Security incident report

Section 1: Identify the network protocol involved in the incident
The protocol involved in the incident would be Hypertext Transfer Protocol (HTTP). By running tcpdump and accessing the yummyrecipesforme.com website to detect the problem, capture the protocol, and analyze the traffic activity in a DNS & HTTP traffic log file, it provided the evidence needed to come to this conclusion. The malicious file is observed being transported to the user's computers using the HTTP protocol at the application layer.

Section 2: Document the incident
Several customers contacted the website owner stating that when they visited the website, they were prompted to download and run a file that allowed them to access free recipes. Their personal computers have been operating slowly ever since. The website owner tried logging into the web server but noticed they were locked out of their account. The cybersecurity analyst used a sandbox environment to observe and test the website without impacting the company network. Then, the analyst ran network protocol analyzer tcpdump to capture the network and protocol traffic packets produced by interacting with the website. The analyst was prompted to download a file claiming it would update the user's browser, accepted the download and ran it. The browser then redirected the analyst to a fake/different website (greatrecipesforme.com) that looked identical to the original site (yummyrecipesforme.com). The cybersecurity analyst inspected the tcpdump log and observed that the browser initially requested the IP address for the yummyrecipesforme.com website. Once the connection with the website was established over the HTTP protocol, the analyst recalled downloading and executing the file. The logs showed a sudden change in network traffic as the browser requested a new IP resolution for the greatrecipesforme.com URL. The network traffic was then

rerouted to the new IP address for the greatrecipesforme.com website.

The senior cybersecurity professional analyzed the source code for the websites and the downloaded file and confirmed that the website was compromised. The analyst discovered that an attacker had added javascript code that prompted the users to download a malicious file disguised as a browser update. Analysis of the downloaded file found a script that redirects the visitors' browsers. Since the website owner stated that they had been locked out of their administrator account, the team believes the attacker used a brute force attack to access the account and change the admin password. The attacker was able to guess the password easily because the admin password was still set to default password and no controls were in place to prevent brute force attacks. The execution of the malicious file compromised the end users' computers.

Section 3: Recommend one remediation for brute force attacks
One security measure the team plans to implement to protect against brute force attacks is two-factor authentication (2FA). This 2FA plan will include an additional requirement for users to validate their identification by confirming a one-time password (OTP) sent to either their email or phone. Once the user confirms their identity through their login credentials and the OTP, they will gain access to the system. Any malicious actor that attempts a brute force attack will not likely gain access to the system because it requires additional authorization. Another security measure could be to limit login attempts. Brute force attacks rely on attempting multiple passwords and accounts. By restricting login attempts to a small amount per user, attackers won't be able to try more than a few passwords. A common way to restrict login attempts is to temporarily ban an IP from logging in after five failed login attempts, where subsequent attempts at a login will be blocked