# Security incident report

|  |
| --- |
| **Section 1: Identify the network protocol involved in the incident** |
| * IP (internet layer) to route packets to their intended destinations * TCP (transport layer) to establish connection and stream data * DNS protocol (application layer) to communicate with DNS server * HTTP (application layer) to enable requests/transfer of data *E.g.: Web page HTML documents, malicious program download* |
|

|  |
| --- |
| **Section 2: Document the incident** |
| The following were captured in a sandbox environment to test:   * Events that occur upon visiting the original website * Events that occur after downloading the file embedded by the attacker   First, a connection request is sent to the intended (original) website from the local system, using DNS to resolve the website’s domain name to its address. Upon successful redirecting of the request, the website host and the local system establish a connection via TCP through a TCP handshake (SYN – SYN, ACK – ACK).  Upon successful connection, the local system then requests data (as seen by the data push + acknowledgement flag [P.] and the HTTP GET function in line 8 of the log file). This is followed by an acknowledgement from the website host (seen by the acknowledgement flag [.]), followed by traffic on port 80, which is associated with HTTP communications. In this context, the traffic involves loading the website’s web page and downloading the file embedded in the website by the attacker.  Following this, we see another connection request to the DNS server, indicating a browser refresh/redirection, this time for the URL of the spoofed website “greatrecipesforme.com” instead of the original “yummyrecipesforme.com”. What follows is similar to the previous connection, with a TCP handshake followed by a HTTP GET request from the local system (further indicated by the data push + acknowledgement flag [P.]), followed by an acknowledgement from the malicious website and a traffic on port 80, which, as stated before, is associated with HTTP communications. In this context, the traffic involves loading the spoofed website’s web page and downloading malware from this website. |

|  |
| --- |
| **Section 3: Recommend one remediation for brute force attacks** |
| The attacker was able to find the password through brute force using common default admin passwords; this is a dictionary brute-force attack. A clear weakness here is weak password policies that permit the use of weak passwords that can be guessed with relative ease. Hence, one remediation for brute-force attacks here is a **password policy that ensures strong passwords** (e.g. through the use of alphanumeric + symbol combinations, the enforcement of a sufficiently large password length (e.g. 12 characters), etc.). Such a policy drastically expands the search space of potential passwords while avoiding common guesses, making it infeasible for an attacker to guess the password through brute-force methods. |