Web Security Assignment

IT19130026

Samarasinghe M.L.S

Content Page

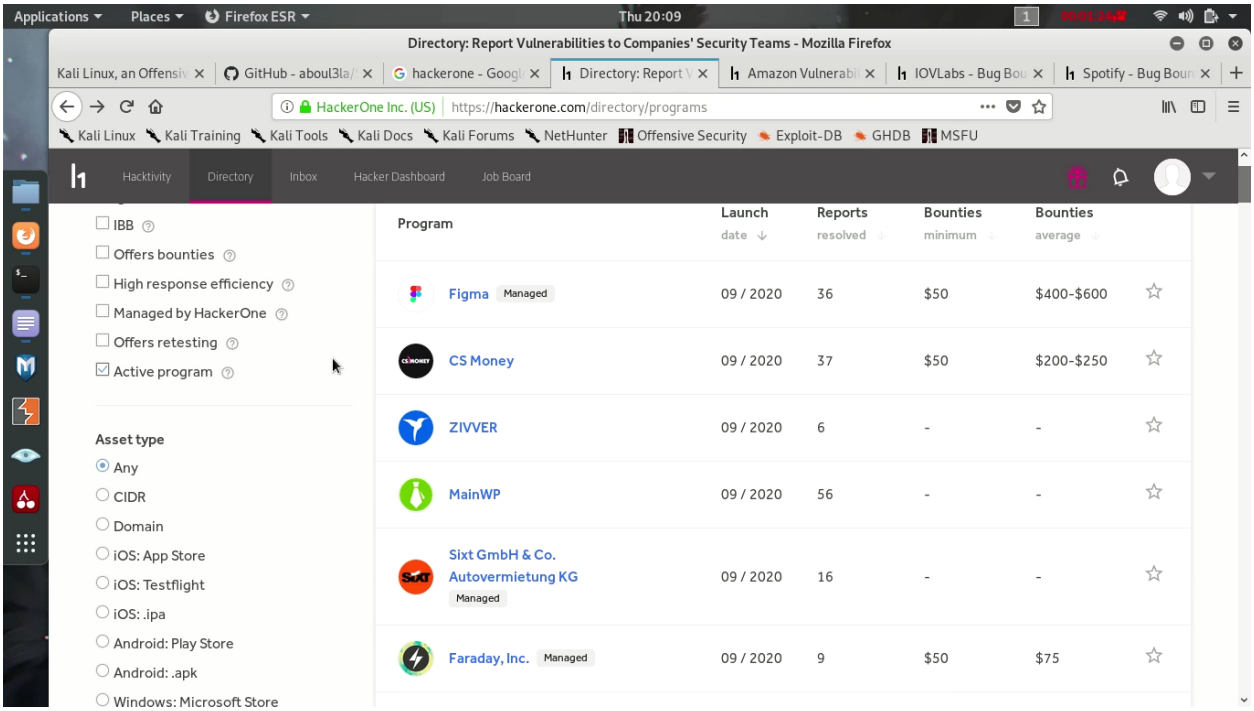
Introduction

To complete this assignment, I used a few basic scans using few tools to check for the Top 10 Vulnerabilities listed by OWASP

- Injection
- Broken Authentication
- Sensitive Data Exposure
- XML External Entities (XXE)
- Broken Access Control
- Security Misconfigurations
- Cross Site Scripting (XSS)
- Insecure Deserialization
- Using Components with Known Vulnerability
- Insufficient logging and monitoring

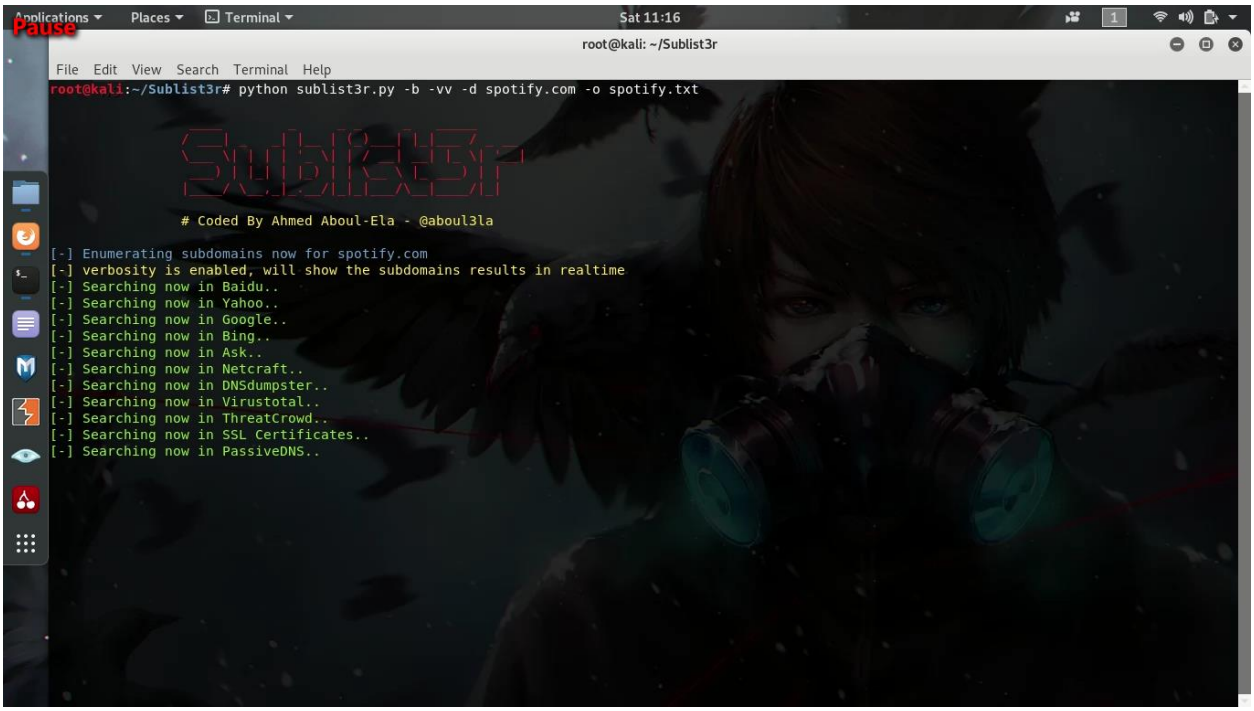
I used **Sblist3r, Nikto, Netspaker pro, lazyrecon** for this scanning process.

Selecting a Domain



To select a Subdomain, I logged in to the hackerone bug bounty program and I selected "**spotofy.com**" as my domain.

Getting Subdomains count



I used this python tool called **Sublist3r** to scan my domain "**Spotify.com**" for its subdomain count.

You can use this link to watch the video hoe I selected a domain, how I installed this tool and scanned using this tool.

Link for the recorded Video #1:
https://drive.google.com/file/d/1y4OAqTkz3TSrkJJY196rl3U74uexdqSP/view?usp=sharing

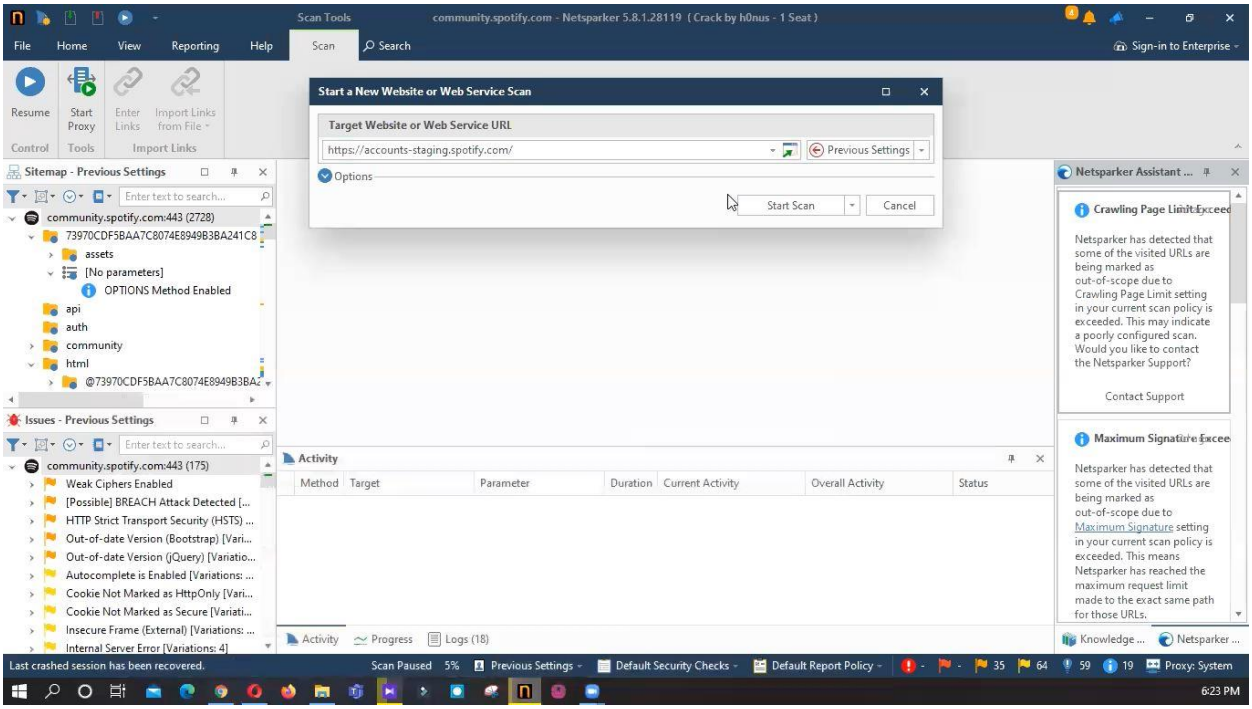
After that scan I got 158 subdomains as the results. Then I decided to use the subdomain list of 20 given in the hackerone platform's spotify page.

- accounts-staging.spotify.com
- accounts.spotify.com
- ads.spotify.com
- artists.spotify.com
- canvas.spotify.com
- certomato.spotify.com
- community.spotify.com
- csat-support-help-page-mobile.spotify.com
- developer.spotify.com
- homething.spotify.com
- hrblog.spotify.com
- newsroom.spotify.com
- podcasters.spotify.com
- promo.spotify.com
- purple.spotify.com
- spotify.com
- support.spotify.com
- surveys.spotify.com
- uplink.spotify.com
- works.spotify.com

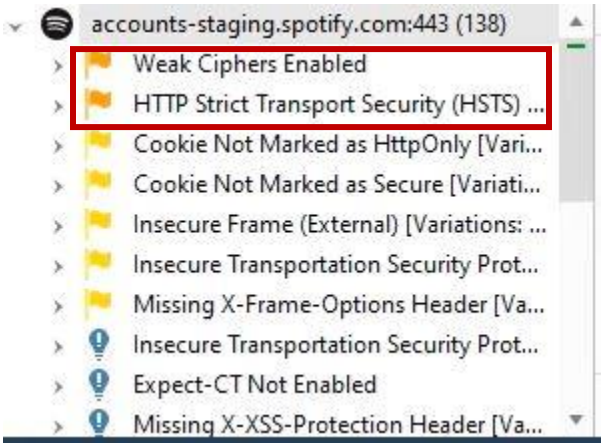
First Scan set

After I got subdomain list, I tried to scan few of them using that Netspaker Pro tool.

Accounts-staging.spotify.com

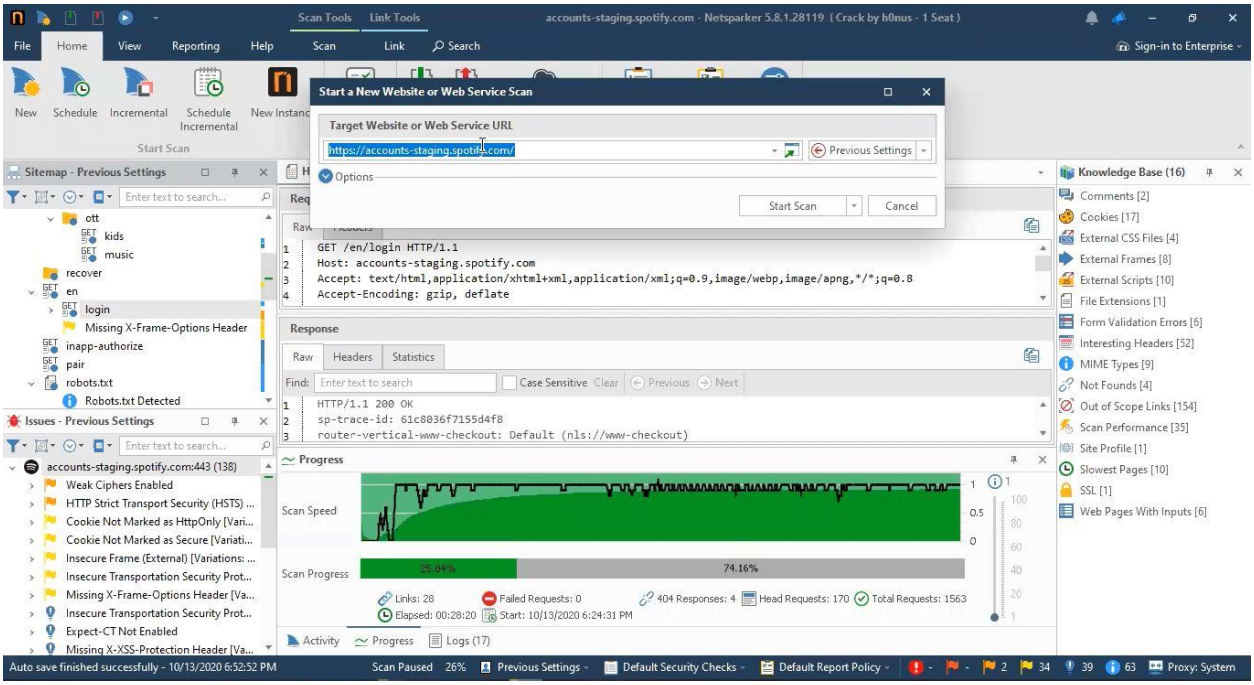


Result:

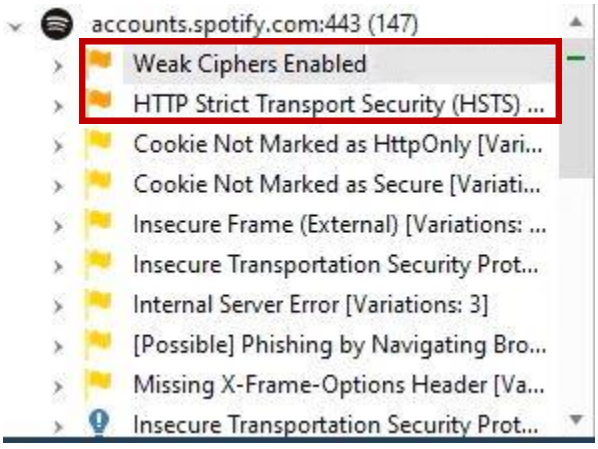


Here I get these two medium range vulnerabilities only. Other low range vulnerabilities wont effect to the domain.

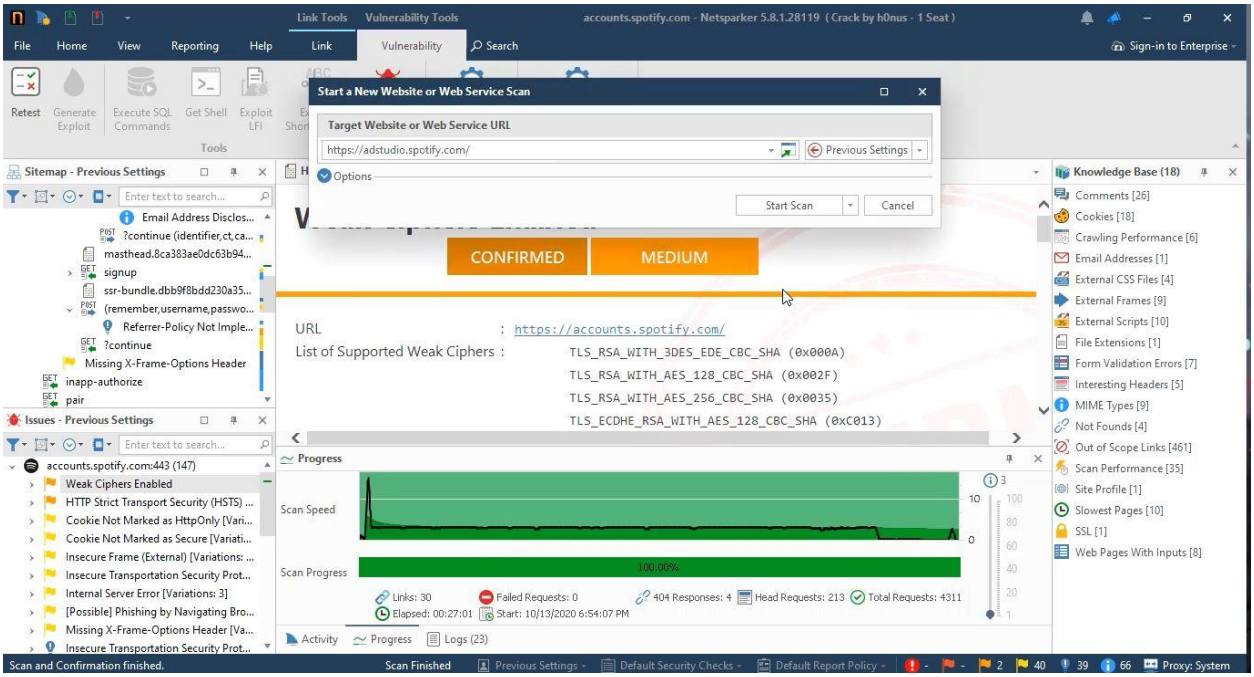
Account.spotify.com



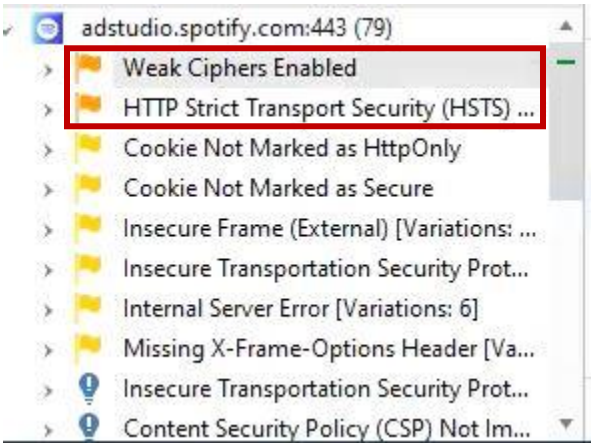
Results:



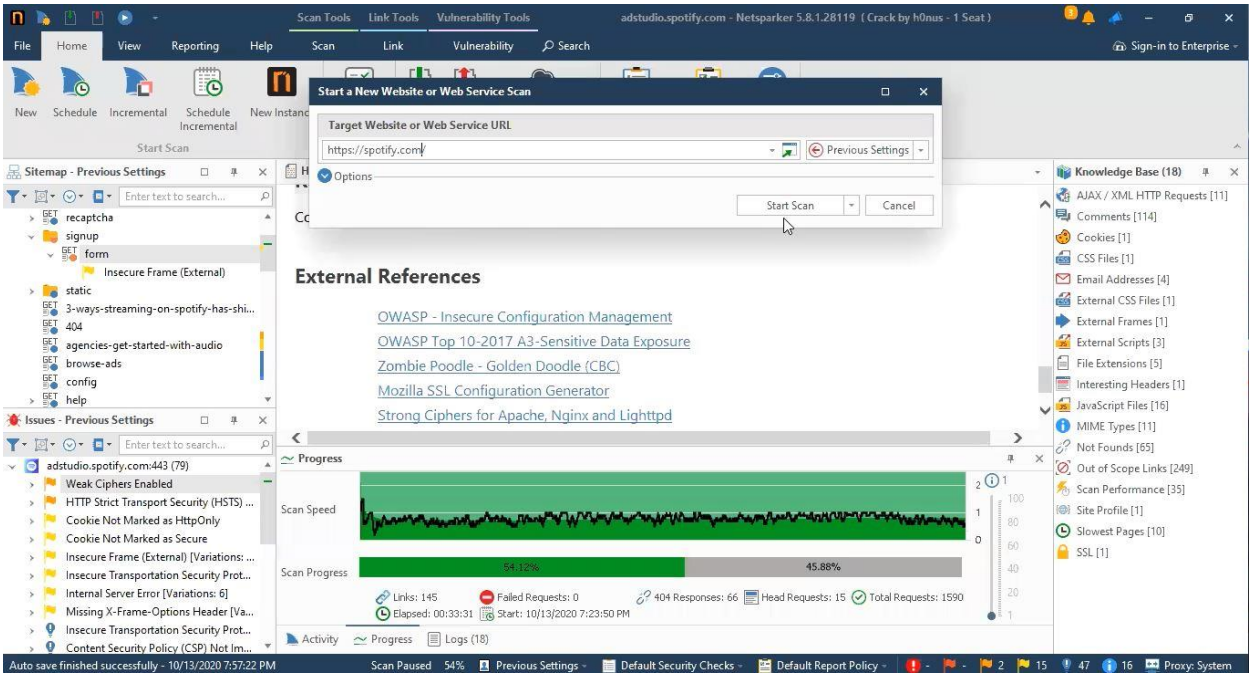
adstudio.spotify.com



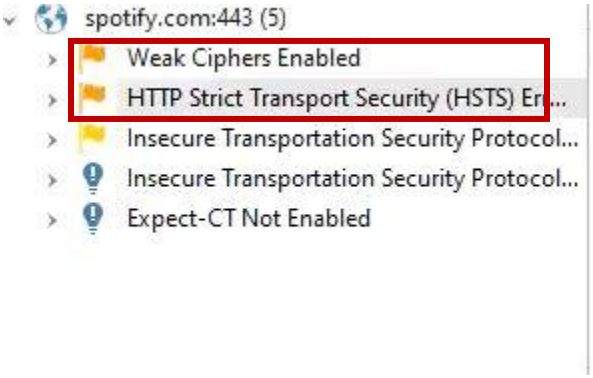
Results:



spotify.com



Results:



Here you can see I got the same results after scanning these subdomains.

