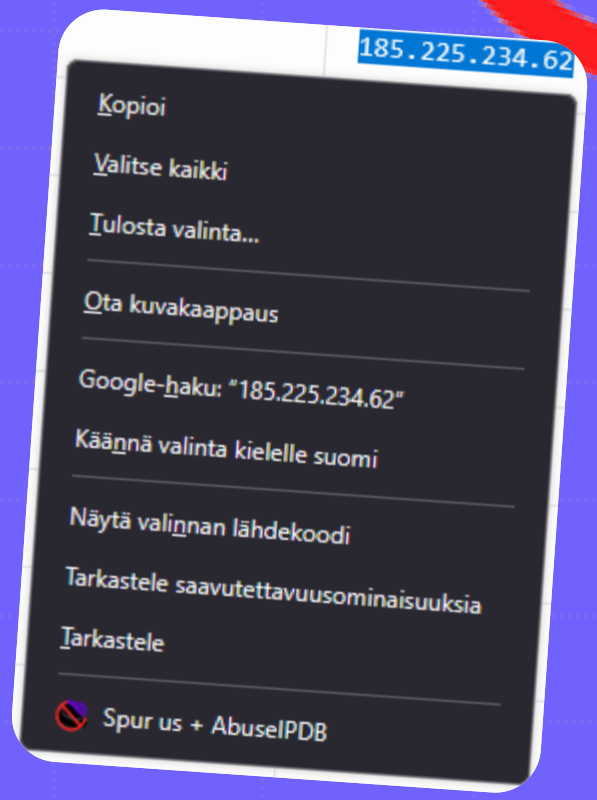
COMP.SEC.300-2024-2025-1 Secure Programming

Nea Parjanen

Spur + ABUSEIPDB IP Address Check Mozilla Extension

Checks simultaneously if the IP Address is a VPN and which country it is from using two IP checker websites



What is this used for?

- In cyber security related work

Why it should be secure?

- API key management
- Extensions used in attacks
- Every code needs to be secure

www.abuseipdb.com/check/82.149.81.31

AbuseIPDB

Home | Report IP | Bulk Reporter | Pricing | About | FA

AbuseIPDB » *82.149.81.31*

Check an IP Address, Domain Name, or Subnet
e.g. **85.156.59.100**, **microsoft.com**, or **5.188.10.0/24**

82.149.81.31 was not found in our database

ISP	Packethub S.A.
Usage Type	Data Center/Web Hosting/Transit
ASN	Unknown
Domain Name	packethub.net
Country	Italy
City	Milan, Lombardy

IP info including ISP, Usage Type, and Location provided by IPInfo. Updated biweekly.

REPORT 82.149.81.31 | WHOIS 82.149.81.31

IP Abuse Reports for **82.149.81.31**:

This IP address has not been reported. File Report

spur.us/context/82.149.81.31

SPUR

Products | Resources | About | Contact | Sign in | Sign up

82.149.81.31 - Nord VPN

82.149.81.31 belongs to the **Nord VPN** anonymization network. Nord VPN users route traffic through 82.149.81.31 to obscure their traffic from ISPs and mask their identity from servers on the internet.

Unlock The Full Data >>>

Few Devices Online

This IP address (82.149.81.31) is not being used by many devices. The low device count for 82.149.81.31 indicates that it may be privately allocated to specific customers. Datacenter IPs like 82.149.81.31 can be used in fake user or bot activity which is not accounted for in this metric. IP Addresses that have fewer unique users are generally less effective at anonymizing activity online.

DATACENTER

This IP is owned by PacketHub S.A. and is hosted in Monte Carlo, MC. This IP belongs to DATACENTER infrastructure. Datacenter IPs like 82.149.81.31 typically route user traffic when they are acting as a VPN, Proxy, Cloud Gateway or are performing some automated activity.

