Applicazioni *mobile* : [in]sicurezza e supply chain

Denise Nanni, Gabriele D'Angelo

Alma Mater Studiorum - Università di Bologna

HACK IN BO®
Winter 2024 Edition
23ª EDIZIONE

Tabella dei Contenuti

- Chi siamo
- Introduzione
- Metodologia
- Risultati
- Conclusioni e Sviluppi Futuri

HACK IN BO® Winter 2024 Edition
23ª EDIZIONE

~$ whoami

HACK IN BO®
Winter 2024 Edition
23ª EDIZIONE

Denise Nanni

HACK IN BO®
Winter 2024 Edition
23ª EDIZIONE

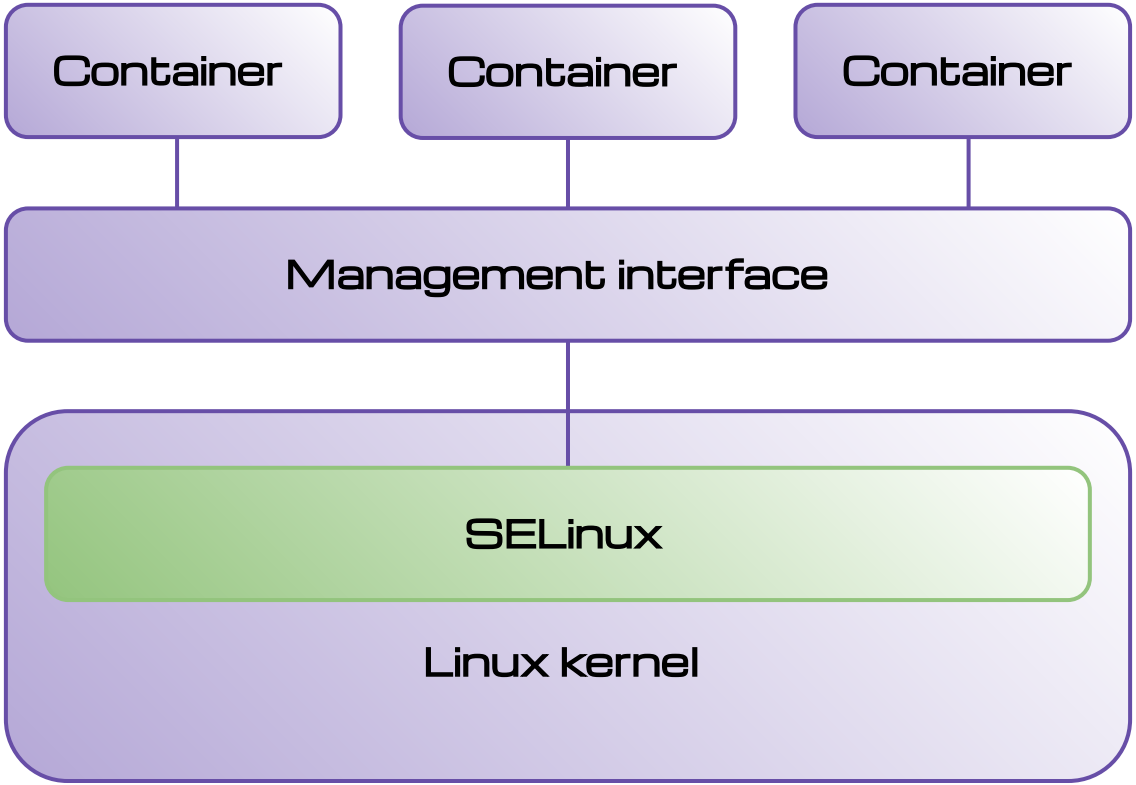
Gabriele D'Angelo

~$ man intro

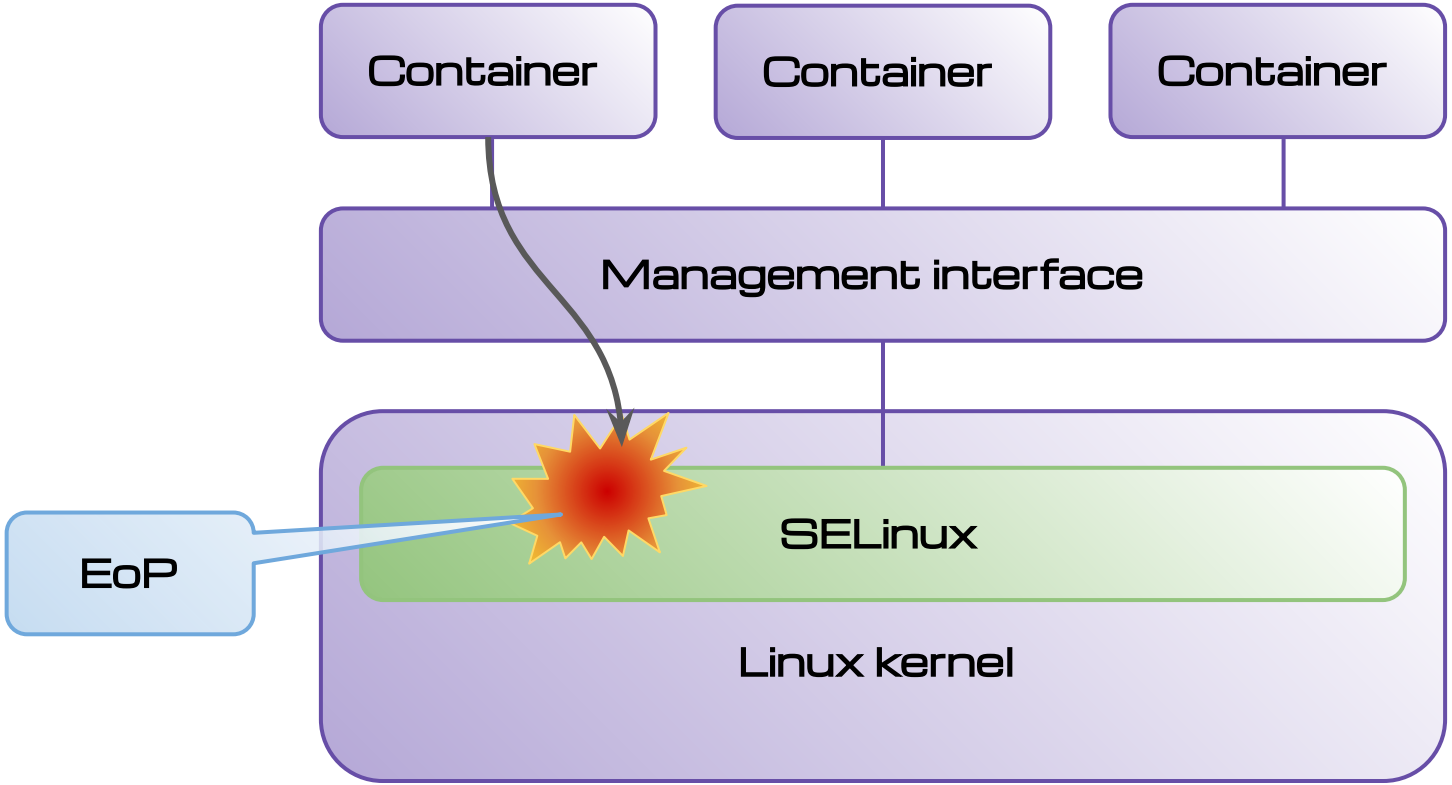
Mobile digital transformation

Fonte: Microsoft Copilot

- sempre più "**valore**" contenuto negli smartphone
- **superficie d'attacco** molto vasta

