Security incident report

Section 1: Identify the network protocol involved in the incident
The protocol involved in this incident is the Hypertext Transfer Protocol (HTTP). Since the issue occurred when accessing the web server for yummyrecipesforme.com, it's evident that the traffic to the web server involved HTTP. Additionally, when we ran `tcpdump` and accessed the yummyrecipesforme.com website, the corresponding log file confirmed the use of the HTTP protocol during the connection. The malicious file was delivered to users' computers via HTTP at the application layer.

Section 2: Document the incident
Several customers of the website yummyrecipesforme.com contacted the website's helpdesk, reporting that when they attempt to access the website, they are prompted to download a file offering free recipes, and their computers subsequently become slower. The website owner also tried to log in to the admin panel but was unable to do so. As soon as our team was assigned this mission, we created a sandbox environment to observe the website's behavior without risking infection of the company's network. We then used `tcpdump` to capture the network traffic associated with the website. When we attempted to access the website, we were prompted to download a file claiming to provide free recipes. The browser then redirected us to a fake website called greatrecipesforme.com. By examining the `tcpdump` logs, we observed that the browser initially requested the IP address of the legitimate website, yummyrecipesforme.com. Once the connection was established over the HTTP protocol, we noted the download and execution of the file. The logs show that, following this, the browser unexpectedly initiated a DNS resolution request for the website greatrecipesforme.com, resulting in a redirection and a new connection over

HTTP.

A senior cybersecurity professional analyzed the source code of both the legitimate website and the downloaded file. The analyst discovered that the attacker had altered the website's source code, adding a JavaScript function that forced the download of the malicious file and redirected users to the fake website. Since the administrators are unable to access their accounts, the team believes the attacker used a brute force attack to gain access and then changed the account's password once logged in. The execution of the malicious file compromised the users' computers, which explains why they experienced performance slowdowns.

Section 3: Recommend one remediation for brute force attacks

To mitigate the risk of brute force attacks, our team recommends the following measures:

1. **Use Strong Passwords**: According to NIST password guidelines, user-generated passwords should be between 8 and 64 characters long. This length makes brute force attacks significantly more time-consuming and resource-intensive for the attacker.
2. **Enforce Multi-Factor Authentication (MFA)**: Even if an attacker manages to obtain a password, they would still be unable to access the website without additional authentication factors, such as a verification code or biometric data.
3. **Implement Delayed Authentication Responses**: Introducing a slight delay in the authentication process won't inconvenience regular users but will make brute force attacks more difficult by slowing down the attacker's attempts.