



How to Build SBOM from Binaries: A Round About Story

September 2022

Changing the World's Energy Future

Robert J Erbes



INL is a U.S. Department of Energy National Laboratory operated by Battelle Energy Alliance, LLC

DISCLAIMER

This information was prepared as an account of work sponsored by an agency of the U.S. Government. Neither the U.S. Government nor any agency thereof, nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness, of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the U.S. Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the U.S. Government or any agency thereof.

How to Build SBOM from Binaries: A Round About Story

Robert J Erbes

September 2022

**Idaho National Laboratory
Idaho Falls, Idaho 83415**

<http://www.inl.gov>

**Prepared for the
U.S. Department of Energy
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517**

HOW TO BUILD SBOM FROM BINARIES

a round about story

Robert.Erbes @ Idaho National Laboratory

INL/MIS-22-68677

“Top Down” SBOM



Product A

- * Product C
- * Product D



Product B

- * Library 1
- * Product E
- * OpenSSL

Scenario: Product SBOM says “OpenSSL”

■ UH OH!

Openssl » Openssl : Security Vulnerabilities							
CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9							
Sort Results By : CVE Number Descending CVE Number Ascending CVSS Score Descending Number Of Exploits Descend							
Total number of vulnerabilities : 215 Page : 1 (This Page) 2 3 4 5							
Copy Results Download Results							
#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score
1	CVE-2022-2274	787		Exec Code Mem. Corr.	2022-07-01	2022-07-15	10.0
The OpenSSL 3.0.4 release introduced a serious bug in the RSA implementation for X86_64 CPUs supporting the on such machines and memory corruption will happen during the computation. As a consequence of the memory computation. SSL/TLS servers or other servers using 2048 bit RSA private keys running on machines supporting							
2	CVE-2022-2097	326			2022-07-05	2022-07-23	5.0
AES OCB mode for 32-bit x86 platforms using the AES-NI assembly optimised implementation will not encrypt the preexisting in the memory that wasn't written. In the special case of "in place" encryption, sixteen bytes of the p they are both unaffected. Fixed in OpenSSL 3.0.5 (Affected 3.0.0-3.0.4). Fixed in OpenSSL 1.1.1q (Affected 1.1.							
3	CVE-2022-2068	78		Exec Code	2022-06-21	2022-07-23	10.0
In addition to the c_rehash shell command injection identified in CVE-2022-1292, further circumstances where t found by code review. When the CVE-2022-1292 was fixed it was not discovered that there are other places in t through the shell. This script is distributed by some operating systems in a manner where it is automatically exe the script. Use of the c_rehash script is considered obsolete and should be replaced by the OpenSSL rehash com (Affected 1.1.1-1.1.1o). Fixed in OpenSSL 1.0.2zf (Affected 1.0.2-1.0.2ze).							
4	CVE-2022-1473	404		DoS	2022-05-03	2022-06-02	5.0
The OPENSSL_LH_flush() function, which empties a hash table, contains a bug that breaks reuse of the memory a long lived process periodically decodes certificates or keys its memory usage will expand without bounds and t empty hash table entries will take increasingly more time. Typically such long lived processes might be TLS client OpenSSL 3.0 version thus older releases are not affected by the issue. Fixed in OpenSSL 3.0.3 (Affected 3.0.0,3.							
5	CVE-2022-1434	327			2022-05-03	2022-06-02	4.3

2022

[CVE-2022-2274 \(OpenSSL advisory\)](#) [[High severity](#)] 05 July 2022: [↑](#)

The OpenSSL 3.0.4 release introduced a serious bug in the RSA implementation for X86_64 CPUs supporting the AVX512IFMA instructions. This issue makes the RSA implementation with 2048 bit private keys incorrect on such machines and memory corruption will happen during the computation. As a consequence of the memory corruption an attacker may be able to trigger a remote code execution on the machine performing the computation. SSL/TLS servers or other servers using 2048 bit RSA private keys running on machines supporting AVX512IFMA instructions of the X86_64 architecture are affected by this issue. Reported by Xi Ruoyao.

- Fixed in OpenSSL 3.0.5 ([git commit](#)) (Affected 3.0.4)

2022

[CVE-2022-2274 \(OpenSSL advisory\)](#) [[High severity](#)] 05 July 2022: [↑](#)

The OpenSSL 3.0.4 release introduced a serious bug in the RSA implementation for X86_64 CPUs supporting the AVX512IFMA instructions. This issue makes the RSA implementation with 2048 bit private keys incorrect on such machines and memory corruption will happen during the computation. As a consequence of the memory corruption an attacker may be able to trigger a remote code execution on the machine performing the computation. SSL/TLS servers or other servers using 2048 bit RSA private keys running on machines supporting AVX512IFMA instructions of the X86_64 architecture are affected by this issue. Reported by Xi Ruoyao.

- Fixed in OpenSSL 3.0.5 ([git commit](#)) (Affected 3.0.4)

What if we developed a “bottom up” SBOM?

- Verify vendor SBOM
- No vendor SBOM available
- Just curious

Can be used for:

- Incident Response
- Supply Chain attack detection
- Risk Management



Let's look at PowerPoint

```
[robert@big Microsoft PowerPoint.app % pwd  
/Applications/Microsoft PowerPoint.app  
[robert@big Microsoft PowerPoint.app % find . | wc -l  
28509  
robert@big Microsoft PowerPoint.app %
```

```
robert@Big Contents % otool -l MacOS/Microsoft PowerPoint
MacOS/Microsoft PowerPoint (architecture x86_64):
 @path/ADAL4.framework/Versions/A/ADAL4 (compatibility version 0.0.0, current version 0.0.0)
 @path/Chart.framework/Versions/A/Chart (compatibility version 0.0.0, current version 0.0.0)
 @path/CocoaUI.framework/Versions/A/CocoaUI (compatibility version 0.0.0, current version 0.0.0)
 @path/COMBase.framework/Versions/A/COMBase (compatibility version 0.0.0, current version 0.0.0)
 @path/MicrosoftConversionLibrary.framework/Versions/A/MicrosoftConversionLibrary (compatibility version 0.0.0, current version 0.0.0)
 @path/MicrosoftCredui.framework/Versions/A/MicrosoftCredui (compatibility version 0.0.0, current version 0.0.0)
 @path/MicrosoftCSI.framework/Versions/A/MicrosoftCSI (compatibility version 0.0.0, current version 0.0.0)
 @path/dwrite10.framework/Versions/A/dwrite10 (compatibility version 0.0.0, current version 0.0.0)
 @path/MicrosoftFBA.framework/Versions/A/MicrosoftFBA (compatibility version 0.0.0, current version 0.0.0)
 @path/Forms.framework/Versions/A/Forms (compatibility version 0.0.0, current version 0.0.0)
 @path/Gfx.framework/Versions/A/Gfx (compatibility version 0.0.0, current version 0.0.0)
 @path/Hlink.framework/Versions/A/Hlink (compatibility version 0.0.0, current version 0.0.0)
 @path/SmartArt.framework/Versions/A/SmartArt (compatibility version 0.0.0, current version 0.0.0)
 @path/MicrosoftCloudServices.framework/Versions/A/MicrosoftCloudServices (compatibility version 0.0.0, current version 0.0.0)
 @path/MicrosoftFontLibrary.framework/Versions/A/MicrosoftFontLibrary (compatibility version 0.0.0, current version 0.0.0)
 @path/mbuinstrument.framework/Versions/A/mbuinstrument (compatibility version 0.0.0, current version 0.0.0)
 @path/mbukernel.framework/Versions/A/mbukernel (compatibility version 0.0.0, current version 0.0.0)
 @path/mbulocale.framework/Versions/A/mbulocale (compatibility version 0.0.0, current version 0.0.0)
 @path/merp.framework/Versions/A/merp (compatibility version 0.0.0, current version 0.0.0)
 @path/MetEx.framework/Versions/A/MetEx (compatibility version 0.0.0, current version 0.0.0)
 @path/SAExt.framework/Versions/A/SAExt (compatibility version 0.0.0, current version 0.0.0)
 @path/mso20.framework/Versions/A/mso20 (compatibility version 0.0.0, current version 0.0.0)
 @path/mso30.framework/Versions/A/mso30 (compatibility version 0.0.0, current version 0.0.0)
 @path/mso4oui.framework/Versions/A/mso4oui (compatibility version 0.0.0, current version 0.0.0)
 @path/mso99.framework/Versions/A/mso99 (compatibility version 0.0.0, current version 0.0.0)
 @path/MsoCF.framework/Versions/A/MsoCF (compatibility version 0.0.0, current version 0.0.0)
 @path/MSXML.framework/Versions/A/MSXML (compatibility version 0.0.0, current version 0.0.0)
 @path/OfficeArt.framework/Versions/A/OfficeArt (compatibility version 0.0.0, current version 0.0.0)
 @path/OcsClient.framework/Versions/A/OcsClient (compatibility version 0.0.0, current version 0.0.0)
 @path/OLE.framework/Versions/A/OLE (compatibility version 0.0.0, current version 0.0.0)
 @path/FluentUI.framework/Versions/A/FluentUI (compatibility version 0.0.0, current version 0.0.0)
 @path/osf.framework/Versions/A/osf (compatibility version 0.0.0, current version 0.0.0)
 @path/osstorage2.framework/Versions/A/osstorage2 (compatibility version 0.0.0, current version 0.0.0)
 @path/libmsix.dylib (compatibility version 0.0.0, current version 0.0.0)
 /System/Library/Frameworks/Security.framework/Versions/A/Security (compatibility version 1.0.0, current version 60158.100.133)
 @path/ProofingUI.framework/Versions/A/ProofingUI (compatibility version 0.0.0, current version 0.0.0)
 @path/MicrosoftPTLS.framework/Versions/A/MicrosoftPTLS (compatibility version 0.0.0, current version 0.0.0)
 @path/MicrosoftPTLS7.framework/Versions/A/MicrosoftPTLS7 (compatibility version 0.0.0, current version 0.0.0)
 @path/MicrosoftRichEdit.framework/Versions/A/MicrosoftRichEdit (compatibility version 0.0.0, current version 0.0.0)
 @path/MicrosoftWebServices.framework/Versions/A/MicrosoftWebServices (compatibility version 0.0.0, current version 0.0.0)
 /System/Library/Frameworks/Accelerate.framework/Versions/A/Accelerate (compatibility version 1.0.0, current version 4.0.0)
 /System/Library/Frameworks/ApplicationServices.framework/Versions/A/ApplicationServices (compatibility version 1.0.0, current version 56.0.0)
 /System/Library/Frameworks/AudioToolbox.framework/Versions/A/AudioToolbox (compatibility version 1.0.0, current version 1000.0.0)
 /System/Library/Frameworks/AVFoundation.framework/Versions/A/AVFoundation (compatibility version 1.0.0, current version 2.0.0)
 /System/Library/Frameworks/Cocoa.framework/Versions/A/Cocoa (compatibility version 1.0.0, current version 23.0.0)
 /System/Library/Frameworks/CoreAudio.framework/Versions/A/CoreAudio (compatibility version 1.0.0, current version 1.0.0)
 /System/Library/Frameworks/CoreMedia.framework/Versions/A/CoreMedia (compatibility version 1.0.0, current version 1.0.0)
 /System/Library/Frameworks/CoreServices.framework/Versions/A/CoreServices (compatibility version 1.0.0, current version 1141.1.0)
 /System/Library/Frameworks/CoreVideo.framework/Versions/A/CoreVideo (compatibility version 1.2.0, current version 1.5.0)
 /System/Library/Frameworks/IOKit.framework/Versions/A/IOKit (compatibility version 1.0.0, current version 275.0.0)
 /System/Library/Frameworks/MediaAccessibility.framework/Versions/A/MediaAccessibility (compatibility version 1.0.0, current version 62.0.0)
 /System/Library/Frameworks/OpenGL.framework/Versions/A/OpenGL (compatibility version 1.0.0, current version 1.0.0)
 /System/Library/Frameworks/QuartzCore.framework/Versions/A/QuartzCore (compatibility version 1.2.0, current version 1.11.0)
 /System/Library/Frameworks/ScriptingBridge.framework/Versions/A/ScriptingBridge (compatibility version 1.0.0, current version 1.0.0)
 /System/Library/Frameworks/SystemConfiguration.framework/Versions/A/SystemConfiguration (compatibility version 1.0.0, current version 1163.1)
 /System/Library/Frameworks/VideoToolbox.framework/Versions/A/VideoToolbox (compatibility version 1.0.0, current version 1.0.0)
 /System/Library/Frameworks/WebKit.framework/Versions/A/WebKit (compatibility version 1.0.0, current version 613.1.17)
 @path/Uniscribe.framework/Versions/A/Uniscribe (compatibility version 0.0.0, current version 0.0.0)
 @path/Visual Basic for Applications.framework/Versions/A/Visual Basic for Applications (compatibility version 0.0.0, current version 0.0.0)
 @path/WinCrypto.framework/Versions/A/WinCrypto (compatibility version 0.0.0, current version 0.0.0)
 @path/OLEAutomation.framework/Versions/A/OLEAutomation (compatibility version 0.0.0, current version 0.0.0)
 @path/rpcrt4.framework/Versions/A/rpcrt4 (compatibility version 0.0.0, current version 0.0.0)
 @path/WMGraphicsDevice.framework/Versions/A/WMGraphicsDevice (compatibility version 0.0.0, current version 0.0.0)
 @path/WMKernel.framework/Versions/A/WMKernel (compatibility version 0.0.0, current version 0.0.0)
 @path/Xmllite.framework/Versions/A/Xmllite (compatibility version 0.0.0, current version 0.0.0)
 /System/Library/Frameworks/Foundation.framework/Versions/C/Foundation (compatibility version 300.0.0, current version 1858.112.0)
 /usr/lib/libobjc.A.dylib (compatibility version 1.0.0, current version 228.0.0)
 /usr/lib/libc++.1.dylib (compatibility version 1.0.0, current version 1300.23.0)
 /usr/lib/libSystem.B.dylib (compatibility version 1.0.0, current version 1311.108.3)
 /System/Library/Frameworks/AppKit.framework/Versions/C/AppKit (compatibility version 45.0.0, current version 2113.40.126)
 /System/Library/Frameworks/Combine.framework/Versions/A/Combine (compatibility version 1.0.0, current version 280.101.0)
 /System/Library/Frameworks/CoreFoundation.framework/Versions/A/CoreFoundation (compatibility version 150.0.0, current version 1858.112.0)
 /System/Library/Frameworks/CoreGraphics.framework/Versions/A/CoreGraphics (compatibility version 64.0.0, current version 1557.5.4)
 /System/Library/Frameworks/CoreText.framework/Versions/A/CoreText (compatibility version 1.0.0, current version 1.0.0)
 /System/Library/Frameworks/ImageIO.framework/Versions/A/ImageIO (compatibility version 1.0.0, current version 1.0.0)
 /System/Library/Frameworks/Metal.framework/Versions/A/Metal (compatibility version 1.0.0, current version 261.13.0)
 /System/Library/Frameworks/UIKit.framework/Versions/A/UIKit (compatibility version 0.0.0, current version 0.0.0)
 /usr/lib/swift/libswiftAVFoundation.dylib (compatibility version 1.0.0, current version 2080.20.4, weak)
 /usr/lib/swift/libswiftAppKit.dylib (compatibility version 1.0.0, current version 189.0.0)
 /usr/lib/swift/libswiftCore.dylib (compatibility version 1.0.0, current version 5.6.0)
 /usr/lib/swift/libswiftCoreAudio.dylib (compatibility version 1.0.0, current version 1.1.0, weak)
 /usr/lib/swift/libswiftCoreData.dylib (compatibility version 1.0.0, current version 19.0.0, weak)
 /usr/lib/swift/libswiftCoreFoundation.dylib (compatibility version 1.0.0, current version 14.0.0)
 /usr/lib/swift/libswiftCoreGraphics.dylib (compatibility version 1.0.0, current version 2.0.0)
 /usr/lib/swift/libswiftCoreImage.dylib (compatibility version 1.0.0, current version 2.0.0, weak)
 /usr/lib/swift/libswiftCoreMIDI.dylib (compatibility version 1.0.0, current version 5.0.0, weak)
 /usr/lib/swift/libswiftCoreMedia.dylib (compatibility version 1.0.0, current version 2940.20.4)
 /usr/lib/swift/libswiftDarwin.dylib (compatibility version 1.0.0, current version 0.0.0)
 /usr/lib/swift/libswiftDispatch.dylib (compatibility version 1.0.0, current version 11.0.0)
 /usr/lib/swift/libswiftFoundation.dylib (compatibility version 1.0.0, current version 72.105.0)
 /usr/lib/swift/libswiftIOKit.dylib (compatibility version 1.0.0, current version 1.0.0, weak)
 /usr/lib/swift/libswiftMetal.dylib (compatibility version 1.0.0, current version 261.13.0, weak)
 /usr/lib/swift/libswiftObjectiveC.dylib (compatibility version 1.0.0, current version 3.0.0)
 /usr/lib/swift/libswiftQuartzCore.dylib (compatibility version 1.0.0, current version 3.0.0, weak)
 /usr/lib/swift/libswiftUniformTypeIdentifiers.dylib (compatibility version 1.0.0, current version 722.6.0, weak)
 /usr/lib/swift/libswiftWebKit.dylib (compatibility version 1.0.0, current version 613.1.17, weak)
 /usr/lib/swift/libswiftXPC.dylib (compatibility version 1.0.0, current version 1.1.0, weak)
 /usr/lib/swift/libswiftos.dylib (compatibility version 1.0.0, current version 1023.0.0)
 /usr/lib/swift/libswiftsimd.dylib (compatibility version 1.0.0, current version 9.0.0, weak)
 @path/libswift_Concurrency.dylib (compatibility version 1.0.0, current version 5.6.0, weak)
```

