**Activity File: Exploring Kibana**

* You are a DevOps professional and have set up monitoring for one of your web servers. You are collecting all sorts of web log data and it is your job to review the data regularly to make sure everything is running smoothly.
* Today, you notice something strange in the logs and you want to take a closer look.
* Your task: Explore the web server logs to see if there's anything unusual. Specifically, you will:

:warning: **Heads Up**: These sample logs are specific to the time you view them. As such, your answers will be different from the answers provided in the solution file.

**Instructions**

1. Add the sample web log data to Kibana.
2. Answer the following questions:
   * In the last 7 days, how many unique visitors were located in India?
   * In the last 24 hours, of the visitors from China, how many were using Mac OSX?
   * In the last 2 days, what percentage of visitors received 404 errors? How about 503 errors?
   * In the last 7 days, what country produced the majority of the traffic on the website?
   * Of the traffic that's coming from that country, what time of day had the highest amount of activity?
   * List all the types of downloaded files that have been identified for the last 7 days, along with a short description of each file type (use Google if you aren't sure about a particular file type).
3. Now that you have a feel for the data, Let's dive a bit deeper. Look at the chart that shows Unique Visitors Vs. Average Bytes.
   * Locate the time frame in the last 7 days with the most amount of bytes (activity).
   * In your own words, is there anything that seems potentially strange about this activity?
4. Filter the data by this event.
   * What is the timestamp for this event?
   * What kind of file was downloaded?
   * From what country did this activity originate?
   * What HTTP response codes were encountered by this visitor?
5. Switch to the Kibana Discover page to see more details about this activity.
   * What is the source IP address of this activity?
   * What are the geo coordinates of this activity?
   * What OS was the source machine running?
   * What is the full URL that was accessed?
   * From what website did the visitor's traffic originate?
6. Finish your investigation with a short overview of your insights.
   * What do you think the user was doing?
   * Was the file they downloaded malicious? If not, what is the file used for?
   * Is there anything that seems suspicious about this activity?
   * Is any of the traffic you inspected potentially outside of compliance guidlines?

© 2020 Trilogy Education Services, a 2U, Inc. brand. All Rights Reserved.

**Solution Guide: Exploring Kibana**

1. Start by adding the sample web log data to Kibana.
   * You can import it by clicking **Try our sample data**.
   * Or you can import it from the homepage by clicking on **Load a data set and a Kibana dashboard** under **Add sample data**.
   * Click **Add Data** under the **Sample Web Logs** data pane.
   * Click **View Data** to pull up the dashboard.
2. Answer the following questions:
   * In the last 7 days, how many unique visitors were located in India?
     + **Example Answer:** 253
   * In the last 24 hours, of the visitors from China, how many were using Mac OSX?
     + **Example Answer:** 7
   * In the last 2 days, what percentage of visitors received 404 errors? How about 503 errors?
     + **Example Answer:** 404: 6.667% and 503: 13.333%
   * In the last 7 days, what country produced the majority of the traffic on the website?
     + **Example Answer:** China
   * Of the traffic that's coming from that country, what time of day had the highest amount of activity?
     + **Example Answer:** 12 p.m. and 1 p.m. (hours 12 and 13)
   * List all the types of downloaded files that have been identified for the last 7 days, along with a short description of each file type (use Google if you aren't sure about a particular file type).
     + **Example Answer:**
       - **gz:** .gz files are compressed files created using the gzip compression utility.
       - **css:** .css files can help define font, size, color, spacing, border and location of HTML information on a webpage. They are downloaded with their .html counterparts and rendered by the browser.
       - **zip:** A lossless compression format. A .zip file may contain one or more files or directories that have been compressed.
       - **deb:** A file with the .deb file extension is a Debian (Linux) Software Package file. These files are installed when using the apt package manager.
       - **rpm:** .rpm file formats are a Red Hat Software Package file. RPM stands for Red Hat Package Manager.
3. Look at the chart that shows Unique Visitors Vs. Average Bytes.
   * Locate the time frame in the last 7 days with the most amount of bytes (activity).
   * In your own words, is there anything that seems potentially strange about this activity?

