TU Graz

SCIENCE
PASSION
TECHNOLOGY

Kernel-Sicherheit in realen Szenarien:

Angriffe, Schutzmechanismen und Sicherheitslücken im Linux-Kernel

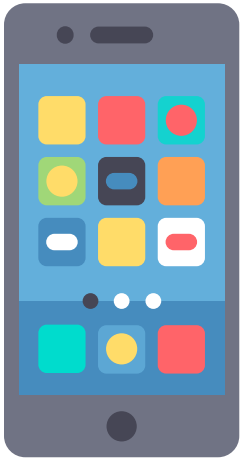
Lukas Maar

11. Mai 2026

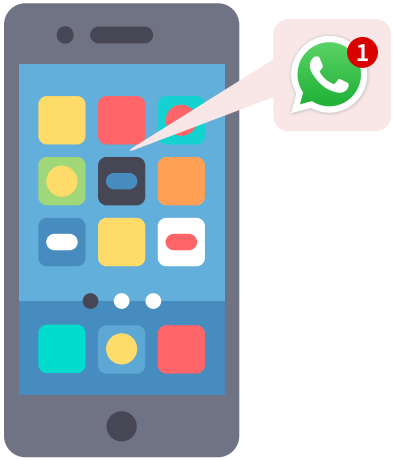
Kolloquim zum GI-Dissertationspreis

› isec.tugraz.at

Vom Alltag zum Kernel-Angriff



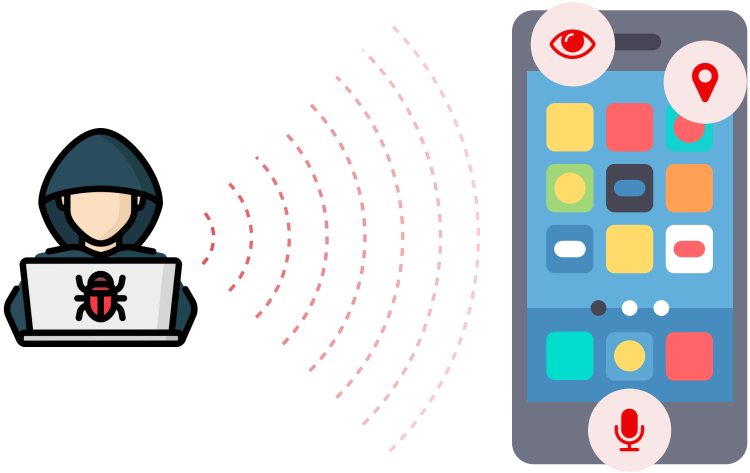
Vom Alltag zum Kernel-Angriff



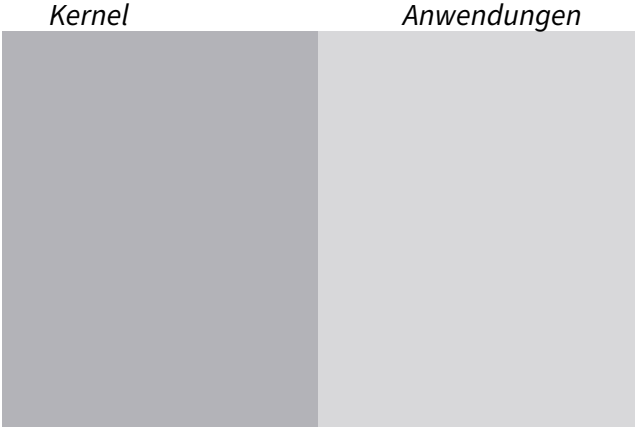
Vom Alltag zum Kernel-Angriff

Vom Alltag zum Kernel-Angriff

Vom Alltag zum Kernel-Angriff



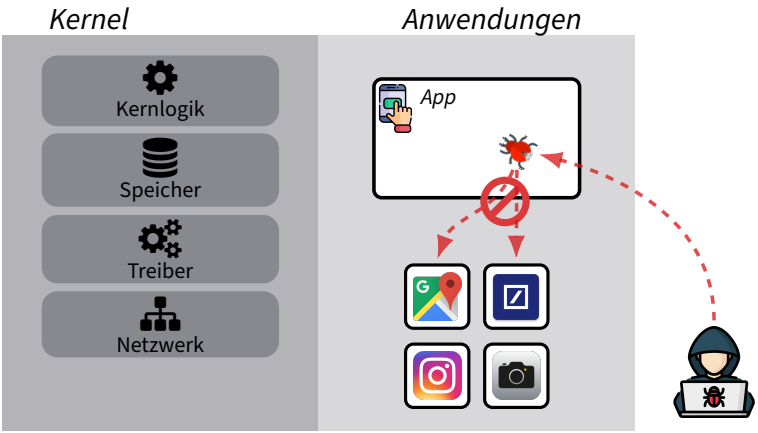
Wie funktionieren Kernel-Angriffe?



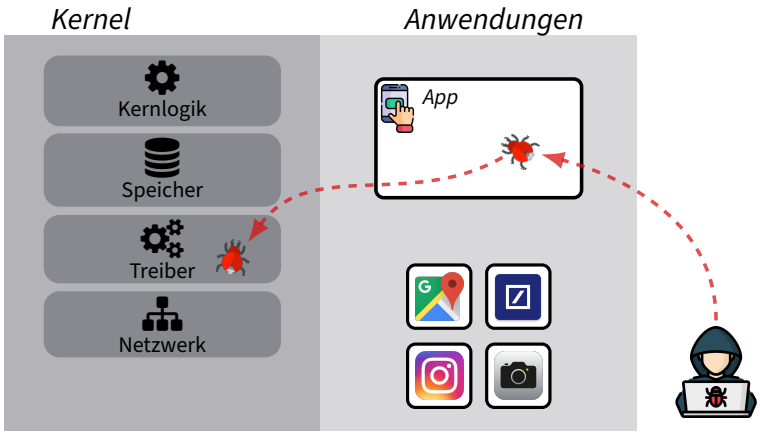
Wie funktionieren Kernel-Angriffe?

