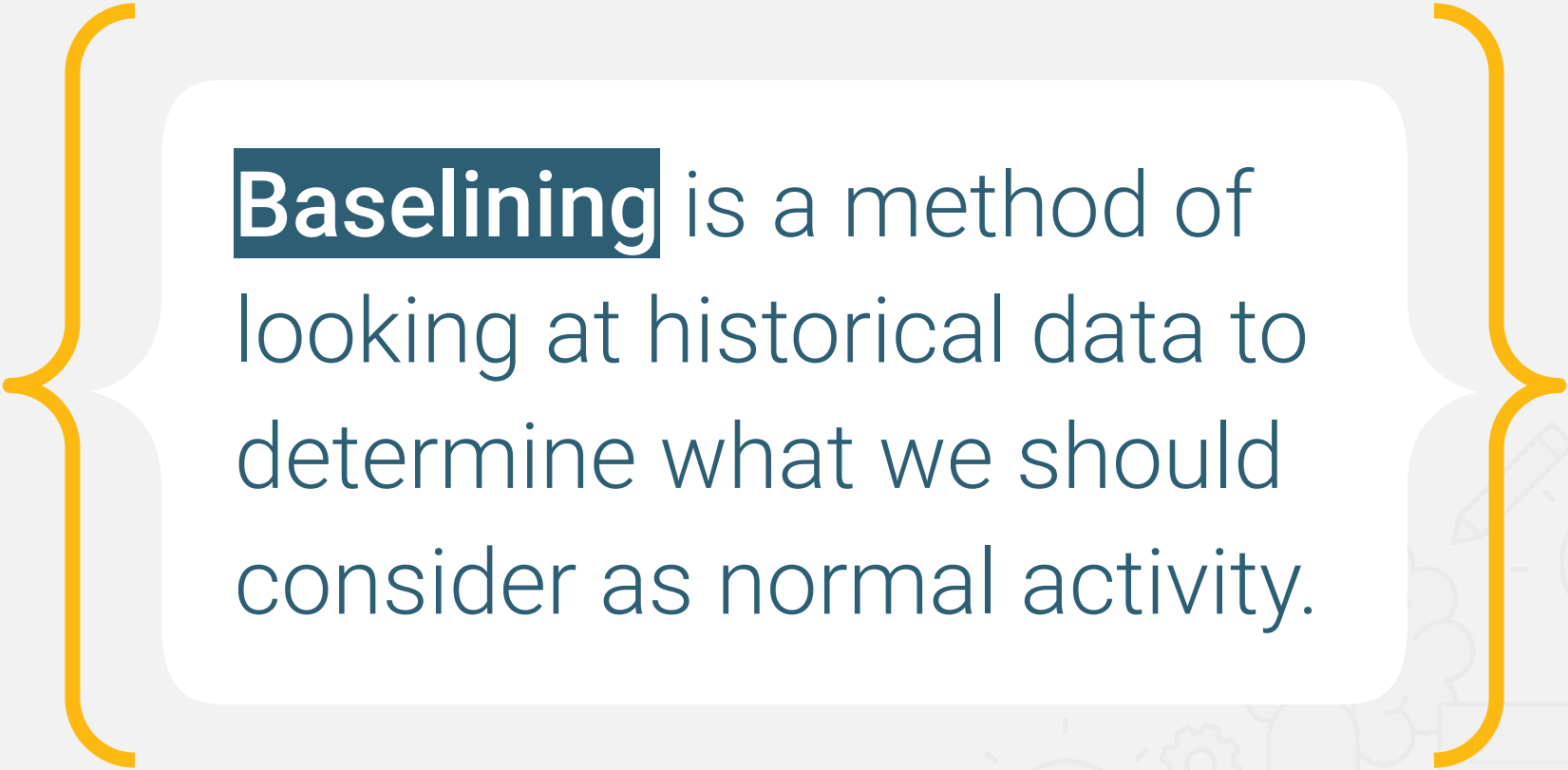
2 The chosen criteria checks activity every hour and creates an alert when 500 login attempts occur within an hour.

3 Suspicious login activity did occur on the server when an attacker tried to brute force the Linux server with 400 attempts, but no alerts were triggered.

