BCSE410L - CYBER SECURITY

PROJECT REPORT

Member 1: Shantanu Anand , 21BCE3137

Member 2: Aryan Gupta , 21BCE0210

Topic: **Cloud Posture Security Management**

Faculty Name: SAIRABANU J

Slot : F2+TF2

Class Id: VL2024250101887

CLOUD POSTURE SECURITY MANAGEMENT

Introduction to Cloud Posture Security Management

Cloud computing has come to form the basis of infrastructure for many organizations, being a ubiquitous feature of the current digital business environment. The flexibility, flexibility and economical points of the cloud services make adoption fast and have made cloud infrastructure strategic in most industries. But this transition to cloud means that the security threats have increased at the same time and as a result, cloud assets and settings require protection from threats. It is where Cloud Posture Security Management (CPSM) comes in to provide its invaluable services. CPSM is the ongoing process of assessing and mitigating the risks in cloud systems so as to mitigate threats, loss of information and unauthorized access to systems.

CPSM is at its core a real-time approach to continuously verifying the correctness of cloud security configurations, identifying misconfigured vulnerabilities, and correcting them as well as adherence to compliance regulations. Originally CPSM is about considering infrastructural and application layers due to high dynamism of cloud environments. Compared to traditional security it is not just centered around reactive security where threats are only mitigated after being identified. This is important because the threats that cloud becomes vulnerable to range from data leakage, insecure interfaces, account hijacking, to even advanced persistent threats (APTs).

Another fascinating feature of mainstream cloud security models is the so-called shared responsibility model. In contrast to on-premise models where security is a full responsibility of the organization, the cloud providers and the customers divide it in between. The cloud provider's responsibility is to protect the underlying infrastructure which could be physical hardware, storage and networking etc., the customer on the other hand intercepts the responsibility of protecting data, applications and configurations that run on those infrastructures. This scattered responsibility necessitates Cloud Posture Security Management; a single misstep in an organization's security management – for instance, a misconfiguration or unpatched weakness – effectively puts the cloud environment in a state of vulnerability.

For this goal in this project, Cloud Posture Security Management will be implemented using two renowned opensource tools namely Skipfish and Wapiti which are compatible with Linux and widely used to detect security holes. All of these tools were chosen based on its potential in threats identification and the ability to work in cloud platform. Skipfish offers quickly both reconnaissance and mapping, which are valuable for a wider picture of security within the application. Wapiti, on the other hand, provides a step by step, more sophisticated, fine-grained vulnerability scanning where the program mimics an attack that will attempt to determine what specific vulnerability exist in the system. Through the use of these instruments, this project will show a holistic method of managing and eradicating risks in cloud systems.

As much as this report will present the general process of CPSM, it will also contain a detailed description of the process of Skipfish and Wapiti installation and use in cloud environments, how effective they are, instructions on how to implement them and results of experiments carried out. By means of this project, we hope to demonstrate the relevance of CPSM and the proper prevention of cloud risks and the lessons to be learned when it comes to adopting a secure and effective cloud environment.

Detailed Description of Cloud Posture Security Management

Formally, cloud posture is known as Cloud Posture Security Management or CPSM which is a critical activity for organizations that have shifted or are planning to shift to cloud computing infrastructure for the management of their IT systems with a view of enhancing their security level. With the advancement in cloud deployment, the security of data, application and workloads also becomes a challenge. CPSM provides systematic approach in overall Security Configuration & practice and emphasizes the principle of continuous monitoring, assessing the risks and protecting cloud resources.

As with most technologies, special issues of security when using cloud systems because of the shared responsibility model, where the cloud provider company is only guarding the infrastructure and the user is responsible for data, applications, and configurations. This model coupled with the dynamic nature of cloud and the ability to make substantial changes in configuration in a real short time means that security cannot be reactive as misconfiguration, lack of compliance or exposed vulnerabilities are significant problems.

Key components of CPSM include:

- **Continuous Monitoring and Assessment:**
  - These constant checks guarantee that security settings are in compliance with acceptable policies always. In contrast to the relevant check which happen occasionally, continuous monitoring recognizes cloud settings, access controls and policies where security mishaps may be most likely to occur and scrutinizes them in real-time or as often as possible.
  - CPSM continuously scans the cloud environment, and thus, in case of misconfiguration or vulnerability in it, such issue will be quickly discovered by the tool, which will minimize the chances of attackers taking advantage of the situation.

- **Vulnerability Management:**
  - Vulnerability management in CPSM deals with risks that can have an impact that can compromise cloud resources to threats. Security threats include: SQL injection ; Cross-site scripting and open ports that prevail in cloud systems in case they are uncontrolled.
  - Such issues are addressed using vulnerability management tools so that the security teams can target on addressing the most important issues first. Weekly or even daily scanning and then fixing the vulnerabilities is the fundamental steps within the vulnerability management lifecycle.

- **Compliance and Governance:**
  - Regulation is a necessary condition for many enterprises that work in industries with requirements set by the GDPR, HIPAA and ISO 27001, where strict requirements for security are prescribed. CPSM serves the important purpose of applying these standards to cloud environments since it checks the handling and accessibility of information and security settings against industry guidelines.
  - CPSM incorporates compliance checks into the automated process, meaning that consistent tests on cloud settings ensure compliance with standards, and thus helps organizations avoid fines.

- **Risk Mitigation and Incident Response:**
    - CPSM has comprehensive involvement in managing and recognizing hazards in Cloud platforms. The appraisal of risks in terms of impact gives the CPSM a measure to direct resources on the essential areas that need attention.
    - Furthermore, CPSM has established guidelines on how to deal with incidents since they equip organizations with the right response to a security breach, control the extent of the damage and contain any potential future losses. Incident response plans make sure that any time some problems are pointed out, the various corporate entities have measures in place in order to contain the situation.

- **Automated Alerts and Reporting**:
    - The integration of automation in CPSM provides real-time notifications and analytical representations of cloud security status. For critical events related to security breaches or defined threats and vulnerabilities, automated alert accumulations promptly inform the security teams.
    - **Reporting Features**: These features enable a comprehensive view of security, including vulnerability identification, compliance assessments, and trend presentation. This helps organizations monitor progress in specific areas, discover new threats and vulnerabilities, and maintain a record of the company's defensive efforts at any given time.

For this project, the CPSM framework was implemented using the open-source tools Skipfish and Wapiti, which facilitate vulnerability scanning and assessment of cloud security configurations. These tools enable a realistic emulation of vulnerability assessments, demonstrating how CPSM can enhance cloud security by identifying and mitigating risks effectively.

Description of the Tools Used in Cloud Posture Security Management

For the Cloud Posture Security Management (CPSM) implementation, this project leverages two powerful open-source tools: Skipfish and Wapiti. These tools were carefully selected for their advanced capabilities in vulnerability detection and security assessment, making them suitable for evaluating and securing cloud-based applications. Below, we dive into the unique functionalities of each tool, along with their roles and applications within the CPSM framework.

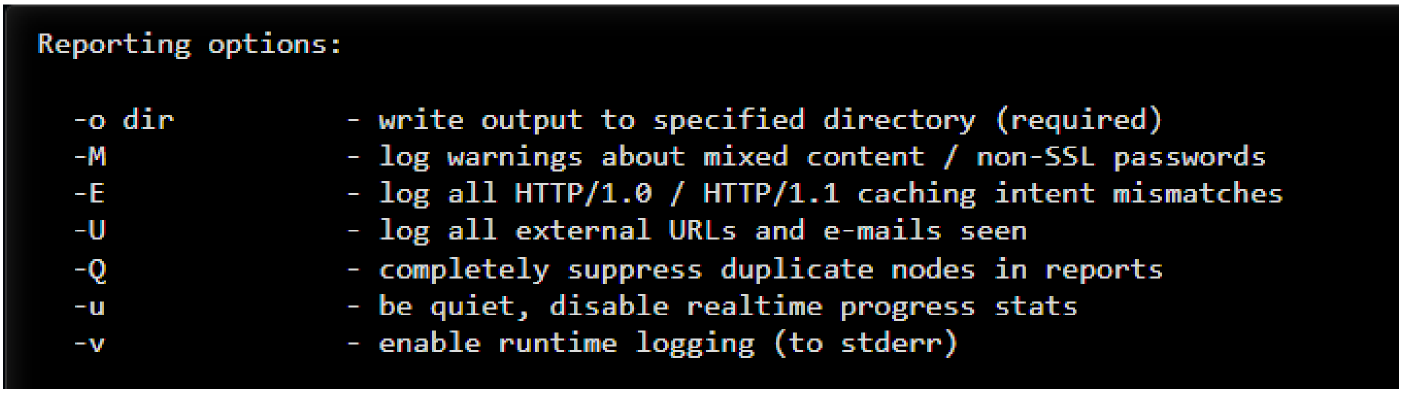
1. Skipfish

Skipfish is an active web application security reconnaissance tool that is developed by Google. Accustomed to fast and high-speed work, Skipfish is created specifically for performing exhaustive scans to create a detailed map of the application structure and reveal a wide range of possible security issues.

Key Features:

- **High-Speed and Optimized Scanning**: The UI of Skipfish is designed for speed since it uses optimized code to enable the scan to happen very fast. It attains fast scan through HTTP pipelining, simultaneous requests, and have a low CPU utilization which makes it among the fast scanners.

- **Interactive Site Mapping**: Skipfish has a number of features with one of the most important being the ability to generate an interactive map of the structure of the target application. It visualizes all reachable pages, endpoints, resources, and assets and demonstrates the application, as well as identifies possible attack vectors.
- **Advanced Vulnerability Detection and Categorization:** In its default setup, Skipfish classifies vulnerabilities by type of attack, such as SQL injections, cross-site scripting (XSS), insecure cookies, troubles with file handling, and directory traversal. This categorization is useful in determining the kinds of risks in the application and helps give more focus on the important issues.
- **Dictionary-Based Fuzzing:** Skipfish comes with a payload dictionary used for checking for injection and other input-based vulnerabilities. Depending on the specificities of the target vulnerabilities, this dictionary can be tuned for finer quality of the test.
- **Detailed and Customizable Reporting**: As for reporting, Skipfish creates HTML with hyperlinked results of each found issue with URLs, the kind of problems, and ways to fix them. It also presents a layout of the whole application structure which will enable the security teams to have a clearer vision of vulnerability distribution.



```
Reporting options:

  -o dir          - write output to specified directory (required)
  -M              - log warnings about mixed content / non-SSL passwords
  -E              - log all HTTP/1.0 / HTTP/1.1 caching intent mismatches
  -U              - log all external URLs and e-mails seen
  -Q              - completely suppress duplicate nodes in reports
  -u              - be quiet, disable realtime progress stats
  -v              - enable runtime logging (to stderr)
```

Role in Project:

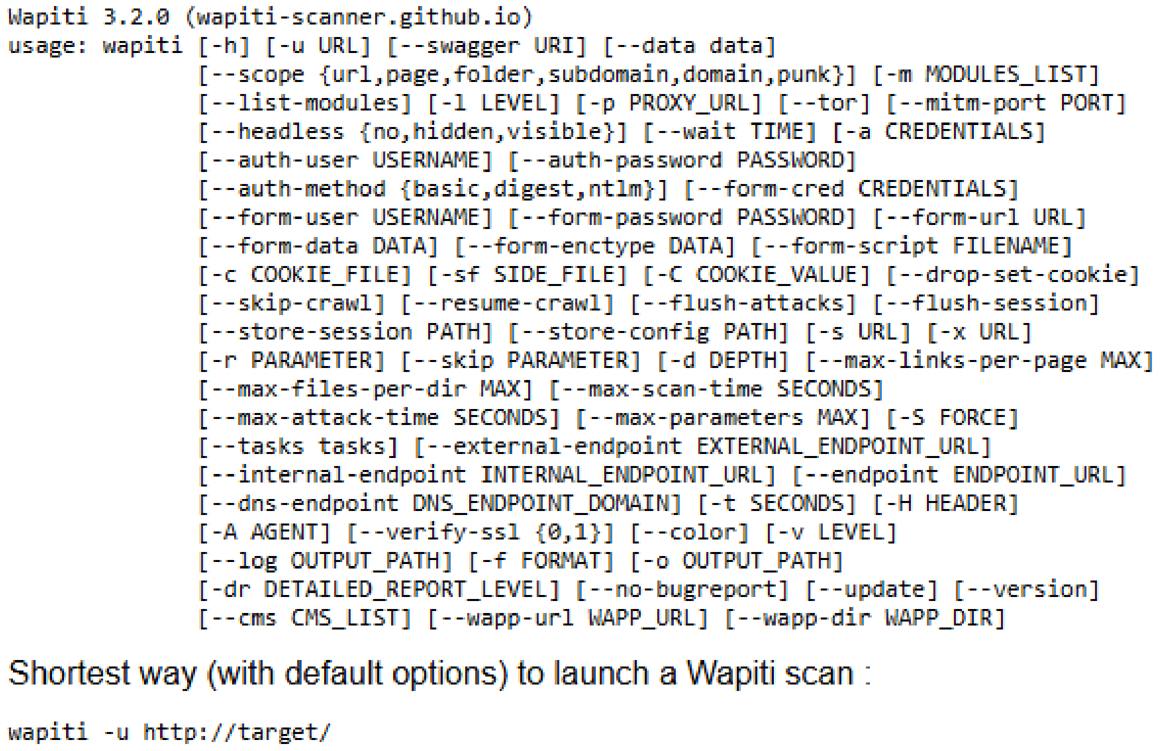
- **Initial Reconnaissance and Mapping:** In this project, Skipfish tool was utilized in conducting a preliminary, high-level reconnaissance seeking to map the flow of the cloud application. With an application map, it offered general information about the structure of the cloud environment exhibits or weak points through which intruders could enter or pathways that should be secured strictly.
- **Baseline Vulnerability Identification:** Using the Skipfish tool for the first scan, prequisites were established for evaluating the general security of the whole application. From this scan it was possible to identify regions with high density of weak points, including login forms, search fields, directories, etc, all of which had to be studied in a more detailed manner.
- **Prioritization for Deeper Scans:** The high-level results from Skipfish were further used to decide which parts warranted a more detailed scan by Wapiti. It made the process more time-saving while ensuring that the severe vulnerabilities are accorded a more elaborate look.

2. Wapiti

Wapiti is another modular web application vulnerability scanner, which has significantly different approach comparing with Skipfish, this tool is based on the simulated attack. Wapiti is to simulate real attacks on applications to specify types of flaws that can be discovered, making it very valuable to focus on the vulnerability kind.

Key Features:

- **Comprehensive Vulnerability Coverage**: Wapiti searches for a broad variety of weaknesses of the kind:
    - **SQL Injection:** Checks for exploitable conditions that would enable injected and malicious SQL statements to be executed on a specific website with resulting disclosure and or alteration of data.
    - **Cross-Site Scripting (XSS):** SCANS for weakness in input fields, with an intention of injecting scripts that could anyway jeopardise users or the application.
    - **File Disclosure:** Recognises scenarios where files, or directories could be vulnerable to being accessed by another user.
    - **Command Injection and CRLF Injection:** Checks input data fields that can cause the execution of unauthorized commands or HTTP response splitting affecting the server's security.
- **Customizable Attack Modules:** Wapiti depends on a modular structure, which makes it possible for the solution to address different attacks as well as techniques. This flexibility allows an additional configuration of attacks or changes of attack types, so the tool can be adapted to a new threat or other needs regarding the security process.
- **Configurable Scanning Modes:** Wapiti allows the user to select between different scan types-the basic scan, which checks for basic vulnerabilities, and the advanced scan, which scans for almost anything an attacker will be able to see.
- **Automated Report Generation:** For each identified threat, Wapiti produces reports with commentary, the hosts impacted, and ways to address the issues. Information or findings can be presented in different formats including HTML and JSON and can be used in other tools or for other security documentation.



```
Wapiti 3.2.0 (wapiti-scanner.github.io)
usage: wapiti [-h] [-u URL] [--swagger URI] [--data data]
              [--scope {url,page,folder,subdomain,domain,punk}] [-m MODULES_LIST]
              [--list-modules] [-l LEVEL] [-p PROXY_URL] [--tor] [--mitm-port PORT]
              [--headless {no,hidden,visible}] [--wait TIME] [-a CREDENTIALS]
              [--auth-user USERNAME] [--auth-password PASSWORD]
              [--auth-method {basic,digest,ntlm}] [--form-cred CREDENTIALS]
              [--form-user USERNAME] [--form-password PASSWORD] [--form-url URL]
              [--form-data DATA] [--form-enctype DATA] [--form-script FILENAME]
              [-c COOKIE_FILE] [-sf SIDE_FILE] [-C COOKIE_VALUE] [--drop-set-cookie]
              [--skip-crawl] [--resume-crawl] [--flush-attacks] [--flush-session]
              [--store-session PATH] [--store-config PATH] [-s URL] [-x URL]
              [-r PARAMETER] [--skip PARAMETER] [-d DEPTH] [--max-links-per-page MAX]
              [--max-files-per-dir MAX] [--max-scan-time SECONDS]
              [--max-attack-time SECONDS] [--max-parameters MAX] [-S FORCE]
              [--tasks tasks] [--external-endpoint EXTERNAL_ENDPOINT_URL]
              [--internal-endpoint INTERNAL_ENDPOINT_URL] [--endpoint ENDPOINT_URL]
              [--dns-endpoint DNS_ENDPOINT_DOMAIN] [-t SECONDS] [-H HEADER]
              [-A AGENT] [--verify-ssl {0,1}] [--color] [-v LEVEL]
              [--log OUTPUT_PATH] [-f FORMAT] [-o OUTPUT_PATH]
              [-dr DETAILED_REPORT_LEVEL] [--no-bugreport] [--update] [--version]
              [--cms CMS_LIST] [--wapp-url WAPP_URL] [--wapp-dir WAPP_DIR]
```

Shortest way (with default options) to launch a Wapiti scan :

```
wapiti -u http://target/
```

Role in Project:

- **Focused Vulnerability Analysis:** Upon engaging Skipfish for the initial sweep of the website, more detailed and specific areas which had been flagged as vulnerable were scanned with Wapiti, specifically the user input fields and points of authentication. It also let a stronger focus on certain types of risk that should be ranked higher, e. g. , SQL injection and XSS.
- **Simulated Attack Emulation:** Wapiti was used to attacked simulation in order to observe how this cloud application reacted to different manipulations of the input data resulting in discovering the issues in input validation and security measures. This helped determine whether the identified vulnerabilities by Skipfish could be exploited.
- **Detailed Reporting for Remediation:** On Wapiti' s side, the reports contain recommendations for rectification actions . For each of the vulnerabilities found, their risk assessment including proposed solutions was provided which made it easier to know how to deal with each of the shortcomings. This level of detail allowed for targets to be set for remediation of what was most problematic, to begin with.