Wie funktionieren Kernel-Angriffe?

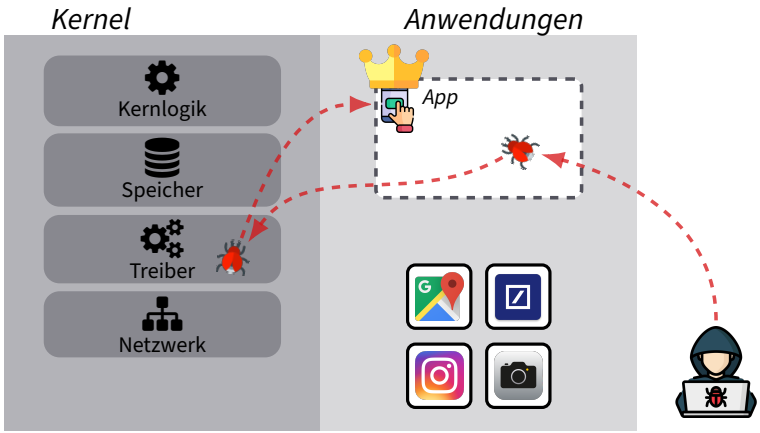
Wie funktionieren Kernel-Angriffe?



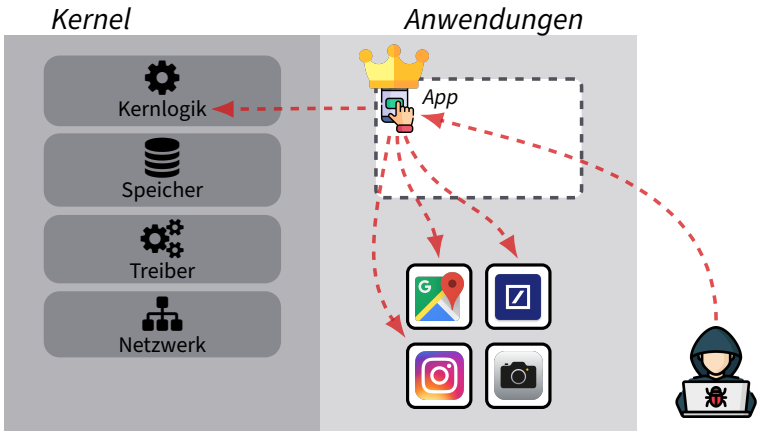
Wie funktionieren Kernel-Angriffe?



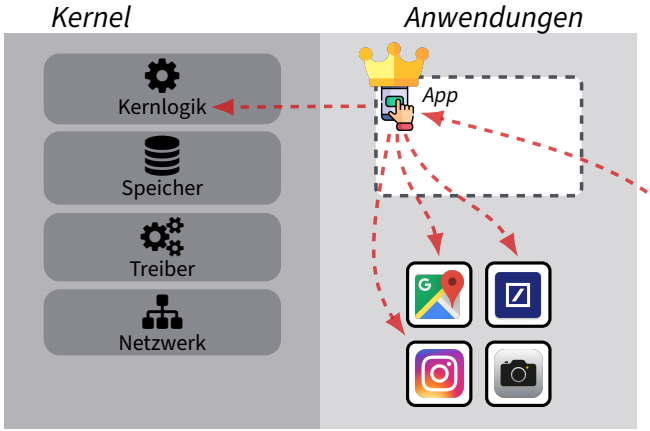
Wie funktionieren Kernel-Angriffe?



Wie funktionieren Kernel-Angriffe?



Wie funktionieren Kernel-Angriffe?



Android | Desktop | Server | Industrie | Krit. Infr. | Energie | Gesundheit | ...

Was Angreifer brauchen

Was Angreifer brauchen

- **Zuverlässige Umsetzbarkeit**

  Angriffe müssen zuverlässig und praktisch umsetzbar sein

Was Angreifer brauchen

- **Zuverlässige Umsetzbarkeit**
  Angriffe müssen zuverlässig und praktisch umsetzbar sein
- **Schutzmechanismen**
  Angriffe müssen trotz Schutzmechanismen funktionieren

Was Angreifer brauchen

- **Zuverlässige Umsetzbarkeit**
  - Angriffe müssen zuverlässig und praktisch umsetzbar sein
- **Schutzmechanismen**
  - Angriffe müssen trotz Schutzmechanismen funktionieren
- **Reale Geräte**
  - Angriffe müssen auf realen Geräten funktionieren

Was ~~Angreifer brauchen~~ Verteidiger verhindern müssen

- **Zuverlässige Umsetzbarkeit**
  - Angriffe müssen zuverlässig und praktisch umsetzbar sein
- **Schutzmechanismen**
  - Angriffe müssen trotz Schutzmechanismen funktionieren
- **Reale Geräte**
  - Angriffe müssen auf realen Geräten funktionieren

Was ~~Angreifer brauchen~~ Verteidiger verhindern müssen

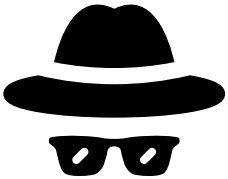
Zuverlässige Umsetzbarkeit

Angriffe

Praktische Kernel-Angriffe so schwer wie möglich machen

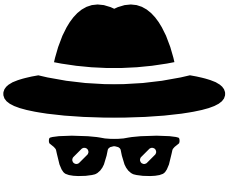
Angriffe müssen auf realen Geräten funktionieren