POWERPOINT & SHARED LIBRARY DEPENDENCIES

Hey look, it's WebKit

```
/System/Library/Frameworks/VideoToolbox.framework/Versions/A/VideoToolbox (compatibility version 1.0.0, current version 1.0.0)
/System/Library/Frameworks/WebKit.framework/Versions/A/WebKit (compatibility version 1.0.0, current version 613.1.17)
@rpath/Uniscribe.framework/Versions/A/Uniscribe (compatibility version 0.0.0, current version 0.0.0)
@rpath/Visual Basic for Applications.framework/Versions/A/Visual Basic for Applications (compatibility version 0.0.0, current version 0.0.0)
@rpath/WinCrypto.framework/Versions/A/WinCrypto (compatibility version 0.0.0, current version 0.0.0)
@rpath/OLEAutomation.framework/Versions/A/OLEAutomation (compatibility version 0.0.0, current version 0.0.0)
```

Is there CYBER RISK to using WebKit?

[Webkit](#) » [Webkit](#) : Security Vulnerabilities

CVSS Scores Greater Than: [0](#) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#)

Sort Results By : [CVE Number Descending](#) [CVE Number Ascending](#) [CVSS Score Descending](#) [Number Of Exploits Descending](#)

[Copy Results](#) [Download Results](#)

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score
1	CVE-2016-9643	400		DoS	2017-03-07	2017-07-12	5.0
The regex code in Webkit 2.4.11 allows remote attackers to cause a denial of service (memory consumption) as demon number of +) (plus close parenthesis).							
2	CVE-2016-9642	125		DoS	2017-02-03	2017-07-12	4.3
JavaScriptCore in WebKit allows attackers to cause a denial of service (out-of-bounds heap read) via a crafted Javascript							
3	CVE-2010-1766	189		DoS Mem. Corr.	2010-07-22	2013-02-07	7.5
Off-by-one error in the WebSocketHandshake::readServerHandshake function in websockets/WebSocketHandshake.cpp websockets servers to cause a denial of service (memory corruption) or possibly have unspecified other impact via an u							
4	CVE-2008-6059	264		+Info	2009-02-05	2017-08-08	5.0
xml/XMLHttpRequest.cpp in WebCore in WebKit before r38566 does not properly restrict access from web pages to the obtain sensitive information from cookies via XMLHttpRequest calls, related to the HTTPOnly protection mechanism.							
Total number of vulnerabilities : 4 Page : 1 (This Page)							

- Only 4 CVE! Nice.
- Latest CVE affects regex code in... WebKit 2.4.11 ??
- otool -L said we had WebKit version 613.1.17 ???

Apple » Webkit : Security Vulnerabilities

CVSS Scores Greater Than: [0](#) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#)

Sort Results By : [CVE Number Descending](#) [CVE Number Ascending](#) [CVSS Score Descending](#) [Number Of Exploits Descending](#)

Total number of vulnerabilities : **93** Page : [1](#) (This Page) [2](#)

[Copy Results](#) [Download Results](#)

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score
1	CVE-2017-13870	119		DoS Exec Code Overflow Mem. Corr.	2017-12-25	2019-03-22	6.8
An issue was discovered in certain Apple products. iOS before 11.2 is affected. Safari before 11.0.2 is affected. iCloud is affected. The issue involves the "WebKit" component. It allows remote attackers to execute arbitrary code or cause							
2	CVE-2017-13866	119		DoS Exec Code Overflow Mem. Corr.	2017-12-25	2019-03-22	6.8
An issue was discovered in certain Apple products. iOS before 11.2 is affected. Safari before 11.0.2 is affected. iCloud is affected. The issue involves the "WebKit" component. It allows remote attackers to execute arbitrary code or cause							
3	CVE-2017-13856	119		DoS Exec Code Overflow Mem. Corr.	2017-12-25	2019-03-22	6.8
An issue was discovered in certain Apple products. iOS before 11.2 is affected. Safari before 11.0.2 is affected. iCloud is affected. The issue involves the "WebKit" component. It allows remote attackers to execute arbitrary code or cause							
4	CVE-2017-13803	119		DoS Exec Code Overflow Mem. Corr.	2017-11-13	2019-03-22	6.8
An issue was discovered in certain Apple products. iOS before 11.1 is affected. Safari before 11.0.1 is affected. iCloud is affected. The issue involves the "WebKit" component. It allows remote attackers to execute arbitrary code or cause							
5	CVE-2017-13802	119		DoS Exec Code Overflow Mem. Corr.	2017-11-13	2019-03-22	6.8
An issue was discovered in certain Apple products. iOS before 11.1 is affected. Safari before 11.0.1 is affected. iCloud is affected. The issue involves the "WebKit" component. It allows remote attackers to execute arbitrary code or cause							
6	CVE-2017-13798	119		DoS Exec Code Overflow Mem. Corr.	2017-11-13	2019-03-22	6.8
An issue was discovered in certain Apple products. iOS before 11.1 is affected. Safari before 11.0.1 is affected. iCloud is affected. The issue involves the "WebKit" component. It allows remote attackers to execute arbitrary code or cause							

APPLE WEBKIT

- Way more CVE (93)
- "certain Apple products"
- Stopped counting in 2017?
 - ...pretty sure there's been newer vulns

<https://usa.kaspersky.com> › ... › Threats

Update iOS, there is a dangerous vulnerability in WebKit

Feb 11, 2022 — Dangerous vulnerability in **WebKit** (**CVE-2022-22620**) is believed to be actively exploited by hackers. Update your iOS devices as soon as possible!

<https://googleprojectzero.github.io> › CVE-2021-30858

CVE-2021-30858: WebKit use-after-free in IndexedDB

CVE-2021-30858: WebKit use-after-free in IndexedDB. Maddie Stone, Google Project Zero. The Basics. Disclosure or Patch Date: 13 September 2021. Product: Apple ...

CVE-2022-*?!?!?

<https://threatpost.com> › Web Security

Apple Patches Actively Exploited WebKit Zero Day - Threatpost

Feb 11, 2022 — The zero-day, tracked as **CVE-2022-22620**, is a Use-After-Free issue, which is related to incorrect use of dynamic memory during program operation ...

<https://support.apple.com> › en-us

About the security content of Safari 15.6 - Apple Support

Jul 20, 2022 — **CVE-2022-32784:** Young Min Kim of CompSec Lab at Seoul National University.

