Security Incident Report

Applying OS Hardening Techniques

Wendy Alayon
11/06/2024 **@syszern**

Table of Contents

Security incident report

Scenario

Assume the role of 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.

A malicious actor executed a brute force attack to gain access to the web host. They repeatedly entered several known default passwords for the administrative account until correctly guessing the right one. Upon obtaining the login credentials, they accessed the admin panel and modified 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 attacker changed the password to the administrative account. When customers downloaded the file, they were redirected to a fake version of the website that contained 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 reported 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 attempted to log in to the admin panel but was unable to do so. Consequently, they reached out to the website hosting provider. The cybersecurity team, including the security analyst, was tasked with investigating this security event.

To address the incident, the analyst created a sandbox environment to observe the suspicious website behavior. They ran the network protocol analyzer tcpdump and then entered the URL for the website, yummyrecipesforme.com. Upon loading the website, a prompt appeared to download an executable file to update the browser. The analyst accepted the download and allowed the file to run. The browser then redirected to a different URL, greatrecipesforme.com, which contained 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 noticed 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.

The security analyst's job is to document the incident in detail, including identifying the network protocols used to establish the connection between the user and the website. The analyst is also tasked with recommending a security action to prevent future brute force attacks.

tcpdump 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
…<a lot of traffic on the port 80>...


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
…<a lot of traffic on the port 80>...
```

Determining One Network Protocol Implicated in the Incident

The network protocol involved in the problem is Hypertext Transfer Protocol (HTTP), as it is related to accessing the web server hosting the company's website (yummyrecipesforme.com). Requests made to web servers typically utilize HTTP traffic. This is confirmed by running a network protocol analyzer tool, tcpdump, which revealed log entries showing the usage of the HTTP protocol when contacting the web server hosting the company's website. Furthermore, the issue pertaining to the malicious file download prompt is also observed to be utilizing the same protocol (HTTP) to transport data to the user's computer. HTTP operates at the application layer of the TCP/IP model.

Section 2: Incident Documentation

Several customers contacted the company's website helpdesk, informing them that upon attempting to access the site, they were prompted to download a file to access free recipes. After downloading and running the file, they were redirected to a different URL. Additionally, they reported experiencing decreased PC performance following the execution of the file. In response to these reports, the website owner tried to access the admin panel but was unable to log in to the administrative account. Consequently, they reached out to their web hosting provider for assistance.

The security analyst established a sandbox environment to monitor suspicious website activities without risking the company network. They utilized tcpdump to capture packets generated during interactions with the website. Once the website loaded, the analyst was immediately prompted to download an executable file, supposedly for a browser update. The analyst accepted to the download and allowed the executable file to run. Immediately thereafter, the browser redirected the analyst to a different URL.

Analysis of log entries presented by tcpdump show that initially, the browser initiated a DNS request to acquire the IP address associated with the URL yummyrecipesforme.com from the DNS server. Upon receiving the correct IP address from the DNS server, the browser proceeded to send an HTTP request for the webpage hosted at that address. Unexpectedly, during this interaction, the browser also initiated the download of malicious file. After this event, the browser again engaged in a DNS request, this time for a different URL greatrecipesforme.com. The DNS server promptly responded with the corresponding IP address. Finally, the browser sent an HTTP request to the IP address linked to greatrecipesforme.com, completing the sequence of events as documented in the logs.

The senior cybersecurity analyst confirmed that the website had been compromised. Following a thorough analysis of the website's source code, it was discovered that a

JavaScript code had been inserted. This code prompted visitors to download an executable file, disguised as a "browser update." Embedded within this file was a script that redirected visitors' browsers from the correct URL to a fake one. Consequently, the execution of this file compromised the computers of website visitors. Meanwhile, the cybersecurity team reported that the web server was targeted by a brute force attack. Given the website owner's inability to log in to the admin panel, the analysts concluded that the attacker likely used a brute force attack to guess the correct password for the admin account and then changed it, effectively denying access to the website owner. It's noteworthy that the admin account retained its default password during the attack. Moreover, it was noted that there were no controls in place to prevent such an attack.

Section 3: One Recommended Security Measure

One recommended security measure to implement is password policies that enforce strong passwords and regular password changes. This can prevent attackers from using easily guessable passwords, especially in this case where the the administrator account (admin account) still had the default password assigned to it, rather than having been changed to a more secure password, making the account vulnerable to unauthorized access. Another supportive security measure is implementing account lockout mechanisms that temporarily lock an account after a certain number of failed login attempts. This can thwart brute force attacks by preventing attackers from making unlimited login attempts until the correct password has been guessed. Implementing two-factor authentication where users must provide a second form of verification, in addition to their password also adds an extra layer of security even if the password is compromised.