COMP 8003
Delivery Lab(6)
Report

Brandon Rada
A01345707
Feb 18th, 2026

Purpose

This lab is designed to make the Delivery phase concrete. You will experience how a prepared mechanism moves into an environment through trusted workflows, why that movement produces little signal, and why delivery is often misunderstood or noticed only in hindsight.

Requirements

Task	Status
This lab covers:	
Part 1 – Defender: map ingress paths	Fully implemented
Part 2 – Defender: Establish a baseline	Fully implemented
Part 3 – Delivery designer: prepare a benign mechanism	Fully implemented
Part 4 – Delivery: cross the boundary	Fully implemented
Part 5 – Defender: observe without hindsight	Fully implemented
Part 6 – Boundary analysis and reflection	Fully implemented

Platforms

This lab has been tested on:

- Kali Linux
- Windows 11

Language

- html

Documents

- Design
- Testing
- User Guide

Findings

Part 1 — Defender: map ingress paths

Ingress path 1: Email

Content entering:
- Attachments like PDFs, docs, images
- Links to external websites
- Plain-text messages

Identities/services handling it:
- External senders
- Email provider/handler servers

Where its logged:
- Email inbox

What “normal” looks like:
- Frequent daily traffic
- Internal and external senders
- Attachments like documents, schedules, and updates

Ingress path 2: Messaging services

Content entering:
- Shared files like PDFs, docs, images
- Links to external websites
- Chat messages

Identities/services handling it:
- Authenticated users
- Messaging services servers

Where its logged:
- Server logs of the chats (like downloads that are sent)

What “normal” looks like:
- Chatting throughout the day
- Files being shared to either a group of people or specific people
- Links to documents and resources.

Ingress path 3: Web based submission/portals

Content entering:
- Uploaded documents like PDFs, ZIP files, images

Identities/services handling it:
- Authenticates users
- Backend services

Where its logged:

- Submission logs
- Server logs

What “normal” looks like:

- Spike in uploads near typical submission times
- Submissions from authenticated accounts
- File types like PDF, ZIP, etc.

Ingress path 4: Web access

Content entering:

- Web pages loaded by users
- Downloaded files like PDFs, installers, images, files.

Identities/services handling it:

- User browsers
- Network firewall

Where its logged:

- DNS logs
- Firewall logs

What “normal” looks like:

- Regularly accessed websites, and downloads
- For the most part HTTPS traffic

Part 2 — Defender: Establish a baseline

Baseline Behaviour: Email Ingress Path

In this environment, email is used routinely throughout the day for communication. Messaging arrives at a steady pace with most email activity occurring during standard working hours. The majority of senders are internal users and automated systems and services. External services may also be common, for example for creating accounts like github or youtube.

Email attachments usually include PDFs, word documents, plain text messages, excel spreadsheets, images, ZIP files which are related to the work. Links are also common, like google drive or external resources. Emails are found in the user's inbox, with logs handled by the mail service servers.

Part 3 — Delivery designer: prepare a benign mechanism

Neutral Email Message:

Subject:
Shared Document Update
Body:

Hi,
I have added some new notes to the document we're keeping updated.
You can view it here when you get the chance:
https://brandonrada.github.io/COMP8003-lab6-website/

Let me know if anything needs to be changed.

thanks.

Benign Web Page:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Shared Document</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <style>
        body {
            margin: 0;
            font-family: Arial, Helvetica, sans-serif;
            background-color: #f1f3f4;
        }

        /* Top navigation bar */
        .topbar {
            background-color: #ffffff;
            padding: 12px 20px;
            box-shadow: 0 1px 3px rgba(0,0,0,0.1);
            display: flex;
            align-items: center;
            justify-content: space-between;
        }

        .logo {
            font-weight: bold;
            font-size: 18px;
            color: #202124;
        }
```

```css
.right-section {
    display: flex;
    align-items: center;
    gap: 15px;
}

.status {
    font-size: 14px;
    color: #5f6368;
}

.login-btn {
    padding: 6px 14px;
    border-radius: 4px;
    border: 1px solid #1a73e8;
    background-color: #1a73e8;
    color: white;
    font-size: 14px;
    cursor: default;
}