Results of first Scan Set

Weak Ciphers Enabled

List of Supported Weak Ciphers:

- TLS_RSA_WITH_3DES_EDE_CBC_SHA (0x000A)
- TLS_RSA_WITH_AES_128_CBC_SHA (0x002F)
- TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xC013)
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xC014)

CLASSIFICATION

- PCI DSS 3.26.5.4
- OWASP 2013A6
- OWASP 2017A3
- CWE327
- CAPEC217
- WASC4
- ISO27001A.14.1.3

CVSS 3.0 SCORE

Base6.8 (Medium)

- Temporal6.8 (Medium)
- Environmental6.8 (Medium)
- CVSS Vector String
- CVSS:3.0/AV:A/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:N

CVSS 3.1 SCORE

- Base6.8 (Medium)
- Temporal6.8 (Medium)
- Environmental6.8 (Medium)
- CVSS Vector String
- CVSS:3.1/AV:A/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:N

Vulnerability Details

Netsparker detected that weak ciphers are enabled during secure communication (SSL).

You should allow only strong ciphers on your web server to protect secure communication with your visitors.

Impact

Attackers might decrypt SSL traffic between your server and your visitors.

Actions to Take

For Apache, you should modify the SSLCipherSuite directive in the httpd.conf.

SSLCipherSuite HIGH: MEDIUM: !MD5: !RC4

Lighttpd:

- ssl.honor-cipher-order = "enable"
- ssl.cipher-list = "EECDH+AESGCM:EDH+AESGCM"

For Microsoft IIS, you should make some changes to the system registry. Incorrectly editing the registry may severely damage your system. Before making changes to the registry, you should back up any valued data on your computer.

a. Click Start, click Run, type regedt32 or type regedit, and then click OK.

b. In Registry Editor, locate the following registry key:
`HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders`

c. Set "Enabled" DWORD to "0x0" for the following registry keys:

- `SCHANNEL\Ciphers\DES 56/56`
- `SCHANNEL\Ciphers\RC4 64/128`
- `SCHANNEL\Ciphers\RC4 40/128`
- `SCHANNEL\Ciphers\RC2 56/128`
- `SCHANNEL\Ciphers\RC2 40/128`
- `SCHANNEL\Ciphers\NULL`
- `SCHANNEL\Hashes\MD5Remedy`

Configure your web server to disallow using weak ciphers.

External References

- OWASP - Insecure Configuration Management
- OWASP Top 10-2017 A3-Sensitive Data Exposure
- Zombie Poodle - Golden Doodle (CBC)

- Mozilla SSL Configuration Generator
- Strong Ciphers for Apache, Nginx and Lighttpd

HTTP strict Transport Security (HSTS) Errors and Warnings

CLASSIFICATION:

- OWASP 2013A5
- OWASP 2017A6
- CWE16
- WASC15
- ISO27001A.14.1.2

Error Resolution

preload directive not present Submit domain for inclusion in browsers' HTTP Strict Transport Security (HSTS) preload list.

Vulnerability Details

Netspaker detected errors during parsing of Strict-Transport-Security header.

Impact

The HSTS Warning and Error may allow attackers to bypass HSTS, effectively allowing them to read and modify your communication with the website.

Remedy

Ideally, after fixing the errors and warnings, you should consider adding your domain to the the HSTS preload list. This will ensure that browsers automatically connect your website by using HTTPS, actively preventing users from visiting your site using HTTP. Since this list is hardcoded in users' browsers, it will enable HSTS even before they visit your page for the first time, eliminating the need for Trust On First Use (TOFU) with its associated risks and disadvantages. Unless you fix the errors and warnings your website won't meet the conditions required to enter the browser's preload list.

Browser vendors declared:

Serve a valid certificate,

If you are listening on port 80, redirect all domains from HTTP to HTTPS on the same host. Serve all subdomains over HTTPS:

- In particular, you must support HTTPS for the www subdomain if a DNS record for that subdomain exists

Serve an HSTS header on the base domain for HTTPS requests:

- The max-age must be at least 31536000 seconds (1 year)
- The include Subdomains directive must be specified
- The preload directive must be specified
- If you are serving an additional redirect from your HTTPS site, that redirect must have the HSTS header (rather than the page it redirects to)

External References

- HTTP Strict Transport Security (HSTS) HTTP Header
- Wikipedia - HTTP Strict Transport Security Implementation
- Check HSTS Preload status and eligibility

You can use this link to watch how I scanned those subdomains using Netspaker Pro tool.

Link for the recorded Video #2:
https://drive.google.com/file/d/183Q2k5rshggQe78FGcbklpTkze48B0iW/view?usp=sharing

Scan using Nikto Tool (port 80)

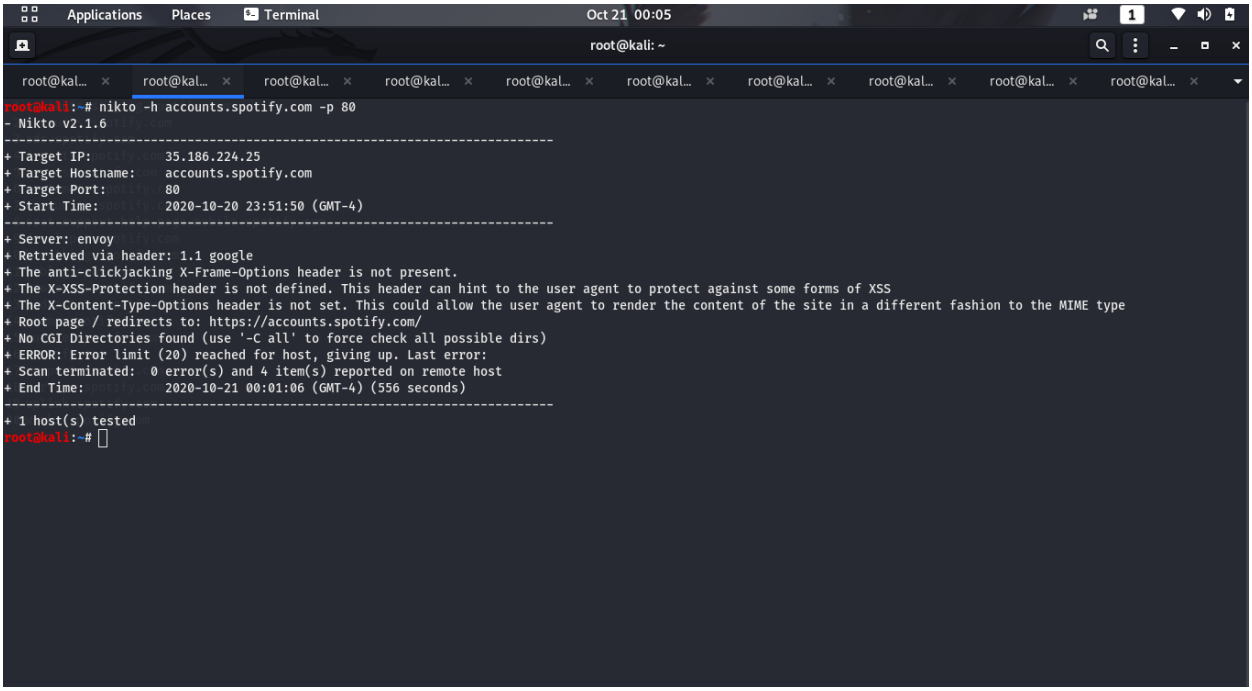
After above scans, I realized Netspaker takes time. Then I decided to use Nikto tool before Netspaker because Nikto is speeder than Netspaker and after that scan few specific subdomains using Netspaker.

First, I scanned all subdomains for port 80.

accounts-staging.spotify.com

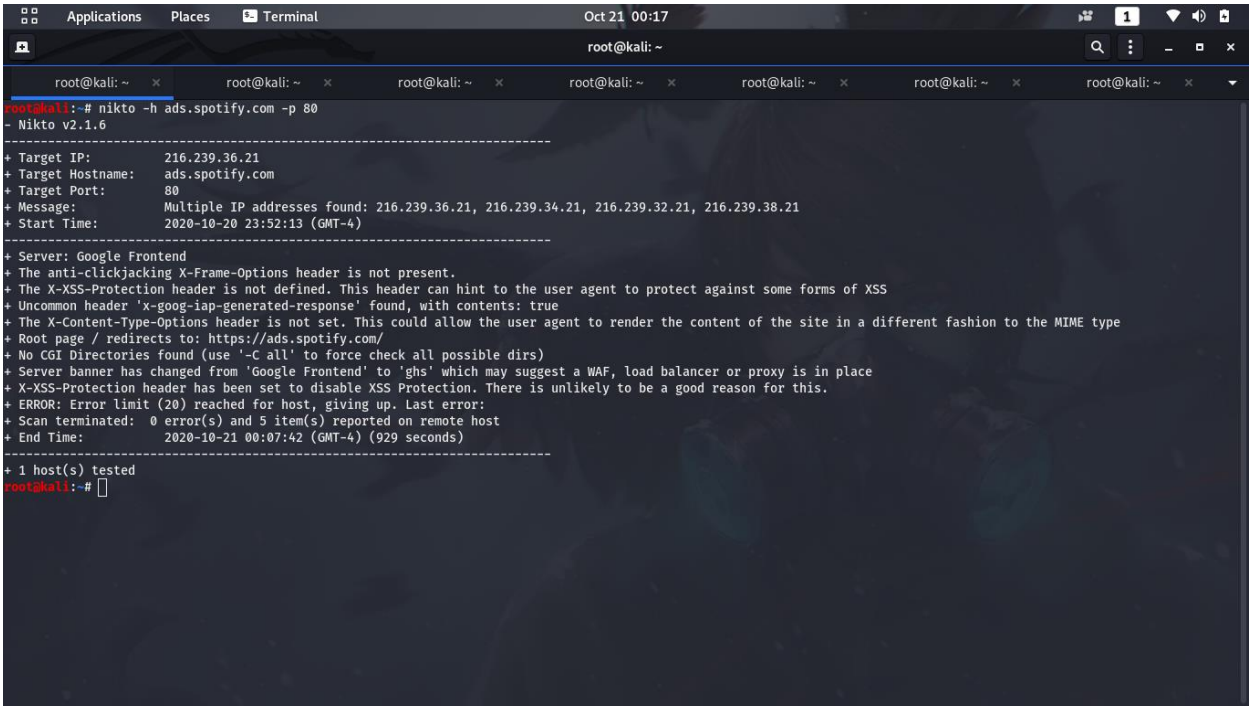
1. **Clickjacking**
2. **X-XSS-Protection header is not defined**
3. **X-Content-Type-Options header is not set**

accounts.spotify.com



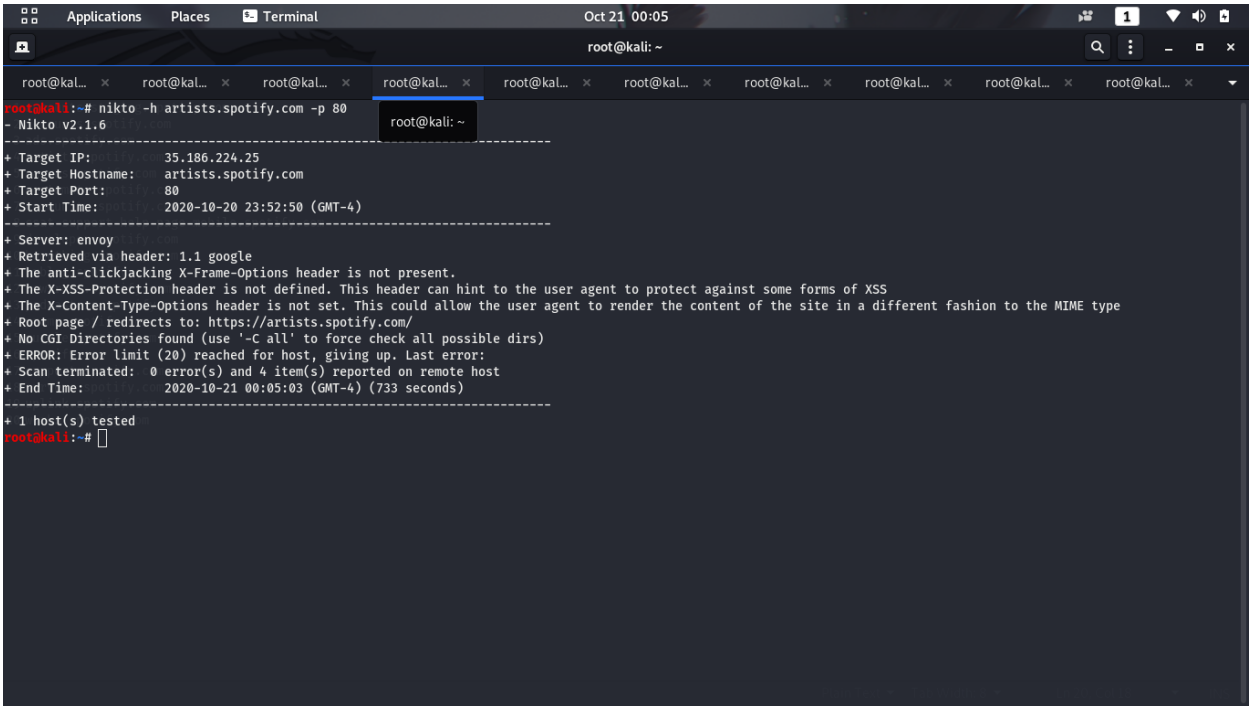
1. **Clickjacking**
2. **X-XSS-Protection header is not defined**
3. **X-Content-Type-Options header is not set**

ads.spotify.com



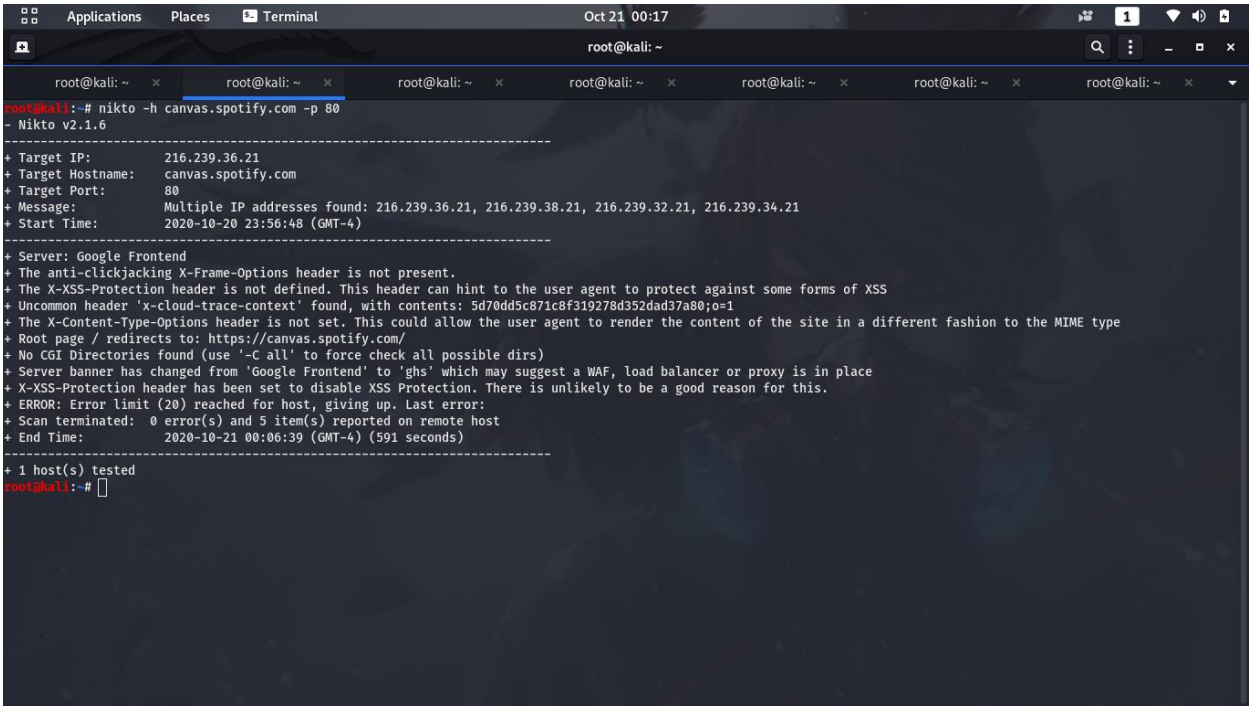
1. **Clickjacking**
2. **X-XSS-Protection header is not defined**
3. **X-Content-Type-Options header is not set**

artists.spotify.com



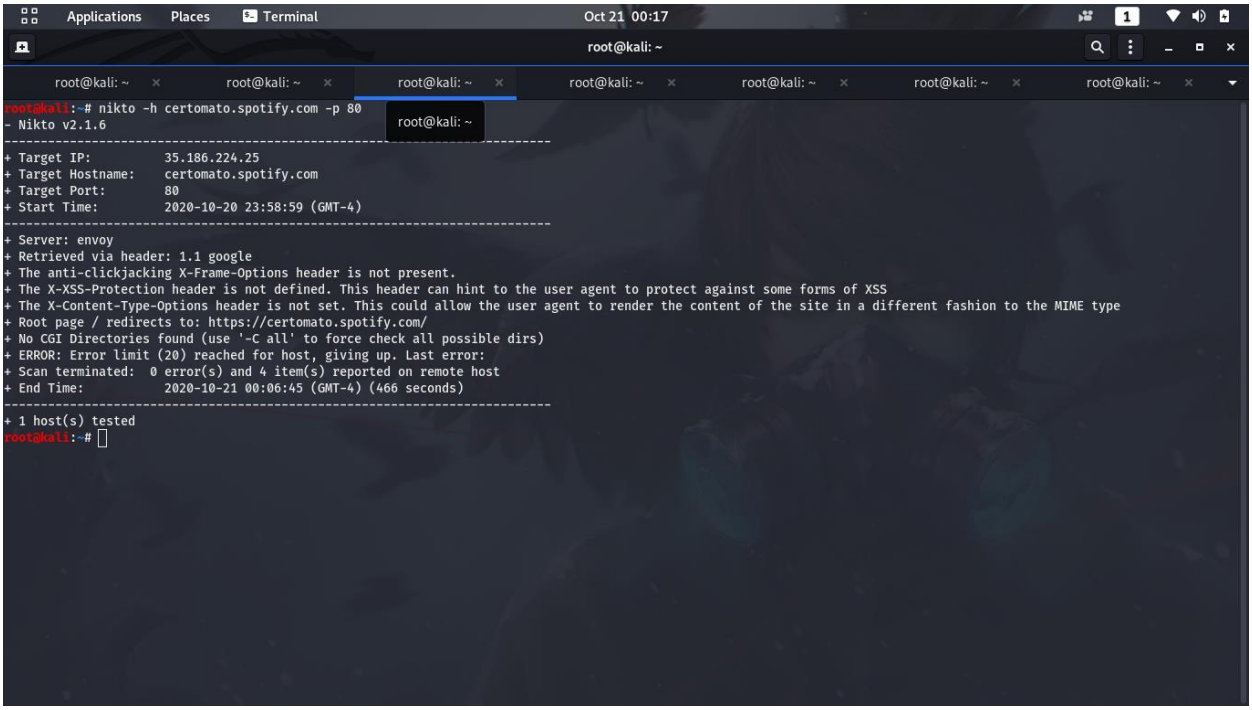
1. **Clickjacking**
2. **X-XSS-Protection header is not defined**
3. **X-Content-Type-Options header is not set**

