**Security Incident Report: Compromise of Recipe Website and Malware Distribution**

### **Introduction**

This report details the investigation into a security incident involving the website *yummyrecipesforme.com*, which was compromised to distribute malware to visitors. The attack involved a combination of brute force credential theft, website defacement, and malicious file distribution—all conducted over unencrypted HTTP connections. Below is a comprehensive analysis of the incident, including the attack methodology, forensic evidence, and recommended security improvements.

### **1. Protocol and Traffic Analysis**

The attack leveraged the **HTTP protocol** to deliver malicious content and redirect users to a fraudulent domain (*greatrecipesforme.com*). Key observations from the network traffic logs include:

* **Initial DNS Query:**  
  The user’s machine first resolved *yummyrecipesforme.com* to 203.0.113.22 via a DNS request (14:18:32). This is standard behavior when accessing a website.
* **HTTP Session Establishment:**  
  A TCP three-way handshake (SYN, SYN-ACK, ACK) was completed, followed by an HTTP GET / request (14:18:36). This unencrypted connection allowed the attacker to inject a malicious download prompt.
* **Malicious Redirection:**  
  After the user downloaded and executed the file, a second DNS query was made for *greatrecipesforme.com* (14:20:32), resolving to 192.0.2.17. The browser then established a new HTTP session with this domain (14:25:29), indicating a possible command-and-control (C2) server.

**Why HTTP Was Exploited:**

* No encryption (unlike HTTPS), making traffic manipulation easier.
* Lack of integrity checks allowed the attacker to modify web content.
* The protocol’s simplicity facilitated malware delivery without detection.

### **2. Incident Timeline and Attack Chain**

#### **Initial Compromise**

* **Brute Force Attack:**  
  The attacker gained access to the website’s admin panel by systematically guessing weak credentials. Once inside, they altered the site’s code to inject a fake "Download Free Recipes" prompt.
* **Malware Deployment:**  
  When users visited the site, they were tricked into downloading a malicious executable disguised as a recipe file. Execution of this file led to:
  + System slowdowns (reported by users).
  + Unauthorized DNS redirection to *greatrecipesforme.com*.

#### **Post-Infection Activity**

* **C2 Communication:**  
  The malware established connections to the attacker’s server (*greatrecipesforme.com*), likely exfiltrating data or awaiting further instructions.
* **Persistence Mechanism:**  
  Forensic analysis suggested registry modifications to maintain access.

#### **Impact**

* **Users:** Compromised devices, potential data theft.
* **Website Owner:** Locked out of the admin panel, reputational damage.

### **3. Security Recommendations**

#### **Immediate Actions**

**-Disable HTTP:** Enforce HTTPS site-wide to prevent traffic tampering.  
 **-Password Reset:** Force all admin accounts to update credentials.  
 **-Malware Scanning:** Deploy endpoint detection on affected devices.

#### **Long-Term Defenses**

**-Multi-Factor Authentication (MFA):**

* Require TOTP or hardware keys for admin access.
* Prevents brute force attacks even if passwords are weak.

**-Rate-Limiting & Account Lockouts:**

* Lock accounts after 5 failed attempts.
* Implement progressive delays to slow brute force attempts.

**-Web Application Firewall (WAF):**

* Block SQLi, XSS, and unauthorized file uploads.
* Alert on suspicious DNS redirections.

**-User Awareness Training:**

* Educate visitors on risks of downloading unexpected files.
* Teach staff to recognize phishing and social engineering.

### **Conclusion**

This attack exploited weak authentication, unencrypted HTTP, and user trust to distribute malware. By implementing MFA, enforcing HTTPS, and deploying a WAF, future incidents can be mitigated. Additionally, continuous monitoring for unusual DNS requests will help detect similar redirection attacks early.

**Next Steps:**

* Conduct a full security audit of the web server.
* Monitor for further C2 traffic from infected devices.
* Notify affected users and recommend malware scans.

This version maintains technical accuracy while presenting the findings in a more narrative, less bullet-point-heavy format. It flows like an investigative report rather than a checklist, making it more engaging for stakeholders. Let me know if you'd like any refinements!