Complementary Use of Skipfish and Wapiti

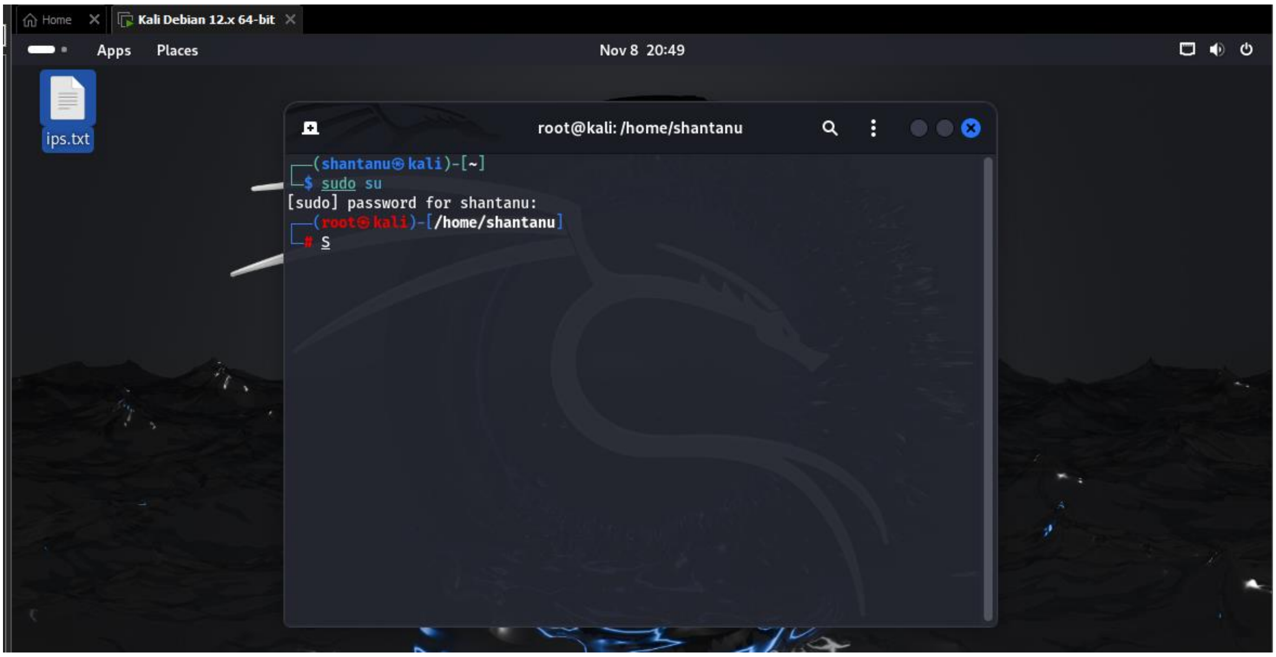
In this project, Skipfish and Wapiti worked in tandem to deliver a comprehensive assessment of the cloud application's security posture. Skipfish provided a fast, initial reconnaissance that mapped out the application's structure and highlighted areas of concern, while Wapiti performed a more detailed, targeted analysis of critical vulnerabilities. By combining the breadth of Skipfish's scans with the depth of Wapiti's emulated attacks, this CPSM approach effectively identified and analyzed a wide array of security risks, enhancing the overall security posture of the cloud environment.

STEP BY STEP IMPLEMENTATION OF TOOL

SKIPFISH IMPLEMENTATION BY SHANTANU (21BCE3137)

STEP 1: OPEN YOUR TERMINAL ON KALI LINUZ SYSTEM AND ENTER THIS COMMAND

sudo su



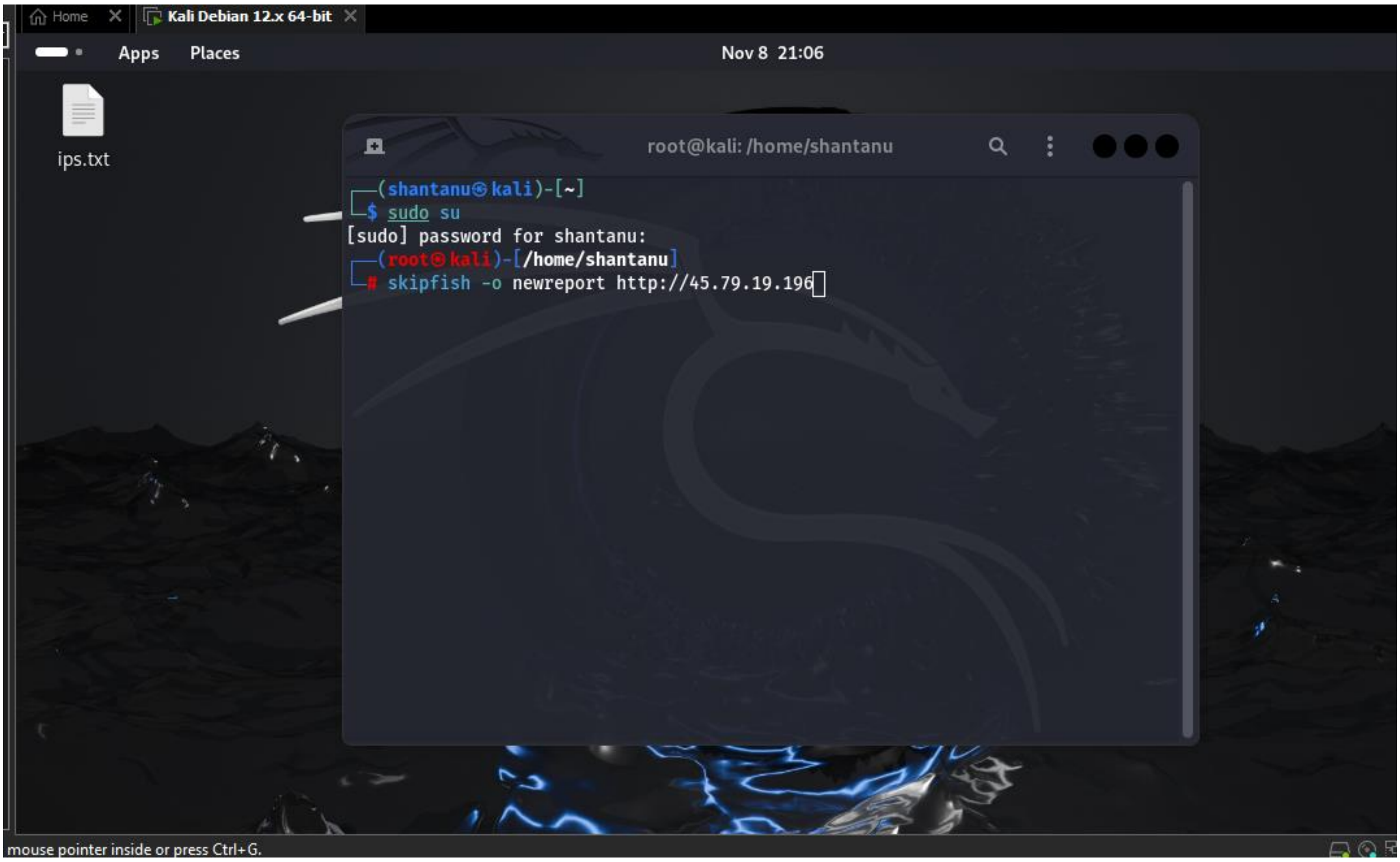
```
┌──(shantanu㉿kali)-[~]
└─$ sudo su
[sudo] password for shantanu:
┌──(root㉿kali)-[/home/shantanu]
└─# S
```

STEP 2: CONSIDER A SUITABLE IP ADDRESS FOR OUR ANALYSIS

IN OUR CASE WE TAKE THIS IP http://45.79.19.196

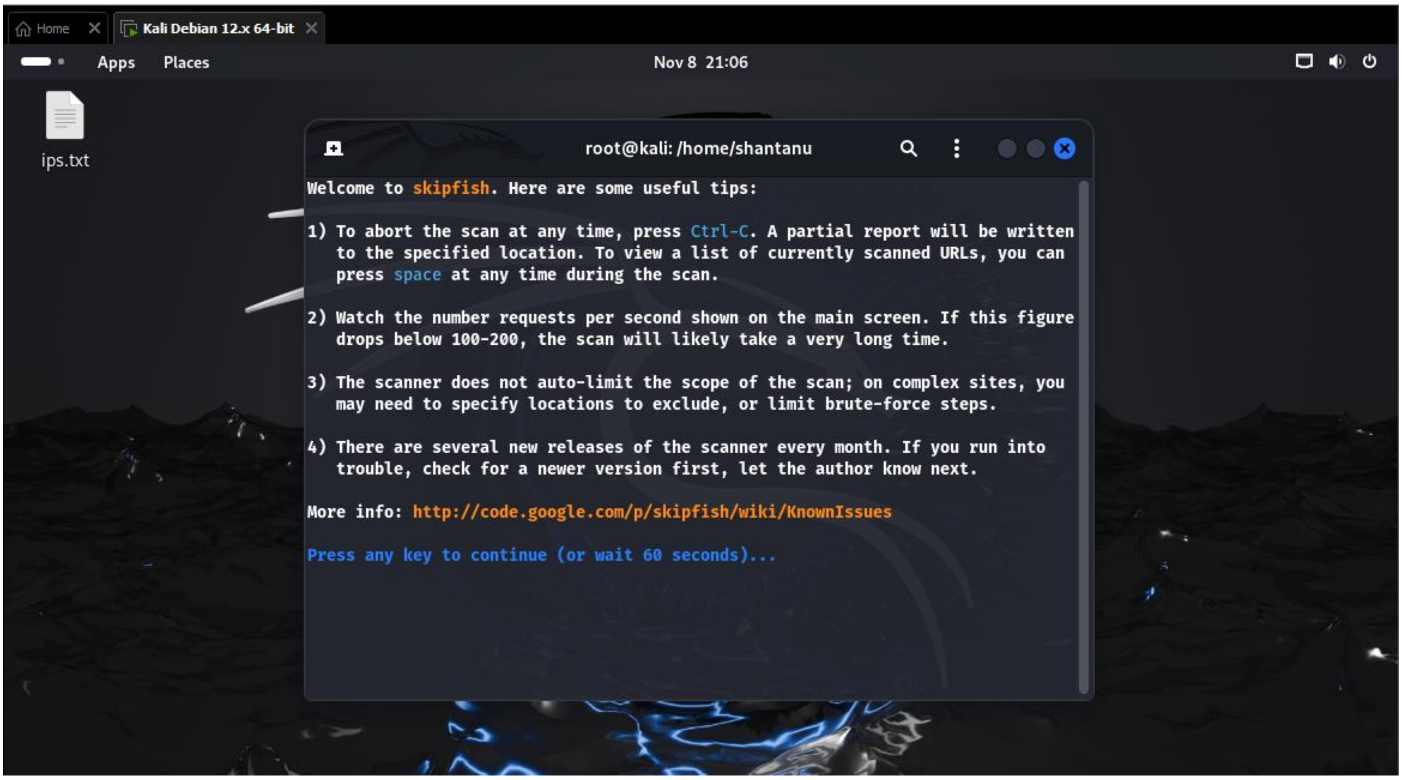
THEN ENTER THIS COMMAND TO RUN SKIPFISH TOOL

skipfish -o newreport http://45.79.19.196


Home
Kali Debian 12.x 64-bit
Apps
Places
Nov 8 21:06
ips.txt
root@kali: /home/shantanu
mouse pointer inside or press Ctrl+G.

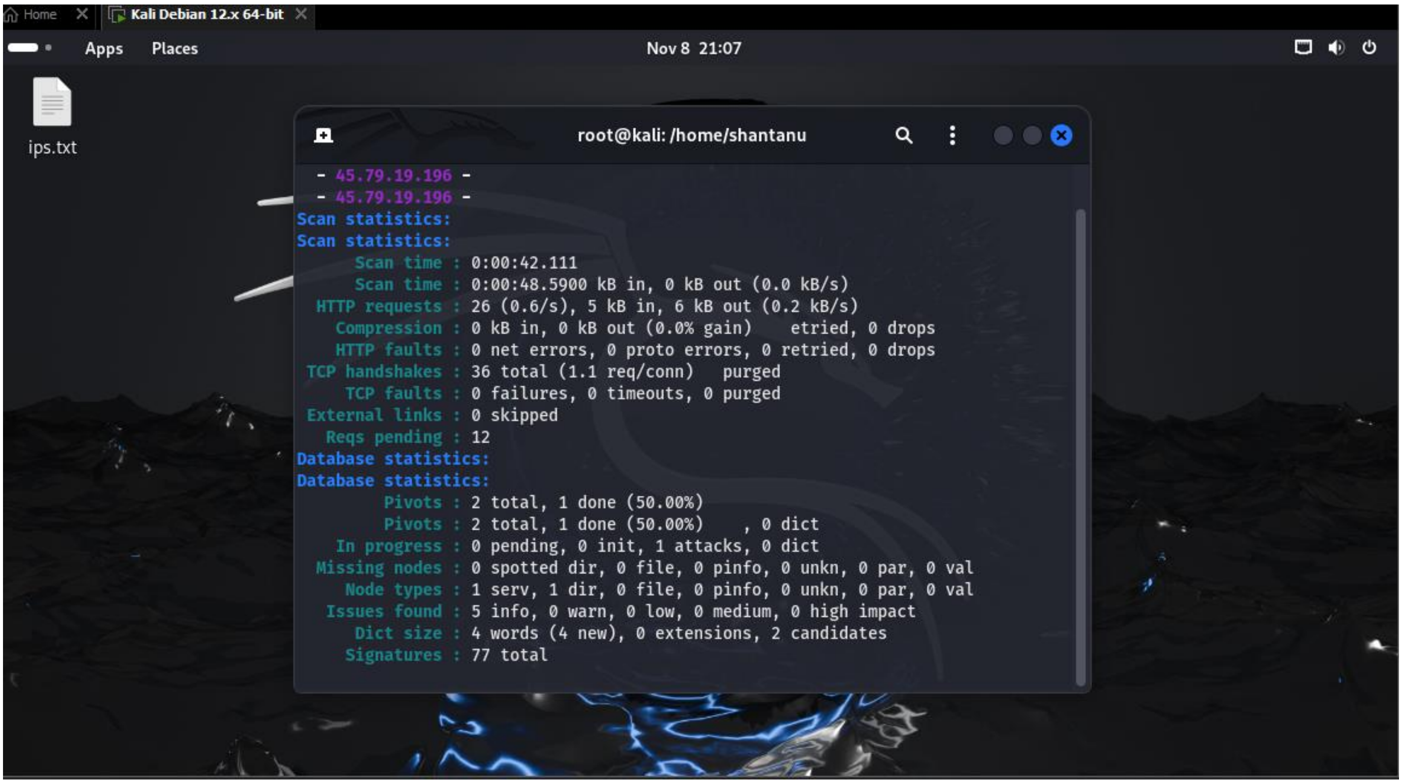
```
┌──(shantanu㉿kali)-[~]
└─$ sudo su
[sudo] password for shantanu:
┌──(root㉿kali)-[/home/shantanu]
└─# skipfish -o newreport http://45.79.19.196
```

