



# Splunk Reports and Alerts

Cybersecurity

SIEM Day 3



# Class Objectives

---

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



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



Schedule statistical reports in Splunk.



Determine baselines of normal activity to trigger alerts.



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



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:

## Monitor

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

This method is commonly used by businesses to monitor their production environment.

## Forward

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

Forwarders forward logs from a device into the Splunk system.

## Upload

**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.

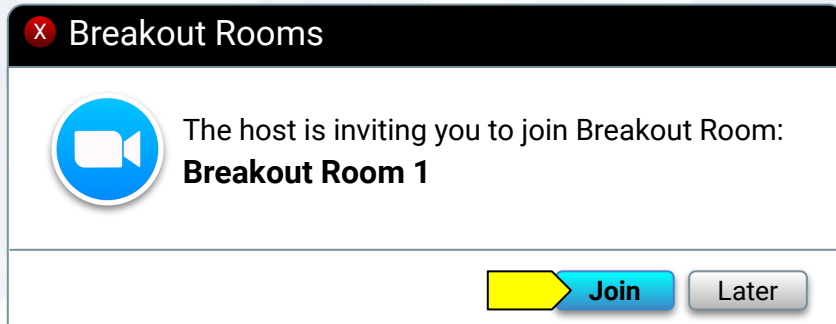
The screenshot shows the Splunk Search & Reporting interface. At the top, there's a navigation bar with 'splunk>cloud' and various menu items like 'App: Search & Reporting', 'Messages', 'Settings', 'Activity', and a 'Find' search bar. Below this is a sub-navigation bar with 'Search', 'Analytics', 'Datasets', 'Reports', 'Alerts', and 'Dashboards'. The main content area is titled 'New Search' and shows a search query 'categoryid=sports'. Below the query, it indicates '115 events' and a time range from '4/6/21 6:04:59.000 PM to 4/7/21 6:04:56.000 PM'. There's a 'Date time range' dropdown and a search button (magnifying glass icon) which is highlighted by a yellow arrow. Below the search bar, there's a visualization section with a bar chart showing event counts over time. At the bottom, there's a table of search results with columns for 'Time' and 'Event'. The table shows two events from 4/7/21, both at 5:12:50.000 PM and 5:12:48.000 PM respectively, with detailed log entries for each.

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&JSESSIONID=SD0SL9FF7ADFF52798 HTTP/1.1" 200 2383 "http://www.buttercupgames.com/cart.do?action=addtocart&itemId=EST-21&categoryId=SPORTS&productId=CU-PG-G06" "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" 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&JSESSIONID=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.



Suggested Time:

15 Minutes





Time's Up! Let's Review.

