

Hi,

Thanks for taking the time to work on this assignment!

This assignment is a close simulation of the work of a solution architect at Salt, and is meant to assess your engineering, communication and problem solving skills.

As part of this assignment we will also simulate a standard working process within a team, that will include 3 meetings:

1. The assignment will be presented to you (about 30 mins)
2. A follow up meeting will be scheduled, where you'll present a technical design, receive feedback and give an estimation for when you will complete the task (about 30-60 mins).
3. You'll present your design and code and discuss them. (about 30-60 mins)

Below you can find the assignment. Feel free to ask any questions you might have throughout the process

## Objective

By the end of this assignment, we will have a functional reverse proxy to <https://httpbin.org/> services where all of the proxied traffic is recorded and backed up on S3, in its unencrypted form.

## Reverse Proxy

Set up a reverse proxy server to <http://httpbin.org/> that meets the below requirements:

- The reverse proxy shall be deployed on an AWS t3.micro VM
- The reverse proxy shall be deployed on a standard linux distribution, Ubuntu, CentOS, etc
- The reverse proxy shall proxy any request to <http://httpbin.org/> and return the original response (valid or error) from the <http://httpbin.org/> server
- The upstream ([httpbin.org](http://httpbin.org/)) can be accessed over HTTP only as it's safe to assume the upstream (<http://httpbin.org/>) is on-premise in the customer's data center
- The reverse proxy shall have a public IP and is open to the internet
- The reverse proxy shall serve clients on HTTPS, and can use a self-signed certificate
- A client that access the reverse proxy server via HTTP will be redirected to the equivalent HTTPS interface

To avoid consuming the VM's resources we would like the traffic recording to be performed on a separated and dedicated VM - Salt Security's Sensor VM. Therefore, we would like to use [AWS Traffic Mirroring](#) capabilities to set up a tapping VM.

## Traffic Recording

Set up a tapping and recording VM that meets the below requirements:

- The Sensor shall be deployed on an AWS t3.micro VM
- The Sensor shall record both request and response
- The Sensor shall upload the captured traffic to a dedicated S3 bucket
- The recorded traffic shall be stored in pcap format, and it shall be possible to view it by standard pcap utilities
- The recorded traffic shall include the clients sessions on its unencrypted form, meaning that the SSL/TLS encryption must somehow unfold, so it shall be possible to view the original client's request + the original upstream's response as plain text.
- The Sensor must constantly upload requests mirrored by it, without manual triggering on the sensor machine

## Acceptance Criteria

The solution must meet the above requirements as well as tested on the below criteria:

- As a user I would be able to execute requests to <https://httpbin.org/> via the reverse proxy
- As a user I would be able to download a recorded session and view it on a pcap client (wireshark, tcpdump, etc) - unencrypted
- As a user I would be able to trigger a request and view the recorded traffic no more than 2 minutes after the request triggering

## Deliverables

To conclude this project the below deliverables shall be provided:

For the design review meeting:

- A diagram of the suggested solution, explaining the flows, used technologies, etc

Before the final meeting (presentation of the implemented solution):

- A step-by-step tutorial describing the tools and commands used to create the solution's infrastructure and setup
- The URL/IP address of the reverse proxy server
- A postman collection with example requests to the reverse proxy server

During the final meeting - you will be asked to walk us through your solution and we will discuss the architecture, decisions and choices made, and what is needed to fully productize such a solution and deliver it to customers.

## General guidelines:

- You can use any tool and online resource you want, and reach out to us for consultation - as you would if you were working on this task as a solution architect at Salt Security :-)
- The implementation should be clean, optimal, readable and extensible.
- However, to avoid exceeding the expected time, it is possible to make tradeoffs to simplify the solution alongside explaining how a full production-grade solution would be implemented.

Good luck!