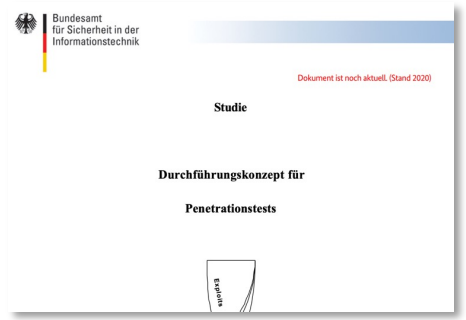
Best practices bei der Beauftragung von Pentests

Kriterien gemäß DiGAV

- **Penetrationstests**: Mit dem DVPMG wurde diese Anforderung für alle DiGA in die DiGAV aufgenommen. Die Sicherheit der Daten über den gesamten Anwendungsprozess und alle erdenklichen Nutzungsszenarien hinweg sicherzustellen, ist essentielle Anforderung an DiGA. Penetrationstests ermöglichen die Nachbildung möglicher Angriffsmuster und können so dazu beitragen, Sicherheitslücken aufzudecken. Für die Produktversion, für die eine Aufnahme in das DiGA-Verzeichnis beantragt wird, muss für alle Komponenten ein Penetrationstest durchgeführt worden sein. Diese Tests sind anforderungsbezogen zu wiederholen, z. B. wenn neue Schnittstellen in das Internet hinzukommen. Als Basis für die Testkonzeption sind das Durchführungskonzept für Penetrationstests des BSI sowie die jeweils aktuellen OWASP Top-10 Sicherheitsrisiken heranzuziehen. Dem BfArM muss auf Verlangen ein Nachweis über die Durchführung der entsprechenden Tests und die Behebung der dabei gefundenen Schwachstellen vorgelegt werden. Das BSI hat zu Thema Penetrationstest den Praxis-Leitfaden „*Praxis-Leitfaden: IT-Sicherheits-Penetrationstest*“ veröffentlicht.

OWASP TOP10

https://owasp.org/Top10/



https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Studien/Penetrationstest/penetrationstest.pdf?__blob=publicationFile&v=3

https://www.bfarm.de/SharedDocs/Downloads/DE/Medizinprodukte/diga_leitfaden.html

Checkliste vor Beauftragung eines Pentesters

1. Kompetenz und Erfahrung des Anbieters prüfen:

- Kann der Anbieter frühere Erfahrungen und **Referenzen** zur Durchführung von Pentests bei **DiGA** / bei der Verarbeitung von Gesundheitsdaten vorweisen?
- Kann der Anbieter beurteilen, inwieweit die richtigen / alle geforderten Security Measures vorhanden sind, beispielsweise im DiGA-Kontext?
- Erfahrung mit der verwendeten Entwicklungsumgebung/Programmiersprache, Web/Mobile/Android/iOS?
- Zertifizierungen des Unternehmens sind in der Regel wenig aussagekräftig
- Wichtiger ist die Kompetenz der Mitarbeiter: aktives Mitglied der Security Community? Publikationen?

2. Anpassung an spezifische Anforderungen:

- **White-Box-Test:** sollte immer die Empfehlung des Anbieters sein! Benötigt Zugang zu Ressourcen (z.B. Quellcode, Systemdokumentation)
- **Menschliche Analyse:** Trotz der Verwendung automatisierter Tools sollte der Test eine beträchtliche menschliche Analyse enthalten, um komplexe Sicherheitslücken zu identifizieren
- Abstimmung zur Vorgehensweise: OWASP, BSI etc.?

Checkliste vor Beauftragung eines Pentesters

3. Testumfang definieren:

- **Abdeckung:** Der Test sollte alle relevanten Systeme, Netzwerke und Anwendungen abdecken, einschließlich Schnittstellen und ggf. Cloud-Dienste, nicht nur AppSec
- **Grenzen** des Tests (z.B. keine Denial-of-Service-Attacken) festgelegt?
- Ggf. weitere Angriffsvektoren einschließen, z.B. Social Engineering, Phishing, (theoretische) Denial of Service-Attacken, und Injection-Angriffe.

