(Mock) Response to Abnormal Web Server Activity

A red text with a black swoosh

Description automatically generated

Material & instructions developed by: Google Cybersecurity Professional Certificate Course

Completed by: Alexander Herman on 01/16/24

**Table of Contents:**

Activity Overview (Provided): \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**2**

Scenario (Provided): \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**2-3**

DNS & HTTP Log (Provided):\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**3-4**

Log Quick Notes (My Work): \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**4**

Incident Report (My Work): \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**5**

**Activity Overview**



In this activity, you will take on the role of a cybersecurity analyst working for a company that hosts the cooking website, yummyrecipesforme.com. Visitors to the website experience a security issue when loading the main webpage. Your job is to investigate, identify, document, and recommend a solution to the security problem.

When investigating the security event, you will review a tcpdump log. You will need to identify the network protocols used to establish the connection between the user and the website. Network protocols are the communication rules and standards networked devices use to transmit data. Unfortunately, malicious actors can also use network protocols to invade and attack private networks. Knowing how to identify the protocols commonly used in attacks will help you protect your organization’s network against these types of security events.

To complete the assignment, you will also need to document what occurred during the security incident. Then, you will recommend one security measure to implement to prevent similar security problems in the future.

**Scenario**

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.

**DNS & HTTP Traffic Log**

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)

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

14:18:36.786529 IP your.machine.36086 > yummyrecipesforme.com.http: Flags

[.], ack 1, win 512, options [nop,nop,TS val 3302576859 ecr 3302576859],

length 0

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

14:18:36.786595 IP yummyrecipesforme.com.http > your.machine.36086: Flags

[.], ack 74, win 512, options [nop,nop,TS val 3302576859 ecr 3302576859],

length 0

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

192.0.2.17 (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

14:25:29.576524 IP your.machine.56378 > greatrecipesforme.com.http: Flags

[.], ack 1, win 512, options [nop,nop,TS val 3302989649 ecr 3302989649],

length 0

14:25:29.576590 IP your.machine.56378 > greatrecipesforme.com.http: Flags

[P.], seq 1:74, ack 1, win 512, options [nop,nop,TS val 3302989649 ecr

3302989649], length 73: HTTP: GET / HTTP/1.1

14:25:29.576597 IP greatrecipesforme.com.http > your.machine.56378: Flags

[.], ack 74, win 512, options [nop,nop,TS val 3302989649 ecr 3302989649],

length 0

**Log Quick Notes**

* your.machine sent a valid DNS request but when the connection start acknowledgement was processed, a weird HTTP download request (data push) was sent from the web server (HTTP: GET / HTTP/1.1)
* After accepting the request, my.machine was forced to send a new DNS request for a (greatrecipesforme.com)
* google returned the IP of 192.0.2.17 for greatrecipesforme.com
* A lot of traffic on the HTTP port 80 (unsecure)

**Security Incident Report**

**Problem Summary**:

After receiving reports from customers that when attempting to access the website (www.yummyrecipesforme.com), they were prompted to download & run a file that resulted in them being directed to a new website & their computers began running slower, I reviewed the DNS/HTTP logs. Upon reviewing the logs, it was clear that there was a problem with the company web server and the HTTP network protocol at the application layer.

**Analysis**:

This afternoon, several customers reported problems with the company website. In response, I set up a sandbox environment and ran tcpdump to capture the DNS/HTTP logs. At 14:18:32.192571, when the connection to the website (yummyrecipesforme.com), began I was promoted to download and run an executable file with (HTTP: GET / HTTP/1.1). This is not a normal data push for the web server to be executing suggesting that there is a problem with the web server. Sure enough, the web server admin later confirmed that there was a brute force attack on the web server and the threat actor was able to gain admin level access, lock the web server admins out by changing the password, and injected malicious JavaScript into the source code. These new lines of code are what prompted users to download the executable file that was disguised as a browser update. After downloading & running the file, I was redirected to a similar website (greatrecipesforme.com) where all the products sold on the original site were downloadable for free. This resulted in end users’ device being compromised. Looking forward there are several hardening recommendations of changes to make to prevent this from occurring again. To protect the web server from brute force attacks, the company should implement MFA or 2FA to add a layer of security to the admin login portal. Additionally, they should implement strong password policies to prevent threat actors from being able to easily guess the password. Finally, reCAPTCHA or CAPTCHA should be implemented to protect against software or bots that may attempt a dictionary attack.