Ex. No:1 Date:

Image Steganography

Aim:

Algorithm:

Code:

```
import cv2 import string import os d={}
c={}

for i in range(255): d[chr(i)]=i
c[i]=chr(i)

x=cv2.imread(r"C:\Users\TCS\Desktop\img.jpg") i=x.shape[0]
j=x.shape[1] print(i,j)

key=input("Enter key to edit(Security Key) : ") text=input("Enter text to hide : ")

kl=0 tln=len(text) z=0
n=0 m=0

l=len(text)

for i in range(l): x[n,m,z]=d[text[i]]^d[key[kl]] n=n+1
m=m+1 m=(m+1)%3
kl=(kl+1)%len(key)

cv2.imwrite("encrypted_img.jpg",x) os.startfile("encrypted_img.jpg")
print("Data Hiding in Image completed successfully.") #x=cv2.imread(“encrypted_img.jpg")


kl=0 tln=len(text) z=0
n=0 m=0
ch = int(input("\nEnter 1 to extract data from Image : ")) if ch == 1:
key1=input("\n\nRe enter key to extract text : ") decrypt=""

if key == key1 : for i in range(l):
decrypt+=c[x[n,m,z]^d[key[kl]]] n=n+1
m=m+1 m=(m+1)%3
kl=(kl+1)%len(key)
print("Encrypted text was : ",decrypt)
else:
print("Key doesn't matched.")
else:
print("Thank you. EXITING.")
```

Output:

```
Python 3.8.1 Shell
File  Edit  Shell  Debug  Options  Window  Help
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:/Users/TCS/AppData/Local/Programs/Python/Python38-32/steg.py ===
309 464
Enter key to edit(Security Key) : 54637
Enter text to hide : harini
Data Hiding in Image completed successfully.

Enter 1 to extract data from Image : 1


Re enter key to extract text : 54637
Encrypted text was :  h
Encrypted text was :  ha
Encrypted text was :  har
Encrypted text was :  hari
Encrypted text was :  harin
Encrypted text was :  harini
>>>
```

Result:

Ex. No: 2 Date:

Configuration of Network – commands

After undertaking initial stealthy reconnaissance to identify IP address spaces of interest, network scanning is an intrusive and aggressive process used to identify accessible hosts and their network services. The rationale behind IP network scanning is to gain insight into the following elements of a given network:

- ICMP message types that generate responses from target hosts
- Accessible TCP and UDP network services running on the target hosts
- Operating platforms of target hosts and their configurations
- Areas of vulnerability within target host IP stack implementations (including sequence number predictability for TCP spoofing and session hijacking)
- Configuration of filtering and security systems (including firewalls, border routers, switches, and IDS/IPS mechanisms)

Performing both network scanning and reconnaissance tasks paints a clear picture of the network topology and its security features. Before penetrating the target network, specific network service probing is undertaken to enumerate vulnerabilities and weaknesses, covered in later chapters of this book.

ICMP Probing

Internet Control Message Protocol (ICMP) probes can be used to identify potentially weak and poorly protected networks and hosts. ICMP is a short messaging protocol, used by systems administrators for continuity testing of networks in particular (using tools such as ping and traceroute). From a network scanning perspective, the following types of ICMP messages are useful:

Type 8 (echo request)

Echo request messages are also known as ping packets. You can use a scanning tool such as Nmap to perform ping sweeping and easily identify hosts that are accessible.

Type 13 (timestamp request)

A timestamp request message is used to obtain the system time information from the target host. The response is in a decimal format and is the number of milliseconds elapsed since midnight GMT.

Type 15 (information request)

The ICMP information request message was intended to support self-configuring systems such as diskless workstations at boot time to allow them to discover their network addresses. Protocols such as RARP, BOOTP, or DHCP achieve this more robustly, so type 15 messages are rarely used.

Type 17 (subnet address mask request)

An address mask request message reveals the subnet mask used by the target host. This information is useful when mapping networks and identifying the size of subnets and network spaces used by organizations.

Firewalls of security-conscious organizations often blanket-filter inbound ICMP messages, and so ICMP probing isn't effective; however, ICMP isn't filtered in most cases, as these messages are useful during network troubleshooting.

ICMP Probing Tools
A number of tools can be used to perform ICMP probing, including SING, Nmap, and ICMP Scan. These utilities and their benefits are discussed here.

SING
Send ICMP Nasty Garbage (SING) is a command-line utility that sends customizable ICMP probes. The main purpose of the tool is to replace the ping command with certain enhancements, including the ability to transmit and receive spoofed packets, send MAC-spoofed packets, and support the transmission of many other message types, including ICMP address mask, timestamp, and information requests, as well as router solicitation and router advertisement messages.

In these examples, I direct probes at broadcast addresses and individual hosts.

Using SING to send broadcast ICMP echo request messages:

$ sing -echo 192.168.0.255
SINGing to 192.168.0.255 (192.168.0.255): 16 data bytes
16 bytes from 192.168.0.1: seq=0 ttl=64 TOS=0 time=0.230 ms 16 bytes from 192.168.0.155: seq=0 ttl=64 TOS=0 time=2.267 ms 16 bytes from 192.168.0.126: seq=0 ttl=64 TOS=0 time=2.491 ms 16 bytes from 192.168.0.50: seq=0 ttl=64 TOS=0 time=2.202 ms 16 bytes from 192.168.0.89: seq=0 ttl=64 TOS=0 time=1.572 ms Using SING to send ICMP timestamp request messages:

$ sing -tstamp 192.168.0.50
SINGing to 192.168.0.50 (192.168.0.50): 20 data bytes
20 bytes from 192.168.0.50: seq=0 ttl=128 TOS=0 diff=327372878 20 bytes from 192.168.0.50: seq=1 ttl=128 TOS=0 diff=1938181226* 20 bytes from 192.168.0.50: seq=2 ttl=128 TOS=0 diff=1552566402* 20 bytes from 192.168.0.50: seq=3 ttl=128 TOS=0 diff=1183728794* Using SING to send ICMP address mask request messages:

$ sing -mask 192.168.0.25
SINGing to 192.168.0.25 (192.168.0.25): 12 data bytes
12 bytes from 192.168.0.25: seq=0 ttl=236 mask=255.255.255.0 12 bytes from 192.168.0.25: seq=1 ttl=236 mask=255.255.255.0 12 bytes from 192.168.0.25: seq=2 ttl=236 mask=255.255.255.0 12 bytes from 192.168.0.25: seq=3 ttl=236 mask=255.255.255.0
There are a handful of other ICMP message types that have other security implications, such as ICMP type 5 redirect messages sent by routers, which allow for traffic redirection. These messages aren't related to network scanning, and so they are not detailed here. For details of traffic redirection using ICMP, including exploit code.

Nmap

Nmap can perform ICMP ping sweep scans of target IP blocks easily. Many hardened networks will blanket-filter inbound ICMP messages at border routers or firewalls, so sweeping in this fashion isn't effective in some cases. Nmap can be run from a Unix-based or Windows command prompt to perform an ICMP ping sweep against 192.168.0.0/24, as shown in Example 4-1.

Example 4-1. Performing a ping sweep with Nmap

$ nmap -sP -PI 192.168.0.0/24

Starting Nmap 4.10 at 2007-04-01 20:39 UTC Host 192.168.0.0 seems to be a subnet broadcast address (2 extra pings).
Host 192.168.0.1 appears to be up. Host 192.168.0.25 appears to be up. Host 192.168.0.32 appears to be up. Host 192.168.0.50 appears to be up. Host 192.168.0.65 appears to be up. Host 192.168.0.102 appears to be up. Host 192.168.0.110 appears to be up. Host 192.168.0.155 appears to be up.
Host 192.168.0.255 seems to be a subnet broadcast address (2 extra pings). Nmap finished: 256 IP addresses (8 hosts up) scanned in 17.329 seconds

ICMPScan

ICMPScan is a bulk scanner that sends type 8, 13, 15, and 17 ICMP messages, derived from Nmap. The tool is very useful in that it can process inbound responses by placing the network interface into promiscuous mode, thereby identifying internal IP addresses and machines that respond from probes sent to subnet network and broadcast addresses. Example 4-2 shows ICMPScan being run against an internal network block. Because ICMP is a connectionless protocol, it is best practice to resend each probe (using -r 1) and set the timeout to 500 milliseconds (using -t 500). We also set the tool to listen in promiscuous mode for unsolicited responses (using the -c flag).

Example 4-2. Running ICMPScan

```
$ icmpscan
Usage: icmpscan [ options ] target [...]
-i <interface> Specify interface.
-c          Enable promiscuous mode.
-A <address>   Specify source address of generated packets.
-t <timeout>   Specify timeout for probe response.
-r <retries>   Retries per probe.
-f <filename> Read targets from the specified file.
-E, -P   ICMP Echo Probe
-T, -S   Timestamp
-N, -M Netmask
-I          Info
-R          Router solicitation
-h          Display usage information
```

-v Increase verbosity
-B Enable debugging output.
-n Numeric output (do not resolve hostnames)

$ icmpscan -c -t 500 -r 1 192.168.1.0/24 192.168.1.0: Echo (From 192.168.1.17!)
192.168.1.0: Address Mask [255.255.255.0] (From 192.168.1.17!)
192.168.1.7 : Echo
192.168.1.7: Timestamp [0x03ab2db0, 0x02d4c507, 0x02d4c507] 192.168.1.7: Address Mask [255.255.255.0]
192.168.1.8 : Echo
192.168.1.8: Address Mask [255.255.255.0]

Identifying Subnet Network and Broadcast Addresses
Nmap identifies subnet network and broadcast addresses by counting the number of ICMP echo replies for each IP address during an ICMP ping sweep. Such addresses respond with multiple replies, providing insight into the target network and its segmentation. In Example 4-3 we use Nmap to enumerate subnet network and broadcast addresses in use for a given network (154.14.224.0/26).

Example 4-3. Enumerating subnet network and broadcast addresses with Nmap

$ nmap -sP 154.14.224.0/26

Starting Nmap 4.10 (http://www.insecure.org/nmap/) at 2007-04-01 20:39 UTC Host 154.14.224.16 seems to be a subnet broadcast address (returned 1 extra pings). Host pipex-gw.abc.co.uk (154.14.224.17) appears to be up.
Host mail.abc.co.uk (154.14.224.18) appears to be up. Host 154.14.224.25 appears to be up.
Host intranet.abc.co.uk (154.14.224.26) appears to be up. Host 154.14.224.27 appears to be up.
Host 154.14.224.30 appears to be up.
Host 154.14.224.31 seems to be a subnet broadcast address (returned 1 extra pings). Host 154.14.224.32 seems to be a subnet broadcast address (returned 1 extra pings). Host pipex-gw.smallco.net (154.14.224.33) appears to be up.
Host mail.smallco.net (154.14.224.34) appears to be up.
Host 154.14.224.35 seems to be a subnet broadcast address (returned 1 extra pings). Host 154.14.224.40 seems to be a subnet broadcast address (returned 1 extra pings). Host pipex-gw.example.org (154.14.224.41) appears to be up.
Host gatekeeper.example.org (154.14.224.42) appears to be up.
Host 154.14.224.43 appears to be up.
Host 154.14.224.47 seems to be a subnet broadcast address (returned 1 extra pings). This scan has identified six subnets within the 154.14.224.0/26 network, as follows:

An unused or filtered block from 154.14.224.0 to 154.14.224.15 (14 usable addresses) The abc.co.uk block from 154.14.224.16 to 154.14.224.31 (14 usable addresses)
The smallco.net block from 154.14.224.32 to 154.14.224.35 (2 usable addresses)

An unused or filtered block from 154.14.224.36 to 154.14.224.39 (2 usable addresses) The example.org block from 154.14.224.40 to 154.14.224.47 (6 usable addresses)
An unused or filtered block from 154.14.224.48 to 154.14.224.63 (14 usable addresses)

Gleaning Internal IP Addresses

In some cases, it is possible to gather internal IP address information by analyzing ICMP responses from an ICMP ping sweep. Upon sending ICMP echo requests to publicly accessible IP addresses, firewalls often use Network Address Translation (NAT) or similar IP masquerading to forward the packets on to internal addresses, which then respond to the probes. Other scenarios include poor routing configuration on routers that are probed using ICMP, where they respond to the probes from a different interface.

Stateful inspection mechanisms and sniffers can be used to monitor for ICMP responses from internal IP addresses in relation to your original probes. Tools such as Nmap and SING don't identify these responses from private addresses, as low-level stateful analysis of the traffic flowing into and out of a network is required. A quick and simple example of this behavior can be seen in the ISS BlackICE personal firewall event log in Figure 4-1 as a simple ICMP ping sweep is performed.

ISS BlackICE used to statefully glean internal IP addresses
Figure 4-1. ISS BlackICE used to statefully glean internal IP addresses
This figure shows that BlackICE has identified four unsolicited ICMP echo replies from private addresses (within the 172.16.0.0/12 space in this case, but they are often within 192.168.0.0/16 or 10.0.0.0/8).

ICMP Scan supports this type of internal IP address discovery when in promiscuous mode. It is beneficial to run a network sniffer such as Ethereal or tcpdump during testing to pick up on unsolicited ICMP responses, including "ICMP TTL exceeded" (type 11 code 0) messages, indicating a routing loop, and "ICMP administratively prohibited" (type 3 code 13) messages, indicating an ACL in use on a router or firewall.

OS Fingerprinting Using ICMP
Ofir Arkin's Xprobe2 utility performs OS fingerprinting by primarily analyzing responses to ICMP probes. See the Sys-Security Group web site (http://www.sys-security.com) for further details, including white papers and presentations that describe the Xprobe2 fingerprinting technology and approach. Example 4-4 shows Xprobe2 being used to fingerprint a remote host.

Example 4-4. Operating system fingerprinting using Xprobe 2

$ xprobe2 -v 192.168.0.174

[+] Target is 192.168.0.174
[+] Loading modules.
[+] Following modules are loaded:

[x] [1] ping:icmp_ping - ICMP echo discovery module
[x] [2] ping:tcp_ping - TCP-based ping discovery module
[x] [3] ping:udp_ping - UDP-based ping discovery module
[x] [4] infogather:ttl_calc - TCP and UDP based TTL distance calculation
[x] [5] infogather:portscan - TCP and UDP PortScanner
[x] [6] fingerprint:icmp_echo - ICMP Echo request fingerprinting module
[x] [7] fingerprint:icmp_tstamp - ICMP Timestamp request fingerprinting module
[x] [8] fingerprint:icmp_amask - ICMP Address mask request fingerprinting module
[x] [9] fingerprint:icmp_port_unreach - ICMP port unreachable fingerprinting module
[x] [10] fingerprint:tcp_hshake - TCP Handshake fingerprinting module
[x] [11] fingerprint:tcp_rst - TCP RST fingerprinting module
[x] [12] fingerprint:smb - SMB fingerprinting module
[13] fingerprint:snmp - SNMPv2c fingerprinting module [+] 13 modules registered
[+] Initializing scan engine [+] Running scan engine
[+] Host: 192.168.0.174 is up (Guess probability: 100%)
[+] Target: 192.168.0.174 is alive. Round-Trip Time: 0.00015 sec [+] Selected safe Round-Trip Time value is: 0.00030 sec
[+] Primary guess:
[+] Host 192.168.0.174 Running OS: "Sun Solaris 5 (SunOS 2.5)" (Guess probability: 100%)
[+] Other guesses:
[+] Host 192.168.0.174 Running OS: "Sun Solaris 6 (SunOS 2.6)" (Guess probability: 100%)
[+] Host 192.168.0.174 Running OS: "Sun Solaris 7 (SunOS 2.7)" (Guess probability: 100%)
[+] Host 192.168.0.174 Running OS: "Sun Solaris 8 (SunOS 2.8)" (Guess probability: 100%)
[+] Host 192.168.0.174 Running OS: "Sun Solaris 9 (SunOS 2.9)" (Guess probability: 100%)
[+] Host 192.168.0.174 Running OS: "Mac OS 9.2.x" (Guess probability: 95%) [+] Host 192.168.0.174 Running OS: "HPUX B.11.0 x" (Guess probability: 95%) [+] Host 192.168.0.174 Running OS: "Mac OS X 10.1.5" (Guess probability: 87%) [+] Host 192.168.0.174 Running OS: "FreeBSD 4.3" (Guess probability: 87%)
[+] Host 192.168.0.174 Running OS: "FreeBSD 4.2" (Guess probability: 87%) TCP Port Scanning
Accessible TCP ports can be identified by port scanning target IP addresses. The following nine different types of TCP port scanning are used in the wild by both attackers and security consultants:

Standard scanning methods Vanilla connect ( ) scanning

Half-open SYN flag scanning

Stealth TCP scanning methods Inverse TCP flag scanning

ACK flag probe scanning TCP fragmentation scanning
Third-party and spoofed TCP scanning methods FTP bounce scanning

Proxy bounce scanning

Sniffer-based spoofed scanning IP ID header scanning

What follows is a technical breakdown for each TCP port scanning type, along with details of Windows- and Unix-based tools that can perform scanning.