# Questions?



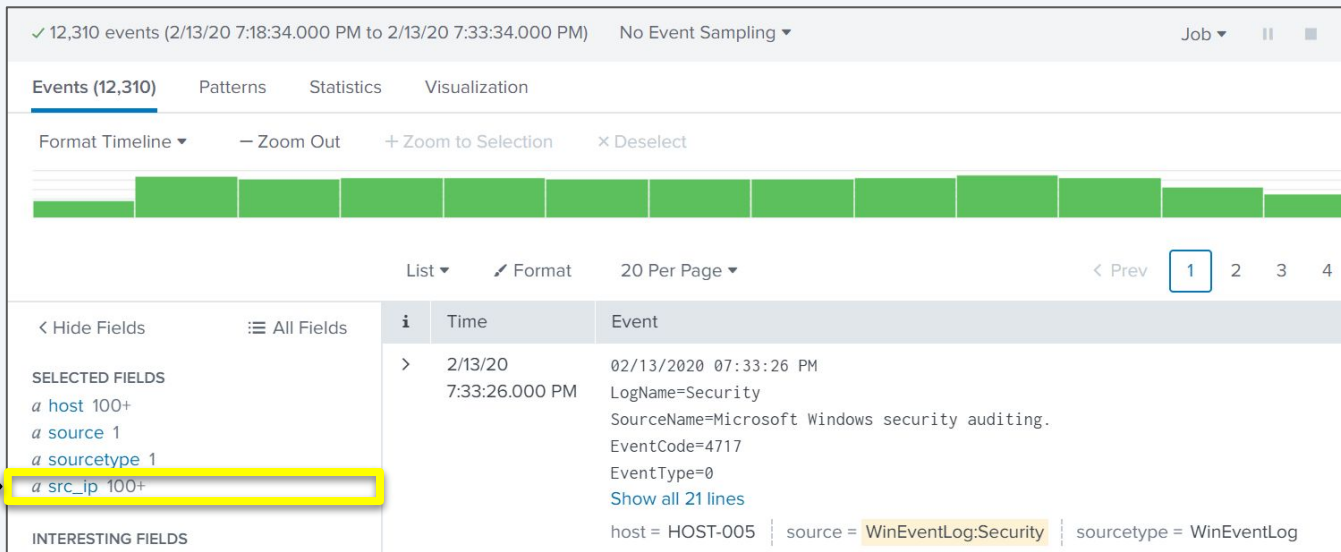
# Splunk Statistics

# 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.



# 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 modal window titled 'Account\_Name' is open, showing '11 Values, 80% of events'. Below this, there are tabs for 'Reports', including 'Top values', 'Top values by time', and 'Rare values'. The 'Top 10 Values' report is displayed as a table with columns for the field name, count, and percentage.

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

---

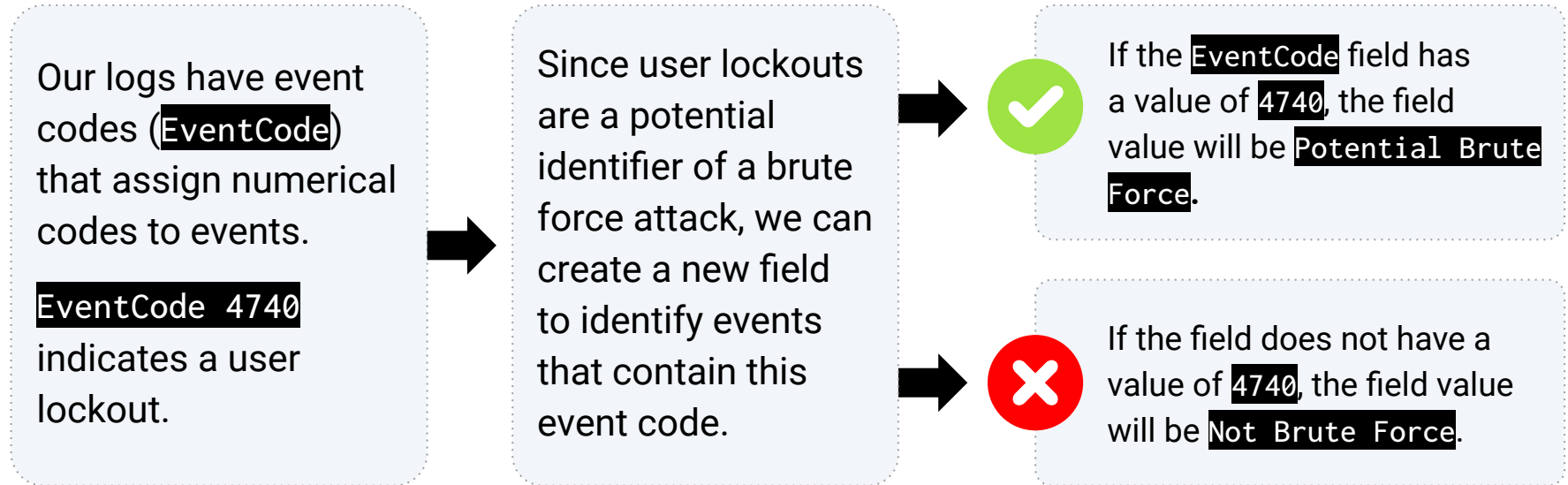
## Splunk Statistics

# 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.





# The eval Command

---

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



The `eval` command calculates an expression (such as `if then`) and places the resulting values into a search field.