.profile-circle {
    width: 32px;
    height: 32px;
    border-radius: 50%;
    background-color: #5f6368;
    color: white;
    display: flex;
    align-items: center;
    justify-content: center;
    font-size: 14px;
    font-weight: bold;
}

/* Document container */
.document-wrapper {
    display: flex;
    justify-content: center;
```

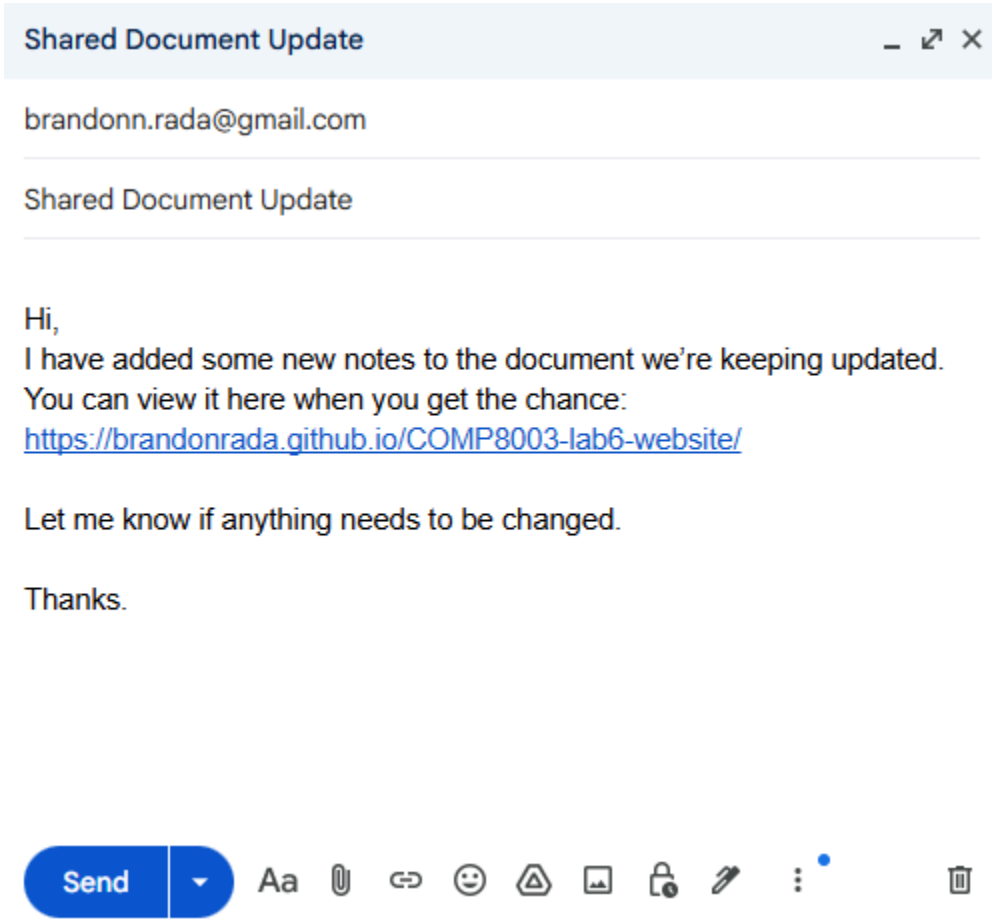
```
            padding: 40px 20px;
        }

        .document {
            background: #ffffff;
            width: 800px;
            min-height: 900px;
            padding: 60px;
            box-shadow: 0 2px 8px rgba(0,0,0,0.1);
            line-height: 1.6;
        }

        h1 {
            margin-top: 0;
            font-size: 26px;
        }

        .footer {
            text-align: center;
            padding: 20px;
            font-size: 12px;
            color: #80868b;
        }
    </style>
</head>
<body>

    <div class="topbar">
        <div class="logo">Shared Document Viewer</div>

        <div class="right-section">
            <div class="status">View Only</div>
            <button class="login-btn">Login</button>
            <div class="profile-circle">BR</div>
        </div>
    </div>

    <div class="document-wrapper">
        <div class="document">
```

```
            <h1>Project Notes</h1>

            <p><strong>Last Updated:</strong> February 18, 2026</p>

            <p>
                These notes summarize the latest updates to the project.
                Please review the following sections carefully.
            </p>

            <h3>Section 1: Overview</h3>
            <p>
                This document contains working notes and updates.
                All collaborators should review changes and suggest edits
                if necessary.
            </p>

            <h3>Section 2: Recent Changes</h3>
            <ul>
                <li>Added new milestone timeline</li>
                <li>Updated task assignments</li>
                <li>Clarified documentation requirements</li>
            </ul>

            <h3>Comments</h3>
            <p>
                Please reply to the original email if modifications are required.
            </p>
        </div>
    </div>

    <div class="footer">
        Document sharing service | Internal Use Only
    </div>

</body>
</html>
```

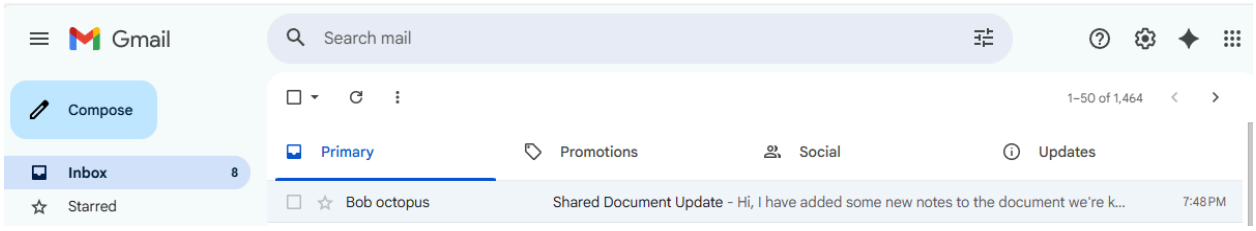
Part 4 — Delivery: cross the boundary

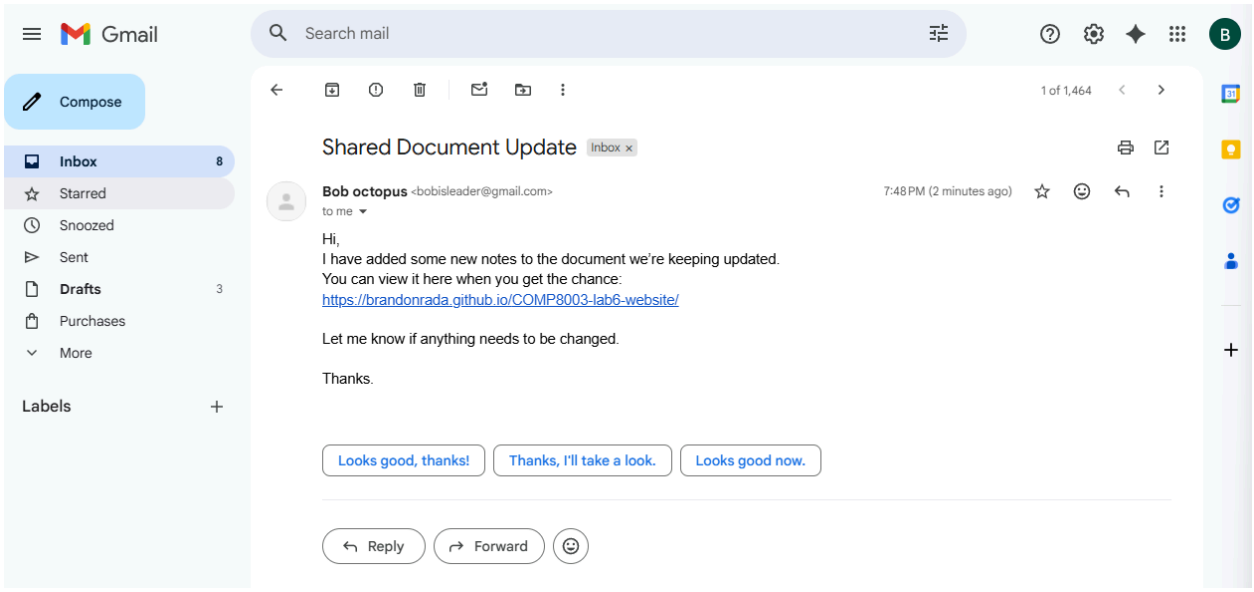
Sender:

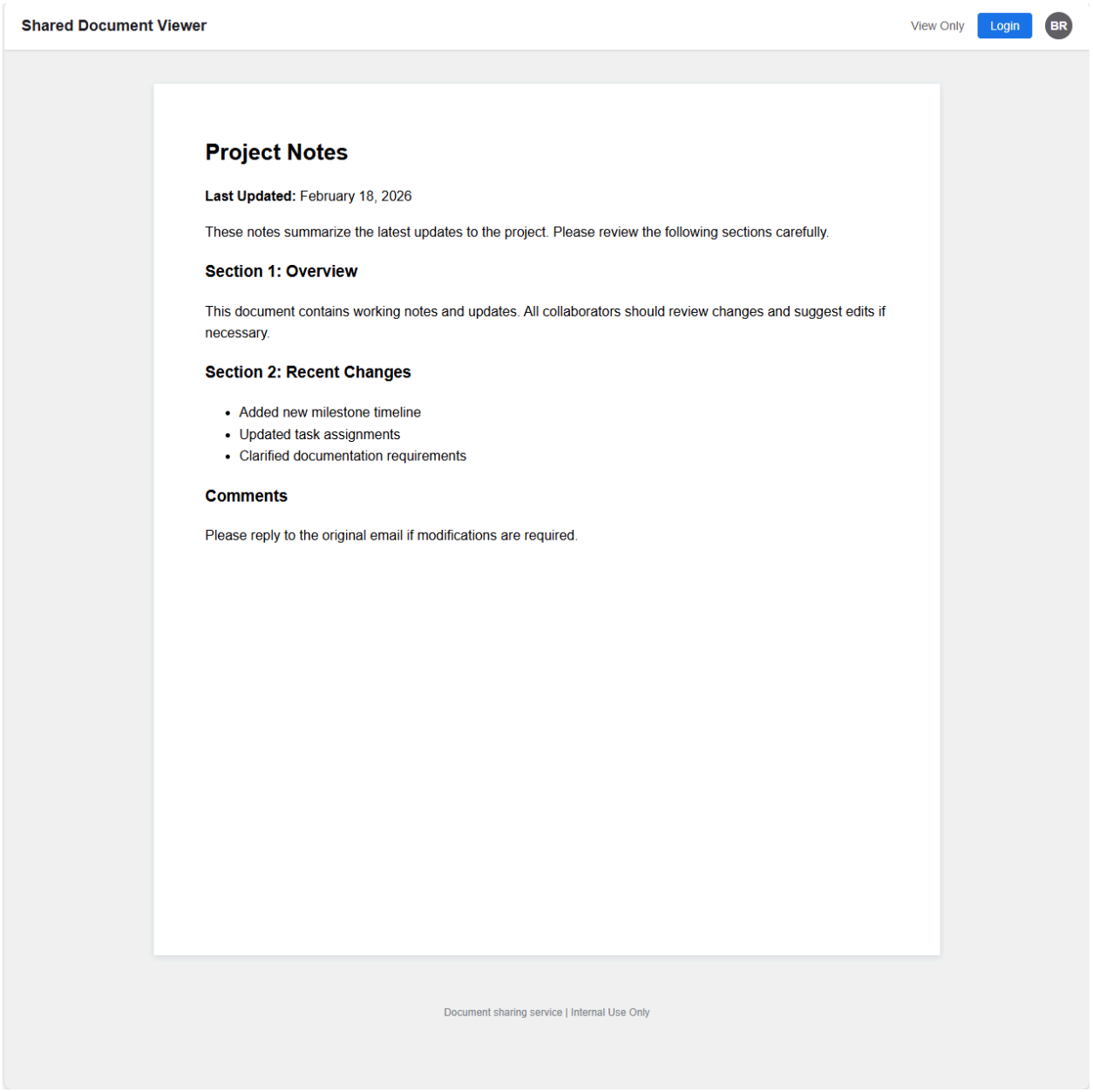
The email being sent:



Receiver:




Gmail
Search mail
Compose
Inbox 8
Starred
Snoozed
Sent
Drafts 3
Purchases
More
Labels
1 of 1,464
Shared Document Update
Inbox ×
Bob octopus <bobisleader@gmail.com>
to me
7:48 PM (2 minutes ago)
Hi,
I have added some new notes to the document we're keeping updated.
You can view it here when you get the chance:
https://brandonrada.github.io/COMP8003-lab6-website/
Let me know if anything needs to be changed.
Thanks.
Looks good, thanks!
Thanks, I'll take a look.
Looks good now.
Reply
Forward



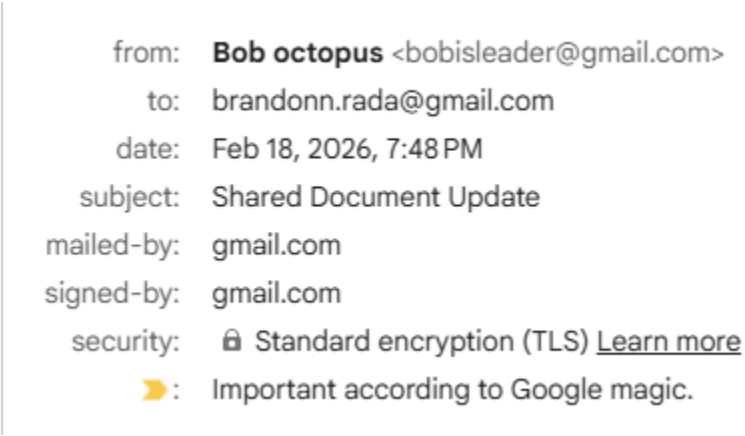
Part 5 — Defender: observe without hindsight

What Can Be Observed:

Email Delivery Logs:

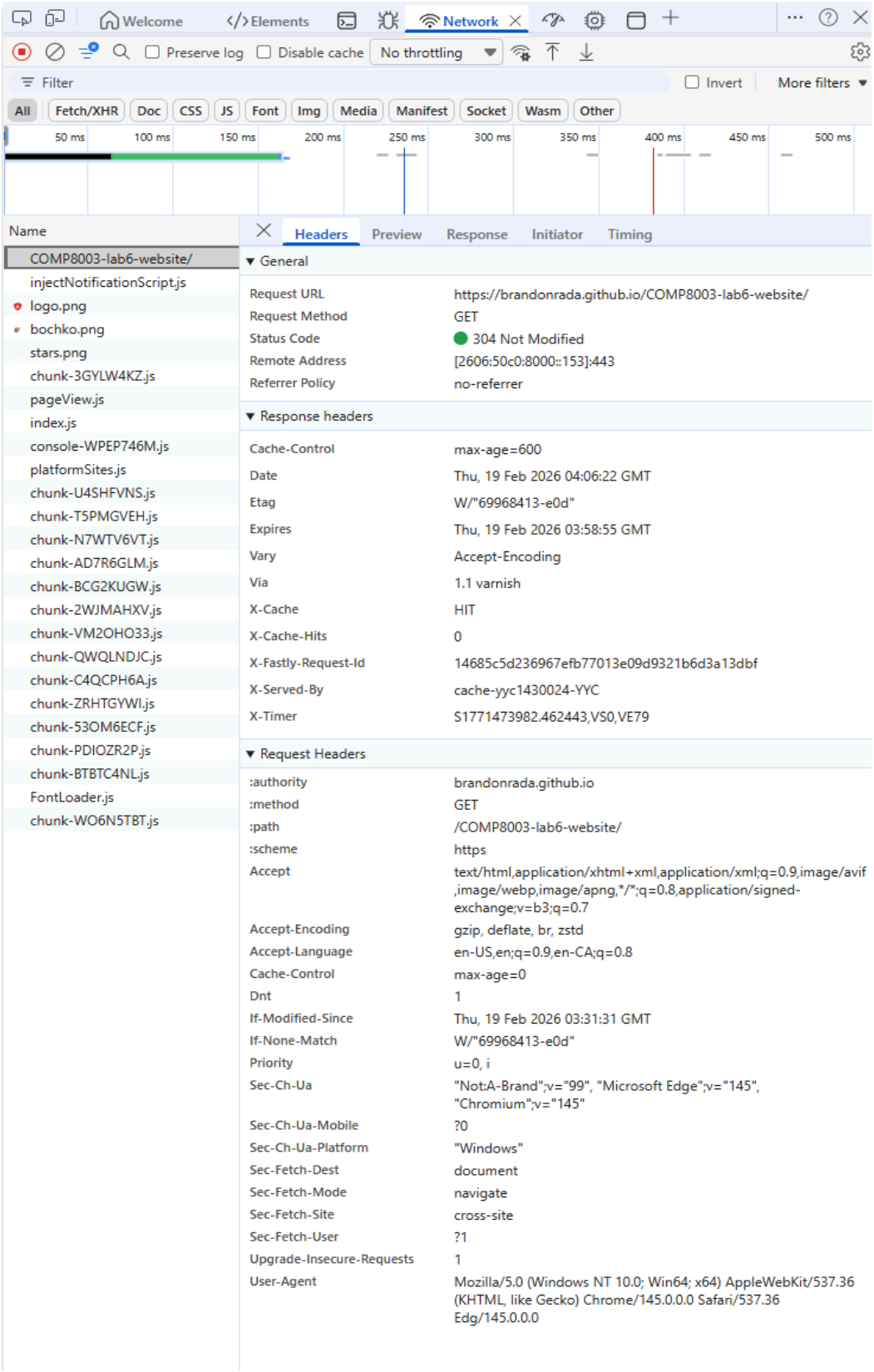
- An email titled “Shared Document Update” appears in the defenders email (brandonn.rada@gmail.com).
- The sender is bobisleader@gmail.com, which looks like a regular external email address.

- The message was delivered successfully with no warnings/flags.
- The content of the email is normal, it mentions updates to shared notes and contains a link to a document viewer



- The headers show standard routing through gmail servers.

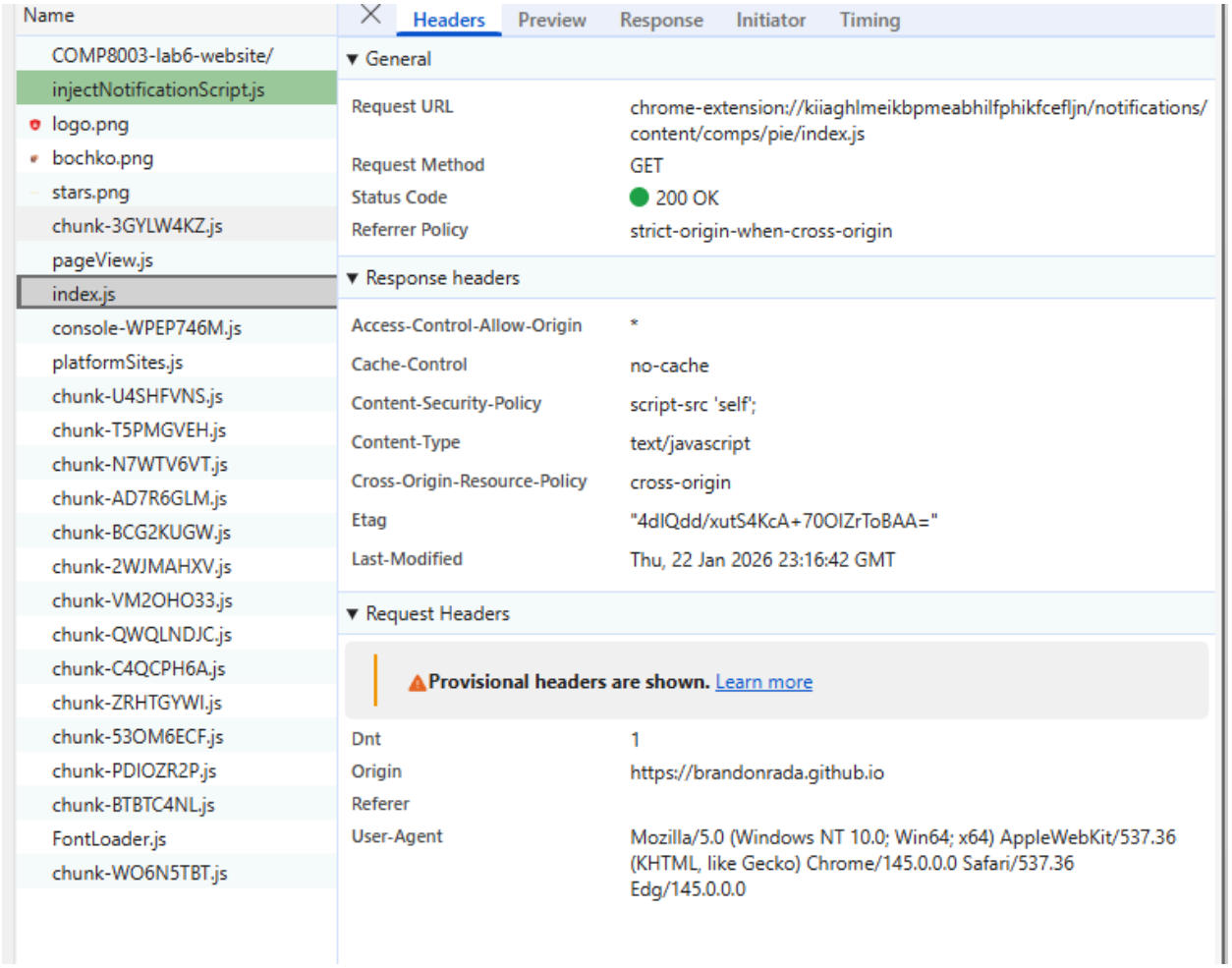
DNS/Network Logs:



- The link leading to a hosted website resolves normally and follows a HTTPS connection

- No redirects, scripts, or outbound connections are made.

Web Server Access Logs:



- A Get request for the main HTML file is logged.
- Nothing other than the static page is requested.
- The page loads successfully with a 200 OK response

What Cannot Be Observed:

There is no evidence to suggest any code execution, since the page is static and contains no scripts. No evidence of credential collection or authentication attempts.
There is no visible reasoning for why the sender (bobisleader@gmail.com) has created and sent the email to the user (brandonn.rada@gmail.com).

Part 6 — Boundary analysis and reflection

What boundary was crossed? :

The prepared external object (the email and link to a website) crossed from outside the environment into an internal trusted communication channel. The boundary was crossed through the email, where an external message was accepted into a user's internal environment.

Which trust assumptions were used? :

The delivery used many trust assumptions:

- That the environment trusts that email is a legitimate and allowed way of communication.
- External senders are able to deliver messages.
- Links to external websites are normal in communication.
- Internal users click on links to access external resources through email.

What made this delivery look routine?:

The delivery looked routine as it was short, neutral, and aligned with what may be normal document sharing experienced by the user.
The sender's address may also look normal, with the link going to a HTTPS website.

What evidence exists that delivery occurred? :

It is possible to see that the delivery occurred as the email appears in the user's inbox and network logs show that there was a GET request for the html file.

What evidence does not exist yet? :

Since it did not seem that any actions were taken other than the delivery, we don't see any evidence of credential collection or authentication attempts. There is also no indication that this delivery will be used later or a part of something bigger.

What would have to happen next for this to become exploitation? :

For exploitation to occur, something would have to utilize the delivered mechanism. For example, having the user interact with a login box, executing scripts, or triggering some known vulnerability.