canvas.spotify.com



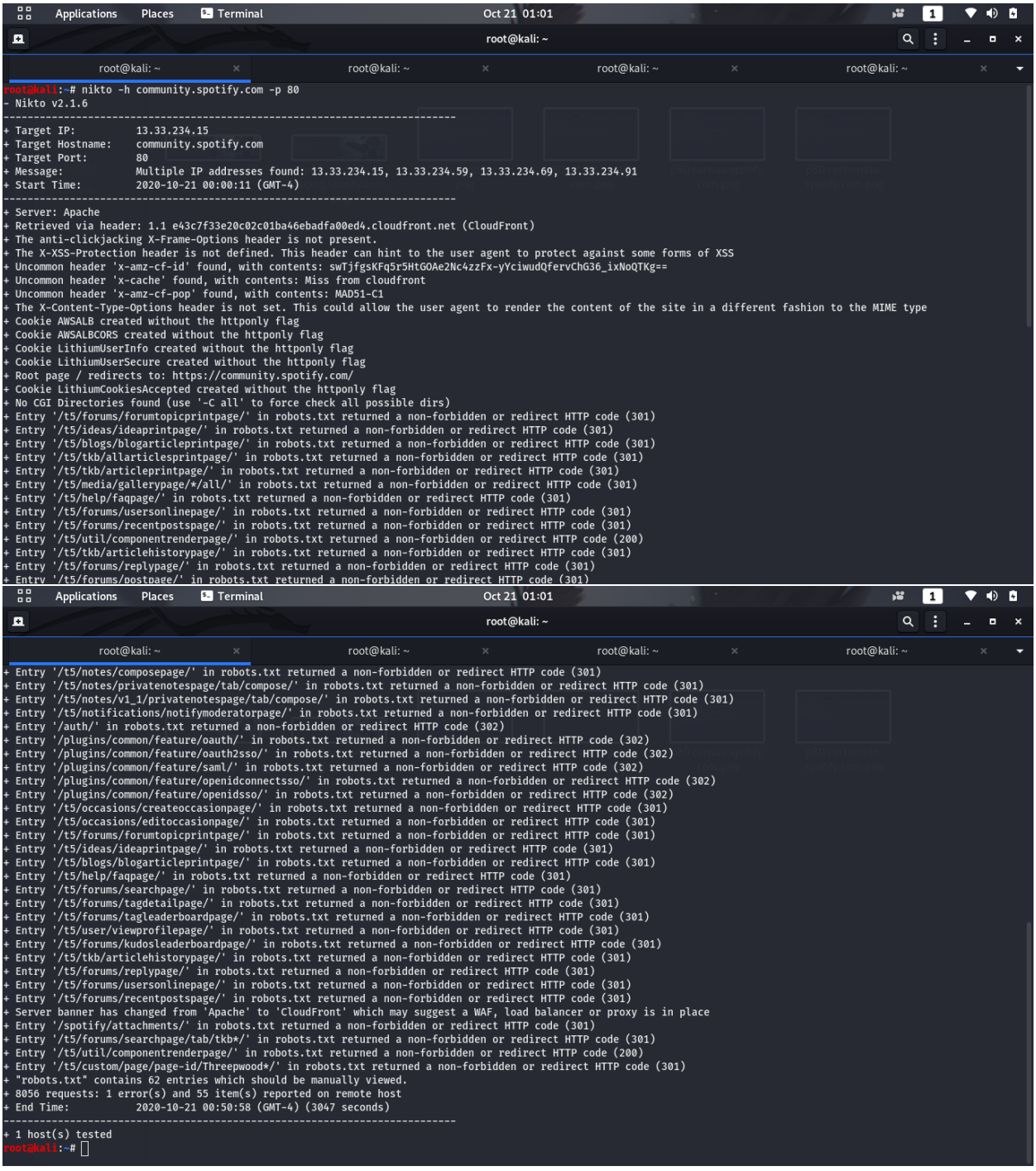
1. **Clickjacking**
2. **X-XSS-Protection header is not defined**
3. **X-Content-Type-Options header is not set**

certomato.spotify.com

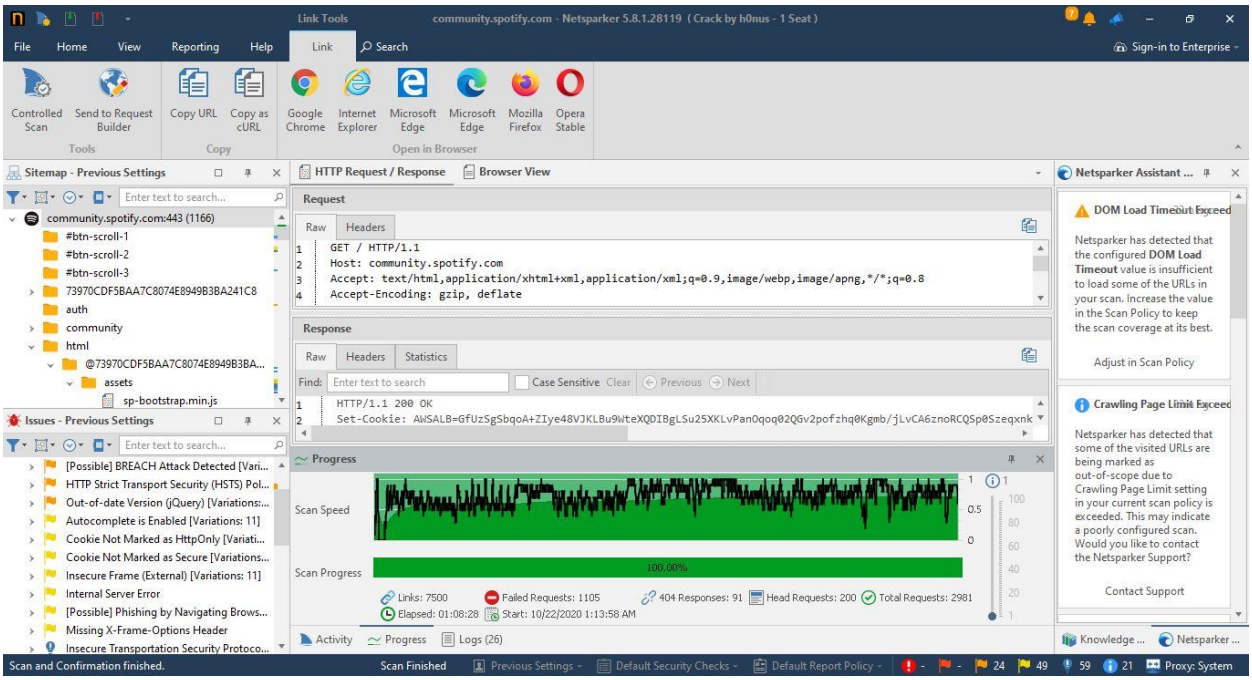


1. **Clickjacking**
2. **X-XSS-Protection header is not defined**
3. **X-Content-Type-Options header is not set**

community.spotify.com

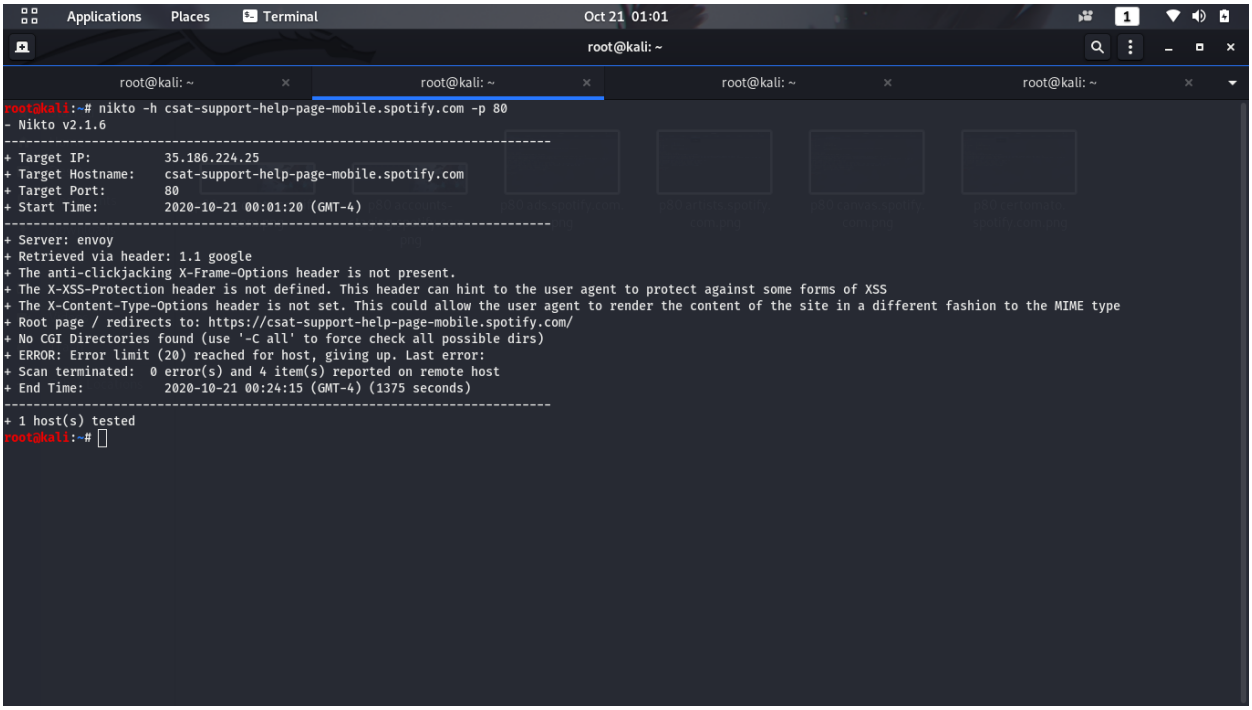


Since this report is hard to understand, I used Netspaker Pro to scan this subdomain again.



1. **BREACH Attack (Possible)**
2. **HTTP Strict Transport Security (HSTS) Policy Not Enabled**
3. **Out-of-date Version (jQuery)**
4. **Autocomplete is Enabled**
5. **Cookie Not Marked as Http Only**
6. **Cookie Not Marked as Secure**
7. **Insecure Frame (External)**
8. **Missing X-Frame-Options Header**

csat-support-help-page-mobile.spotify.com

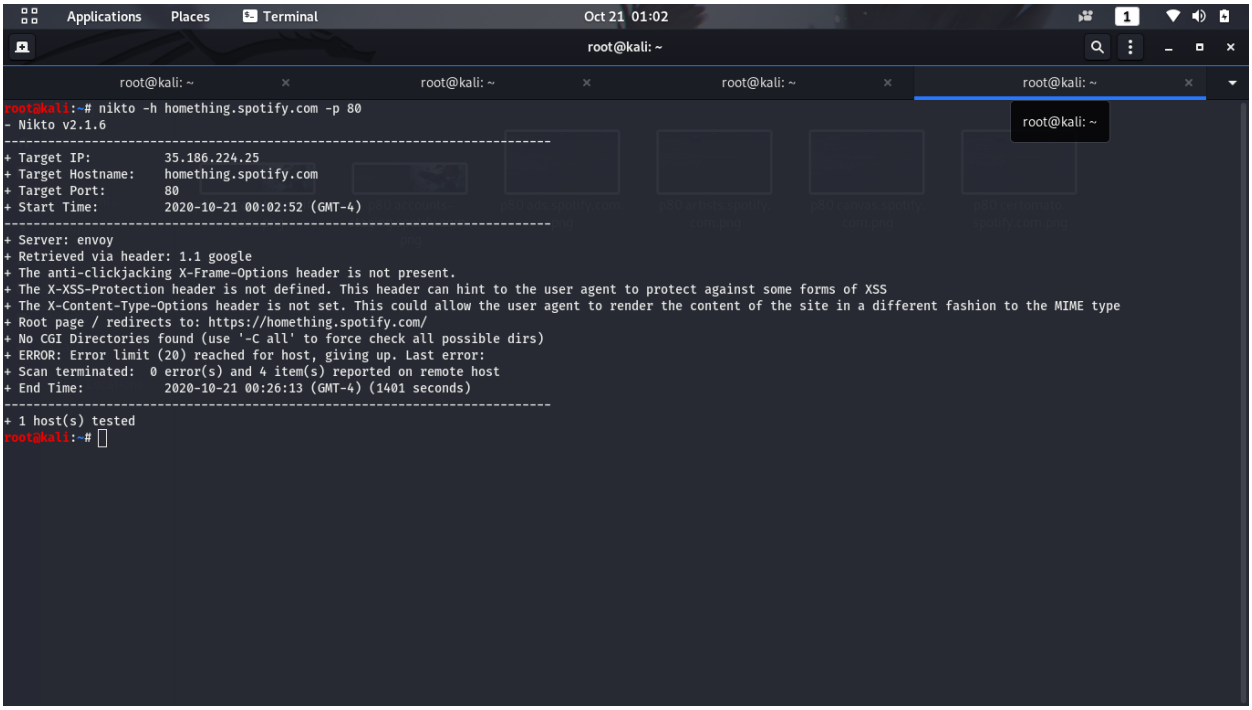


1. **Clickjacking**
2. **X-XSS-Protection header is not defined**
3. **X-Content-Type-Options header is not set**

developer.spotify.com

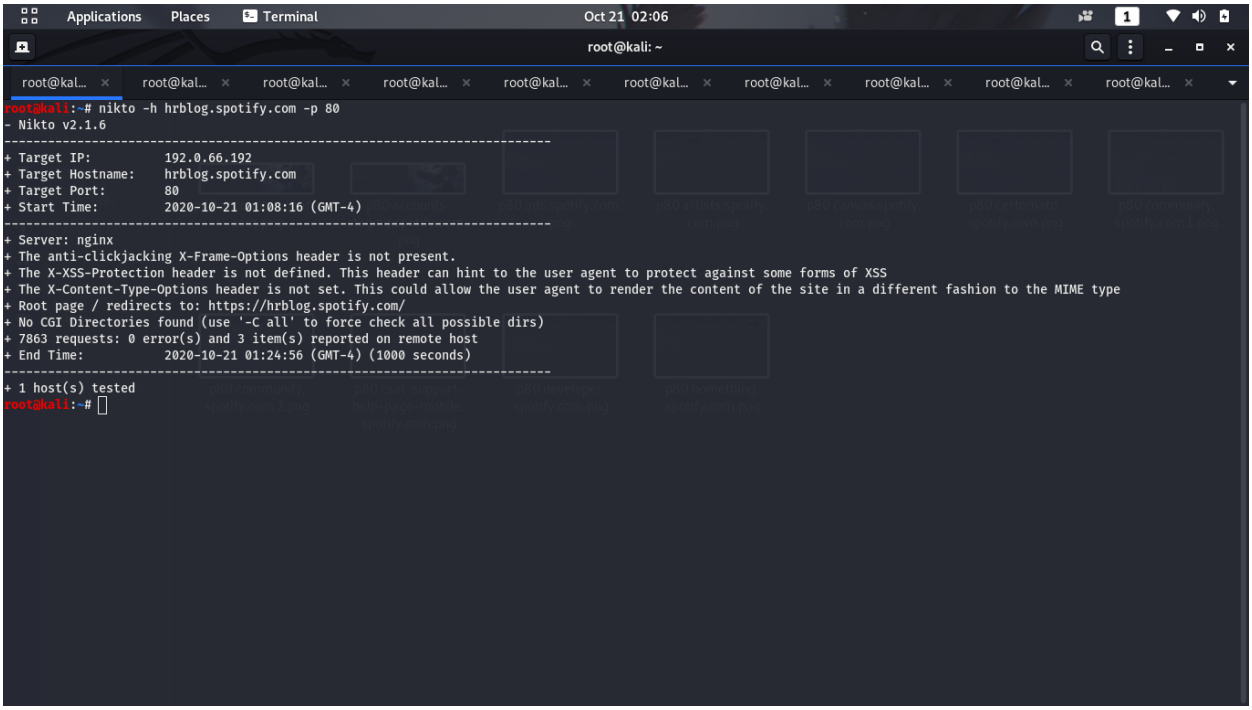
1. **Clickjacking**
2. **X-XSS-Protection header is not defined**
3. **X-Content-Type-Options header is not set**

homething.spotify.com



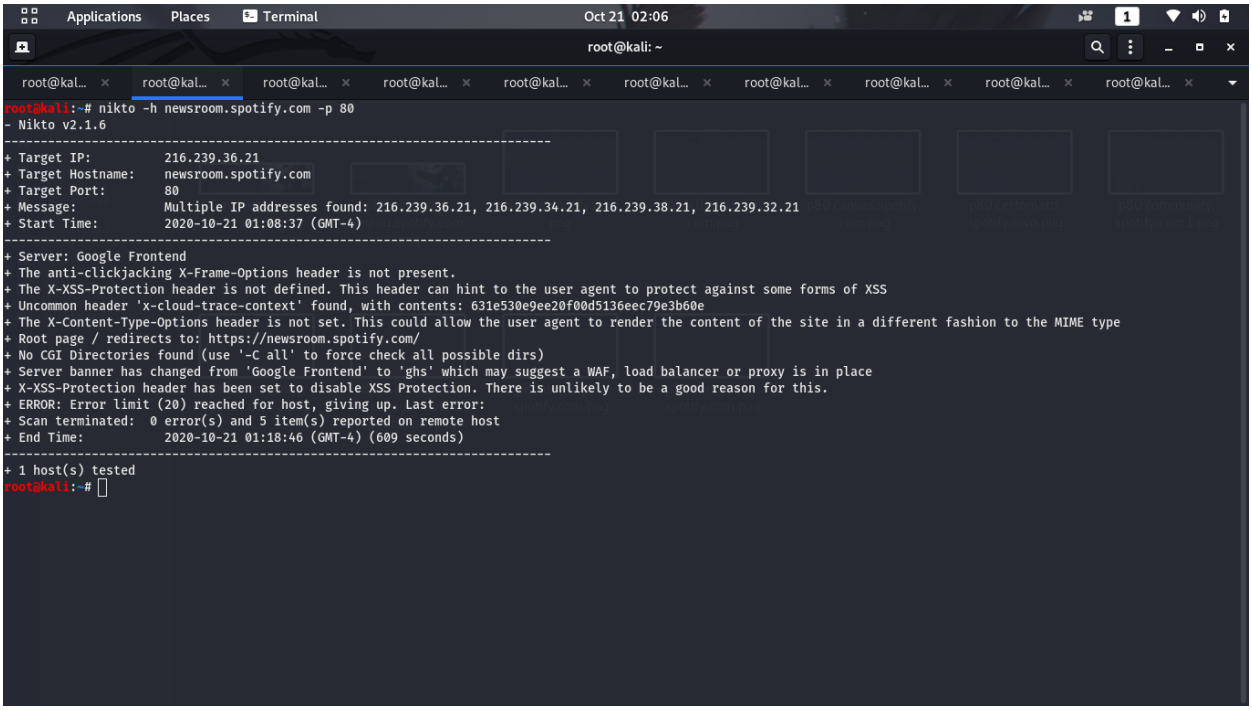
1. **Clickjacking**
2. **X-XSS-Protection header is not defined**
3. **X-Content-Type-Options header is not set**

hrblog.spotify.com



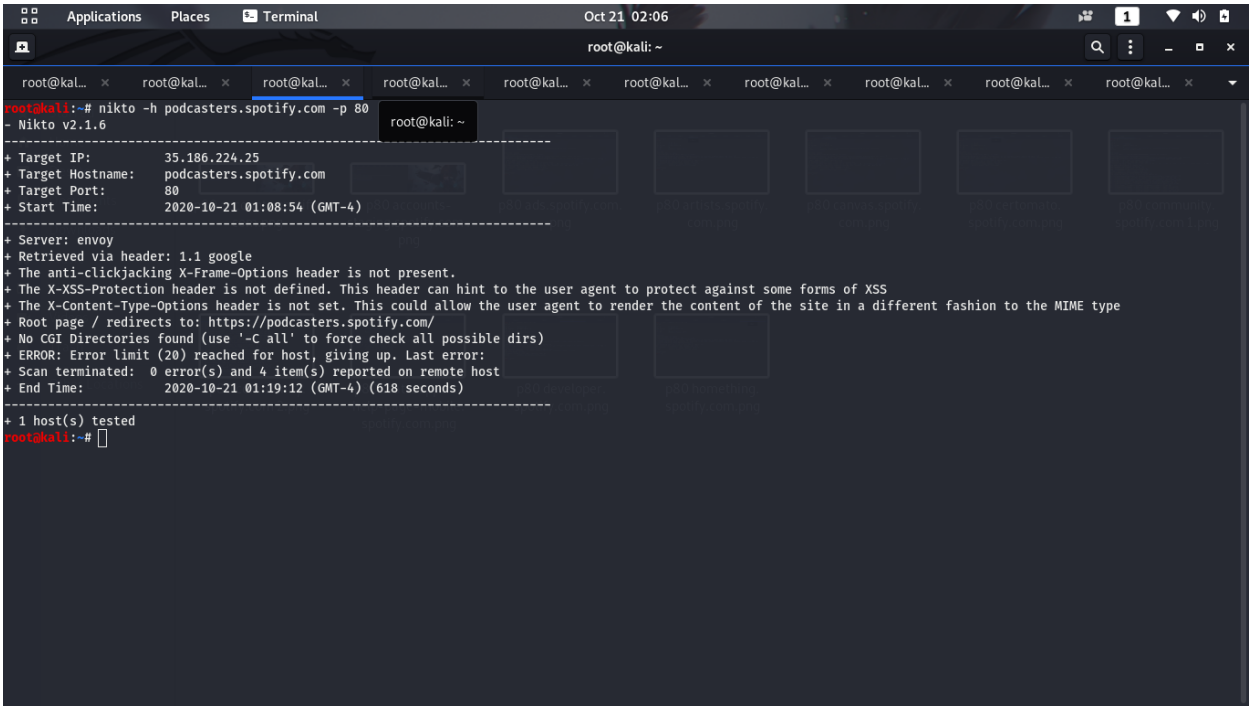
1. **Clickjacking**
2. **X-XSS-Protection header is not defined**
3. **X-Content-Type-Options header is not set**

newsroom.spotify.com



1. **Clickjacking**
2. **X-XSS-Protection header is not defined**
3. **X-Content-Type-Options header is not set**