If the search field doesn't exist, it creates a new search field.



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")
```



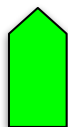
States the following expression: "If the **EventCode** field has a value of **4740**."

# 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")
```



Continues 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.

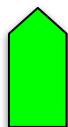


Creates a new field called `BruteForce`.

```
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

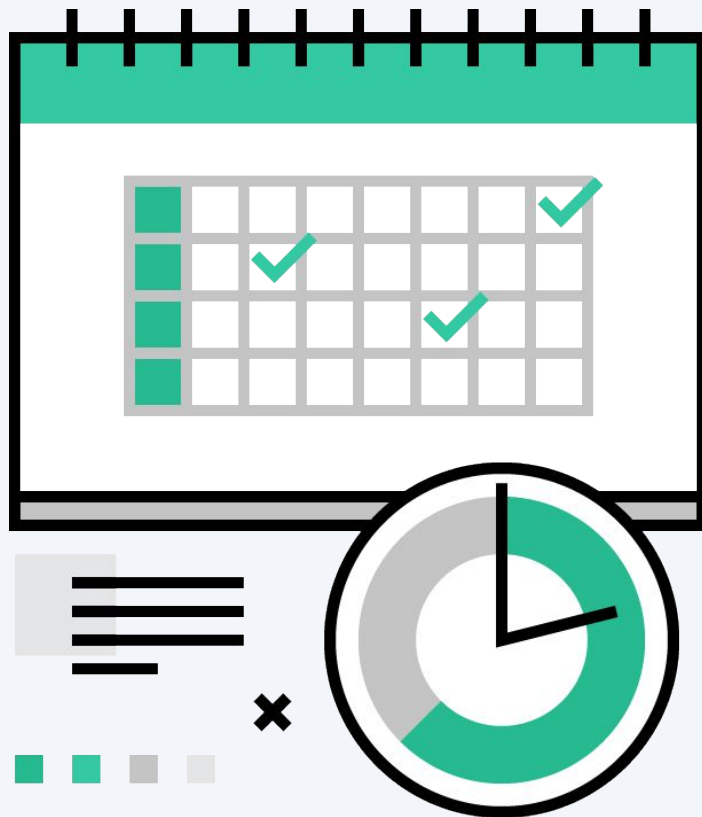
# Splunk Reports

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

## For example:

If an organization is experiencing suspicious network attacks around 12 a.m., they would want to analyze their 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:



We were notified that the most recent brute force attacks happened around 12 a.m.



Therefore, we will run a report at 1 a.m. each night to view activity for the past several hours.



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?





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 an individual or 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**:
  - Title**: Alert File Size