WebKit. Available for: macOS Big Sur and macOS Catalina

Let us not worry about WebKit

- We know it's there.
 - We know it's a dependency.
 - We can hash it and worry about versions later.
-
- How about statically compiled libraries? Y'know. Like...

...what about OpenSSL?

```
[robert@big Microsoft PowerPoint.app % cd Contents  
[robert@big Contents % grep -ir openssl *  
Binary file Frameworks/MSRightsManagement.framework/Versions/A/MSRightsManagement matches  
Binary file Frameworks/MicrosoftCSI.framework/Versions/A/MicrosoftCSI matches  
Binary file Frameworks/libmsix.dylib matches  
Binary file MacOS/Microsoft PowerPoint matches  
Binary file SharedSupport/Open XML for Excel.app/Contents/MacOS/Open XML for Excel matches  
robert@big Contents % █
```

```
robert@big Contents % strings Frameworks/MSRightsManagement.framework/Versions/A/MSRightsManagement | grep -i openssl
OpenSSL CMAC method
OpenSSL default
openssl_conf
OPENSSL_CONF
openssl.cnf
OPENSSL_init
OPENSSL_finish
OPENSSL_atexit
OPENSSL_buf2hexstr
openssl_fopen
OPENSSL_hexstr2buf
OPENSSL_init_crypto
OPENSSL_LH_new
OPENSSL_sk_deep_copy
OPENSSL_sk_dup
OPENSSL_ia32cap
%s:%d: OpenSSL internal error: %s
OpenSSL PKCS#3 DH method
OpenSSL X9.42 DH method
OpenSSL DH Method
OpenSSL DSA method
OpenSSL 'dlfcn' shared library method
OpenSSL EC algorithm
OpenSSL EC_KEY method
OpenSSL X25519 algorithm
OpenSSL X448 algorithm
OpenSSL ED25519 algorithm
OpenSSL ED448 algorithm
OPENSSL_ENGINES
openssl
(TEST_ENG_OPENSSL_RC4) test_init_key() called
(TEST_ENG_OPENSSL_PKEY)Loading Private key %s
OpenSSL HMAC method
OPENSSL_asc2uni
OPENSSL_uni2asc
OPENSSL_uni2utf8
OPENSSL_utf82uni
OpenSSL POLY1305 method
OpenSSL RSA method
OpenSSL RSA-PSS method
OpenSSL PKCS#1 RSA
OpenSSL SIPHASH method
OpenSSL NULL UI
OpenSSL default user interface
crypto/ui/ui_openssl.c
OpenSSL NIST SP 800-90A DRBG
OpenSSL NIST SP 800-90A DRBG
OPENSSL_armcap
OpenSSL CMAC method
```

YEP.
OPENSSL

```

robert@big Contents % otool -L Frameworks/MSRightsManagement.framework/Versions/A/MSRightsManagement
Frameworks/MSRightsManagement.framework/Versions/A/MSRightsManagement (architecture x86_64):
  @rpath/MSRightsManagement.framework/Versions/A/MSRightsManagement (compatibility version 1.0.0, current version 1.0.0)
  /usr/lib/libz.1.dylib (compatibility version 1.0.0, current version 1.2.11)
  /usr/lib/libc++.1.dylib (compatibility version 1.0.0, current version 905.6.0)
  /usr/lib/libsqlite3.dylib (compatibility version 9.0.0, current version 321.3.0)
  /usr/lib/libresolv.9.dylib (compatibility version 1.0.0, current version 1.0.0)
  /System/Library/Frameworks/SystemConfiguration.framework/Versions/A/SystemConfiguration (compatibility version 1.0.0, current version 1109.101.1)
  /System/Library/Frameworks/AppKit.framework/Versions/C/AppKit (compatibility version 45.0.0, current version 2022.44.149)
  /System/Library/Frameworks/Security.framework/Versions/A/Security (compatibility version 1.0.0, current version 59754.100.106)
  /System/Library/Frameworks/Foundation.framework/Versions/C/Foundation (compatibility version 300.0.0, current version 1775.118.101)
  /System/Library/Frameworks/Cocoa.framework/Versions/A/Cocoa (compatibility version 1.0.0, current version 23.0.0)
  /usr/lib/libobjc.A.dylib (compatibility version 1.0.0, current version 228.0.0)
  /usr/lib/libSystem.B.dylib (compatibility version 1.0.0, current version 1292.100.5)
  /System/Library/Frameworks/CoreData.framework/Versions/A/CoreData (compatibility version 1.0.0, current version 1048.0.0)
  /System/Library/Frameworks/CoreFoundation.framework/Versions/A/CoreFoundation (compatibility version 150.0.0, current version 1775.118.101)
Frameworks/MSRightsManagement.framework/Versions/A/MSRightsManagement (architecture arm64):
  @rpath/MSRightsManagement.framework/Versions/A/MSRightsManagement (compatibility version 1.0.0, current version 1.0.0)
  /usr/lib/libz.1.dylib (compatibility version 1.0.0, current version 1.2.11)
  /usr/lib/libc++.1.dylib (compatibility version 1.0.0, current version 905.6.0)
  /usr/lib/libsqlite3.dylib (compatibility version 9.0.0, current version 321.3.0)
  /usr/lib/libresolv.9.dylib (compatibility version 1.0.0, current version 1.0.0)
  /System/Library/Frameworks/SystemConfiguration.framework/Versions/A/SystemConfiguration (compatibility version 1.0.0, current version 1109.101.1)
  /System/Library/Frameworks/AppKit.framework/Versions/C/AppKit (compatibility version 45.0.0, current version 2022.44.149)
  /System/Library/Frameworks/Security.framework/Versions/A/Security (compatibility version 1.0.0, current version 59754.100.106)
  /System/Library/Frameworks/Foundation.framework/Versions/C/Foundation (compatibility version 300.0.0, current version 1775.118.101)
  /System/Library/Frameworks/Cocoa.framework/Versions/A/Cocoa (compatibility version 1.0.0, current version 23.0.0)
  /usr/lib/libobjc.A.dylib (compatibility version 1.0.0, current version 228.0.0)
  /usr/lib/libSystem.B.dylib (compatibility version 1.0.0, current version 1292.100.5)
  /System/Library/Frameworks/CoreData.framework/Versions/A/CoreData (compatibility version 1.0.0, current version 1048.0.0)
  /System/Library/Frameworks/CoreFoundation.framework/Versions/A/CoreFoundation (compatibility version 150.0.0, current version 1775.118.101)
robert@big Contents % otool -L Frameworks/MSRightsManagement.framework/Versions/A/MSRightsManagement | grep -i openssl
robert@big Contents % █

```

Is OpenSSL dynamically loaded?

– *Nope. It's compiled.*

We could stop here

- We know OpenSSL is compiled into MSRightsManagement.
 - Absent any other information we could list OpenSSL as a “component” within MSRightsManagement, which itself is a shared library dependency of PowerPoint.
-
- But that doesn't tell us anything about those 215 CVEs.
 - Before we start RE how OpenSSL is used by MSRightsManagement, we can do more digging.