podcasters.spotify.com

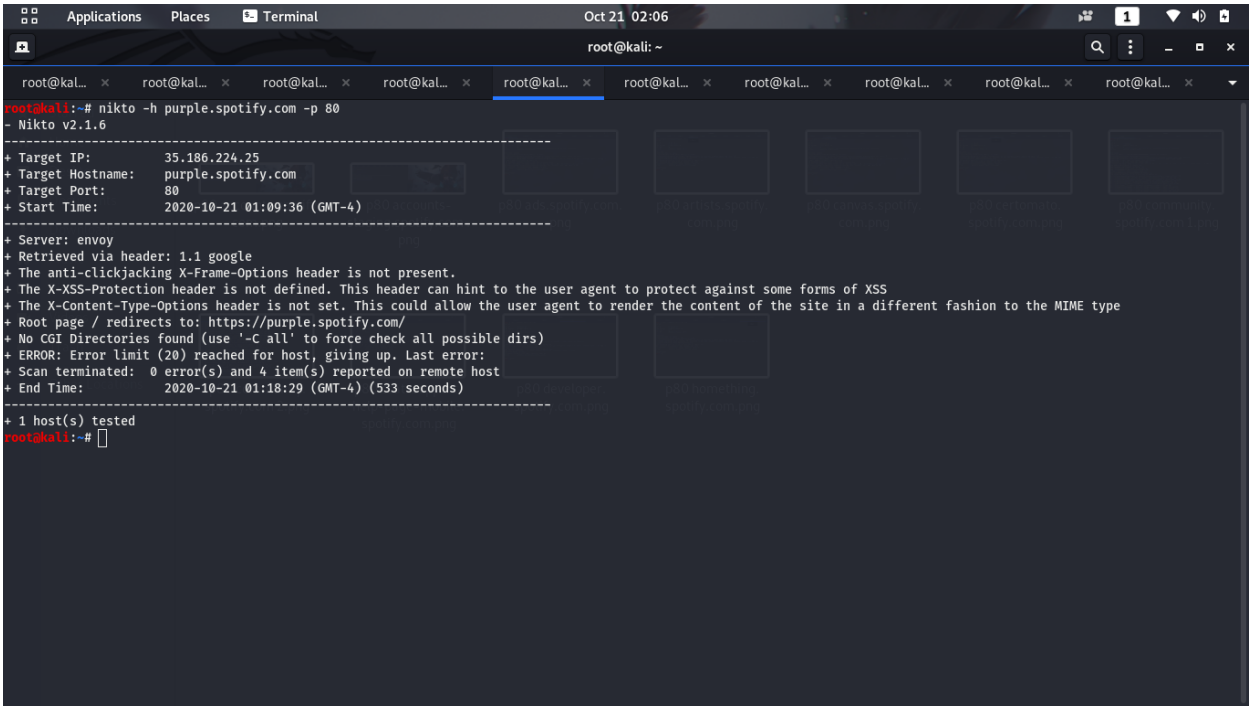


1. **Clickjacking**
2. **X-XSS-Protection header is not defined**
3. **X-Content-Type-Options header is not set**

promo.spotify.com

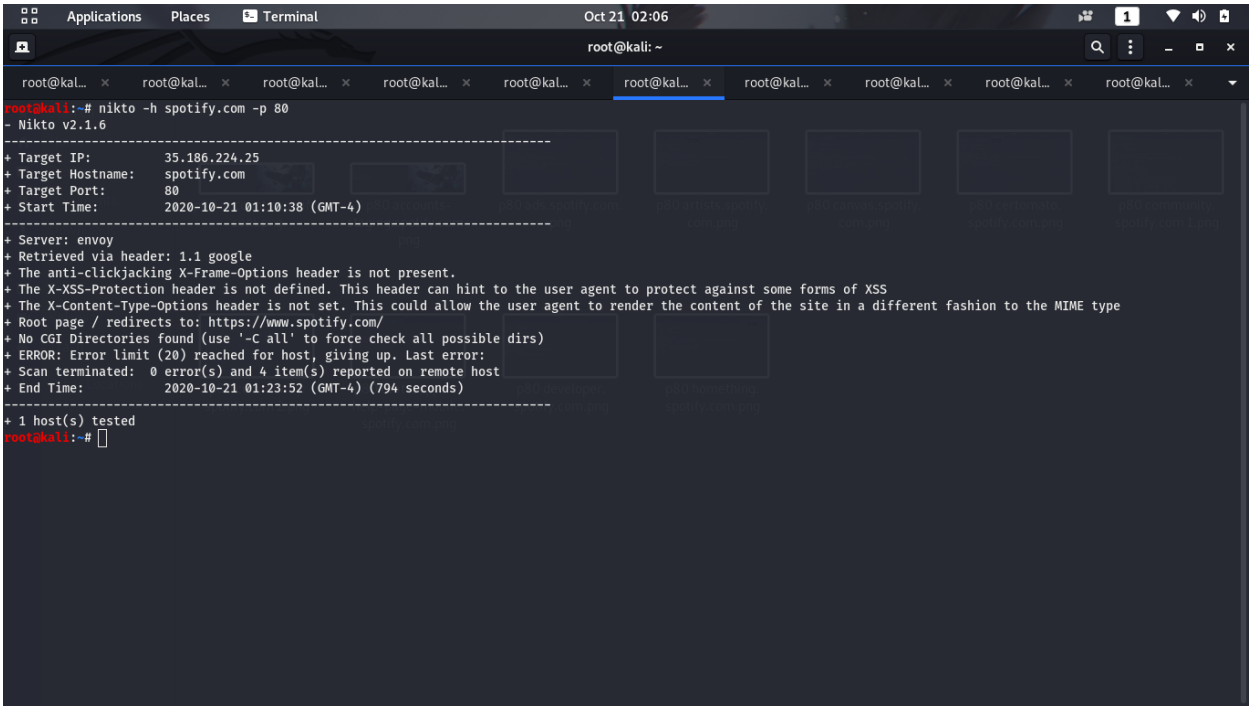
1. **Clickjacking**
2. **X-XSS-Protection header is not defined**
3. **X-Content-Type-Options header is not set**

purple.spotify.com



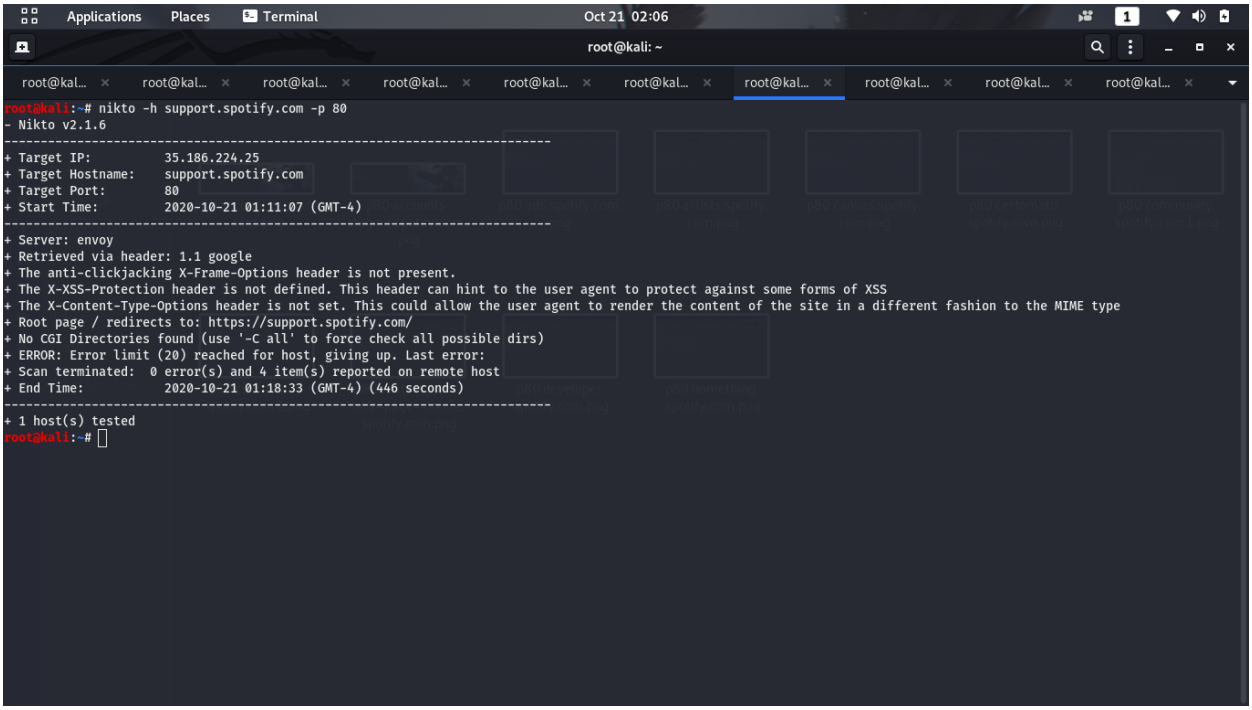
1. **Clickjacking**
2. **X-XSS-Protection header is not defined**
3. **X-Content-Type-Options header is not set**

spotify.com



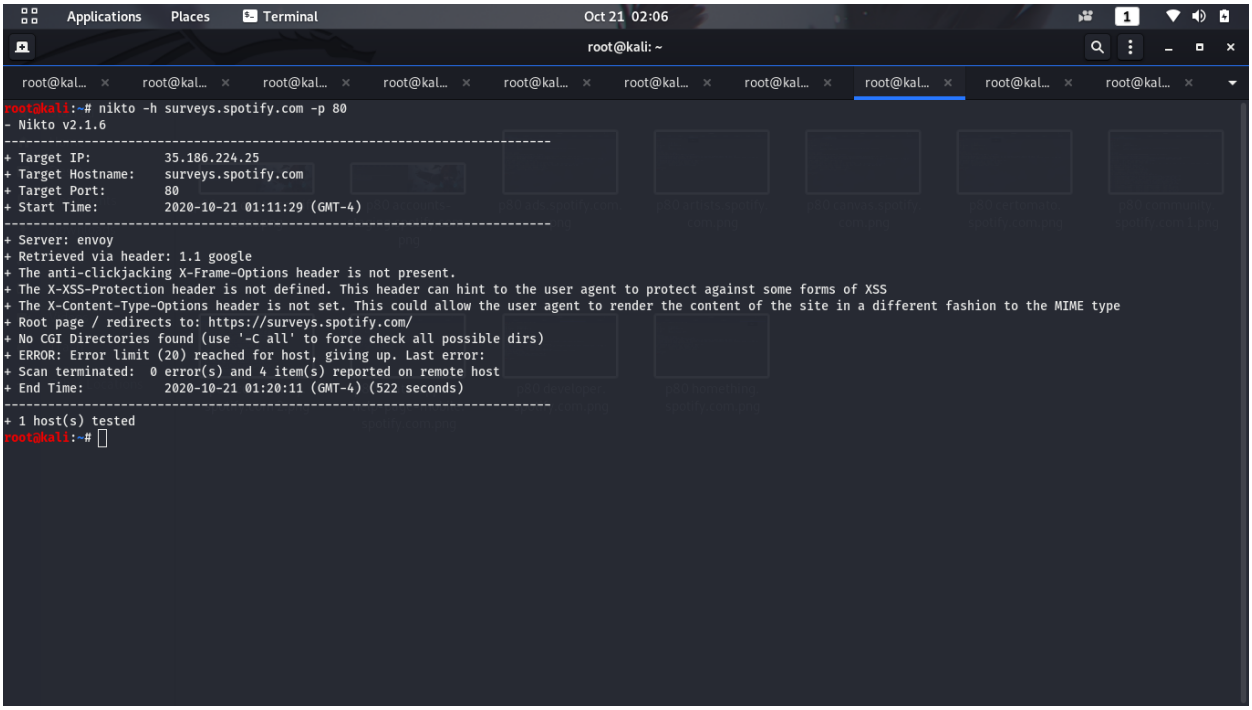
1. **Clickjacking**
2. **X-XSS-Protection header is not defined**
3. **X-Content-Type-Options header is not set**

support.spotify.com



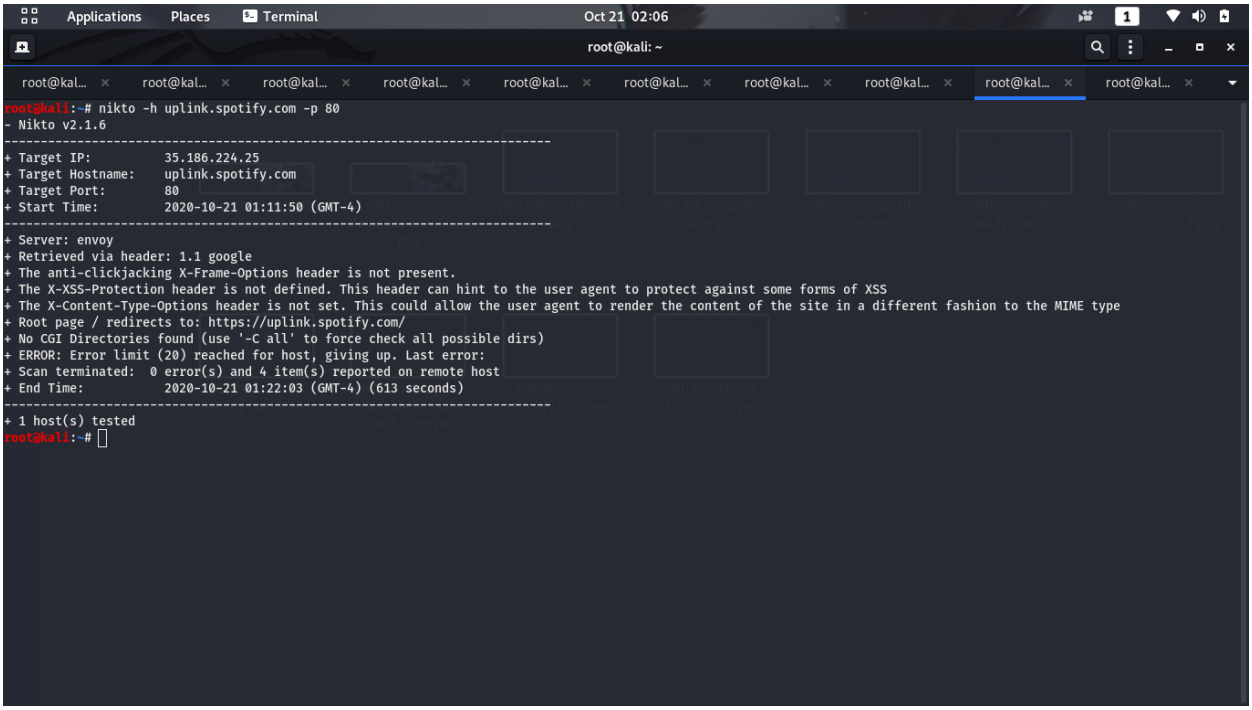
1. **Clickjacking**
2. **X-XSS-Protection header is not defined**
3. **X-Content-Type-Options header is not set**

surveys.spotify.com



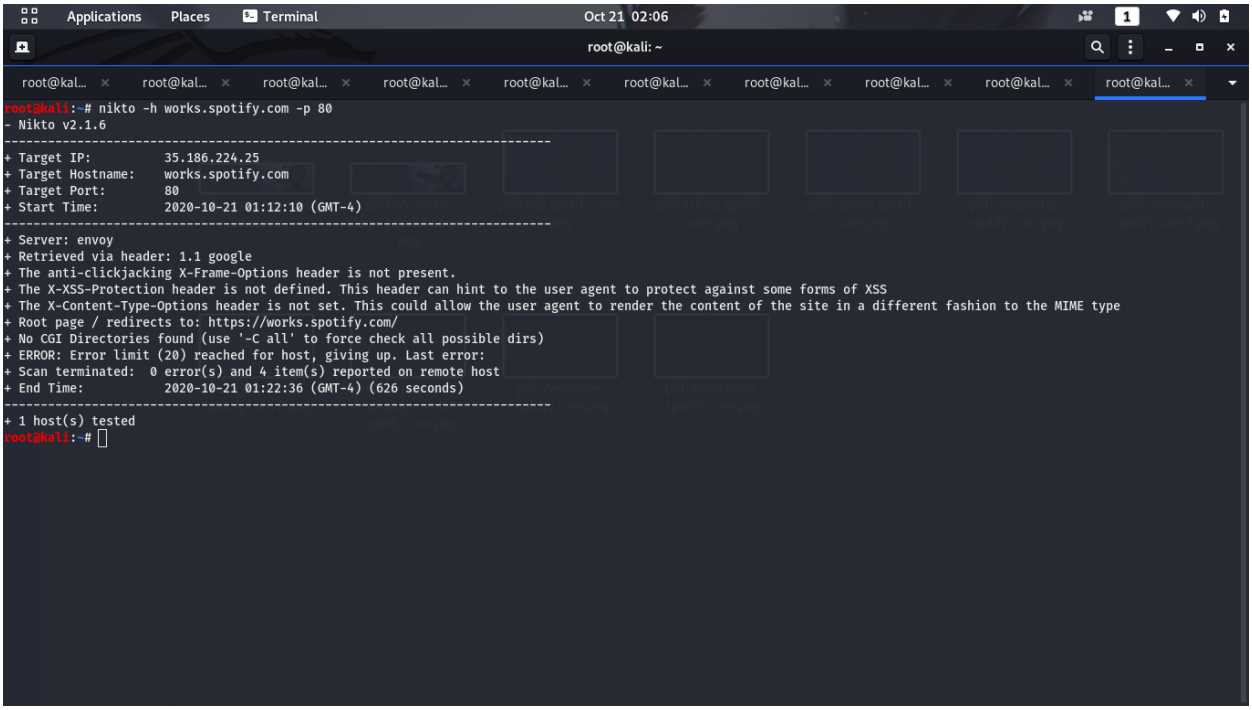
1. **Clickjacking**
2. **X-XSS-Protection header is not defined**
3. **X-Content-Type-Options header is not set**

uplink.spotify.com



1. **Clickjacking**
2. **X-XSS-Protection header is not defined**
3. **X-Content-Type-Options header is not set**

works.spotify.com

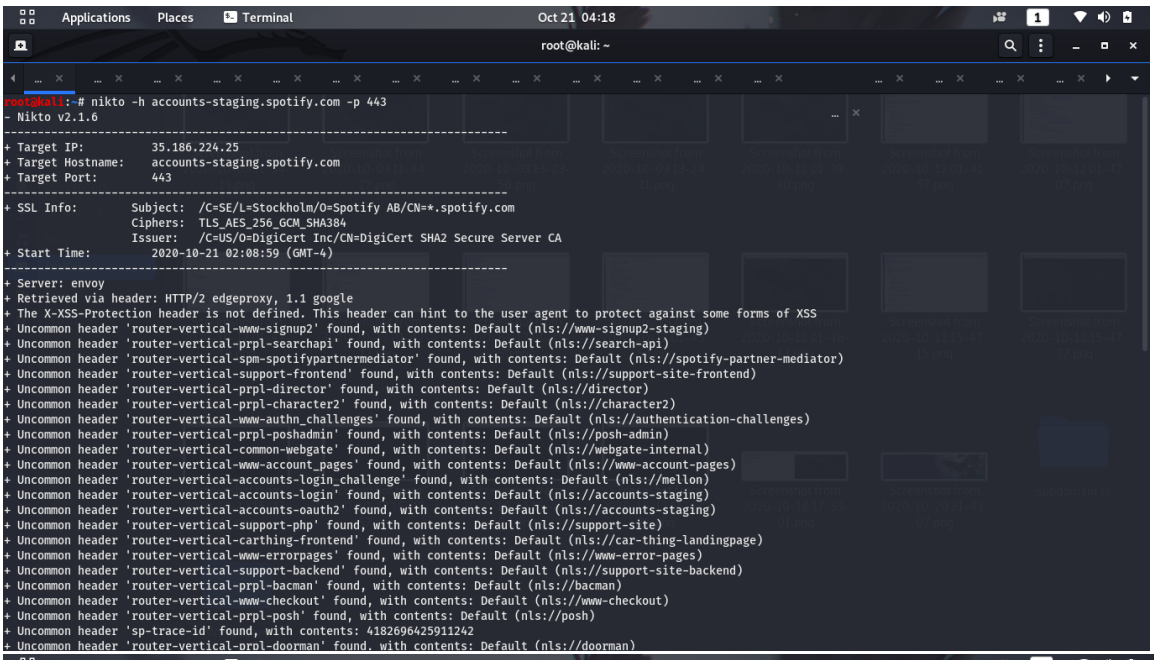


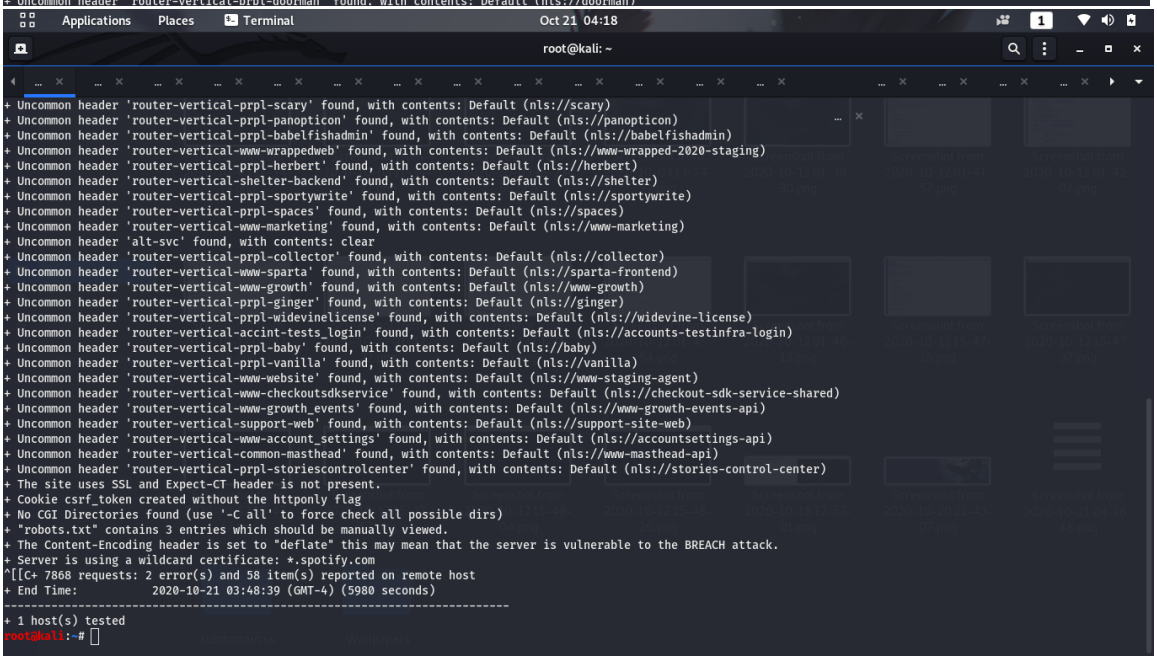
1. **Clickjacking**
2. **X-XSS-Protection header is not defined**
3. **X-Content-Type-Options header is not set**

Scan using Nikto Tool (port 443)

After scanning all subdomains for port 80, I Scanned all of them again for port 443 for further more.