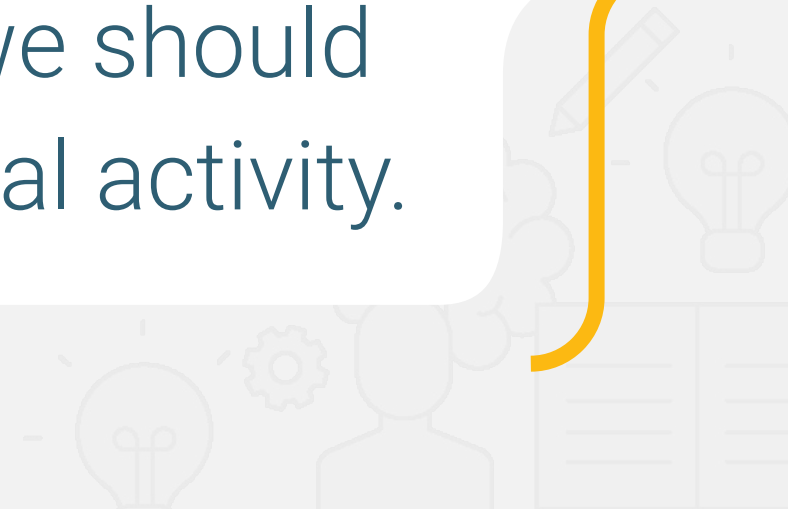


Security professionals can avoid these false results by using **baselines** to design their alerts.





**Baselining** is a method of looking at historical data to determine what we should consider as normal activity.



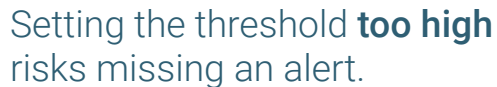


# Instructor **Demonstration**

Baselining

# Setting a Baseline Threshold

Baselining is a method of looking at historical data to determine typical activity, known as a threshold. When the **threshold** is exceeded, an alert is triggered.



Setting the threshold **too high**  
risks missing an alert.

Setting the threshold **too low**  
creates too many false positives.



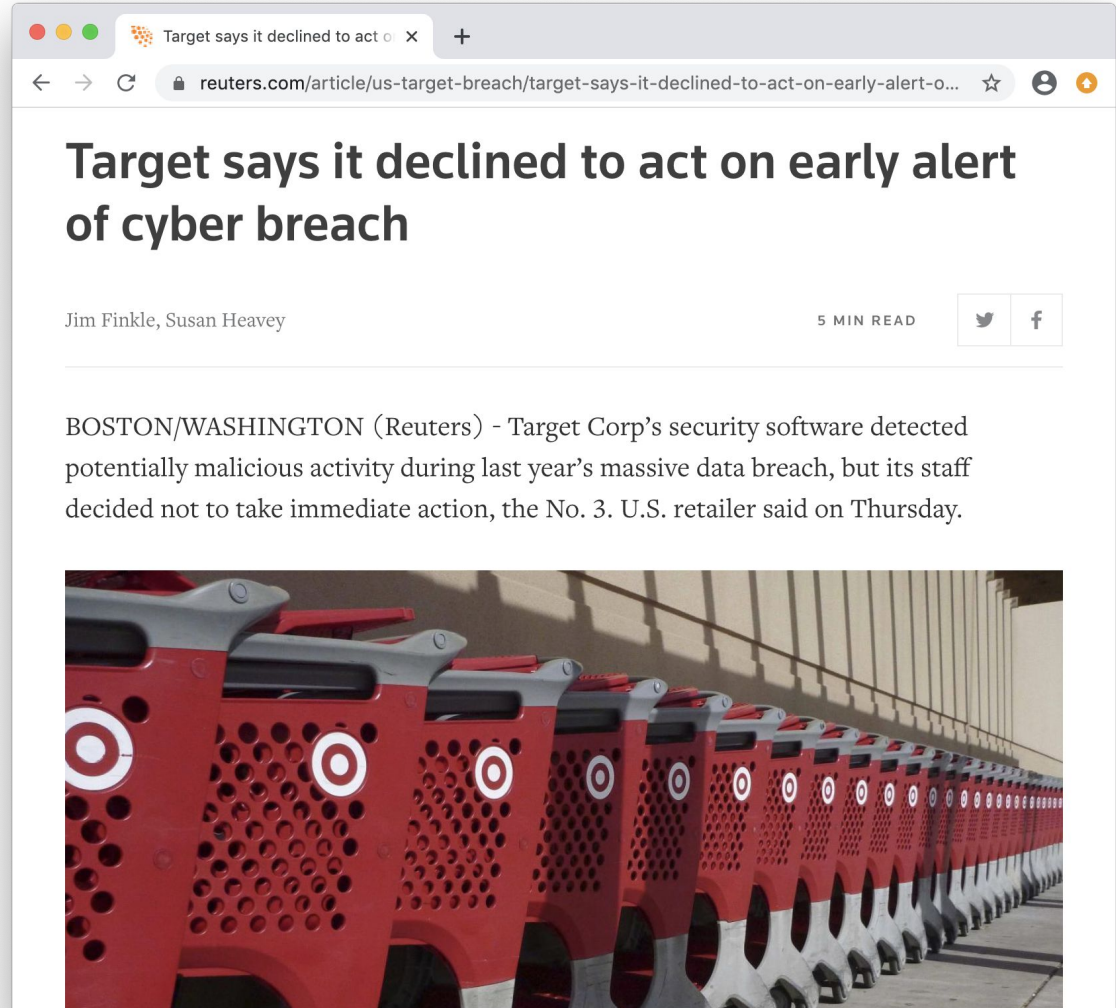
**Alert fatigue** occurs when security professionals receive so many alerts that they cannot adequately respond to each one.