Meine Hauptbeiträge



- **Zuverlässige Umsetzbarkeit**
- **Schutzmechanismen**
- **Reale Geräte**

Meine Hauptbeiträge

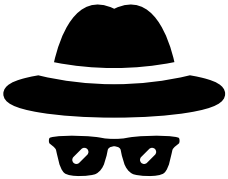


- **Zuverlässige Umsetzbarkeit**
  Drei neue Angriffstechniken:

- **Schutzmechanismen**

- **Reale Geräte**

Meine Hauptbeiträge

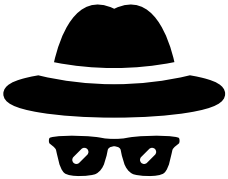


- **Zuverlässige Umsetzbarkeit**
  Drei neue Angriffstechniken:

- **Schutzmechanismen**
  Zwei neue Schutzmechanismen:

- **Reale Geräte**

Meine Hauptbeiträge



Zuverlässige Umsetzbarkeit

Drei neue Angriffstechniken:

Schutzmechanismen

Zwei neue Schutzmechanismen:

Reale Geräte

Zwei neue Android-Analysen:

Zuverlässige Umsetzbarkeit

durch Seitenkanal-unterstützte Angriffstechniken

Interaktives Beispiel

Was sind Seitenkanäle?

Interaktives Beispiel

- Zahl aus $\{0, \ldots, 9\}$ wählen

Was sind Seitenkanäle?

Interaktives Beispiel

- Zahl aus $\{0, \ldots, 9\}$ wählen
- Mit 1337 multiplizieren

Aufzeigen, wenn ihr fertig seid

Was sind Seitenkanäle?

Interaktives Beispiel

- Zahl aus $\{0, \ldots, 9\}$ wählen
- Mit 1337 multiplizieren
  Aufzeigen, wenn ihr fertig seid
- Zufallsauswahl
  Wahrscheinlichkeit 10 %

Was sind Seitenkanäle?

Interaktives Beispiel

- Zahl aus $\{0, \ldots, 9\}$ wählen
- Mit 1337 multiplizieren
  Aufzeigen, wenn ihr fertig seid
- Zufallsauswahl
  Wahrscheinlichkeit 10 %

Beobachtung

Was sind Seitenkanäle?

Interaktives Beispiel

- Zahl aus $\{0, \ldots, 9\}$ wählen
- Mit 1337 multiplizieren
  Aufzeigen, wenn ihr fertig seid
- Zufallsauswahl
  Wahrscheinlichkeit 10 %

Beobachtung

- Schnell aufgezeigt

Was sind Seitenkanäle?

Interaktives Beispiel

- Zahl aus $\{0, \ldots, 9\}$ wählen
- Mit 1337 multiplizieren
  Aufzeigen, wenn ihr fertig seid
- Zufallsauswahl
  Wahrscheinlichkeit 10 %

Beobachtung

- Schnell aufgezeigt
  Wahrscheinlich 0 oder 1

Was sind Seitenkanäle?

Interaktives Beispiel

- Zahl aus $\{0, \ldots, 9\}$ wählen
- Mit 1337 multiplizieren
  Aufzeigen, wenn ihr fertig seid
- Zufallsauswahl
  Wahrscheinlichkeit 10 %

Beobachtung

- Schnell aufgezeigt
  Wahrscheinlich 0 oder 1
  Oder sehr intelligent

Was sind Seitenkanäle?

Interaktives Beispiel

- Zahl aus $\{0, \ldots, 9\}$ wählen
- Mit 1337 multiplizieren
  Aufzeigen, wenn ihr fertig seid
- Zufallsauswahl
  Wahrscheinlichkeit 10 %

Beobachtung

- Schnell aufgezeigt
  Wahrscheinlich 0 oder 1
  Oder sehr intelligent

Über die Laufzeit kann ich etwas über die geheime Eingabe lernen.

Meine Seitenkanäle

Meine Seitenkanäle

Meine Seitenkanäle nutzen Laufzeit, um den internen Systemzustand messbar zu machen.

Meine Seitenkanäle

Meine Seitenkanäle nutzen Laufzeit, um den internen Systemzustand messbar zu machen.

SLUBStick

Meine Seitenkanäle

Meine Seitenkanäle nutzen Laufzeit, um den internen Systemzustand messbar zu machen.

SLUBStick

Allokationen im Kernel-Allokator

Meine Seitenkanäle

Meine Seitenkanäle nutzen Laufzeit, um den internen Systemzustand messbar zu machen.

SLUBStick

Allokationen im Kernel-Allokator

Macht sichtbar:
Speicherwiederverwendung

Meine Seitenkanäle

Meine Seitenkanäle nutzen Laufzeit, um den internen Systemzustand messbar zu machen.

SLUBStick

Allokationen im Kernel-Allokator

Macht sichtbar:
Speicherwiederverwendung

KernelSnitch

Meine Seitenkanäle

Meine Seitenkanäle nutzen Laufzeit, um den internen Systemzustand messbar zu machen.

SLUBStick

- Allokationen im Kernel-Allokator

Macht sichtbar:
Speicherwiederverwendung

KernelSnitch

- Zugriffe auf Kernel-Datenstrukturen

Meine Seitenkanäle

Meine Seitenkanäle nutzen Laufzeit, um den internen Systemzustand messbar zu machen.

SLUBStick

Allokationen im Kernel-Allokator

Macht sichtbar:
Speicherwiederverwendung

KernelSnitch

Zugriffe auf Kernel-Datenstrukturen

Macht sichtbar:
Speicherorte von Kernel-Objekten

Meine Seitenkanäle

Meine Seitenkanäle nutzen Laufzeit, um den internen Systemzustand messbar zu machen.