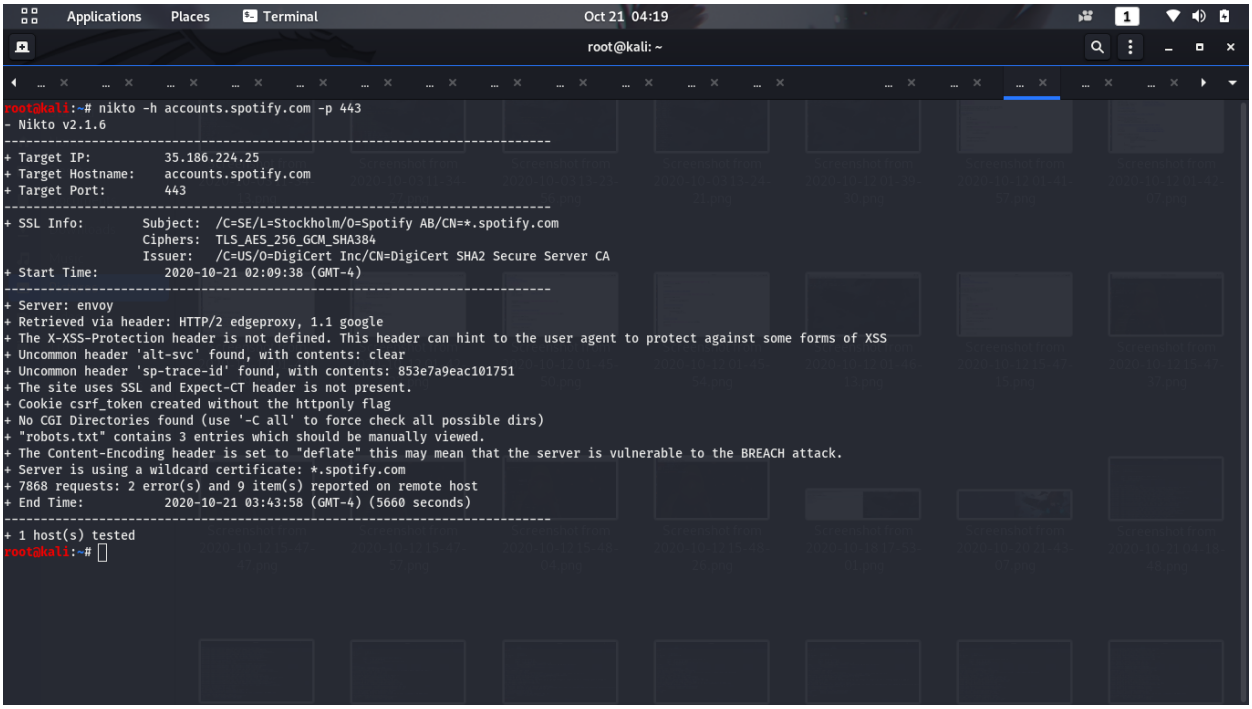
accounts-staging.spotify.com





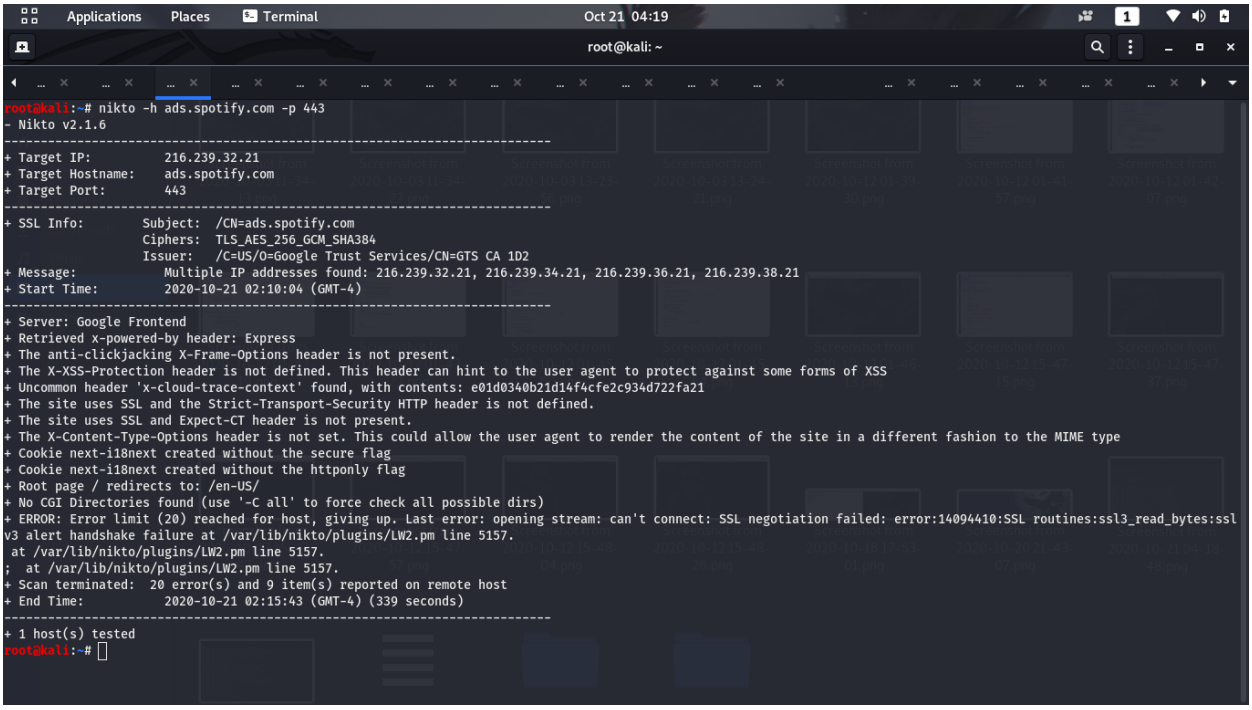
1. **X-XSS-Protection header is not defined**
2. **Server is using Wildcard Certification**

accounts.spotify.com



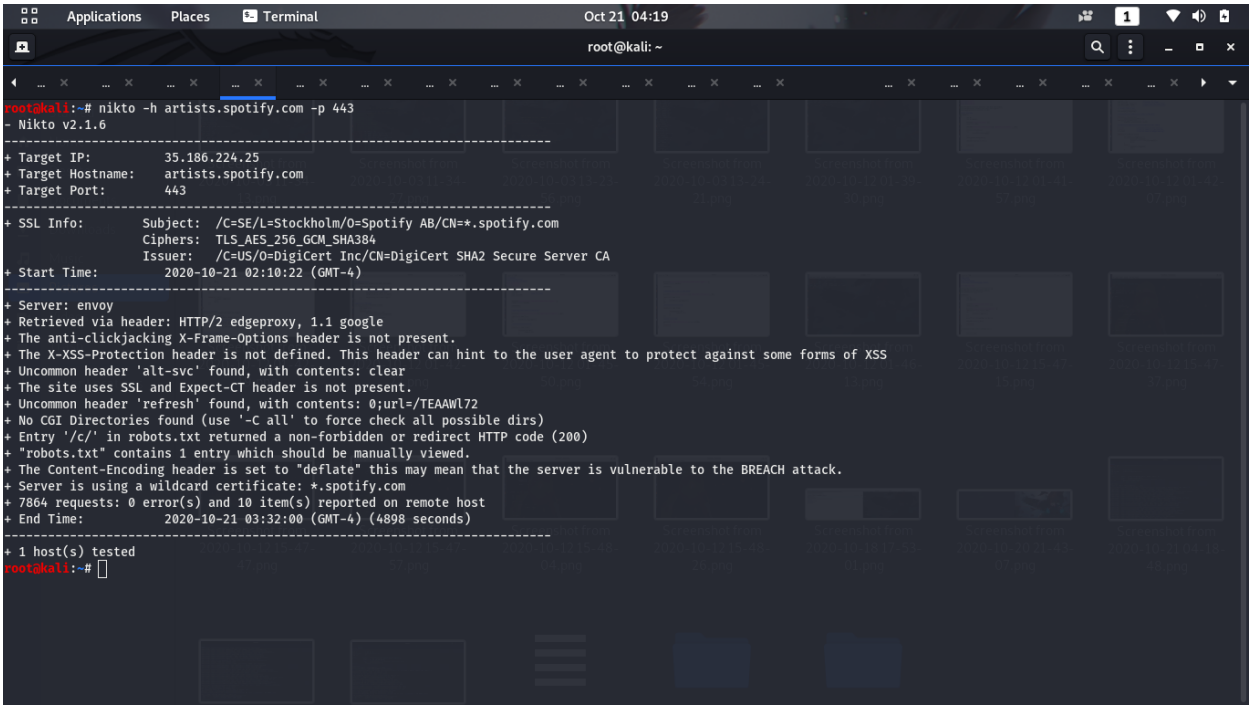
1. **X-XSS-Protection header is not defined**
2. **Server is using Wildcard Certification**

ads.spotify.com



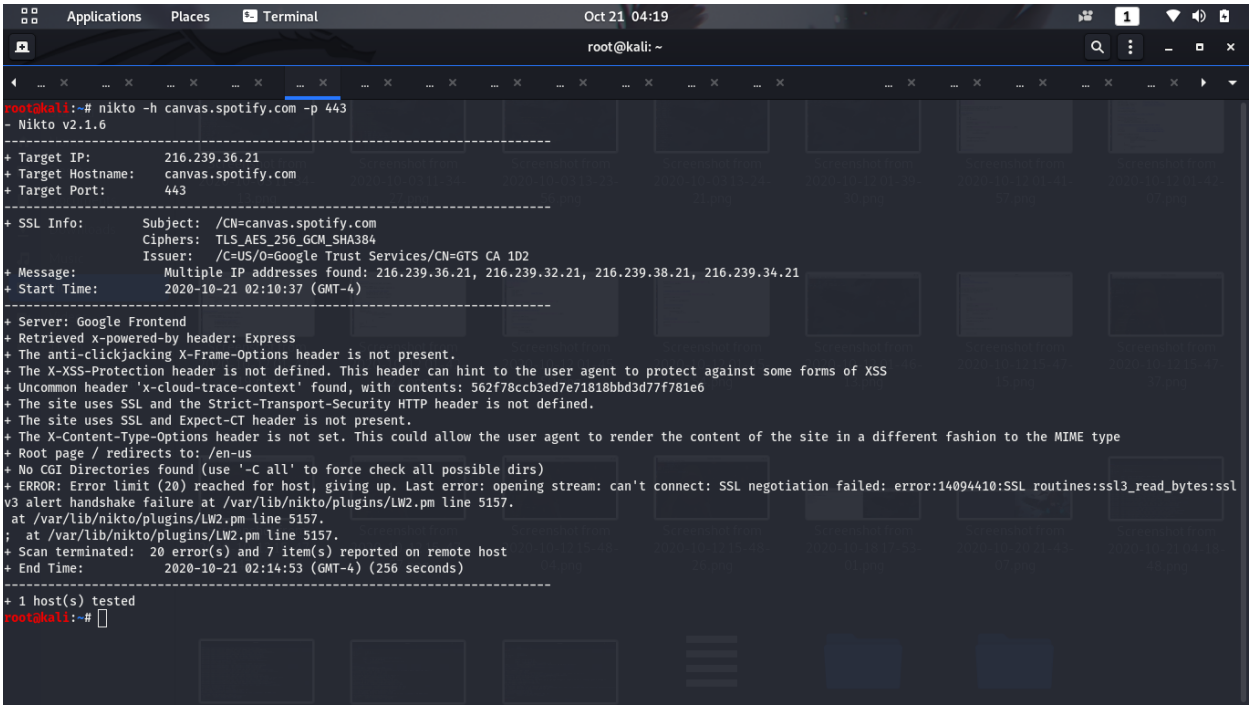
1. **Clickjacking**
2. **X-XSS-Protection header is not defined**
3. **X-Content-Type-Options header is not set**

artists.spotify.com



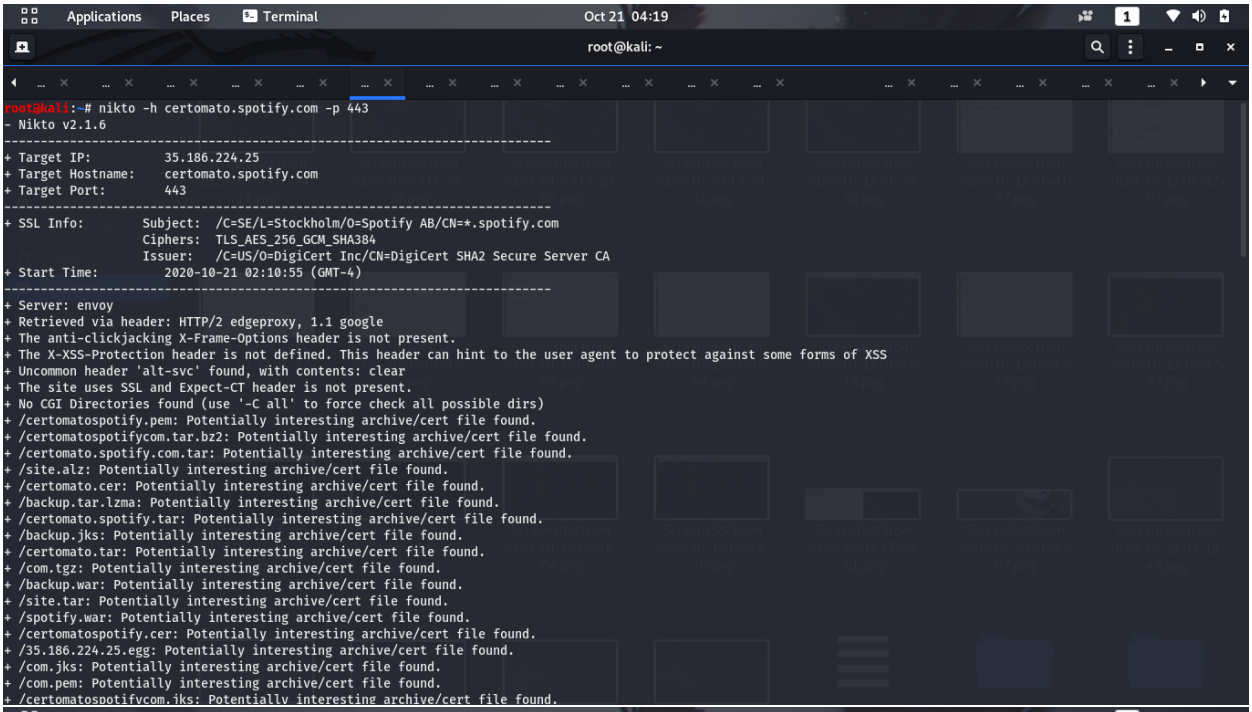
1. **Clickjacking**
2. **X-XSS-Protection header is not defined**
3. **Server is using Wildcard Certification**

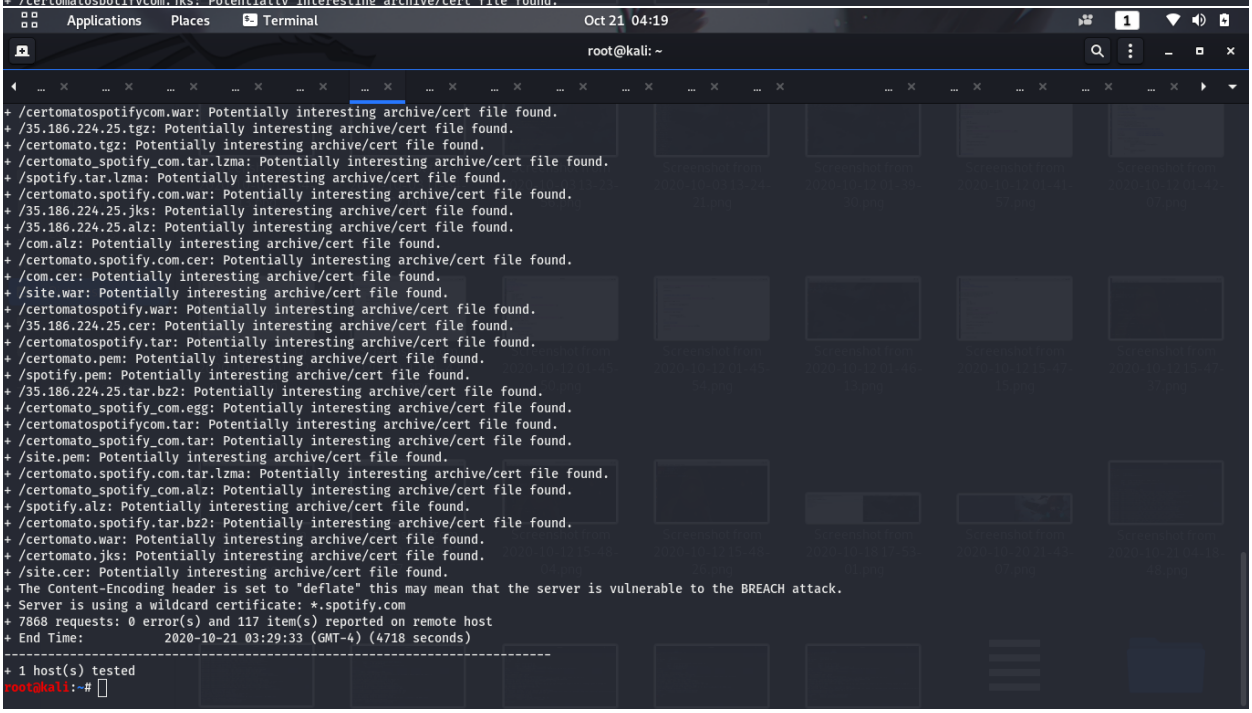
canvas.spotify.com



1. **Clickjacking**
2. **X-XSS-Protection header is not defined**
3. **X-Content-Type-Options header is not set**