**Example Answer:** (Your results may be different.) In our example, it seems strange that *one* visitor is using a number of bytes that is considerably higher than all other usage.

1. Filter the data by this event.
   * What is the timestamp for this event?
     + **Example Answer:** The time filter shows Sep 13, 2020 @ 21:00 -> Sep 14, 2020 @ 00:00. The time stamp is 22:55.
   * What kind of file was downloaded?
     + **Example Answer:** An RPM file
   * From what country did this activity originate?
     + **Example Answer:** India
   * What HTTP response codes were encountered by this visitor?
     + **Example Answer:** 200 OK
2. Switch over to the Kibana Discover page to see more details about this activity.
   * What is the source IP address of this activity?
     + **Example Answer:** 35.143.166.159
   * What are the geo coordinates of this activity?
     + **Example Answer:** { "lat": 43.34121, "lon": -73.6103075 }
   * What OS was the source machine running?
     + **Example Answer:** Windows 8
   * What is the full URL that was accessed?
     + **Example Answer:** <https://artifacts.elastic.co/downloads/beats/metricbeat/metricbeat-6.3.2-i686.rpm>
   * From what website did the visitor's traffic originate?
     + **Example Answer:** Facebook
3. Finish your investigation with a short overview of your insights.
   * What do you think the user was doing?
     + **Example Answer:** This event appears to be a user downloading a Linux package from the website being monitored.
   * Was the file they downloaded malicious? If not, what is the file used for?
     + Linux packages aren't typically malicious, but they could be. Depending on the website, this could be harmless traffic from a sysadmin performing an update.
   * Was there anything that seems suspicious about this activity?
   * Is any of the traffic you inspected potentially outside of compliance guidelines?
     + **Example Answer:** The main concern is the referral link from Facebook, as it's probably not within compliance to post package update links on Facebook.
     + This user could be further investigated and monitored for suspicious activity.

© 2020 Trilogy Education Services, a 2U, Inc. brand. All Rights Reserved.

## Activity File: Interview Questions

* This first project covers a wide range of topics including cloud, network security, and logging and monitoring.
* When networking and talking to potential employers, you should be able to reference the work done on this project to answer specific interview questions or demonstrate your skills within a specific domain.
* You will choose a domain that you're interested in pursuing as a career and answer mock questions based on the suggested response format.​

### Instructions

1. Choose one of the following domains:
   * Network security
   * Cloud security
   * Logging and monitoring

If you are unsure of which domain you want to focus on, that's okay. You can either choose the one you're most comfortable discussing, or complete the tasks in two or all three domains.

1. Select one domain and one question.
   * Questions are provided for each domain. Choose one to answer from your chosen domain.​
2. Write a one-page response that answers the question using specific examples from your work on Project 1. Your response should flow and read like a presentation while keeping the general structure of the technical question response guidelines.

You will submit this one-page response.

#### Reminder: Response Guidelines

As a reminder, good responses do the following.​

1. Restate the problem.
2. Provide a concrete example scenario.
3. Explain the solution requirements.
4. Explain the solution details.
5. Identify advantages and disadvantages of the solution​.​Including each of these components will ensure you prove your competency of subject matter and critical thinking.​

### Interview Questions by Domain

Below you will find a list of questions, grouped by specific domains. Select one question to answer.​

For each question, where appropriate, we have provided you with specific prompts to consider as you structure each section of your response. Feel free to use these prompts or your own examples.

#### Domain: Network Security

**Question 1: Faulty Firewall**

Suppose you have a firewall that's supposed to block SSH connections, but instead lets them through. How would you debug it?