SLUBStick

- Allokationen im Kernel-Allokator

Macht sichtbar:
Speicherwiederverwendung

KernelSnitch

- Zugriffe auf Kernel-Datenstrukturen

Macht sichtbar:
Speicherorte von Kernel-Objekten

TLB-Seitenkanal

Meine Seitenkanäle

Meine Seitenkanäle nutzen Laufzeit, um den internen Systemzustand messbar zu machen.

SLUBStick

- Allokationen im Kernel-Allokator

Macht sichtbar:
Speicherwiederverwendung

KernelSnitch

- Zugriffe auf Kernel-Datenstrukturen

Macht sichtbar:
Speicherorte von Kernel-Objekten

TLB-Seitenkanal

- Adressübersetzungen in Hardware-Caches

Meine Seitenkanäle

Meine Seitenkanäle nutzen Laufzeit, um den internen Systemzustand messbar zu machen.

SLUBStick

- Allokationen im Kernel-Allokator

Macht sichtbar:
Speicherwiederverwendung

KernelSnitch

- Zugriffe auf Kernel-Datenstrukturen

Macht sichtbar:
Speicherorte von Kernel-Objekten

TLB-Seitenkanal

- Adressübersetzungen in Hardware-Caches

Macht sichtbar:
Speicherorte von Kernel-Objekten

KernelSnitch als Beispiel

