Security incident report - Brute Force Attack

Section 1: Identify the network protocol involved in the incident
The protocol involved in the incident is the **Hypertext Transfer Protocol (HTTP)**. Since the issue was with accessing the web server for `yummyrecipesforme.com`, we know that web page requests involve HTTP traffic. Additionally, when the cybersecurity analyst ran `tcpdump` and accessed the website, the corresponding `tcpdump` log file showed HTTP traffic when contacting the server. The malicious file was observed being transported to users' computers via the HTTP protocol at the **application layer**. Other network protocols involved in the attack: • **Domain Name System (DNS)** – Used to resolve the IP addresses of `yummyrecipesforme.com` and `greatrecipesforme.com`. • **Transmission Control Protocol (TCP)** – Used to establish the connection between the client and web server.

Section 2: Document the incident
Incident Summary Several customers contacted the website's helpdesk stating that when they visited `yummyrecipesforme.com`, they were prompted to download and run a file that contained access to new recipes. After running the file, their personal computers began operating **slowly**, and they noticed the website address had changed.

The website owner attempted to log in to the web server but discovered they were **locked out** of their admin account.

Investigation & Findings

A cybersecurity analyst conducted an investigation using a **sandbox environment** to safely interact with the website without impacting the company network.

Steps Taken by the Analyst:

1. Opened `yummyrecipesforme.com` in a sandbox environment.
2. Ran `tcpdump` to capture network traffic packets.
3. Observed a prompt to download a file labeled as a **browser update**.
4. Downloaded and executed the file.
5. Noticed the browser redirected to **`greatrecipesforme.com`**, a fake version of the original website.

Network Log Analysis:

- The **browser initially requested the IP address** for `yummyrecipesforme.com` from the DNS server.
- Once the **HTTP connection** was established, the analyst **downloaded and executed the file**.
- The logs showed **a sudden change in network traffic** as the browser requested a new **IP address for `greatrecipesforme.com`**.
- The network traffic was then **rerouted to `greatrecipesforme.com`**, exposing users to additional malware.

Root Cause Analysis:

- A **senior cybersecurity professional** analyzed the **website's source code** and the downloaded malware file.
- The analysis revealed that **malicious JavaScript code** was injected into `yummyrecipesforme.com`, prompting visitors to download the malware.
- The attacker **used a brute force attack** to guess the admin password and gain access to the website's **admin panel**.
- Once inside, they **changed the admin password** to lock out the website owner.
- The malware **redirected users to `greatrecipesforme.com`**, which contained more malicious content.

Impact on the Business & Users:

- **Customers' personal computers were compromised** after downloading the malicious file.
- **The website's credibility and trust were damaged** due to the unauthorized redirection.
- **The website owner lost control** of the admin panel, preventing them from resolving the issue immediately.

Section 3: Recommend remediation for brute force attacks

To prevent future brute force attacks, the cybersecurity team recommends implementing the following security measures:

1. Enforce Strong Password Policies

- Require **complex passwords** with at least 12 characters, including uppercase, lowercase, numbers, and special characters.
- **Prohibit the use of default passwords** and enforce **password history** policies to prevent password reuse.
- Implement **automatic password expiration** requiring users to update credentials regularly.

2. Implement Multi-Factor Authentication (MFA)

- Enable **two-factor authentication (2FA)** for all administrator accounts.
- Require **OTP (One-Time Passcode) verification** via email, SMS, or authentication apps (e.g., Google Authenticator).

3. Deploy Account Lockout & Rate Limiting

- Implement an **account lockout policy** after a set number of failed login attempts. Example:
    - **Lock account after 5 failed attempts** within **10 minutes**.
    - **Require admin intervention** or **temporary cooldown** before allowing retries.
- Use **rate-limiting mechanisms** to slow down login attempts and

prevent automated brute force attacks.

4. Monitor Web Traffic with Intrusion Detection Systems (IDS)

- Deploy **Intrusion Detection & Prevention Systems (IDS/IPS)** such as **Snort** or **Suricata** to detect suspicious login attempts.
- Configure real-time **alerts** for multiple failed login attempts from the same IP address.

5. Conduct Regular Security Audits & Updates

- **Change default passwords** immediately upon system deployment.
- Perform **regular penetration testing** to identify security weaknesses.
- Schedule **frequent security updates** for web applications and servers.
-
- **Review admin logs** for unauthorized access attempts.
- **Scan for malware and unauthorized script modifications** regularly.