4. Berichterstattung:

- Wird der Anbieter einen umfassenden Bericht erstellen, der sowohl technische Details als auch eine Zusammenfassung auf Managementebene und klare Empfehlungen zur Behebung der identifizierten Schwachstellen enthält?
- Welche Angaben werden zu Schwachstellen gemacht? Das "Common Vulnerability Scoring System" (CVSS) ist aufwändig und nicht immer hilfreich. Eine Erläuterung von Eintrittswahrscheinlichkeit und Schadensszenarien zahlt sich auch auf eine interne Betrachtung durch das Risikomanagement mehr aus.

Checkliste vor Beauftragung eines Pentesters

5. Rechtliche und ethische Standards:

- Hat der Anbieter einen Prozess, um alle notwendigen Genehmigungen vor Beginn des Pentests einzuholen?
- Wird der Anbieter vertrauliche Informationen und Zugriffsdaten sicher behandeln und nach Abschluss des Tests ordnungsgemäß entsorgen (NDA)?

6. Nachtest-Unterstützung:

- Bietet der Anbieter Unterstützung nach dem Test an, um bei der Interpretation der Testergebnisse und der Behebung der identifizierten Schwachstellen zu helfen?

7. Regelmäßige Tests:

- Haben Sie einen Plan für regelmäßige Pentests, um auf neue Sicherheitslücken und Bedrohungen zu reagieren?
- Wird ein erneuter (teilweiser) Test bei Änderungen in der Software angeboten?

Was günstige Vulnerabilitäts-Scans/Pentests oft (nur) testen

1. **Betriebssystem-Schwachstellen**, die bereits bekannt sind
2. **Bekannte Vulnerabilitäten** in installierten Softwarepaketen
3. **Fehlkonfigurationen**, also als unsicher klassifizierte System- oder Anwendungseinstellungen
4. **Veraltete Software**
5. **Fehlende Patches** und dadurch bestehende Sicherheitslücken
6. **Unsichere (Netzwerk-)Protokolle**
7. **Offene Ports** zur Reduktion von Eintrittspunkten ins System