Kernel-Angriff

KernelSnitch als Beispiel

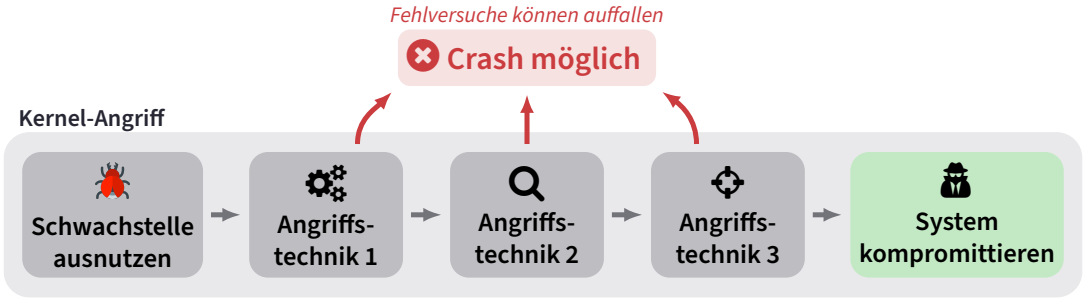
Kernel-Angriff

Schwachstelle ausnutzen

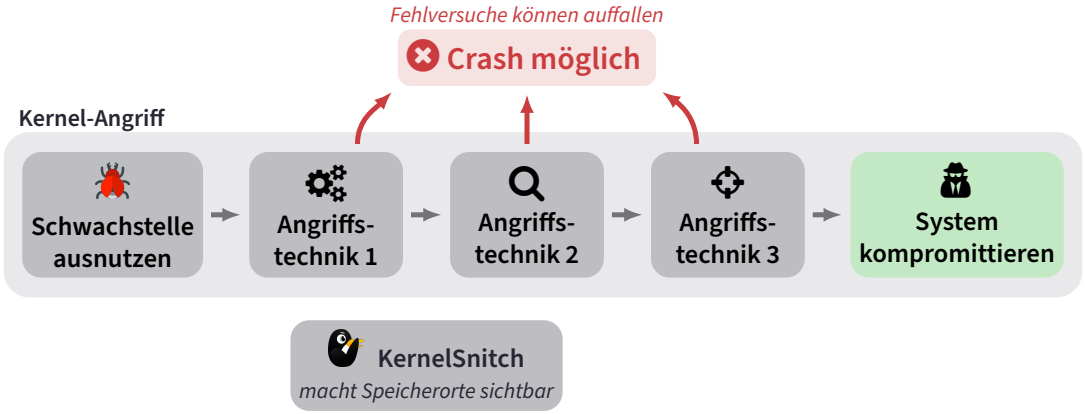
KernelSnitch als Beispiel

KernelSnitch als Beispiel

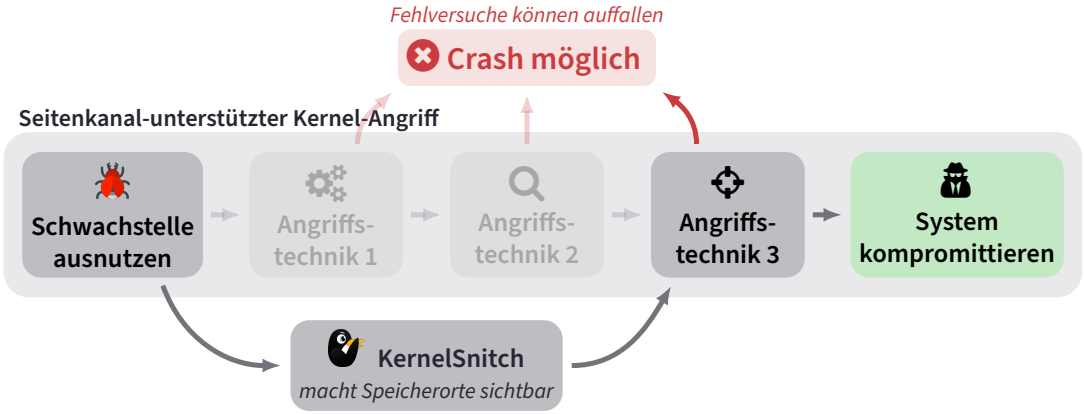
KernelSnitch als Beispiel



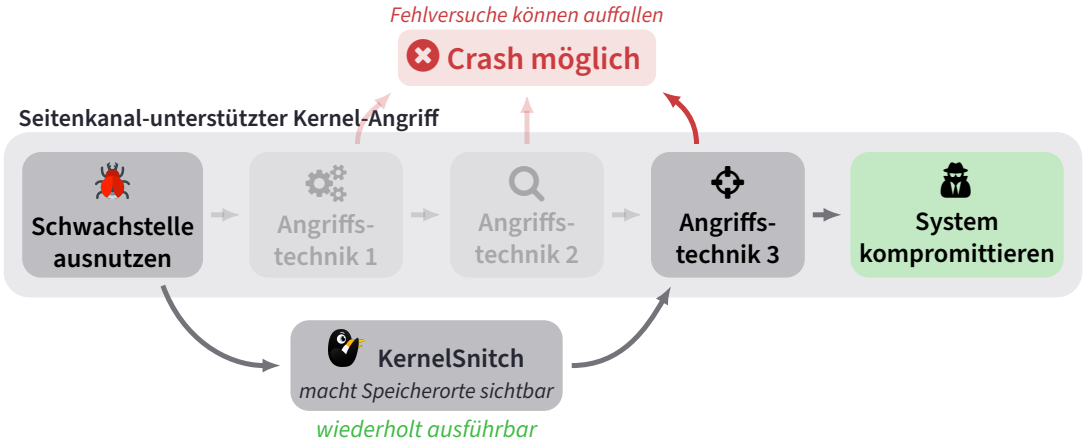
KernelSnitch als Beispiel



KernelSnitch als Beispiel



KernelSnitch als Beispiel

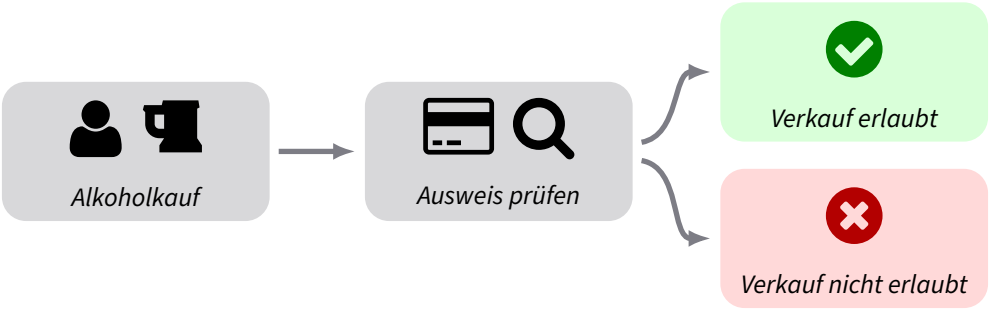


Schutzmechanismen

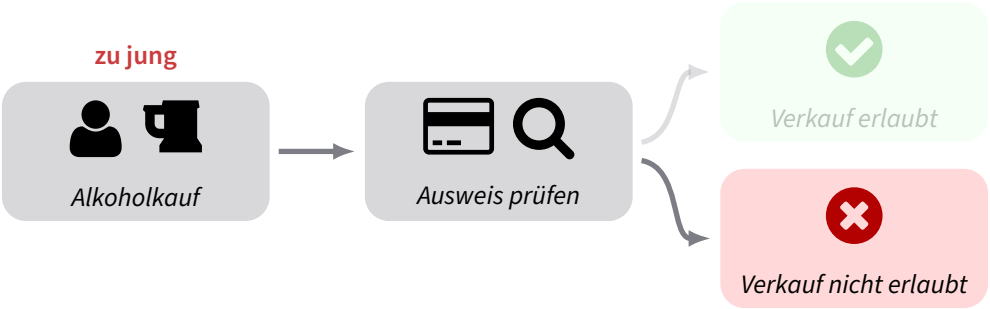
Sicherheitskritische Kernel-Daten schützen

Kernel-Angriffe: Ablauf oder Daten manipulieren

Kernel-Angriffe: Ablauf oder Daten manipulieren



Kernel-Angriffe: Ablauf oder Daten manipulieren



Kernel-Angriffe: Ablauf oder Daten manipulieren

Kontrollfluss-Manipulationsangriff:

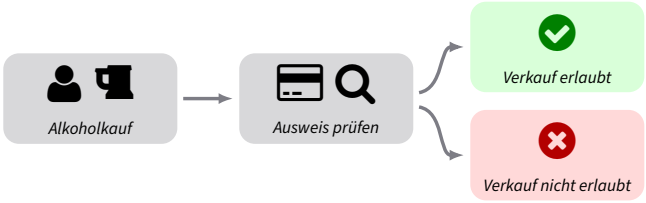
Ausweisprüfung wird übersprungen

Kernel-Angriffe: Ablauf oder Daten manipulieren

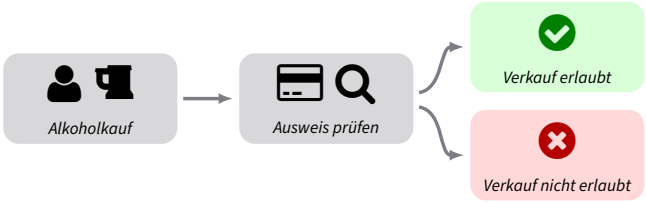
Datenorientierter Angriff:

Ausweisdaten werden manipuliert

Meine Beiträge

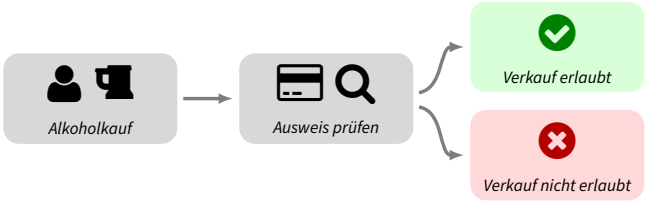


Meine Beiträge



DOPE

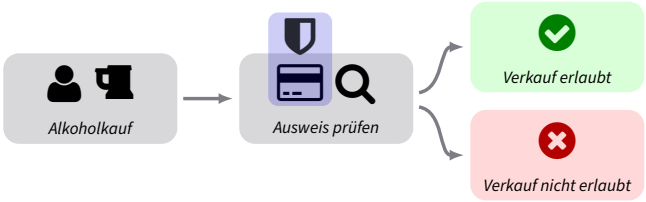
Meine Beiträge



DOPE

Verwendet Memory Protection Keys

Meine Beiträge

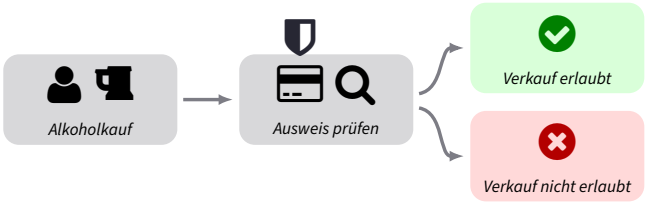


DOPE

Verwendet Memory Protection Keys

Schützt:
Sicherheitskritische Kernel-Daten
wie Credentials und Page Tables

Meine Beiträge



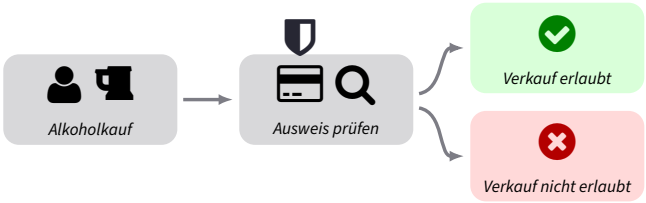
DOPE

Verwendet Memory Protection Keys

Schützt:
Sicherheitskritische Kernel-Daten wie Credentials und Page Tables

HEK-CFI

Meine Beiträge



DOPE

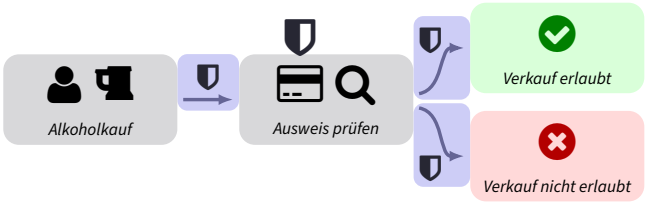
Verwendet Memory Protection Keys

Schützt:
Sicherheitskritische Kernel-Daten wie Credentials und Page Tables

HEK-CFI

Verwendet Intel CET

Meine Beiträge



DOPE

Verwendet Memory Protection Keys

Schützt:
Sicherheitskritische Kernel-Daten wie Credentials und Page Tables

HEK-CFI

Verwendet Intel CET

Schützt:
Kontrollflussdaten wie Rückkehradressen, Code-Zeiger und Thread-State


STOP

Reale Geräte

Großflächige Analysen zeigen kritische Lücken

Wie gut geschützt sind reale Android-Geräte?

Kernel-Angriff

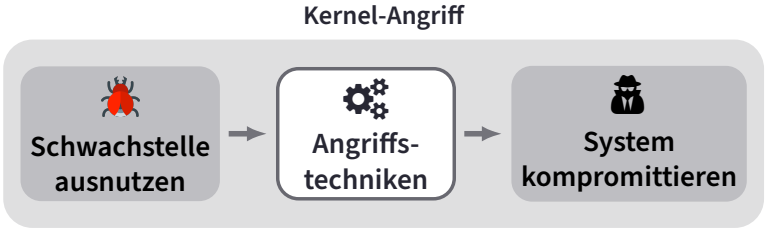
Wie gut geschützt sind reale Android-Geräte?

Kernel-Angriff

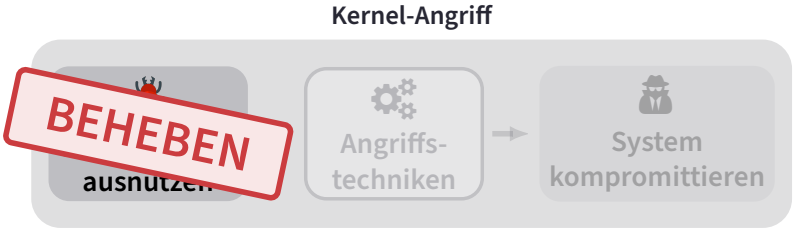
Schwachstelle ausnutzen

Wie gut geschützt sind reale Android-Geräte?

Wie gut geschützt sind reale Android-Geräte?



Wie gut geschützt sind reale Android-Geräte?



Behebungen:

Konkrete Schwachstellen beheben

Wie gut geschützt sind reale Android-Geräte?

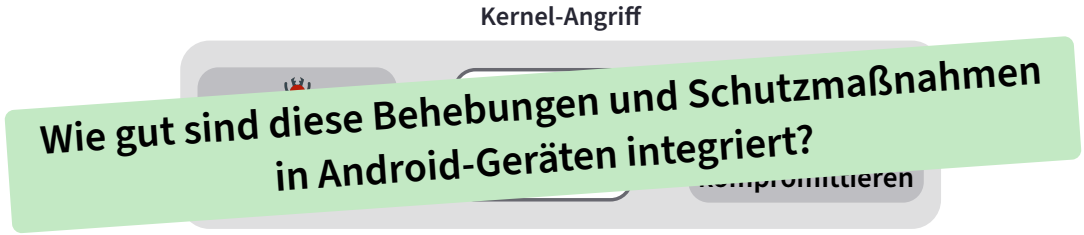
Behebungen:

Konkrete Schwachstellen beheben

Proaktive Schutzmaßnahmen:

Angriffstechniken verhindern

Wie gut geschützt sind reale Android-Geräte?



