Security incident report

Section 1: Identify the network protocol involved in the incident
The network protocol that was affected in this incident was Hypertext transfer protocol (HTTP). This protocol falls within the 4th layer of the TCP/IP model, the application layer. The application layer is responsible for creating or replying to network requests. The attacker changed the source code of the website to cause the HTTP protocol to redirect users to the unsafe site.

Section 2: Document the incident
A little while before 2:18 PM, numerous customers called and reported that when going to the website yummyrecipiesforme.com, the website prompts them to download a file to update their browser. When they downloaded it and ran it, the website URL changed, and their computers ran slowly. The website owner attempted to go to the admin panel of their website but was unable to do so. They also reported that all the recipes that the company sells are free on the new site. Testing this issue in a sandbox and using tcpdump, I went to the URL yummyrecipiesforme.com. The website prompted me to download an executable file to update my browser. The URL changed from yummyrecipesforme.com to greatrecipesforme.com, but did not update the browser. The fake website is identical, save for the recipes being free. The computer did start running slowly as the customers reported initially. Reviewing tcpdump's logs, my machine (initially using port 52444) first sent a DNS resolution request to dns.google.domain at 2:18 PM. Then the DNS server responded with the destination URL, 203.0.113.22. Then, my computer (using port 36086) was able to connect directly with yummyrecipesforme.http. The log shows "HTTP: GET / HTTP/1.1". At 2:20 PM, my machine makes another DNS resolution request, this time receiving a new destination URL, 192.0.2.172. This is when my computer (using port 56378) connects to greatrecipesforme.com.

My team's senior cybersecurity professional found that the admin password was not strong; It was the default password. This vulnerability allowed the attacker to successfully perform a brute force attack, gain access to the admin panel, and change the source code. They also changed the admin password to lock out the true admin user. It looks as though "HTTP: GET / HTTP/1.1" was the change that causes the website to prompt users to download an executable file to bring them to the unsafe website with the free recipes.

Section 3: Recommend one remediation for brute force attacks
To prevent an attack like this from happening again, our team will reassess the password policy. This will make it more difficult for future attackers to brute force a password. Implementing 2FA will mitigate risk, as the true admin will then be prompted to confirm signing in an extra way that will cause future attackers to have issues with verification. Additional things to implement is the monitoring of login attempts, and also having a limiter on how many times the user can enter in their password before locking them out. This will reduce the attack surface in the future.