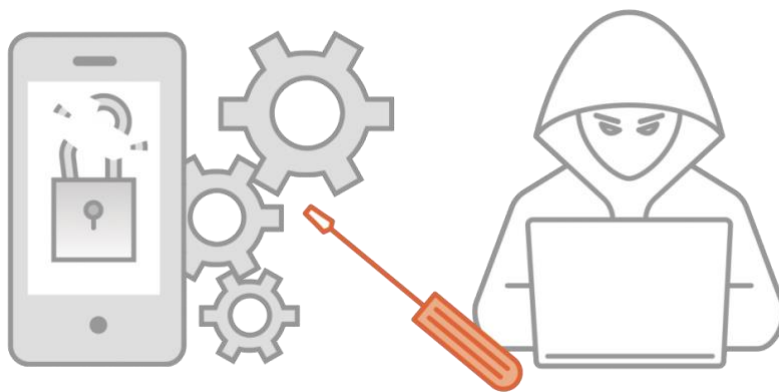


Developer Challenge



The information in this document, including possible attachment(s), is confidential and solely intended for the addressee. Publication, multiplication, distribution, use, or viewing of this document is only permitted if the addressee explicitly granted permission to do so.

ThreatFabric and ThreatFabric's logos are trademarks of ThreatFabric B.V. The trademarks, names, and illustrations of other organizations and products are the property of their respective owners. © ThreatFabric B.V.

1. Introduction

1.1 Description

The aim of this challenge is for both the ThreatFabric team and yourself to validate the typical type of tasks/projects that the successful candidate will perform in this job, through a practical exercise. In order to match the usual type of tasks to perform in this job, there are several exercises, each representing a different task type, based on the profile you applied for.

The exercise should allow you to get grasp of the common challenges that you will be facing in this job and simplifies the recruitment process by shortening it.

You **only need to perform one of the exercises**. The one that matches your profile and the position you applied for most.

The most important aspect of the exercise is the quality of the resulting code and/or documentation. Although time also plays an important role in our jobs, we do understand that your actual job already keeps you busy, meaning that it can take some time for you to get this exercise done and therefore that we won't judge the time but only the result.

2. Challenges

There are several different exercises, detailed below. **You only need to perform one.** As part of the job the successful candidate will have to be able to do both. Depending on your knowledge of the subject it should take you a few hours to complete an exercise. Select the one you are the most skilled/comfortable with to ensure you will also be able to explain the result during the exercise follow-up meeting.

2.1 Challenge 1 – Android manifest analyzer app

This is a development exercise; the goal of the exercise is to build an Android application (Java) that retrieves the Android manifest of another application installed on the same device (application of your choice). Running the manifest analyzer should result in having the content of the manifest file available in a human readable format. This action should be performed without using the Android API but finding the manifest file and parsing it yourself instead. You can find more information about Android application manifest in the official documentation: <https://developer.android.com/guide/topics/manifest/manifest-intro>

Please build a simple application that can perform the task and write short documentation (user interface isn't important for us, just the process details and the steps followed by your code):

- Describe steps followed by the code to fulfill the task
- Detail how someone can make use of the application

2.2 Challenge 2 – Architecture design Cloud resource manager

This is an architecture exercise; the goal of the exercise is to design (properly document) one or more solutions to edit static resources (based on a regularly changing configuration) served by AWS CloudFront from an origin like AWS S3. The configuration determines the name of the file to be edited and the changes to be made to it. A file cannot be edited directly in the S3

bucket it is served from and must be edited before leaving CloudFront to be sent to the web client.

There are multiple ways to perform this technical task, as we also haven't provided many details to go on, so feel free to make some assumptions.

More documentation about respective mentioned services is available here:

<https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/Introduction.htm>

!

<https://docs.aws.amazon.com/AmazonS3/latest/dev/Welcome.html>

Please write clear documentation about design and implementation of the solution: ▪

Describe on a high-level what are the possible ways to fulfill the task:

- Describe steps followed by your favored solution to fulfill the task.
- Use pictures and diagrams to complement the description.
- Explain strengths/weaknesses compared to other possible solutions.
- Estimate the development work required to build the solution.

2.3 Challenge 3 – User interface conception and design

This is an interface designing exercise; the goal of the exercise is to build an interface (with the framework of your choice, ex: Angular, React) based on the ThreatFabric house style, as per the screenshots and color pallet [available in the appendix](#). Be creative and think of what is the most relevant/useful for the end-user based on your own knowledge and thoughts (there are no wrong answers).

Please create some interface drafts that can fulfill following:

- 1) Interface1 - List of devices connected to the system (Android / iOS / PC):
 - The list contains some generic device information, such as “device id”, “device type”, “device risk level/score” and “last seen timestamp”
 - There is an option to search for devices based on parameters like the device id, device type, risk level/score and last seen timestamp
 - Devices can be sorted on different fields
 - Clicking on a device (or some button for this device) will navigate to the device details page
- 2) Interface 2 - Dashboard:
 - The dashboard contains charts with information about different statistics detected on a daily basis (number of infected devices, number of rooted devices, number of devices with abnormal security settings).
 - The dashboard shows detections of infected devices, per device type.
 - The dashboard contains date range selection options to define data boundaries
- 3) Interface 3 - App details:
 - The interface shows the details of a specific Android application (it's like when you select one item from a whole list).
 - Example of info shown: “installation timestamp”, “last update timestamp”, “installation file”, hash”, “app name”, “app version”, “app permissions”, “app signing certificate”

3. Report

Once you performed the exercise, please reply to the email via which you received the exercise details, attaching the resulting file(s).

