Standalone Client Documentation

Manuel Anderson

December 2019

Contents

1 Introduction **3**

2 How does it work? **3**

2.1 The client 4

2.2 Orchestrator 5

2.3 sampleFunction 5

1 Introduction

Here we will try to explain how our *standaloneClient* works behind the scenes. The objective of this program is to perform a file exfiltration in silent and hard-to-detect manner.

The focus of this project was to use AWS enviroment for exfiltration, specially the Lambda functions.

Generally, detection occurs when either the domains the data is being sent to are not trustworthy or the volumne of data being transfered in a single request is massive.

By using Lambda functions we will try to mitigate this issues.

2 How does it work?

The main idea of the algorithm is to perform a file division based on the size of the file we want to exfiltrate. This way, the data being sent on each subsequent request will be tiny compared to the size of the whole file. Therefore, the chance of detection based on request size will be lowered.

The file division is quite easy. We basically take a *partitionSize* provided by the user (bytes)and divide the main file into "n" parts.

Then, we create "n" lambda functions with our orchestrator and finally, we exfiltrate each part of the file with its corresponding lambda function to a S3 Bucket.

Lets talk a little bit about each component so we can better understand how this works.

2.1 The client

The client is in charge of doing the file partitions needed for the lambda invocations.

The following diagram represents how the process works in a general manner.

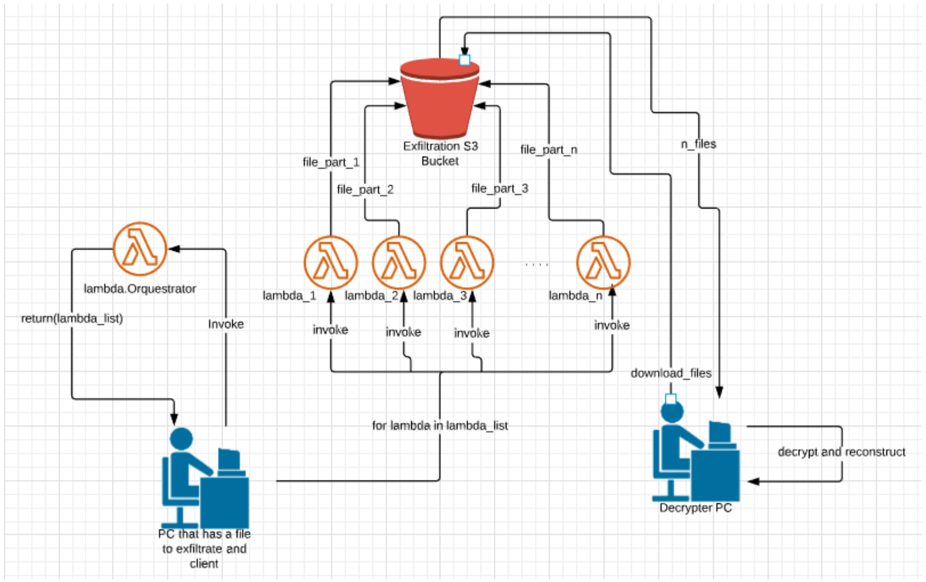

Figure 1: Standalone Client Diagram.

Lets explain what's going on.

1. Client is called from PC which has a file to exfiltrate
2. Client connects to AWS
3. Client divides the file into "n" parts, each parts weighs *partitionSize*
4. The "n" value is sent to the orchestrator function, which creates "n" lambda functions (One for each partition). Lambda names are formed by an UUID created randomly.
5. For each partition, the information gets encrypted and put into a payload, which is then sent to the lambda function assigned to that partition.
6. Each lambda function that is called uploads a file to our exfiltration S3 Bucket. Files follow the label HOSTNAME-FILENAME
7. After the upload, the lambda function is deleted.

One the exfiltration is completed, we can recover the file with the decrypter.

1. Decrypter is called
2. Decrypter downloads each file, which is then read and decrypted. When each part is decrypted, information is put into a file called ”reconstructedFile”

2.2 Orchestrator

The orchestrator is the one in charge of receiving a number and then creating that amount of lambda functions based on a template.

A list of the lambda functions is then returned to the client, those functions will be the ones to exfiltrate the data.

The template for the lambda functions is taken from the template bucket we create, in which we put our compressed template.

2.3 sampleFunction

This function is really basic, its only job is to take the payload it receives from the client with the fileName and put that into an S3 bucket.