***Yummyrecipes.com Incident Report and OS Hardening***

You are a cybersecurity analyst for yummyrecipesforme.com, a website that sells recipes and cookbooks. A disgruntled baker has decided to publish the website’s best-selling recipes for the public to access for free.

The baker 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 running the downloaded file, the customers are redirected to a fake version of the website where the seller’s recipes are now available for free.

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 update their browsers. 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 is designed to look like the original site. However, the recipes your company sells are now posted for free on the new website.

The logs show the following process:

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.

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 baker 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.

How to read the DNS & HTTP traffic log

This reading explains how to identify the brute force attack using tcpdump.

|  |
| --- |
| 14:18:32.192571 IP **your.machine.52444 > dns.google.domain**: 35084+ A? **yummyrecipesforme.com**. (24)  14:18:32.204388 IP **dns.google.domain** > **your.machine.52444**: 35084 1/0/0 A **203.0.113.22** (40) |

The first section of the DNS & HTTP traffic log file shows the source computer (**your.machine.52444**) using port **52444** to send a DNS resolution request to the DNS server (**dns.google.domain**) for the destination URL (**yummyrecipesforme.com**). Then the reply comes back from the DNS server to the source computer with the IP address of the destination URL **(203.0.113.22**).

|  |
| --- |
| 14:18:36.786501 IP **your.machine.36086** > **yummyrecipesforme.com.http**: **Flags [S]**, seq 2873951608, win 65495, options [mss 65495,sackOK,TS val 3302576859 ecr 0,nop,wscale 7], length 0  14:18:36.786517 IP yummyrecipesforme.com.http > your.machine.36086: **Flags** **[S.]**, seq 3984334959, ack 2873951609, win 65483, options [mss 65495,sackOK,TS val 3302576859 ecr 3302576859,nop,wscale 7], length 0 |

The next section shows the source computer sending a connection request (**Flags [S]**) from the source computer (**your.machine.36086**) using port **36086** directly to the destination (**yummyrecipesforme.com.http**). The **.http** suffix is the port number; **http** is commonly associated with port 80. The reply shows the destination acknowledging it received the connection request (**Flags [S.]**). The communication between the source and the intended destination continues for about 2 minutes, according to the timestamps between this block (**14:18**) and the next DNS resolution request (see below for the **14:20** timestamp).

**TCP Flag codes include:**

**Flags [S]**  - Connection **S**tart

**Flags [F]**  - Connection **F**inish

**Flags [P]**  - Data **P**ush

**Flags [R]**  - Connection **R**eset

**Flags [.]**  - Acknowledgment

|  |
| --- |
| 14:18:36.786589 IP your.machine.36086 > **yummyrecipesforme.com.http**: Flags [P.], seq 1:74, ack 1, win 512, options [nop,nop,TS val 3302576859 ecr 3302576859], length 73: **HTTP: GET / HTTP/1.1** |

The log entry with the code **HTTP: GET / HTTP/1.1** shows the browser is requesting data from **yummyrecipesforme.com** with the **HTTP: GET** method using **HTTP** protocol version **1.1**. This could be the download request for the malicious file.

|  |
| --- |
| **14:20**:32.192571 IP **your.machine.52444 > dns.google.domain**: 21899+ A? greatrecipesforme.com. (24)  14:20:32.204388 IP **dns.google.domain > your.machine.52444**: 21899 1/0/0 A **192.0.2.172** (40)  14:25:29.576493 **IP your.machine.56378 > greatrecipesforme.com.http**: Flags [S], seq 1020702883, win 65495, options [mss 65495,sackOK,TS val 3302989649 ecr 0,nop,wscale 7], length 0  14:25:29.576510 IP **greatrecipesforme.com.http > your.machine.56378**: Flags [S.], seq 1993648018, ack 1020702884, win 65483, options [mss 65495,sackOK,TS val 3302989649 ecr 3302989649,nop,wscale 7], length 0 |

Then, a sudden change happens in the logs. The traffic is routed from the source computer to the DNS server again using port **.52444** (**your.machine.52444 > dns.google.domain**) to make another DNS resolution request. This time, the DNS server routes the traffic to a new IP address (**192.0.2.172)** and its associated URL (**greatrecipesforme.com.http**). The traffic changes to a route between the source computer and the spoofed website (outgoing traffic: **IP your.machine.56378 > greatrecipesforme.com.http** and incoming traffic: **greatrecipesforme.com.http >** **IP your.machine.56378**). Note that the port number (**.56378**) on the source computer has changed again when redirected to a new website.

Apply OS hardening techniques

|  |
| --- |
| **Section 1: Identify the network protocol involved in the incident** |
| The Hypertext transfer protocol (HTTP) was the protocol that was impacted by the incident. Subsequent to running tcpdump and accessing the yummyrecipesforme.com website to detect the problem, capture protocol, and traffic activity in a DNS & HTTP traffic log file provided the evidence needed to come to this conclusion. At the application layer the malicious file was detected  being transported to the users’ computers using the HTTP protocol. |
|

|  |
| --- |
| **Section 2: Document the incident** |
| Clients emailed the website owner stating that they were prompted to download and run a file that asked them to update their browsers after visiting the website. Downloading the file on their computers resulted in slower operating performance on thier machines. The owner of the website observed that. The website owner tried logging into the web server but noticed they were locked out of their account.  A sandbox tool was employed by the analyst to test the website while not exposing the company network to vulnerabilities. Then, the analyst ran tcpdump to capture the network and protocol traffic packets produced by interacting with the website. The analyst proceeded to download the file that claimed to update the browser and ran it. The browser then rerouted the network traffic to a fabricated website (greatrecipesforme.com) that was designed to mirror the original site (yummyrecipesforme.com).  Upon through inspection of the tcpdumplog the analyst that observed the browser initially requested the IP address for the yummyrecipesforme.com website. The logs showed a sudden change in network traffic as the browser requested a new IP resolution for the greatrecipesforme.com URL. The network traffic was then rerouted to the new IP address for the greatrecipesforme.com website.  The senior cybersecurity professional analyzed the source code for the websites and the downloaded file. An attacker compromised the website to add code that prompted the users to download a malicious file disguised as a browser update. Since the website owner stated that they had been locked out of their administrator account, the team believes the attacker used a brute force attack to access the account and change the admin password. The execution of the malicious file compromised the end users’ computers. |

|  |
| --- |
| **Section 3: Recommend one remediation for brute force attacks** |
| The security measure that the SOC team will implement to protect against brute force attacks is two-factor authentication (2FA). This 2FA plan will include an additional requirement for users to validate their identification by confirming a one-time password (OTP) sent to either their email or phone. Once the user confirms their identity through their login credentials and the OTP, they will gain access to the system. Any malicious actor that attempts a brute force attack will not likely gain access to the system because it requires additional authorization. |