Penetrationstestbericht

Mobile App 'DubMoney'

der Dubius Payment Ltd.

EXAMPLE

6. August 2024

v1.1

Inhaltsverzeichnis

i Änderungsverzeichnis

Version	Beschreibung	Autor	Datum
1.0	Reporterstellung	Dominik Sauer	5. August 2024
1.1	Qualitätssicherung	QA-Team	6. August 2024

1 Ansprechpartner

1.1 Ansprechpartner Dubius Payment Ltd.

Damian Westcott
CEO

✉ d.westcott@dubius-payment.com

Dubius Payment Ltd.
71 Peachfield Road
SO53 4NE CHANDLER
United States

1.2 Ansprechpartner binsec GmbH

Dominik Sauer
Head of Penetration Testing

☎ +49 69247560713
✉ ds@binsec.com

binsec GmbH
Solmsstraße 41
60486 Frankfurt am Main
Germany

1.3 Über den Pentester

Seit 2013 obliegt Herrn Dominik Sauer die Leitung und Durchführung von Penetrationstests der binsec GmbH. Er startete seine Karriere bereits während seines Informatikstudiums an der Hochschule Darmstadt und beendete seine akademische Laufbahn mit einem sehr guten Masterabschluss in Informatik mit Vertiefungsrichtung IT-Sicherheit. Ebenfalls kann er die führenden Zertifikate im Bereich Penetration-Testing vorweisen und ist seit 2013 „Offensive Security Certified Professional (OSCP)“ sowie „Offensive Security Certified Expert (OSCE)“ seit 2015.

Darüber hinaus engagiert er sich seit 2017 als Dozent in Forschung und Lehre an der Hochschule Darmstadt (HDA) und an der Technischen Hochschule Mittelhessen (THM). So hält er beispielsweise Lehrveranstaltungen zu Penetration Testing oder zur digitalen Forensik und betreut wissenschaftliche Arbeiten.

2 Projektübersicht

2.1 Einführung

Die Dubius Payment Ltd. ließ ihre mobile Anwendung 'DubMoney' einem Penetrationstest unterziehen. Im Detail können Kunden über die Mobile Anwendungen Transaktionen tätigen. Das Backend bzw. die angebundene API sollte nicht getestet werden, da sie bereits in einem vorherigen Penetrationstests untersucht wurde.

2.2 Rahmenbedingungen

Der Penetrationstest wurde zwischen dem 29. Juli 2024 und 9. August 2024 als externer Grey-Box-Test durchgeführt, ohne aggressive Angriffstechniken wie DDoS-Attacken zu verwenden. Die vollständige Klassifizierung ist in Kapitel 5.1 dargestellt. Die grundsätzliche Vorgehensweise ist in Kapitel 5.2 beschrieben. Alle Tests wurden von den folgenden IP-Adressen ausgeführt:

IPv4

- 185.156.254.128/25
- 217.111.127.122/32
- 162.55.59.194/32

IPv6

- 2a07:a1c1:1:600::/63
- 2001:920:1914:3464::/64

2.3 Scope

Für den Penetrationstest wurden die folgenden Installationspakete für Android und iOS bereitgestellt:

Betriebssystem	Bundle ID	Version	Anmerkung
Android	dp.dub.dev	2.6.2 (Dev)	Eine Root-Erkennung war aktiv.
iOS	dp.dub.dev	2.6.2 (Dev)	Eine Jailbreak-Erkennung war aktiv.

Die Apps wurden auf den folgenden mobilen Endgeräten installiert:

Smartphone	Betriebssystemversion	Zugriffsberechtigung
Samsung Galaxy S9	Android 10	Das Endgerät war gerooted.
Pixel 6	Android 14	-
iPhones 12	iOS 16.5.1	Ein Jailbreak war auf dem Endgerät vorhanden.

Um Störungen in der Produktionsumgebung zu vermeiden, wurde der Penetrationstest in der Entwicklungsumgebung durchgeführt. Für die Durchführung des Penetrationstests wurden die folgenden Benutzerkonten über die öffentliche Registrierung angelegt:

- ds@binsec.com
- ds+1@binsec.com
- ds+2@binsec.com

2.4 Durchgeführte Prüfungen

In dem zuvor genannten Scope wurden nachfolgende Prüfpunkte bei den einzelnen Zielsystemen bzw. -anwendungen untersucht. Diese Auflistung wird automatisiert aus dem Dokumentationswerkzeug der binsec GmbH erzeugt.