Standard Scanning Methods

Standard scanning methods, such as vanilla and half-open SYN scanning, are extremely simple direct techniques used to accurately identify accessible TCP ports and services. These scanning methods are reliable but are easily logged and identified.

Vanilla connect( ) scanning

TCP connect( ) port scanning is the simplest type of probe to launch. There is no stealth whatsoever involved in this form of scanning, as a full TCP/IP connection is established with each port of the target host.

TCP/IP robustness means that connect( ) port scanning is an accurate way to determine which TCP services are accessible on a given host. However, due to the way that a full three-way handshake is performed, an aggressive connect( ) scan could antagonize or break poorly written network services. Figure 4-2 and Figure 4-3 show the various TCP packets involved and their flags.

In Figure 4-2, the attacker first sends a SYN probe packet to the port he wishes to test. Upon receiving a packet from the port with the SYN and ACK flags set, he knows that the port is open. The attacker completes the three-way handshake by sending an ACK packet back.

A vanilla TCP scan result when a port is open

Figure 4-2. A vanilla TCP scan result when a port is open

If, however, the target port is closed, the attacker receives an RST/ACK packet directly back, as shown in Figure 4-3.

A vanilla TCP scan result when a port is closed

Figure 4-3. A vanilla TCP scan result when a port is closed Tools that perform connect( ) TCP scanning.

Nmap can perform a TCP connect( ) port scan using the -sT flag. A benefit of this scanning type is that superuser root access is not required, as raw network sockets are not used. Other very simple scanners exist, including pscan.c, which is available as source code from many sites including Packet Storm.

Ex. No:3 Date:

WIRELESS AUDIT

Aim:

Algorithm:

Output:

root@kali:~# iwconfig
eth0 no wireless extensions.

wlan0 IEEE 802.11bgn ESSID:off/any
Mode:Managed Access Point: Not-Associated Tx-Power=20 dBm Retry short limit:7 RTS thr:off
Fragment thr:off Encryption
key:off Power Management:of f
lo no wireless extensions.

root@kali:~# iwlist wlan0 scanning
wlan0 Scan completed :
Cell 01 - Address: 14:F6:5A:F4:57:22
Channel:6
Frequency:2.437 GHz (Channel 6) Quality=70/70 Signal level=- 27 dBm Encryption key:on
ESSID:"BENEDICT"
Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s
Bit Rates:6 Mb/s; 9 Mb/s; 12 Mb/s; 18 Mb/s; 24 Mb/s
36 Mb/s; 48 Mb/s; 54 Mb/s
Mode:Master Extra:tsf=00000000425 b0a37 Extra: Last beacon: 548ms ago IE: WPA Version
1Group Cipher : TKIP
Pairwise Ciphers (2) : CCMP TKIP Authentication Suites (1) : PSK

root@kali:~# airmon-ng start wlan0

Found 2 processes that could cause trouble.
If airodump-ng, aireplay-ng or airtun-ng stops working after a short period of time, you may want to kill (some of) them!

PID Name
1148 NetworkManager
1324 wpa_supplicant

PHY Interface	Driver	Chipset
phy0 wlan0	ath9k_htc	Atheros Communications, Inc. AR9271 802.11n

Newly created monitor mode interface wlan0mon is *NOT* in monitor mode. Removing non-monitor wlan0mon interface...

WARNING: unable to start monitor mode, please run "airmon-ng check kill"

root@kali:~# airmon-ng check kill
Killing these processes: PID Name
1324 wpa_supplicant

root@kali:~# airmon-ng start wlan0

PHY Interface	Driver	Chipset
phy0 wlan0	ath9k_htc	Atheros Communications, Inc. AR9271 802.11n

(mac80211 monitor mode vif enabled for [phy0]wlan0 on [phy0]wlan0mon) (mac80211 station mode vif disabled for [phy0]wlan0)

root@kali:~# airodump-ng -w atheros -c 6 --bssid 14:F6:5A:F4:57:22 wlan0mon CH 6][Elapsed: 5 mins][2016-10-05 01:35][WPA handshake: 14:F6:5A:F4:57:
BSSID PWR RXQ Beacons #Data, #/s CH MB ENC CIPHER AUTH E

14:F6:5A:F4:57:22 -31 100 3104 10036 0 6 54e. WPA CCMP

PSK B

BSSID STATION PWR Rate Lost Frames Probe

14:F6:5A:F4:57:22 70:05:14:A3:7E:3E -32 2e- 0

0 10836
root@kali:~# ls -l
total 10348
-rw-r--r-- 1 root root 10580359 Oct 5 01:35 atheros-01.cap
-rw-r--r-- 1 root root 481 Oct 5 01:35 atheros-01.csv
-rw-r--r-- 1 root root 598 Oct 5 01:35 atheros-01.kismet.csv
-rw-r--r-- 1 root root 2796 Oct 5 01:35 atheros-01.kismet.netxml

root@kali:~# aircrack-ng -a 2 atheros-01.cap -w /usr/share/wordlists/rockyou.txt
[00:00:52] 84564 keys tested (1648.11 k/s)

KEY FOUND! [rec12345]

Master Key : CA 53 9B 5C 23 16 70 E4 84 53 16 9E FB 14 77 49 A9 7A A0
2D 9F BB 2B C3 8D 26 D2 33 54 3D 3A 43

Transient Key : F5 F4 BA AF 57 6F 87 04 58 02 ED 18 62 37 8A 53
38 86 F1 A2 CA 0D 4A 8D D6 EC ED 0D 6C 1D C1 AF
81 58 81 C2 5D 58 7F FA DE 13 34 D6 A2 AE FE 05 F6 53 B8 CA A0 70 EC 02 1B EA 5F 7A DA
7A EC 7D

EAPOL HMAC 0A 12 4C 3D ED BD EE C0 2B C9 5A E3 C1 65 A8 5C

Result:

Ex. No:4 Date:

LINUX AUDITING USING LYNIS

Aim:

Description:

Output:
root@kali:~/Downloads# lynis audit system [Lynis 2.6.2]
#####
Lynis comes with ABSOLUTELY NO WARRANTY. This is free software, and you are welcome to redistribute it under the terms of the GNU General Public License.
See the LICENSE file for details about using this software.

2007-2018, CISOfy – https://cisofy.com/lynis/
Enterprise support available (compliance, plugins, interface and tools)
#####

[+] Initializing program

- Detecting OS… [DONE]
- Checking profiles… [DONE]

Program version: 2.6.2
Operating system: Linux Operating system name: Debian
Operating system version: kali-rolling Kernel version: 5.2.0
Hardware platform: x86_64
Hostname: kali

Profiles: /etc/lynis/default.prf
Log file: /var/log/lynis.log
Report file: /var/log/lynis-report.dat
Report version: 1.0
Plugin directory: /etc/lynis/plugins

Auditor: `[Not Specified]
Language: en
Test category: all
Test group: all

Program update status… [WARNING] [+] System Tools
- Scanning available tools…
- Checking system binaries… [+] Software: firewalls
- Checking iptables kernel module [FOUND]
- Checking iptables policies of chains [FOUND]
- Checking for empty ruleset [WARNING]
- Checking for unused rules [OK]
- Checking host based firewall [ACTIVE] [+] Security frameworks
- Checking presence AppArmor [FOUND]
- Checking AppArmor status [DISABLED]
- Checking presence SELinux [NOT FOUND]
- Checking presence grsecurity [NOT FOUND]

- Checking for implemented MAC framework [NONE] [+] Software: file integrity
- Checking file integrity tools
- Checking presence integrity tool [NOT FOUND] [+] Software: System tooling
- Checking automation tooling
- Automation tooling [NOT FOUND]
- Checking for IDS/IPS tooling [NONE] [+] Software: Malware
- Checking chkrootkit [FOUND] [+] File Permissions
- Starting file permissions check

Lynis security scan details:

Hardening index : 56 [###########]
Tests performed : 222
Plugins enabled : 1

Result:

Ex. No:5 Date:

SNORT IDS

Aim:

Description:

Algorithm:

Output:

[root@localhost security lab]# cd /usr/src

[root@localhost security lab]# wget https://www.snort.org/downloads/snort/daq- 2.0.7.tar.gz

[root@localhost security lab]# wget https://www.snort.org/downloads/snort/snort- 2.9.16.1.tar.gz

[root@localhost security lab]# tar xvzf daq-2.0.7.tar.gz

[root@localhost security lab]# tar xvzf snort-2.9.16.1.tar.gz [root@localhost security lab]# yum install libpcap* pcre* libdnet* -y [root@localhost security lab]# cd daq-2.0.7

[root@localhost security lab]# . /configure [root@localhost security lab]# make [root@localhost security lab]# make install

[root@localhost security lab]# cd snort-2.9.16.1 [root@localhost security lab]# . /configure

[root@localhost security lab]# make [root@localhost security lab]# make install [root@localhost security lab]# snort --version

,,_ -*> Snort! <*-

o")~ Version 2.9.8.2 GRE (Build 335) ""   By Martin Roesch & The Snort Team:

http://www.snort.org/contact#team Copyright (C) 2014-2015 Cisco and/or its affiliates. All rights reserved. Copyright (C) 1998-2013 Sourcefire, Inc., et al.

Using libpcap version 1.7.3 Using PCRE version: 8.38 2015-

11-23 Using ZLIB version: 1.2.8 [root@localhost security lab]# mkdir

/etc/snort [root@localhost security lab]# mkdir /etc/snort/rules [root@localhost security lab]# mkdir /var/log/snort [root@localhost security lab]# vi

/etc/snort/snort.conf

add this line- include /etc/snort/rules/icmp.rules

[root@localhost security lab]# vi /etc/snort/rules/icmp.rules

alert icmp any any -> any any (msg:"ICMP Packet"; sid:477; rev:3;)

[root@localhost security lab]# snort -i enp3s0 -c /etc/snort/snort.conf -l /var/log/snort/ Another terminal

[root@localhost security lab]# ping www.yahoo.com Ctrl + C

[root@localhost security lab]# vi /var/log/snort/alert

[**] [1:477:3] ICMP

Packet [**] [Priority: 0]

10/06-15:03:11.187877 192.168.43.148 -> 106.10.138.240

ICMP TTL:64 TOS:0x0 ID:45855 IpLen:20

DgmLen:84 DF Type:8 Code:0 ID:14680 Seq:64 ECHO

[**] [1:477:3] ICMP

Packet [**] [Priority: 0]

10/06-15:03:11.341739 106.10.138.240 -> 192.168.43.148

ICMP TTL:52 TOS:0x38 ID:2493 IpLen:20

DgmLen:84 Type:0 Code:0 ID:14680 Seq:64 ECHO REPLY

[**] [1:477:3] ICMP

Packet [**] [Priority: 0]

10/06-15:03:12.189727 192.168.43.148 -> 106.10.138.240

ICMP TTL:64 TOS:0x0 ID:46238 IpLen:20

DgmLen:84 DF Type:8 Code:0 ID:14680 Seq:65 ECHO

[**] [1:477:3] ICMP

Packet [**] [Priority: 0]

10/06-15:03:12.340881 106.10.138.240 -> 192.168.43.148

ICMP TTL:52 TOS:0x38 ID:7545 IpLen:20

DgmLen:84 Type:0 Code:0 ID:14680 Seq:65 ECHO REPLY

Result:

Ex. No:6 Date:

LINUX OS HARDENING

Aim:

Algorithm:

Output:

[root@localhost ~]# chkconfig --list |grep '3:on'

Note: This output shows SysV services only and does not include native systemd services. SysV configuration data might be overridden by native systemd configuration.

If you want to list systemd services use 'systemctl list-unit-files'.
To see services enabled on particular target use 'systemctl list-dependencies [target]'.

snortd 0:off 1:off 2:on 3:on 4:on 5:on 6:off [root@localhost ~]# chkconfig snortd off

[root@localhost ~]# chkconfig --list|grep snortd

Note: This output shows SysV services only and does not include native systemd services. SysV configuration data might be overridden by native systemd configuration.

If you want to list systemd services use 'systemctl list-unit-files'.
To see services enabled on particular target use 'systemctl list-dependencies [target]'.
snortd 0:off 1:off 2:off 3:off 4:off 5:off 6:off [root@localhost ~]# yum update all
[root@localhost ~]# vi /etc/modprobe.d/no-usb [root@localhost ~]# sestatus
SELinux status: enabled
SELinuxfs mount: /sys/fs/selinux SELinux root directory:
/etc/selinux Loaded policy name: targeted Current mode:
permissive Mode from config file: permissive Policy MLS status:
enabled
Policy deny_unknown status: allowed Memory protection checking: actual (secure)
Max kernel policy version: 31

[root@localhost ~]# passwd -l p201711 Locking password for user p201711. passwd: Success

[root@localhost ~]# cat /etc/shadow | awk -F: '($2==""){print $1}' [root@localhost ~]# systemctl enable iptables

[root@localhost ~]# echo ALL >>/etc/cron.deny [root@localhost ~]# vi /etc/sysconfig/network

Result:

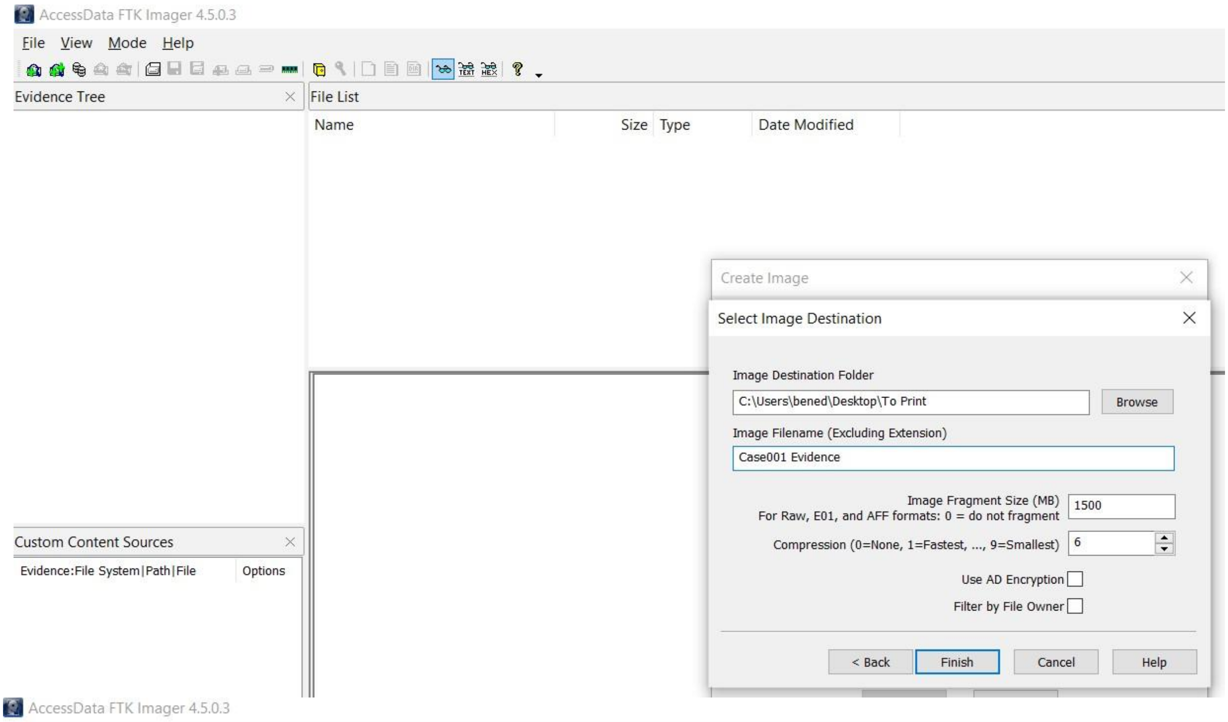
Ex. No:7 Date:

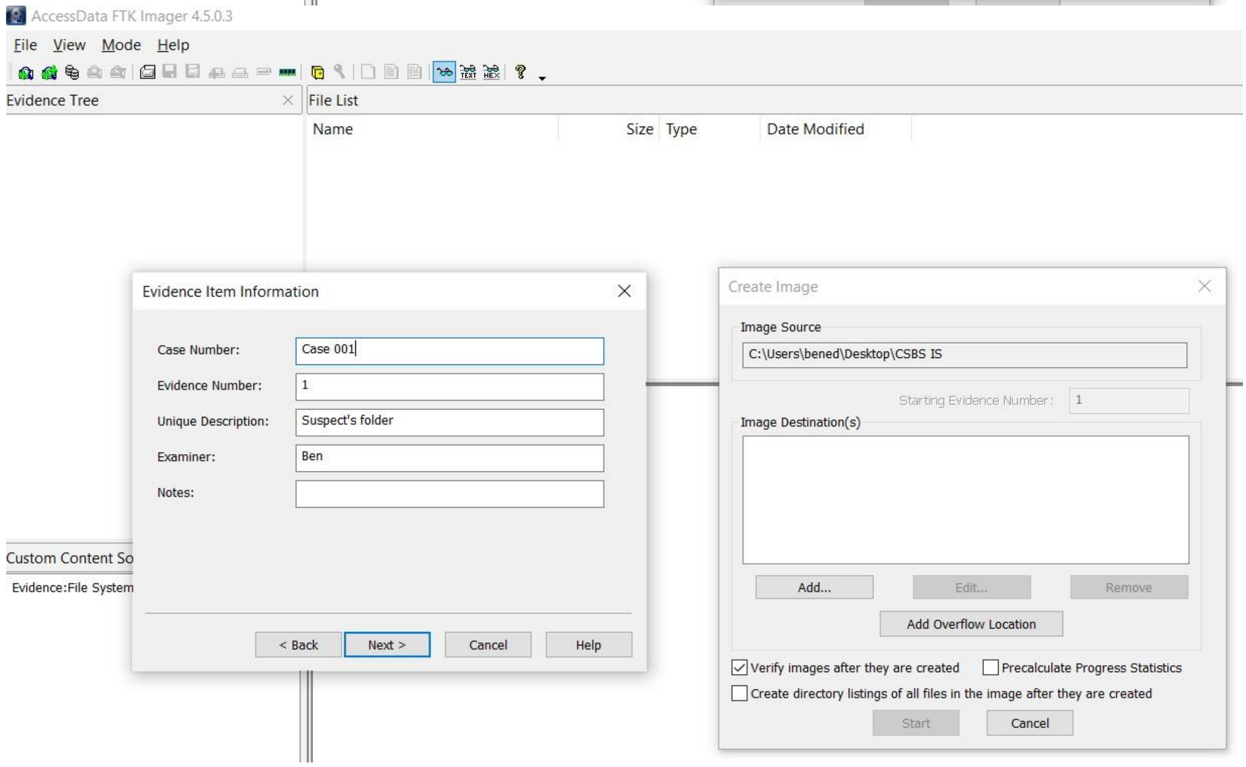
LIVE DATA ACQUISITION OF A FOLDER

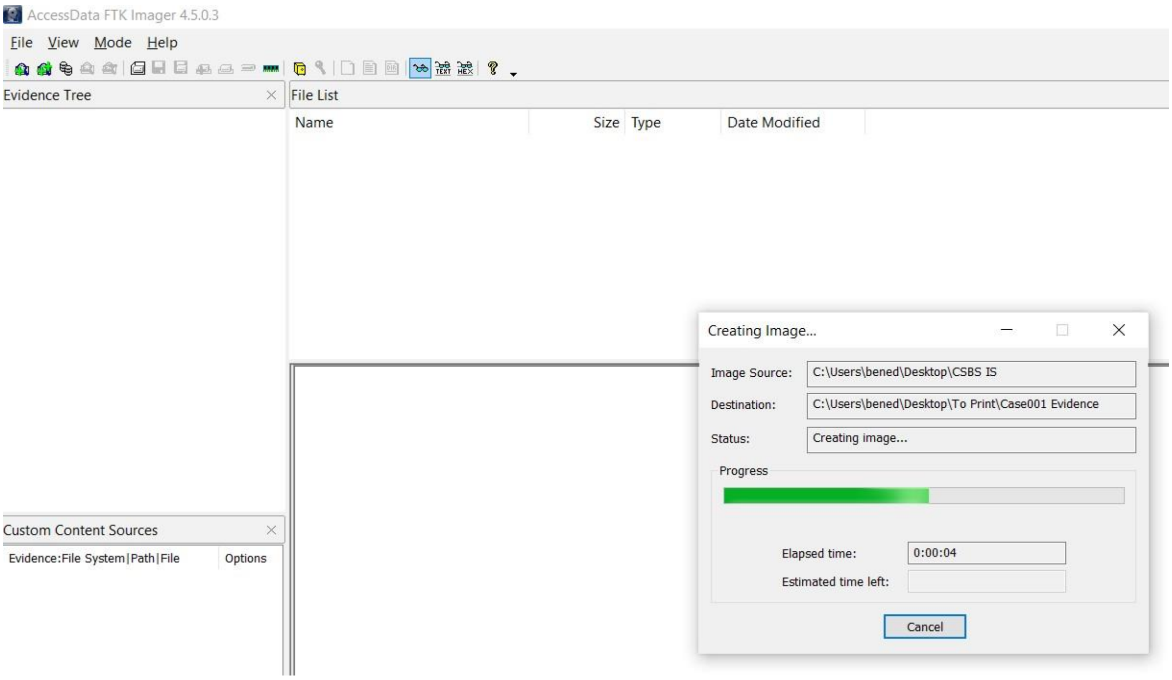
Aim:

Algorithm:

Output:


AccessData FTK Imager 4.5.0.3
File View Mode Help
Evidence Tree
File List
Name
Size
Type
Date Modified
Custom Content Sources
Evidence:File System|Path|File
Options
Create Image
Select Image Destination
Image Destination Folder
C:\Users\bened\Desktop\To Print
Browse
Image Filename (Excluding Extension)
Case001 Evidence
Image Fragment Size (MB)
For Raw, E01, and AFF formats: 0 = do not fragment
1500
Compression (0=None, 1=Fastest, ..., 9=Smallest)
6
Use AD Encryption
Filter by File Owner
< Back
Finish
Cancel
Help


AccessData FTK Imager 4.5.0.3
File View Mode Help
Evidence Tree
File List
Name
Size
Type
Date Modified
Evidence Item Information
Case Number:
Case 001
Evidence Number:
1
Unique Description:
Suspect's folder
Examiner:
Ben
Notes:
Custom Content So
Evidence:File System
< Back
Next >
Cancel
Help
Create Image
Image Source
C:\Users\bened\Desktop\CSBS IS
Starting Evidence Number:
1
Image Destination(s)
Add...
Edit...
Remove
Add Overflow Location
Verify images after they are created
Precalculate Progress Statistics
Create directory listings of all files in the image after they are created
Start
Cancel



Created By AccessData® FTK® Imager 4.5.0.3 Case Information:
Acquired using: ADI4.5.0.3 Case Number: case001 Evidence Number: 1
Unique Description: CSBS IS Examiner: Benedict
Notes:

Information for C:\Users\bened\Desktop\To Print\CSBSFOLDIMG.ad1: [Computed Hashes]
MD5 checksum: 61ccca5209aa38b26609597275da9b24
SHA1 checksum: 7d3c0de861466782e07d20b64c899959bdbd8ce6 Image information:
Acquisition started: Fri May 27 10:34:25 2022
Acquisition finished: Fri May 27 10:34:36 2022 Segment list:
C:\Users\bened\Desktop\To Print\CSBSFOLDIMG.ad1 Image Verification Results:
Verification started: Fri May 27 10:34:36 2022
Verification finished: Fri May 27 10:34:37 2022
MD5 checksum: 61ccca5209aa38b26609597275da9b24 : verified
SHA1 checksum: 7d3c0de861466782e07d20b64c899959bdbd8ce6 : verified

Result:

Ex. No.: 8 Date:

WEB VULNERABILITES USING O-SAFT

Aim:

Description:

Algorithm:

Output:
[root@localhost]# tar xvjf o-saft.tgz [root@localhost]#cd O-Saft
[root@localhost O-Saft]# ./o-saft +info rajalakshmi.org
./o-saft.pl +info rajalakshmi.org | cat
**WARNING: 149: no executable for '/usr/local/openssl/bin/openssl' found; all openssl functionality disabled
!!Hint: consider using '--openssl=/path/to/openssl'
**WARNING: 058: given path '/etc/ssl/certs/' does not contain a CA file
**WARNING: 060: no PEM file for CA found; using '--ca-file=/etc/ssl/certs/ca- certificates.crt'
Use of uninitialized value $_no in regexp compilation at ./o-saft.pl line 230. Use of uninitialized value $_no in regexp compilation at ./o-saft.pl line 230. Use of uninitialized value $_no in regexp compilation at ./o-saft.pl line 230.
**WARNING: if default file does not exist, some certificate checks may fail
!!Hint: use '--ca-file=/full/path/ca-certificates.crt' Given hostname: rajalakshmi.org
IP for given hostname: 14.99.10.232
Reverse resolved hostname: static-232.10.99.14-tataidc.co.in
DNS entries for given hostname: 14.99.10.232 static-232.10.99.14-tataidc.co.in;
**WARNING: 204: Can't make a connection to 'rajalakshmi.org:443' without SNI; no initial data (compare with and without SNI not possible)
**WARNING: 203: connection without SNI succeded with errors; errors ignored
!!Hint: use '--v' to show more information about Net::SSLinfo::do_ssl_open() errors
**WARNING: 205: Can't make a connection to 'rajalakshmi.org:443'; target ignored
!!Hint: use '--v' to show more information
!!Hint: use '--socket-reuse' it may help in some cases
!!Hint: use '--ignore-no-conn' to disable this check [root@localhost O-Saft]# ./o-saft +cipher rajalakshmi.org
./o-saft.pl +cipher rajalakshmi.org | cat
!!Hint: +cipher : functionality changed, please see 'o-saft.pl --help=TECHNIC'
**WARNING: 149: no executable for '/usr/local/openssl/bin/openssl' found; all openssl functionality disabled
!!Hint: consider using '--openssl=/path/to/openssl'
**WARNING: 058: given path '/etc/ssl/certs/' does not contain a CA file
**WARNING: 060: no PEM file for CA found; using '--ca-file=/etc/ssl/certs/ca- certificates.crt'
Use of uninitialized value $_no in regexp compilation at ./o-saft.pl line 230. Use of uninitialized value $_no in regexp compilation at ./o-saft.pl line 230. Use of uninitialized value $_no in regexp compilation at ./o-saft.pl line 230.
**WARNING: if default file does not exist, some certificate checks may fail
!!Hint: use '--ca-file=/full/path/ca-certificates.crt'
**WARNING: 409: SSLv2 does not support SNI; cipher checks are done without SNI
**WARNING: 409: SSLv3 does not support SNI; cipher checks are done without SNI ECDHE-RSA-AES256-SHA yes HIGH
DHE-RSA-AES256-SHA yes HIGH
DHE-RSA-CAMELLIA256-SHA yes HIGH ECDHE-RSA-AES128-SHA yes HIGH
DHE-RSA-AES128-SHA yes HIGH
DHE-RSA-CAMELLIA128-SHA yes HIGH AES256-SHA yes HIGH
CAMELLIA256-SHA yes HIGH AES128-SHA yes HIGH CAMELLIA128-SHA yes HIGH
DHE-RSA-SEED-SHA yes MEDIUM SEED-SHA yes MEDIUM
IDEA-CBC-SHA yes weak ECDHE-RSA-AES256-SHA yes HIGH DHE-RSA-

AES256-SHA yes HIGH
DHE-RSA-CAMELLIA256-SHA yes HIGH ECDHE-RSA-AES128-SHA yes HIGH
DHE-RSA-AES128-SHA yes HIGH
DHE-RSA-CAMELLIA128-SHA yes HIGH AES256-SHA yes HIGH
CAMELLIA256-SHA yes HIGH AES128-SHA yes HIGH CAMELLIA128-SHA yes HIGH
DHE-RSA-SEED-SHA yes MEDIUM SEED-SHA yes MEDIUM
IDEA-CBC-SHA yes weak ECDHE-RSA-AES256-GCM-SHA384 yes HIGH
DHE-RSA-AES256-GCM-SHA384 yes HIGH
ECDHE-RSA-CHACHA20-POLY1305-SHA256yes HIGH DHE-RSA-CHACHA20-POLY1305-SHA256 yes HIGH DHE-RSA-AES256-CCM8 yes HIGH
DHE-RSA-AES256-CCM yes HIGH ECDHE-ARIA256-GCM-SHA384 yes HIGH
DHE-RSA-ARIA256-GCM-SHA384 yes HIGH ECDHE-RSA-AES128-GCM-SHA256 yes HIGH DHE-RSA-AES128-GCM-SHA256 yes HIGH DHE-RSA-AES128-CCM8 yes HIGH
DHE-RSA-AES128-CCM yes HIGH ECDHE-ARIA128-GCM-SHA256 yes HIGH
DHE-RSA-ARIA128-GCM-SHA256 yes HIGH ECDHE-RSA-AES256-SHA384 yes HIGH DHE-RSA-AES256-SHA256 yes HIGH ECDHE-RSA-CAMELLIA256-SHA384 yes HIGH DHE-RSA-CAMELLIA256-SHA256yes HIGH ECDHE-RSA-AES128-SHA256 yes HIGH DHE-RSA-AES128-SHA256 yes HIGH ECDHE-RSA-CAMELLIA128-SHA256 yes HIGH DHE-RSA-CAMELLIA128-SHA256yes HIGH
ECDHE-RSA-AES256-SHA yes HIGH
DHE-RSA-AES256-SHA yes HIGH
DHE-RSA-CAMELLIA256-SHA yes HIGH ECDHE-RSA-AES128-SHA yes HIGH
DHE-RSA-AES128-SHA yes HIGH
DHE-RSA-CAMELLIA128-SHA yes HIGH AES256-GCM-SHA384 yes HIGH AES256-CCM8 yes HIGH
AES256-CCMyes HIGH ARIA256-GCM-SHA384 yes HIGH AES128-GCM-SHA256 yes HIGH AES128-CCM8 yes HIGH
AES128-CCMyes HIGH ARIA128-GCM-SHA256 yes HIGH AES256-SHA256 yes HIGH CAMELLIA256-SHA256 yes HIGH AES128-SHA256 yes HIGH CAMELLIA128-SHA256 yes HIGH AES256-SHA yes HIGH
CAMELLIA256-SHA yes HIGH AES128-SHA yes HIGH CAMELLIA128-SHA yes HIGH
DHE-RSA-SEED-SHA yes MEDIUM SEED-SHA yes MEDIUM
TLS_AES_256_GCM_SHA384 yes HIGH TLS_CHACHA20_POLY1305_SHA256 yes HIGH TLS_AES_128_GCM_SHA256 yes HIGH

SSLv3: 0 0 0 0 0 0
TLSv1: 10 2 0 1 7 13 ECDHE-RSA-AES256-SHA
TLSv11: 10 2 0 1 7 13 ECDHE-RSA-AES256-SHA
TLSv12: 44 2 0 0 29 46 ECDHE-RSA-AES256-GCM-SHA384
TLSv13: 3 0 0 0 0 3 TLS_AES_256_GCM_SHA384

Selected Cipher: **[root@localhost O-Saft]#**

Result:

Ex. No: 9 Date:

MALWARE ANALYSIS

Aim:

Description:

Algorithm:

Yara Script:

rule spyeye : banker
{
meta:
author = "Ben"
description = "SpyEye X.Y memory" date = "2022-05-25"
version = "1.0" filetype = "memory"

strings:
$g = "bot_version"
$h = "bot_guid"

condition:
any of ($g,$h) and filesize >50000

Output:

[root@localhost Downloads]# ll malware.exe
-rw-r--r--. 1 root root 148480 May 26 11:17 malware.exe

[root@localhost Downloads]# yara spyeye.yara malware.exe spyeye malware.exe

Result:

Ex. No: 10 Date:

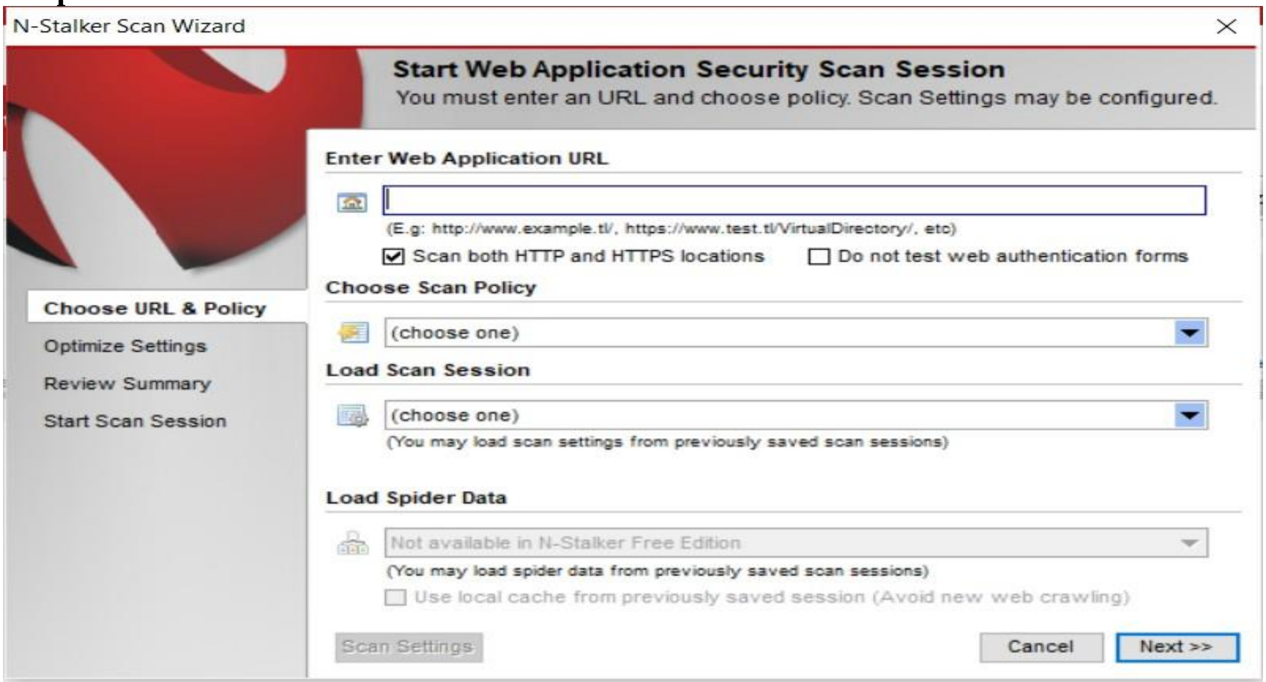
N-STALKER

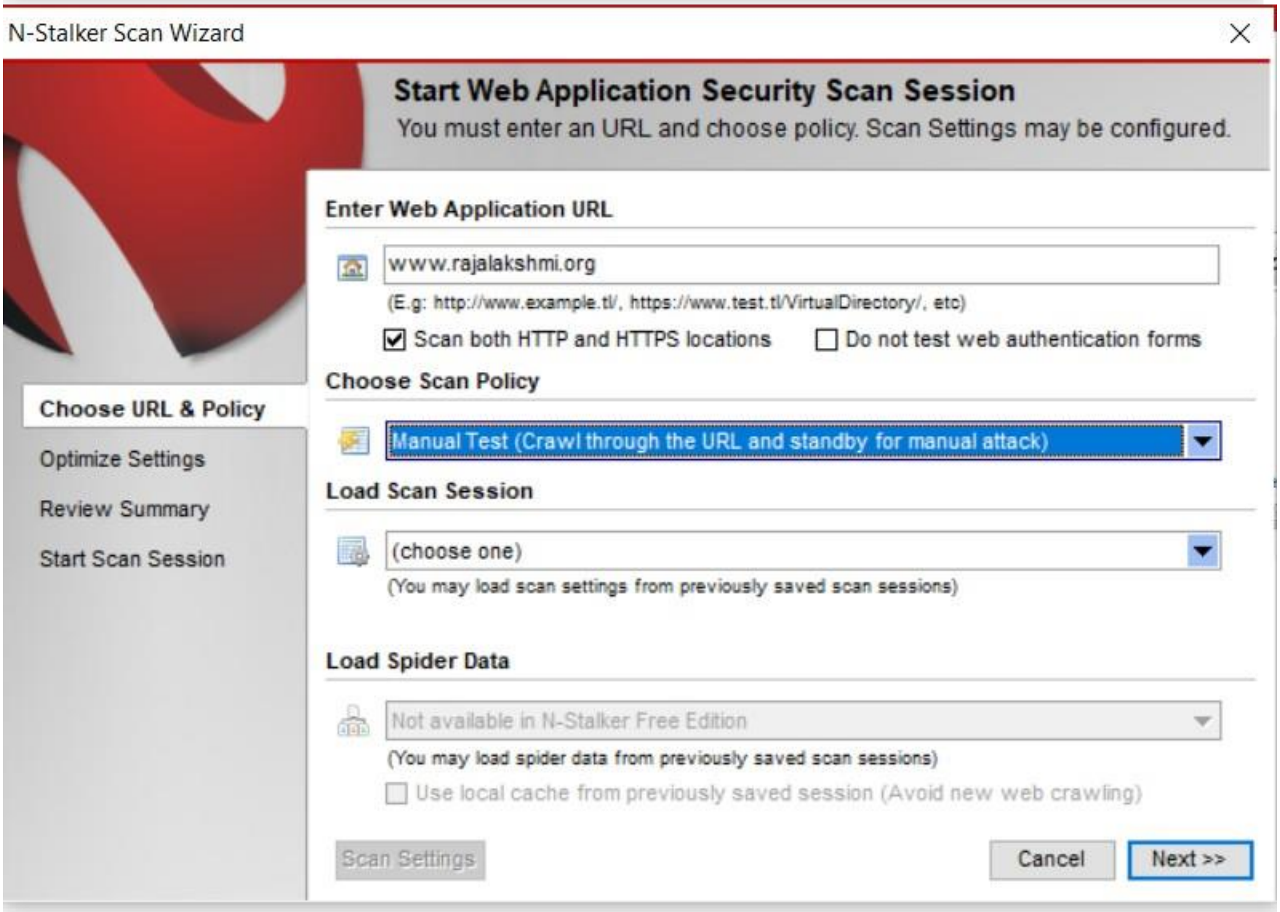
Aim:

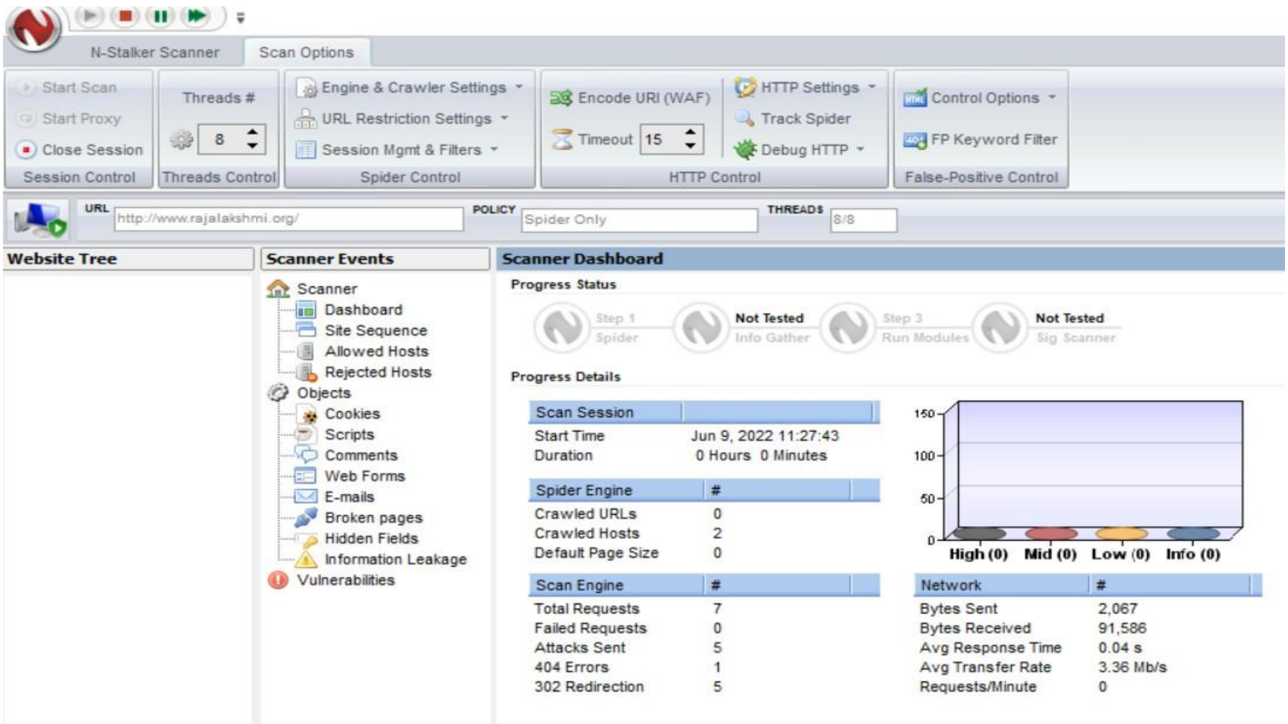
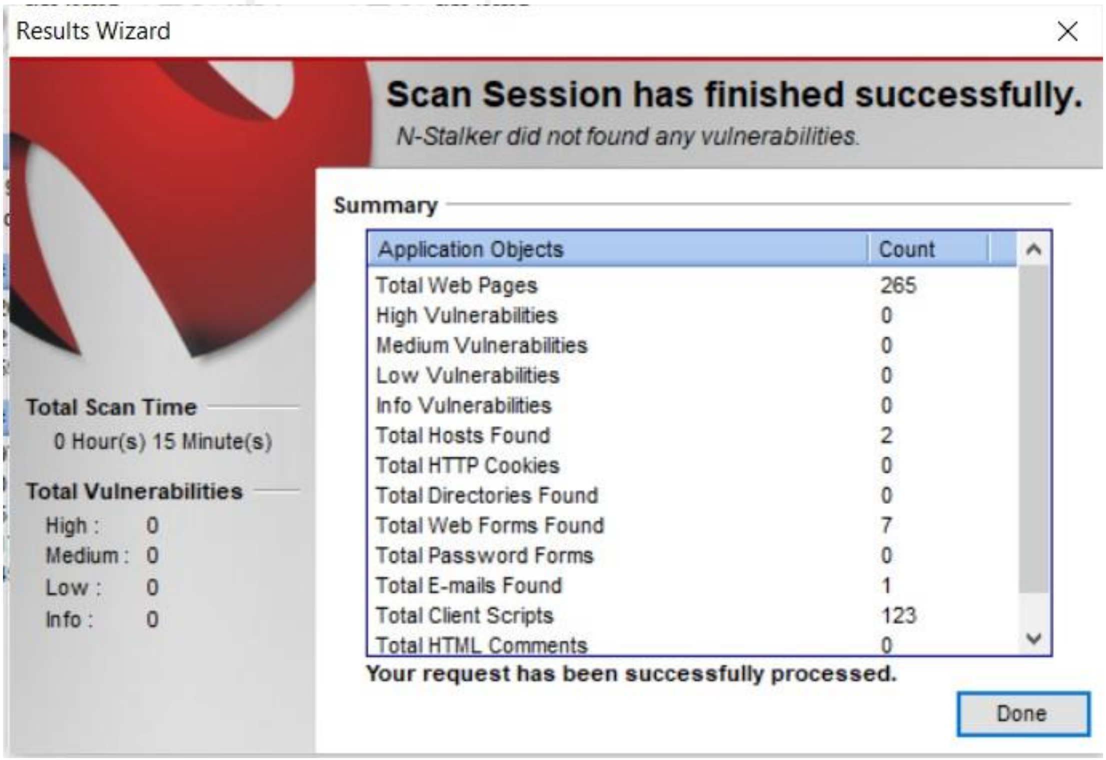
Description:

Algorithm:

Output:


N-Stalker Scan Wizard
Start Web Application Security Scan Session
You must enter an URL and choose policy. Scan Settings may be configured.
Enter Web Application URL
(E.g: http://www.example.tl/, https://www.test.tl/VirtualDirectory/, etc)
Scan both HTTP and HTTPS locations
Do not test web authentication forms
Choose Scan Policy
(choose one)
Load Scan Session
(choose one)
(You may load scan settings from previously saved scan sessions)
Load Spider Data
Not available in N-Stalker Free Edition
(You may load spider data from previously saved scan sessions)
Use local cache from previously saved session (Avoid new web crawling)
Choose URL & Policy
Optimize Settings
Review Summary
Start Scan Session
Scan Settings
Cancel
Next >>


N-Stalker Scan Wizard
Start Web Application Security Scan Session
You must enter an URL and choose policy. Scan Settings may be configured.
Enter Web Application URL
www.rajalakshmi.org
(E.g: http://www.example.tl/, https://www.test.tl/VirtualDirectory/, etc)
Scan both HTTP and HTTPS locations
Do not test web authentication forms
Choose Scan Policy
Manual Test (Crawl through the URL and standby for manual attack)
Load Scan Session
(choose one)
(You may load scan settings from previously saved scan sessions)
Load Spider Data
Not available in N-Stalker Free Edition
(You may load spider data from previously saved scan sessions)
Use local cache from previously saved session (Avoid new web crawling)
Choose URL & Policy
Optimize Settings
Review Summary
Start Scan Session
Scan Settings
Cancel
Next >>


N-Stalker Scanner
Scan Options
Start Scan
Start Proxy
Close Session
Session Control
Threads #
8
Threads Control
Engine & Crawler Settings
URL Restriction Settings
Session Mgmt & Filters
Spider Control
Encode URI (WAF)
Timeout
15
HTTP Settings
Track Spider
Debug HTTP
HTTP Control
Control Options
FP Keyword Filter
False-Positive Control
URL
http://www.rajalakshmi.org/
POLICY
Spider Only
THREADS
8/8
Website Tree
Scanner Events
Scanner Dashboard
Scanner
Dashboard
Site Sequence
Allowed Hosts
Rejected Hosts
Objects
Cookies
Scripts
Comments
Web Forms
E-mails
Broken pages
Hidden Fields
Information Leakage
Vulnerabilities
Progress Status
Step 1 Spider
Not Tested Info Gather
Step 3 Run Modules
Not Tested Sig Scanner
Progress Details
Scan Session
Start Time Jun 9, 2022 11:27:43
Duration 0 Hours 0 Minutes
Spider Engine #
Crawled URLs 0
Crawled Hosts 2
Default Page Size 0
Scan Engine #
Total Requests 7
Failed Requests 0
Attacks Sent 5
404 Errors 1
302 Redirection 5
150
100
50
0
High (0) Mid (0) Low (0) Info (0)
Network #
Bytes Sent 2,067
Bytes Received 91,586
Avg Response Time 0.04 s
Avg Transfer Rate 3.36 Mb/s
Requests/Minute 0

Results Wizard
Scan Session has finished successfully.
N-Stalker did not found any vulnerabilities.
Summary
Application Objects Count
Total Web Pages 265
High Vulnerabilities 0
Medium Vulnerabilities 0
Low Vulnerabilities 0
Info Vulnerabilities 0
Total Hosts Found 2
Total HTTP Cookies 0
Total Directories Found 0
Total Web Forms Found 7
Total Password Forms 0
Total E-mails Found 1
Total Client Scripts 123
Total HTML Comments 0
Your request has been successfully processed.
Done
Total Scan Time
0 Hour(s) 15 Minute(s)
Total Vulnerabilities
High : 0
Medium : 0
Low : 0
Info : 0

Result:

Ex. No: 11 Date:

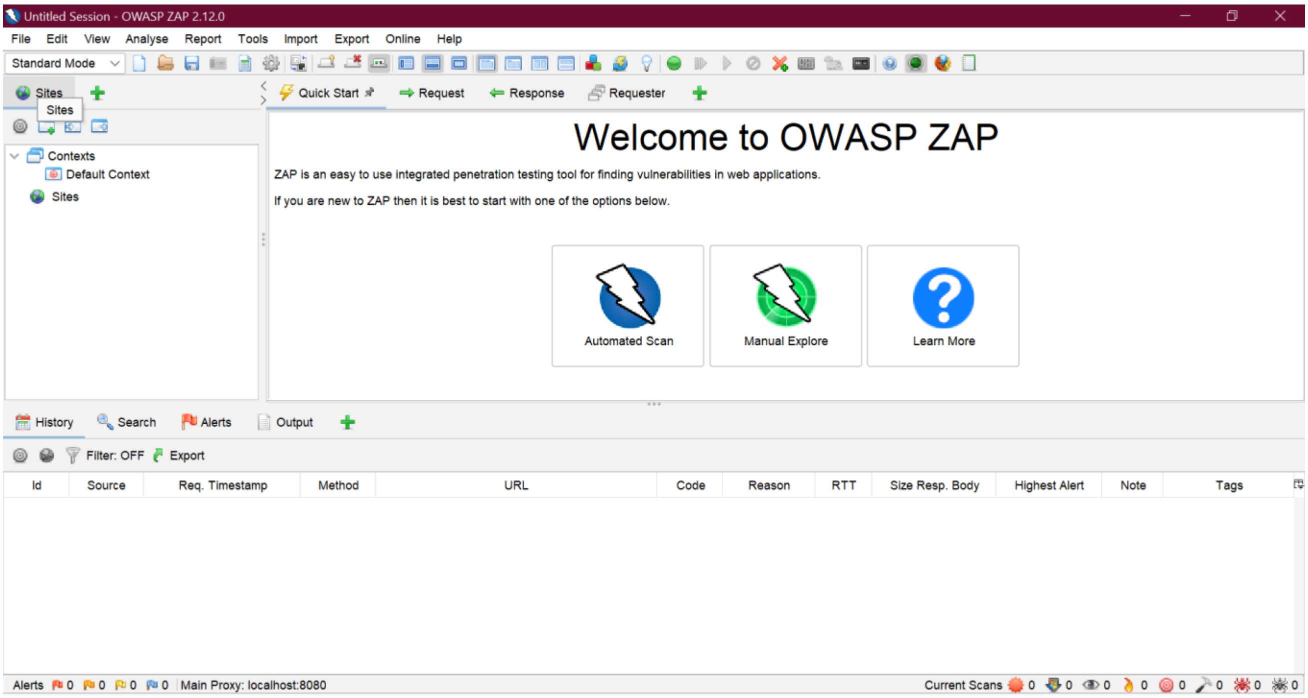
OWASP VULNERABILITY TEST

Aim:

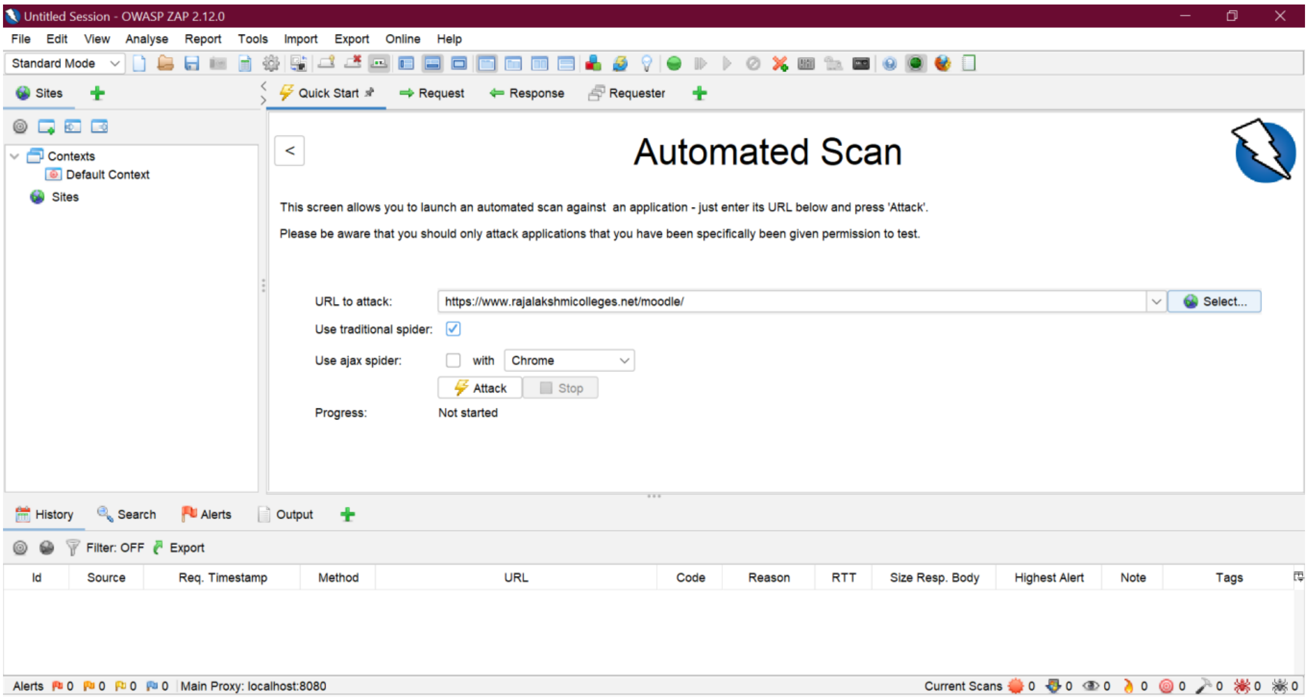
Description:

Algorithm:

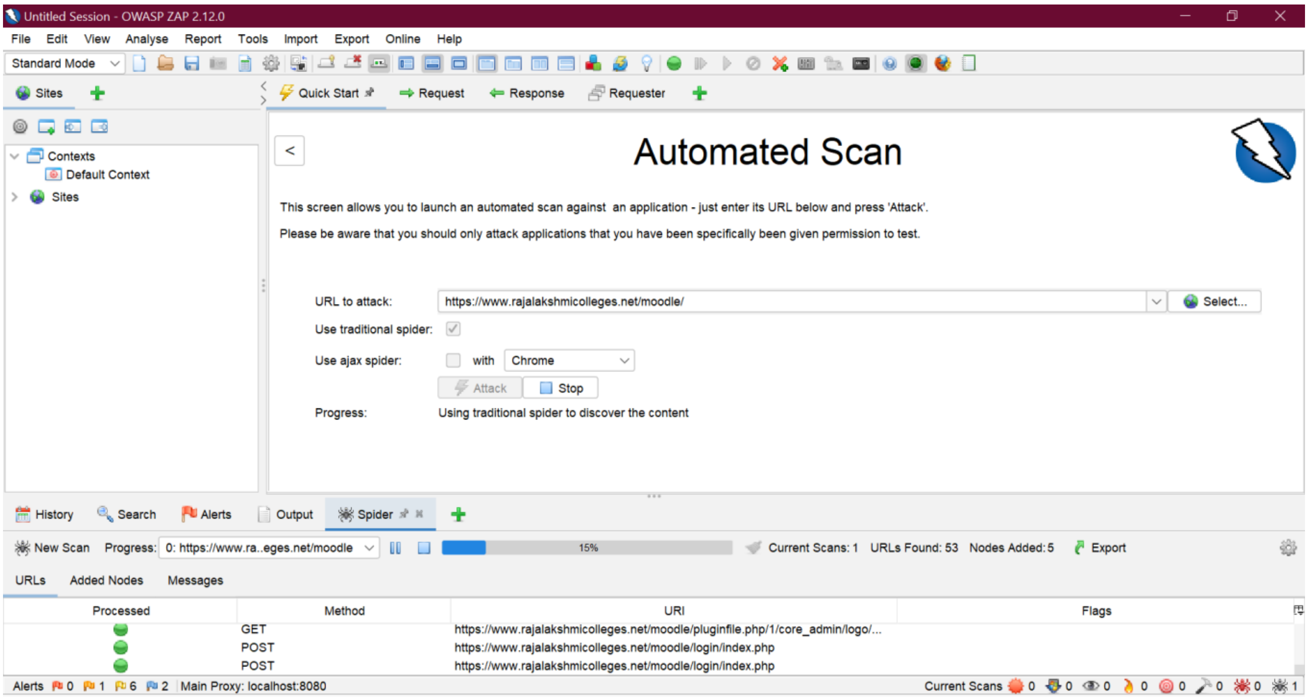
Output:

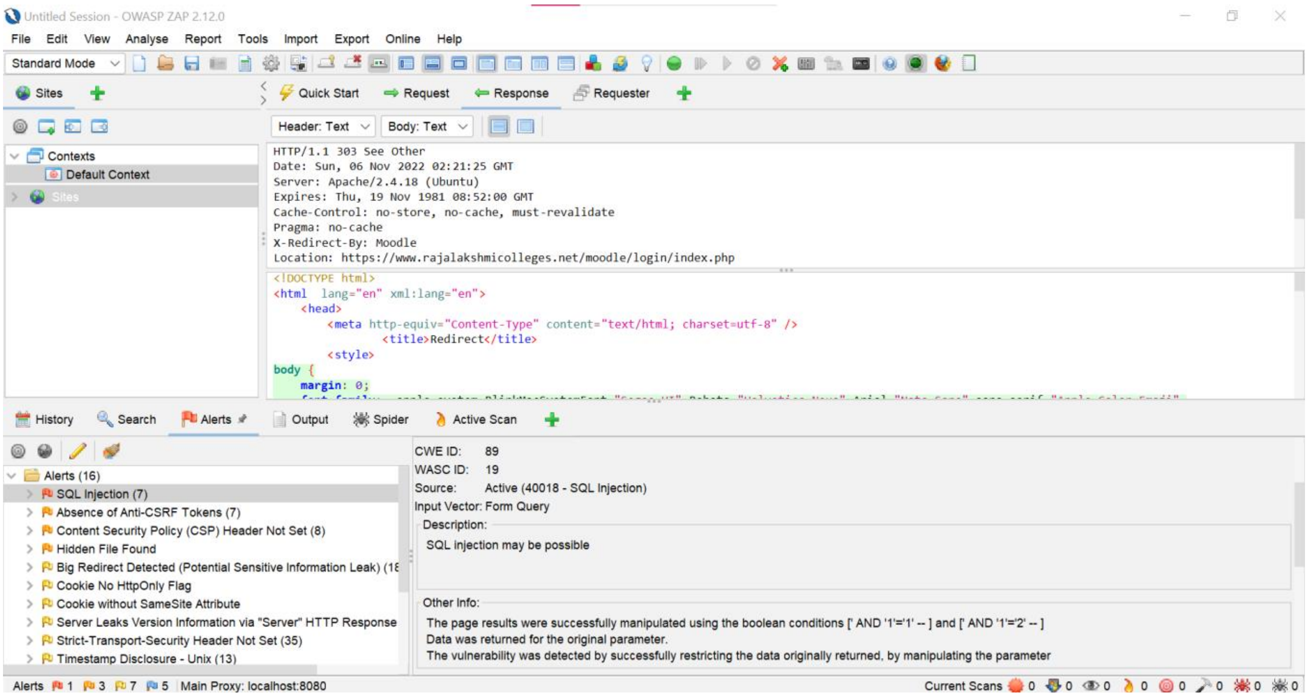


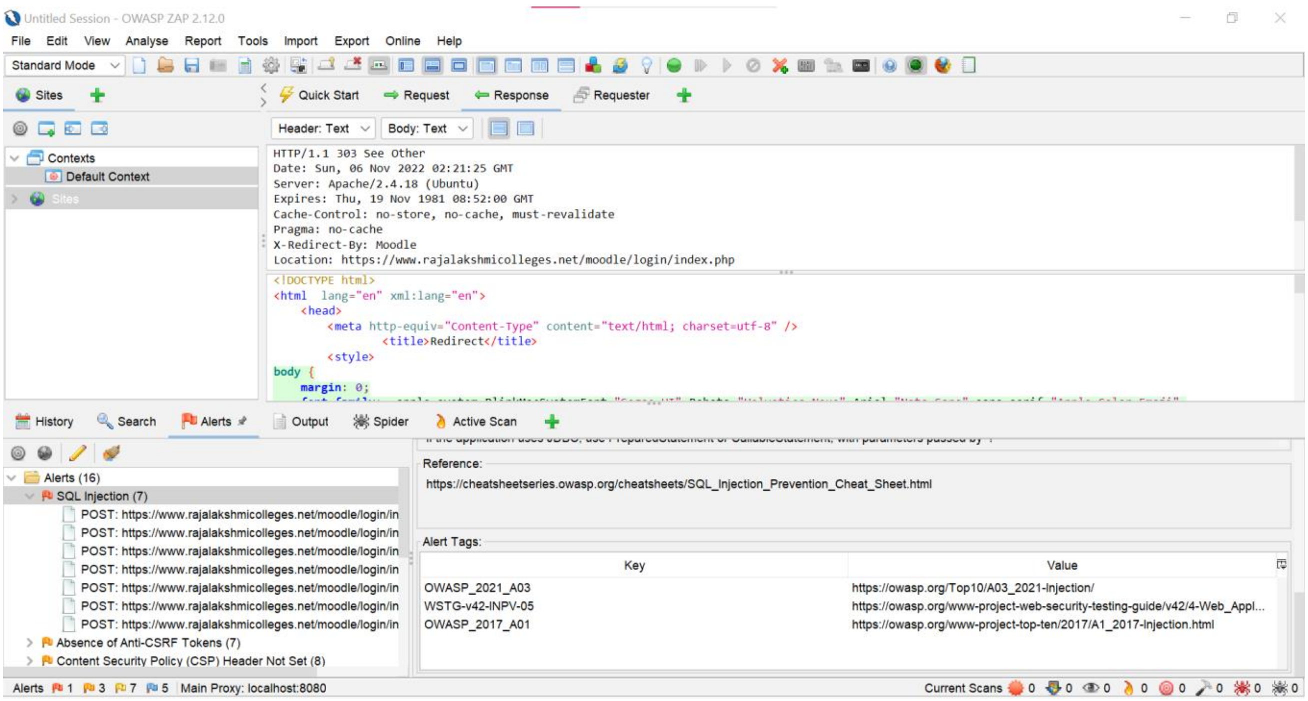
Untitled Session - OWASP ZAP 2.12.0
File Edit View Analyse Report Tools Import Export Online Help
Standard Mode
Sites
Quick Start
Request
Response
Requester
Contexts
Default Context
Sites
Welcome to OWASP ZAP
ZAP is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications.
If you are new to ZAP then it is best to start with one of the options below.
Automated Scan
Manual Explore
Learn More
History
Search
Alerts
Output
Filter: OFF
Export
Id
Source
Req. Timestamp
Method
URL
Code
Reason
RTT
Size Resp. Body
Highest Alert
Note
Tags
Main Proxy: localhost:8080
Current Scans

