russellhaering / goxmldsig Public

Code Issues 14 Pull requests 13 Actions Projects Security 1 ...

New issue Jump to bottom

Crash on nil-pointer dereference with malformed input #48

Closed stevenjohnstone opened this issue on Aug 14, 2019 · 10 comments · Fixed by #71

stevenjohnstone commented on Aug 14, 2019

See russellhaering/gosaml2#59 for background.

Program which exhibits the issue:

```go
package main

import (
        "crypto/x509"
        "encoding/base64"
        "encoding/xml"
        "fmt"
        "time"

        saml2 "github.com/russellhaering/gosaml2"
        "github.com/russellhaering/gosaml2/types"
        dsig "github.com/russellhaering/goxmldsig"
)

const (
        timeString   = "2019-08-12T12:00:52.718Z"
        oktaMetadata = `<?xml version="1.0" encoding="UTF-8"?><md:EntityDescriptor entityID="http://www.okta.com/exk133onomIuOW98z357" xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata">
A1UECAwKQ2FsaWZvcm5pYTEWMBQGA1UEBwwNU2FuIEZyYW5jaXNjbzENMAsGA1UECgwET2t0YTEU
MBIGA1UECwwLU1NPUHJvdmlkZXIxEzARBgNVBAMMCmRldi05MDUyNTExHDAaBgkqhkiG9w0BCQEW
DWluZm9Ab2t0YS5jb20wHhcNMTkwODA4MjA1MzMzWhcNMjkwODA4MjA1NDMzWjCBkjELMAkGA1UE
BhMCVVMxEzARBgNVBAgMCkNhbGlmb3JuaWExFjAUBgNVBAcMDVNhbiBGcmFuY2lzY28xDTALBgNV
BAoMBE9rdGExFDASBgNVBAsMC1NTT1Byb3ZpZGVyMRMwEQYDVQQDDApkZXYtOTA1MjUxMRwwGgYJ
KoZIhvcNAQkBFg1pbmZvQG9rdGEuY29tMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA
m+ZZF6aEG6ehLLIV6RPA+i1z6ss3HBG2bZD3efwKCDDXYUkp59AE7JsjVHMtpJPHhzHuScuHDMlu
HmkBQTW7j9XpnaRn8SfZXkwlCUHTo+HAC9lwbQxO4d4wnwgnm6FAjm1I/gbfFAobd8BR9pDxHuXE
MQ0DtQu/W3WbDUrz/bhSxPJAoVy2koQn9G0y3unm7eRwYWHeuW6GdPWV2szTtDS0c3qtUXVF5Ugg
iQYlwQu6xkfy4l8iGJL7ETa2BmJzwCFecMIct87SqNhYQwCBH54MBaHcaSsCKyimNvMY9B7RmC+H
4+awePPA1q3R/UQ3Pfom8mx6yDdKIWqlkG3MsQIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQAiURCZ
P4oJWcf1o5nm4yG15UH01g/S6Y4OUWMi6BFJy9fCrJ0h/2BZKi68SQ0uMAbdK6anxCzq3Rr5MSzW
OWPQ1Zljn3LGPsiTFdFca/GVRen5IYQ7Dr2Mvhtm+QVscEY9TDjtETbTAHEVEjwXmB21wtdIhizv
sQS7wz0A8LV+Atpbev45RiV6COmB6T6vJuFQ7ZsDZMSHZriTYiETTJvHBGd7PtbCxYNc6LRB2JDb
wlekRhVEjR0UhnM+nn2sqqbv7tDEPs63lZSDXCnR1PhscHrEuQ04rHI3OL0gCULVQFvJrj85IAZF
1QQuGUK8ozfOyFpQWAJUW71INnF/SLWv</ds:X509Certificate></ds:X509Data></ds:KeyInfo></md:KeyDescriptor><md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified</md:NameIDFor

        badInput = `<p ID="5"o=""t=""n=""xmlns:saml2p=""s=""><saml2 t=""l=""></l><s s=""><s><s I=""><s><s m=""></m></s><s></e></e></o><s></e><s><s><s></X></X></o></e><saml2p l=""></l>
)

func newServiceProvider() *saml2.SAMLServiceProvider {
        metadata := &types.EntityDescriptor{} // may need to support EntityDescriptors (plural)
        if err := xml.Unmarshal([]byte(oktaMetadata), metadata); err != nil {
                panic(err)
        }

        certStore := dsig.MemoryX509CertificateStore{
                Roots: []*x509.Certificate{},
        }

        for _, kd := range metadata.IDPSSODescriptor.KeyDescriptors {
                for _, xcert := range kd.KeyInfo.X509Data.X509Certificates {
                        if xcert.Data == "" {
                                panic("nope")
                        }
                        certData, err := base64.StdEncoding.DecodeString(xcert.Data)
                        if err != nil {
                                panic(err)
                        }

                        idpCert, err := x509.ParseCertificate(certData)
                        if err != nil {
                                panic(err)
                        }

                        certStore.Roots = append(certStore.Roots, idpCert)
                }
        }

        SSOs := metadata.IDPSSODescriptor.SingleSignOnServices

        tenantURI := "test.example.com"

        fakeTime, err := time.Parse(time.RFC3339, timeString)
        if err != nil {
                panic(err)
        }

        clock := dsig.NewFakeClockAt(fakeTime)

        return &saml2.SAMLServiceProvider{
                Clock:                       clock,
                IdentityProviderSSOURL:      SSOs[0].Location,
                IdentityProviderIssuer:      metadata.EntityID,
                ServiceProviderIssuer:       tenantURI,
                AssertionConsumerServiceURL: "https://127.0.0.1/login",
                SignAuthnRequests:           true,
                AudienceURI:                 tenantURI,
                IDPCertificateStore:         &certStore,
                ValidateEncryptionCert:      true,
        }
}

func main() {
        base64Input := base64.StdEncoding.EncodeToString([]byte(badInput))
        if _, err := newServiceProvider().RetrieveAssertionInfo(base64Input); err != nil {
                fmt.Printf("error %v\n", err)
        }
}
```