Prüfobjekt → Android-App: DubMoney

Ergebnis	Task: Reverse Engineering	Findings
✘	Identifizieren der erforderlichen Plattform Version	Seite 11
✔	Überprüfung der App auf den Einsatz von veralteten Softwarekomponenten	-
✔	Rekonstruktion des Source Codes anhand der APK	-
✔	Identifikation der verwendeten Kryptoalgorithmen	-
✔	Umgehung der Root-Erkennung in Android	-
✔	Überprüfung, ob die Anwendung gedebugged werden kann	-
Ergebnis	**Task: Plattformnutzung**	**Findings**
✔	Identifikation der angeforderten Berechtigungen seitens der mobilen Anwendung	-
✔	Überprüfung, ob die Ausführung der App auf einem mobilen Endgerät ohne Displaysperre unterbunden wird	-
✔	Überprüfung, ob Screen-Overlay-Angriffe unterbunden werden	-
✔	Überprüfung, ob sensible Daten über IPC-Mechanismen (Inter Process Communication) offengelegt werden	-
Ergebnis	**Task: Manipulation des Source Codes**	**Findings**
✔	Überprüfung, ob das Serverzertifikat eines angebundenen Backends verifiziert wird	-
✔	Überprüfung, ob JavaScript in WebViews ausgeführt werden kann	-
Ergebnis	**Task: Data Storage**	**Findings**
✔	Überprüfung, ob Datenbackups sensitive Informationen enthalten	-
✔	Überprüfung, ob Screen Capturing unterbunden wird	-
✘	Überprüfung, ob der Datenspeicher der App nach mehrfach fehlerhaften Anmeldeversuchen gelöscht wird	Seite 12
✘	Überprüfung der Zeitdauer von sensitiven Daten im Hauptspeicher	Seite 13
✔	Überprüfung, ob sensitive Daten über die Benutzeroberfläche angezeigt werden	-
✔	Überprüfung, ob der Tastatur-Cache für Texteingabefelder deaktiviert ist	-
✔	Überprüfung, ob sensible Daten in Anwendungsprotokolle geschrieben werden	-

✔	Überprüfung, ob sensitive Daten unsicher auf dem mobilen Engerät abgelegt werden.	-

Prüfobjekt → iOS-App: DubMoney

Ergebnis	Task: Reverse Engineering	Findings
✘	Identifizieren der erforderlichen Plattform Version	Seite 11
✔	Überprüfung der App auf den Einsatz von veralteten Softwarekomponenten	-
✔	Umgehung der Jailbreak-Erkennung von iOS-Geräten	-
✔	Identifikation der verwendeten Kryptoalgorithmen	-
✔	Überprüfung, ob die Anwendung gedebugged werden kann	-
Ergebnis	**Task: Plattformnutzung**	**Findings**
✔	Überprüfung, ob Drittanbieter-Tastaturen verwendet werden können	-
✔	Überprüfung, ob die Ausführung der App auf einem mobilen Endgerät ohne Displaysperre unterbunden wird	-
✔	Überprüfung, ob sensible Daten über IPC-Mechanismen (Inter Process Communication) offengelegt werden	-
Ergebnis	**Task: Manipulation des Source Codes**	**Findings**
✔	Überprüfung, ob das Serverzertifikat eines angebundenen Backends verifiziert wird	-
✔	Überprüfung, ob JavaScript in WebViews ausgeführt werden kann	-
Ergebnis	**Task: Data Storage**	**Findings**
✔	Überprüfung, ob Datenbackups sensitive Informationen enthalten	-
✔	Überprüfung, ob Screen Capturing unterbunden wird	-
✘	Überprüfung, ob der Datenspeicher der App nach mehrfach fehlerhaften Anmeldeversuchen gelöscht wird	Seite 12
✘	Überprüfung der Zeitdauer von sensitiven Daten im Hauptspeicher	Seite 13
✔	Überprüfung, ob sensitive Daten über die Benutzeroberfläche angezeigt werden	-
✔	Überprüfung, ob der Tastatur-Cache für Texteingabefelder deaktiviert ist	-
✔	Überprüfung, ob sensible Daten in Anwendungsprotokolle geschrieben werden	-
✔	Überprüfung, ob sensitive Daten unsicher auf dem mobilen Engerät abgelegt werden.	-