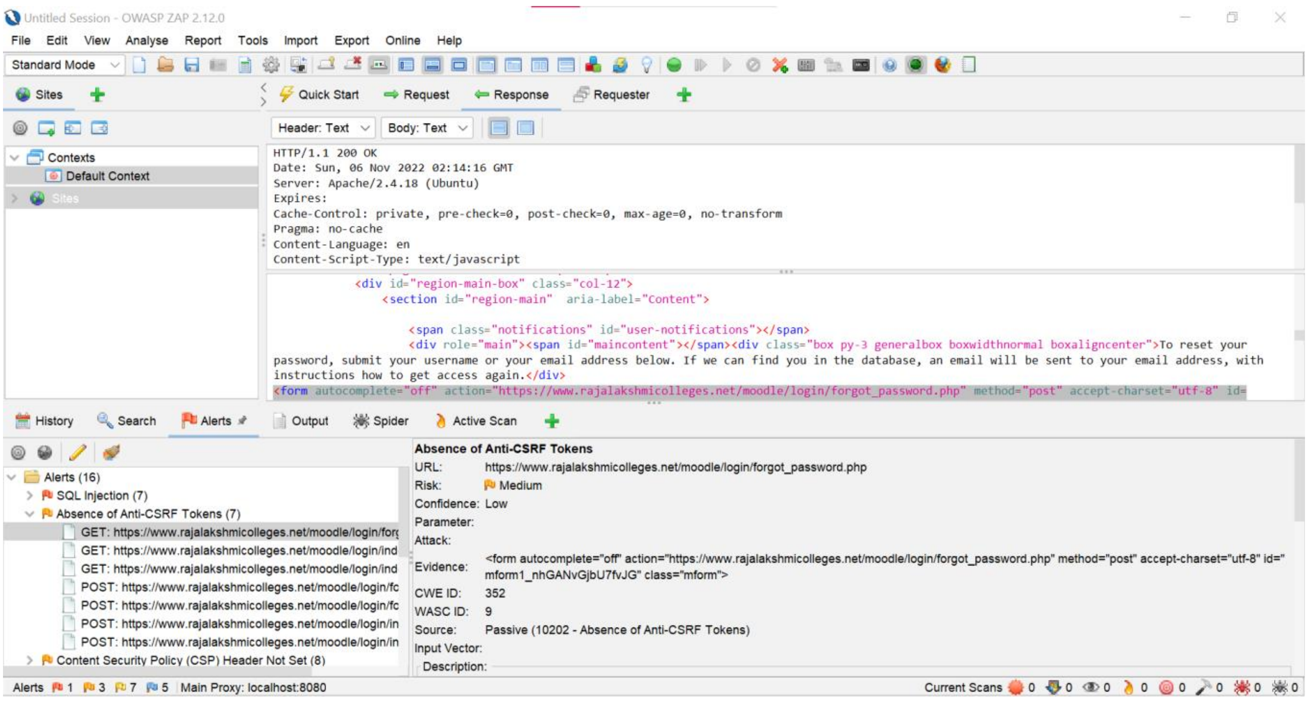


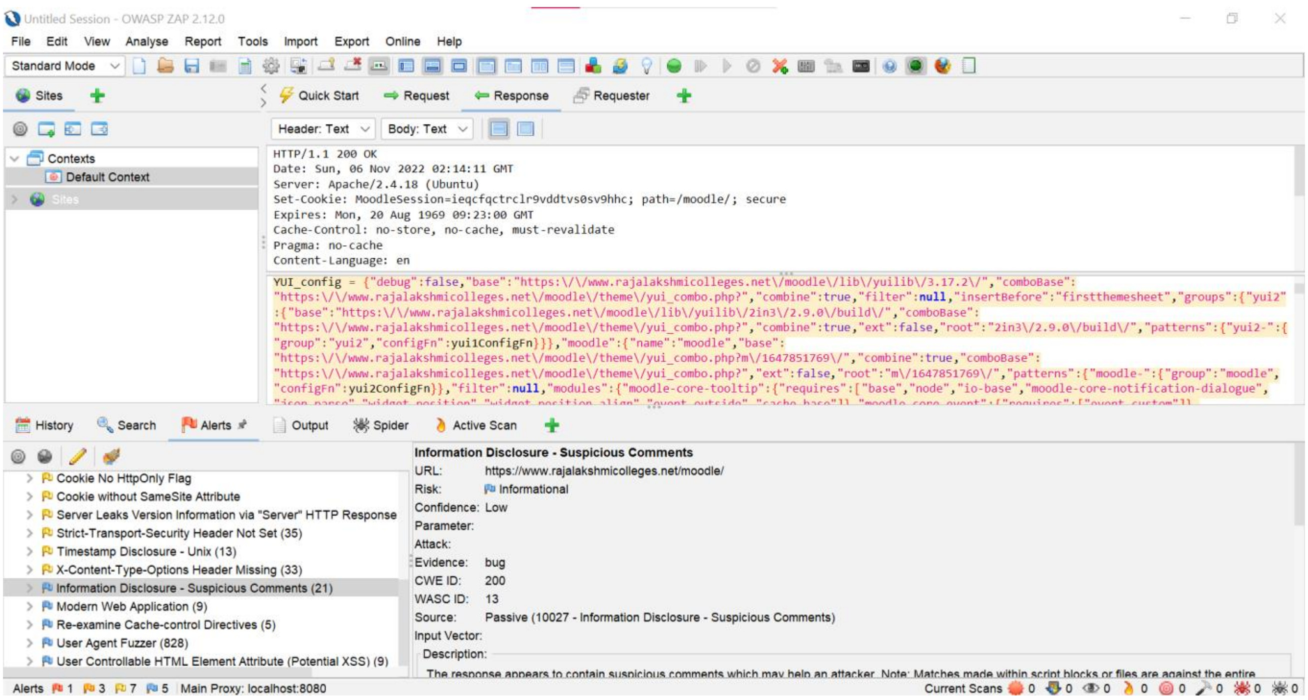
Untitled Session - OWASP ZAP 2.12.0
File Edit View Analyse Report Tools Import Export Online Help
Automated Scan
This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'.
Please be aware that you should only attack applications that you have been specifically been given permission to test.
URL to attack:
https://www.rajalakshmicolleges.net/moodle/
Select...
Use traditional spider:
Use ajax spider:
with
Chrome
Attack
Stop
Progress:
Not started
Main Proxy: localhost:8080
Current Scans

