Security incident report

Section 1: Identify the network protocol involved in the incident

The network protocols involved in the incident were **DNS (Domain Name System)** and **HTTP (Hypertext Transfer Protocol)**. DNS was used to resolve domain names (e.g., yummyrecipesforme.com and greatrecipesforme.com) to IP addresses, and HTTP was used to request and deliver web content, including the malicious file.

Section 2: Document the incident

On 6th April 2025, several customers contacted the helpdesk of **yummyrecipesforme.com**, reporting that the website prompted them to download a file in order to access free recipes.

Upon downloading and running the file, their browsers redirected to a different website; **greatrecipesforme.com -** and they noticed slower computer performance.

A cybersecurity investigation revealed that a **former employee** gained unauthorized access to the admin panel of the website by executing a **brute force attack**.

The attacker successfully guessed the default admin password and logged in. Once access was obtained, the attacker altered the website's source code by injecting a malicious JavaScript function that automatically prompted visitors to download an executable file.

The attacker also changed the admin password after modifying the site, locking out legitimate access by the website owner.

To safely test what was happening on the website, we created a secure testing area called a **sandbox environment**. While checking the website, we were able to find the below during the analysis:

- A DNS request was made for yummyrecipesforme.com which resolved to IP 203.0.113.22

(The computer first **asked a DNS server** for the IP address of yummyrecipesforme.com. The DNS server replied with the IP address 203.0.113.22)

- An HTTP GET request was made to retrieve the site's content.

(Then, the computer used **HTTP** (a web communication protocol) to **ask for the content** of the website)

- A download prompt appeared for a browser "update" executable.

(As soon as the site loaded, a message popped up asking the user to download a file that claimed to be a browser update)

- Upon execution, a second DNS request was made for greatrecipesforme.com, resolving to IP 192.0.2.17.

(After running that file, the computer sent another **DNS request**, this time asking for the IP address of a different site: greatrecipesforme.com. It got back the IP address 192.0.2.17)

- The browser was redirected to this fake site where malware was hosted.

(The browser was then **redirected to this new fake website**, which contained harmful software [malware]).

The **tcpdump traffic log** confirmed these steps, showing the DNS lookups and HTTP traffic between the client and the malicious sites.
The HTTP protocol was used for both the initial site access and the redirection to the malware-laden domain.
(We used a tool called **tcpdump** to check all this. It confirmed that DNS and HTTP were the two protocols used during this attack. DNS was used to find the website addresses, and HTTP was used to load the pages and download the dangerous file.)

This security incident was traced to poor password management practices and the lack of brute force protections on the web server.

Section 3: Recommend one remediation for brute force attacks

Recommended Security Measure:

Enforce **two-factor authentication (2FA)** on all administrative accounts.

Rationale: Two-factor authentication adds an extra layer of security by requiring not only a password but also a second form of verification, such as a temporary code sent to a mobile device. Even if an attacker guesses or obtains the correct password through brute force, they would still be unable to access the account without the second authentication factor. This drastically reduces the risk of unauthorized access, even when passwords are weak or reused.