3 Managementübersicht

3.1 Zusammenfassung

Der Penetrationstest wurde zwischen dem 29. Juli 2024 und 9. August 2024 durchgeführt. Dabei konnten 6 Schwachstellen identifiziert werden, die zu 3 Findings zusammengefasst und einer initialen Risikobewertung unterzogen wurden. Insgesamt wurde kein Finding mit unmittelbarem Handlungsbedarf und 2 Findings mit Handlungsbedarf bewertet. Das Finding mit der Bewertung Hinweis ist als Vorschlag zur Erhöhung des Sicherheitsniveaus zu verstehen.

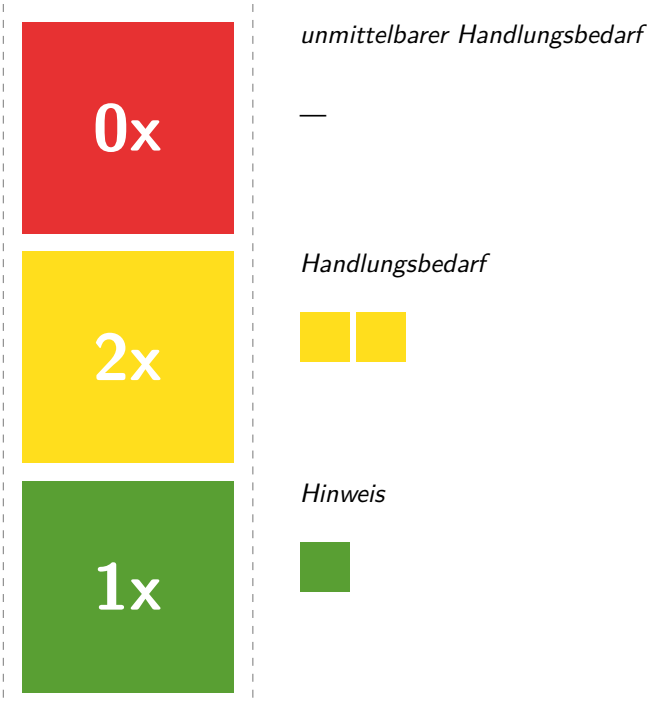


Abb. Risikoverteilung der Findings

Während des Penetrationstests konnten Schwachstellen in der mobilen Datenverarbeitung sowie -ablage identifiziert werden.

3.2 Liste der Findings

1 

Nicht behoben: *Fehlkonfiguration*

Die mobile Anwendung unterstützt Betriebssysteme, die vom Hersteller nicht länger mit Sicherheitsupdates versorgt werden. Siehe Seite 11.

System: Android, iOS

2 

Nicht behoben: *Datenspeicherung*

Die mobile Anwendung setzt sich bei mehrfach falscher Passworteingabe für das registrierte Benutzerkonto nicht zurück. Siehe Seite 12.

System: Android, iOS

3 

Nicht behoben: *Fehlkonfiguration*

Die mobile Anwendung behält sensible Daten länger als nötig im Hauptspeicher. Siehe Seite 13.

System: Android, iOS

4 Technischer Bericht

4.1 Installation der mobilen Anwendung auf veralteten Betriebssystemen

- Handlungsbedarf
- Schaden: Hoch
- Eintrittswahrscheinlichkeit: Gering

Finding #1 **Nicht behoben**

Bei der Entwicklung von iOS- und Android-Apps haben Entwickler die Möglichkeit festzulegen, welche Betriebssystemversionen von ihrer App unterstützt werden. Dies erfolgt über spezifische Attribute in den Konfigurationsdateien der jeweiligen App. Für iOS-Apps wird die minimale unterstützte Betriebssystemversion in der 'Info.plist' mit dem Attribut 'MinimumOSVersion' festgelegt. Bei Android-Apps wird dies in der 'AndroidManifest.xml' durch das Attribut 'minSdkVersion' bestimmt. Durch das Setzen dieser Attribute lässt sich sicherstellen, dass die Apps nur auf Betriebssystemversionen laufen, die als sicher gelten und über die von der App benötigten APIs verfügen.

