Sergei Kuštšenko

Go SSH 0.0.2 Denial-of-Service (CVE-2020-9283)

Computer Security (LTAT.06.002) term paper

Tartu 2020
1 Introduction

This work contains a description of Go SSH server vulnerability found and reported by Mark Adams [1]. Before we start with the description it is necessary to explain different terms and shorthands [2, 3, 4, 5]:

- **Go**: Is a statically typed open source programming language.

- **SSH**: Is a cryptographic network protocol for secure data exchange between two computers over an untrusted network.

- **DoS attack**: A denial-of-service attack occurs when some malicious actor tries to manipulate network target or host in such a way that it can crash it or make unresponsive.

- **Ed25519**: Is a public-key digital signature system from EdDSA.

2 Vulnerability

The vulnerability itself is a possibility of client application crash a Go SSH server which is configured with PublicKeyCallback by sending a signature for verification with an invalid public key. The problem is that at some point of signature verification server discovers that received public key has an invalid length and starts to panic, which means that it stops code execution and exits with an error. Affected all Go SSH servers with goolang.org/x/crypto package before version v0.0.0-20200220183623-bac4c82f6975. Figure 1 shows how vulnerable SSH server behaves with an invalid public key.

```
2020/04/27 13:59:01 Vulnerable SSH server running on 0.6.6.6:2622
pentc: ed25519: bad public key length: 21
goroutine 6 [running]:
crypto/ed25519/Verify(0x0000942cf, 0x15, 0x2c, 0x000006700, 0x88, 0x100, 0x0000042f7, 0x0, 0x0, 0x888112280)
  /usr/local/go/src/crypto/ed25519/ed25519.go:175 +0x458
goolang.org/x/crypto/ed25519/Verify( ...)
  /go/pkg/mod/goolang.org/x/crypto@v0.8.8-282021020234226-1ae671f0f4f4/ed25519/ed25519.go:173
  /go/pkg/mod/goolang.org/x/crypto@v0.8.8-282021020234226-1ae671f0f4f4/sh/key.go:157 +0x1a0
  /go/pkg/mod/goolang.org/x/crypto@v0.8.8-282021020234226-1ae671f0f4f4/sh/key.go:157 +0x1a0
  /go/pkg/mod/goolang.org/x/crypto@v0.8.8-282021020234226-1ae671f0f4f4/sh/key.go:157 +0x1a0

Figure 1. Log of a vulnerable SSH server.
```

Figure 2 shows the place where the panic function is called.
Considering that this vulnerability can be triggered quite easily by a malicious actor or even by an ordinary user who sent an invalid or corrupted key occasionally it is necessary to update all SSH servers up to the latest version.

3 Fix

This vulnerability is fixed in golang.org/x/crypto package starting from version v0.0.0-20200220183623-bac4c82f6975. The fix contains public key length check before the signature verification in the golang.org/x/crypto/ed25519/ed25519.go and if this check fails, it returns an error to the client, rather than crashes server with panic. One place where this check is added is shown in Figure 3.

In the end, Mark Adams [1] validated the fix. Figure 4 shows that the server is actually fixed and it stays up even if it receives an invalid public key.

Figure 2. Place of panic in golang.org/x/crypto/ed25519/ed25519.go.

Figure 3. Modification in golang.org/x/crypto/ssh/keys.go.

Figure 4. Log of fixed SSH server.
References


Appendix

A License

I, Sergei Kuštšenko (15.03.1995) herewith grant the University of Tartu a free permit to publicly export this term paper until the year 2023.