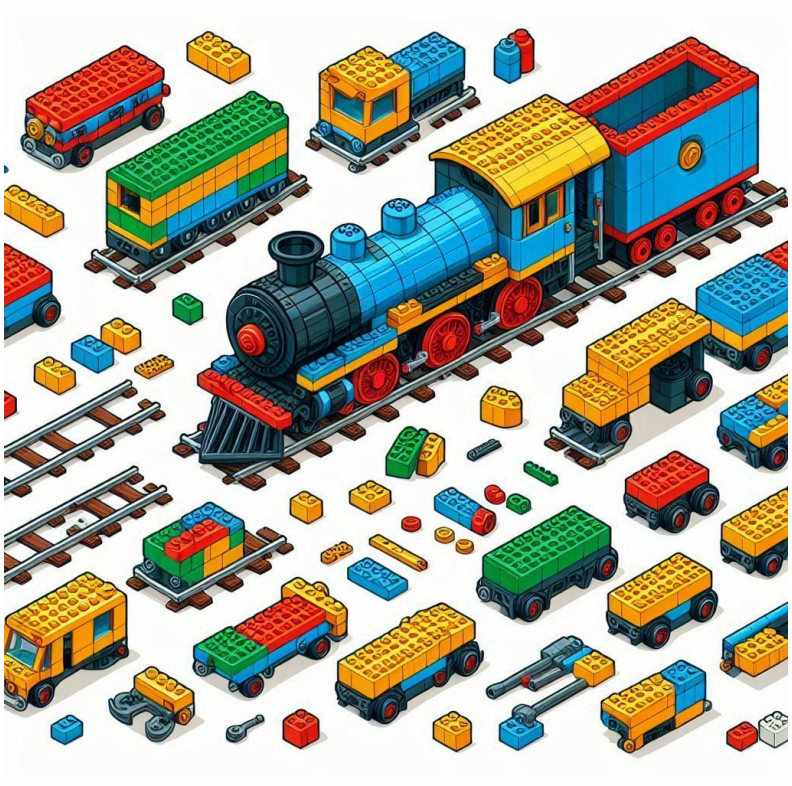


Mobile security model



Mobile security model

App: reinventare la ruota?



Fonte: Microsoft Copilot

Software

- componenti (già sviluppate) + colla
- aspetti positivi di security (es. librerie crittografiche)

Sicurezza delle librerie / componenti

Fonte: Microsoft Copilot

Librerie

- compromesse
- vulnerabili

WIKIPEDIA
The Free Encyclopedia

Search Wikipedia | Search

XZ Utils backdoor

11 languages

Contents hide

(Top)
Background
Mechanism
Response
Remediation
Broader response
Notes
References
External links

Article Talk | Read Edit View history Tools

From Wikipedia, the free encyclopedia

In February 2024, a malicious backdoor was introduced to the Linux build of xz utility within the liblzma library in versions 5.6.0 and 5.6.1 by an account using the name "Jia Tan".[b][2] The backdoor gives an attacker who possesses a specific Ed448 private key remote code execution capabilities on the affected Linux system. The issue has been given the Common Vulnerabilities and Exposures number CVE-2024-3094 and has been assigned a CVSS score of 10.0, the highest possible score.[3][4]

While xz is commonly present in most Linux distributions, at the time of discovery the backdoored version had not yet been widely deployed to production systems, but was present in development versions of major distributions.[5] The backdoor was discovered by the software developer Andres Freund, who announced his findings on 29 March 2024.[6]

XZ Utils backdoor

CVE identifier(s)	CVE-2024-3094
Date discovered	29 March 2024; 7 months ago
Date patched	29 March 2024; 7 months ago[a][1]
Discoverer	Andres Freund
Affected software	xz / liblzma library
Website	tukaani.org/xz-backdoor/

Compromesse: software supply-chain

HACK IN BO® Winter 2024 Edition
23ª EDIZIONE

... oppure vulnerabili

software packaging

Software package format

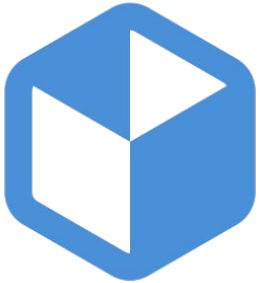
Fonte: Microsoft Copilot

Package manager

- gestione delle dipendenze
- **singola copia** installata di ciascuna libreria

App bundle

AppImage



Flatpak

snap

All-in-one

- **comodo** per gli sviluppatori
- **multiple** (tante) copie della stessa libreria

App bundle: vulnerabilità

Fonte: Microsoft Copilot

- Ciascuna app va **aggiornata singolarmente**
- SVILUPPATORI !!!