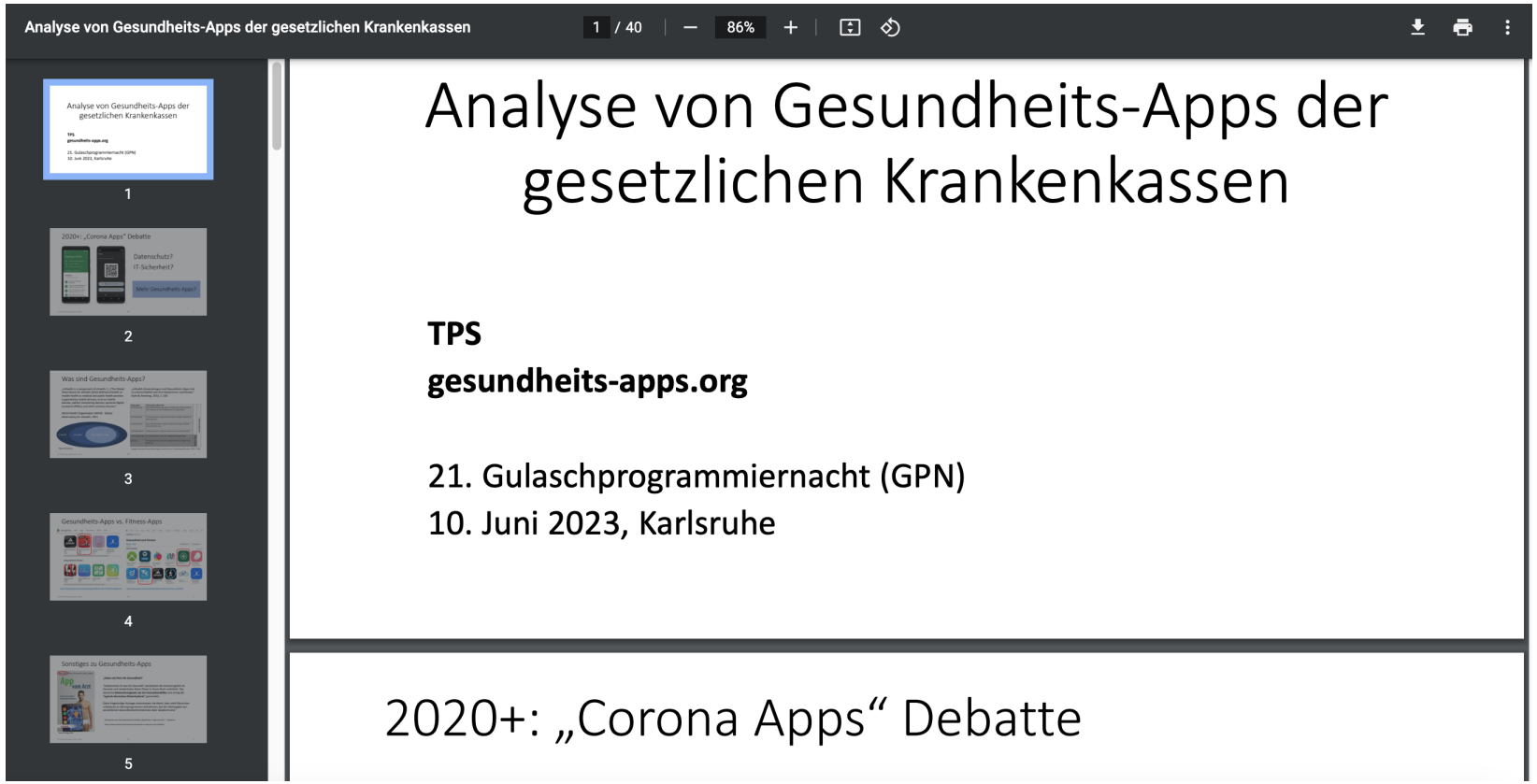
Schwachstellen unzureichender Tests

1. **Fehlende Anpassung:** potenziell kritische Teile der Anwendung wurden nicht ausreichend getestet.
2. **Fehlalarme (False Positives):** Erzeugt unnötige Arbeit für das Unternehmen und zu Verwirrung über die tatsächliche Sicherheitslage.
3. **Verpasste Schwachstellen:** ohne eine menschliche Planung und Überprüfung des Tests können echte Sicherheitsprobleme übersehen werden.
4. **Schlechte Kommunikation:** oft gibt es bei billigen Tests nur den Ausdruck des Outputs des Tools, ohne zusätzliche Erklärungen oder Kontext. Dies machte es für das Unternehmen schwierig zu verstehen, was die Probleme tatsächlich waren und wie sie behoben werden könnten.

