# Security incident report

|  |
| --- |
| **Section 1: Identify the network protocol involved in the incident** |
| The protocol implicated in the incident is the Hypertext Transfer Protocol (HTTP). Given that the issue occurred while trying to access the web server for yummyrecipesforme.com, it's clear that web page requests to servers involve HTTP traffic. Furthermore, when we executed tcpdump while visiting the yummyrecipesforme.com website, the generated tcpdump log file exhibited the use of the HTTP protocol during the connection. The malicious file was detected being transmitted to users' computers via the HTTP protocol at the application layer. |
|

|  |
| --- |
| **Section 2: Document the incident** |
| Numerous customers reached out to the website's helpdesk, reporting that when they visited, they were prompted to download and run a file that purported to offer access to new recipes. Their personal computers have been running slowly ever since. Meanwhile, the website owner attempted to log into the web server but found themselves locked out of their account.  The cybersecurity analyst employed a sandbox environment to access the website without affecting the company network. Following this, the analyst ran tcpdump to capture the network traffic packets generated during the website interaction. The analyst was prompted to download a file that claimed to provide free recipes, which they accepted and executed, resulting in a redirection to a fake site (greatrecipesforme.com).  Upon reviewing the tcpdump log, the cybersecurity analyst noticed that the browser first requested the IP address for yummyrecipesforme.com. After establishing a connection over the HTTP protocol, the analyst recalled downloading and running the file. The logs displayed a sudden shift in network traffic as the browser sought a new IP address for the greatrecipesforme.com URL. Consequently, the network traffic rerouted to the new IP address associated with the greatrecipesforme.com website.  The senior cybersecurity professional examined the source code of both the websites and the downloaded file. The analyst found that an attacker had altered the website to insert code causing users to download a malicious file disguised as a browser update. Given that the website owner reported being locked out of their administrator account, the team believes a brute force attack was used by the attacker to access the account and change the admin password. The execution of the malicious file subsequently compromised the end users' computers. |

|  |
| --- |
| **Section 3: Recommend one remediation for brute force attacks** |
| One security measure the team intends to implement to guard against brute force attacks is to prevent the reuse of previous passwords. Given that the vulnerability leading to this attack stemmed from the attacker’s ability to log in using a default password, it's crucial to ensure old passwords, especially default ones, cannot be used when resetting passwords. Another supporting measure is to mandate more frequent password changes, so if an unauthorized individual becomes aware of the password, they will have less time to exploit it due to timely updates. Lastly, implementing two-factor authentication (2FA) is another effective solution. 2FA requires users to validate their identity through both a password and a one-time passcode (OTP) sent to their email or phone. After confirming their identity with both the login credentials and the OTP, users gain access to the system. Any malicious actor attempting a brute force attack will likely be thwarted from gaining access, as the process necessitates additional authentication steps. |