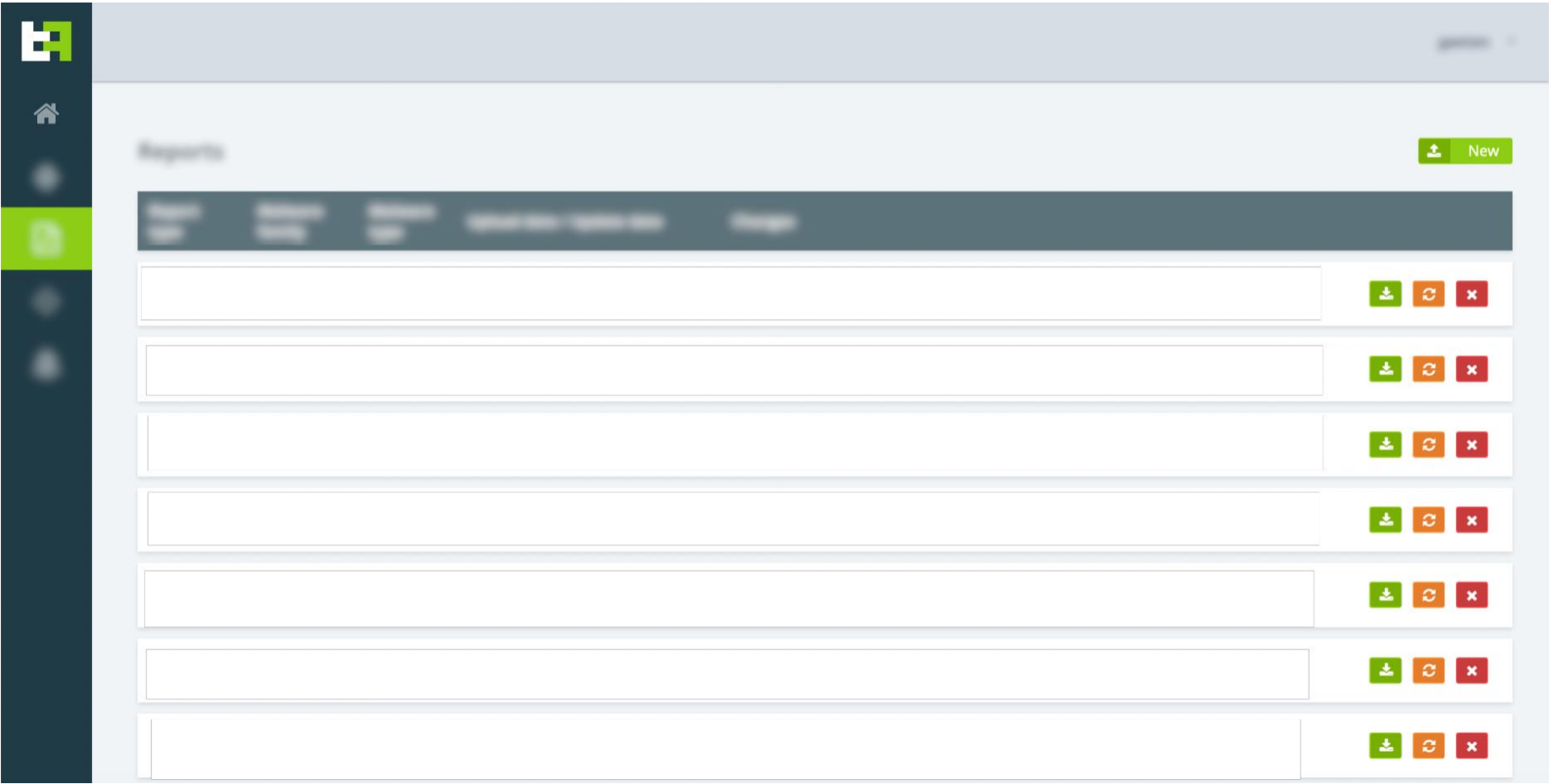
We will acknowledge reception and will analyze the result of your work, after which we will contact you for follow-up.

4. Appendix

ThreatFabric portal template 1:



ThreatFabric portal template 2:



ThreatFabric color pallet:

# 233C45	9ACA3C	D8BA28	F89A1C	F15722	B27E9F	62ACC9	4563AD
C 85	45	18	0	0	32	60	81
M 63	0	23	47	81	57	17	66
Y 54	100	100	100	100	18	14	0
K 47	0	0	0	0	0	0	0
# 6D7F85	BDDA7B	E5D26D	FABC57	F68F59	CBA9C0	98C8DB	7288BF
C 61	29	11	0	0	19	39	59
M 42	0	12	28	54	35	8	43
Y 41	68	70	76	70	10	9	0
K 7	0	0	0	0	0	0	0
# B6BEC1	DDEBBD	F2E8B5	FEDDAB	FCC6AC	E5D4DF	CCE4ED	A1AFD2
C 29	14	5	0	0	8	19	37
M 19	0	5	13	26	16	2	25
Y 20	32	34	36	30	4	4	0
K 0	0	0	0	0	0	0	0
# 536064	8DA659	AC9D51	BB8D41	BB6C42	997F90	7296A4	516389
C 68	49	34	25	21	42	59	76
M 52	20	31	43	64	51	31	61
Y 50	81	82	87	82	31	29	27
K 23	2	4	4	7	2	1	7
# 686C6E	7D866B	8A8468	907D61	8E7061	81777D	738186	52596A
C 60	52	46	42	41	51	58	71
M 50	36	39	44	52	49	41	60
Y 48	62	62	64	59	41	40	42
K 17	10	11	12	15	9	7	21