Security by Design

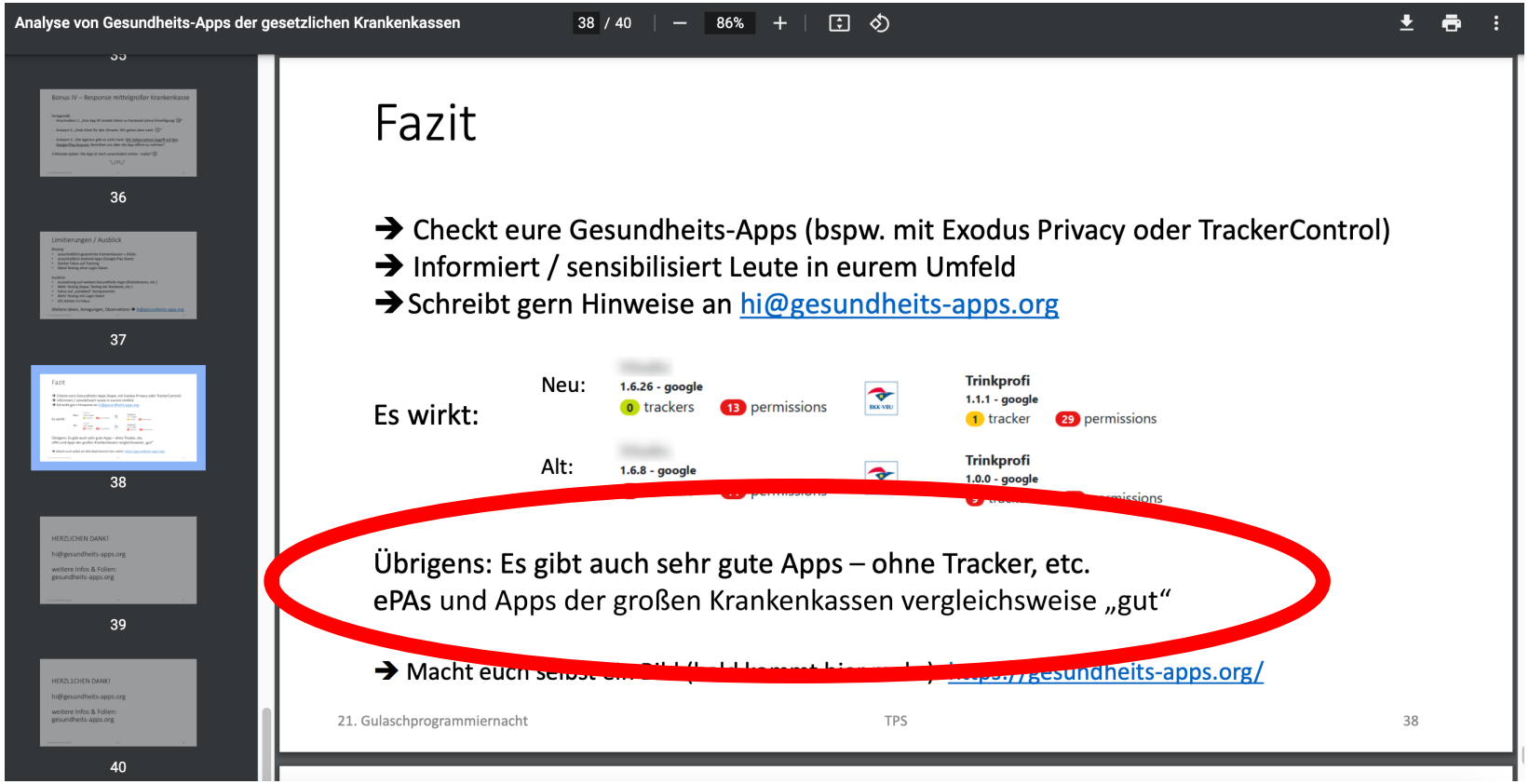
1. **Minimale Datenerhebung:** Die App erfasst nur die absolut notwendigen Daten und nichts mehr. Je weniger Daten erfasst werden, desto weniger können im Falle eines Datenschutzvorfalls kompromittiert werden.
2. **Verschlüsselung:** Alle Daten, die von der App erfasst oder übertragen werden, werden verschlüsselt, sowohl während der Übertragung (Transportverschlüsselung) als auch beim Speichern auf dem Gerät (Datenspeicherverschlüsselung).
3. **Zugriffskontrollen:** Die App implementiert strenge Zugriffskontrollen, um sicherzustellen, dass nur autorisierte Benutzer auf sensible Daten zugreifen können. Dies kann durch Passwörter, biometrische Authentifizierung oder Zwei-Faktor-Authentifizierung erreicht werden.
4. **Regelmäßige Updates und Patches:** Die App wird regelmäßig aktualisiert, um bekannte Sicherheitslücken zu schließen und auf neue Bedrohungen zu reagieren.
5. **Offene Schnittstellen (APIs) schützen:** Gesundheitsapps verwenden oft APIs, um Daten auszutauschen. Diese APIs müssen geschützt und auf Sicherheit überprüft werden, um sicherzustellen, dass sie nicht als Einfallstor für Angriffe dienen.
6. **Sichere Software-Entwicklungspraktiken:** Bei der Entwicklung der App werden sichere Programmierpraktiken verwendet, um Sicherheitslücken zu vermeiden. Dies kann die Verwendung von sicheren Programmierbibliotheken, regelmäßige Codeüberprüfungen und automatisierte Sicherheitstests umfassen.
7. **Detailliertes Monitoring und Logging:** Im Falle eines Angriffs erfolgt eine umgehende Benachrichtigung und bei erfolgreichem Eindringen kann eine adäquate forensische Untersuchung erfolgen.
8. **Incident Response Plan:** Ein gut durchdachter Incident Response Plan ist integraler Bestandteil des App-Designs. Das erlaubt schnelles und effektives Handeln im Falle einer Sicherheitsverletzung.
9. **Penetrationstests und Sicherheitsüberprüfungen:** Vor der Veröffentlichung der App und auch danach in regelmäßigen Abständen werden Penetrationstests und Sicherheitsüberprüfungen durchgeführt, um potenzielle Sicherheitslücken zu identifizieren und zu beheben.