<div> <div>Library function</div> <div>Regular function</div> <div>Instruction</div> <div>Data</div> <div>Unexplored</div> <div>External symbol</div> <div>Lumina function</div> </div>				
<div> <div>Functions</div> <div>IDA View-A</div> <div>Strings</div> <div>Hex View-1</div> <div>Structures</div> </div>				
Function name	Address	Length	Type	String
<code>__cstring:0000000001FE426</code>	<code>__cstring:0000000001FE426</code>	00000017	C	rsa_pss_keygen_mgf1_md
<code>__cstring:0000000001FE43D</code>	<code>__cstring:0000000001FE43D</code>	00000012	C	rsa_pss_keygen_md
<code>__cstring:0000000001FE44F</code>	<code>__cstring:0000000001FE44F</code>	00000017	C	rsa_pss_keygen_saltlen
<code>__cstring:0000000001FE466</code>	<code>__cstring:0000000001FE466</code>	0000000C	C	rsa_oaep_md
<code>__cstring:0000000001FE472</code>	<code>__cstring:0000000001FE472</code>	0000000F	C	rsa_oaep_label
<code>__cstring:0000000001FE481</code>	<code>__cstring:0000000001FE481</code>	00000015	C	crypto/rsa/rsa_pss.c
<code>__cstring:0000000001FE496</code>	<code>__cstring:0000000001FE496</code>	00000016	C	crypto/rsa/rsa_saos.c
<code>__cstring:0000000001FE4AC</code>	<code>__cstring:0000000001FE4AC</code>	00000016	C	crypto/rsa/rsa_sign.c
<code>__cstring:0000000001FE4C2</code>	<code>__cstring:0000000001FE4C2</code>	00000015	C	crypto/rsa/rsa_ssl.c
<code>__cstring:0000000001FE4D7</code>	<code>__cstring:0000000001FE4D7</code>	00000016	C	crypto/rsa/rsa_x931.c
<code>__cstring:0000000001FE4ED</code>	<code>__cstring:0000000001FE4ED</code>	00000008	C	SIPHASH
<code>__cstring:0000000001FE4F5</code>	<code>__cstring:0000000001FE4F5</code>	00000017	C	OpenSSL SIPHERHASH method
<code>__cstring:0000000001FE50C</code>	<code>__cstring:0000000001FE50C</code>	0000001F	C	crypto/siphhash/siphhash_pmeth.c
<code>__cstring:0000000001FE52B</code>	<code>__cstring:0000000001FE52B</code>	0000000B	C	digestsize
<code>__cstring:0000000001FE536</code>	<code>__cstring:0000000001FE536</code>	0000000F	C	SM2_Ciphertext
<code>__cstring:0000000001FE545</code>	<code>__cstring:0000000001FE545</code>	00000017	C	crypto/sm2/sm2_crypt.c
<code>__cstring:0000000001FE56A</code>	<code>__cstring:0000000001FE56A</code>	00000017	C	crypto/sm2/sm2_pmeth.c
<code>__cstring:0000000001FE581</code>	<code>__cstring:0000000001FE581</code>	00000016	C	crypto/sm2/sm2_sign.c
<code>__cstring:0000000001FE597</code>	<code>__cstring:0000000001FE597</code>	00000015	C	crypto/stack/stack.c
<code>__cstring:0000000001FE5AC</code>	<code>__cstring:0000000001FE5AC</code>	0000001B	C	crypto/store/loader_file.c

Find tokens in the binary we can use to compare against the open-source repository

```

__text:0000000001809B0 ; ===== S U B R O U T I N E =====
__text:0000000001809B0
__text:0000000001809B0 ; Attributes: bp-based frame
__text:0000000001809B0
__text:0000000001809B0 ; int __cdecl RSA_padding_add_SSLv23(unsigned __int8 *to, int tlen, const unsigned __int8 *f, int fl)
__text:0000000001809B0         public _RSA_padding_add_SSLv23
__text:0000000001809B0         _RSA_padding_add_SSLv23 proc near ; CODE XREF: sub_17B8B0+1F3+p
v__text:0000000001809B0         push     rbp
__text:0000000001809B1         mov      rbp, rsp
__text:0000000001809B4         push     r15
__text:0000000001809B6         push     r14
__text:0000000001809B8         push     r13
__text:0000000001809BA         push     r12
__text:0000000001809BC         push     rbx
__text:0000000001809BD         push     rax
__text:0000000001809BE         mov      r12d, esi
__text:0000000001809C1         add      r12d, 0FFFFFF5h
__text:0000000001809C5         sub      r12d, ecx
__text:0000000001809C8         jge      short loc_1809F2
__text:0000000001809CA         lea      rcx, aCryptoRsaRsaSs ; "crypto/rsa/rsa_ssl.c"
__text:0000000001809D1         mov      edi, 4 ; lib
__text:0000000001809D6         mov      esi, 6Eh ; 'n' ; func
__text:0000000001809DB         mov      edx, 6Eh ; 'n' ; reason
__text:0000000001809E0         mov      r8d, 19h ; line
__text:0000000001809E6         call     _ERR_put_error
__text:0000000001809EB         xor      eax, eax
__text:0000000001809ED         jmp      loc_180A72
__text:0000000001809F2 ;

```

Error strings are perfect.

Find crypto/rsa/rsa_ssl.c and look at line 0x19 (25) for a call to RSAerr()

CRYPTO/RSA/RSA_
SSL.C
...IS *MISSING*

github.com/openssl/openssl/tree/master/crypto/rsa

Product Team Enterprise Explore Marketplace Pricing Search Sign in Sign up

openssl / openssl Public Sponsor Notifications Fork 8.3k Star 19.1k

<> Code Issues 1.6k Pull requests 273 Actions Projects 2 Wiki Security Insights

master openssl / crypto / rsa / Go to file

slontis and t8m Fix memory leak in ossl_rsa_fromdata. 28adea9 on Jun 28 History

..		
build.info	Remove RSA SSLv23 padding mode	2 years ago
rsa_acvp_test_params.c	Update copyright year	16 months ago
rsa_ameth.c	Add sensitive memory clean in priv encode	2 months ago
rsa_asn1.c	Update copyright year	16 months ago
rsa_backend.c	Fix memory leak in ossl_rsa_fromdata.	2 months ago
rsa_chk.c	The rsa_validate_keypair_multiprime() function return is not boolean	2 months ago
rsa_crpt.c	Convert all (NAME)err() in crypto/ to their corresponding ERR_raise()...	2 years ago
rsa_depr.c	Update copyright year	2 years ago
rsa_err.c	crypto: updates to pass size_t to RAND_bytes_ex()	15 months ago
rsa_gen.c	RSA Keygen update - When using the default provider fallback to defau...	2 months ago
rsa_lib.c	Fix the check of evp_pkey_ctx_set_params_strict	2 months ago
rsa_local.h	Fix change in behaviour of EVP_PKEY_CTRL_RSA_KEYGEN_BITS	17 months ago
rsa_meth.c	Convert all (NAME)err() in crypto/ to their corresponding ERR_raise()...	2 years ago
rsa_mp.c	Update copyright year	16 months ago
rsa_mp_names.c	rsa: add ossl_ prefix to internal rsa_ calls.	2 years ago
rsa_none.c	Convert all (NAME)err() in crypto/ to their corresponding ERR_raise()...	2 years ago
rsa_oaep.c	Rename all getters to use get/get0 in name	15 months ago
rsa_ossl.c	Update copyright year	3 months ago
rsa_pk1.c	Update copyright year	14 months ago
rsa_pmeth.c	fix some code with obvious wrong coding style	10 months ago
rsa_prn.c	Update copyright year	13 months ago
rsa_pss.c	Rename all getters to use get/get0 in name	15 months ago
rsa_saos.c	Convert all (NAME)err() in crypto/ to their corresponding ERR_raise()...	2 years ago
rsa_schemes.c	rsa: add ossl_ prefix to internal rsa_ calls.	2 years ago
rsa_sign.c	fips module header inclusion fine-tuning	14 months ago
rsa_sp800_56b_check.c	Allow small RSA exponents in the default provider	12 months ago
rsa_sp800_56b_gen.c	fix some code with obvious wrong coding style	10 months ago
rsa_x931.c	Convert all (NAME)err() in crypto/ to their corresponding ERR_raise()...	2 years ago
rsa_x931g.c	Update copyright year	2 years ago

← → ↺ 🏠 github.com/openssl/openssl/blob/OpenSSL_1_1_1-stable/crypto/rsa/rsa_ssl

Product Team Enterprise Explore Marketplace Pricing

openssl / openssl Public Sponsor Noti

<> Code Issues 1.6k Pull requests 273 Actions Projects 2

🔗 OpenSSL_1_1_1-... openssl / crypto / rsa / rsa_ssl.c

mattcaswell Update copyright year [...]

