# Security incident report

| **Section 1: Identify the network protocol involved in the incident** | |
| --- | --- |
| The capture shows DNS **A-record lookups** followed by **TCP/HTTP** web traffic. After resolving yummyrecipesforme.com, the host completes a TCP three-way handshake ([S], [S.], [:]) and issues an **HTTP GET** on port **80**. Later, a new DNS lookup resolves greatrecipesforme.com, and the browser again performs a TCP handshake and **HTTP GET** to that **port 80** site.  **Protocols observed:** DNS → **TCP (HTTP)**  **Ports:** 53 (DNS), **80 (HTTP)** | |
|

| **Section 2: Document the incident** |
| --- |
| **What happened (timeline & evidence):**   * **14:18:32** — DNS query for yummyrecipesforme.com returns **203.0.113.22**. * **14:18:36** — Browser initiates **TCP/80** session to yummyrecipesforme.com and sends **HTTP: GET / HTTP/1.1** (likely fetching page content/download). * **14:20:32** — New DNS query for **greatrecipesforme.com** returns **192.0.2.17/172** (spoof site in exercise). * **14:25:29** — Browser starts a new **TCP/80** session and sends **HTTP GET** to greatrecipesforme.com.   **Analysis (what the data means):**   * The normal browse to yummyrecipesforme.com quickly pivots to **greatrecipesforme.com**, after which traffic continues over **HTTP**. This pattern matches a **malicious redirect to a spoofed site** used to deliver malware or harvest credentials. * The repeated handshakes and GETs on **port 80** (unencrypted HTTP) expose users to interception/tampering risks and make it trivial for an attacker to **steal entered passwords** on the spoof site.   **Impact:**   * Users visiting the cooking site are **redirected to an attacker-controlled domain**, where credentials and/or malware downloads may occur. The incident threatens **confidentiality and integrity** of user accounts and endpoints.   **Likely root cause:**   * A **malicious redirect** (e.g., injected link/ad/script or prior download) steering victims from the legitimate site to a **look-alike domain** (greatrecipesforme.com) over **HTTP**. |

| **Section 3: Recommend one remediation for brute force attacks** |
| --- |
| * **Implement account lockout / throttling on authentication endpoints** (e.g., lock account or introduce exponential back-off after **5** failed attempts; add CAPTCHA after repeated failures). This directly limits online **brute-force** and **credential-stuffing** effectiveness and should be paired with **MFA** for privileged/admin accounts.   *(All timestamps/domains and packet details above are taken from the provided tcpdump reading and log excerpts.* |