Fonte: Google Gemini

~$ ./how.sh

HACK IN BO®
Winter 2024 Edition
23ª EDIZIONE

Concetti chiave

APK

Formato di distribuzione delle applicazioni Android

TPL

Libreria di terze parti inclusa nell'applicazione per l'utilizzo di funzionalità generiche

Vulnerabilità

Debolezza hardware o software che, se sfruttata, può compromettere le informazioni

CVE

Programma di censimento e valutazione di vulnerabilità note

CVE

Anno

Identificativo

CVE-2023-33220 Detail

Description

During the retrofit validation process, the firmware doesn't properly check the boundaries while copying some attributes to check. This allows a stack-based buffer overflow that could lead to a potential Remote Code Execution on the targeted device

Metrics

CVSS Version 4.0 | CVSS Version 3.x | CVSS Version 2.0

NVD enrichment efforts reference publicly available information to associate vector strings. CVSS information contributed by other sources is also displayed.

CVSS 3.x Severity and Vector Strings:

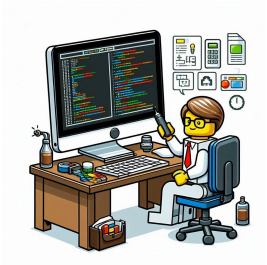
NIST: NVD	**Base Score:** 9.8 CRITICAL	**Vector:** CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
CNA: IDEMIA	**Base Score:** 9.1 CRITICAL	**Vector:** CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N