Bei der Überprüfung der mobile App wurde festgestellt, dass Betriebssystemversionen bei Android und iOS unterstützt werden, welche keine Updates vom Hersteller mehr erhalten:

```
minSdkVersion : '24
```

Unterstützung von Android 7.0

```
<key>MinimumOSVersion</key>
  <string>12.0</string>
```

Unterstützung von iOS 12

Im Allgemeinen können veraltete Betriebssysteme über bekannte Schwachstellen verfügen. Darüber hinaus erhalten sie keine Sicherheitsupdates mehr und verfügen gegebenenfalls nicht über aktuelle Sicherheitsfeatures, die vor unbefugtem Zugriff auf sensitive App-Inhalte schützen.

Empfehlung

In Anlehnung an die End-of-Life-Zyklen von Android[1] und iOS[2] sollten einzig Betriebssysteme unterstützt werden, die Security Updates durch den Hersteller erhalten.

[1] https://endoflife.date/android
[2] https://endoflife.date/ios

4.2 Fehlende Datenlöschung bei mehrfach falscher Passworteingabe

Handlungsbedarf
Schaden: Mittel
Eintrittswahrscheinlichkeit: Mittel

Finding #2 **Nicht behoben**

Nach Security Best Practices sollen die Daten einer mobilen Anwendung gelöscht bzw. zurückgesetzt werden, wenn eine falsche Passworteingabe für das registrierte Benutzerkonto mehrfach hintereinander erfolgt. Während des Penetrationstests konnte jedoch kein Zähler festgestellt werden, wie häufig ein falsches Passwort in der 'DubMoney'-App eingegeben werden darf. Stattdessen blieben die Daten auch nach 15 fehlerhaften Anmeldeversuchen auf dem mobilen Endgerät erhalten.

Empfehlung

Es sollte eine Datenlöschung nach mehrfach falscher Passworteingabe in der mobilen Anwendung implementiert werden.

4.3 Fehlende Speicherfreigabe nach Datenverarbeitung

Hinweis
Schaden: Warnung
Eintrittswahrscheinlichkeit: Gering

Finding #3 **Nicht behoben**

Nach Security Best Practices sollen sensitive Daten nur so lange wie nötig im Hauptspeicher gehalten werden. Obwohl das Passwort nach einer Benutzeranmeldung nicht mehr von der mobilen Anwendung benötigt wird, konnte es trotz 5 Minuten Inaktivität des Nutzers auf der App-Startseite im Hauptspeicher aufgefunden werden:

```
# memory search "54 65 73 74 31 32 33 34 21 61 62 63 64"
Searching for: 54 65 73 74 31 32 33 34 21 61 62 63 64
137b0da8  54 65 73 74 31 32 33 34 21 61 62 63 64 00 00 00  Test1234!abcd...
137b0db8  98 e7 78 70 00 00 00 00 00 00 00 00 00 00 00 00  ..xp............
137b0dc8  58 6a 80 70 00 00 00 00 02 00 00 00 08 ff ba 70  Xj.p...........p
Pattern matched at 1 addresses
```