  - Description**: Email Alert when the file size report is run
  - Permissions**: Private (selected), Shared in App
- Alert type**: Scheduled (selected), Real-time
- Frequency**: Run every week ▼
- On**: Monday ▼ **at**: 6:00 ▼
- Trigger Conditions**:
  - Trigger alert when**: Number of Results ▼
  - Comparison**: is greater than ▼
  - Value**: 0
  - Trigger**: Once (selected), For each result
  - Throttle**: ☐
- Trigger Actions**: + Add Actions ▼

At the bottom right, there are 'Cancel' and 'Save' buttons.

# 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 by which these criteria are checked.

### **For example:**

Every day at 12 p.m.

# 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](mailto:soc_manager@acme.com)."

In summary, the complete alert would be:

Every day at 12 p.m., 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](mailto: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.



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



The chosen criteria checks activity every hour and creates an alert when 10 login attempts occur within an hour.



Several alerts were triggered per these conditions, but further research determined the alerts were set off by normal user activity.



SOC realizes that 10 login attempts within an hour is not very suspicious.

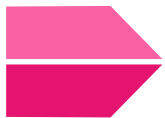
# False Negatives

---

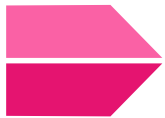
**False negatives** occur when the condition is met and an alert is not triggered, meaning the security situation occurred undetected.



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



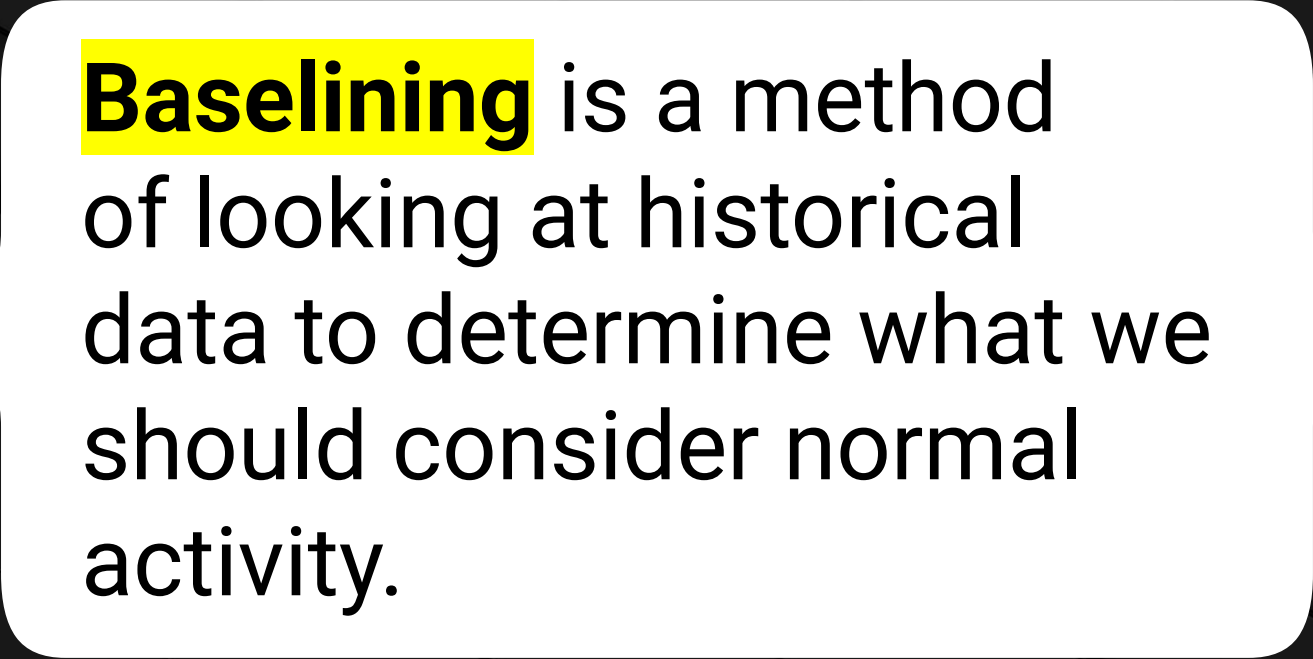
The chosen criteria checks activity every hour and creates an alert when 500 login attempts occur within an hour.



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 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.



[illegible]

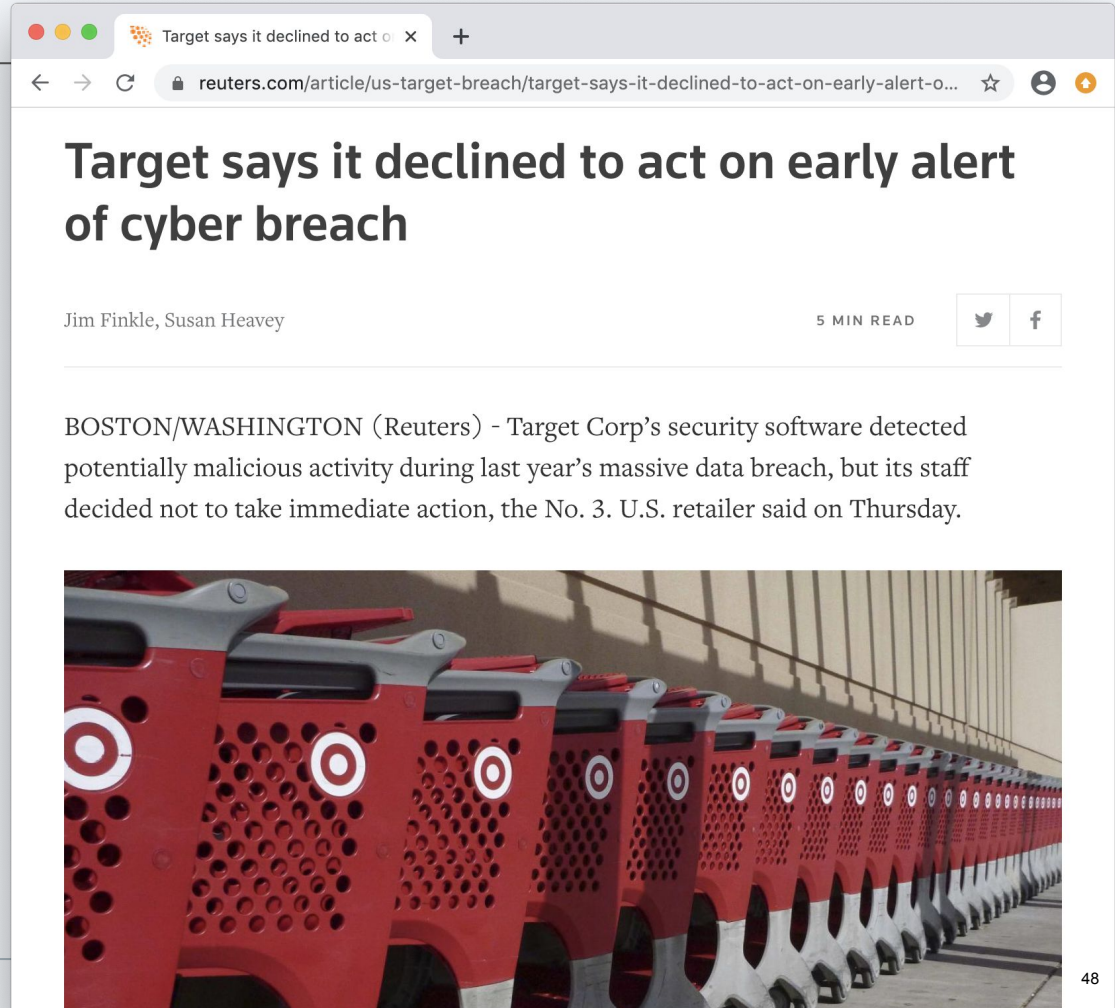
**Alert fatigue** occurs when security professionals receive so many alerts that they are prevented from adequately responding to each one.

- 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 \$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: Baselineing

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

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

Remove

Cancel

Save



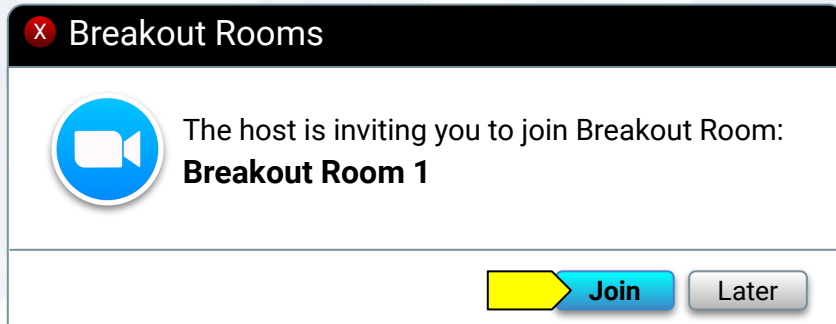
# Instructor Demonstration

---

## 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*