STEP 3: PRESS ANY KEY TO INTIATE SCANING



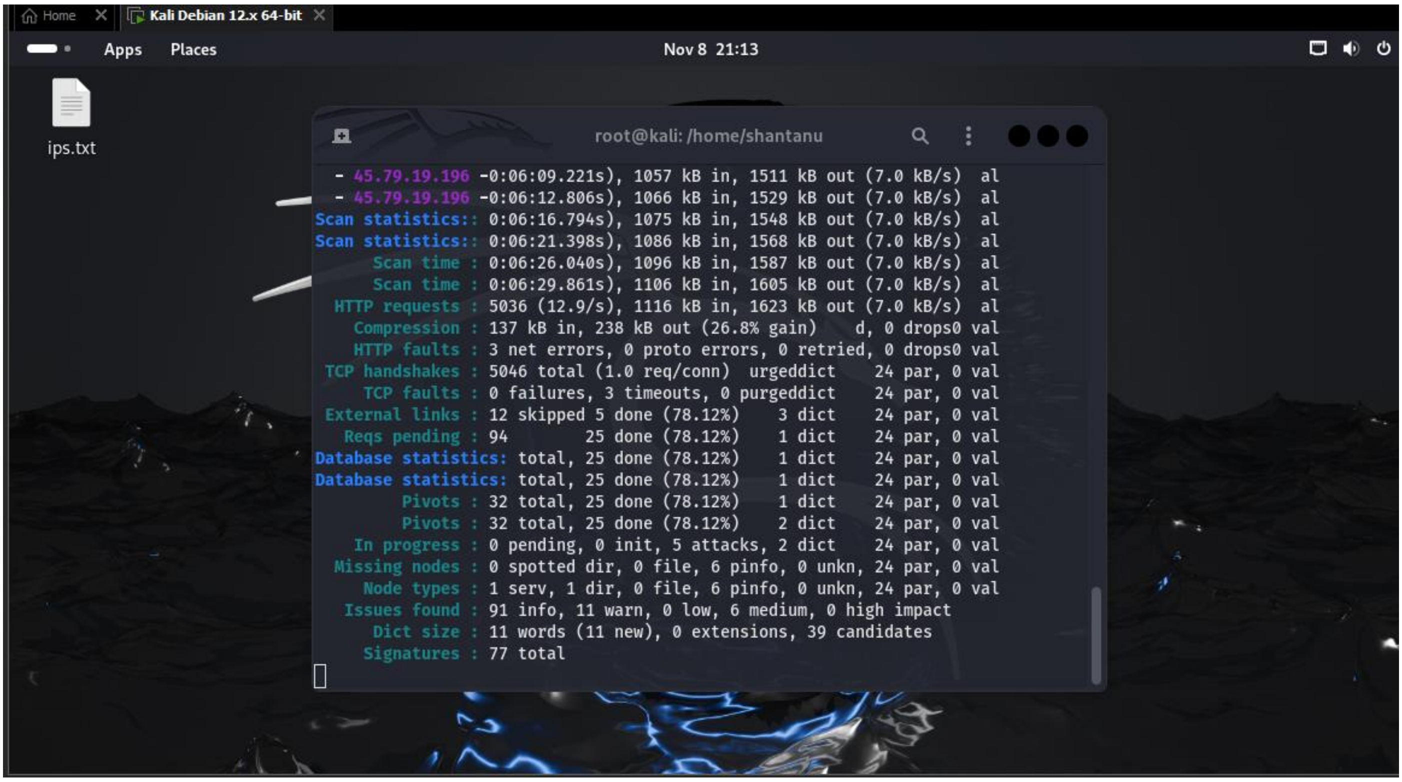
```
Welcome to skipfish. Here are some useful tips:

1) To abort the scan at any time, press Ctrl-C. A partial report will be written
   to the specified location. To view a list of currently scanned URLs, you can
   press space at any time during the scan.

2) Watch the number requests per second shown on the main screen. If this figure
   drops below 100-200, the scan will likely take a very long time.

3) The scanner does not auto-limit the scope of the scan; on complex sites, you
   may need to specify locations to exclude, or limit brute-force steps.

4) There are several new releases of the scanner every month. If you run into
   trouble, check for a newer version first, let the author know next.

More info: http://code.google.com/p/skipfish/wiki/KnownIssues

Press any key to continue (or wait 60 seconds)...
```

STEP 4: WAIT FOR IT TO FINISH SCAN AND GENERATE REPORT



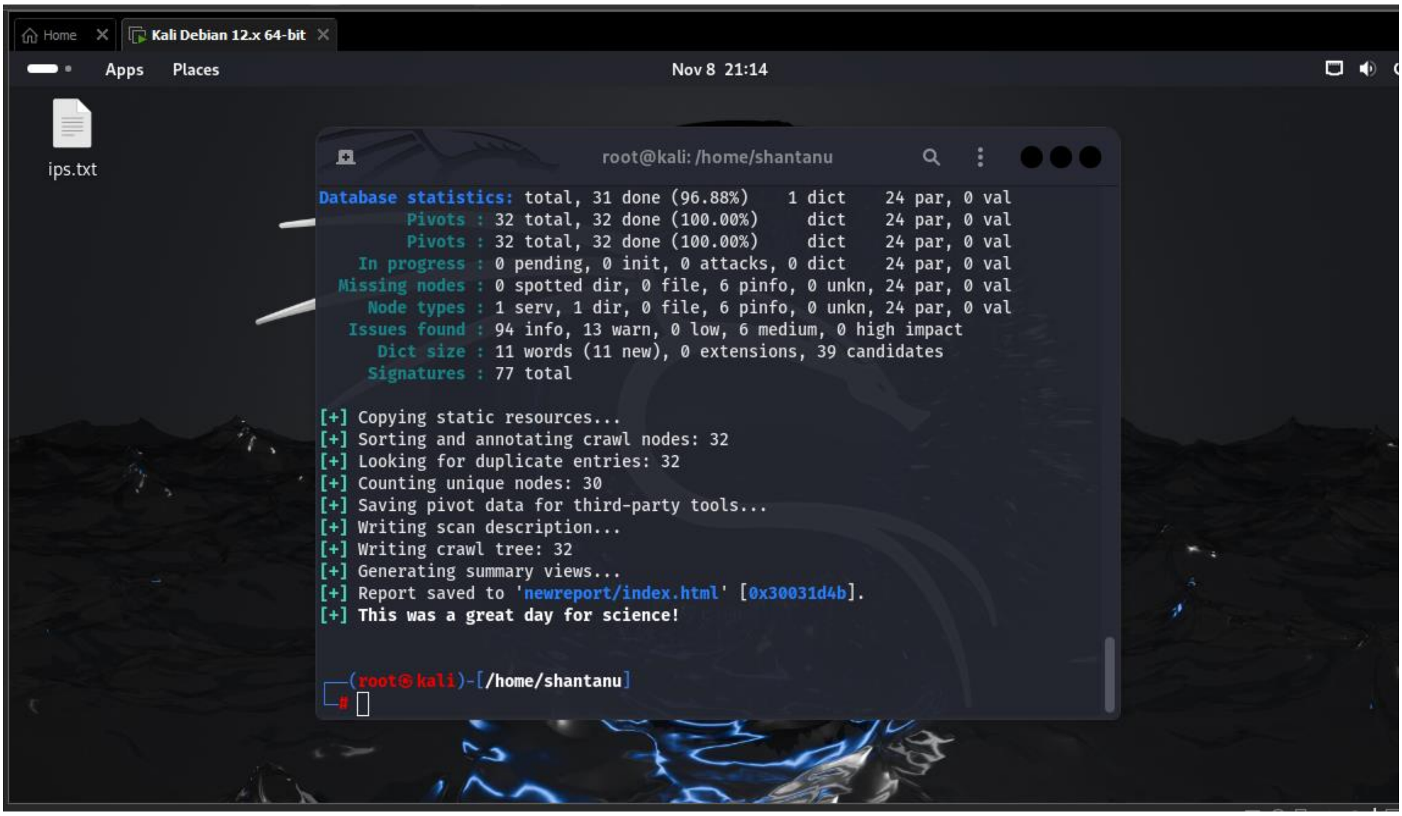
```
  - 45.79.19.196 -
  - 45.79.19.196 -
Scan statistics:
Scan statistics:
      Scan time : 0:00:42.111
      Scan time : 0:00:48.5900 kB in, 0 kB out (0.0 kB/s)
  HTTP requests : 26 (0.6/s), 5 kB in, 6 kB out (0.2 kB/s)
    Compression : 0 kB in, 0 kB out (0.0% gain)    etried, 0 drops
    HTTP faults : 0 net errors, 0 proto errors, 0 retried, 0 drops
 TCP handshakes : 36 total (1.1 req/conn)   purged
     TCP faults : 0 failures, 0 timeouts, 0 purged
 External links : 0 skipped
   Reqs pending : 12
Database statistics:
Database statistics:
         Pivots : 2 total, 1 done (50.00%)
         Pivots : 2 total, 1 done (50.00%)    , 0 dict
    In progress : 0 pending, 0 init, 1 attacks, 0 dict
  Missing nodes : 0 spotted dir, 0 file, 0 pinfo, 0 unkn, 0 par, 0 val
     Node types : 1 serv, 1 dir, 0 file, 0 pinfo, 0 unkn, 0 par, 0 val
   Issues found : 5 info, 0 warn, 0 low, 0 medium, 0 high impact
      Dict size : 4 words (4 new), 0 extensions, 2 candidates
     Signatures : 77 total
```

It will take a while



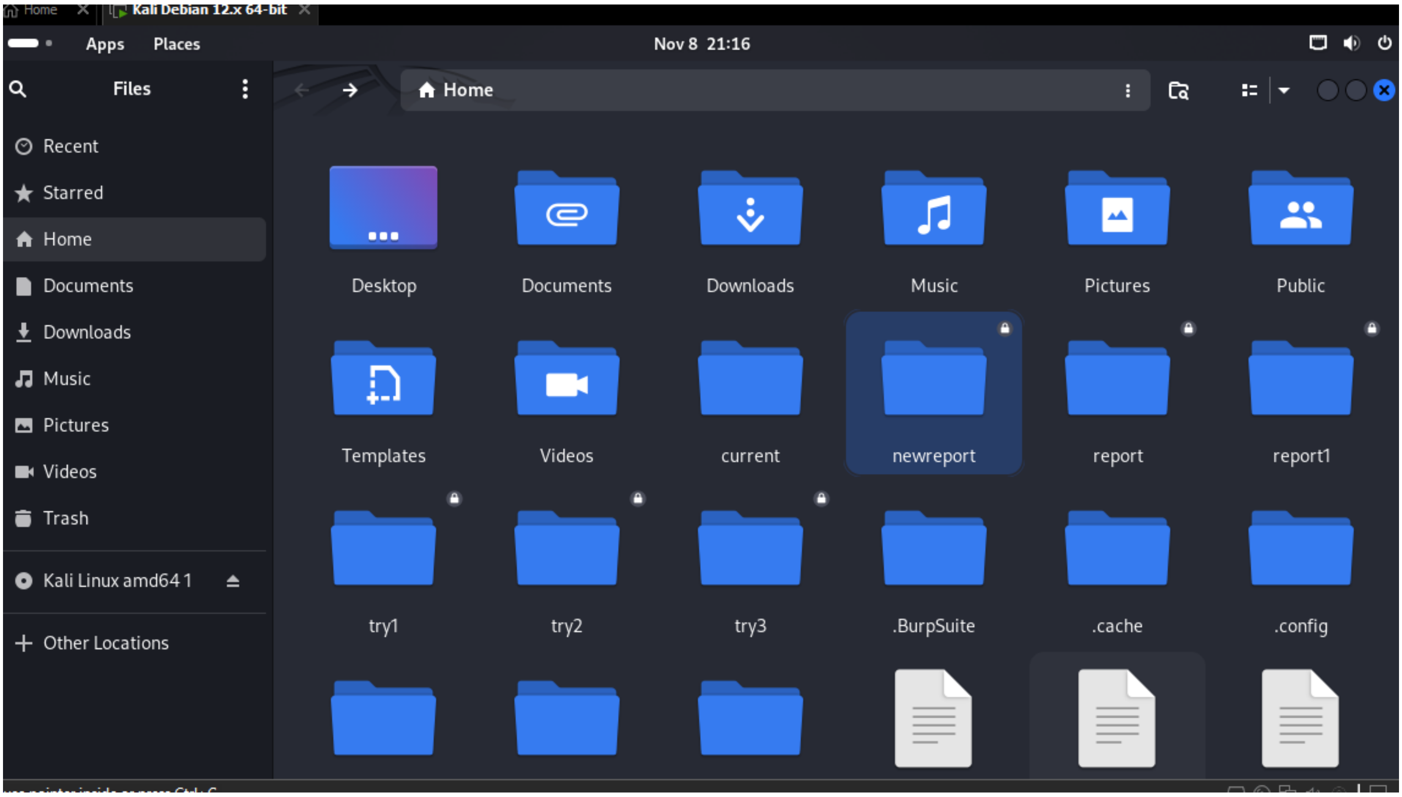
```
  - 45.79.19.196 -0:06:09.221s), 1057 kB in, 1511 kB out (7.0 kB/s)  al
  - 45.79.19.196 -0:06:12.806s), 1066 kB in, 1529 kB out (7.0 kB/s)  al
Scan statistics:: 0:06:16.794s), 1075 kB in, 1548 kB out (7.0 kB/s)  al
Scan statistics:: 0:06:21.398s), 1086 kB in, 1568 kB out (7.0 kB/s)  al
      Scan time : 0:06:26.040s), 1096 kB in, 1587 kB out (7.0 kB/s)  al
      Scan time : 0:06:29.861s), 1106 kB in, 1605 kB out (7.0 kB/s)  al
  HTTP requests : 5036 (12.9/s), 1116 kB in, 1623 kB out (7.0 kB/s)  al
    Compression : 137 kB in, 238 kB out (26.8% gain)    d, 0 drops0 val
    HTTP faults : 3 net errors, 0 proto errors, 0 retried, 0 drops0 val
 TCP handshakes : 5046 total (1.0 req/conn)  urgeddict    24 par, 0 val
     TCP faults : 0 failures, 3 timeouts, 0 purgeddict    24 par, 0 val
 External links : 12 skipped 5 done (78.12%)    3 dict    24 par, 0 val
   Reqs pending : 94        25 done (78.12%)    1 dict    24 par, 0 val
Database statistics: total, 25 done (78.12%)    1 dict    24 par, 0 val
Database statistics: total, 25 done (78.12%)    1 dict    24 par, 0 val
         Pivots : 32 total, 25 done (78.12%)    1 dict    24 par, 0 val
         Pivots : 32 total, 25 done (78.12%)    2 dict    24 par, 0 val
    In progress : 0 pending, 0 init, 5 attacks, 2 dict    24 par, 0 val
  Missing nodes : 0 spotted dir, 0 file, 6 pinfo, 0 unkn, 24 par, 0 val
     Node types : 1 serv, 1 dir, 0 file, 6 pinfo, 0 unkn, 24 par, 0 val
   Issues found : 91 info, 11 warn, 0 low, 6 medium, 0 high impact
      Dict size : 11 words (11 new), 0 extensions, 39 candidates
     Signatures : 77 total
```