Erfolgreiche Suche nach dem Benutzerpasswort 'Test1234!abcd'

```
Using USB device `iPhone`
Agent injected and responds ok!

     _  _
 ___| |_|_|___ ___| |_|_|___ ___
| . | . | | -_|  _|  _| | . |   |
|___|___| |___|___|_| |_|___|_|_|
      |___|(object)inject(ion) v1.11.0

     Runtime Mobile Exploration
        by: @leonjza from @sensepost

[tab] for command suggestions
# memory search "54 65 73 74 31 32 33 34 21 61 62"
Searching for: 54 65 73 74 31 32 33 34 21 61 62
12bd66820  54 65 73 74 31 32 33 34 21 61 62 63 64 00 00 ed  Test1234!abcd...
12bd66830  17 cb af fb a1 25 00 00 00 00 00 00 00 00 00 00  .....%..........
12bd66840  20 fa 35 81 02 00 00 00 d0 67 60 82 02 00 00 00   .5......g`.....
28088c880  54 65 73 74 31 32 33 34 21 61 62 63 64 00 00 00  Test1234!abcd...
28088c890  1f 64 65 2d 44 45 2d 28 6e 75 6c 6c 29 2d 30 2d  .de-DE-(null)-0-
28088c8a0  57 49 54 48 4f 55 54 2d 44 65 66 61 75 6c 74 73  WITHOUT-Defaults
Pattern matched at 2 addresses
```

Erfolgreiche Suche nach dem Benutzerpasswort 'Test1234!abcd'

Die Hauptspeicheranalyse wurde sowohl für Android als auch für iOS mit dem Runtime-Mobile-Exploration-Toolkit 'objection'[3] durchgeführt, welches administrativen Zugriff auf die mobilen Endgeräte erfordert.

Empfehlung

Die Datenobjekte sollten nach Verarbeitung überschrieben und anschließend gelöscht werden.

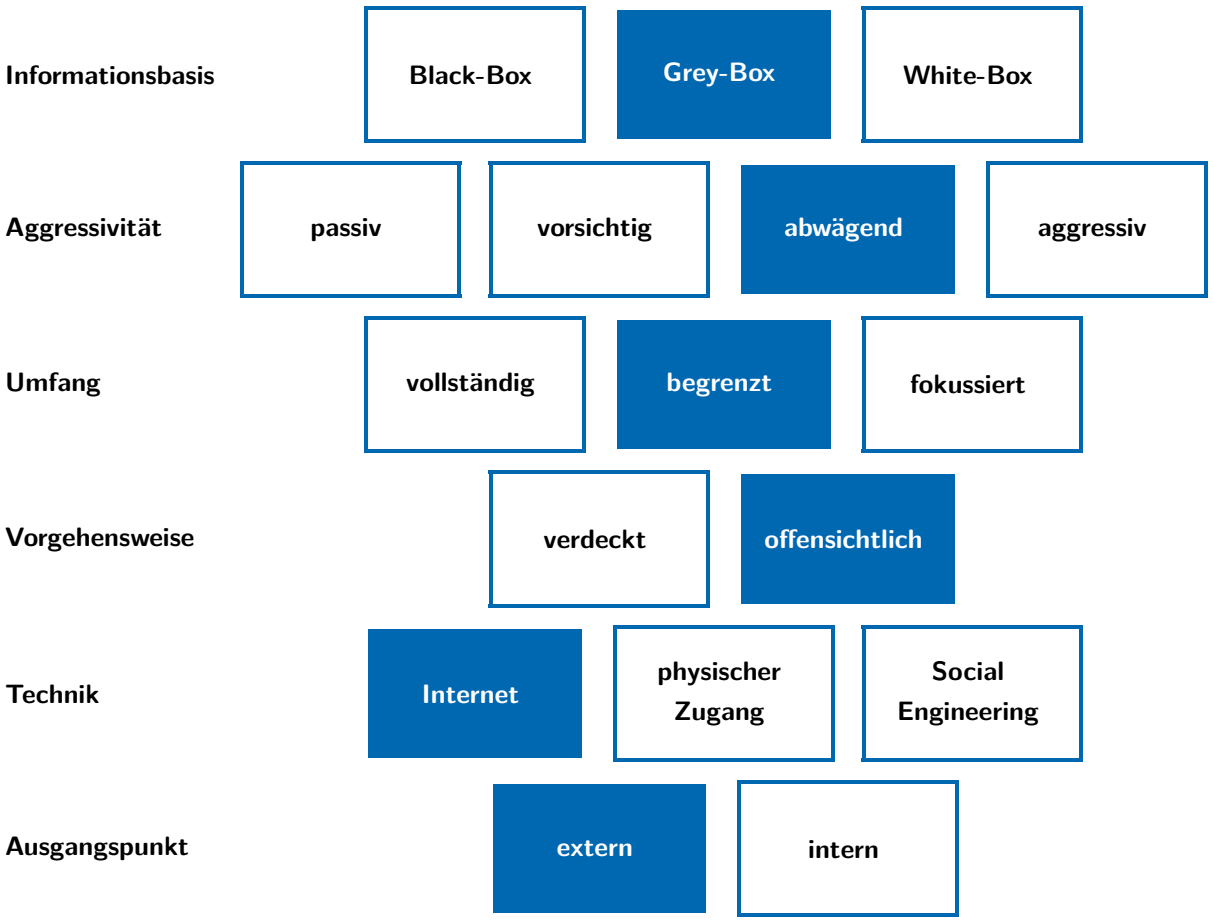
[3] https://github.com/sensepost/objection

