# Security incident report

| **Section 1: Identify the network protocol involved in the incident** | |
| --- | --- |
| The protocols identified in the tcpdump log are:  DNS(Domain Name System): Used to resolve domain name IP addresses. DNS queries were made to translate [yummyrecipesforme.com](http://yummyrecipesforme.com) and [greatrecipesforme.com](http://greatrecipesforme.com) to their respective Ip addresses.  HTTP(Hypertext Transfer Protocol): Used to initiate and transfer web content. HTTP requests were made to both the original and malicious websites, initiating the download of a malicious file and redirecting the browser. | |
|

| **Section 2: Document the incident** |
| --- |
| Several customers of [yummrecipesforme.com](http://yummrecipesforme.com) reported being prompted to download a file to access free recipes. After executing the file,their browsers were redirected to [greatrecipesforme.com](http://greatrecipesforme.com) and their systems began running slowly.  An investigation revealed that a former employee executed a brute force attach on the website’s admin panel. This exploited a weak default password. Upon gaining unauthorized access, the attacker embedded malicious JavaScript in the site’s source code. This script initiated the download of an executable file upon page load. When users rna the file, it redirected their browsers to [greatrecipesforme.com](http://greatrecipesforme.com), a fake website designed to distribute malware.  Tcpdump logs confirmed the sequence:   * A DNS request and HTTP GEt to [yummrecipesforme.com](http://yummrecipesforme.com) initiated the download * A second DNS request for [greatrecipesforme.com](http://greatrecipesforme.com) was followed by HTTP traffic to that site, confirming the redirection * The website owner was unable to login due to changed admin credentials * The incident was confirmed by analyzing the tcpdump logs and reviewing the altered source code |

| **Section 3: Recommend one remediation for brute force attacks** |
| --- |
| To prevent brute force attacks, implement a login attempt limiting policy that locks accounts after a defined number of failed login attempts.  Why this works:  Limiting log attempts stops attackers from guessing passwords repeatedly. Once the threshold is reached, the system can lock the account or delay further attempts. This reduces the risk of unauthorized access.  Additional measures to consider:   * Enforce complex password requirements * Enable two-factor authentication (2FA) * Monitor login behavior and set alerts for suspicious activity * Avoid using default passwords and perform regular password audits |