# Network protocol involved in the incident

The protocol involved in this incident is the **Hypertext Transfer Protocol (HTTP)**. Since the problem occurred when users tried to access the web server for *yummyrecipesforme.com*, it shows that the requests for the web pages were using HTTP traffic. In addition, when we ran **tcpdump** and visited the website, the tcpdump log confirmed that the **HTTP** protocol was being used during the connection. The malicious file was also seen being transferred to the users' computers through HTTP at the application layer.

# Incident Documentation

Several customers contacted the helpdesk and reported that when they visited the website, they were asked to download and run a file that claimed to give them access to new recipes. After doing this, their personal computers began to run slowly. The website owner also tried to log into the web server but found that they were locked out of their account.

To safely investigate the issue, the cybersecurity analyst used a sandbox environment to access the website without risking the company network. The analyst then ran **tcpdump** to capture the network traffic packets while interacting with the website. During the visit, the analyst was also prompted to download a file offering free recipes. After downloading and running the file, the browser redirected them to a fake website, *greatrecipesforme.com*.

The analyst reviewed the tcpdump log and saw that the browser first requested the IP address for *yummyrecipesforme.com*. Once the connection over the **HTTP** protocol was established, the analyst downloaded and executed the file. The logs then showed an unexpected change in network traffic as the browser requested a new IP address for the *greatrecipesforme.com* URL. At that point, network traffic began routing to the new website.

A senior cybersecurity professional analyzed the source code of both websites and the downloaded file. They found that an attacker had modified the website to add code that forced users to download a malicious file disguised as a browser update. Since the website owner reported being locked out of the admin account, the team believes that the attacker used a **brute force attack** to access the account and change the password. Running the malicious file led to the compromise of the end users' computers.

# Recommendations

One security measure the team plans to introduce to protect against brute force attacks is to prevent the reuse of previous passwords. Since the attacker was able to take advantage of a default password, it is important to make sure that old or default passwords cannot be used again for resetting an account. Another measure is to require users to update their passwords more frequently. This reduces the chance that an unauthorized person can continue using a known password for a long time.

Finally, a strong additional protection is to implement **two-factor authentication (2FA)**. With 2FA, users must log in using their password and also confirm a **one-time passcode (OTP)** sent to their email or phone. After entering their credentials and the OTP, they gain access. This makes it much harder for a malicious actor performing a brute force attack to succeed, because they would also need the second form of authentication.