- Even when an organization builds good alerts and an alert gets triggered, security professionals will need to research and respond to it.
- If an organization's system triggers too many alerts, even if they are good alerts, security professionals will often miss issues as they get lost in the noise.

# Alert Fatigue

Alert fatigue can have a major impact on organizations:

- In 2014, a breach at Target cost the company US\$252 million and led to the resignation of its CIO and CEO.
- One of the company's security products actually detected the breach.
- But due to the high quantity of alerts and the frequency of false alerts, the company's IT security team ignored it.





# Alert Fatigue

To prevent alert fatigue:





## Activity:

### Baselining

---

In this activity, you will review logs and create a baseline of typical hourly login counts.

**Suggested Time:**

15 Minutes



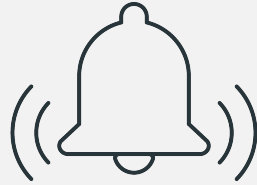


**Time's up!**  
Let's review



# Questions?





# **Creating** and **Scheduling** Alerts

# Creating and Scheduling Alerts

Now that we can determine accurate baselines, we can continue with our scenario and design the alerts.

- We will design an alert to trigger when 30 login attempts occur in an hour.
- We will run this alert to check the count every hour.
- Once the alert is triggered, an email will be sent.

### Save As Alert

When triggered

▼

✉ Send email

Remove

To

soc@securityteam.com

Comma separated list of email addresses.  
[Show CC and BCC](#)

Priority

Normal ▼

Subject

Log Validation Alert

The email subject, recipients and message can include tokens that insert text based on the results of the search.  
[Learn More](#)

Message

Logins went over 30 in an hour, please investigate

Cancel

Save



# Instructor **Demonstration**

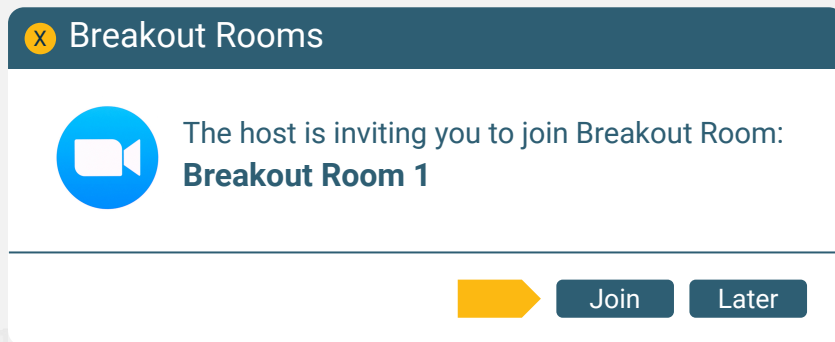
Creating and Scheduling Alerts



## Activity:

Creating and Scheduling Alerts

In this activity, you will design and schedule an alert to notify your team if a brute-force attack is occurring.



**Suggested Time:**  
15 Minutes





**Time's up!**  
Let's review



# Questions?





**The End**