By: caami.sec

Security incident report

Section 1: Identify the network protocol involved in the incident
The network protocol used to investigate the problem was **HTTP**. The incident involved accessing the web server for yummyrecipesforme.com, and requests to web servers for web pages use HTTP traffic. The tcpdump log showed HTTP protocol usage when contacting the website, specifically showing "HTTP: GET /" requests to both yummyrecipesforme.com and greatrecipesforme.com.

Section 2: Document the incident
Several customers contacted the website's helpdesk stating that when they visited yummyrecipesforme.com, they were prompted to download a file that contained access to new recipes. Their personal computers have been operating slowly ever since. The website owner tried logging into the admin panel but noticed they were locked out of their account. The cybersecurity analyst used a sandbox environment to investigate the website without impacting the company network. The analyst ran tcpdump to capture network traffic and observed the suspicious behavior. When accessing yummyrecipesforme.com, the analyst was prompted to download an executable file to update the browser. After downloading and running the file, the browser redirected to a different website (greatrecipesforme.com). The tcpdump log shows the browser initially requested the IP address for yummyrecipesforme.com through DNS. After establishing the HTTP connection, the malicious download occurred. Then the log shows a sudden change in network traffic as the browser requested a new IP address for greatrecipesforme.com and was redirected to this fake website. The investigation revealed that a former employee had used a brute force attack to gain access to the admin account using default passwords. After gaining access, they embedded malicious JavaScript code in the website that prompted users to download the malware file, which then redirected them to the fake website.

Section 3: Recommend one remediation for brute force attacks
Since the hacker successfully gained access to the web host through a brute force attack by guessing the default password, it's highly recommended to implement strong password policies with complex passwords and enable MFA to prevent unauthorized access to web servers. These policies will help prevent vulnerabilities by making passwords more difficult to guess through brute force attacks. Complex passwords should use special characters, numbers, and mixed cases, which make them significantly more difficult to guess. MFA requires an additional verification step before allowing login, requiring users to verify their identity using extra authentication methods such as fingerprint, face ID, SMS codes, or authenticator apps. This makes it much harder for attackers to gain unauthorized access even if they guess the password.