Gulaschprogrammiernacht 2023



https://gesundheits-apps.org/gpn.pdf

Gulaschprogrammiernacht 2023



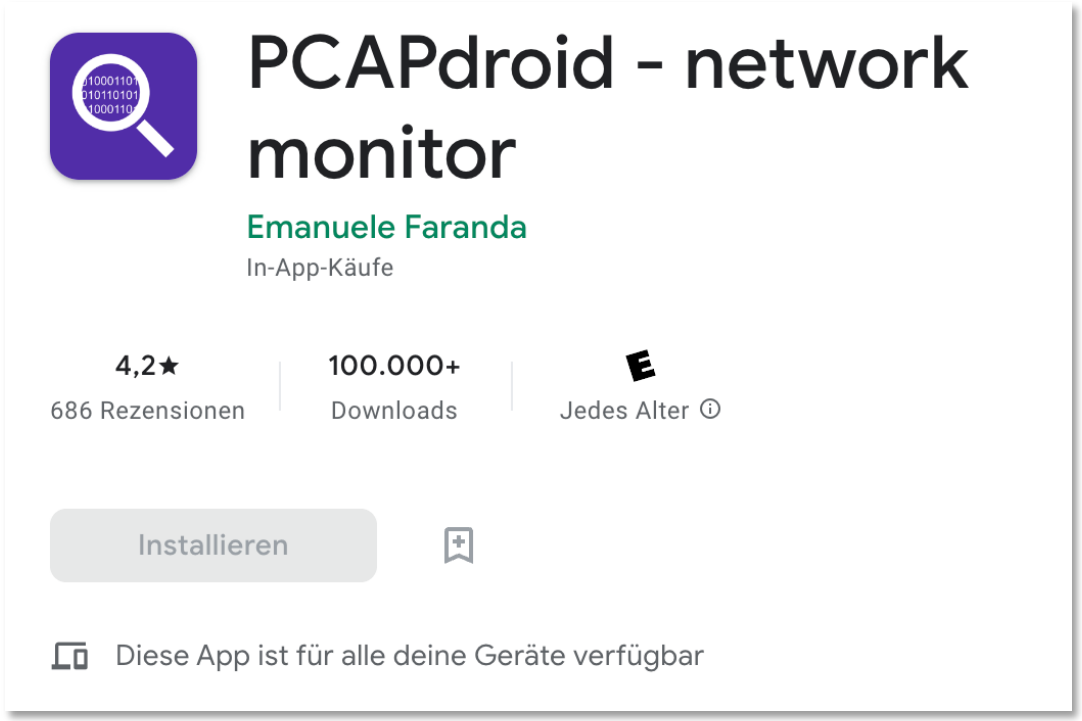
https://gesundheits-apps.org/gpn.pdf

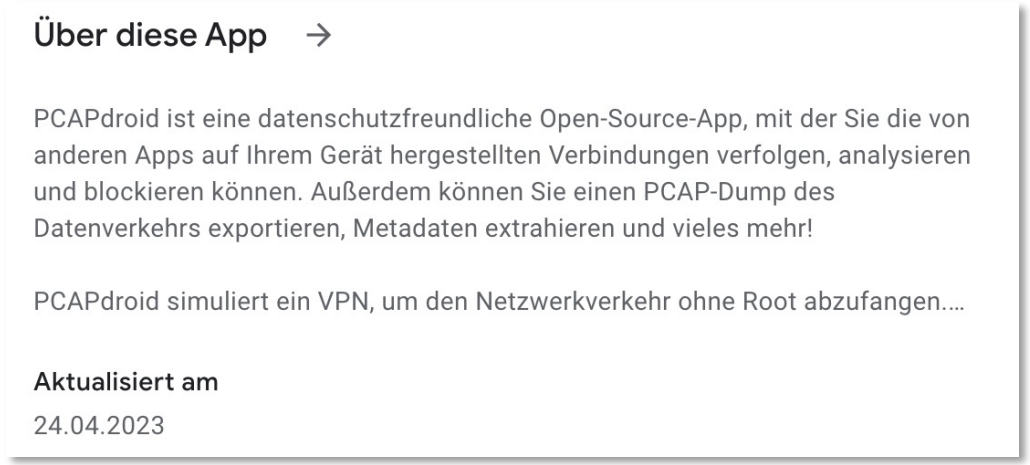
Gulaschprogrammiernacht 2023

Direkte Kontaktaufnahme (Appstart	Erkannte Tracker Bibliotheken
firebaseinstallations.googleapis.com	Facebook Analytics, Facebook Login, Google Firebase Analyti
-	Facebook Flipper, Google Firebase Analytics
-	Google Firebase Analytics
d2atvv2ifireay.cloudfront.net	-
-	Batch, Google CrashLytics, Snowplow
ws.batch.com	Batch, Google CrashLytics, Snowplow
-	-
-	-
-	Google Firebase Analytics
-	-
nta -	-
-	-
-	-
api.mixpanel.com	MixPanel
-	-
-	Google Firebase Analytics, Huawei Mobile Services (HMS) Co
-	Facebook Flipper, Google Firebase Analytics, Matomo (Piwik)
-	Facebook Flipper, Google Firebase Analytics, Matomo (Piwik)
-	Facebook Flipper, Google Firebase Analytics, Matomo (Piwik)
-	Facebook Flipper, Google Firebase Analytics, Matomo (Piwik)
-	Facebook Flipper, Google Firebase Analytics, Matomo (Piwik)
googleapis.com	-