STEP 5: AFTER THE SCANING IS FINISHED WE CAN SEE THE LOCATION OF GENERATED REPORT

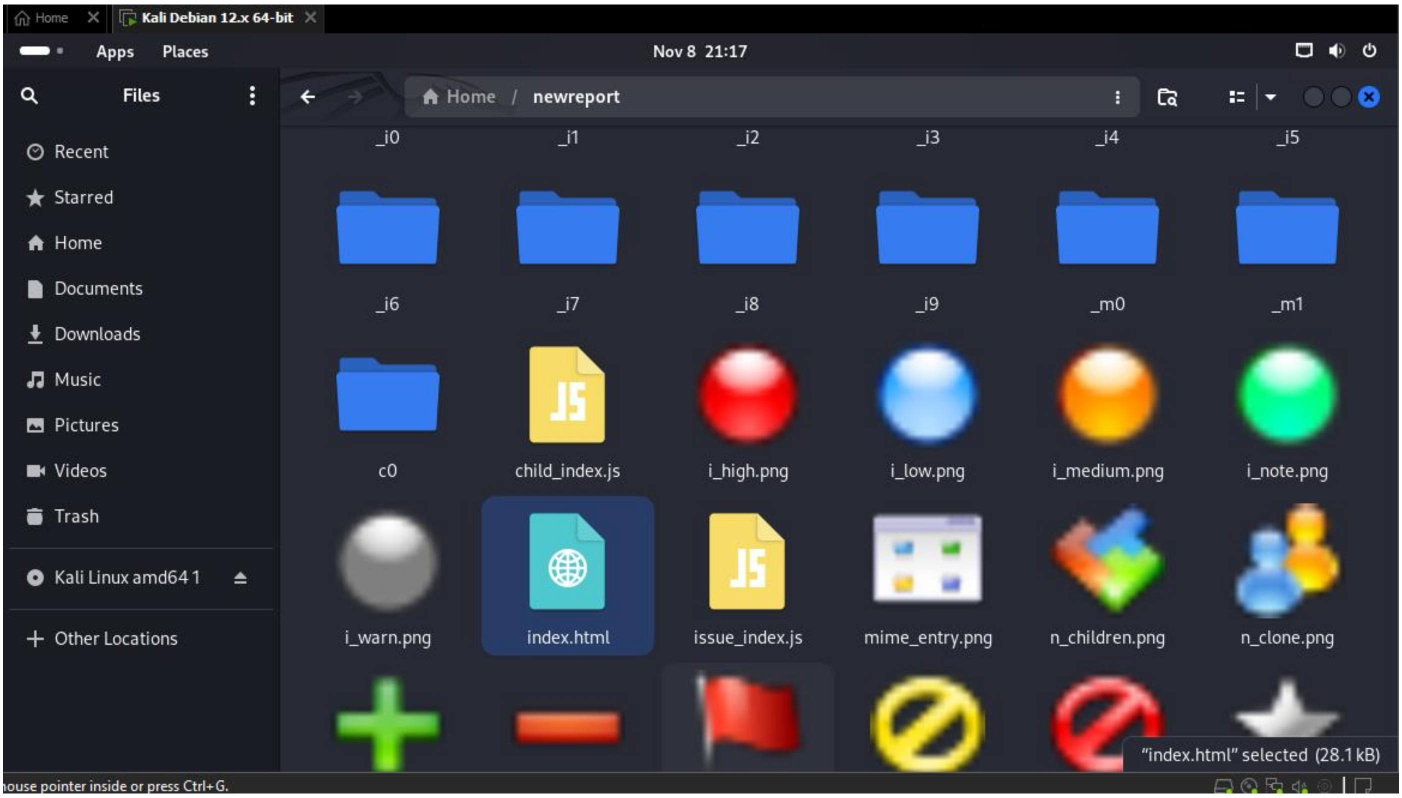


```
Database statistics: total, 31 done (96.88%)    1 dict    24 par, 0 val
         Pivots : 32 total, 32 done (100.00%)     dict    24 par, 0 val
         Pivots : 32 total, 32 done (100.00%)     dict    24 par, 0 val
    In progress : 0 pending, 0 init, 0 attacks, 0 dict    24 par, 0 val
  Missing nodes : 0 spotted dir, 0 file, 6 pinfo, 0 unkn, 24 par, 0 val
     Node types : 1 serv, 1 dir, 0 file, 6 pinfo, 0 unkn, 24 par, 0 val
   Issues found : 94 info, 13 warn, 0 low, 6 medium, 0 high impact
      Dict size : 11 words (11 new), 0 extensions, 39 candidates
     Signatures : 77 total

[+] Copying static resources...
[+] Sorting and annotating crawl nodes: 32
[+] Looking for duplicate entries: 32
[+] Counting unique nodes: 30
[+] Saving pivot data for third-party tools...
[+] Writing scan description...
[+] Writing crawl tree: 32
[+] Generating summary views...
[+] Report saved to 'newreport/index.html' [0x30031d4b].
[+] This was a great day for science!


┌──(root㉿kali)-[/home/shantanu]
└─# 
```

STEP 6: GO TO THE DIRECTORY AND OPEN newreport folder

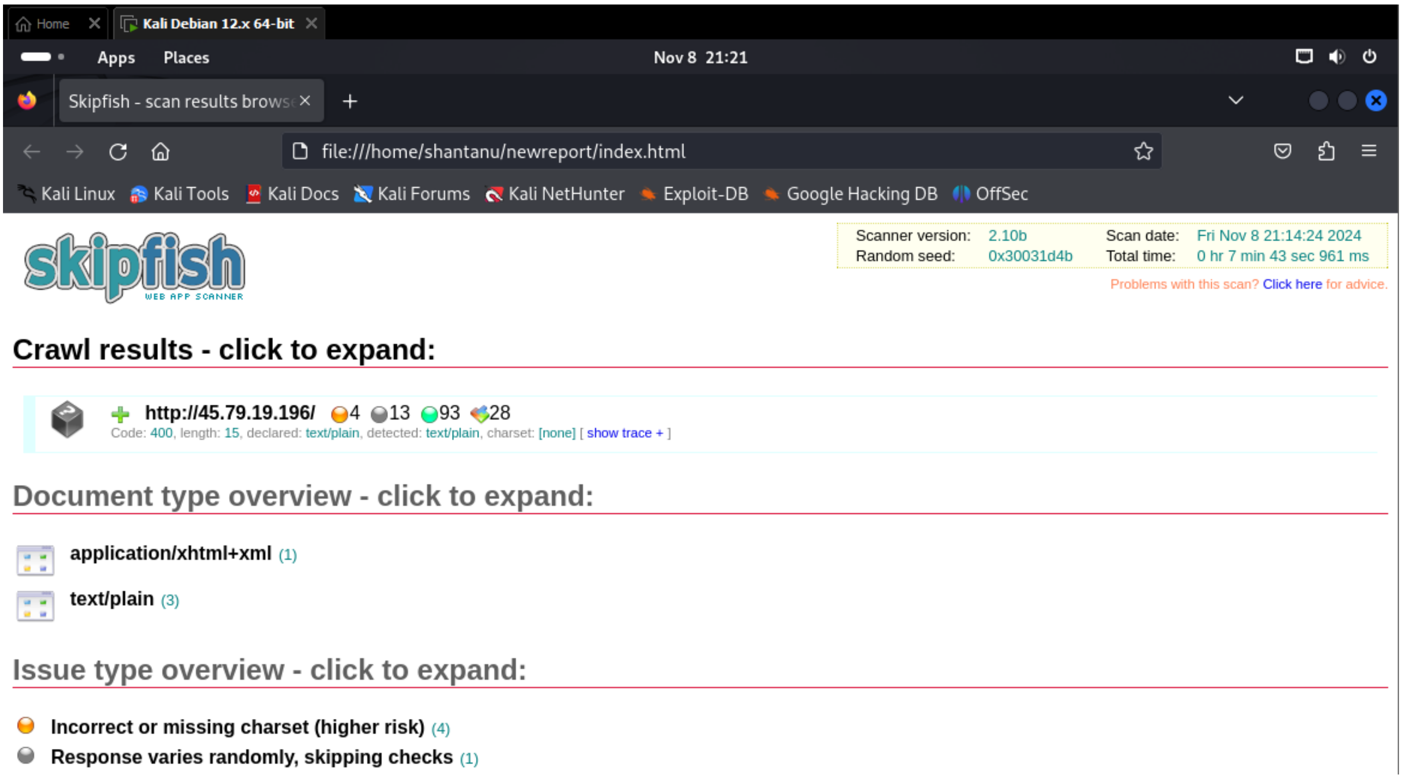


STEP 7: CHECK FOR THE index.html file and open it to see results as it contains the result of analysis



"index.html" selected (28.1 kB)

RESULTS :



Scanner version: 2.10b Scan date: Fri Nov 8 21:14:24 2024
Random seed: 0x30031d4b Total time: 0 hr 7 min 43 sec 961 ms

Problems with this scan? Click here for advice.

Crawl results - click to expand:

http://45.79.19.196/ 4 13 93 28
Code: 400, length: 15, declared: text/plain, detected: text/plain, charset: [none] [show trace +]

Document type overview - click to expand:

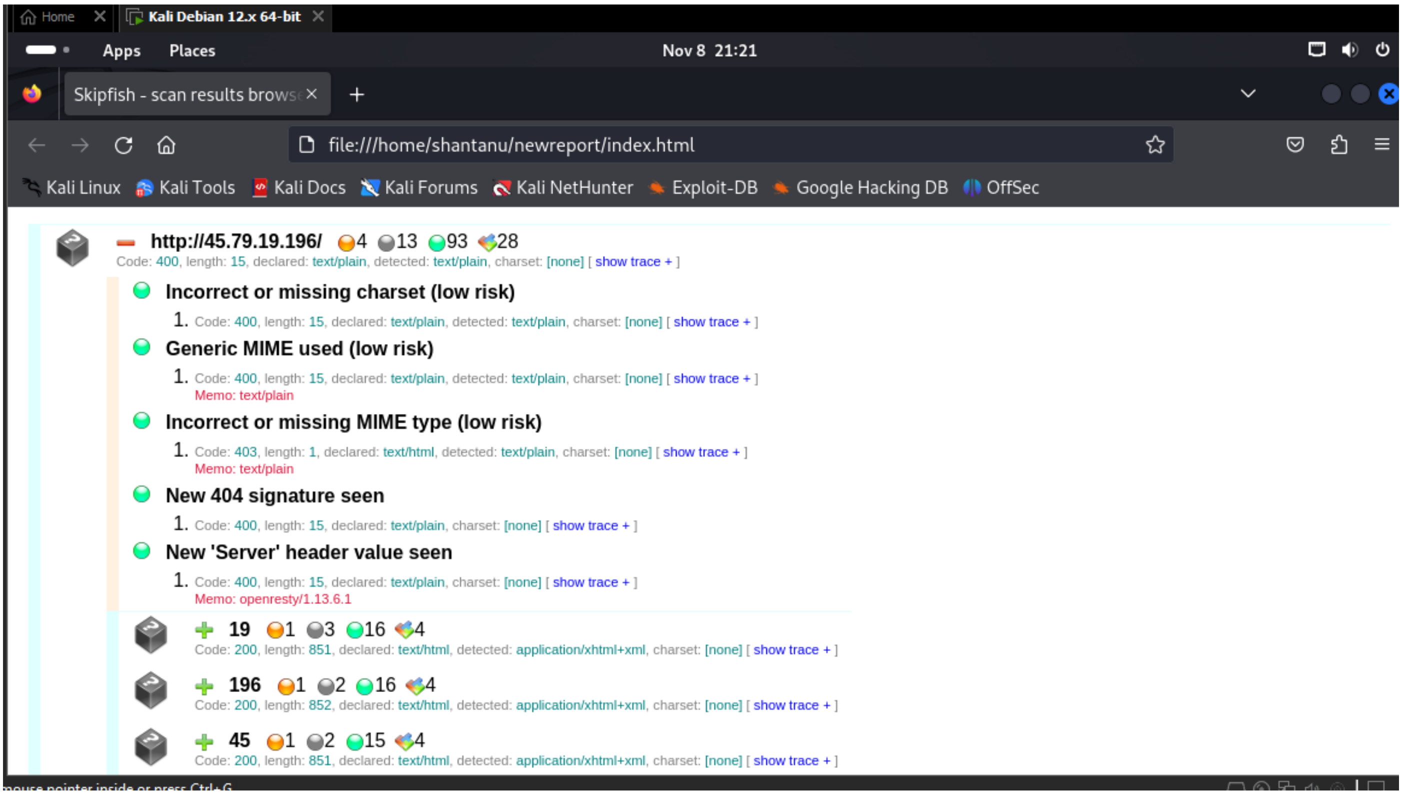
application/xhtml+xml (1)

text/plain (3)

Issue type overview - click to expand:

Incorrect or missing charset (higher risk) (4)

Response varies randomly, skipping checks (1)



http://45.79.19.196/ ●4 ●13 ●93 28
Code: 400, length: 15, declared: text/plain, detected: text/plain, charset: [none] [show trace +]

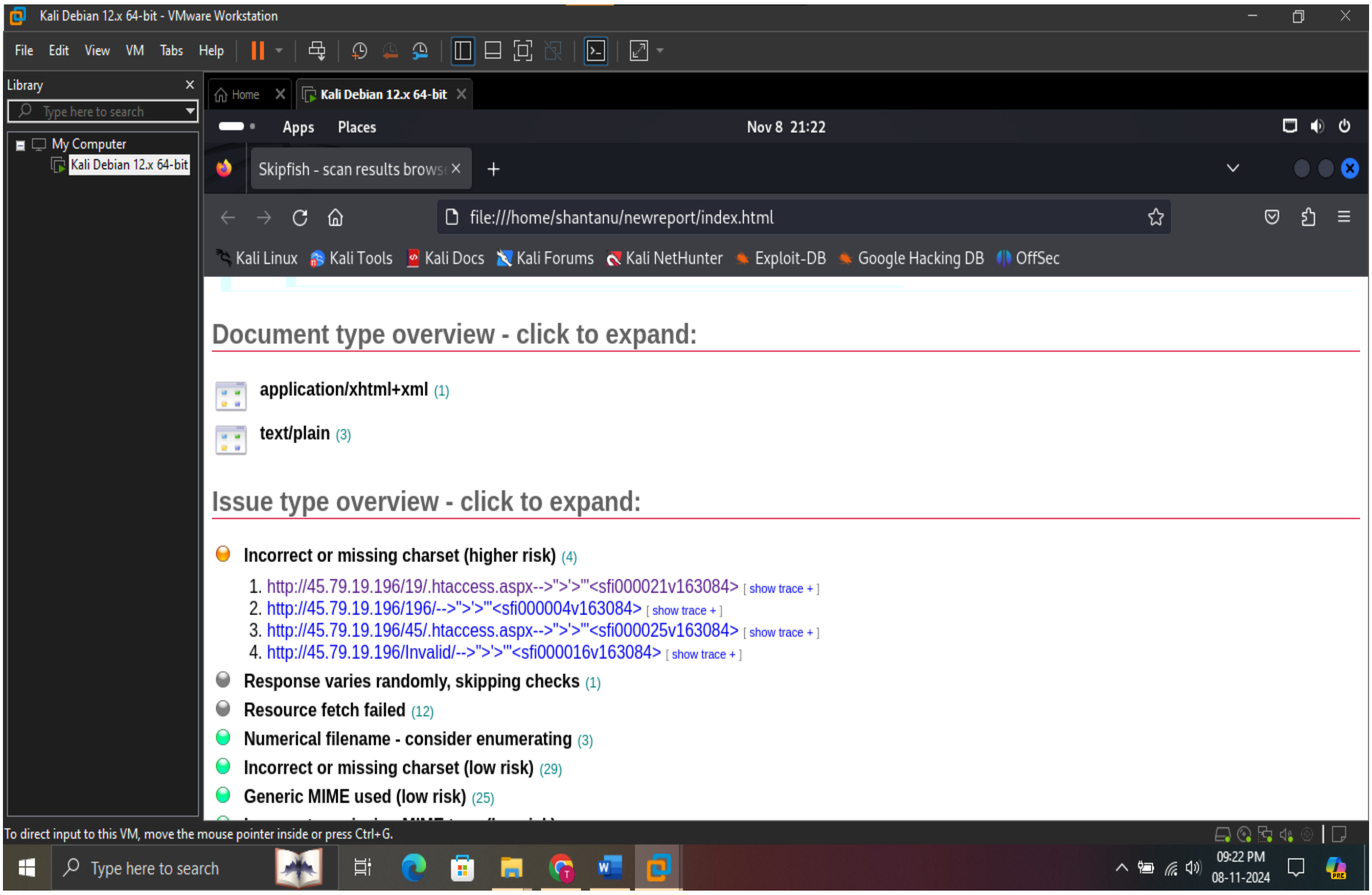
- **Incorrect or missing charset (low risk)**
  1. Code: 400, length: 15, declared: text/plain, detected: text/plain, charset: [none] [show trace +]
- **Generic MIME used (low risk)**
  1. Code: 400, length: 15, declared: text/plain, detected: text/plain, charset: [none] [show trace +]
     Memo: text/plain
- **Incorrect or missing MIME type (low risk)**
  1. Code: 403, length: 1, declared: text/html, detected: text/plain, charset: [none] [show trace +]
     Memo: text/plain
- **New 404 signature seen**
  1. Code: 400, length: 15, declared: text/plain, charset: [none] [show trace +]
- **New 'Server' header value seen**
  1. Code: 400, length: 15, declared: text/plain, charset: [none] [show trace +]
     Memo: openresty/1.13.6.1

19 ●1 ●3 ●16 4
Code: 200, length: 851, declared: text/html, detected: application/xhtml+xml, charset: [none] [show trace +]

196 ●1 ●2 ●16 4
Code: 200, length: 852, declared: text/html, detected: application/xhtml+xml, charset: [none] [show trace +]

45 ●1 ●2 ●15 4
Code: 200, length: 851, declared: text/html, detected: application/xhtml+xml, charset: [none] [show trace +]

mouse pointer inside or press Ctrl+G.



