


master

...

security / advisories / SICK-2021-015.md

 sickcodes [CVE-2021-29923] + [CVE-2021-29921] + [CVE-2021-29922] Update CVSS ✓

History

1 contributor

141 lines (92 sloc) | 5.52 KB

Title

CVE-2021-29922 rust standard library "net" - Improper Input Validation of octal literals in rust 1.52.0 std::net and below results in indeterminate SSRF & RFI vulnerabilities.

CVE ID

CVE-2021-29922

CVSS Score

9.1 CRITICAL

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

Internal ID

SICK-2021-015

Vendor

rust-lang

Product

rust std::net

Product Versions

1.52.1 and below

Vulnerability Details

Improper input validation of octal strings in rust-lang standard library "net" allows unauthenticated remote attackers to perform indeterminate SSRF, RFI, and LFI attacks on many programs that rely on rust-lang std::net. IP address octets are left stripped instead of evaluated as valid IP addresses. For example, an attacker submitting an IP address to a web application that relies on std::net::IpAddr, could cause SSRF via inputting octal input data; An attacker can submit exploitable IP addresses if the octet is 3 digits, with the minimum exploitable octet being 08 (Denial of Service) and the maximum exploitable octet is 099 (Denial of Service). For example, an attacker can submit 010.8.8.8, which is 8.8.8.8 (RFI), yet std::net::IpAddr will evaluate this as 10.8.8.8. Equally, an attacker can input 127.0.026.1 which is really 127.0.22.1 but rust evaluates it as 127.0.26.1.

Vendor Response

Fixed in rust 1.53: [rust-lang/rust#83652](https://github.com/rust-lang/rust/pull/83652)

Proofs of Concept

```
#!/bin/bash
# Authors:      https://github.com/sickcodes, https://twitter.com/sickcodes
#              https://doc.rust-lang.org/std/net/struct.Ipv4Addr.html#method.is_private
# License:      GPLv3+
# https://play.rust-lang.org/?version=stable&mode=release&edition=2015&gist=62f6f98dc1de7d162ee4e62d09825035

cat <<EOF>poc.rs
#![allow(unused)]

fn main() {
    use std::net::Ipv4Addr;

    let localhost = Ipv4Addr::new(127, 0, 0, 1);
    assert_eq!("0127.0.0.1".parse(), Ok(localhost));
    assert_eq!(localhost.is_loopback(), true);
}
EOF

# vulnerable
docker run -it -v ~/poc.rs:/poc.rs rust:1.51 /bin/bash -c "rustc -V; rustc /poc.rs; ./poc && echo VULNERABLE"

# vulnerable
docker run -it -v ~/poc.rs:/poc.rs rust:1.52 /bin/bash -c "rustc -V; rustc /poc.rs; ./poc && echo VULNERABLE"
```

```
# fixed
docker run -it -v ~/poc.rs:/poc.rs rust:1.53 /bin/bash -c "rustc -V; rustc /poc.rs; ./poc && echo VULNERABLE"

# fixed
docker run -it -v ~/poc.rs:/poc.rs rust:1.54 /bin/bash -c "rustc -V; rustc /poc.rs; ./poc && echo VULNERABLE"

# fixed
docker run -it -v ~/poc.rs:/poc.rs rust:1.58 /bin/bash -c "rustc -V; rustc /poc.rs; ./poc && echo VULNERABLE"

// ##!/usr/bin/env rustc
// # Authors:      https://twitter.com/sickcodes, https://twitter.com/kaoudis
// # License:      GPLv3+

use std::net::IpAddr;

fn main() {
    let addr = "127.0.0.1".parse::<IpAddr>().unwrap();
    println!("{}", addr.to_string());
    let addr1 = "127.0.0.26.1".parse::<IpAddr>().unwrap();
    println!("{}", addr1.to_string());
    let addr2 = "127.0.0.093".parse::<IpAddr>().unwrap();
    println!("{}", addr2.to_string());
    let addr3 = "099.0.0.01".parse::<IpAddr>().unwrap();
    println!("{}", addr3.to_string());
}

// $ rustc -o main main.rs
// $ ./main
// 127.0.0.1
// 127.0.26.1
// 127.0.0.93
// 99.0.0.1
```

#### Disclosure Timeline

- 2021-03-29 - Researchers discover vulnerability
- 2021-03-29 - CVE requested
- 2021-03-30 - Fix merged in master [rust-lang/rust#83652](https://github.com/rust-lang/rust/pull/83652)
- 2021-08-07 - Release @ DEF CON 29 [https://www.youtube.com/watch?v=\\_o1RPJAe4kU](https://www.youtube.com/watch?v=_o1RPJAe4kU)

#### Links

[rust-lang/rust#83648](https://github.com/rust-lang/rust/pull/83648)

[rust-lang/rust#83652](https://github.com/rust-lang/rust/pull/83652)

<https://doc.rust-lang.org/beta/std/net/struct.Ipv4Addr.html>

<https://github.com/sickcodes/security/blob/master/advisories/SICK-2021-015.md>

<https://sick.codes/sick-2021-015>

[https://www.youtube.com/watch?v=\\_o1RPJAe4kU](https://www.youtube.com/watch?v=_o1RPJAe4kU)

<https://defcon.org/html/defcon-29/dc-29-speakers.html#kaoudis>

#### Researchers

Cheng Xu: <https://github.com/xu-cheng> || <https://xuc.me/>

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://github.com/johnjhacking> || <https://www.twitter.com/johnjhacking>

Nick Sahler: <https://github.com/nicksahler> || [https://twitter.com/tensor\\_bodega](https://twitter.com/tensor_bodega)

#### CVE Links

<https://sick.codes/sick-2021-015>

<https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-29922>

<https://nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-29922>