Security incident report

Section 1: Identify the network protocol involved in the incident
The protocol involved in the incident is the **Hypertext Transfer Protocol (HTTP)**. Because the issue was related to accessing the web server for `yummyrecipesforme.com`, it is clear that requests for webpages were sent over HTTP traffic. The tcpdump log confirms this, showing communication between the browser and the server over port 80, which is the standard port for HTTP. The malicious file was delivered to users' systems through this same HTTP connection, which operates at the application layer of the TCP/IP model. The tcpdump log shows that the browser first requested the IP address of `yummyrecipesforme.com` through a **DNS query**, and the DNS server replied with the correct IP address. Once the browser established a connection over HTTP, it received the malicious prompt to download a file. After the file was run, the browser sent another DNS request for `greatrecipesforme.com`, and traffic was redirected to that malicious site over HTTP as well. This confirms that HTTP was the primary protocol involved in delivering the attack.

Section 2: Document the incident
Several customers contacted the company's helpdesk to report that when visiting `yummyrecipesforme.com`, they were prompted to download and run a file claiming to provide access to free recipes. After running the file, their browsers redirected to a different website and their personal computers began to run more slowly. The website owner attempted to log in to the admin panel but was unable to access the account, indicating that the administrator credentials had been changed. A cybersecurity analyst recreated the incident in a sandbox environment to safely investigate the issue. The analyst visited `yummyrecipesforme.com`, ran tcpdump to capture network activity, and observed the same prompt to

download and run a file. Once executed, the browser redirected to another website, `greatrecipesforme.com`, which contained the malware.

The tcpdump log confirmed these actions, showing a DNS request for `yummyrecipesforme.com`, followed by HTTP traffic delivering the malicious file. The logs then showed a second DNS request for `greatrecipesforme.com` and an HTTP connection to that domain. This sequence of events indicates that the malicious code embedded in the website redirected traffic after the user executed the downloaded file.

The senior cybersecurity analyst inspected the website's source code and the malicious file. The analysis confirmed that the attacker added JavaScript to the site to prompt visitors to download malware. The attacker accessed the admin panel through a **brute force attack**, successfully guessing the default password and changing it to lock out the legitimate website owner. The execution of the malicious file compromised visitors' systems and redirected them to a fake version of the website.

Section 3: Recommend one remediation for brute force attacks

A key security measure to prevent brute force attacks is implementing **multi-factor authentication (MFA)** for all administrative accounts. MFA adds another verification step, such as a one-time code sent to a trusted device or email, making it harder for attackers to gain access even if they guess the password.

Additional protective measures include enforcing **strong password policies** to ensure passwords are unique and difficult to guess, and implementing **account lockouts or rate limiting** after several failed login attempts. These steps would make brute force attacks significantly less effective and help prevent similar incidents in the future.