NVIDIA

Galleria

Nauhoita 00:00:00
Alt+F9 - Pysäytä ja tallenna

Pikatoisto
Alt+Shift+F10 - Ota käyttöön/pois käytöstä

Kuvakaappaus
Alt+F1 - Tallenna kuvakaappaus

Valokuvatila
Alt+F2 - Aloita

Kohokohdat
Kaappaa parhaat pelihetkesi automaattisesti

Pelisuodatin
None - Muokkaa suodattimia

Mikrofoni Aina käytössä
Ctrl+Alt+M - Ota käyttöön/pois käytöstä

Tilastot
Alt+R - Vaihda tilastojen päällekkäistaso päällä/pois

osoite

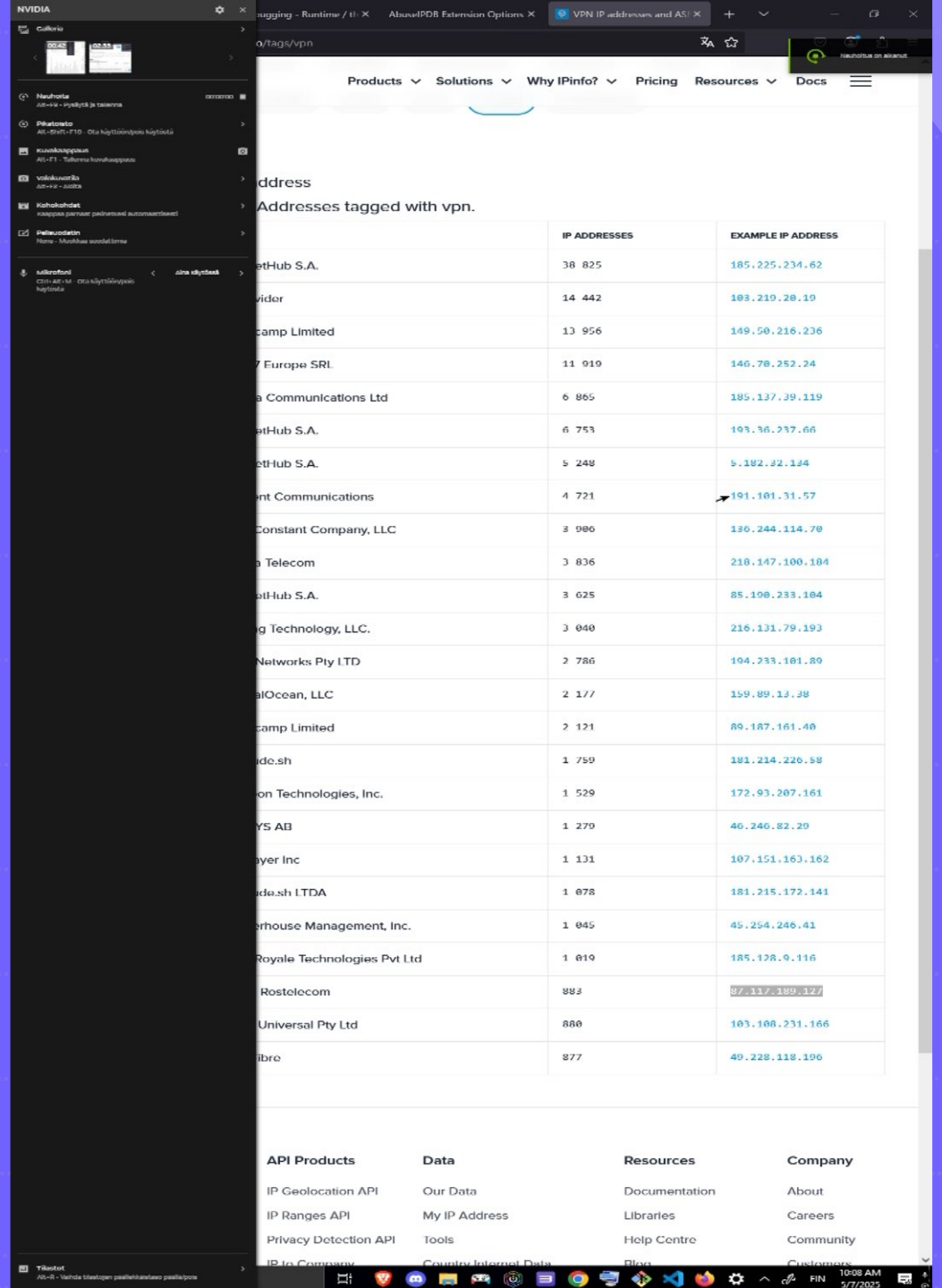
Nauhoitus on alkanut

Firefox

Hae hakukoneella Google tai kirjoita osoite

VPN IP dresses an... | 216.131.79.102 (Whitelabel VP... | 45.80.28.52 | Packethub S.A.... | Wikipedia | YouTube | Reddit | Add-ons for Firefox | Lisää pikavalinta

FIN 10:00 AM 5/7/2025


NVIDIA
Galleria
Products
Solutions
Why IPinfo?
Pricing
Resources
Docs
ddress
Addresses tagged with vpn.
IP ADDRESSES
EXAMPLE IP ADDRESS
etHub S.A. 38 825 185.225.234.62
vider 14 442 103.219.20.19
camp Limited 13 956 149.50.216.236
7 Europe SRL 11 919 146.70.252.24
a Communications Ltd 6 865 185.137.39.119
etHub S.A. 6 753 193.36.237.66
etHub S.A. 5 248 5.182.32.134
nt Communications 4 721 191.101.31.57
Constant Company, LLC 3 986 136.244.114.70
Telecom 3 836 218.147.100.184
etHub S.A. 3 625 85.190.233.104
g Technology, LLC. 3 040 216.131.79.193
Networks Pty LTD 2 786 194.233.101.89
alOcean, LLC 2 177 159.89.13.38
camp Limited 2 121 89.187.161.40
ade.sh 1 759 181.214.226.58
on Technologies, Inc. 1 529 172.93.207.161
YS AB 1 279 46.246.82.29
yer Inc 1 131 107.151.163.162
ade.sh LTDA 1 078 181.215.172.141
erhouse Management, Inc. 1 045 45.254.246.41
Royale Technologies Pvt Ltd 1 019 185.128.9.116
Rostelecom 883 87.117.189.127
Universal Pty Ltd 880 103.108.231.166
ibre 877 49.228.118.196
API Products
IP Geolocation API
IP Ranges API
Privacy Detection API
Data
Our Data
My IP Address
Tools
Resources
Documentation
Libraries
Help Centre
Company
About
Careers
Community

My API key

Before securing it

- Hardcoded API key

After securing it

- Encrypted API key in local storage with **AES-GCM** encryption
  - **PBKDF2** key with **PIN**, **random salt** and **IV**
- **DecryptedAPIkey** is in-memory only.

My sanitations and validations

Before securing it

- Attacker could input malicious code to input fields
- The AbuseIPDB API could send harmful information

After securing it

- When setting up API key the format is **validated** for API key and the PIN code.
- Trust but verify: All fetched information via API is **escaped.**

Mozilla extension security measures

Before securing it

- Attacker could run scripts from dev tools
- The extension could be permissioned to do something harmful

After securing it

- **content_security_policy** blocks sources that are not relevant and scripts.
- Only relevant permissions are given to extension which are used

Code stuff

Used to register the extension

manifest.json

content security policy

Permissions

Style of all html

style.css

The main algorithm

background.js

Opens stored IP in spur.us and opens the AbuseIPDB window

On right click opens shortcut menu

IP validation ipv4/ipv6

Decrypting API key
API key: AES-GCM
PIN: PBKDF2

Encrypts the API key

options.js

Encrypting API key
API key: AES-GCM
PIN: PBKDF2
browser.storage.local

Makes random salt + IV

Shows the UI for API setup

options.html

AbuseIPDB info fetcher

popup.js

Calls background.js functions to validate Highlighted IP

Escapes all API information

Asks for PIN

Shows the UI for AbuseIPDB window

popup.html

API fetch HTML escaping

popup.js

```
// Escapes dangerous HTML characters that could go through API
function escapeHtml(text) {

    return text.replace(/[&<>"']/g, match => { // Replace if any of these characters are present

        const escapeMap = {
            "&": "&amp;",
            "<": "&lt;",
            ">": "&gt;",
            "\"": "&quot;",
            "'": "&#039;"
        };

        return escapeMap[match] || match;

    });
}
```

options.js

```
try {
    // Creates random salt 16 bytes -> Harder to guess the PIN
    const salt = crypto.getRandomValues(new Uint8Array(16));

    // Creates a random IV with 12 bytes -> Encryption is unique with the same key
    const iv   = crypto.getRandomValues(new Uint8Array(12));

    // String to binary: e.g 1234 -> list of bytes
    const enc = new TextEncoder();

    // Takes the user entered PIN and makes it a crypto key
    const baseKey = await crypto.subtle.importKey(
        "raw",                  // raw binary of the PIN
        enc.encode(pin),        // PIN string -> PIN binary
        { name: "PBKDF2" },     // We use PBKDF2 from the library
        false,                  // Do not allow exporting
        ["deriveKey"]           // Can generate another key with this key
    );

    // Create a secure AES-GCM key using the PIN which uses PBKDF2
    const aesKey = await crypto.subtle.deriveKey(
        {
            name: "PBKDF2",         // Using PBKDF2
            salt: salt,             // Adding salt prevents rainbow table attacks
            iterations: 100000,     // Defends from brute-force, doing the thing many times
            hash: "SHA-256"         // Using secure hash function
        },
        baseKey,                            // Key made using the PIN
        { name: "AES-GCM", length: 256 }, // Makes AES key
        false,                              // key can't be extracted
        ["encrypt", "decrypt"]              // key can be encrypted/decrypted
    );
```

PBKDF2
Password-Based Key Derivation Function 2
= Turns PIN code to a strong key

AES-GCM
Advanced Encryption Standard Galois/Counter Mode
= Widely used symmetric encryption algorithm (Encrypts data with confidentiality + integrity)

```
// Encrypting the API key using the new AES-GCM key
const ciphertext = await crypto.subtle.encrypt(
    { name: "AES-GCM", iv: iv },// using AES-GCM and IV
    aesKey,                     // using the AES key that was made
    enc.encode(apiKey)          // convert from api key -> encrypted binary
```

Extension permissions

manifest.json

```
"content_security_policy": "default-src 'none'; script-src 'self'; style-src 'self'; connect-src https://api.abuseipdb.com/; img-src 'self';",
```

```
"permissions": [
  "contextMenus",
  "storage",
  "tabs",
  "notifications",
  "https://api.abuseipdb.com/",
  "https://spur.us/context/*"
],
```

My learnings:
Literally everything

- Making of Mozilla extension
- Security measures for Mozilla Extensions
- TESTING?
- AbuseIPDB API integration
- Secure API key handling
- All implementation of other security stuff
- Usage of ChatGPT in learning to program

Thank you!

Spur us + ABUSEIPDB check
Mozilla Extension