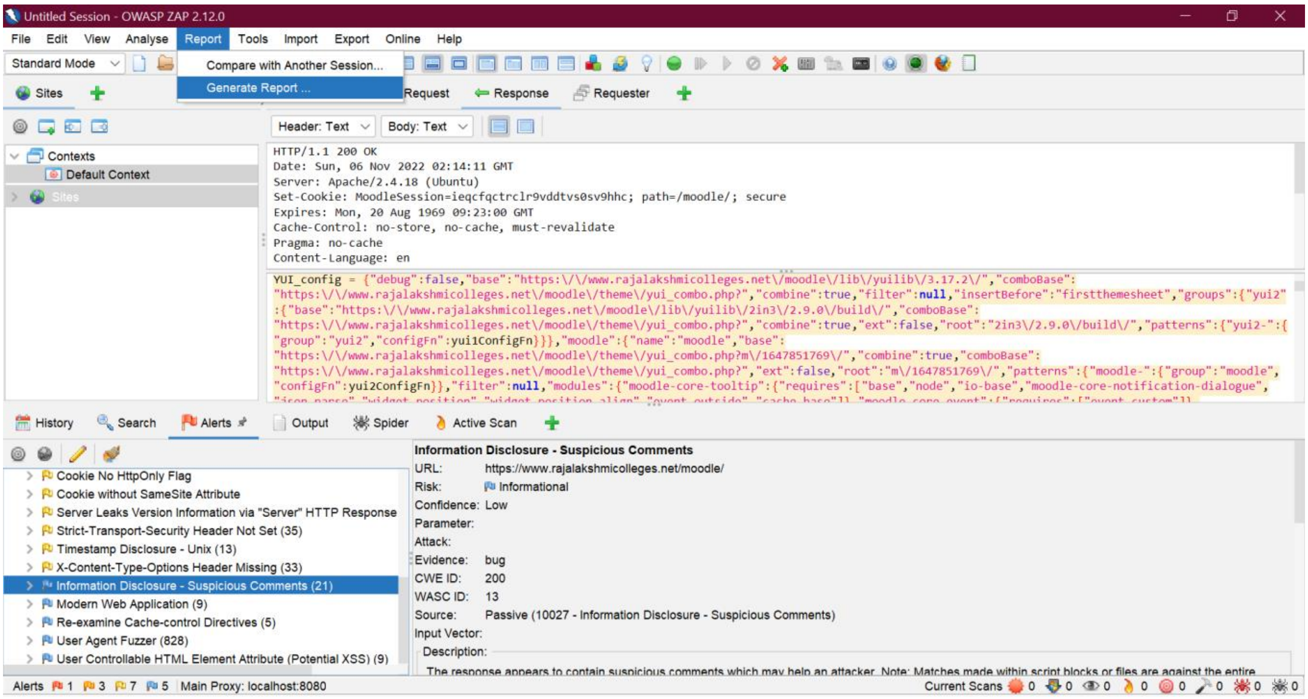


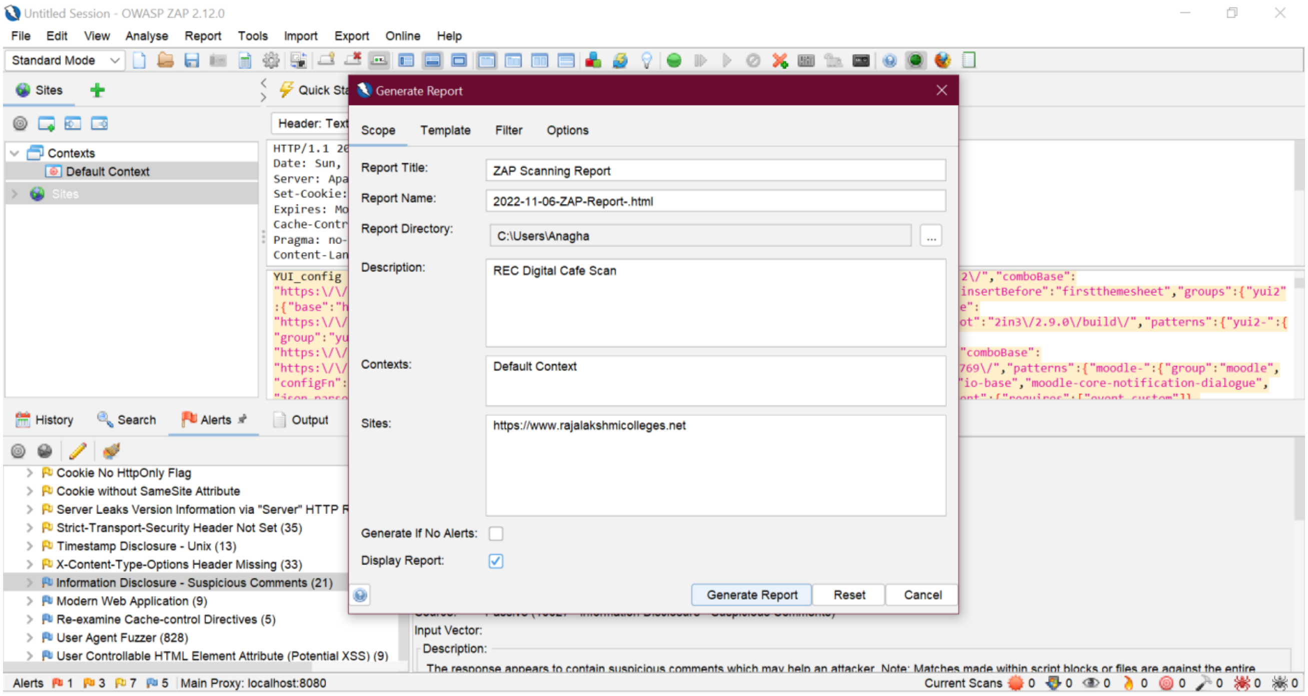


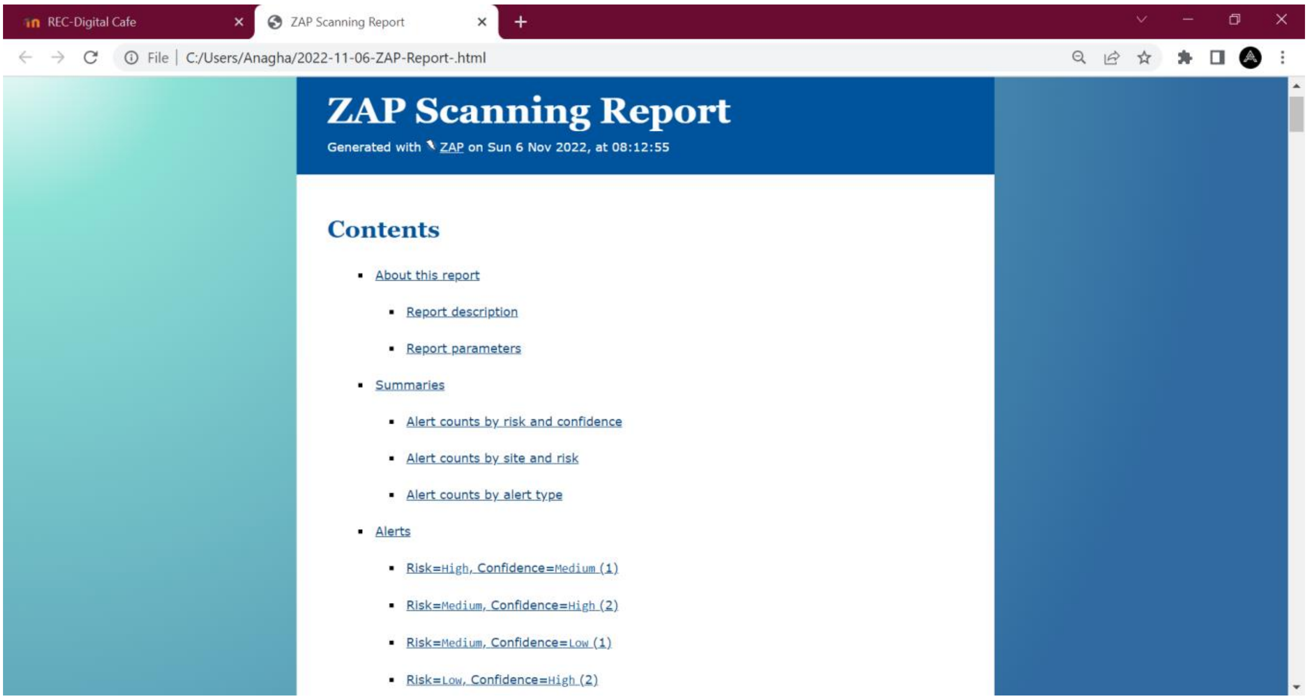


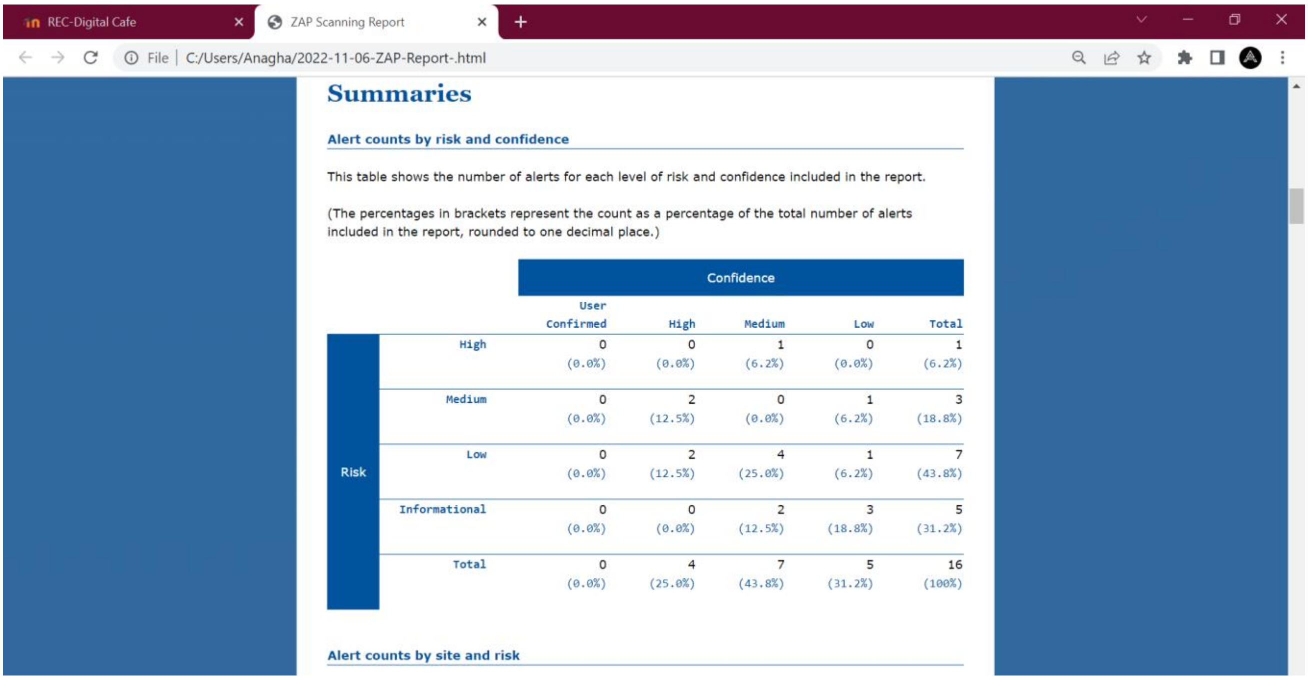












Summaries

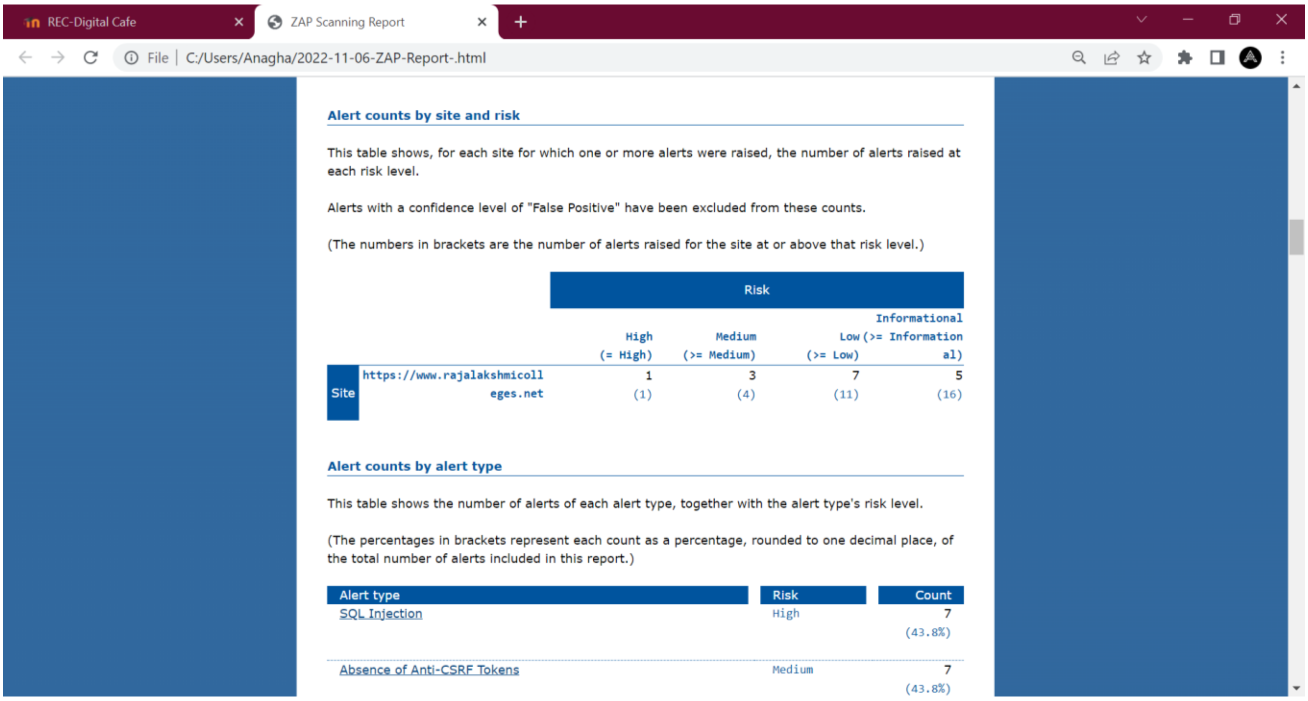
Alert counts by risk and confidence

This table shows the number of alerts for each level of risk and confidence included in the report.

(The percentages in brackets represent the count as a percentage of the total number of alerts included in the report, rounded to one decimal place.)

		Confidence				
		User Confirmed	High	Medium	Low	Total
Risk	High	0 (0.0%)	0 (0.0%)	1 (6.2%)	0 (0.0%)	1 (6.2%)
	Medium	0 (0.0%)	2 (12.5%)	0 (0.0%)	1 (6.2%)	3 (18.8%)
	Low	0 (0.0%)	2 (12.5%)	4 (25.0%)	1 (6.2%)	7 (43.8%)
	Informational	0 (0.0%)	0 (0.0%)	2 (12.5%)	3 (18.8%)	5 (31.2%)
	Total	0 (0.0%)	4 (25.0%)	7 (43.8%)	5 (31.2%)	16 (100%)

Alert counts by site and risk



Alert counts by site and risk

This table shows, for each site for which one or more alerts were raised, the number of alerts raised at each risk level.

Alerts with a confidence level of "False Positive" have been excluded from these counts.

(The numbers in brackets are the number of alerts raised for the site at or above that risk level.)

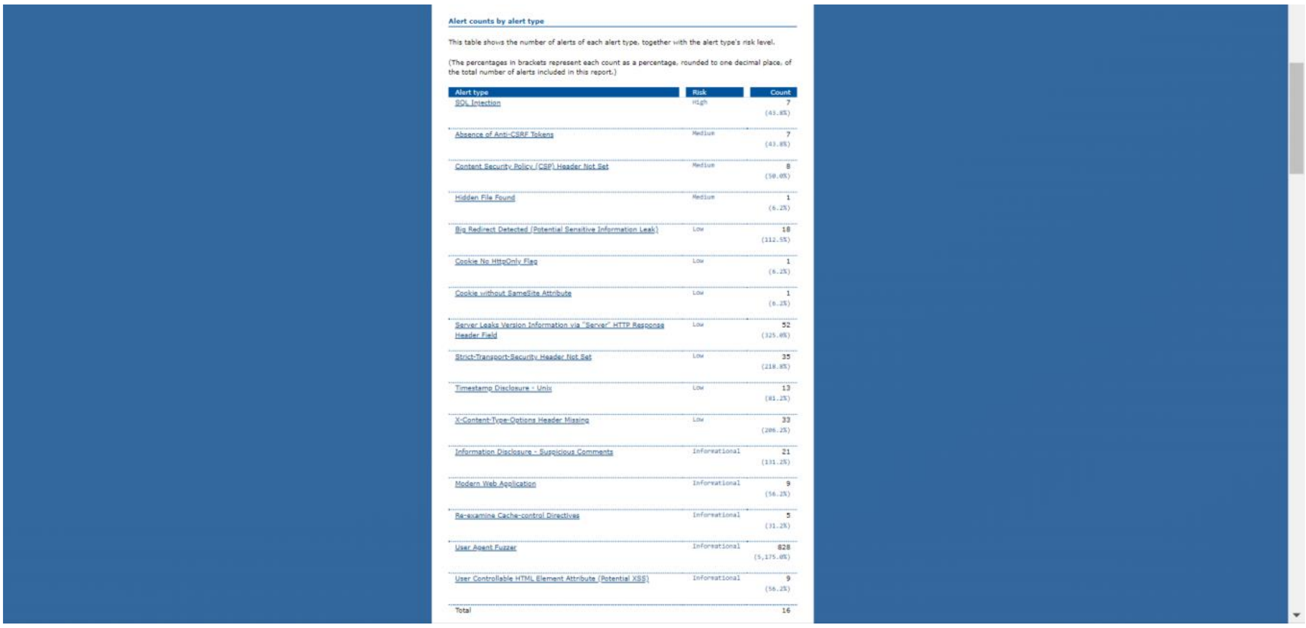
		Risk			
		High (= High)	Medium (>= Medium)	Low (>= Low)	Informational (>= Informational)
Site	https://www.rajalakshmicolleges.net	1 (1)	3 (4)	7 (11)	5 (16)

Alert counts by alert type

This table shows the number of alerts of each alert type, together with the alert type's risk level.

(The percentages in brackets represent each count as a percentage, rounded to one decimal place, of the total number of alerts included in this report.)

Alert type	Risk	Count
SQL Injection	High	7 (43.8%)
Absence of Anti-CSRF Tokens	Medium	7 (43.8%)

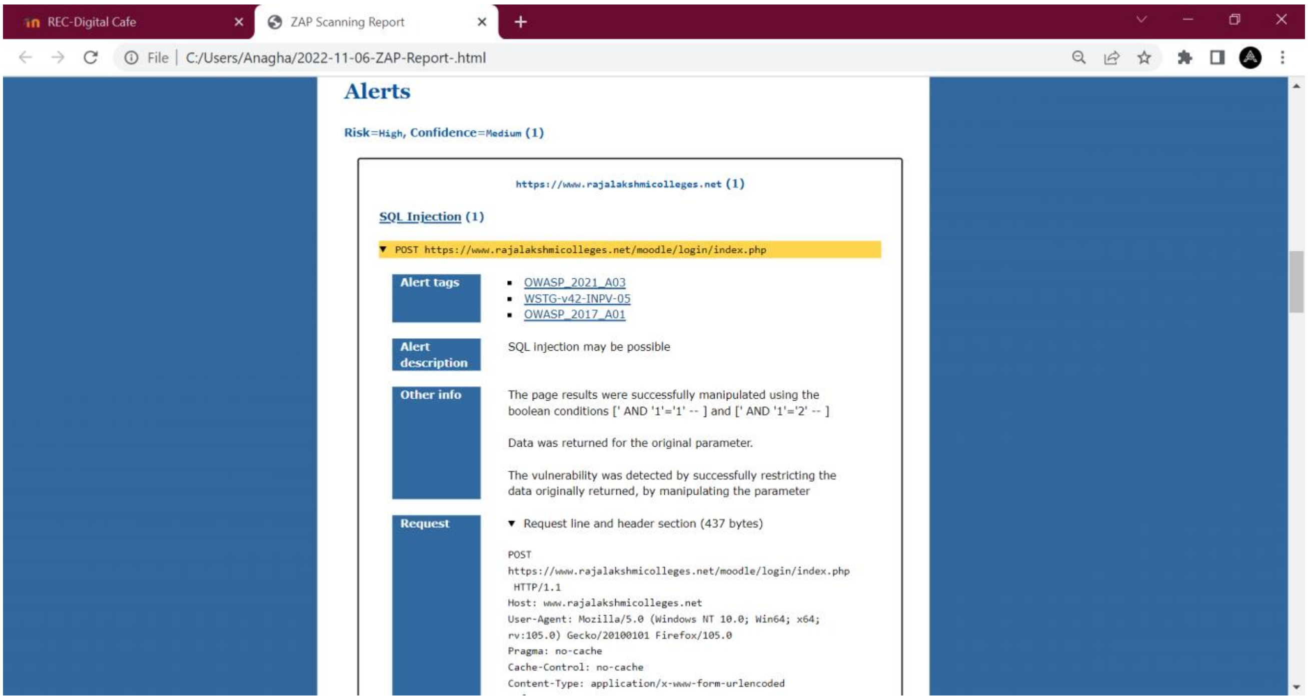


Alert counts by alert type

This table shows the number of alerts of each alert type, together with the alert type's risk level.

(The percentages in brackets represent each count as a percentage, rounded to one decimal place, of the total number of alerts included in this report.)

Alert type	Risk	Count
SQL Injection	High	7 (43.8%)
Absence of Anti-CSRF Tokens	Medium	7 (43.8%)
Content Security Policy (CSP) Header Not Set	Medium	8 (50.0%)
Hidden File Found	Medium	1 (6.2%)
Big Redirect Detected (Potential Sensitive Information Leak)	Low	18 (112.5%)
Cookie No HttpOnly Flag	Low	1 (6.2%)
Cookie without SameSite Attribute	Low	1 (6.2%)
Server Leaks Version Information via "Server" HTTP Response Header Field	Low	52 (325.0%)
Strict-Transport-Security Header Not Set	Low	35 (218.8%)
Timestamp Disclosure - Unix	Low	13 (81.2%)
X-Content-Type-Options Header Missing	Low	33 (206.2%)
Information Disclosure - Suspicious Comments	Informational	21 (131.2%)
Modern Web Application	Informational	9 (56.2%)
Re-examine Cache-control Directives	Informational	5 (31.2%)
User Agent Fuzzer	Informational	828 (5,175.0%)
User Controllable HTML Element Attribute (Potential XSS)	Informational	9 (56.2%)
Total		16



REC-Digital Cafe | ZAP Scanning Report

File | C:/Users/Anagha/2022-11-06-ZAP-Report-.html

Alerts

Risk=High, Confidence=Medium (1)

https://www.rajalakshmicolleges.net (1)

SQL Injection (1)

▼ POST https://www.rajalakshmicolleges.net/moodle/login/index.php

Alert tags	OWASP_2021_A03 WSTG-v42-INPV-05 OWASP_2017_A01
Alert description	SQL injection may be possible
Other info	The page results were successfully manipulated using the boolean conditions [' AND '1'='1' --] and [' AND '1'='2' --] Data was returned for the original parameter. The vulnerability was detected by successfully restricting the data originally returned, by manipulating the parameter
Request	▼ Request line and header section (437 bytes)

```
POST
https://www.rajalakshmicolleges.net/moodle/login/index.php
 HTTP/1.1
Host: www.rajalakshmicolleges.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64;
rv:105.0) Gecko/20100101 Firefox/105.0
Pragma: no-cache
Cache-Control: no-cache
Content-Type: application/x-www-form-urlencoded
```

Result:

Ex. No: 11 Date:

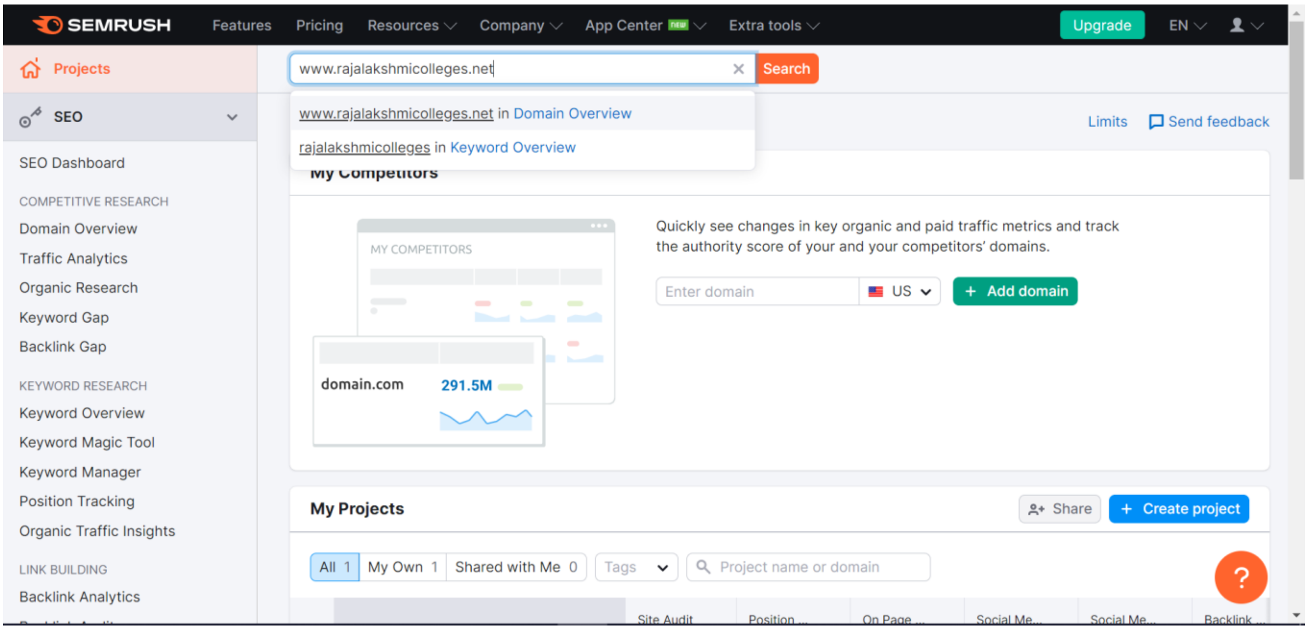
WEBSITE AUDIT

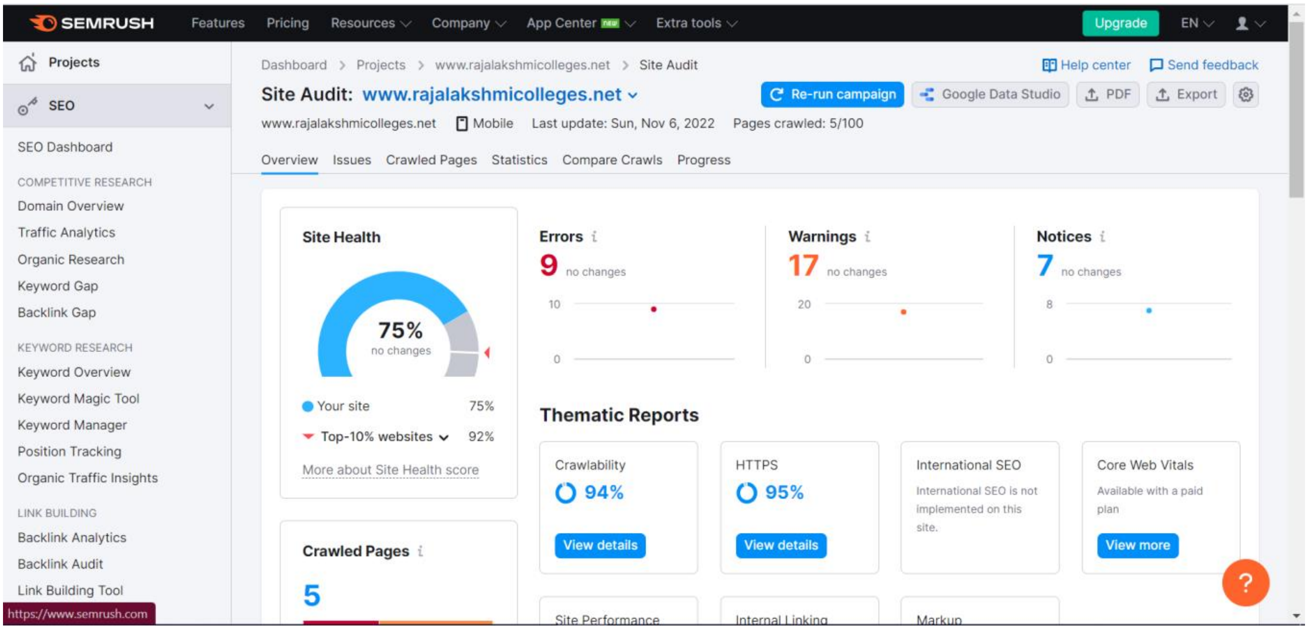
Aim:

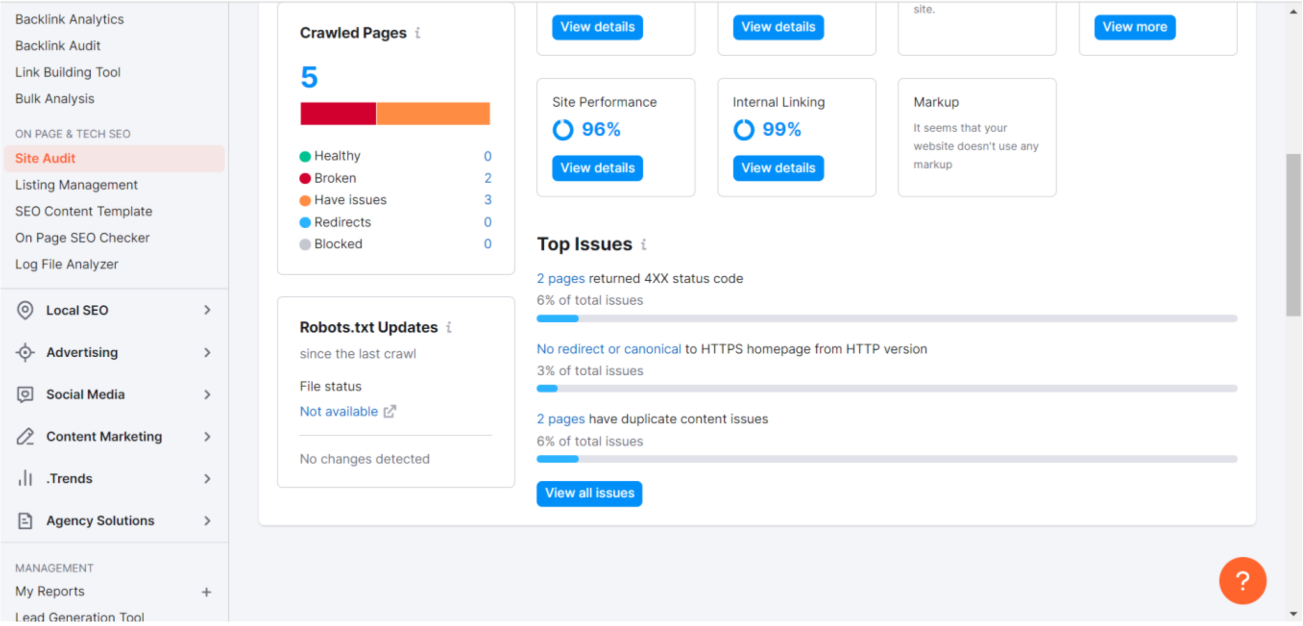
Description:

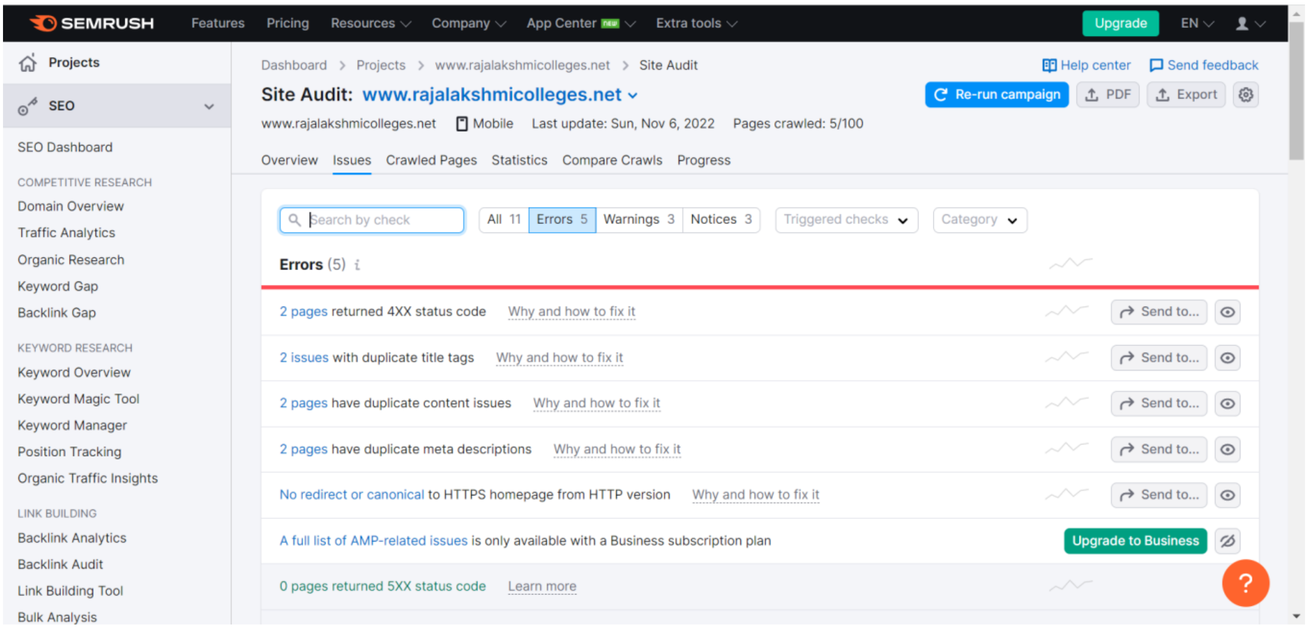
Algorithm:

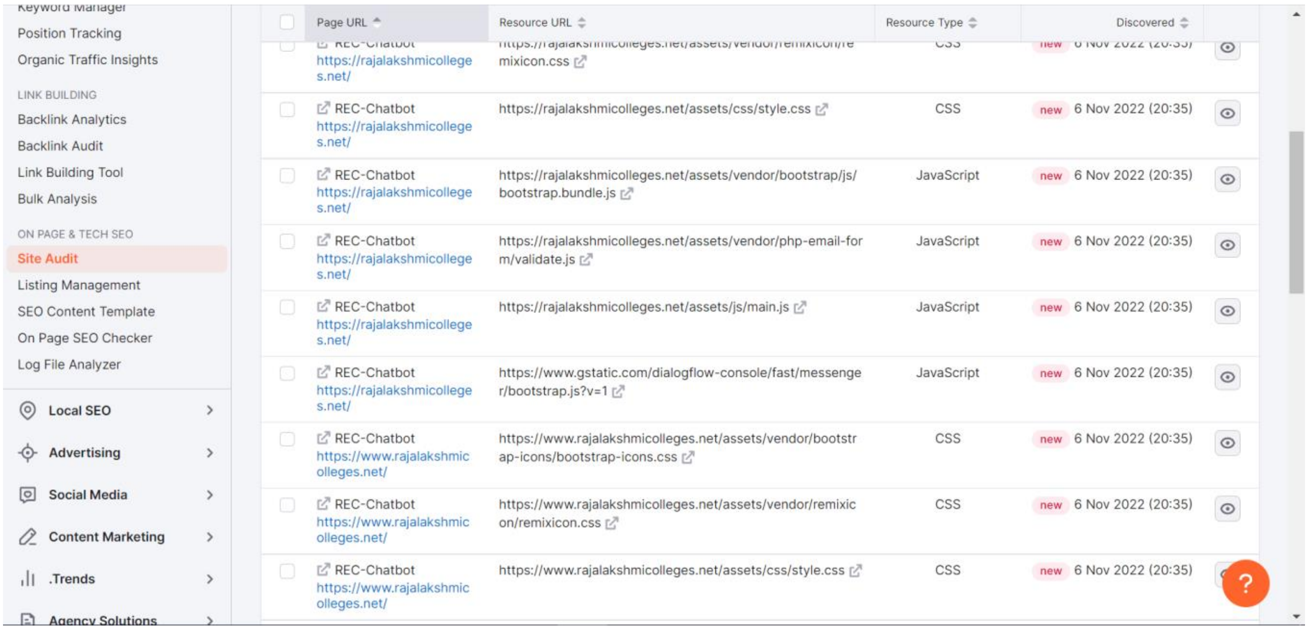
Output:

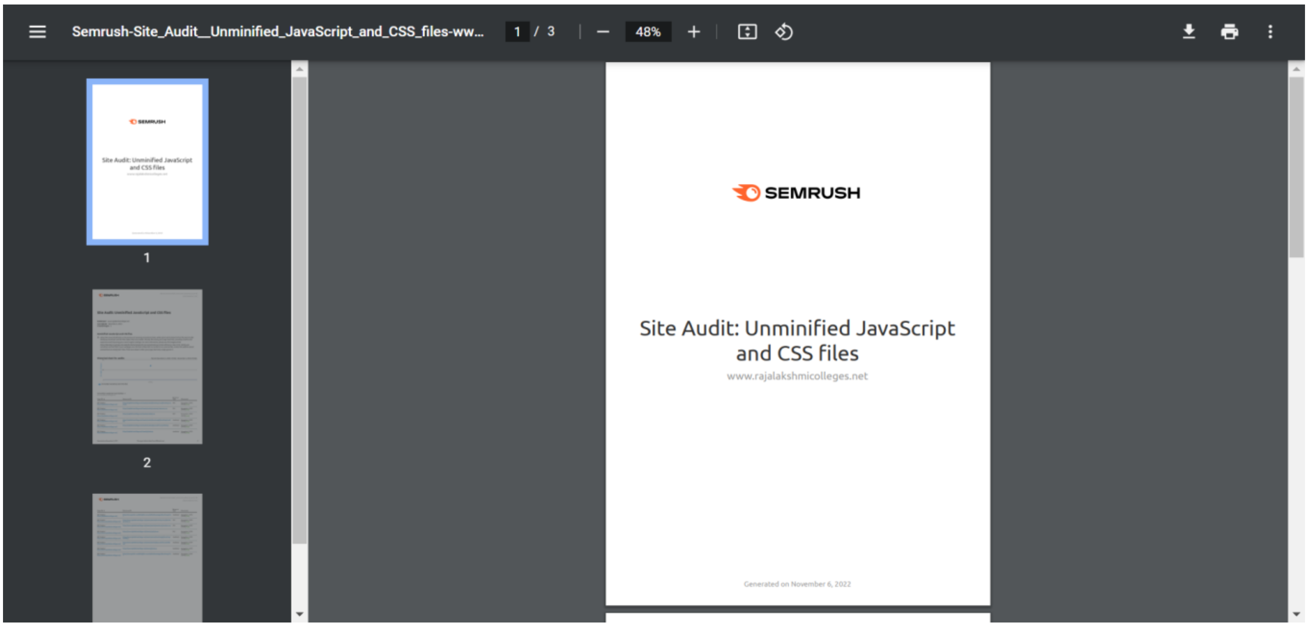

SEMRUSH
Features
Pricing
Resources
Company
App Center
Extra tools
Upgrade
EN
Projects
www.rajalakshmicolleges.net
Search
www.rajalakshmicolleges.net in Domain Overview
rajalakshmicolleges in Keyword Overview
SEO
Limits
Send feedback
SEO Dashboard
COMPETITIVE RESEARCH
Domain Overview
Traffic Analytics
Organic Research
Keyword Gap
Backlink Gap
KEYWORD RESEARCH
Keyword Overview
Keyword Magic Tool
Keyword Manager
Position Tracking
Organic Traffic Insights
LINK BUILDING
Backlink Analytics
MY COMPETITORS
domain.com
291.5M
Quickly see changes in key organic and paid traffic metrics and track the authority score of your and your competitors' domains.
Enter domain
US
+ Add domain
My Projects
Share
+ Create project
All 1
My Own 1
Shared with Me 0
Tags
Project name or domain


SEMRUSH
Features
Pricing
Resources
Company
App Center
Extra tools
Upgrade
EN
Projects
SEO
Dashboard > Projects > www.rajalakshmicolleges.net > Site Audit
Help center
Send feedback
Site Audit: www.rajalakshmicolleges.net
Re-run campaign
Google Data Studio
PDF
Export
www.rajalakshmicolleges.net
Mobile
Last update: Sun, Nov 6, 2022
Pages crawled: 5/100
Overview
Issues
Crawled Pages
Statistics
Compare Crawls
Progress
SEO Dashboard
COMPETITIVE RESEARCH
Domain Overview
Traffic Analytics
Organic Research
Keyword Gap
Backlink Gap
KEYWORD RESEARCH
Keyword Overview
Keyword Magic Tool
Keyword Manager
Position Tracking
Organic Traffic Insights
LINK BUILDING
Backlink Analytics
Backlink Audit
Link Building Tool
https://www.semrush.com
Site Health
75%
no changes
Your site 75%
Top-10% websites 92%
More about Site Health score
Crawled Pages
5
Errors
9 no changes
Warnings
17 no changes
Notices
7 no changes
Thematic Reports
Crawlability
94%
View details
HTTPS
95%
View details
International SEO
International SEO is not implemented on this site.
Core Web Vitals
Available with a paid plan
View more
Site Performance
Internal Linking
Markup


Backlink Analytics
Backlink Audit
Link Building Tool
Bulk Analysis
ON PAGE & TECH SEO
Site Audit
Listing Management
SEO Content Template
On Page SEO Checker
Log File Analyzer
Local SEO
Advertising
Social Media
Content Marketing
.Trends
Agency Solutions
MANAGEMENT
My Reports
Lead Generation Tool
Crawled Pages
5
Healthy 0
Broken 2
Have issues 3
Redirects 0
Blocked 0
Robots.txt Updates
since the last crawl
File status
Not available
No changes detected
View details
View details
site.
View more
Site Performance
96%
View details
Internal Linking
99%
View details
Markup
It seems that your website doesn't use any markup
Top Issues
2 pages returned 4XX status code
6% of total issues
No redirect or canonical to HTTPS homepage from HTTP version
3% of total issues
2 pages have duplicate content issues
6% of total issues
View all issues


SEMRUSH
Features
Pricing
Resources
Company
App Center
Extra tools
Upgrade
EN
Projects
SEO
SEO Dashboard
COMPETITIVE RESEARCH
Domain Overview
Traffic Analytics
Organic Research
Keyword Gap
Backlink Gap
KEYWORD RESEARCH
Keyword Overview
Keyword Magic Tool
Keyword Manager
Position Tracking
Organic Traffic Insights
LINK BUILDING
Backlink Analytics
Backlink Audit
Link Building Tool
Bulk Analysis
Dashboard > Projects > www.rajalakshmicolleges.net > Site Audit
Help center
Send feedback
Site Audit: www.rajalakshmicolleges.net
Re-run campaign
PDF
Export
www.rajalakshmicolleges.net Mobile Last update: Sun, Nov 6, 2022 Pages crawled: 5/100
Overview Issues Crawled Pages Statistics Compare Crawls Progress
Search by check
All 11
Errors 5
Warnings 3
Notices 3
Triggered checks
Category
Errors (5)
2 pages returned 4XX status code Why and how to fix it
2 issues with duplicate title tags Why and how to fix it
2 pages have duplicate content issues Why and how to fix it
2 pages have duplicate meta descriptions Why and how to fix it
No redirect or canonical to HTTPS homepage from HTTP version Why and how to fix it
A full list of AMP-related issues is only available with a Business subscription plan
Upgrade to Business
0 pages returned 5XX status code Learn more
Send to...





Result: