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). Since the issue was with accessing the web server for yummyrecipesforme.com, we know that requests to web servers for web pages involve http traffic. Also, when we ran tcpdump and accessed the yummyrecipesforme.com website the corresponding tcpdump log file showed the usage of the http protocol when contacting the . The malicious file is observed being transported to the users' computers using the HTTP protocol at the application layer.

Section 2: Document the incident
The incident came into radar when multiple customers emailed yummyrecipesforme's helpdesk. They complained that the company's website had prompted them to download a file to access free recipes. The customers claimed that, after running the file, the address of the website changed and their personal computers began running more slowly. It was brought to light that the admin password had been tampered with when the website owner tried to login to the admin panel but was unable to do so. The security team mitigated the incident by running network protocol analyzer tcpdump inside a sandbox environment, then typing in the URL for the website, yummyrecipesforme.com. The result was the same as raised in complaint by the customers. The logs showed the following - 1. The browser initiates a DNS request: It requests the IP address of the yummyrecipesforme.com URL from the DNS server. 2. The DNS replies with the correct IP address. 3. The browser initiates an HTTP request: It requests the yummyrecipesforme.com webpage using the IP address sent by the DNS server. 4. The browser initiates the download of the malware. 5. The browser initiates a DNS request for greatrecipesforme.com. 6. The DNS server responds with the IP address for greatrecipesforme.com. 7. The browser initiates an HTTP request to the IP address for greatrecipesforme.com

On reviewing the source code it was found that a block of javascript code had been added to prompt website visitors to download an executable file. Analysis of the downloaded file found a script that redirects the visitors' browsers from yummyrecipesforme.com to greatrecipesforme.com.

Thus, it can be concluded that the web server was impacted by a brute force attack. The disgruntled hacker was able to guess the password easily because the admin password was still set to the default password. Additionally, there were no controls in place to prevent a brute force attack.

Section 3: Recommend one remediation for brute force attacks
Stricter password controls and usage of Multi Factor Authentication (MFA) could have prevented the threat actor from logging into the admit panel through brute force. Stricter password rules would ensure that the password is difficult to guess through brute force and limiting the number of login attempts would have set off an alert when the number of attempts exceeded the limit during bruteforce attack. MFA requires the user to prove their identity in more than one way. This would have also prevented the attacker from gaining access to the website source code.