Title

CVE-2021-29923 golang standard library "net" - Improper Input Validation of octal literals in golang 1.16.2 and below standard library "net" results in indeterminate SSRF & RFI vulnerabilities.

CVE ID

CVE-2021-29923

CVSS Score

7.5 HIGH

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

Internal ID

SICK-2021-016

Vendor

golang

Product

golang standard library: net

Product Versions

1.16.3 and below

Vulnerability Details

Improper input validation of octal strings in golang standard library "net" allows unauthenticated remote attackers to perform indeterminate SSRF, RFI, and LFI attacks on many programs that rely on golang builtin net.ParseCIDR function. CIDR IP individual octects are left stripped instead of evaluated as valid IP octets. For example, an attacker submitting an IP CIDR to a web application that relies on net.ParseCIDR(ipaddress), could cause SSRF via inputting octal input data; A remote, local, authenticated or unauthenticated attacker can submit a myriad of exploitable IP addresses to applications that rely on golang net standard library. For example, an attacker can submit 00000177.0.0.1, which is 127.0.0.1 (RFI), yet net.ParseCIDR(00000177.0.0.1/24) will evaluate this as 177.0.0.1/24, when this is infact 127.0.0.1/24.

Vendor Response

Accepted to be fixed in 1.17 golang/go#30999

Proof of Concept

```
//usr/bin/go run $0 $@ ; exit
// Authors:      https://twitter.com/sickcodes, https://twitter.com/kaoudis
// License:      GPLv3+

package main

import "fmt"
import "net"

func main() {
  fmt.Println("octet 1: ")
  var input = "00000177.0.0.1/24"
  var addr, mask, err = net.ParseCIDR(input)
  fmt.Println("input: ", input, " result: ", addr, mask, err)

  fmt.Println("octet 2: ")
  var input1 = "192.0169.1.1/5"
  var addr1, mask1, err1 = net.ParseCIDR(input1)
  fmt.Println("input: ", input1, " result: ", addr1, mask1, err1)

  fmt.Println("octet 3: ")
  var input2 = "5.5.035.5/8"
  var addr2, mask2, err2 = net.ParseCIDR(input2)
   fmt.Println("input: ", input2, " result: ", addr2, mask2, err2)
```

```
    fmt.Println("trying an octal not in dot-decimal: ")
    var input3 = "035"
    var addr3, mask3, err3 = net.ParseCIDR(input3)
     fmt.Println("input: ", input3, " result: ", addr3, mask3, err3)

    fmt.Println("trying an octet in hexadecimal: ")
    var input4 = "1.0xff.2.3/19"
    var addr4, mask4, err4 = net.ParseCIDR(input4)
    fmt.Println("input: ", input4, " result: ", addr4, mask4, err4)

    fmt.Println("trying a hexadecimal not in dot-decimal: ")
    var input5 = "0xffff"
    var addr5, mask5, err5 = net.ParseCIDR(input5)
     fmt.Println("input: ", input5, " result: ", addr5, mask5, err5)
}
```

Disclosure Timeline

- **2019-03-22** - @opennota opens issue in 2019: net: reject leading zeros in IP address parsers #30999
- **2021-03-29** - Researchers discover vulnerability
- **2021-03-29** - CVE requested
- **2021-08-07** - Release @ DEF CON 29 https://www.youtube.com/watch?v=_o1RPJAe4kU

Links

https://golang.org/pkg/net/#ParseCIDR

golang/go#43389

golang/go#30999

https://go-review.googlesource.com/c/go/+/325829/

https://github.com/sickcodes/security/blob/master/advisories/SICK-2021-016.md

https://sick.codes/sick-2021-016

https://www.youtube.com/watch?v=_o1RPJAe4kU

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

Researchers

opennota: https://gitlab.com/opennota https://github.com/

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

CVE Links

https://sick.codes/sick-2021-016

https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-29923

https://nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-29923