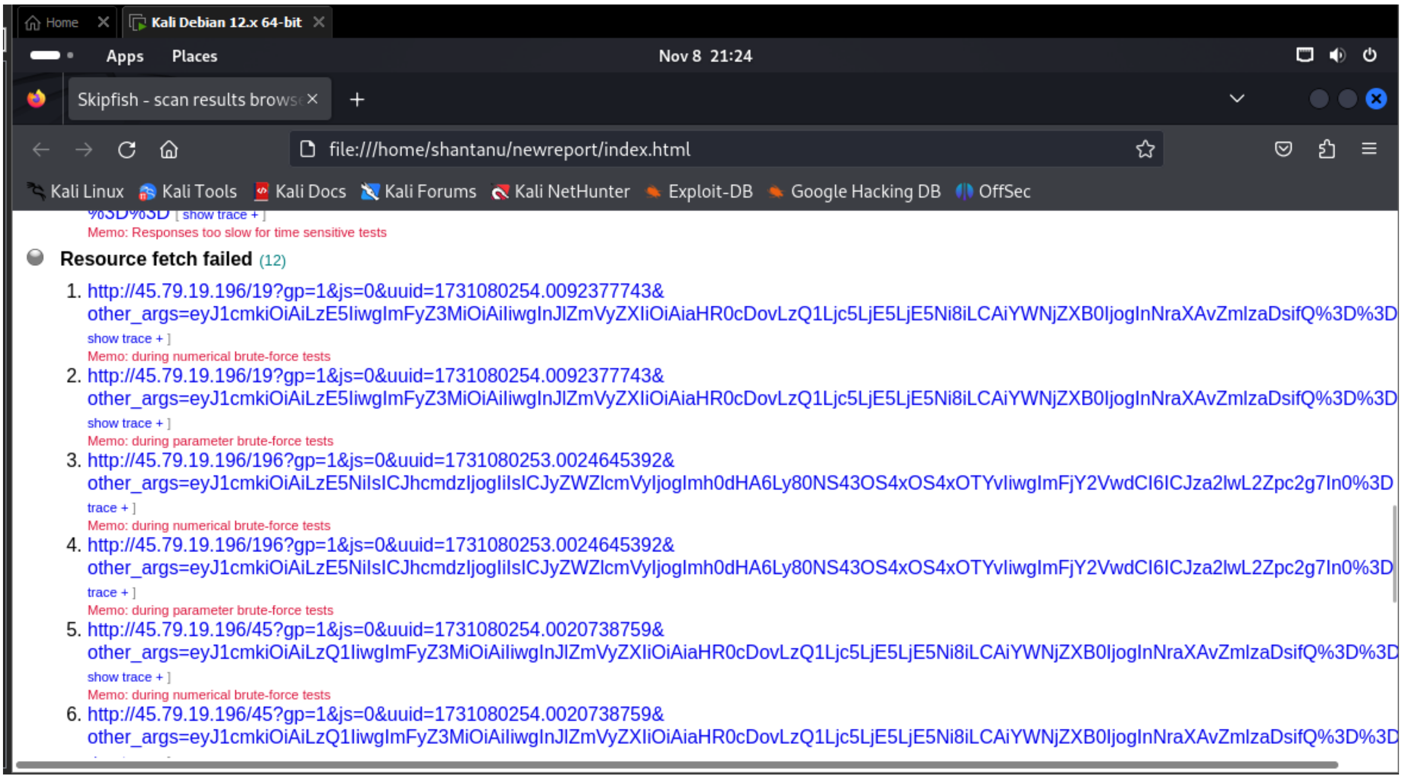
Document type overview - click to expand:

- **application/xhtml+xml** (1)
- **text/plain** (3)

Issue type overview - click to expand:

- **Incorrect or missing charset (higher risk)** (4)
  1. http://45.79.19.196/19/.htaccess.aspx-->">'>'"<sfi000021v163084> [show trace +]
  2. http://45.79.19.196/196/-->">'>'"<sfi000004v163084> [show trace +]
  3. http://45.79.19.196/45/.htaccess.aspx-->">'>'"<sfi000025v163084> [show trace +]
  4. http://45.79.19.196/Invalid/-->">'>'"<sfi000016v163084> [show trace +]
- **Response varies randomly, skipping checks** (1)
- **Resource fetch failed** (12)
- **Numerical filename - consider enumerating** (3)
- **Incorrect or missing charset (low risk)** (29)
- **Generic MIME used (low risk)** (25)

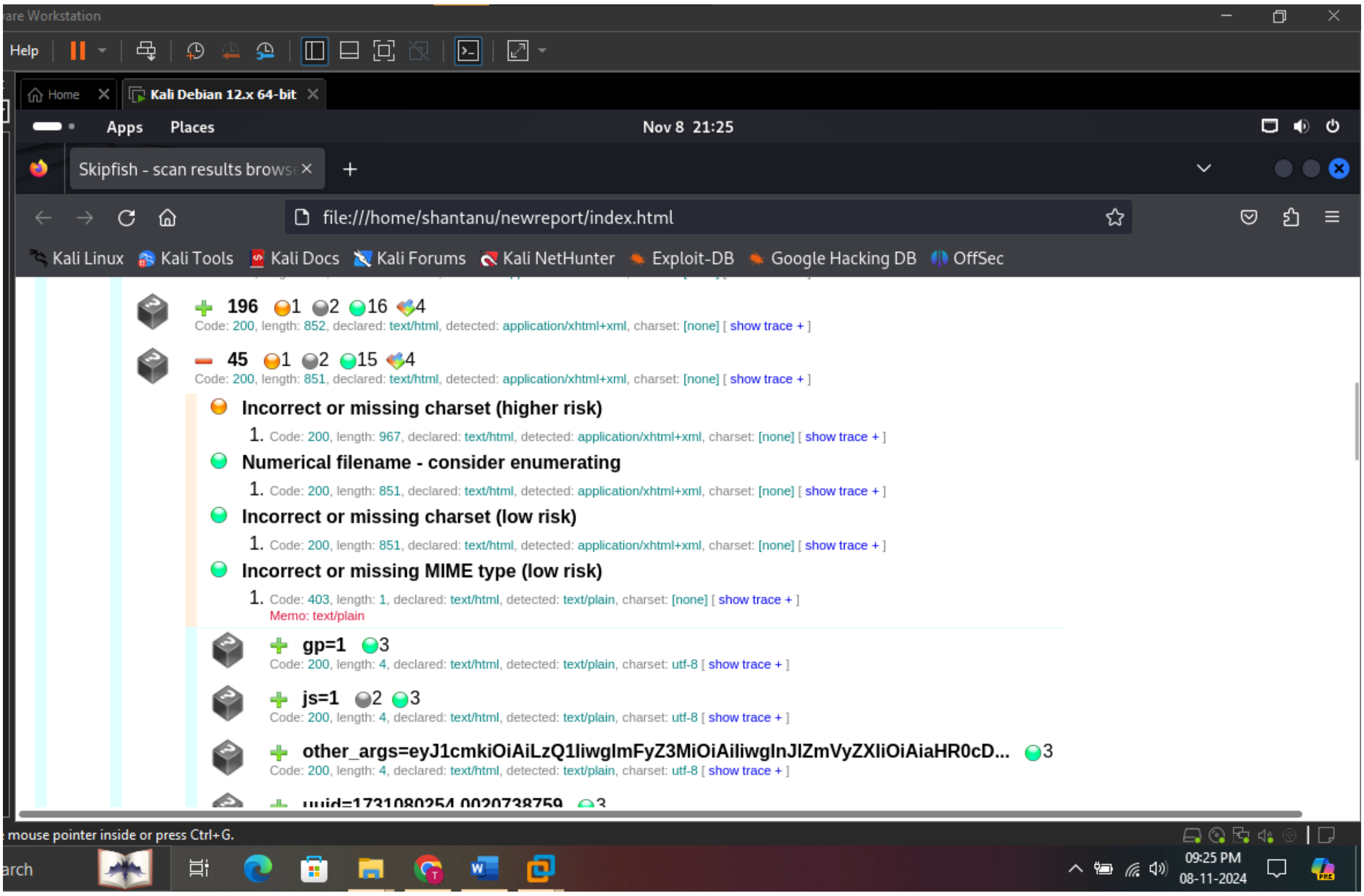
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.



%3D%3D [show trace +]
Memo: Responses too slow for time sensitive tests

Resource fetch failed (12)

1. http://45.79.19.196/19?gp=1&js=0&uuid=1731080254.0092377743&
other_args=eyJ1cmkiOiAiLzE5IiwgImFyZ3MiOiAiIiwgInJlZmVyZXIiOiAiaHR0cDovLzQ1Ljc5LjE5LjE5Ni8iLCAiYWNjZXB0IjogInNraXAvZmlzaDsifQ%3D%3D
show trace +]
Memo: during numerical brute-force tests
2. http://45.79.19.196/19?gp=1&js=0&uuid=1731080254.0092377743&
other_args=eyJ1cmkiOiAiLzE5IiwgImFyZ3MiOiAiIiwgInJlZmVyZXIiOiAiaHR0cDovLzQ1Ljc5LjE5LjE5Ni8iLCAiYWNjZXB0IjogInNraXAvZmlzaDsifQ%3D%3D
show trace +]
Memo: during parameter brute-force tests
3. http://45.79.19.196/196?gp=1&js=0&uuid=1731080253.0024645392&
other_args=eyJ1cmkiOiAiLzE5NiIsICJhcmdzIjogIiIsICJyZWZlcmVyIjogImh0dHA6Ly80NS43OS4xOS4xOTYvIiwgImFjY2VwdCI6ICJza2lwL2Zpc2g7In0%3D
trace +]
Memo: during numerical brute-force tests
4. http://45.79.19.196/196?gp=1&js=0&uuid=1731080253.0024645392&
other_args=eyJ1cmkiOiAiLzE5NiIsICJhcmdzIjogIiIsICJyZWZlcmVyIjogImh0dHA6Ly80NS43OS4xOS4xOTYvIiwgImFjY2VwdCI6ICJza2lwL2Zpc2g7In0%3D
trace +]
Memo: during parameter brute-force tests
5. http://45.79.19.196/45?gp=1&js=0&uuid=1731080254.0020738759&
other_args=eyJ1cmkiOiAiLzQ1IiwgImFyZ3MiOiAiIiwgInJlZmVyZXIiOiAiaHR0cDovLzQ1Ljc5LjE5LjE5Ni8iLCAiYWNjZXB0IjogInNraXAvZmlzaDsifQ%3D%3D
show trace +]
Memo: during numerical brute-force tests
6. http://45.79.19.196/45?gp=1&js=0&uuid=1731080254.0020738759&
other_args=eyJ1cmkiOiAiLzQ1IiwgImFyZ3MiOiAiIiwgInJlZmVyZXIiOiAiaHR0cDovLzQ1Ljc5LjE5LjE5Ni8iLCAiYWNjZXB0IjogInNraXAvZmlzaDsifQ%3D%3D



+ **196** ●1 ●2 ●16 4
Code: 200, length: 852, declared: text/html, detected: application/xhtml+xml, charset: [none] [show trace +]

− **45** ●1 ●2 ●15 4
Code: 200, length: 851, declared: text/html, detected: application/xhtml+xml, charset: [none] [show trace +]

● **Incorrect or missing charset (higher risk)**
1. Code: 200, length: 967, declared: text/html, detected: application/xhtml+xml, charset: [none] [show trace +]

● **Numerical filename - consider enumerating**
1. Code: 200, length: 851, declared: text/html, detected: application/xhtml+xml, charset: [none] [show trace +]

● **Incorrect or missing charset (low risk)**
1. Code: 200, length: 851, declared: text/html, detected: application/xhtml+xml, charset: [none] [show trace +]

● **Incorrect or missing MIME type (low risk)**
1. Code: 403, length: 1, declared: text/html, detected: text/plain, charset: [none] [show trace +]
Memo: text/plain

+ **gp=1** ●3
Code: 200, length: 4, declared: text/html, detected: text/plain, charset: utf-8 [show trace +]

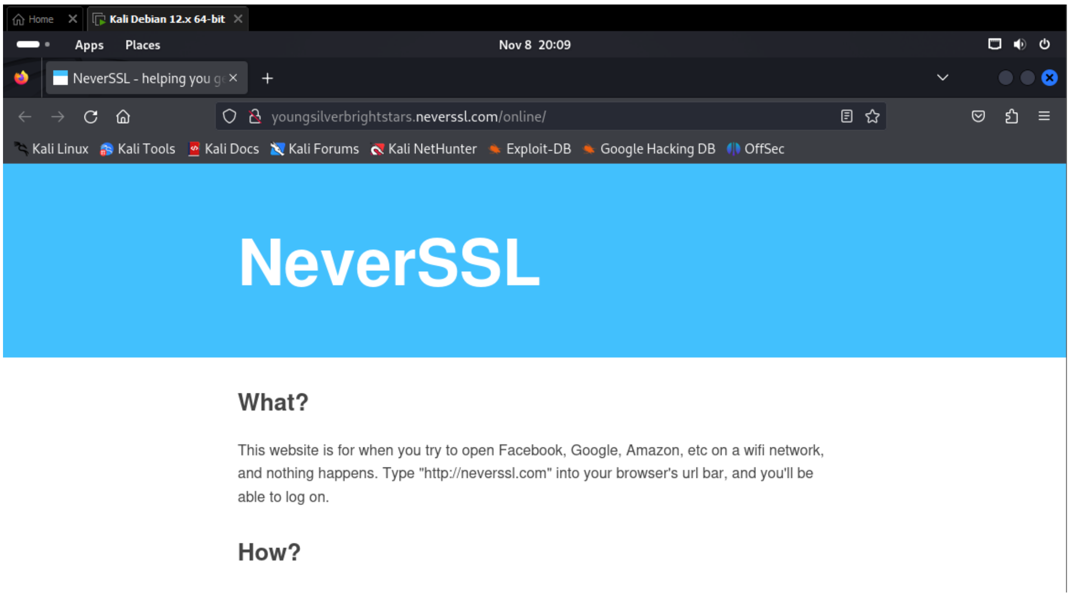
+ **js=1** ●2 ●3
Code: 200, length: 4, declared: text/html, detected: text/plain, charset: utf-8 [show trace +]

+ **other_args=eyJ1cmkiOiAiLzQ1IiwgImFyZ3MiOiAiIiwgInJlZmVyZXIiOiAiaHR0cD...** ●3
Code: 200, length: 4, declared: text/html, detected: text/plain, charset: utf-8 [show trace +]

+ uuid=1731080254.0020738759 ●3

WAPITI IMPLEMENTATION BY ARYAN (21BCE0210)

Step 1: Go to any site and copy its url . In our case its http://youngsilverbrightstars.neverssl.com/online/



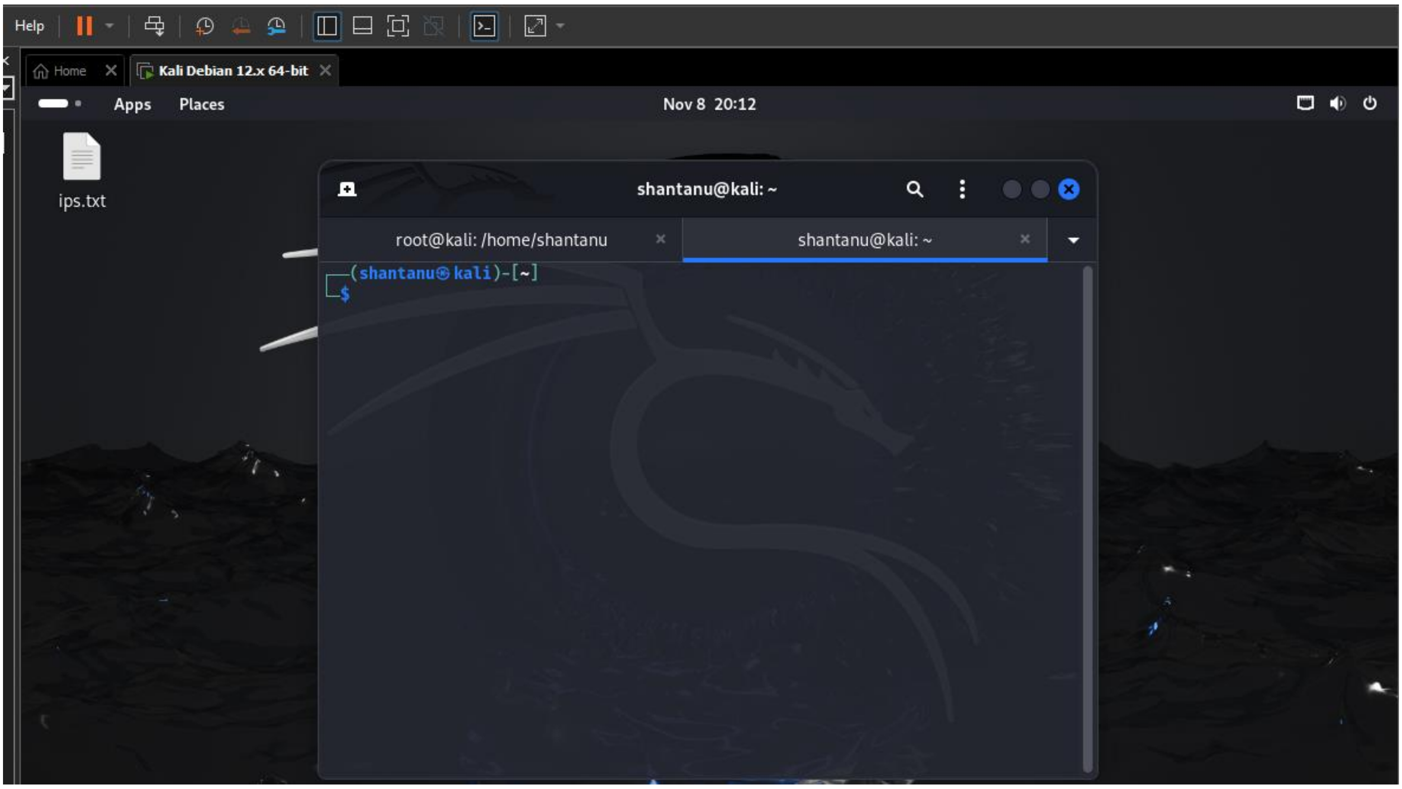
NeverSSL

What?

This website is for when you try to open Facebook, Google, Amazon, etc on a wifi network, and nothing happens. Type "http://neverssl.com" into your browser's url bar, and you'll be able to log on.

How?

Step 2: Open Terminal on your Kali Linux



```
┌──(shantanu㉿kali)-[~]
└─$
```

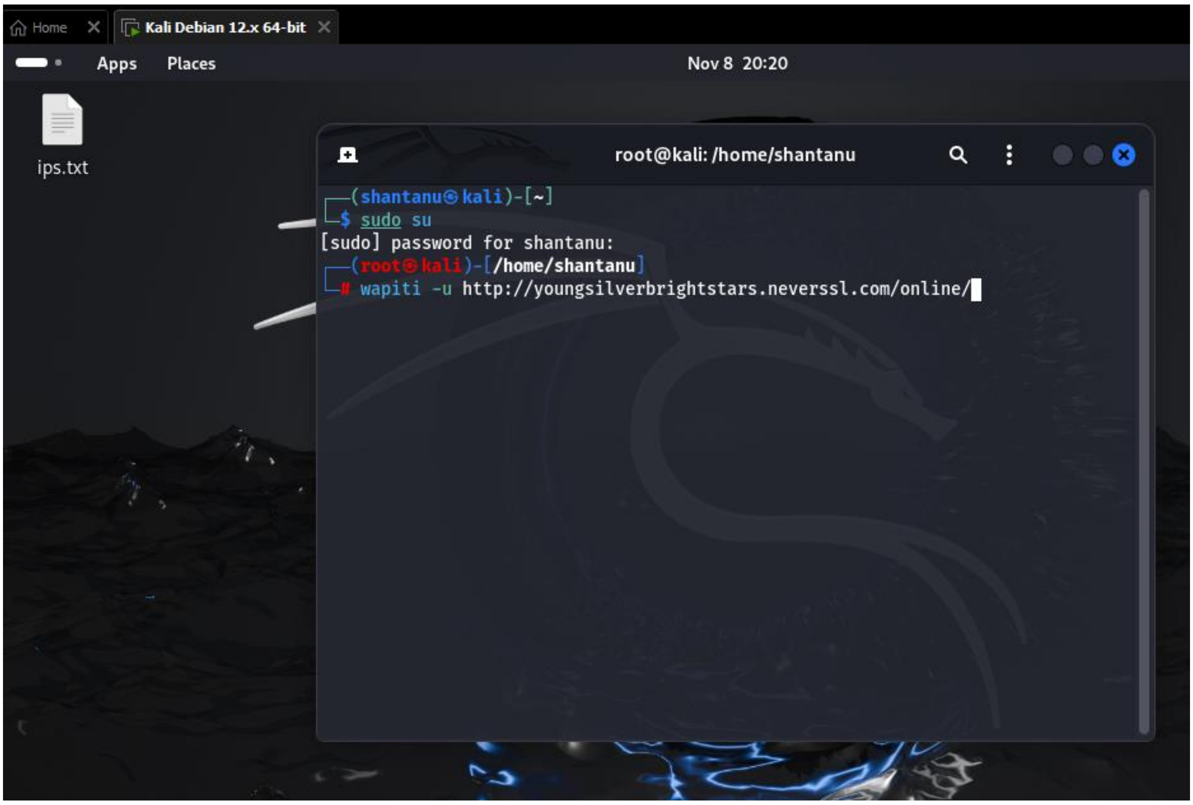
Step 3:

Enter this commands:

sudo su

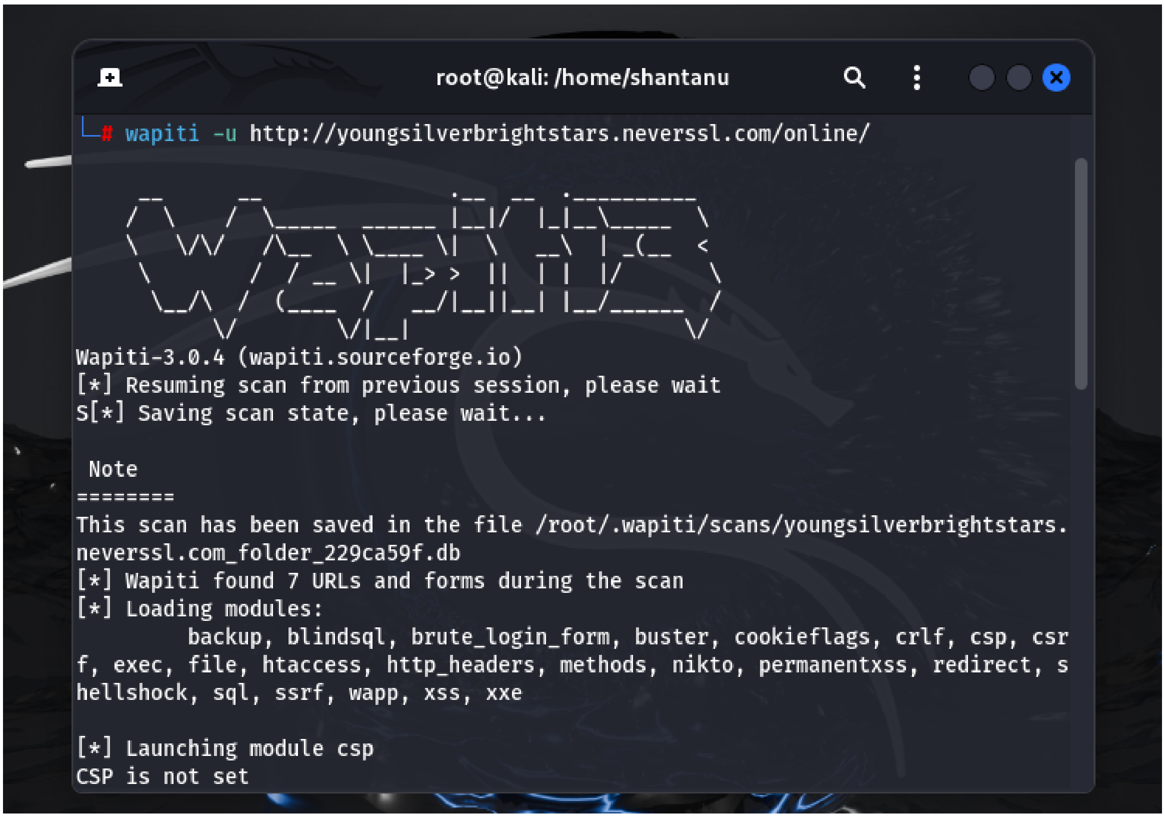
then use this command

wapiti -u http://youngsilverbrightstars.neverssl.com/online/



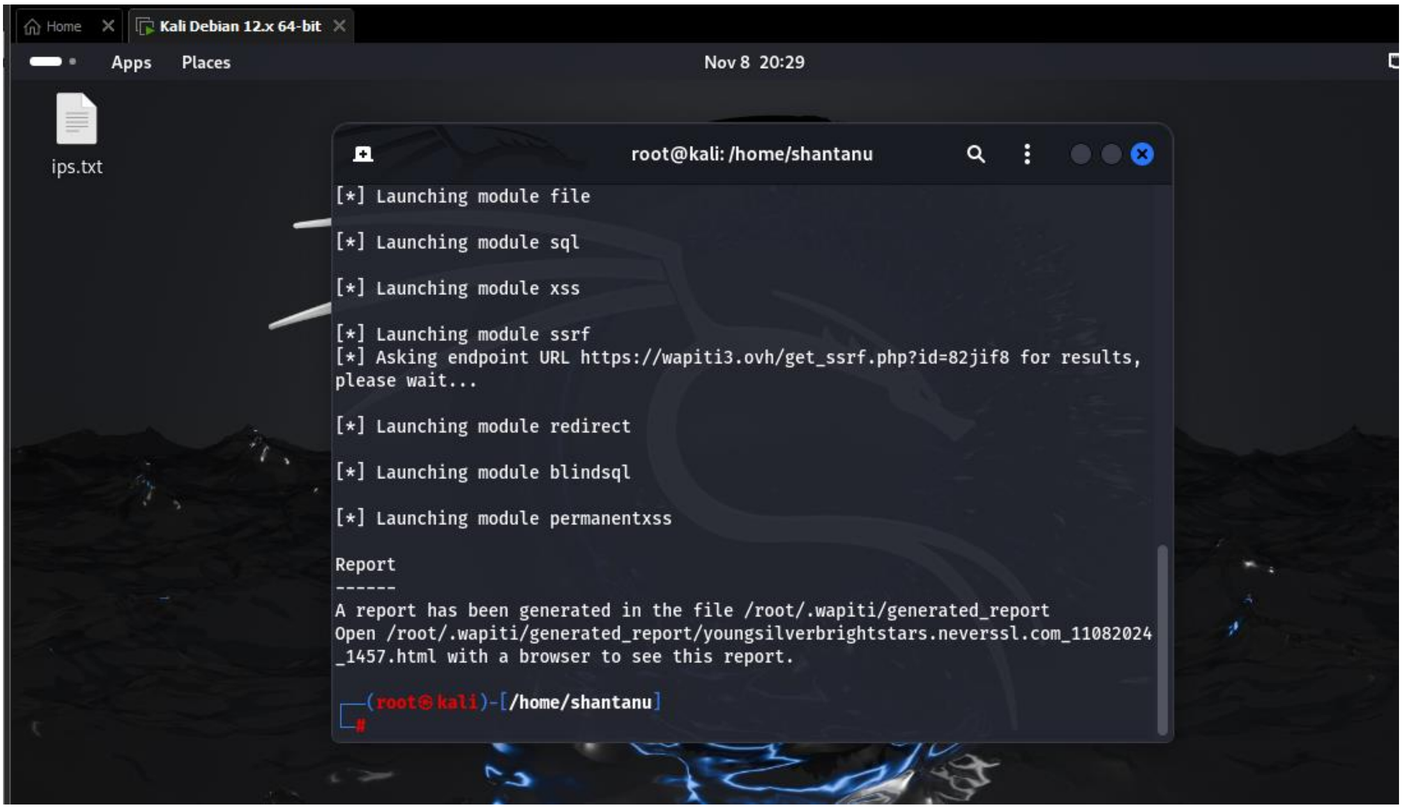
```
┌──(shantanu㉿kali)-[~]
└─$ sudo su
[sudo] password for shantanu:
┌──(root㉿kali)-[/home/shantanu]
└─# wapiti -u http://youngsilverbrightstars.neverssl.com/online/
```

STEP 4: WAIT FOR IT TO GENERATE REPORT



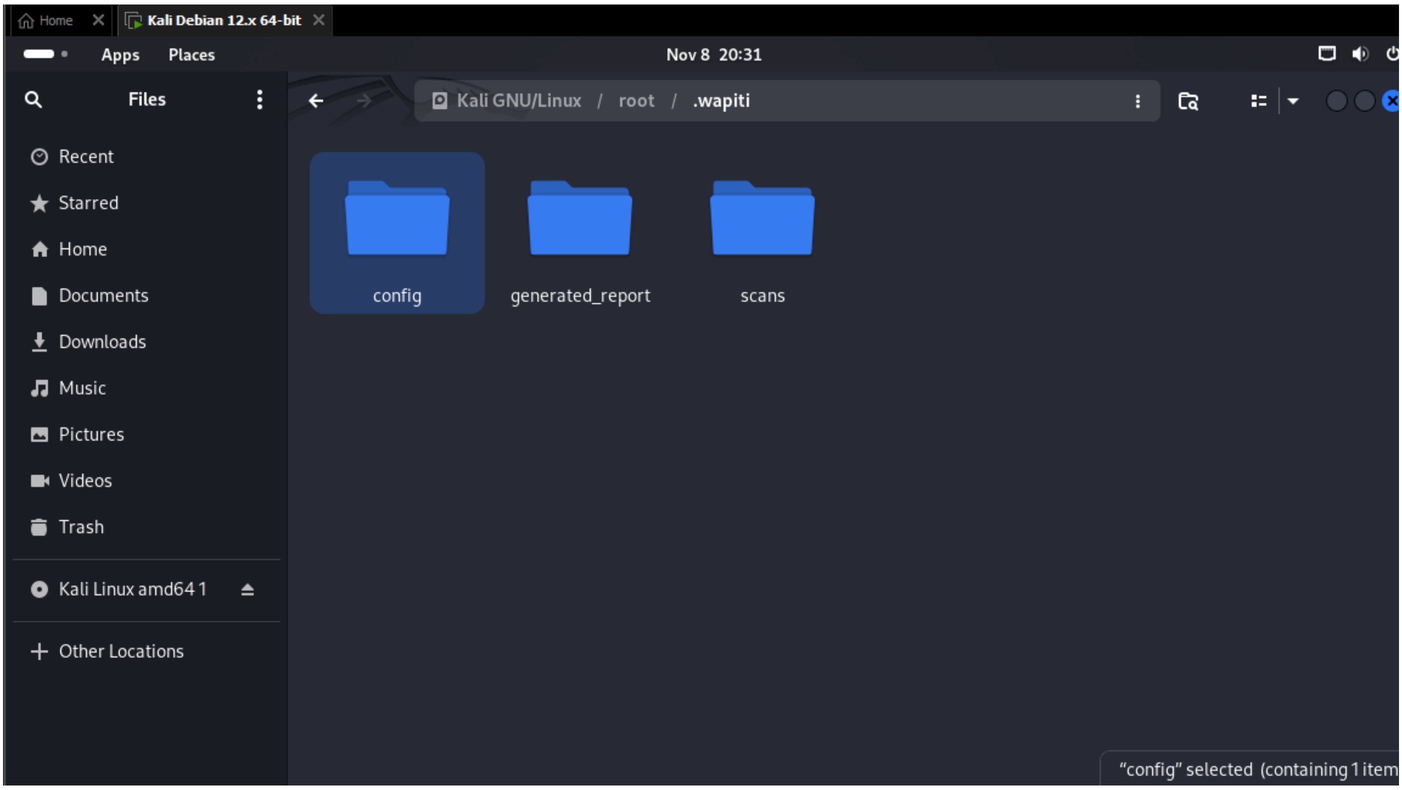
```
└─# wapiti -u http://youngsilverbrightstars.neverssl.com/online/


     __      __               .__  __  .__________
    /  \    /  \_____  ______ |__|/  |_|__\_____  \
    \   \/\/   /\__  \ \____ \|  \   __\  | _(__  <
     \        /  / __ \|  |_> >  ||  | |  |/       \
      \__/\  /  (____  /   __/|__||__| |__/______  /
           \/        \/|__|                      \/
Wapiti-3.0.4 (wapiti.sourceforge.io)
[*] Resuming scan from previous session, please wait
S[*] Saving scan state, please wait...

 Note
========
This scan has been saved in the file /root/.wapiti/scans/youngsilverbrightstars.
neverssl.com_folder_229ca59f.db
[*] Wapiti found 7 URLs and forms during the scan
[*] Loading modules:
         backup, blindsql, brute_login_form, buster, cookieflags, crlf, csp, csr
f, exec, file, htaccess, http_headers, methods, nikto, permanentxss, redirect, s
hellshock, sql, ssrf, wapp, xss, xxe

[*] Launching module csp
CSP is not set
```

STEP 5: Go to directory where it has generated the report ,in our case /root/.wapiti

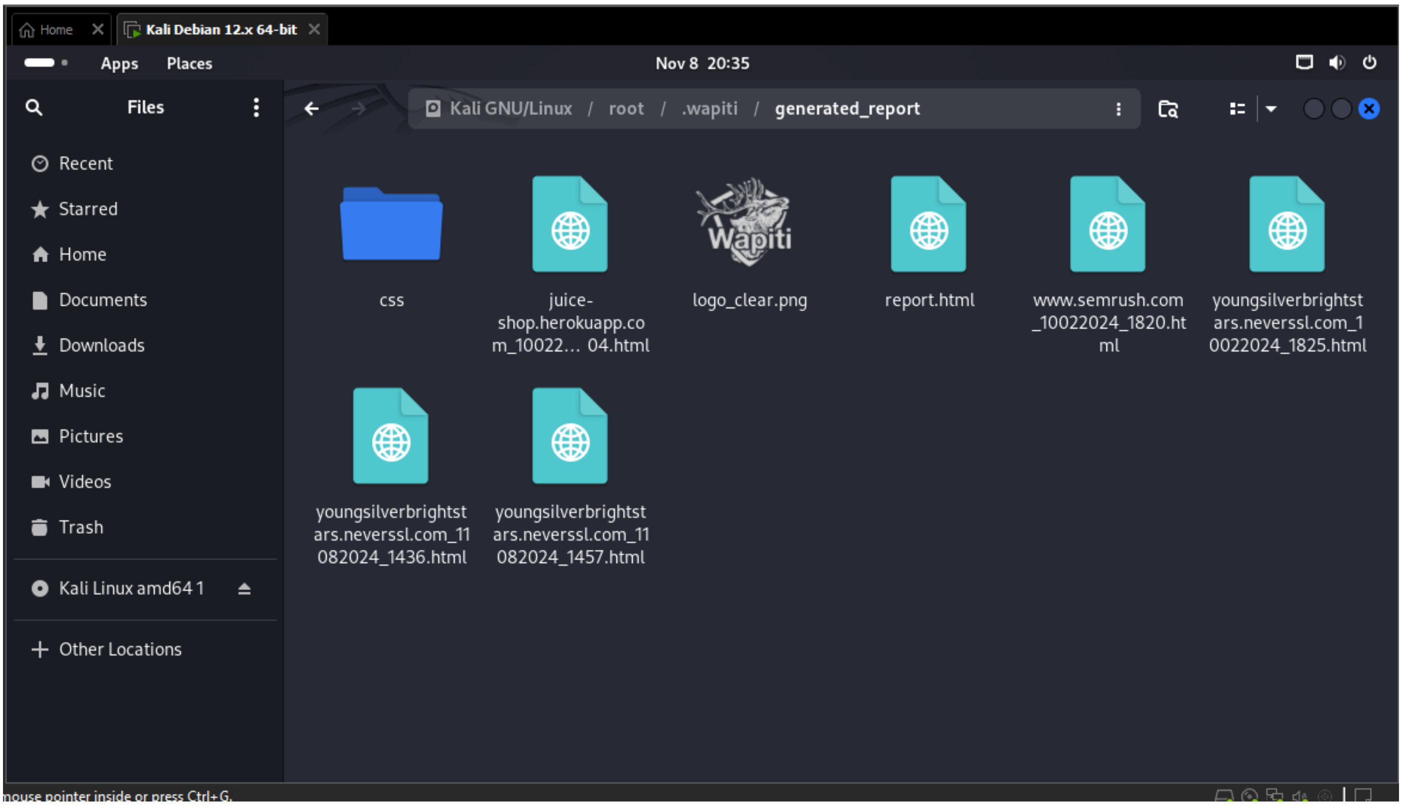


```
[*] Launching module file

[*] Launching module sql

[*] Launching module xss

[*] Launching module ssrf
[*] Asking endpoint URL https://wapiti3.ovh/get_ssrf.php?id=82jif8 for results,
please wait...

[*] Launching module redirect

[*] Launching module blindsql

[*] Launching module permanentxss

Report
------
A report has been generated in the file /root/.wapiti/generated_report
Open /root/.wapiti/generated_report/youngsilverbrightstars.neverssl.com_11082024
_1457.html with a browser to see this report.

┌──(root㉿kali)-[/home/shantanu]
└─#
```

STEP 6: Check for the GENBERATED REPORT FOLDER

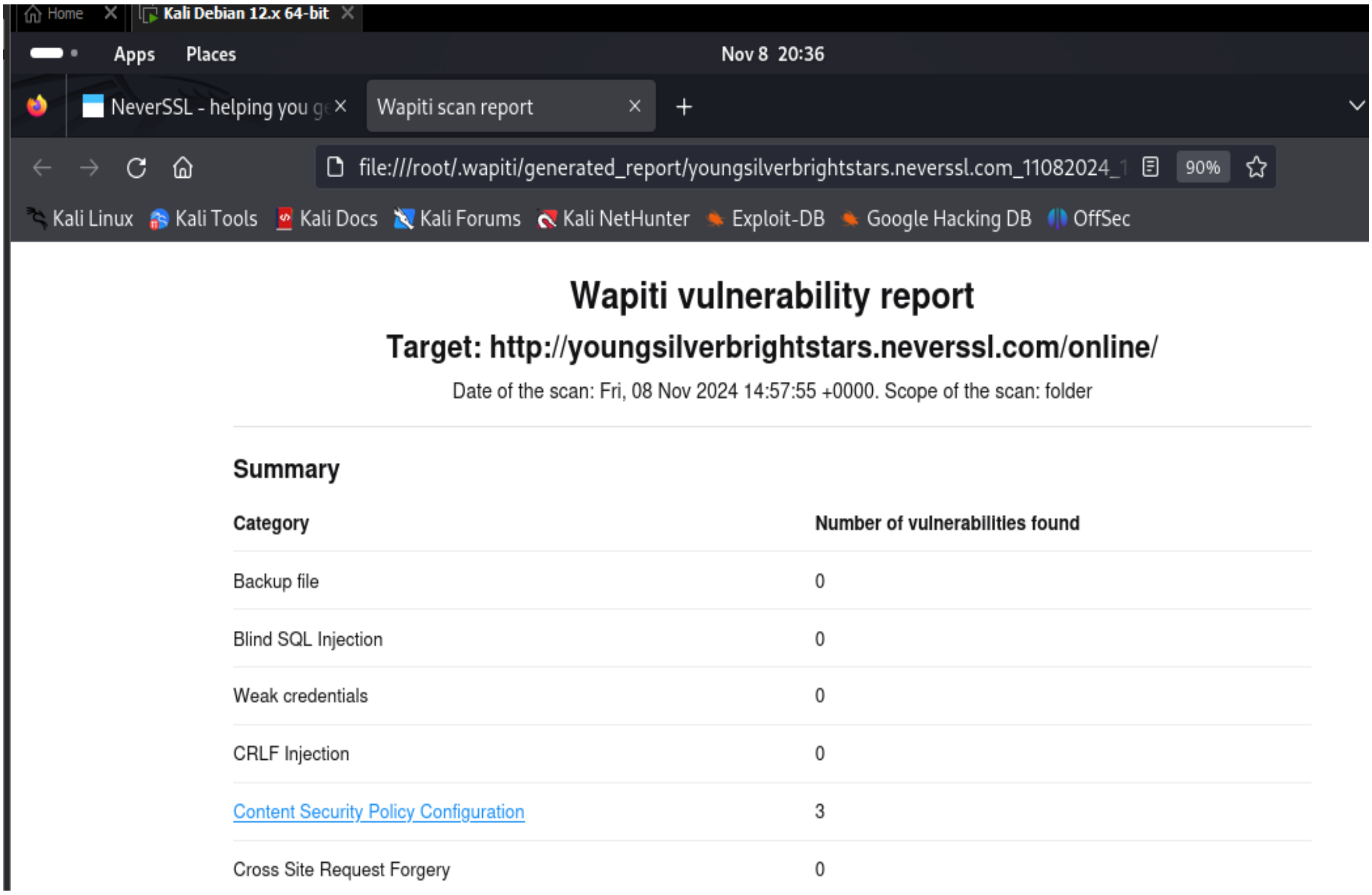


Step 7 : Find the Geneated report



Home
Kali Debian 12.x 64-bit
Apps
Places
Nov 8 20:35
Files
Recent
Starred
Home
Documents
Downloads
Music
Pictures
Videos
Trash
Kali Linux amd64 1
Other Locations
Kali GNU/Linux / root / .wapiti / generated_report
css
juice-shop.herokuapp.com_10022… 04.html
logo_clear.png
report.html
www.semrush.com_10022024_1820.html
youngsilverbrightstars.neverssl.com_10022024_1825.html
youngsilverbrightstars.neverssl.com_11082024_1436.html
youngsilverbrightstars.neverssl.com_11082024_1457.html
nouse pointer inside or press Ctrl+G.

RESULTS :



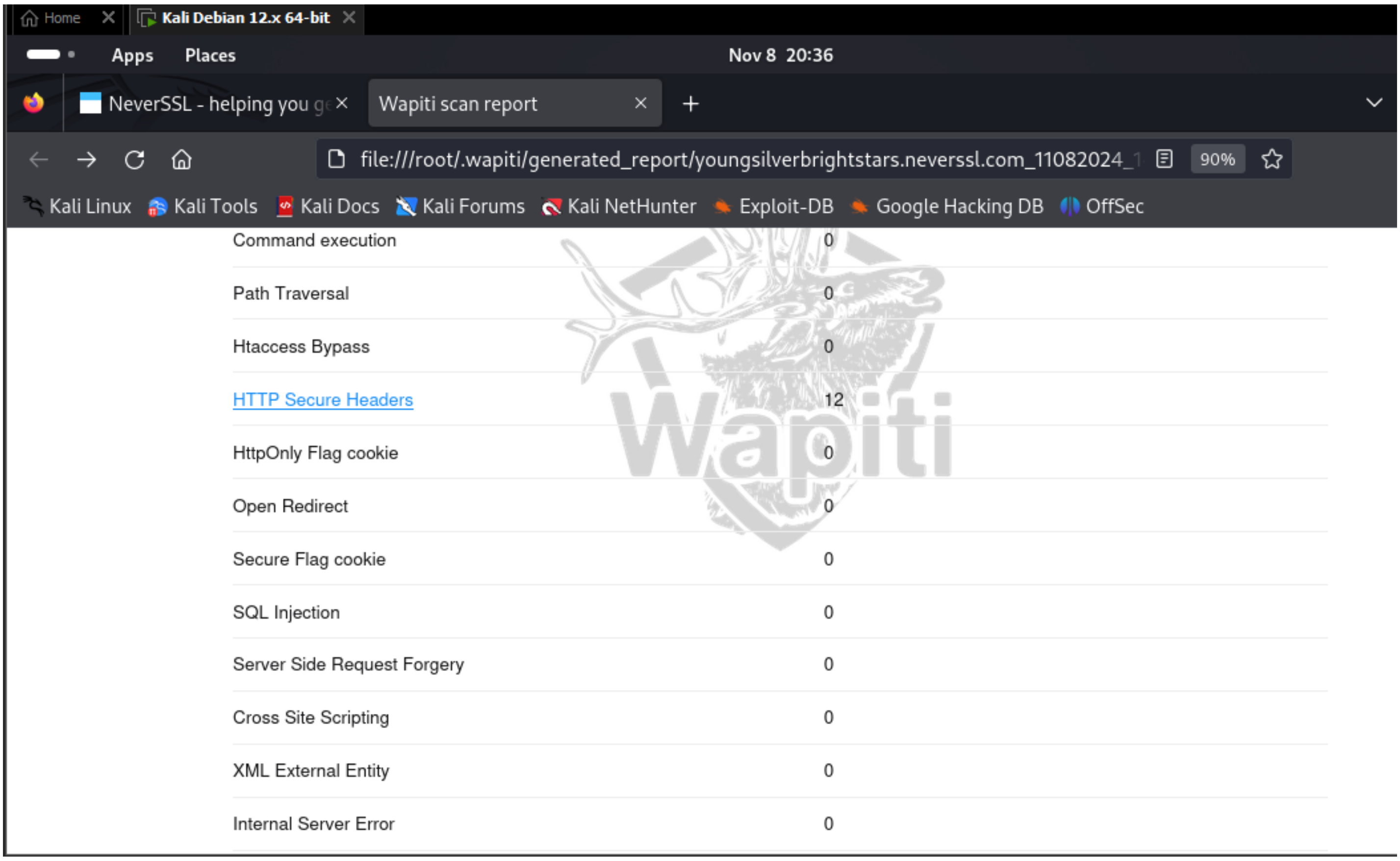
Wapiti vulnerability report

Target: http://youngsilverbrightstars.neverssl.com/online/

Date of the scan: Fri, 08 Nov 2024 14:57:55 +0000. Scope of the scan: folder

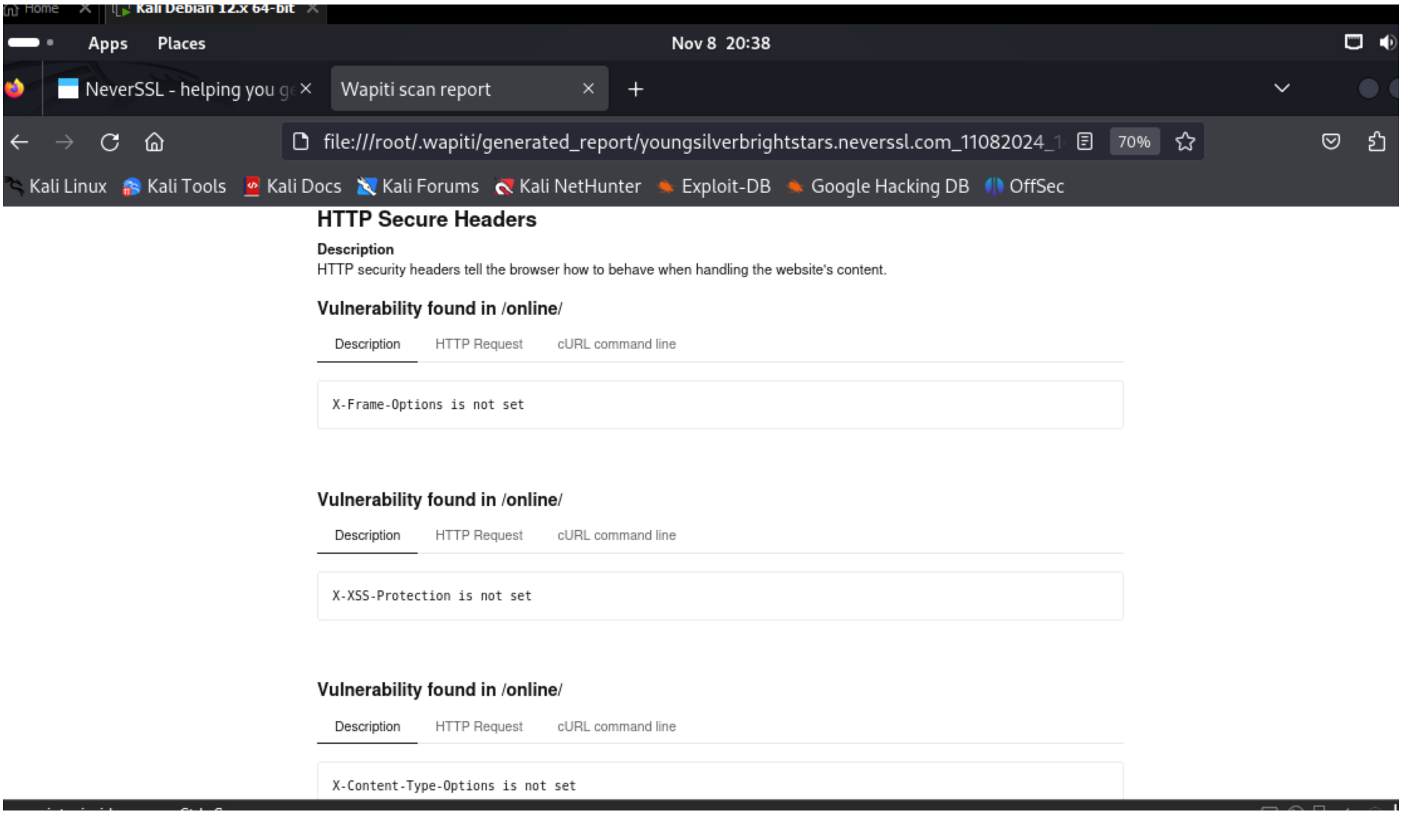
Summary

Category	Number of vulnerabilities found
Backup file	0
Blind SQL Injection	0
Weak credentials	0
CRLF Injection	0
Content Security Policy Configuration	3
Cross Site Request Forgery	0



Command execution	0
Path Traversal	0
Htaccess Bypass	0
HTTP Secure Headers	12
HttpOnly Flag cookie	0
Open Redirect	0
Secure Flag cookie	0
SQL Injection	0
Server Side Request Forgery	0
Cross Site Scripting	0
XML External Entity	0
Internal Server Error	0

EXPAND THE ISSUES TO SEE ISSUES IN DETAIL with Possible Sloutions



HTTP Secure Headers

Description

HTTP security headers tell the browser how to behave when handling the website's content.

