Security Incident Report (Yummyrecipesforme.com)

Section 1: Identify the network protocol involved in the incident
The protocol at the center of this incident is **HTTP (Hypertext Transfer Protocol)**. Because the issue was tied to accessing the web server for *yummyrecipesforme.com*, it immediately pointed us toward web traffic. Requests to web servers for web pages typically run over HTTP, and our tcpdump analysis confirmed this. When we visited the website during testing, the tcpdump logs clearly showed HTTP traffic being generated. More importantly, the malicious file was delivered to users' computers through the HTTP protocol at the application layer, which explains how the attack was carried out.

Section 2: Document the incident
Several customers reached out to the website's helpdesk with the same alarming complaint: when they visited *yummyrecipesforme.com*, they were prompted to download a file that supposedly contained "new recipes." After running it, their personal computers began to slow down noticeably. At the same time, the website owner attempted to log in to the server but found themselves locked out of their administrator account. To investigate safely, the cybersecurity analyst set up a sandbox environment, isolated from the company's main network, and accessed the website. Using **tcpdump**, the analyst captured the network traffic generated while interacting with the site. Just as customers reported, the analyst was prompted to download a file that promised access to free recipes. After downloading and running the file, the browser redirected to a suspicious site: *greatrecipesforme.com*. When the tcpdump logs were reviewed, they confirmed the sequence of events. The browser first requested the IP address for

yummyrecipesforme.com and established a connection using the HTTP protocol. Shortly after the file was executed, the logs showed a sudden shift, the browser began requesting the IP address for *greatrecipesforme.com*, and all subsequent traffic was rerouted there.

A senior cybersecurity professional took the analysis further by inspecting both the website's source code and the downloaded file. They discovered that attackers had injected malicious code into the site, designed to trick visitors into downloading what looked like a browser update. Since the website owner was locked out of their admin account, the team concluded that the attacker likely used a brute force attack to gain unauthorized access and change the admin password. Ultimately, running the malicious file compromised the customers' personal computers.

Section 3: Recommend one remediation for brute force attacks
One strong defense is to put **two-factor authentication (2FA)** in place. With 2FA, a user needs more than just their password to log in, they also have to confirm a one-time passcode (OTP) that's sent to their email or phone. Only after entering both their login credentials and the OTP can they access the system. This extra step makes it extremely difficult for attackers using brute force methods to break in, since a stolen or guessed password alone won't be enough.