Behebungen:
Konkrete Schwachstellen beheben

Proaktive Schutzmaßnahmen:
Angriffstechniken verhindern

Wie gut geschützt sind reale Android-Geräte?

Behebungen:
Konkrete Schwachstellen beheben

Proaktive Schutzmaßnahmen:
Angriffstechniken verhindern

Doom of Device Drivers

Wie gut geschützt sind reale Android-Geräte?

Behebungen:
Konkrete Schwachstellen beheben

Doom of Device Drivers

Proaktive Schutzmaßnahmen:
Angriffstechniken verhindern

Defects-in-Depth

Defects-in-Depth: Wie gut sind Schutzmechanismen integriert?

Defects-in-Depth: Wie gut sind Schutzmechanismen integriert?

Kernel-Angriffe	Bekannte und verfügbare Schutzmechanismen
CVE-2019-2215	
CVE-2019-2025	
CVE-2020-0030	
CVE-2021-1968,-1969,-1940	
CVE-2021-0920	
CVE-2021-1905	
CVE-2022-22265	
CVE-2021-25369,-25370	
CVE-2016-3809,-2021-0399	
CVE-2022-20409	
CVE-2023-21400	
CVE-2022-28350	
CVE-2020-29661	
CVE-2021-22600	
CVE-2020-0423	
CVE-2022-22057	
CVE-2023-26083,-0266	
CVE-2020-0041	
CVE-2019-2205	
CVE-2019-2025	
CVE-2020-3680	
CVE-2022-20421	
CVE-2022-0847	
CVE-2021-4154	
CVE-2021-38001	
NO_NUMBER (∼2021)	

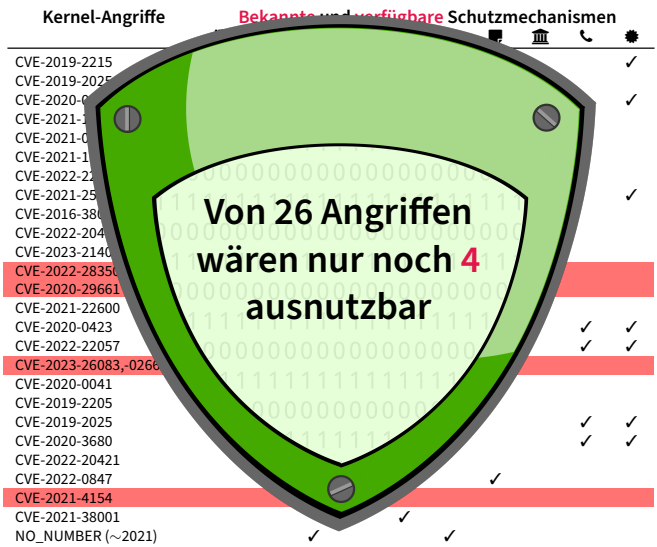
Defects-in-Depth: Wie gut sind Schutzmechanismen integriert?

Kernel-Angriffe	Bekannte und verfügbare Schutzmechanismen									
CVE-2019-2215	✓		✓							✓
CVE-2019-2025	✓		✓							
CVE-2020-0030	✓		✓							✓
CVE-2021-1968,-1969,-1940				✓	✓					
CVE-2021-0920	✓		✓							
CVE-2021-1905				✓	✓					
CVE-2022-22265			✓							
CVE-2021-25369,-25370			✓							✓
CVE-2016-3809,-2021-0399			✓	✓	✓					
CVE-2022-20409			✓							
CVE-2023-21400			✓							
CVE-2022-28350										
CVE-2020-29661										
CVE-2021-22600			✓							
CVE-2020-0423	✓								✓	✓
CVE-2022-22057						✓			✓	✓
CVE-2023-26083,-0266										
CVE-2020-0041	✓		✓							
CVE-2019-2205	✓		✓							
CVE-2019-2025	✓		✓						✓	✓
CVE-2020-3680	✓		✓						✓	✓
CVE-2022-20421			✓							
CVE-2022-0847							✓			
CVE-2021-4154										
CVE-2021-38001				✓	✓					
NO_NUMBER (∼2021)			✓			✓				

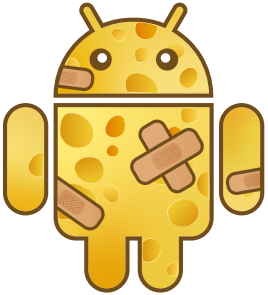
Defects-in-Depth: Wie gut sind Schutzmechanismen integriert?

Kernel-Angriffe	Bekannte und verfügbare Schutzmechanismen									
CVE-2019-2215	✓		✓							✓
CVE-2019-2025	✓		✓							
CVE-2020-0030	✓		✓							✓
CVE-2021-1968,-1969,-1940				✓	✓					
CVE-2021-0920	✓		✓							
CVE-2021-1905				✓	✓					
CVE-2022-22265			✓							
CVE-2021-25369,-25370			✓							✓
CVE-2016-3809,-2021-0399			✓	✓	✓					
CVE-2022-20409			✓							
CVE-2023-21400			✓							
CVE-2022-28350										
CVE-2020-29661										
CVE-2021-22600			✓							
CVE-2020-0423	✓								✓	✓
CVE-2022-22057						✓			✓	✓
CVE-2023-26083,-0266										
CVE-2020-0041	✓		✓							
CVE-2019-2205	✓		✓							
CVE-2019-2025	✓		✓						✓	✓
CVE-2020-3680	✓		✓						✓	✓
CVE-2022-20421			✓							
CVE-2022-0847							✓			
CVE-2021-4154										
CVE-2021-38001				✓	✓					
NO_NUMBER (∼2021)			✓			✓				

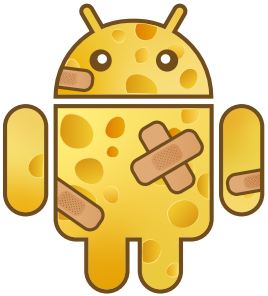
Defects-in-Depth: Wie gut sind Schutzmechanismen integriert?