Vulnerability found in /online/

Description HTTP Request cURL command line

```
X-Frame-Options is not set
```

Vulnerability found in /online/

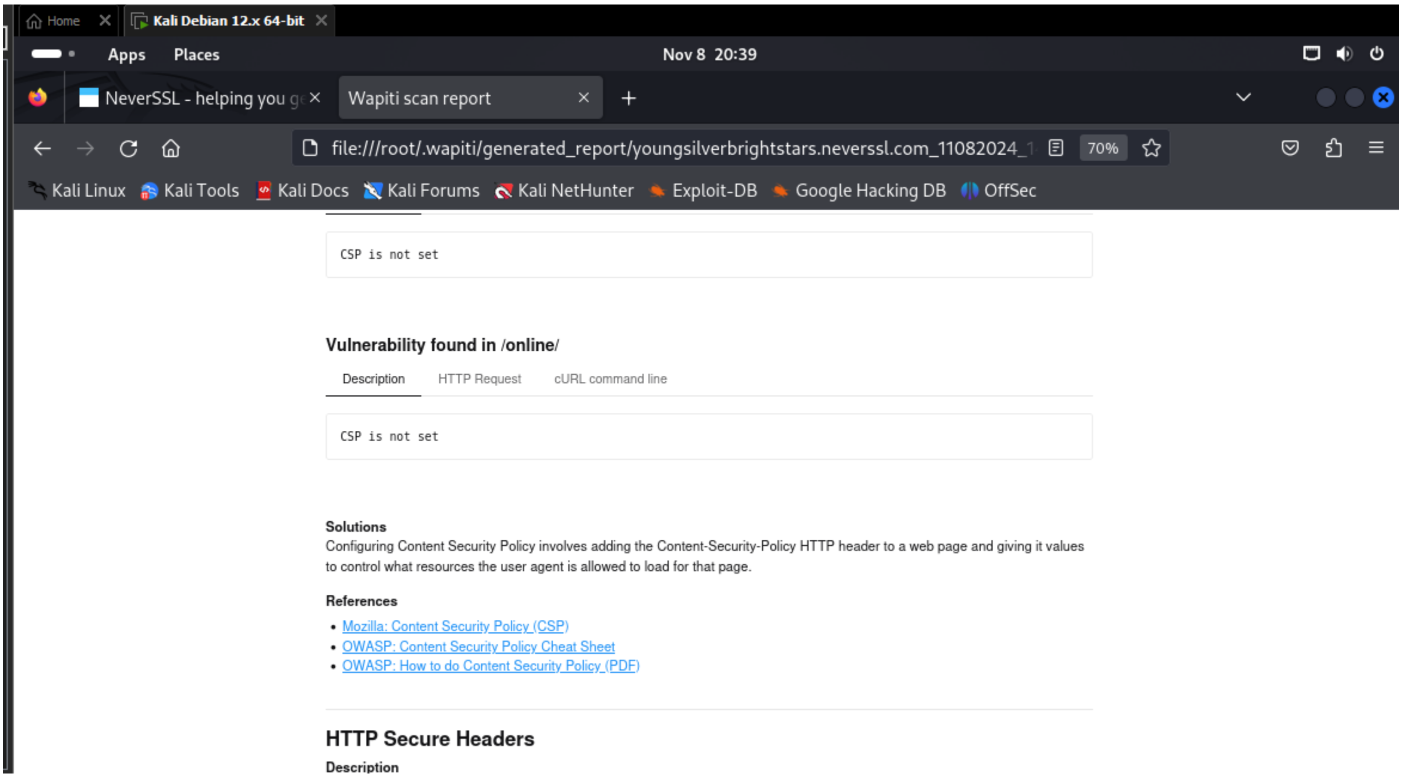
Description HTTP Request cURL command line

```
X-XSS-Protection is not set
```

Vulnerability found in /online/

Description HTTP Request cURL command line

```
X-Content-Type-Options is not set
```



```
CSP is not set
```

Vulnerability found in /online/

Description HTTP Request cURL command line

```
CSP is not set
```

Solutions

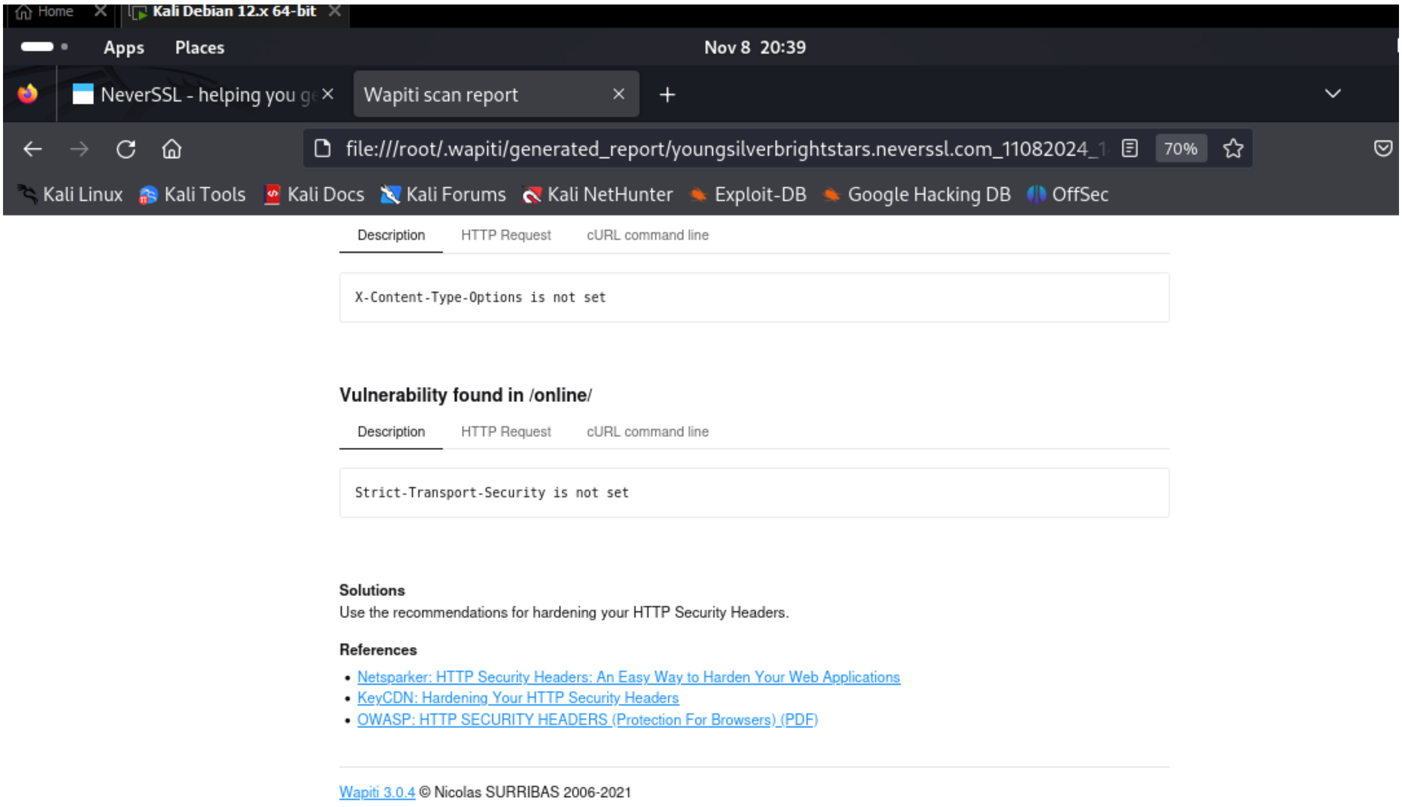
Configuring Content Security Policy involves adding the Content-Security-Policy HTTP header to a web page and giving it values to control what resources the user agent is allowed to load for that page.

References

- Mozilla: Content Security Policy (CSP)
- OWASP: Content Security Policy Cheat Sheet
- OWASP: How to do Content Security Policy (PDF)

HTTP Secure Headers

Description



Description HTTP Request cURL command line

```
X-Content-Type-Options is not set
```

Vulnerability found in /online/

Description HTTP Request cURL command line

```
Strict-Transport-Security is not set
```

Solutions

Use the recommendations for hardening your HTTP Security Headers.

References

- Netsparker: HTTP Security Headers: An Easy Way to Harden Your Web Applications
- KeyCDN: Hardening Your HTTP Security Headers
- OWASP: HTTP SECURITY HEADERS (Protection For Browsers) (PDF)

Wapiti 3.0.4 © Nicolas SURRIBAS 2006-2021

Recent Researches Literature Review on Cloud Security Posture Management (CSPM)

1. The Evolution of Cloud Security to Real-Time CSPM and Beyond

In their most recent writing, Aqua Security (n.d.) explain how the original concept of CSPM and the newer CSPM version have developed in a world that requires live risk analysis and action in cloud settings. The report also discusses shortcomings of earlier forms of CSPM, suggesting that mass, discrete scanning techniques can miss transient issues and real-time threats. It asserts that real-time visibility with contextual understanding, coupled with prioritized remediation is crucial to satisfy the dynamic nature of cloud security.

One strength of the report is that it explicates on context-aware CSPM, a form of program security management that mainly targets risk prioritization as opposed to flooding security teams with notifications. He further supposes that, failing such prioritization, organizations risk having a negative phenomenon known as 'alert fatigue', and end up missing some vital vulnerabilities. Furthermore, the report covers novelties which, for example, include agentless scanning, allowing an effective but rather incomplete assessment of the risk position based on the API of the cloud provider.

The most critical contribution of the report is the advocacy for real-time scanning capabilities in CSPM solutions to address the risks missed by the approaches described above while in workload. : This insight raises the question of the appropriateness of context-based CSPM solutions in the current and future states of the evolving cloud security space.

Citation: Aqua Security. (n.d.). *The Evolution of Cloud Security to Real-Time CSPM and Beyond*. Retrieved from https://www.aquasec.com/resources/cloud-security-posture-management-guide&#8203;:contentReference[oaicite:0]{index=0}.

2. Enhancing Cloud Security Posture Management - A Comprehensive Analysis and Experimental Validation of CSPM Strategies

In this work, Yadav et al. (2024) explore how CSPM can be used to increase compliance and decrease risks in cloud environments. To support this work, this study employs theoretical analysis and experimental validation on Cloud Custodian and Security Monkey to show how automated remediation helps keep multi-cloud systems compliant and addressing misconfigurations.

The research is most useful where it presents experimental outcomes, which give quantification to the given CSPM strategies concerning the cloud security environment. The authors elaborate on the benefits that automated tools afford for compliance enforcement, minimizing human mistakes, and identifying weak spots in real-time, the CSPM tools are crucial for vast cloud operations.

But one of the strong points of this paper is the quantitative analysis of CSPM tools reflecting real-life experience and offering practices for improvement to the readers and security professionals. This is why the adoption of the automated CSPM solutions that involves more than a static scan must be emphasized.

Citation: Yadav, S., Karthick, G., & Mukundha, C. H. (2024). Enhancing Cloud Security Posture Management - A Comprehensive Analysis and Experimental Validation of CSPM Strategies. *Nanotechnology Perceptions, 20*(5), 838-860.

3. An Innovative Model for Both Cloud Security and Privacy in Provision of Secure Cloud Services

Using both qualitative and quantitative factors, Reddy et al. (2019) propose a CSPM model that consists of four layers: the Outer layer, consisting of Privacy, the second layer: Identity layer, the third layer: Compliance layer and the fourth layer: Integration layer. Their model focuses on the several layers of security within networks, data and access proposing a full spectrum solution for dealing with threats from within organizations as well as from the outside world. Due to unique aspects of cloud computing, the authors focus on particular risks that can occur in cloud environments, including internal threats, malware introduction, and unauthorized access, and propose measures to improve the general state of cloud security.

This paper's key value add is in presenting a CSPM framework that compartmentalizes Cloud Security for ease of administration. This model is helpful for security practitioners working by the need to address multifaceted cloud environments in an orderly manner. Further, the study provides preliminary evidence that a layered CSPM model can be adopted by cloud providers to provide a solid CSPM solution.

A weakness of this paper is that there is no practical implementation of the model to benchmark with real Cloud-based applications. Still, it provides good theoretical base to further research of layered cloud security models in future.

Citation: Reddy, G. R., Sreenivasarao, D., & Saheb, S. K. (2019). An Innovative Model for Both Cloud Security and Privacy in Provision of Secure Cloud Services. *International Journal of Innovative Technology and Exploring Engineering (IJITEE), 9*(1), 2785-2789

4. Enhancing Cloud Security Posture for Ubiquitous Data Access with a Cybersecurity Framework-Based Management Tool

Coppola et al. (2023) put forward a Cloud Security Posture Management (CSPM) tool based on NIST Cybersecurity Framework (NIST CSF) that aims at enhancing the cloud security by using AI and big data techniques. The CSPM they developed is used to scan the cloud environment perpetually to make threat identification and misconfiguration alerts in AWS. The tool is expected to enhance security, given that AI can analyze large volumes of security information to arrive at robust security decisions.

Another interesting feature of the present work is the focus on AI implementation in CSPM, which can perform threat identification and mitigation independently. This approach reduces anxiety of human involvement significantly, the occurrence of errors common among humans and shortens the time taken to respond to such incidents. In the paper, the author also von emphasizes on the critical role of big data security to handle the complexities characterizing large cloud environments as data expands exponentially.

This study also has great value in showing how AI integrated CSPM tools can meet the broad and rapidly changing scale of cloud environments while providing a conceptual direction for applying AI in CSPM solutions.

Citation: Coppola, G., Varde, A. S., & Shang, J. (2023). Enhancing Cloud Security Posture for Ubiquitous Data Access with a Cybersecurity Framework-Based Management Tool. *Proceedings of IEEE UEMCON 2023*, 1-10. https://doi.org/10.1109/UEMCON59035.2023.10316003&#8203;:contentReference[oaicite:3]{index=3}.

5. **CSPM was Born to Tackle Cloud Misconfigurations**

Although misconfigurations present a major problem in cloud security, CSPM solutions were first created to guard against it, as noted in this report by Aqua Security (n.d.). The report outlines the shortcoming of a traditional CSPM approach in relation to live visibility and high noise ratios from excessive low-risk alerts. It promotes the effective CSPM solutions from today's perspective which targets real threats and that possesses features like monitoring to adapt to the CSP's continuous evolution.

The strength of the report is in substantiation of context as a requirement in CSPM by elaborating on how context-based solutions can help to eliminate clutter and let the security team concentrate on the challenges that are most critical. This is because the report provides a cogent case on how real-time visibility when combined with contextual risk priority can enhance the cloud security framework of an organisation.

The findings of this report are critical for analyzing the CSPM paradigm shift more deeply and identifying the growth of monitoring solutions that would be significant amid the constant updates of cyber threats on the cloud platform.

Citation: Aqua Security. (n.d.). *CSPM was Born to Tackle Cloud Misconfigurations*. Retrieved from https://www.aquasec.com/resources/cloud-security-posture-management-guide&#8203;:contentReference[oaicite:4]{index=4}

GITHUB LINKS

Skipfish : https://github.com/Shantanu10z/Skipfish

Wapiti: https://github.com/Shantanu10z/Wapiti

Bibliography

1. Aqua Security. (n.d.). *The Evolution of Cloud Security to Real-Time CSPM and Beyond*. Retrieved from https://www.aquasec.com/resources/cloud-security-posture-management-guide

2. Coppola, G., Varde, A. S., & Shang, J. (2023). Enhancing Cloud Security Posture for Ubiquitous Data Access with a Cybersecurity Framework-Based Management Tool. *Proceedings of IEEE UEMCON 2023*, 1-10. https://doi.org/10.1109/UEMCON59035.2023.10316003

3. Offensive Security. (n.d.). *Skipfish*. Retrieved from https://www.kali.org/tools/skipfish/

4. Reddy, G. R., Sreenivasarao, D., & Saheb, S. K. (2019). An Innovative Model for Both Cloud Security and Privacy in Provision of Secure Cloud Services. *International Journal of Innovative Technology and Exploring Engineering (IJITEE), 9*(1), 2785-2789.

5. Wapiti Project. (n.d.). *Wapiti Web Application Vulnerability Scanner*. Retrieved from https://wapiti-scanner.github.io/

6. Yadav, S., Karthick, G., & Mukundha, C. H. (2024). Enhancing Cloud Security Posture Management - A Comprehensive Analysis and Experimental Validation of CSPM Strategies. *Nanotechnology Perceptions, 20*(5), 838-860.

7. Aqua Security. (n.d.). *CSPM was Born to Tackle Cloud Misconfigurations*. Retrieved from https://www.aquasec.com/resources/cloud-security-posture-management-guide