9 contributors

176 lines (152 sloc) 5.96 KB

```
1 /*
2  * Copyright 1995-2021 The OpenSSL Project Authors. All Rights Reserved.
3  *
4  * Licensed under the OpenSSL license (the "License"). You may not use
5  * this file except in compliance with the License. You can obtain a copy
6  * in the file LICENSE in the source distribution or at
7  * https://www.openssl.org/source/license.html
8  */
9
10 #include <stdio.h>
11 #include "internal/cryptlib.h"
12 #include <openssl/bn.h>
13 #include <openssl/rsa.h>
14 #include <openssl/rand.h>
15 #include "internal/constant_time.h"
16
17 int RSA_padding_add_SSLv23(unsigned char *to, int tlen,
18                          const unsigned char *from, int flen)
19 {
20     int i, j;
21     unsigned char *p;
22
23     if (flen > (tlen - RSA_PKCS1_PADDING_SIZE)) {
24         RSAerr(RSA_F_RSA_PADDING_ADD_SSLV23,
25              RSA_R_DATA_TOO_LARGE_FOR_KEY_SIZE);
26         return 0;
27     }
28
29     p = (unsigned char *)to;
```

← Found it!

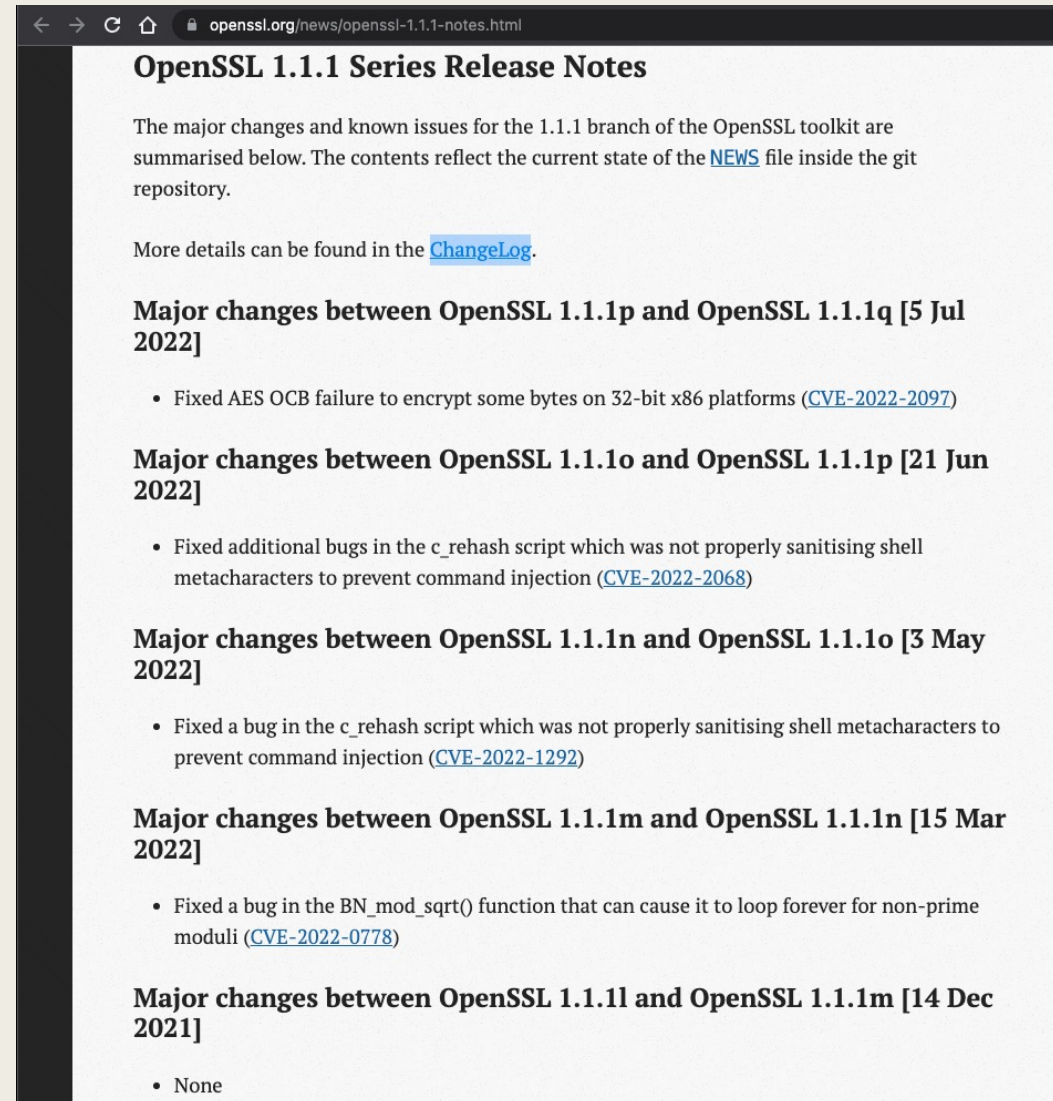
```
__text:0000000001809B0 ; ===== SUBROUTINE =====
__text:0000000001809B0
__text:0000000001809B0 ; Attributes: bp-based frame
__text:0000000001809B0 ; int __cdecl RSA_padding_add_SSLv23(unsigned __int8 *to, int tlen, const unsigned __int8 *f, int fl)
__text:0000000001809B0 public RSA_padding_add_SSLv23
__text:0000000001809B0 RSA_padding_add_SSLv23 proc near ; CODE XREF: sub_17B8B0+1F3+p
__text:0000000001809B0 push rbp
__text:0000000001809B1 mov rbp, rsp
__text:0000000001809B4 push r15
__text:0000000001809B6 push r14
__text:0000000001809B8 push r13
__text:0000000001809BA push r12
__text:0000000001809BC push rbx
__text:0000000001809BD push rax
__text:0000000001809BE mov r12d, esi
__text:0000000001809C1 add r12d, 0FFFFFFFh
__text:0000000001809C5 sub r12d, ecx
__text:0000000001809C8 jge short loc_1809F2
__text:0000000001809CA lea rcx, aCryptoRsaRsaSs ; "crypto/rsa/rsa_ssl.c"
__text:0000000001809D1 mov edi, 4 ; lib
__text:0000000001809D6 mov esi, 6Eh ; 'n' ; func
__text:0000000001809DB mov edx, 6Eh ; 'n' ; reason
__text:0000000001809E0 mov r8d, 19h ; line
__text:0000000001809E6 call _ERR_put_error
__text:0000000001809EB xor eax, eax
__text:0000000001809ED jmp loc_180A72
__text:0000000001809F2 ; =====
```

And it matches the binary

We know it's OpenSSL 1.X.Y

Cool.

We just don't know which release



The screenshot shows a web browser window with the URL `openssl.org/news/openssl-1.1.1-notes.html`. The page title is "OpenSSL 1.1.1 Series Release Notes". The main text states: "The major changes and known issues for the 1.1.1 branch of the OpenSSL toolkit are summarised below. The contents reflect the current state of the [NEWS](#) file inside the git repository." It then says: "More details can be found in the [ChangeLog](#)."

The page lists several major changes between versions:

- Major changes between OpenSSL 1.1.1p and OpenSSL 1.1.1q [5 Jul 2022]**
 - Fixed AES OCB failure to encrypt some bytes on 32-bit x86 platforms ([CVE-2022-2097](#))
- Major changes between OpenSSL 1.1.1o and OpenSSL 1.1.1p [21 Jun 2022]**
 - Fixed additional bugs in the `c_rehash` script which was not properly sanitising shell metacharacters to prevent command injection ([CVE-2022-2068](#))
- Major changes between OpenSSL 1.1.1n and OpenSSL 1.1.1o [3 May 2022]**
 - Fixed a bug in the `c_rehash` script which was not properly sanitising shell metacharacters to prevent command injection ([CVE-2022-1292](#))
- Major changes between OpenSSL 1.1.1m and OpenSSL 1.1.1n [15 Mar 2022]**
 - Fixed a bug in the `BN_mod_sqrt()` function that can cause it to loop forever for non-prime moduli ([CVE-2022-0778](#))
- Major changes between OpenSSL 1.1.1l and OpenSSL 1.1.1m [14 Dec 2021]**
 - None

```

__text:0000000000099EC0 ; int __cdecl BN_mod_exp_mont_word(BIGNUM *r, unsigned __int64 a, const BIGNUM *p,
__text:0000000000099EC0 public _BN_mod_exp_mont_word
__text:0000000000099EC0 _BN_mod_exp_mont_word proc near ; CODE XREF: _BN_mod_exp+D2+j
__text:0000000000099EC0 var_70 = qword ptr -70h
__text:0000000000099EC0 r = qword ptr -68h
__text:0000000000099EC0 w = qword ptr -60h
__text:0000000000099EC0 var_58 = dword ptr -58h
__text:0000000000099EC0 var_54 = dword ptr -54h
__text:0000000000099EC0 rem = qword ptr -50h
__text:0000000000099EC0 var_48 = qword ptr -48h
__text:0000000000099EC0 ctx = qword ptr -40h
__text:0000000000099EC0 a = qword ptr -38h
__text:0000000000099EC0 mont = qword ptr -30h
__text:0000000000099EC0
__text:0000000000099EC0 push rbp
__text:0000000000099EC1 mov rbp, rsp
__text:0000000000099EC4 push r15
__text:0000000000099EC6 push r14
__text:0000000000099EC8 push r13
__text:0000000000099ECA push r12
__text:0000000000099ECC push rbx
__text:0000000000099ECD sub rsp, 48h
__text:0000000000099ED1 mov r12, r9
__text:0000000000099ED4 mov rbx, r8
__text:0000000000099ED7 mov r13, rcx
__text:0000000000099EDA mov r14, rdx
__text:0000000000099EDD mov [rbp+var_48], rsi
__text:0000000000099EE1 mov r15, rdi
__text:0000000000099EE4 mov rdi, rdx
__text:0000000000099EE7 mov esi, 4
__text:0000000000099EEC call _BN_get_flags
__text:0000000000099EF1 test eax, eax
__text:0000000000099EF3 jnz short loc_99F06
__text:0000000000099EF5 mov rdi, r13
__text:0000000000099EF8 mov esi, 4
__text:0000000000099EFD call _BN_get_flags
__text:0000000000099F02 test eax, eax
__text:0000000000099F04 jz short loc_99F3C
__text:0000000000099F06
__text:0000000000099F06 loc_99F06: ; CODE XREF: _BN_mod_exp_mont_word+33+j
__text:0000000000099F06 lea rcx, aCryptoBnBnExpC ; "crypto/bn/bn_exp.c"
__text:0000000000099F0D mov edi, 3 ; lib
__text:0000000000099F12 mov esi, 75h ; 'u' ; func
__text:0000000000099F17 mov edx, 42h ; 'B' ; reason
__text:0000000000099F1C mov r8d, 486h ; line
__text:0000000000099F22
__text:0000000000099F22 loc_99F22: ; CODE XREF: _BN_mod_exp_mont_word+14B+j
__text:0000000000099F22 call _ERR_put_error
__text:0000000000099F27 xor r15d, r15d

```

```

1135     return ret;
1136 }
1137
1138 int BN_mod_exp_mont_word(BIGNUM *rr, BN_ULONG a, const BIGNUM *p,
1139                          const BIGNUM *m, BN_CTX *ctx, BN_MONT_CTX *in_mont)
1140 {
1141     BN_MONT_CTX *mont = NULL;
1142     int b, bits, ret = 0;
1143     int r_is_one;
1144     BN_ULONG w, next_w;
1145     BIGNUM *r, *t;
1146     BIGNUM *swap_tmp;
1147     #define BN_MOD_MUL_WORD(r, w, m) \
1148         (BN_mul_word(r, (w)) && \
1149          (/* BN_ucmp(r, (m)) < 0 ? 1 :*/ \
1150           (BN_mod(t, r, m, ctx) && (swap_tmp = r, r = t, t = swap_tmp, 1))))
1151     /*
1152      * BN_MOD_MUL_WORD is only used with 'w' large, so the BN_ucmp test is
1153      * probably more overhead than always using BN_mod (which uses BN_copy if
1154      * a similar test returns true).
1155      */
1156     /*
1157      * We can use BN_mod and do not need BN_nnmod because our accumulator is
1158      * never negative (the result of BN_mod does not depend on the sign of
1159      * the modulus).
1160      */
1161     #define BN_TO_MONTGOMERY_WORD(r, w, mont) \
1162         (BN_set_word(r, (w)) && BN_to_montgomery(r, r, (mont), ctx))
1163
1164     if (BN_get_flags(p, BN_FLG_CONSTTIME) != 0
1165         || BN_get_flags(m, BN_FLG_CONSTTIME) != 0) {
1166         /* BN_FLG_CONSTTIME only supported by BN_mod_exp_mont() */
1167         BNerr(BN_F_BN_MOD_EXP_MONT_WORD, ERR_R_SHOULD_NOT_HAVE_BEEN_CALLED);
1168         return 0;
1169     }
1170
1171     bn_check_top(p);
1172     bn_check_top(m);
1173
1174     if (!BN_is_odd(m)) {
1175         BNerr(BN_F_BN_MOD_EXP_MONT_WORD, BN_R_CALLED_WITH_EVEN_MODULUS);
1176         return 0;
1177     }

```



```

→ ↻ 🏠 🔒 github.com/openssl/openssl/blob/b5acbf914833a83368e51766de4cf2e2074a9436/crypto/bn/bn_exp.c
1128
1129 int BN_mod_exp_mont_word(BIGNUM *rr, BN_ULONG a, const BIGNUM *p,
1130                          const BIGNUM *m, BN_CTX *ctx, BN_MONT_CTX *in_mont)
1131 {
1132     BN_MONT_CTX *mont = NULL;
1133     int b, bits, ret = 0;
1134     int r_is_one;
1135     BN_ULONG w, next_w;
1136     BIGNUM *r, *t;
1137     BIGNUM *swap_tmp;
1138 #define BN_MOD_MUL_WORD(r, w, m) \
1139     (BN_mul_word(r, (w)) && \
1140      (/* BN_ucmp(r, (m)) < 0 ? 1 : */ \
1141       (BN_mod(t, r, m, ctx) && (swap_tmp = r, r = t, t = swap_tmp, 1))))
1142     /*
1143      * BN_MOD_MUL_WORD is only used with 'w' large, so the BN_ucmp test is
1144      * probably more overhead than always using BN_mod (which uses BN_copy if
1145      * a similar test returns true).
1146      */
1147     /*
1148      * We can use BN_mod and do not need BN_nnmod because our accumulator is
1149      * never negative (the result of BN_mod does not depend on the sign of
1150      * the modulus).
1151      */
1152 #define BN_TO_MONTGOMERY_WORD(r, w, mont) \
1153     (BN_set_word(r, (w)) && BN_to_montgomery(r, r, (mont), ctx))
1154
1155     if (BN_get_flags(p, BN_FLG_CONSTTIME) != 0
1156         || BN_get_flags(m, BN_FLG_CONSTTIME) != 0) {
1157         /* BN_FLG_CONSTTIME only supported by BN_mod_exp_mont() */
1158         BNerr(BN_F_BN_MOD_EXP_MONT_WORD, ERR_R_SHOULD_NOT_HAVE_BEEN_CALLED);
1159         return 0;
1160     }
1161
1162     bn_check_top(p);
1163     bn_check_top(m);
1164
1165     if (!BN_is_odd(m)) {
1166         BNerr(BN_F_BN_MOD_EXP_MONT_WORD, BN_R_CALLED_WITH_EVEN_MODULUS);
1167         return 0;

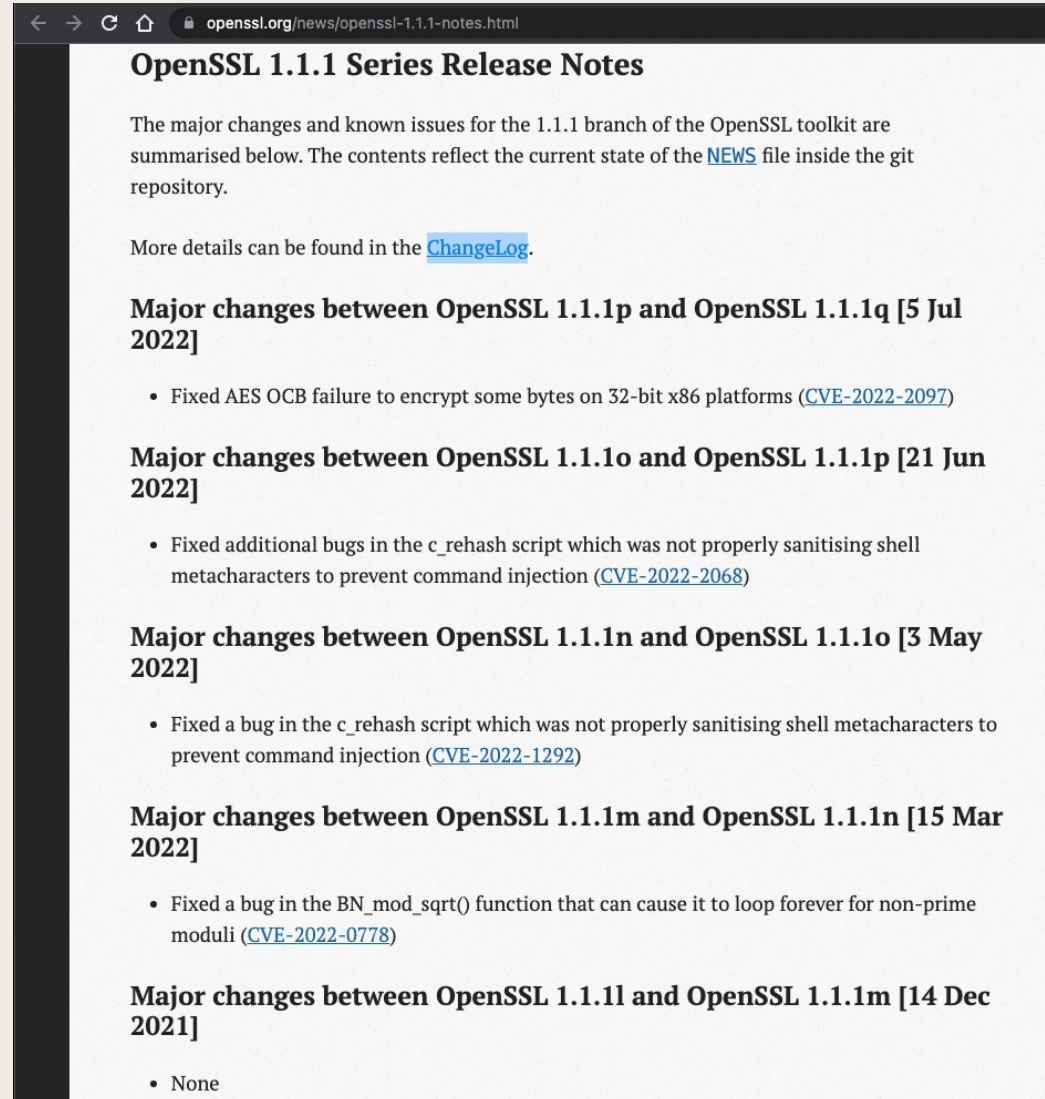
```