Panic:

```
panic: runtime error: invalid memory address or nil pointer dereference
[signal SIGSEGV: segmentation violation code=0x1 addr=0x28 pc=0x66bca4]
```

```
goroutine 1 [running]:
github.com/russellhaering/goxmldsig.(*ValidationContext).validateSignature(0xc000154f88, 0xc00019c240, 0xc00019e4c0, 0xc000126580, 0x0, 0x1, 0x6010105)
	/goroot/go/src/github.com/russellhaering/goxmldsig/validate.go:275 +0x2b4
github.com/russellhaering/goxmldsig.(*ValidationContext).Validate(0xc000154f88, 0xc00017d560, 0xc00017d560, 0x0, 0x0)
	/goroot/go/src/github.com/russellhaering/goxmldsig/validate.go:466 +0xf3
github.com/russellhaering/gosaml2.(*SAMLServiceProvider).validateAssertionSignatures.func1(0xc000191080, 0xc0000ada40, 0x9, 0x1)
	/goroot/go/src/github.com/russellhaering/gosaml2/decode_response.go:178 +0x2fc
github.com/russellhaering/goxmldsig/etreeutils.NSFindIterateCtx.func1(0xc000191080, 0xc0000ada40, 0xc000191080, 0x0)
	/goroot/go/src/github.com/russellhaering/goxmldsig/etreeutils/namespace.go:281 +0x137
github.com/russellhaering/goxmldsig/etreeutils.NSTraverse(0xc000190ba0, 0xc0000ada40, 0xc000190b70, 0x0, 0x0)
	/goroot/go/src/github.com/russellhaering/goxmldsig/etreeutils/namespace.go:148 +0x6e
github.com/russellhaering/goxmldsig/etreeutils.NSTraverse(0xc0000ccc60, 0xc0000ad4a0, 0xc000190b70, 0x30, 0x6ebda0)
	/goroot/go/src/github.com/russellhaering/goxmldsig/etreeutils/namespace.go:155 +0xe5
github.com/russellhaering/goxmldsig/etreeutils.NSFindIterateCtx(0xc0000ccc60, 0xc0000ad4a0, 0x71c4d1, 0x25, 0x712a6e, 0x9, 0xc000190b40, 0x7152c8, 0x12)
	/goroot/go/src/github.com/russellhaering/goxmldsig/etreeutils/namespace.go:268 +0xa4
github.com/russellhaering/goxmldsig/etreeutils.NSFindIterate(...)
	/goroot/go/src/github.com/russellhaering/goxmldsig/etreeutils/namespace.go:257
github.com/russellhaering/gosaml2.(*SAMLServiceProvider).validateAssertionSignatures(0xc000164000, 0xc0000ad4a0, 0x0, 0x0)
	/goroot/go/src/github.com/russellhaering/gosaml2/decode_response.go:200 +0xed
github.com/russellhaering/gosaml2.(*SAMLServiceProvider).ValidateEncodedResponse(0xc000164000, 0xc000143000, 0xe24, 0xc000013510, 0xc00015e000, 0xc0000d8b40)
	/goroot/go/src/github.com/russellhaering/gosaml2/decode_response.go:248 +0x30e
github.com/russellhaering/gosaml2.(*SAMLServiceProvider).RetrieveAssertionInfo(0xc000164000, 0xc000143000, 0xe24, 0xa9b, 0xc000143000, 0xe24)
	/goroot/go/src/github.com/russellhaering/gosaml2/retrieve_assertion.go:40 +0x9b
main.main()
	/goroot/go/src/github.com/russellhaering/gosaml2/crasher/main.go:93 +0xb6
exit status 2
```