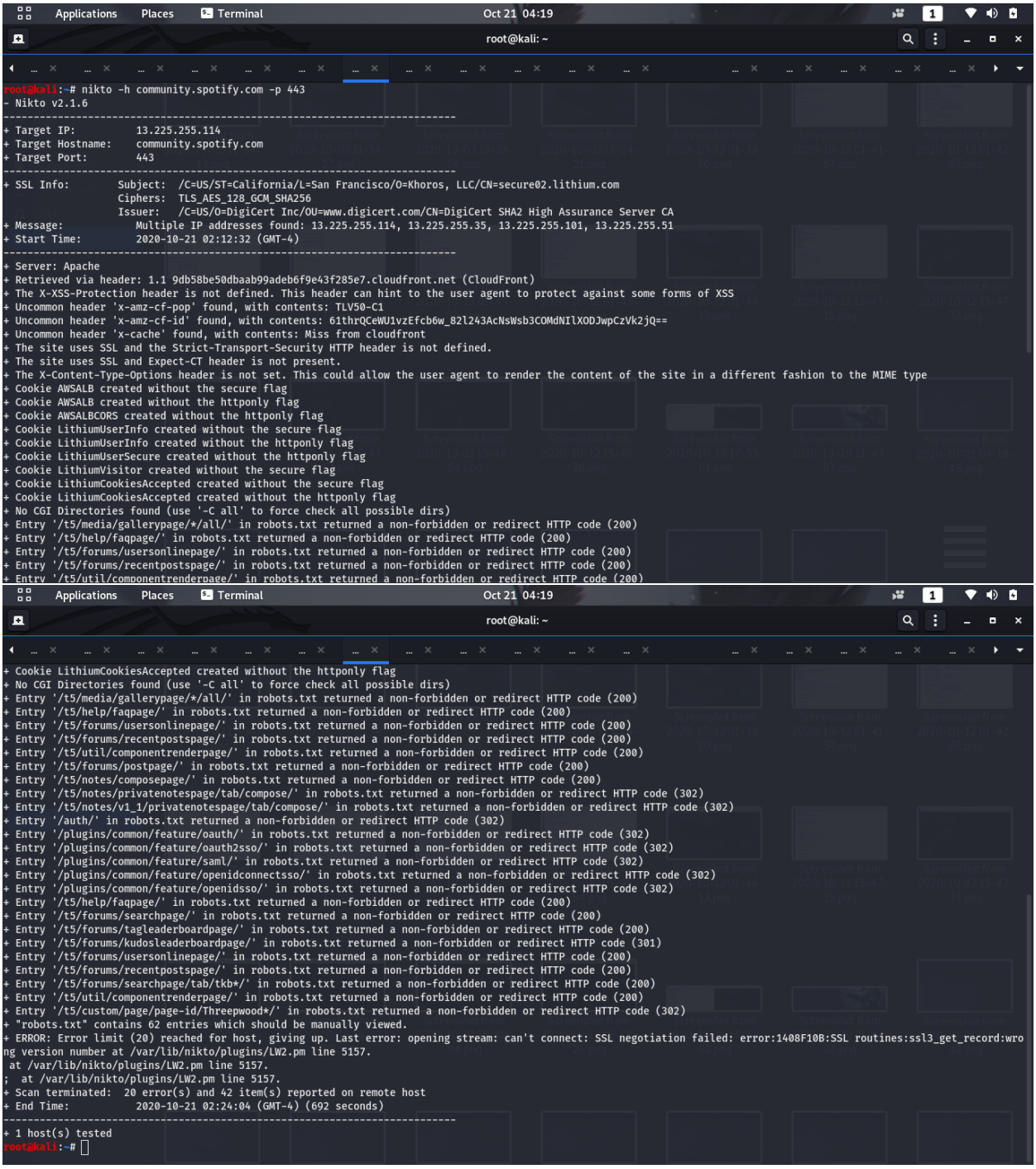
certomato.spotify.com





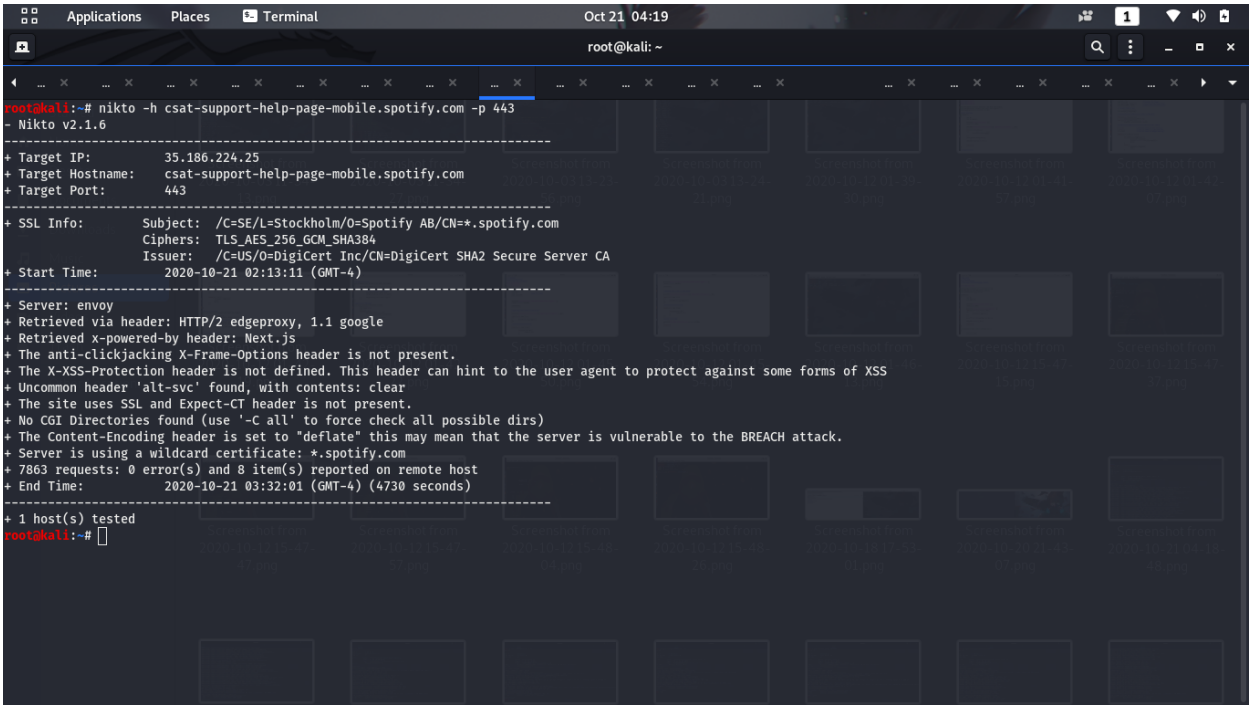
1. **Clickjacking**
2. **X-XSS-Protection header is not defined**
3. **Server is using Wildcard Certification**

community.spotify.com



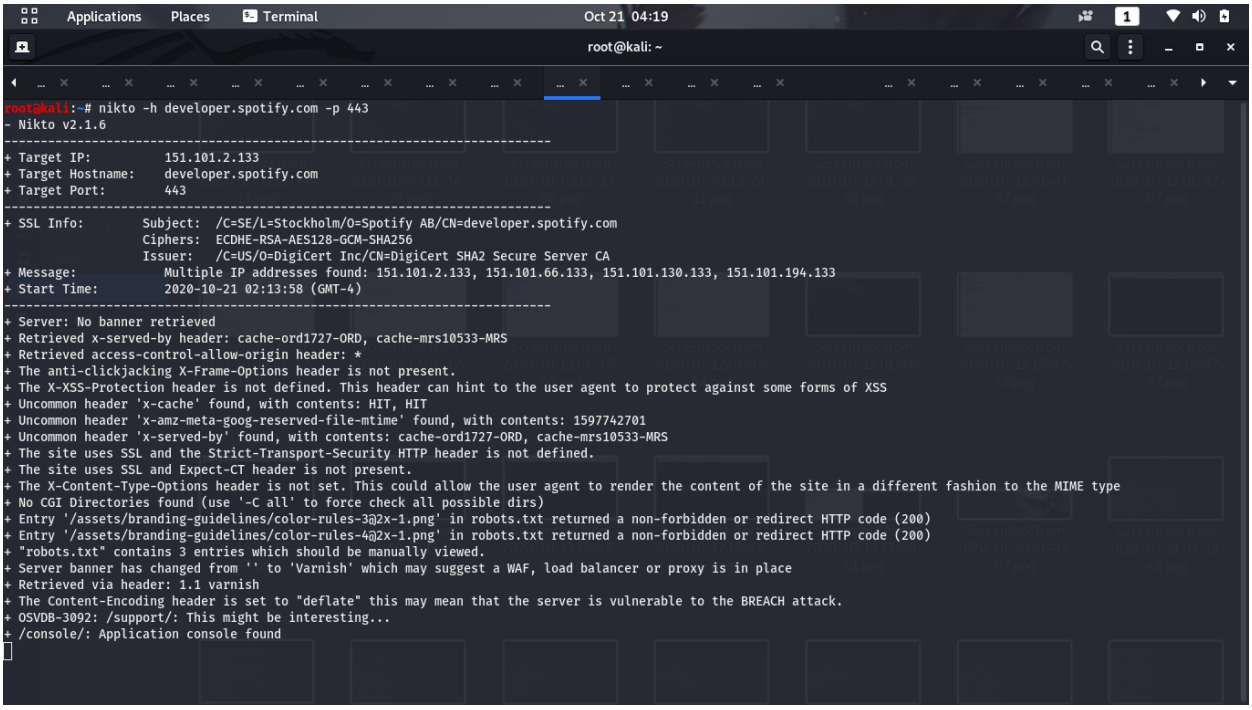
1. **X-XSS-Protection header is not defined**
2. **X-Content-Type-Options header is not set**

csat-support-help-page-mobile.spotify.com



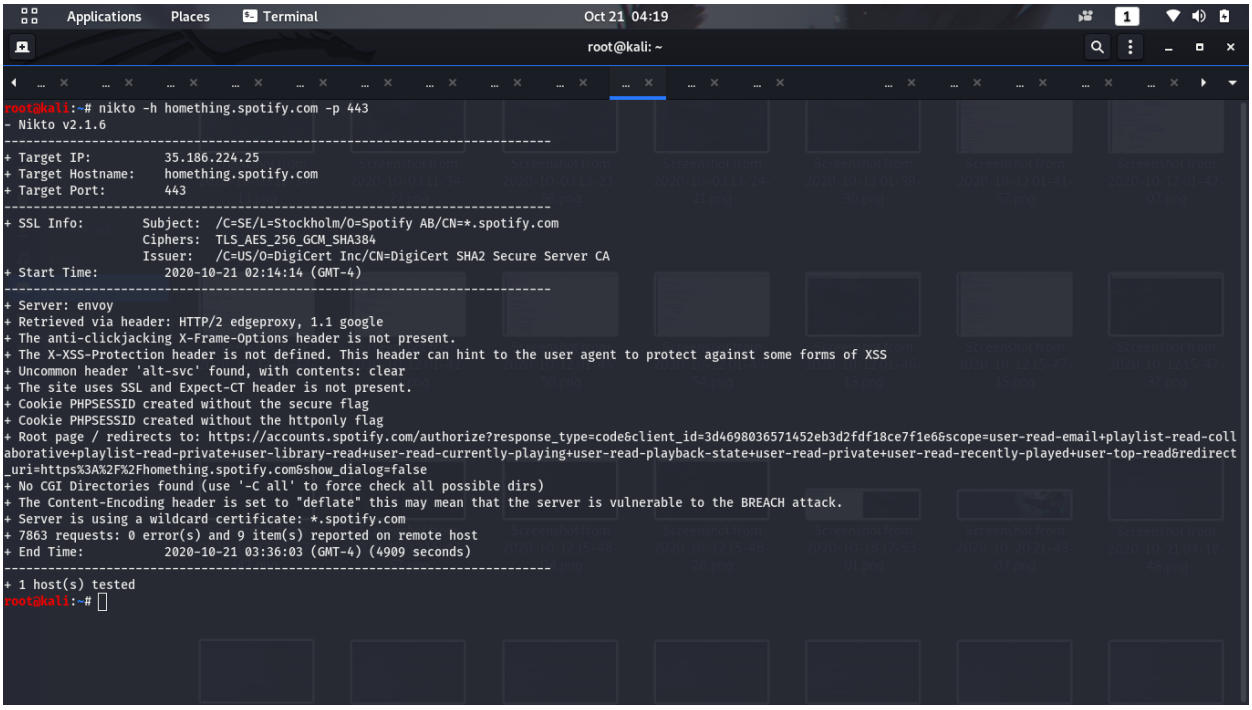
1. **Clickjacking**
2. **X-XSS-Protection header is not defined**
3. **Server is using Wildcard Certification**

developer.spotify.com



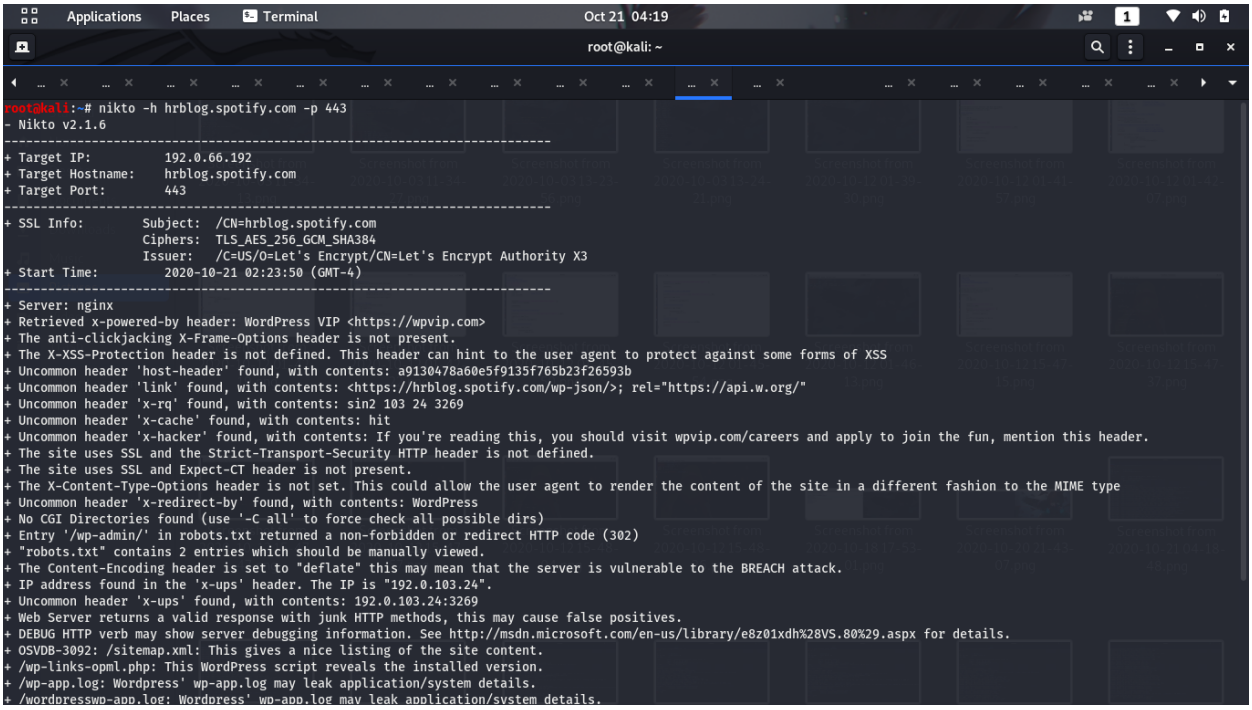
1. **Clickjacking**
2. **X-XSS-Protection header is not defined**

homething.spotify.com



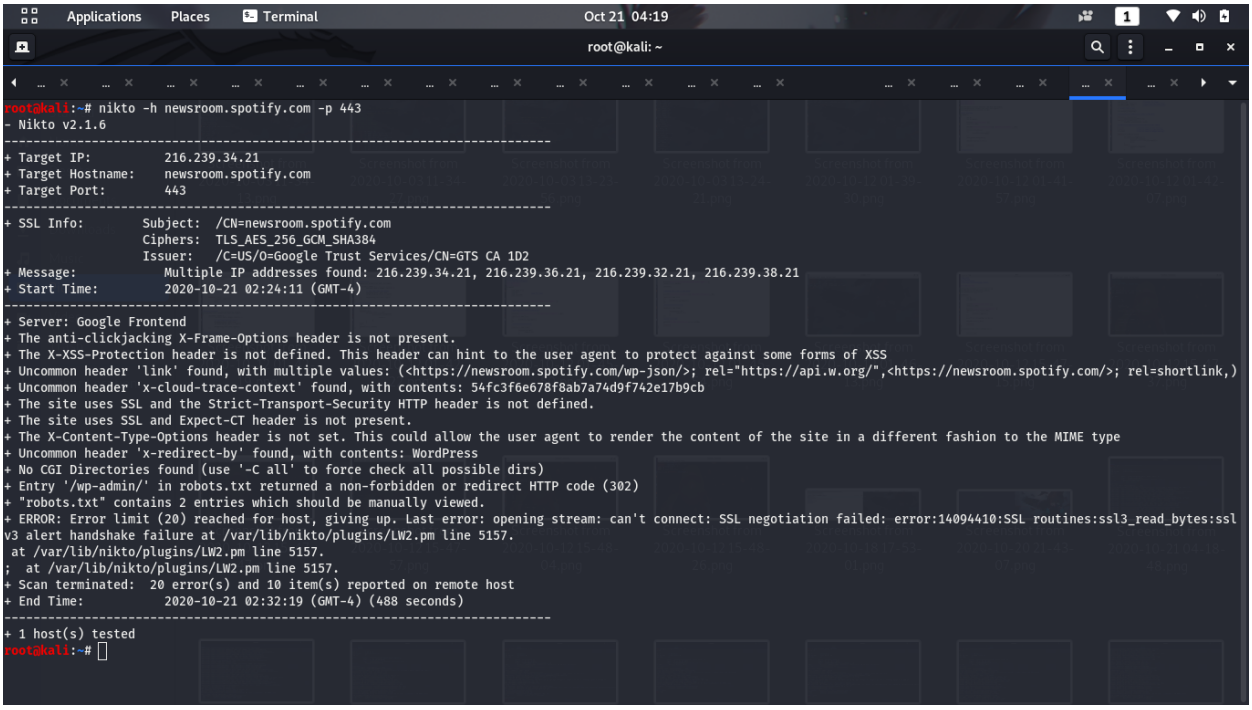
1. **Clickjacking**
2. **X-XSS-Protection header is not defined**
3. **Server is using Wildcard Certification**
4. **cookie PHPSESSID created without the httponly flag**

hrblog.spotify.com



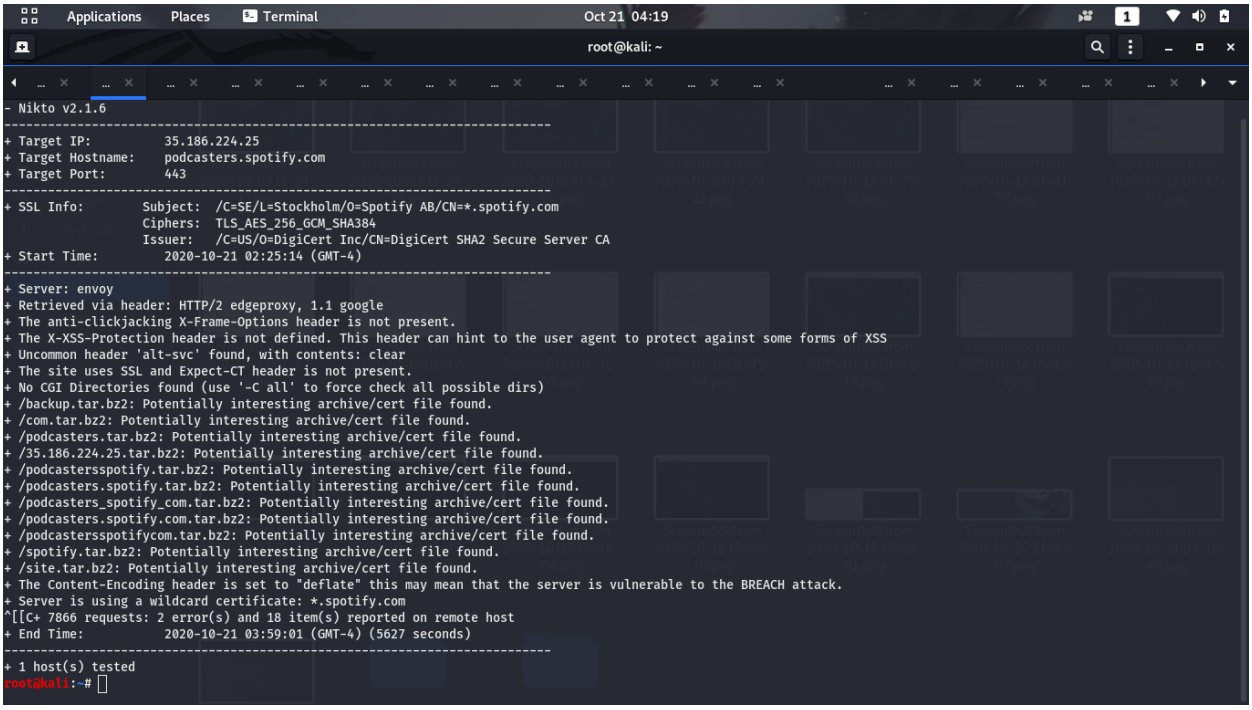
1. **Clickjacking**
2. **X-XSS-Protection header is not defined**
3. **X-Content-Type-Options header is not set**