Found it

- This commit was on 27 September 2019
- Next commit (which didn't match) was on 23 March 2022
- The copy of OpenSSL in PowerPoint was built / released before 23 March 2022.
- Which means...

A release before 23 March 2022 means that **at least three** of these CVE's *might* apply.

And so, it goes on...



The screenshot shows the 'OpenSSL 1.1.1 Series Release Notes' page. The browser's address bar displays 'openssl.org/news/openssl-1.1.1-notes.html'. The page content includes an introductory paragraph, a link to 'ChangeLog', and a series of sections detailing major changes between various OpenSSL versions, each accompanied by a list of CVEs.

OpenSSL 1.1.1 Series Release Notes

The major changes and known issues for the 1.1.1 branch of the OpenSSL toolkit are summarised below. The contents reflect the current state of the [NEWS](#) file inside the git repository.

More details can be found in the [ChangeLog](#).

Major changes between OpenSSL 1.1.1p and OpenSSL 1.1.1q [5 Jul 2022]

- Fixed AES OCB failure to encrypt some bytes on 32-bit x86 platforms ([CVE-2022-2097](#))

Major changes between OpenSSL 1.1.1o and OpenSSL 1.1.1p [21 Jun 2022]

- Fixed additional bugs in the `c_rehash` script which was not properly sanitising shell metacharacters to prevent command injection ([CVE-2022-2068](#))

Major changes between OpenSSL 1.1.1n and OpenSSL 1.1.1o [3 May 2022]

- Fixed a bug in the `c_rehash` script which was not properly sanitising shell metacharacters to prevent command injection ([CVE-2022-1292](#))

Major changes between OpenSSL 1.1.1m and OpenSSL 1.1.1n [15 Mar 2022]

- Fixed a bug in the `BN_mod_sqrt()` function that can cause it to loop forever for non-prime moduli ([CVE-2022-0778](#))

Major changes between OpenSSL 1.1.1l and OpenSSL 1.1.1m [14 Dec 2021]

- None

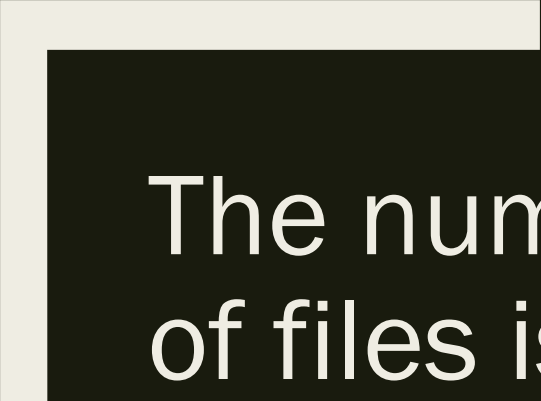
And we can do this.

- which good news!
- What's more, because we know how to do it manually means *we can automate it!*
 - *For OpenSSL*
 - *On architectures we can disassemble*
 - *That haven't been modified*
 - *On binaries we know how to load*
 - *...*
- We absolutely need to automate some of this... because it is not always this easy

One component of one library down

```
[robert@big Microsoft PowerPoint.app % pwd  
/Applications/Microsoft PowerPoint.app  
[robert@big Microsoft PowerPoint.app % find . | wc -l  
28509  
robert@big Microsoft PowerPoint.app %
```

Only 28,508 files to go!



The number of files isn't the whole story

- What files are important files?
 - *(All of them)*
- What about files created / used at runtime?
 - *(Capture those too)*
- What if the product downloads an update?
 - *(Uh-Oh)*
- What if there are files that are only installed / downloaded depending on your configuration or system?
 - *(please stop)*
- What about the files used to build/run the files? Compilers, interpreters, OS Kernels?
 - *(nervous laughter)*

2022

[CVE-2022-2274 \(OpenSSL advisory\)](#) [[High severity](#)] 05 July 2022: [↑](#)

The OpenSSL 3.0.4 release introduced a serious bug in the RSA implementation for X86_64 CPUs supporting the AVX512IFMA instructions. This issue makes the RSA implementation with 2048 bit private keys incorrect on such machines and memory corruption will happen during the computation. As a consequence of the memory corruption an attacker may be able to trigger a remote code execution on the machine performing the computation. SSL/TLS servers or other servers using 2048 bit RSA private keys running on machines supporting AVX512IFMA instructions of the X86_64 architecture are affected by this issue. Reported by Xi Ruoyao.

- Fixed in OpenSSL 3.0.5 ([git commit](#)) (Affected 3.0.4)



WHY “FILES”

and not software packages, like OpenSSL?



The General Process for Building SBOM from a Binary

Input: a file

- Have I seen this file before?
 - Yes: *Done!*
 - No: *Next step*
- Can I pull other files out of it?
 - Yes: *pull the files out and start again*
 - No: *Next step*
- What's inside?
 - "*Nothing*": *You're done!*
 - "*Something*": *Next step*
- Have I seen the something before?
 - Yes: *Done!*
 - No: *Figure out what it is and start again*



The General Process is General

- Works on
 - *“Firmware”*
 - *JavaScript*
 - *PE*
 - *ELF*
 - *PDF*
 - *ISO*
 - *MSI*
 - *DOCX*
 - *Python*
 - *TEXT*
 - ...



HOW TO BUILD SBOM FROM BINARIES

a round about story

Robert.Erbes @ Idaho National Laboratory