Caratteristiche

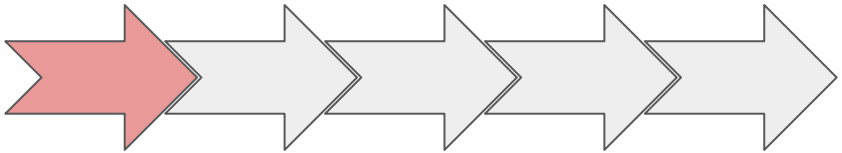
Punteggio

Fonte: NIST

App: dallo sviluppo all'esecuzione



Fonte: Bing Image Creator

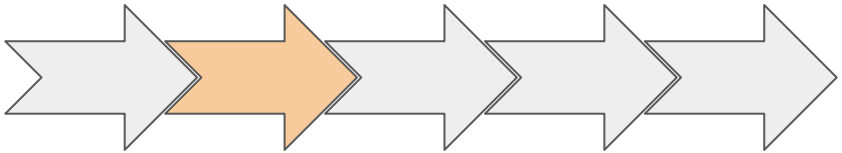


Fonte: Bing Image Creator

Scrittura del codice sorgente,
inclusione e utilizzo delle librerie

App: dallo sviluppo all'esecuzione

Fonte: Bing Image Creator

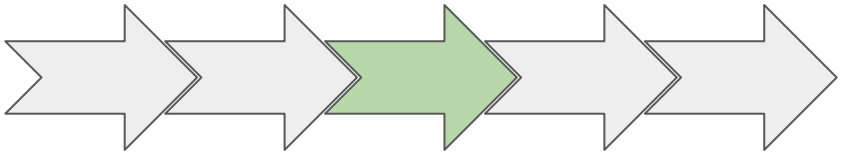


Fonte: Bing Image Creator

Compilazione e generazione dell'APK

App: dallo sviluppo all'esecuzione

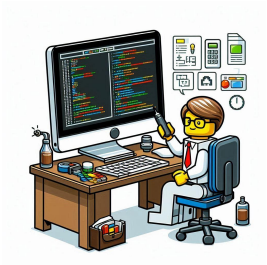
Fonte: Bing Image Creator



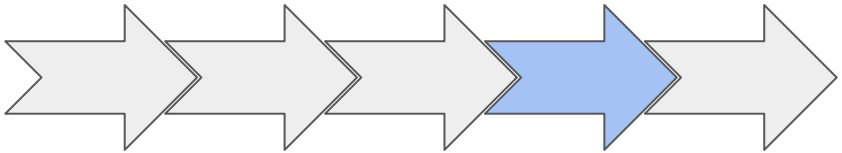
Fonte: Bing Image Creator

Distribuzione dell'APK

App: dallo sviluppo all'esecuzione



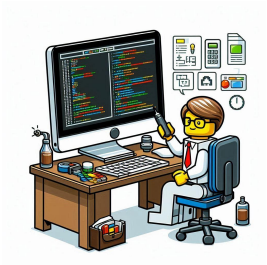
Fonte: Bing Image Creator



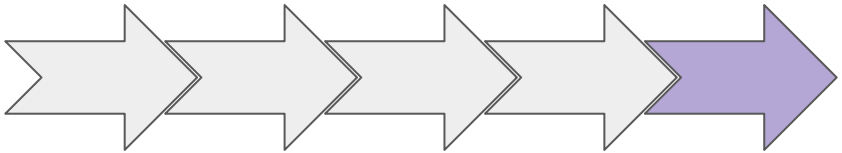
Fonte: Bing Image Creator

Download e installazione sul dispositivo

App: dallo sviluppo all'esecuzione



Fonte: Bing Image Creator



Fonte: Bing Image Creator

Esecuzione dell'applicazione

Collezione degli APK

Utilizzo di un emulatore Android e di *Android Debug Bridge*

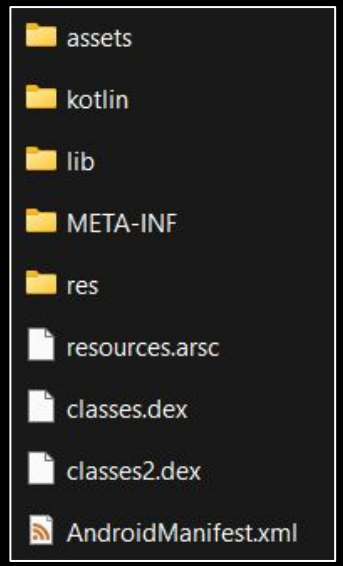
Procedura automatizzabile:

- Ricerca delle app su Google Play Store
- Avvio del download e installazione
- Estrazione dell'APK e merge
- Salvataggio in locale sul PC

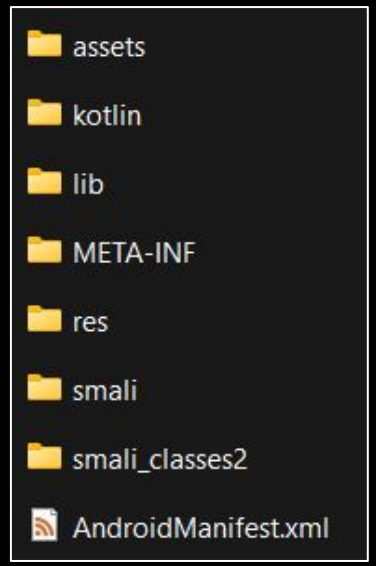
Analisi degli APK

Osservazione del loro contenuto

Unzip



Apktool



Individuazione delle librerie

L'informazione necessaria è la coppia **libreria-versione**

Non esiste un metodo che le riconosca con certezza assoluta, si possono solo fare delle stime

```
└─$ find com.myunibo/smali_classes2/ -mindepth 3
com.myunibo/smali_classes2/com/google/firebase/components/Component.smali
