Security incident report

Section 1: Identify the network protocol involved in the incident
The incident revolves around the use of the Hypertext Transfer Protocol (HTTP). The problem occurred while attempting to reach the web server for yummyrecipesforme.com, which typically handles webpage requests via HTTP. When network traffic was captured using `tcpdump` during an attempt to access the site, the resulting logs clearly indicated HTTP communication. Additionally, the delivery of the malicious file to user systems took place through HTTP, confirming that it was transmitted at the application layer of the network stack.

Section 2: Document the incident
Several users reached out to the site's support team after noticing something unusual. When they visited the site, a pop-up encouraged them to download a file that supposedly gave access to new recipes. After running the file, their computers started slowing down significantly. Meanwhile, the website owner discovered they couldn't log into the server—someone had locked them out of their admin account. To investigate without risking the company's network, a cybersecurity analyst accessed the website in a secure sandbox environment. While browsing the site, the analyst captured network activity using `tcpdump`. During the visit, they encountered the same prompt to download a "free recipes" file. After running it, the browser was redirected to a suspicious site: *greatrecipesforme.com*. Reviewing the tcpdump logs, the analyst noted that the browser first connected to *yummyrecipesforme.com* using HTTP. After the file was downloaded and executed, the traffic shifted—suddenly, the browser started requesting data from a different domain, *greatrecipesforme.com*, and the connection was rerouted to its IP address.

A senior analyst then reviewed the site's code along with the downloaded file. It turned out that someone had injected malicious code into the original website, tricking visitors into downloading malware disguised as a browser update. Given that the site owner had lost access to their account, it's suspected the attacker used a brute-force method to gain control and change the admin password. Once users ran the file, their systems were compromised.

Section 3: Recommend one remediation for brute force attacks
To mitigate brute force attacks, one effective remediation is implementing account lockout policies—temporarily locking or throttling login attempts after a set number of failed tries. This makes automated attacks significantly harder and slows down attackers. Additionally, enforcing strong password requirements and enabling multi-factor authentication (MFA) add critical layers of defense. MFA ensures that even if a password is guessed or stolen, access is still blocked without a second verification method.