Security incident report

Section 1: Documentation of the incident

How it was discovered:

The website's users emailed the helpdesk team alerting them for the fact that all the site's purchasable material was being displayed for free on its own web page. They also reported that whenever they clicked on the download link for the free material, they were being redirected to a similar web page with a completely different domain name, which displayed the reading material but caused slowness on the system.

How it worked:

The way which the malware was implemented used a redirecting system, inserted at source code as a java script segment. It forced the web browser to make a query from the original website's domain (`yummyrecipesforme.com`) to a different one (`greatrecipesforme.com`) with a new IP address, consequently establishing a connection with the malicious server. The new connection would then be authenticated through a TCP-3-way handshake and the contents of the insecure site were displayed using HTTP.

Section 2: The network protocol involved in the incident

The initial connection to the original website was done as any normal connection:

- The web browser would query the **DNS** server for an address resolution for the website (`yummyrecipesforme.com`);

- The server would then answer the query with the corresponding IP address (`203.0.113.22`) to the original website (`yummyrecipesforme.com`);

- The client (the user accessing the website through the web browser) would initiate a **TCP** connection with the server (the web application's host server), doing a TCP-3-way handshake;

- After the established and authenticated TCP connection, the client and the server would communicate through **HTTP**;

The malware's redirection system used a different domain and ip address, but the communication method was the same as the one done to the original website. Using the same protocols for the same purpose:

- **DNS**: Used for querying to resolve the new domain name (`greatrecipesforme.com`) to the unsafe IP address (`192.0.2.17`);

- **TCP**: Used to establish and authenticate, through the 3-way handshake method, the connections for data stream;

- **HTTP**: Used to display the web application's content onto the screen.

Section 3: Recommended remediations for the incident

The other security analysts informed that the way in which the threat actor gained access to the web server, so that he could insert the malicious code into the web application, was through a brute force attack at the administrative backdoor of the web host. The "*admin*" user was utilizing the default password provided by the host, which was weak and, consequently, insecure.

Possible future remediations to be implemented are the following:

- Change the administrative password to a stronger one. The company should also implement a password policy, using a lengthy, multi charactered passcode, making use of upper and lowercase letters, special characters, and numbers. This will make sure employees' accounts cannot be used to gain access to the server, avoiding future breaking in problems caused by brute force attacks.
- The company's IT department should also implement a MFA (Multi Factor Authentication) framework to the web application's server, hardening it against future threats. This implementation will make sure only allowed personnel have access to the web application's source code, which will avoid malicious code insertions and won't make the application rely on a simple username-password security measure.