com.myunibo/smali_classes2/com/google/firebase/components/ComponentContainer$-CC.smali
com.myunibo/smali_classes2/com/google/firebase/components/ComponentContainer.smali
com.myunibo/smali_classes2/com/google/firebase/components/ComponentDiscovery$1.smali
com.myunibo/smali_classes2/com/google/firebase/components/ComponentDiscovery$MetadataRegistrarNameRetriever.smali
com.myunibo/smali_classes2/com/google/firebase/components/ComponentDiscovery$RegistrarNameRetriever.smali
com.myunibo/smali_classes2/com/google/firebase/components/ComponentDiscovery.smali
com.myunibo/smali_classes2/com/google/firebase/components/ComponentDiscoveryService.smali
com.myunibo/smali_classes2/net/minidev/asm/BeansAccess.smali
com.myunibo/smali_classes2/net/minidev/asm/BeansAccessBuilder.smali
com.myunibo/smali_classes2/net/minidev/asm/BeansAccessConfig.smali
com.myunibo/smali_classes2/net/minidev/asm/ConvertDate$StringCmpNS.smali
com.myunibo/smali_classes2/net/minidev/asm/ConvertDate.smali
com.myunibo/smali_classes2/net/minidev/asm/DefaultConverter.smali
com.myunibo/smali_classes2/net/minidev/asm/DynamicClassLoader.smali
com.myunibo/smali_classes2/com/google/zxing/qrcode/encoder/ByteMatrix.smali
com.myunibo/smali_classes2/com/google/zxing/qrcode/encoder/Encoder$1.smali
com.myunibo/smali_classes2/com/google/zxing/qrcode/encoder/Encoder.smali
com.myunibo/smali_classes2/com/google/zxing/qrcode/encoder/MaskUtil.smali
com.myunibo/smali_classes2/com/google/zxing/qrcode/encoder/MatrixUtil.smali
com.myunibo/smali_classes2/com/google/zxing/qrcode/encoder/QRCode.smali
com.myunibo/smali_classes2/com/google/zxing/qrcode/QRCodeReader.smali
com.myunibo/smali_classes2/com/google/zxing/qrcode/QRCodeWriter.smali
```