Make sure each section of your response answers the questions laid out below.​

1. Restate the Problem
2. Provide a Concrete Example Scenario
   * In Project 1, did you allow SSH traffic to all of the VMs on your network?
   * Which VMs did accept SSH connections?
   * What happens if you try to connect to a VM that does not accept SSH connections? Why?
3. Explain the Solution Requirements
   * If one of your Project 1 VMs accepted SSH connections, what would you assume the source of the error is?
   * Which general configurations would you double-check?
   * What actions would you take to test that your new configurations are effective?
4. Explain the Solution Details
   * Which specific panes in the Azure UI would you look at to investigate the problem?
   * Which specific configurations and controls would you check?
   * What would you look for, specifically?
   * How would you attempt to connect to your VMs to test that your fix is effective?
5. Identify Advantages/Disadvantages of the Solution
   * Does your solution guarantee that the Project 1 network is now "immune" to all unauthorized access?
   * What monitoring controls might you add to ensure that you identify any suspicious authentication attempts?​

**Question 2: Unsecured Web Server**

Suppose you find a server running HTTP on port 80, despite compliance guidelines requiring encryption in motion. What do you do?​​

1. Restate the Problem
2. Provide a Concrete Example Scenario
   * In Project 1, did you have servers running HTTP on port 80? If so, why was it permissible to do so?
   * In a real deployment, which specific machine would you configure differently? How, and why?
3. Explain the Solution Requirements
   * Why is running HTTP on port 80 a potential problem?
   * How would you reconfigure a server to serve HTTP traffic safely?
   * How does this solution fix the problem?
4. Explain the Solution Details
   * Which tools and technologies would you use to implement this solution in Project 1?
   * How, specifically, would you use these tools to harden your deployment?
5. Identify Advantages and Disadvantages of the Solution
   * Will your solution break clients that used to communicate with the server over port 80?
   * Do you have to do any work to keep this solution running longterm? Or can you simply "set it and forget it?”

#### Domain: Cloud Security

**Question 1: Cloud Access Control**

How would you control access to a cloud network?

1. Restate the Problem
2. Provide a Concrete Example Scenario
   * In Project 1, did you deploy an on-premises or cloud network?
   * Did you have to configure access controls to this network?
   * What kinds of access controls did you configure, and why were they necessary?
   * How do these details relate to the interview question?
3. Explain the Solution Requirements
   * In Project 1, what kinds of access controls did you have to implement? Consider:
     + NSGs around the VNet? Around the VMs?
     + Local firewalls (ufw, etc.) on each VM?
     + Protocol allow/deny lists?
   * What did each access control achieve, and why was this restriction necessary for the project?
4. Explain the Solution Details
   * Which rules do you set for each NSG in the network?
   * How does access to the jump box work?
   * How does access from the jump box to the web servers work?
5. Identify Advantages/Disadvantages of the Solution
   * Does your solution scale?
   * Is there a better solution than a jump box?
   * What are the disadvantages of implementing a VPN that kept you from doing it this time?
   * What are the advantages of a VPN?
   * When is it appropriate to use a VPN?

**Question 2: Corporate VPN**

What are the advantages and disadvantages of using a corporate VPN, and under what circumstances is using one appropriate?

1. Restate the Problem
2. Provide a Concrete Example Scenario
   * In Project 1, which VMs did you have on the network?
   * Which tools did you use to control access to and from the network?
   * If you didn't use a VPN, what did you use?
   * What disadvantage(s) did your non-VPN solution have?
   * What advantage(s) did your non-VPN solution have?
3. Explain the Solution Requirements
   * Would a VPN meet the access control requirements you had for Project 1?
   * How would a VPN protect the network just as well, or better, than your current solution?
4. Explain the Solution Details
   * Which Azure tools would you use to implement a VPN to your Project 1 network?
   * How would you onboard users to the new VPN system?
5. Identify Advantages and Disadvantages of the Solution
   * In Project 1, would a VPN have been an appropriate access control solution?
   * Under what circumstances is a VPN a good solution?
   * When, if ever, is a VPN "overkill"?

**Question 3: Containers**

When is it appropriate to use containers in cloud deployments, and what are the security benefits of doing so?

