Security incident report

Section 1: Identify the network protocol involved in the incident

Two key network protocols were identified: DNS (Domain Name System) and HTTP (Hypertext Transfer Protocol).

DNS is crucial for resolving domain names into IP addresses. The logs indicate that when the user's browser requested the resolution of yummyrecipesforme.com, DNS provided the correct IP address. This process was repeated for the malicious site, greatrecipesforme.com, directing the user's browser to a different IP address. This redirection is a classic example of DNS manipulation, commonly used in phishing attacks.

HTTP, the protocol used for transmitting web pages on the internet, was also central to this incident. The logs show the browser initiating HTTP requests to both yummyrecipesforme.com and later to greatrecipesforme.com. The attacker leveraged HTTP to serve the malicious website and facilitate the unauthorized download of the malware.

Section 2: Document the incident

A group of customers reported to the owner of yummyrecipesforme.com that when they visited the site, they were asked to download and run a file, supposedly to update their browsers. After doing this, their computers started working slowly. The owner themselves couldn't log into the website's server, suggesting they were locked out of their admin account.

As a cybersecurity analyst, I set up a safe testing area, called a sandbox, to check the website without risking our company's network. I used a tool called tcpdump to record the data traffic when interacting with the site. While doing this, I too got a prompt to download a file claiming to update my browser. After downloading and running it, my browser was redirected to a fake version of our site, called greatrecipesforme.com, which looked just like ours.

Looking at the tcpdump log, I noticed that initially, the browser asked for the IP address of yummyrecipesforme.com. This connection was made using HTTP, a common web protocol. After running the downloaded file, the log showed that the browser asked for a new IP address, this time for greatrecipesforme.com. So, the data traffic was then sent to this new address.

A more experienced team member checked the coding of both our website and the downloaded file. It turns out someone had tampered with our site's code to trick users into downloading a harmful file, disguised as a browser update. Since our website owner was also locked out of the admin account, we think the attacker broke in by guessing the admin password, a technique known as a brute force attack. Running the harmful file on the users' computers caused the issues they experienced.

Section 3: Recommend one remediation for brute force attacks

To enhance security and prevent such brute force attacks in the future, I recommend implementing account lockout policies and multi-factor authentication (MFA). Account lockout policies would temporarily lock an account after several failed login attempts, hindering continuous password guessing attempts. MFA adds an extra security layer, requiring a second verification form, such as a code sent to a phone, even if the password is compromised. Additionally, regular password updates, rate limiting for login attempts, and vigilant monitoring of unusual login activities can further bolster our website's security.