Brute Force Attack: Security incident report

You are a cybersecurity analyst for yummyrecipesforme.com, a website that sells recipes and cookbooks. A former employee has decided to lure users to a fake website with malware.

The former employee/ hacker executed a brute force attack to gain access to the web host. They repeatedly entered several known default passwords for the administrative account until they correctly guessed the right one. After they obtained the login credentials, they were able to access the admin panel and change the website’s source code. They embedded a javascript function in the source code that prompted visitors to download and run a file upon visiting the website. After embedding the malware, the hacker changed the password to the administrative account. When customers download the file, they are redirected to a fake version of the website that contains the malware.

Several hours after the attack, multiple customers emailed yummyrecipesforme’s helpdesk. They complained that the company’s website had prompted them to download a file to access free recipes. The customers claimed that, after running the file, the address of the website changed and their personal computers began running more slowly.

In response to this incident, the website owner tries to log in to the admin panel but is unable to, so they reach out to the website hosting provider. You and other cybersecurity analysts are tasked with investigating this security event.

To address the incident, you create a sandbox environment to observe the suspicious website behavior. You run the network protocol analyzer tcpdump, then type in the URL for the website, yummyrecipesforme.com. As soon as the website loads, you are prompted to download an executable file to update your browser. You accept the download and allow the file to run. You then observe that your browser redirects you to a different URL, greatrecipesforme.com, which contains the malware.

The logs show the following process:

1. The browser initiates a DNS request: It requests the IP address of the yummyrecipesforme.com URL from the DNS server.
2. The DNS replies with the correct IP address.
3. The browser initiates an HTTP request: It requests the yummyrecipesforme.com webpage using the IP address sent by the DNS server.
4. The browser initiates the download of the malware.
5. The browser initiates a DNS request for greatrecipesforme.com.
6. The DNS server responds with the IP address for greatrecipesforme.com.
7. The browser initiates an HTTP request to the IP address for greatrecipesforme.com.

A senior analyst confirms that the website was compromised. The analyst checks the source code for the website. They notice that javascript code had been added to prompt website visitors to download an executable file. Analysis of the downloaded file found a script that redirects the visitors’ browsers from yummyrecipesforme.com to greatrecipesforme.com.

The cybersecurity team reports that the web server was impacted by a brute force attack. The disgruntled hacker was able to guess the password easily because the admin password was still set to the default password. Additionally, there were no controls in place to prevent a brute force attack.

Your job is to document the incident in detail, including identifying the network protocols used to establish the connection between the user and the website.  You should also recommend a security action to take to prevent brute force attacks in the future.

|  |
| --- |
| **Section 1: Identify the network protocol involved in the incident** |
| This is an application layer attack focusing on abusing HTTP and DNS requests to download malicious updates to the user’s browser and redirect them to a fake copy of yummyrecipesforme.com. |
|  |

|  |
| --- |
| **Section 2: Document the incident** |
| A disgruntled user of yummyrecipesforme.com executed a brute force attack on the administrative account of the web server. After obtaining the correct password, the attacker accessed the admin panel to change the website’s source code. A JavaScript function was added to the source code that prompted visitors to download and run a file when they visited the site. After downloading the file, users were redirected to a spoofed version of the website with the domain name greatrecipesforme.com. The attacker uploaded all of a seller’s paid recipes for free to this site, and users also stated that after running the downloaded file their computers began running more slowly.  The cybersecurity analyst has simulated the customer’s actions in a sandbox (Virtual Machine) and confirmed that when visiting yummyrecipesforme.com the following occurs:   1. The browser requests a DNS resolution of the yummyrecipesforme.com URL. 2. The DNS replies with the correct IP address. 3. The browser initiates an HTTP request for the webpage. 4. The browser initiates the download of the malware. 5. The browser requests another DNS resolution for greatrecipesforme.com. 6. The DNS server responds with the new IP address. 7. The browser initiates an HTTP request to the new IP address. |

|  |
| --- |
| **Section 3: Recommend one remediation for brute force attacks** |
| It was later discovered that the password to the admin account was still set to the default password. The best solution to protect against future brute force attacks is to implement secure password policies for this account and the organization. Specific password policies to implement for this account include:   1. Implement techniques to block large amounts of failed password attempts (ex: blocking specific IPs after too many attempts) 2. Update the password requirements to be a certain length and include different types of characters instead of just letters 3. Require recurring password changes 4. Require 2-Factor or Multi-Factor Authentication (2FA or MFA)   One policy to focus on is blocking large amounts of failed password attempts. A brute force attack successfully guesses an accounts credentials by trial-and-error and guesses as many passwords as possible from a list of commonly used passwords. The admin account did not have preventative techniques to detect large amounts of failed password attempts and allowed the attacker to guess the password as many times as they wanted. |