1. Restate the Problem
2. Provide a Concrete Example Scenario
   * In Project 1, when did you use containers?
   * What did you use containers for?
3. Explain the Solution Requirements
   * Why was this an appropriate use for containers?
   * What security benefits did you expect from using containers?
4. Explain the Solution Details
   * In Project 1, how did you configure VMs to be able to run containers?
   * How did you select and install the correct container?
   * How did you verify that it was running correctly?
5. Identify Advantages/Disadvantages of the Solution
   * How would you have achieved the same thing without containers?
   * What are the advantages to doing it without containers?
   * What are the disadvantages?

**Question 4: Cloud Infrastructure as Code**

What are the security benefits of defining cloud infrastructure as code?

1. Restate the Problem
2. Provide a Concrete Example Scenario
   * In Project 1, when did you use infrastructure as code (IaC)?
   * What tool did you use?
   * What did you use it to do?
3. Explain the Solution Requirements
   * Were there any alternatives to IaC?
   * What benefits does IaC have over alternative approaches?
4. Explain the Solution Details
   * In Project 1, which specific configurations did your IaC set up?
   * How did you run and test these configurations?
5. Identify Advantages/Disadvantages of the Solution
   * Are there any disadvantages to using IaC over the "traditional" approach?

#### Domain: Logging and Monitoring

**Question 1: Setting Alerts in a New Monitoring System**

How do you determine which alerts to set in a new monitoring system?

Note: In Project 1, you did not set up any alerts. However, you still have enough experience to answer this question.

1. Restate the Problem
2. Provide a Concrete Example Scenario
   * Describe the network you built for Project 1. Identify the VMs on the network and what they do.
   * Which VMs should be publicly accessible?
   * Which VMs should not be publicly accessible?
3. Explain the Solution Requirements
   * Consider the VMs that should not be publicly accessible from the internet. Which alert(s) should these VMs fire and when?
   * Why should these VMs be associated with these alerts?
4. Explain the Solution Details
   * Which tool in Project 1 would you use to set such an alert?
   * What would the alert rule be? For example, would the alert fire upon a failed SSH attempt or a ping request?
5. Identify Advantages and Disadvantages
   * Are there any malicious circumstances that the alert(s) discussed above do not address?

**Question 2: Challenges of Collecting Large Amounts of Log Data**

What are the challenges of collecting huge amounts of log data? How do security analysts deal with them?

1. Restate the Problem
2. Provide a Concrete Example Scenario
   * In Project 1, when did you deal with log data?
   * What kind(s) of data did you investigate?
   * How much data were you dealing with?
   * What were you looking for?
3. Explain the Solution Requirements
   * What information did you need to find what you were looking for?
   * What does an analyst need to analyze large amounts of log data to find this information?
   * In Project 1, what tools did you use to analyze log data?
4. Explain the Solution Details
   * How did you use these tools to find the log data? E.g., which charts, graphs, etc. were useful for parsing the logs?
5. Identify Advantages and Disadvantages of the Solution
   * What kinds of data did you not inspect during Project 1?
   * Would having access to this additional data have changed your process or conclusions? If so, how?

**Question 3: Escalating Security Events**

How do you determine if a security event or alert is important enough for escalation?

1. Restate the Problem
2. Provide a Concrete Example Scenario
   * What kinds of events and alerts did you encounter in Project 1?
   * Which of these events was most interesting or suspicious?
   * Why was the event suspicious? What led you to investigate it?
3. Explain the Solution Requirements
   * What do you need to figure out in order to determine if this event is worth escalating?
4. Explain the Solution Details
   * How did you use Kibana to find this information?
5. Identify Advantages and Disadvantages of the Solution
   * How confident are you in your conclusion?
   * What additional data would be useful to determine if your conclusions are correct?

© 2020 Trilogy Education Services, a 2U, Inc. brand. All Rights Reserved.