firebaseinstallations.googleapis.com, fire	Facebook Analytics, Facebook Login, Facebook Share, Google
googleapis.com	Google Firebase Analytics
-	-
-	Matomo (Piwik)
googleapis.com	Matomo (Piwik)

https://gesundheits-apps.org/diga/DiGA.htm

Beispiel: eigene DiGA prüfen





Es gibt viele Apps in diese Richtung, die hier genannte ist ein Beispiel

https://play.google.com/store/apps/details?id=com.emanuelef.remote_capture&hl=de&gl=US

Bug Bounty Programm im SVDGV mit Intigriti

Weiterführende Links:

- https://owasp.org/Top10/
- https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Studien/Penetrationstest/penetrationstest.pdf?__blob=publicationFile&v=3
- https://www.bfarm.de/SharedDocs/Downloads/DE/Medizinprodukte/diga_leitfaden.html
- https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Sicherheitsberatung/Pentest_Webcheck/Leitfaden_Penetrationstest.pdf
- https://owasp.org/www-project-web-security-testing-guide/stable/
- https://www.intigriti.com/de
- https://gesundheits-apps.org/diga/DiGA.htm
- https://gesundheits-apps.org/gpn.pdf
- https://play.google.com/store/apps/details?id=com.emanuelef.remote_capture&hl=de&gl=US