Potential fix:

```
diff --git a/validate.go b/validate.go
index 55feb39..fe63af4 100644
--- a/validate.go
+++ b/validate.go
@@ -271,6 +271,10 @@ func (ctx *ValidationContext) validateSignature(el *etree.Element, sig *types.Si
 			return nil, errors.New("Signature could not be verified")
 		}
 
+	if sig.SignatureValue == nil {
+		return nil, errors.New("nil signature value")
+	}
+
 	// Decode the 'SignatureValue' so we can compare against it
 	decodedSignature, err := base64.StdEncoding.DecodeString(sig.SignatureValue.Data)
 	if err != nil {
```

carnil commented on Aug 24, 2020

This issue seem to have CVE-2020-7711 assigned.

p-rog commented on Oct 7, 2020

Will this issue be addressed?

harshithagowda commented on Oct 27, 2020 • edited ▾

Any updates on this fixing this issue?
Snyk is logging a high severity security vulnerability with this repo due to this issue. Any high severity vulnerability in a library of this nature, means it is currently unsuitable for enterprise projects.

Kentamanos commented on Dec 15, 2020

While this bug was filed against goxmldsig, it appears the test program above actually tests it using the saml2 library, and this seems to have been fixed (2 days ago) in the saml2 library. I can't get the test program to crash and it seems **@russellhaering** added code to make sure valid XML was being used here: russellhaering/gosaml2@ `0f0fb74` #diff-184d234308543ccf5984a0bfe952dbde13d02450a1b1d872c19b3184b646007dR375-R376

I suppose this means goxmldsig might technically still have a CVE, but if it's being used from the saml2 library (which is my case and why I was asked to look at this), it appears to be mitigated?

russellhaering-okta commented on Dec 15, 2020 Collaborator

Hey, sorry for the silence here. I think this was fixed as a side-effect of our resolution to another issue a few months ago, but let me push a quick fix to make it explicit.

👍 1

eclipseo commented on Jan 5, 2021

> Hey, sorry for the silence here. I think this was fixed as a side-effect of our resolution to another issue a few months ago, but let me push a quick fix to make it explicit.

Can you please confirm that 1.1.0 is not affected anymore?

aporcupine added a commit to aporcupine/goxmldsig that referenced this issue on Apr 5, 2021

`Explicitly check for case where SignatureValue is nil` ··· `94e448a`

aporcupine mentioned this issue on Apr 5, 2021

Explicitly check for case where SignatureValue is nil #71

Merged

aporcupine commented on Apr 5, 2021 • edited Contributor

Created #71 to explicitly handle the case of SignatureValue being nil which is enough to close this one out once reviewed and merged.

Schparky mentioned this issue on Apr 19, 2021

CVE-2020-7711 (High) detected in github.com/russellhaering/goxmldsig-3541f5e554eefd0d2ef501e27544650d62bf5d22 silinternational/wecarry-api#312

Open

p-rog commented on Apr 29, 2021

When will be a new release with official fix to CVE-2020-7711?
It wasn't confirmed that the patch for CVE-2020-15216 also fixed this vulnerability.

❤️ 1

bssudhir commented on Jul 1, 2021

> Hey, sorry for the silence here. I think this was fixed as a side-effect of our resolution to another issue a few months ago, but let me push a quick fix to make it explicit.

Can you please confirm if this issue is fixed? Synk is still reporting this issue.

❤️ 1

budanm commented on Aug 5, 2021

can you please release a security patch for CVE-2020-7711 and make it explicit? The whitesourcescan is still reporting this vulnerability in spite of having the version github.com/russellhaering/goxmldsig v1.1.0.

russellhaering closed this as completed in #71 on Aug 27, 2021

zserge mentioned this issue on Sep 23, 2021

Chore: Update goxmldsig dependency grafana/grafana#39566

Merged

mfridman mentioned this issue on Sep 30, 2021

Push a new tag and release russellhaering/gosaml2#86

Closed

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Successfully merging a pull request may close this issue.

Explicitly check for case where SignatureValue is nil
aporcupine/goxmldsig

10 participants