## Activity File: Kibana Continued

* This week, you created the infrastructure behind a security information and event management system such as Kibana. Once that set up is complete, you will have finished the project.
* This optional activity tasks you with exploring more Kibana capabilities, some of which you will use in future projects.
* **Note**: In order to complete these activities, you will need to complete the optional Metricbeat configuration.

### Scenario

In this activity, you will suppose the role of a cloud architect that has been tasked with setting up an ELK server to gather logs for the Incident Response team.

Before you hand over the server to the IR team, your senior architect has asked that you verify the ELK server is working as expected and pulling both logs and metrics from the pen-testing web servers.

You will have three tasks:

1. Generate a high amount of failed SSH login attempts and verify that Kibana is picking up this activity.
2. Generate a high amount of CPU usage on the pen-testing machines and verify that Kibana picks up this data.
3. Generate a high amount of web requests to your pen-testing servers and make sure that Kibana is picking them up.

These activities will guide you though generating some data to visualize in Kibana. Each of these activity will require the following high level steps:

1. Use your jump-box to attack your web machines in various ways.
2. Use a Linux utility to stress the system of a webVM directly.
3. Subsequently generate traffic and logs that Kibana will collect.
4. View that traffic in various ways inside Kibanna.

It's also worth noting that these activities comprise different job roles:

* Getting the infrastructure setup and maintaining it is the role of a security engineer or cloud architect.
* Using that infrastructure by creating dashboards and alerts fall under the security analyst role. It would be rare to have a position where you would be required to do both.

That said, now that we have Kibana setup and gathering data from three web servers, its worth learning how to visualize data in Kibana.

Before getting started, we'll have to complete some metrics and logs set up.

### Setup: Kibana Metrics and Logs Orientation

Before we begin generating traffic, locate the two screens inside Kibana that you will use to visualize this traffic:

* Logs
* Metrics

These pages will show you the changes in data that we will create.

#### Logs

* Click **Logs** to see some general system logs coming from the web machines.
* Notice that you can stream logs live from the machines.

#### Metrics

* Next, click **Metrics** on the left side.
  + Here we can see each of our VMs that are sending metrics.
* Click on one of the large squares that represent one of your VMs.
* Choose **View metrics** from the dropdown that appears.
* Notice that you can see CPU and memory usage here.

Now that we know where to look for this data, let's generate some unusual network traffic.

### Activity Tasks

Expand the provided activity files to complete each task. These tasks can be completed in any order.

#### SSH Barrage

Task: Generate a high amount of failed SSH login attempts and verify that Kibana is picking up this activity.

 Activity File: SSH Barrage

#### Linux Stress

Task: Generate a high amount of CPU usage on the pentesting machines and verify that Kibana picks up this data.

 Activity File: Linux Stress

#### wget-DoS

Task: Generate a high amount of web requests to your pen-testing servers and make sure that Kibana is picking them up.

 Activity File: wget-DoS

© 2020 Trilogy Education Services, a 2U, Inc. brand. All Rights Reserved.

**Solution Guide: Kibana Continued**

**Scenario**

In this activity, you played the role of a cloud architect and were tasked with setting up an ELK server to gather logs for the Incident Response team.

Before you hand over the server to the IR team, your senior architect has asked that you verify the ELK server is working as expected and pulling both logs and metrics from the pen-testing web servers.

You had three tasks:

1. Generate a high amount of failed SSH login attempts and verify that Kibana is picking up this activity.
2. Generate a high amount of CPU usage on the pen-testing machines and verify that Kibana picks up this data.
3. Generate a high amount of web requests to your pen-testing servers and make sure that Kibana is picking them up.

**SSH Barrage**

Task: Generate a high amount of failed SSH login attempts and verify that Kibana is picking up this activity.

 Solution Guide: SSH Barrage

**Linux Stress**

Task: Generate a high amount of CPU usage on the pen-testing machines and verify that Kibana picks up this data.

 Solution Guide: Linux Stress

**wget-DoS**

Task: Generate a high amount of web requests to your pen-testing servers and make sure that Kibana is picking them up.

 Solution Guide: wget-DoS

© 2020 Trilogy Education Services, a 2U, Inc. brand. All Rights Reserved.

**Homework: GitHub Fundamentals**

**Background**

To understand GitHub, you need to know the basics of **version control**.

* Version control is a system that allows users to save all versions of a file while working on it. It's like adding an undo function to any document or file. You create save points as you work, which you can revert to at any time.
* **Git** is the most popular software used for version control. It runs on your local computer and allows you create save points (known as **commits**) for your documents.
* You can use Git to manage any directory and track every item inside that directory. At any point, you can revert the Git directory (known as a **repository**) to a previous commit.

GitHub is a website that allows you to sync your local Git repository with a repository in the cloud. This allows you to save your work to the cloud, share your work with others, and easily collaborate on a project.

* Other users can access your online GitHub repository and sync their own changes. They can also make a copy of your repository to create an entirely new project based on your original project. This is known as **forking**.

In this activity, you will:

* Create a new, empty Github repository and sync it to your local machine.
* Once your repository is up, you will add all of your Ansible scripts, Bash scripts, and network diagrams to the repository, and sync it again with the cloud.
* When everything is synced, you will update the GitHub README file, which will explain each of the items in the repo, and display a network diagram. You will then have a GitHub repository to present to future employers.

You will also use your GitHub account for other activities in the course.

**Required Files**

* Ansible YAML scripts from the week on cloud security.
  + Gather all of your Ansible YAML scripts from your Ansible container on your jump box.
  + Copy and paste these into new documents on your local machine.
* Bash scripts from the week on Linux.
  + Gather all of your system configuration scripts you created during the weeks on Linux.
* Network diagrams.
  + Gather all of the network diagrams you created during the weeks on cloud security and networking.

**Your Goals**

1. Create a GitHub repository for all of your files.
2. Copy all of your files into the repository and create a README explaining the repository.

**Topics Covered in this Assignment**

The topics covered in this homework assignment are:

* Creating a new GitHub repository.
* Syncing a local repository.
* Creating a README file.
* [Markdown](https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet)
* The following commands:
  + git pull
  + git add
  + git commit -m
  + git push origin --set-upstream <-branch->

**Instructions**

1. Create your GitHub repository.
   * Go to [github.com](https://github.com/) and sign up for GitHub.
   * Confirm your email address.
   * Click **Create a Repository**.
   * Name your repository and give it a short description.
   * Check the box for **Initialize this repository with a README**.
   * Click **Create Repository**.
2. Download your repository.
   * Click the green **Clone or Download** button on the right side.
   * Copy the link.
   * Go to your command line and run the command: git clone https://github.com/your-username/yourlink.git
     + Enter your GitHub username and password to complete the download.
3. Once you have the repository downloaded, copy your scripts and diagrams into it.
   * Create folders for Linux, Ansible and Diagrams.
   * Copy your scripts and diagrams to the appropriate folder.
4. Sync your local and remote repositories.
   * In your terminal, make sure you're located in the top directory of your repo.
   * Run git add . to specify that you want to sync *all* the items and directories that you just added to your repo. This command stages your files for a commit.
   * Run the command git commit -m "First commit" to confirm the commit and add a note describing it ("First commit").
   * Run git push to finalize the sync.
   * Go to [github.com](http://github.com/) and confirm your content is there.
5. Add the README file you created during Day 3 of the project week.
   * Click on the README.md file in your GitHub repo.
   * Click on the small pencil that reads **Edit this file** on hover.
   * Copy and paste the README.md file you wrote during class.
   * Make any desired changes and click **Commit Changes** at the bottom of the screen.

**Note:** READMEs are written in Markdown. This [Markdown Cheatsheet](https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet) has more information about writing in Markdown.

Check your repo for any errors or typos. You now have a GitHub repository that is ready to present and share with the world.

© 2020 Trilogy Education Services, a 2U, Inc. brand. All Rights Reserved.