newsroom.spotify.com



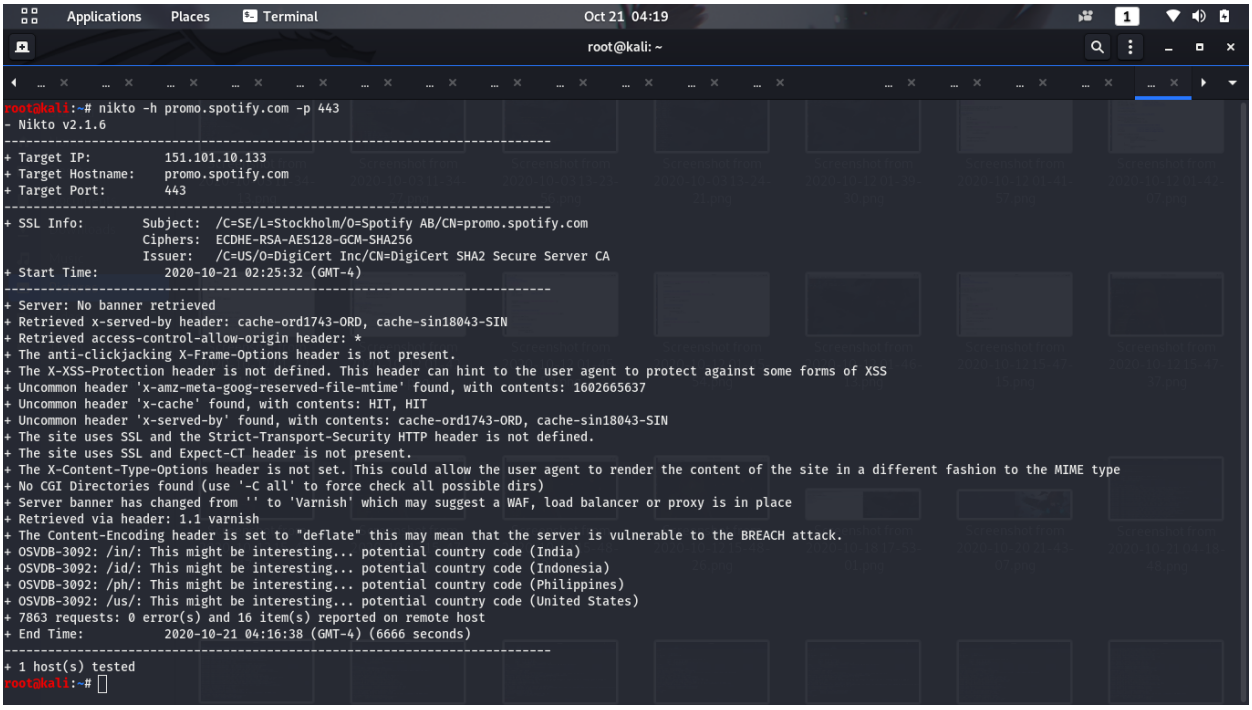
1. **Clickjacking**
2. **X-XSS-Protection header is not defined**
3. **X-Content-Type-Options header is not set**

podcasters.spotify.com



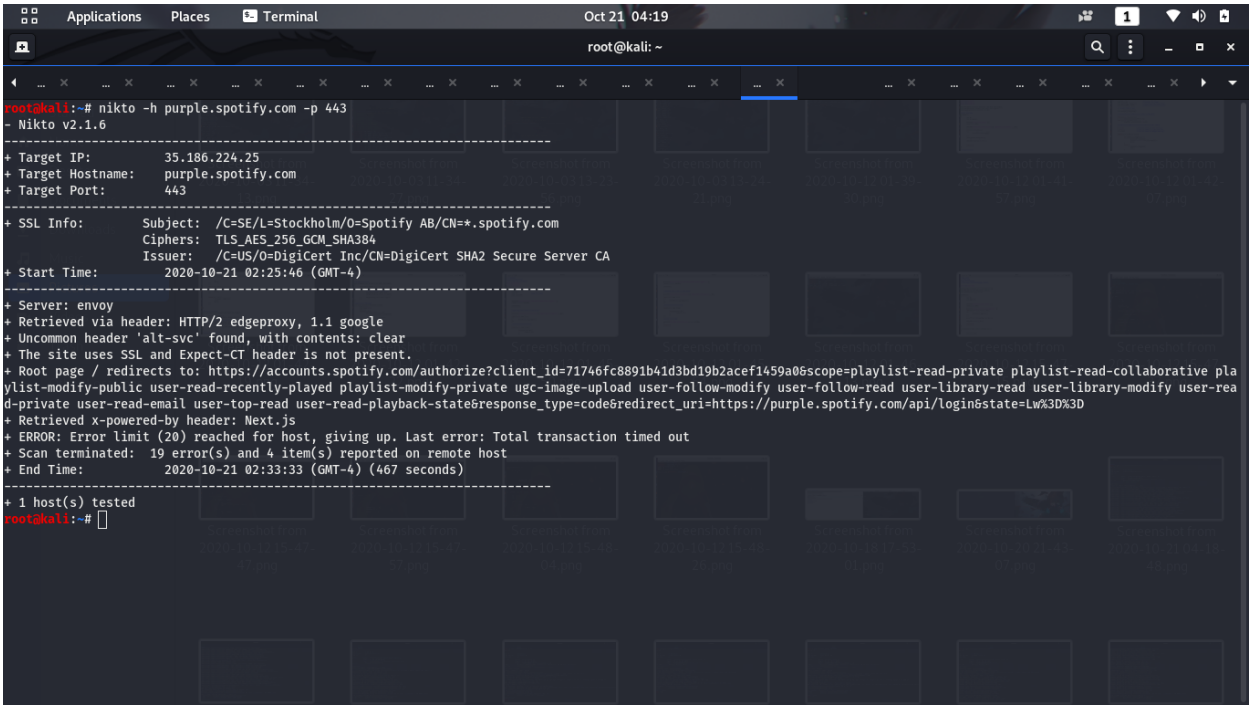
1. **Clickjacking**
2. **X-XSS-Protection header is not defined**
3. **Server is using Wildcard Certification**

promo.spotify.com

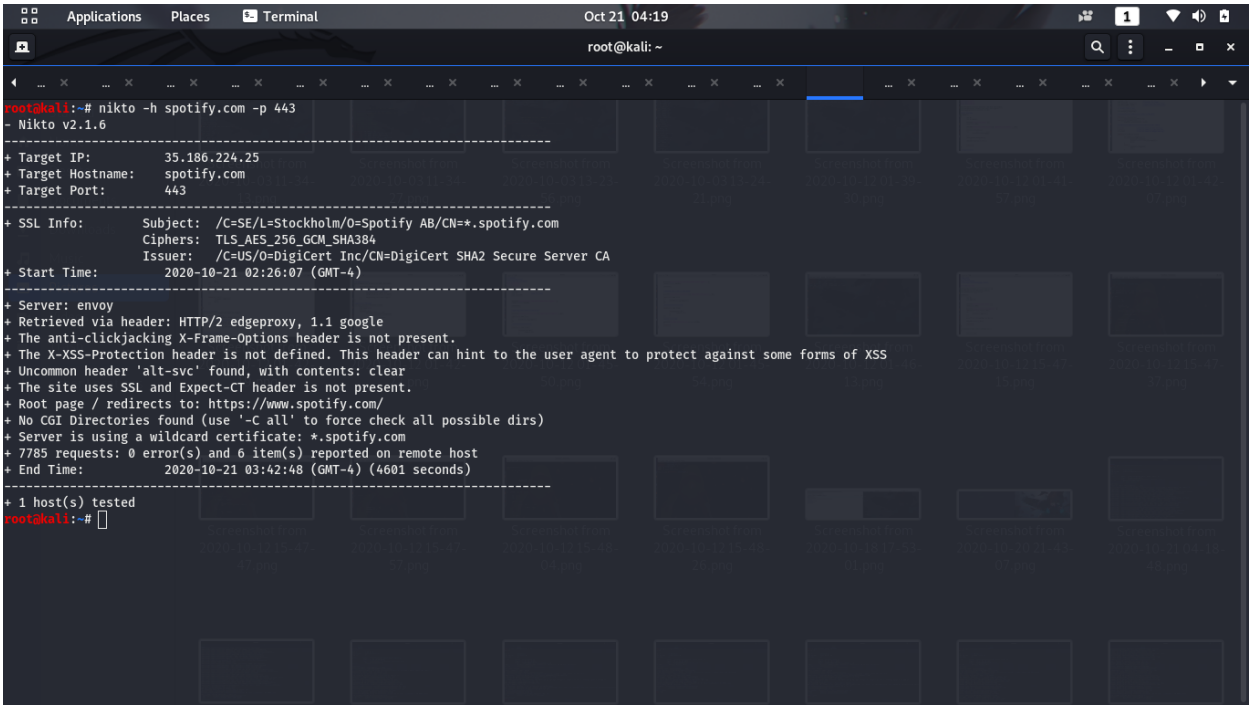


1. **Clickjacking**
2. **X-XSS-Protection header is not defined**
3. **X-Content-Type-Options header is not set**

purple.spotify.com

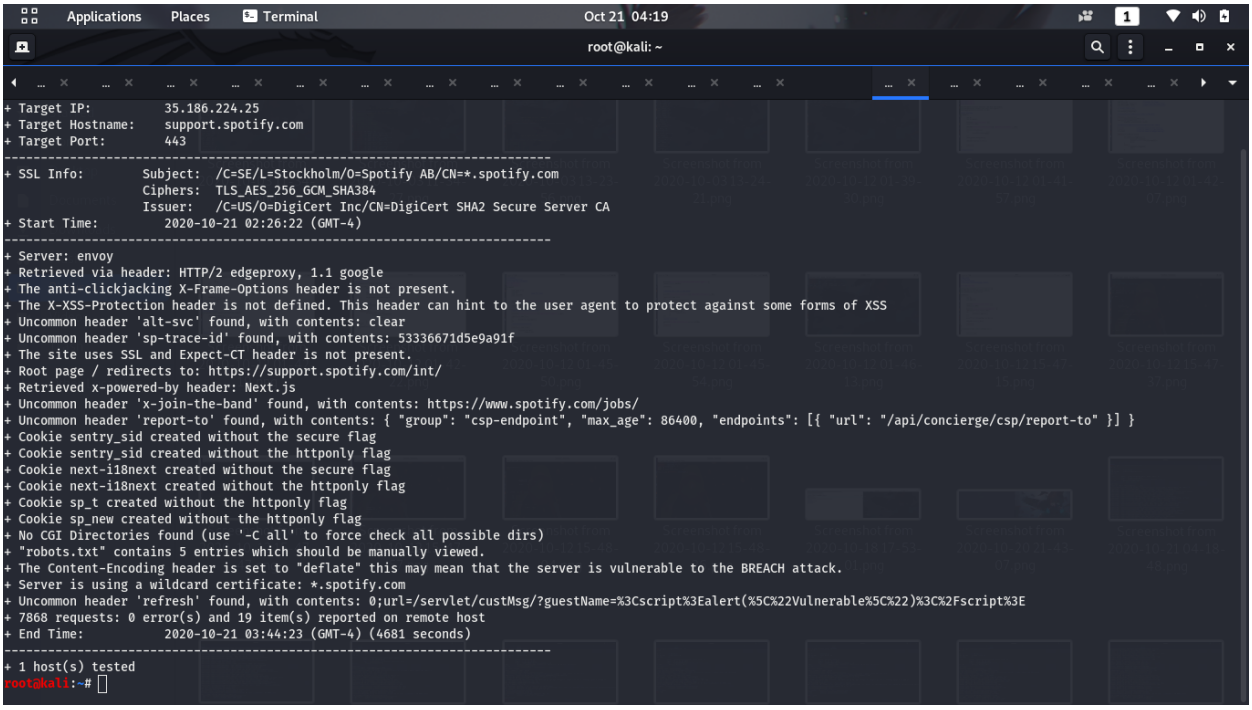


spotify.com



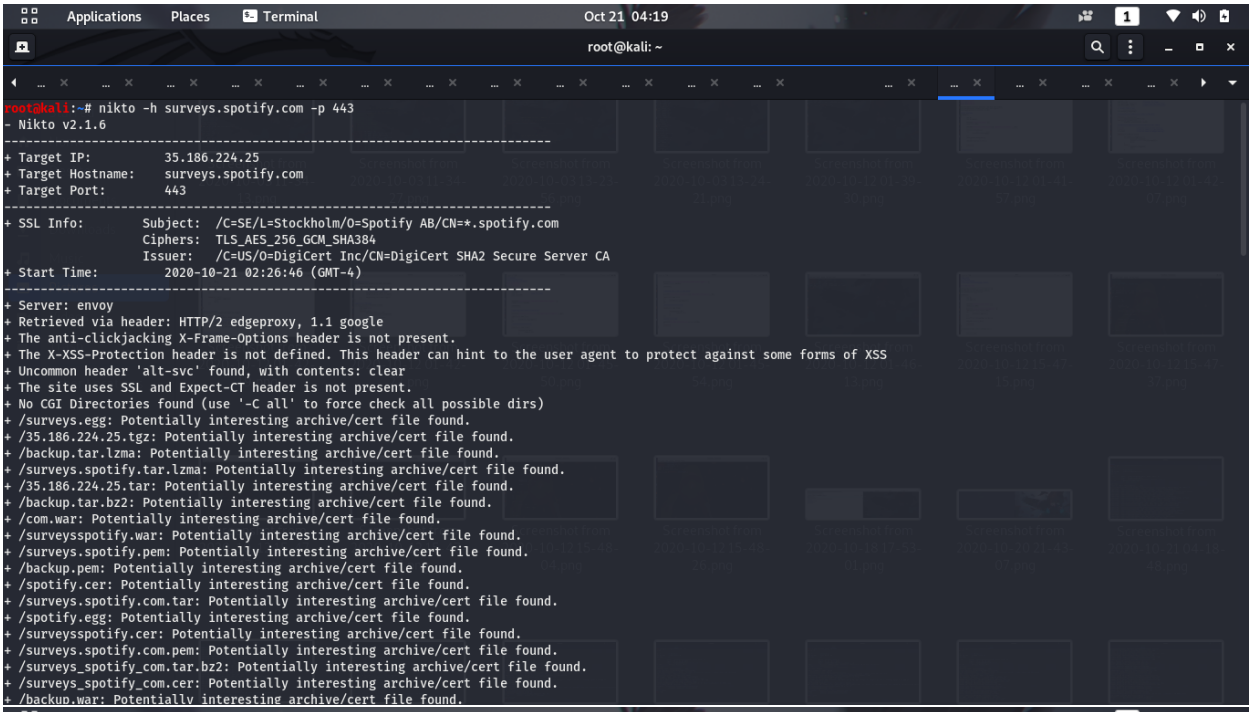
1. **Clickjacking**
2. **X-XSS-Protection header is not defined**
3. **Server is using Wildcard Certification**

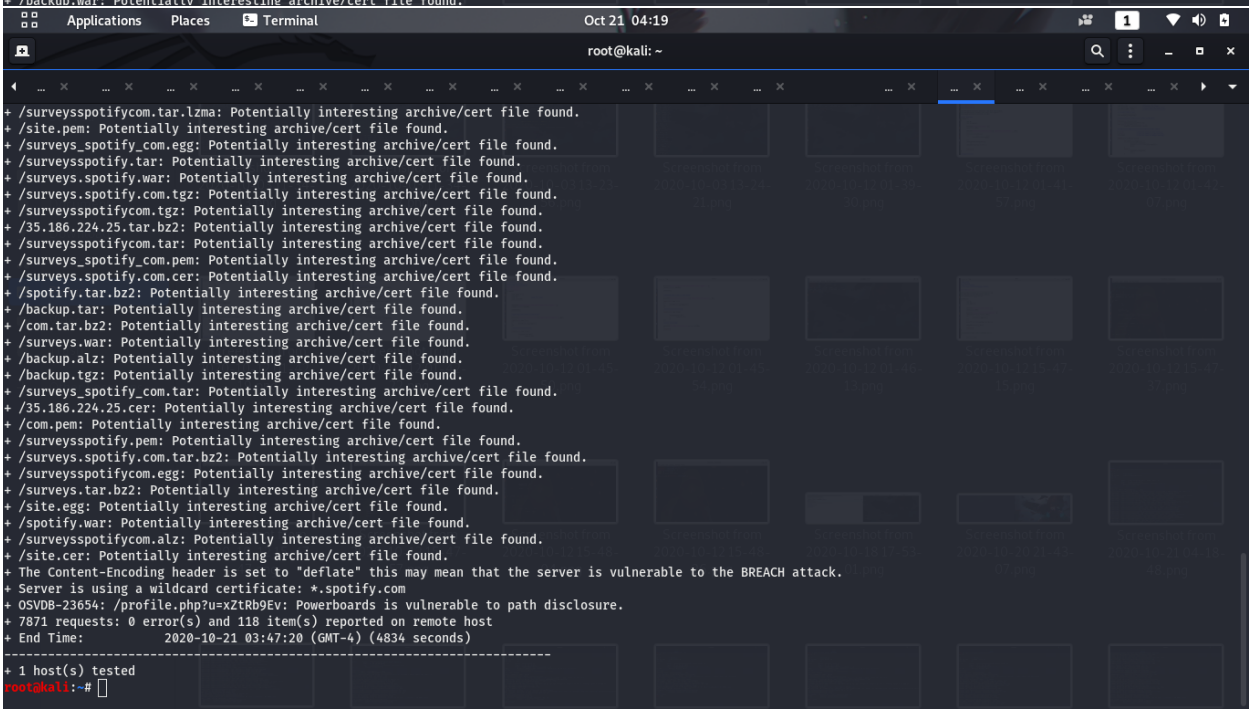
support.spotify.com



1. **Clickjacking**
2. **X-XSS-Protection header is not defined**
3. **Server is using Wildcard Certification**

surveys.spotify.com



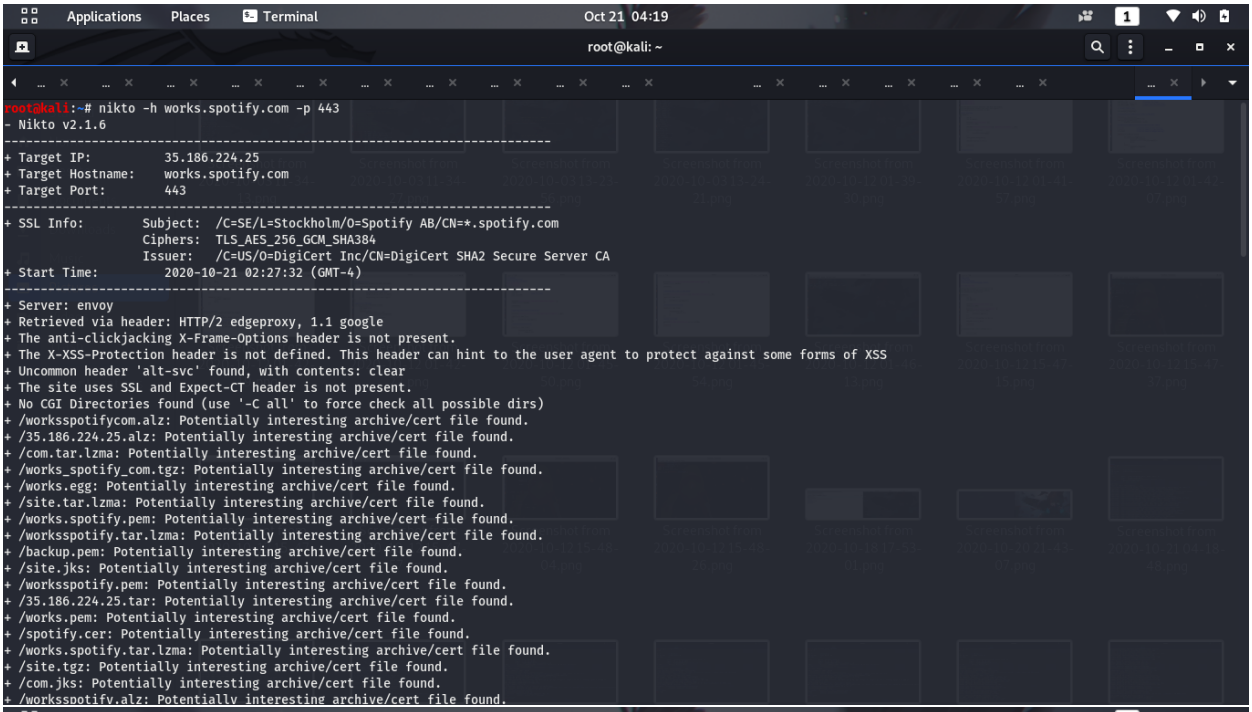


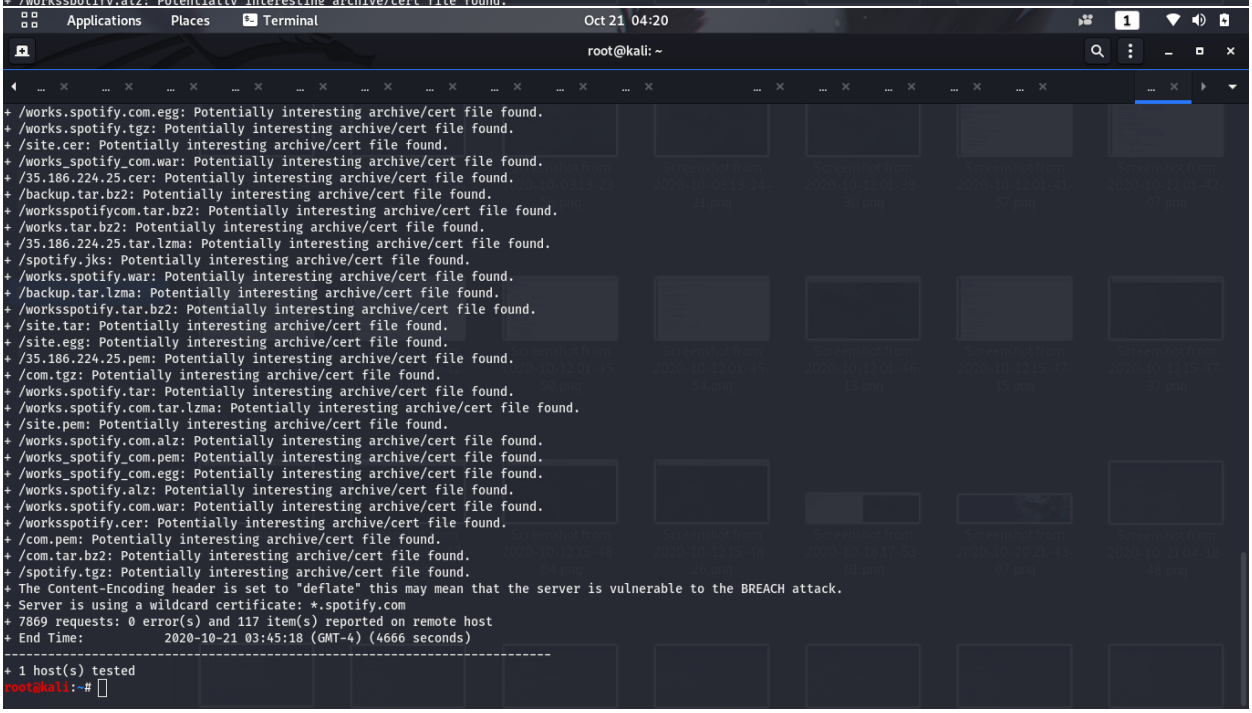
1. **Clickjacking**
2. **X-XSS-Protection header is not defined**
3. **Server is using Wildcard Certification**

uplink.spotify.com

1. **Clickjacking**
2. **X-XSS-Protection header is not defined**
3. **Server is using Wildcard Certification**

works.spotify.com





1. **Clickjacking**
2. **X-XSS-Protection header is not defined**
3. **Server is using Wildcard Certification**

Vulnerability Explanation

1. **Clickjacking** | LOW |CWE-693 | CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:N/I:L/A:N

   Description

   Clickjacking (User Interface Redress Attack, UI Redress Attack, UI Redressing) is a malicious method of tricking a Web user to click on something other than what the user believes they are clicking on, thereby potentially exposing sensitive information or gaining control of their device while clicking on apparently harmless web pages.

   The X-Frame-Options header was not returned by the server, which means this website may be at risk of a clickjacking attack. The HTTP response header for X-Frame-Options can be used to indicate if a browser should be allowed to make a page within a frame or iframe or not. To stop clickjacking attacks, sites may do this by ensuring that their content is not inserted into other sites.

   Remediating

   Set your web server to include an X-Frame-Options header and a Frame-Ancestors Directive CSP header. For more detail about the potential values for this header, check the Web references.

2. **X-XSS-Protection header is not defined** | Best Practice | CWE-16, HIPAA-16, ISO27001-A.14.2.5, WASC-15

   Description

   The response header for HTTP 'X-XSS-Protection' is a feature of modern browsers that allows their XSS auditors to be monitored by websites. The server is not configured to return a 'X-XSS-Protection' header, so any pages on this website may be at risk of an attack by Cross-Site Scripting (XSS).

   Remediation

   Add the X-XSS-Protection header with the value of “1; mode = block”

   - X-XSS-Protection: 1; mode=block

3. **X-Content-Type-Options header is not set** | LOW |CWE-16, ISO27001-A.14.1.2, WASC-15, OWASP 2013-A5, OWASP 2017-A6

Description

The response header of the HTTP 'X-Content-Type-Options' prevents the MIME browser from sniffing a response away from the content-type declared.
The server did not return a proper 'X-Content-Type-Options' header, which means that a Cross-Site Scripting (XSS) attack may be at risk on this website.

Impact

In browsers, MIME style sniffing is a standard function to find a suitable way to make data where the HTTP headers sent by the server are either inconclusive or absent. This allows older versions of Internet Explorer and Chrome to perform MIME sniffing on the response body, possibly allowing the response body to be interpreted and displayed as a type of content other than the intended content.

Once a website allows users to upload content which is then published on the web server, the issue arises. If an attacker is able to perform XSS (Cross-site Scripting) attacks by modifying the content to be accepted by the web application and made by the browser as HTML, it is possible to insert code into an image file, for example, and to have the victim execute it by viewing the image.

Remediation

1) Make sure you send the content-type header while serving resources to correctly fit the type of resource being served. For e.g. if an HTML page is served, you can send an HTTP header
   - Content-Type: text/html

2) Add a "nosniff" X-Content-Type-Options header to tell the browser to trust what the site has sent is the acceptable content-type and not to try to "sniff" the actual content-type.
   - X-Content-Type-Options: nosniff

4. **BREACH Attack (Possible)** |Medium|OWASP 2013-A9, OWASP 2017-A9, CWE310|CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:N/A:N

Description

BREACH (Browser Reconnaissance & Exfiltration via Adaptive Compression of Hypertext) attack is possible on this website.

Due to elements that make BREACH attack possible, SSL/TLS protected traffic remains vulnerable and can be attacked to uncover information from the website.

Regardless of which version of SSL/TLS you use, attacks are still possible. Attacks do not require TLS-layer compression and they can work against any cipher suite.

Impact

Even if you use an SSL/TLS protected connection, an attacker can still view the victim's encrypted traffic and cause the victim to send HTTP requests to the vulnerable web server (by using invisible frames). Following these steps, an attacker could steal information from the website and do the following:

- Inject partial plaintext they have uncovered into a victim's requests
- Measure the size of encrypted traffic

Remediation

Served from a server that uses HTTP-level compression (ie. gzip)

Reflects user-input in the HTTP response bodies

Contains sensitive information (such as a CSRF token) in HTTP response bodies To mitigate the issue, we recommend the following solutions:

- If possible, disable HTTP level compression
- Separate sensitive information from user input

Protect vulnerable pages with CSRF token. The SameSite Cookie attribute will mitigate this issue, because to exploit this issue an attacker forces the victim to visit a target website using invisible frames. With the SameSite cookie attribute added, cookies that belong to the target won't be sent with a request that does not include top level navigation.

Hide the length of the traffic by adding a random number of bytes to the responses.

Add in a rate limit, so that the page maximum is reached five times per minute.

5. **HTTP Strict Transport Security (HSTS) Policy Not Enabled**
6. **Weak Ciphers Enabled**
   (both of above are explained under first set of scans**)**

7. **Out-of-date Version (jQuery)** |Medium| PCI DSS 3.2 -6.2, OWASP 2013 - A9, OWASP2017 - A9, CWE – 829, CAPEC – 310, HIPAA - 164.308(A)(1)(I), ISO27001 - A.14.1.2

   Description
   web site is using jQuery and detected that it is out of date.

   Impact
   Since this is an old version of the software, it may be vulnerable to attacks.

   Remediation
   Please upgrade your installation of jQuery to the latest stable version.

   Known Vulnerabilities in this Version

   jquery Cross-site Scripting (XSS) Vulnerability (CVE-2015-9251)
   - jQuery before 3.0.0 is vulnerable to Cross-site Scripting (XSS) attacks when a cross-domain Ajax request is performed without the dataType option, causing text/javascript responses to be executed.

   Affected Versions - 1.2.1 to 1.12.4

   jquery Cross-site Scripting (XSS) Vulnerability (CVE-2019-11358)
   - jQuery before 3.4.0, as used in Drupal, Backdrop CMS, and other products, mishandles jQuery.extend(true, {}, ...) because of Object.prototype pollution. If an unsanitized source object contained an enumerable __proto__ property, it could extend the native Object.prototype.

   Affected Versions - 1.2.1 to 3.2.1

8. **Autocomplete is Enabled** |LOW| OWASP 2013 - A5, OWASP 2017 - A6,CWE – 16, WASC – 15, ISO27001 - A.14.1.2

Description

Autocomplete is Enabled in one or more of the form fields which might contain sensitive information like "username", "credit card" or "CVV"

Impact

If the user wishes to save, the browser will store the information entered in these fields. This information may be stolen by an intruder who can access the victim's browser. This is particularly important if the program is widely used on shared devices, such as cyber cafes or airport terminals**.**

Remediation

Add the autocomplete="off "attribute to the form tag or individual" input "fields. Since early 2014, however, due to their integrated password management mechanism, major browsers have not complied with this instruction and give users to store passwords internally.

Find all input instances that store and disable autocomplete private data. Fields containing data such as data form "Credit Card" or "CCV" should not be cached**.** You may allow the application to cache usernames and remember passwords, but this is not recommended in most cases.

After fixing the identified problems, re-scan the application to ensure that all the fixes have been implemented correctly**.**

9. **Cookie Not Marked as Http Only** |LOW|OWASP 2013 - A5, OWASP 2017 - A6, CWE - 16, CAPEC - 107, WASC - 15, ISO27001 - A.14.2.5

Description

HTTPOnly cookies cannot be read by client-side scripts, therefore marking a cookie as HTTPOnly can provide an additional layer of protection against cross-site scripting attacks

Impact

An attacker could easily access cookies and hijack the victim's session during a cross-site scripting attack.

Remediation

Label the HTTPOnly cookie. This is going to be an additional security layer against XSS. This is not a magic bullet, however, and will not safeguard the device from cross-site scripting attacks. An attacker can circumvent HTTPOnly security by using a method such as XSS Tunnel.

10. **Cookie Not Marked as Secure** |LOW|PCI DSS 3.26.5.10, OWASP 2013 - A6, OWASP 2017 - A3, CWE – 614, CAPEC -102, WASC – 15, ISO27001 - A.14.1.2

Description

An attacker who can effectively intercept and decode the traffic, or after a successful man-in-the-middle attack, may theoretically steal the cookie.

Impact

This cookie is transmitted over an HTTP link, so if this cookie is significant (such as a session cookie), it can be intercepted by an attacker and hijacked by the victim. If the intruder is able to execute a man-in-the-middle attack, the victim will be forced to make an HTTP request to steal the cookie.

Remediation

Mark all cookies used within the application as secure

11. **Insecure Frame (External)** |LOW|OWASP 2017A6, CWE -m 16, WASC – 15, ISO27001 - A.14.1.2

Description

An attacker who can effectively intercept and decode the traffic, or after a successful man-in-the-middle attack, may theoretically steal the cookie.

Impact

This cookie is transmitted over an HTTP link, so if this cookie is significant (such as a session cookie), it can be intercepted by an attacker and hijacked by the victim. If the intruder is able to execute a man-in-the-middle attack, the victim will be forced to make an HTTP request to steal the cookie.

Remediation

Mark all cookies used within the application as secure

12. **Missing X-Frame-Options Header** |LOW|OWASP 2013A5, OWASP 2017 - A6, CWE – 693, CAPEC – 103, ISO27001A. - 14.2.5

Description

The missing X-Frame-Options header means that a clickjacking attack may be at risk on this website.
The X-Frame-Options HTTP header field indicates a policy that determines if the transmitted resource should be made within a frame or an iframe by the browser. To avoid clickjacking attacks, servers should announce this policy in the header of their HTTP responses, which ensures that their content is not inserted into other pages or frames.

Impact

Clickjacking is when an attacker uses several transparent or opaque layers when they intend to click on the top-level page to trick a user into clicking on a button or connection on a framed page. The intruder then hijacks clicks intended for their page and routes them to another page, most likely owned by another program, domain, or both. Keystrokes may also be hijacked using a similar technique. A user can be led to believe that they are typing in the password to their email or bank account with a carefully designed mix of stylesheets, iframes, and text boxes, but are actually typing in an invisible frame operated by the attacker.

Remediation

Sending correct X-Frame-Options to HTTP response headers that instruct the browser not to allow other domains to be framed.

- X-Frame-Options: DENY This absolutely denies frame / iframe loading.
- X-Frame-Options: SAMEORIGIN This option is only allowed if the site you want to load has the same origin.
- X-Frame-Options: ALLOW-FROM URL A particular URL is given to load itself into an iframe. Please notice that, however, not all browsers support this.

Employing defensive code in the UI to ensure that the current frame is the most top-level window.

13. Server is using Wildcard Certification

Description

There are additional security issues regarding the defense provided by SSL / TLS certificates when a wildcard certificate is reused across multiple subdomains hosted on multiple servers. The certificate will be compromised by adversaries in the case of a breach by one of the servers. There is a danger to the security and credibility of traffic to each site where the certificate is used. It would be possible for an intruder who gets the certificate to decrypt, read or alter, and re-encrypt traffic. This is likely to result in classified information being leaked and more targeted attacks being carried out.

In the event of a server breach, each server should have a unique certificate that is only valid for the subdomains and sites it is hosting, in order to limit the harm. There is almost no need to use wildcard certificates, with services such as Let's Encrypt providing free certificates and numerous solutions for automatic certificate renewal and deployment. In addition, wildcard certificates are ideal for the mishandling of private keys. If not safely stored or transferred, they can be compromised, requiring the revocation of a certificate.

14. cookie PHPSESSID created without the httponly flag

Description

If the HttpOnly attribute is placed on a cookie, then the value of the cookie cannot be read or placed by JavaScript on the client side. This measure makes it marginally harder to exploit such client-side attacks, such as cross-site scripting, by preventing them from trivially capturing the cookie value via an embedded script.

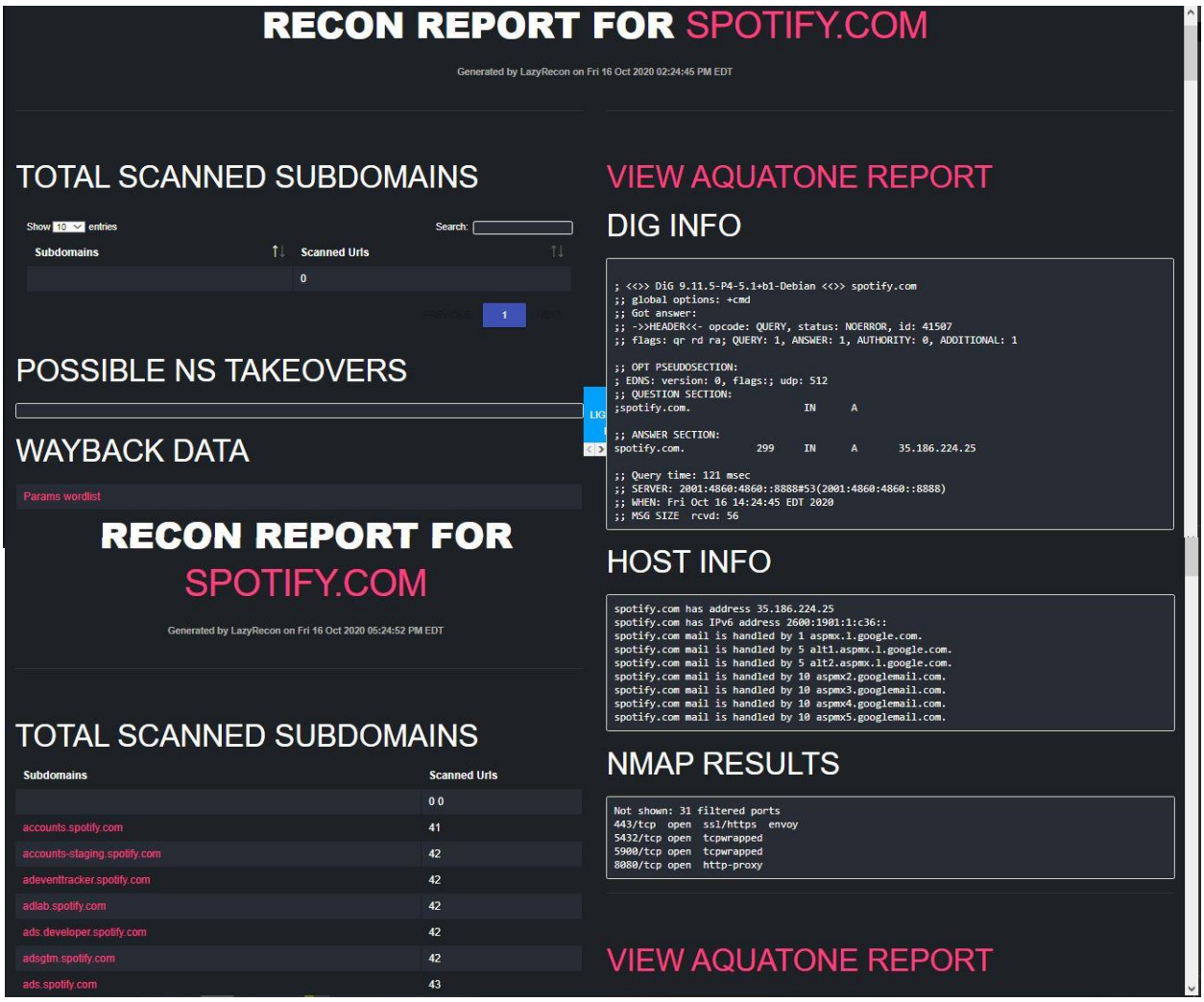
Remediation

Usually, there is no valid reason for not setting the HttpOnly flag on all cookies. You can set the HttpOnly flag by adding this attribute within the applicable Set-cookie directive, unless you explicitly need valid client-side scripts within your application to read or set the value of a cookie.

We should be aware that in certain cases, the constraints placed by the HttpOnly flag can theoretically be circumvented, and that various other severe attacks, apart from basic cookie theft, can be delivered by client-side script injection.

Scan using Lazyrecon

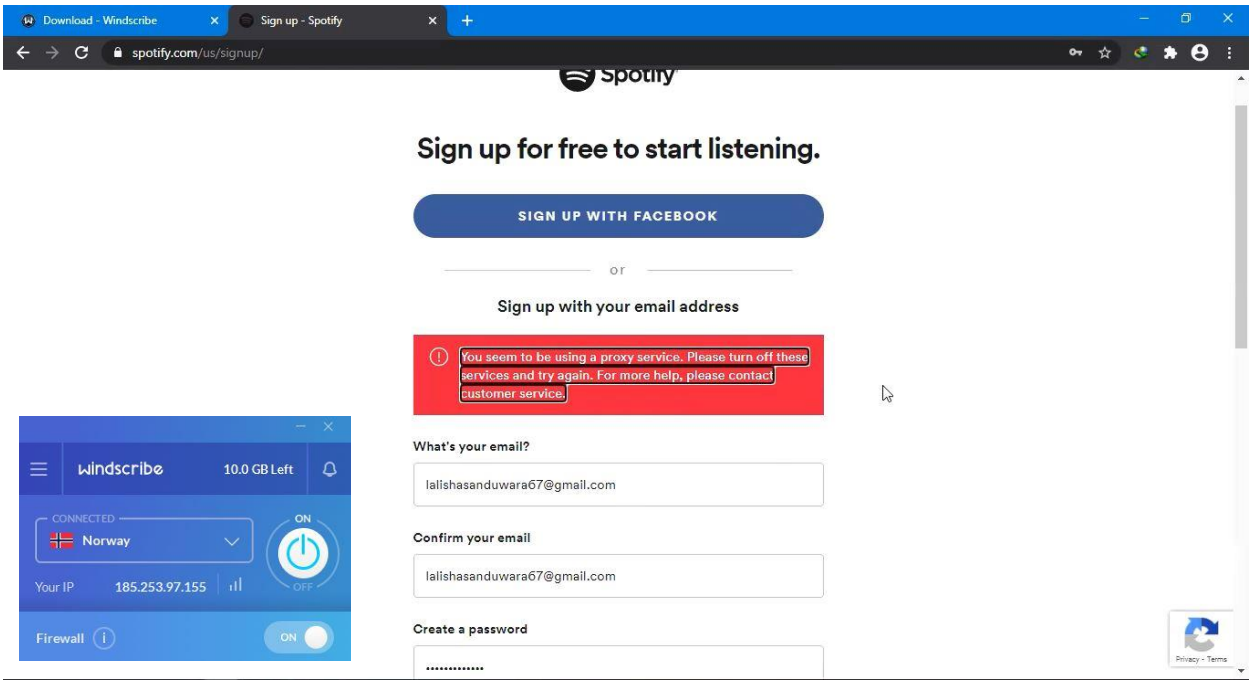
After that I try to scan using Lazyrecon tool. But I wasn't able to understand the report given after the scan.



I wasn't able to configure this out.

Inspect using Burp suite

I also tried to inspect this Spotify.com using Burp suite. Before that I had to create a Spotify account. Since all of you know that Sri Lankans still have no access to create Spotify accounts. So that we have to use a VPN. So, I also installed a Windscribe VPN in my pc and tried to create a Spotify account. But results were as follow.



Because of this situation I was unable to inspect using the Burp suite.

References

Center, S., Definitions, I. and set, C., 2020. *Cookie Without Httponly Flag Set*. [online] Portswigger.net. Available at: <https://portswigger.net/kb/issues/00500600_cookie-without-httponly-flag-set> [Accessed 23 October 2020].

Netsparker.com. 2020. *Missing X-Frame-Options Header | Netsparker*. [online] Available at: <https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-x-frame-options-header/> [Accessed 23 October 2020].

Owasp.org. 2020. *OWASP Foundation | Open Source Foundation For Application Security*. [online] Available at: <https://owasp.org/> [Accessed 23 October 2020].

Academy, W. and scripting, C., 2020. *What Is Cross-Site Scripting (XSS) And How To Prevent It? | Web Security Academy*. [online] Portswigger.net. Available at: <https://portswigger.net/web-security/cross-site-scripting> [Accessed 23 October 2020].

YouTube. 2020. *Web App Penetration Testing Tutorials*. [online] Available at: <https://www.youtube.com/playlist?reload=9&list=PLBf0hzazHTGO3EpGAs718LvLsiMIv9dSC> [Accessed 23 October 2020].

2020. [online] Available at: <https://www.packetlabs.net/wildcard-certificates/#:~:text=Security%20Risks%20of%20Wildcard%20Certificates&text=An%20attacker%20who%20obtains%20the,information%20and%20further%20targeted%20attacks.> [Accessed 23 October 2020].

Final Video Link:

https://drive.google.com/file/d/1uynBledb_zfqRF3sleA5zKDghw5PqMHW/view?usp=sharing