Tecniche di offuscamento

Operazioni sul codice atte a impedire il *reverse engineering* e l'analisi dell'applicazione

Modifiche sul codice ► ridenominazione degli identificatori, *reflection*, ...

Cifratura dei componenti ► stringhe, DEX, applicazione

```
└─$ find it.bancagenerali.mobile/smali_classes*/ -type d
it.bancagenerali.mobile/smali_classes2/ap
it.bancagenerali.mobile/smali_classes2/aq
it.bancagenerali.mobile/smali_classes2/ar
it.bancagenerali.mobile/smali_classes2/as
it.bancagenerali.mobile/smali_classes2/bc
it.bancagenerali.mobile/smali_classes2/bd
it.bancagenerali.mobile/smali_classes2/be
it.bancagenerali.mobile/smali_classes2/bf
it.bancagenerali.mobile/smali_classes2/bg
it.bancagenerali.mobile/smali_classes2/bh
```

Strumenti di rilevazione

- Confronto del codice sorgente dell'applicazione con quello della libreria
- Metodi che superano alcune tecniche di offuscamento
- Fattore di similitudine

```
lib: commons-io-2.5
similarity: 0.5932874516828509

lib: core-3.4.0
similarity: 0.43104514533085964

lib: okhttp-4.2.0
similarity: 0.40401087695445276

lib: okhttp-4.2.1
similarity: 0.4041128484024473

lib: okhttp-4.2.2
similarity: 0.40492694529391776

lib: picasso-2.5.0
similarity: 0.5473217881594845

lib: retrofit-2.5.0
similarity: 0.4694607717863532

lib: retrofit-2.6.0
similarity: 0.6166743224621039

lib: retrofit-2.6.1
similarity: 0.6190799909358713

time: 228s
```

Come funzionano?

- Set di librerie e di APK in input
- Decompilazione
- Soglie di accuratezza per scrematura
- Estrazione delle informazioni e fingerprint
- Similitudine di
  - Firma delle classi
  - Opcode dei metodi
  - Chiamate a funzione

Ricerca delle vulnerabilità

- Scelta della versione con similitudine più alta
- Consultazione dei database di vulnerabilità per ricercare CVE relativi
- Valutazione delle vulnerabilità riscontrate in base ai punteggi e alle caratteristiche

~$ cat results.txt

Set di dati

- Set di librerie comuni con diverse versioni
  - Front-end ▶ caricamento immagini, animazioni, ...
  - Back-end ▶ crittografia, gestione di rete, ...
- Set di APK di vario genere
  - Pubblica amministrazione, banche, social network, ...

50 %

delle applicazioni ha fornito riscontri

Cosa è emerso?

1. Tutte le applicazioni contengono **almeno una** vulnerabilità
2. Molte hanno librerie non aggiornate all'ultima versione
3. Diverse vulnerabilità sono riferite a dipendenze della libreria e non ad essa stessa
4. Alcune vulnerabilità hanno punteggi alti

Alcune considerazioni...

Le app per il settore pubblico sono quelle in cui si riscontrano più librerie non *up-to-date*

La difficoltà di raccoglimento degli APK e i lunghi tempi di calcolo permettono l'uso di un set di dati ristretto

Le app proprietarie sono più protette contro l'analisi statica

~$ echo "whatsnext"

Applicazioni *mobile* : [in]sicurezza e supply chain

Denise Nanni, Gabriele D'Angelo

Alma Mater Studiorum - Università di Bologna

HACK IN BO®
Winter 2024 Edition
23ª EDIZIONE