Defects-in-Depth: Ergebnisse

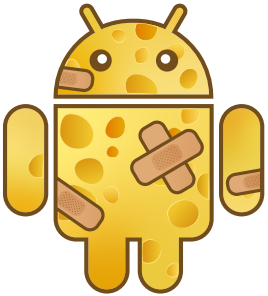


Defects-in-Depth: Ergebnisse



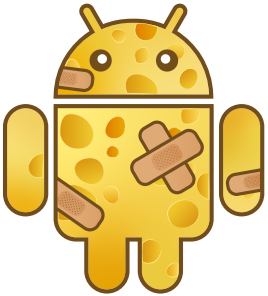
- **Noch mögliche Angriffe mit allen verfügbaren Schutzmechanismen:**
  - *Theoretischer Referenzwert:* **4** von 26

Defects-in-Depth: Ergebnisse



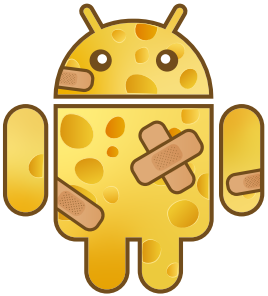
- **Noch mögliche Angriffe mit allen verfügbaren Schutzmechanismen:**
  - *Theoretischer Referenzwert:* **4** von 26
- **Evaluierung:**
  - 994 Android-Geräte
  - 10 Hersteller

Defects-in-Depth: Ergebnisse



- **Noch mögliche Angriffe mit allen verfügbaren Schutzmechanismen:**
  - *Theoretischer Referenzwert:* **4** von 26
- **Evaluierung:**
  - 994 Android-Geräte
  - 10 Hersteller
  - *Realer Durchschnitt:* **15,5** von 26

Defects-in-Depth: Ergebnisse



- **Noch mögliche Angriffe mit allen verfügbaren Schutzmechanismen:**
  - *Theoretischer Referenzwert:* **4** von 26
- **Evaluierung:**
  - 994 Android-Geräte
  - 10 Hersteller
  - *Realer Durchschnitt:* **15,5** von 26
- **Auffällige Ergebnisse:**
  - *Bester:* Google mit **11,8** von 26
  - *Schlechtester:* Fairphone mit **18,5** von 26
  - *Größter Marktanteil:* Samsung mit **16,1** von 26

Doom of Device Drivers: Kommen Behebungen bei Geräten an?

Behebungen bekannter Schwachstellen: Kommen sie bei Android-Geräten an?

Behebungen bekannter Schwachstellen: Kommen sie bei Android-Geräten an?

Analyse:

- 50 bekannte Kernel-Schwachstellen

Behebungen bekannter Schwachstellen: Kommen sie bei Android-Geräten an?

Analyse:

- 50 bekannte Kernel-Schwachstellen

Evaluierung:

- 131 Android-Geräte
- 7 Hersteller

Behebungen bekannter Schwachstellen: Kommen sie bei Android-Geräten an?

Analyse:

- 50 bekannte Kernel-Schwachstellen

Evaluierung:

- 131 Android-Geräte
- 7 Hersteller

Ähnliches Bild: Behebungen fehlen oft

Ergebnis	Anteil
Mind. eine kritische Schwachstelle	59,1 %
Mind. eine Schwachstelle	61,4 %

Behebungen bekannter Schwachstellen: Kommen sie bei Android-Geräten an?

+ Defects-in-Depth:

Öffentlich zugängige Schwachstellen und *bekannte* Angriffstechniken reichen aus, um einen Großteil der Android-Geräte zu kompromittieren

Ähnliches Bild: Behebungen fehlen oft

Evaluierung:

- 131 Android-Geräte
- 7 Hersteller

Mind. eine Schwachstelle


STOP

Beitrag und Wirkung

Beitrag und Wirkung

Beitrag und Wirkung

Wissenschaftliche Beiträge:

- 7 Erstautorarbeiten
  - *5 auf Tier-1-Sicherheits-Konferenzen*
  - *2 auf Tier-2-Sicherheits-Konferenzen*
- 12 Veröffentlichungen insgesamt

Beitrag und Wirkung

Wissenschaftliche Beiträge:

- 7 Erstautorarbeiten
  - *5 auf Tier-1-Sicherheits-Konferenzen*
  - *2 auf Tier-2-Sicherheits-Konferenzen*
- 12 Veröffentlichungen insgesamt

Praktische Wirkung:

- Verantwortungsvolle Offenlegung von mehr als 10 Schwachstellen
  - *Linux-Kernel, Android-Kernel und Android-Applikationen*
  - *Betroffen sind u.a. Systeme auf Milliarden Geräten*
- Bestätigt durch das Linux Kernel Security Team und Industriepartner
- 3 Black-Hat- und 1 NullCon-Vortrag

Beitrag und Wirkung

Wissenschaftliche Beiträge:

- 7 Erstautorarbeiten
  - *5 auf Tier-1-Sicherheits-Konferenzen*
  - *2 auf Tier-2-Sicherheits-Konferenzen*
- 12 Veröffentlichungen insgesamt

Praktische Wirkung:

- Verantwortungsvolle Offenlegung von mehr als 10 Schwachstellen
  - *Linux-Kernel, Android-Kernel und Android-Applikationen*
  - *Betroffen sind u.a. Systeme auf Milliarden Geräten*
- Bestätigt durch das Linux Kernel Security Team und Industriepartner
- 3 Black-Hat- und 1 NullCon-Vortrag

Awards:

- Nominiert für den Pwnie Award
- CSAW Applied Research Competition Finalist


STOP