5 Anhang A

5.1 Klassifizierung-Details

Als Vorgehensweise wurde in Kooperation mit der Dubius Payment Ltd. folgende Klassifizierungsvariante ausgewählt: Der Penetrationstest wurde als externer Grey-Box-Test durchgeführt, ohne aggressive Angriffstechniken wie DDoS-Attacken zu verwenden.

Inhaltlich anlehnend an die Studie - „Durchführungskonzept für Penetrationstests“ - vom Bundesamt für Sicherheit in der Informationstechnik (BSI), ergibt sich folgendes Schema zur Klassifizierung des Penetrationstest:



5.2 Vorgehensweise bei Mobile Apps

Die Untersuchungsmethode der binsec GmbH bei mobilen Anwendungen orientiert sich am OWASP Mobile Application Security Testing Guide und an den OWASP Mobile TOP 10. Das Open Web Application Security Project (OWASP) ist derzeit die weltweit größte Non-Profit-Organisation mit dem Ziel, die Sicherheit von Anwendungen zu erhöhen. Die OWASP Mobile Top 10 beinhalten die zehn kritischsten Schwachstellen bei mobilen Anwendungen. Nach der aktuellen Veröffentlichung von 2024 ist eine der am häufigsten auftretenden Schwachstellen die unsichere Ablage von sensitiven Daten, wie z.B. die Speicherung des Benutzerpassworts in einer Konfigurationsdatei. Sofern nicht anders gewünscht, beziehen wir auch angebundene APIs einer mobilen Anwendung in unseren Penetrationstest mit ein.

Prüfpunkte

Der methodische Prüfungsansatz von der binsec GmbH unterteilt sich grob in die folgenden Prüfphasen. Innerhalb der Prüfphasen wird die mobile Anwendung auf die Schwachstellen der OWASP Mobile TOP 10 untersucht. Während des Penetrationstests werden verschiedene Tools zur Analyse eingesetzt sowie intensive manuelle Prüfungen vorgenommen. Der genaue Verlauf des Penetrationstests ist stark von der Charakteristik der jeweiligen Anwendung abhängig und orientiert sich an der Vorgehensweise eines realen Angreifers. Um die volle Kontrolle über die Kommunikation und Einsicht über die Datenablage zu erhalten, versuchen wir Schutzmechanismen wie eine implementierte Jailbreak/Root-Erkennung oder HTTP-Pinning öffentlicher Schlüssel zu umgehen:

1. Einrichtung einer Testumgebung
2. Reverse Engineering & Manipulation des Source Codes
3. Überprüfung der Plattformnutzung
4. Sichere Datenablage & Kommunikation
5. Überprüfung der serverseitigen Schutzmechanismen

5.3 Risikobewertung

Die binsec GmbH versteht unter dem Begriff „Risiko“ die Kombination aus der Eintrittswahrscheinlichkeit einer Schwachstelle (bzw. der Wahrscheinlichkeit ihrer Ausnutzung) und dem möglichen Schadensausmaß. Die Eintrittswahrscheinlichkeit bzw. die Wahrscheinlichkeit der Ausnutzung einer Sicherheitslücke in IT-Systemen hängt im Wesentlichen von diesen Faktoren ab:

- Wie einfach kann die Schwachstelle identifiziert werden? (Visibility)
- Existieren vorgefertigte Exploits für diese Schwachstelle oder muss der Angreifer einen entsprechenden Wissensstand mitbringen um sie auszunutzen? (Exploitability)
- Setzt die Ausnutzung besondere Rechte voraus? (Privilege Escalation)
- Ist eine Kombination mit anderen Sicherheitslücken erforderlich? (Vulnerability Chaining)
- Ist für die Schwachstelle menschliche Interaktion notwendig? (Social Engineering)

Bestimmend für das mögliche Schadensausmaß sind die folgenden Klassifikationen:

1. Finanzieller Schaden
2. Vollständige Kompromittierung des Systems
3. Verletzung der Sicherheitsziele in Bezug auf Daten oder Benutzerkonten:
    a) Vertraulichkeit (Confidentiality)
    b) Verfügbarkeit (Availability)
    c) Integrität (Integrity)
4. Umgehung von Schutzmechanismen
5. Informationslücke

Aus der Kombination von Eintrittswahrscheinlichkeit und möglichem Schadensausmaß trifft der Penetrationstester eine subjektive Einschätzung des Risikos für jede gefundene Sicherheitslücke. Wir empfehlen zusätzlich eine eigene Bewertung der ermittelten Schwachstellen durchzuführen.

Die Einschätzung wird folgender Einstufung unterzogen:

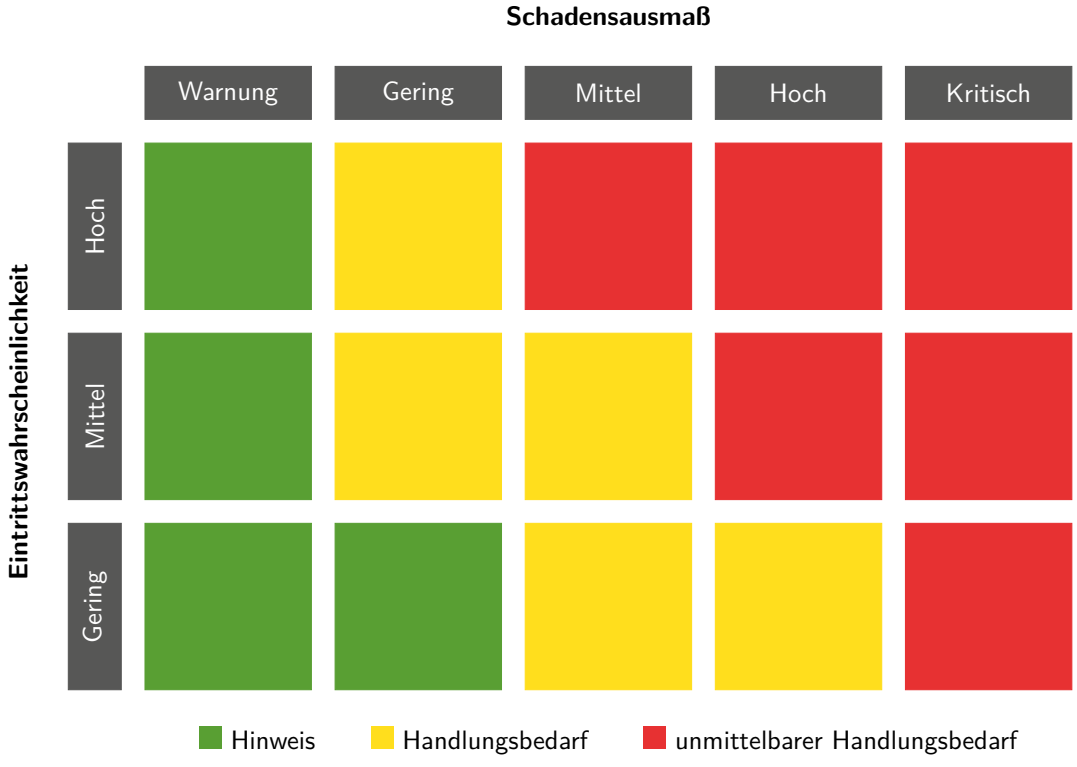
Eintrittswahrscheinlichkeit

Hoch	Die Schwachstelle ist offensichtlich oder Exploits sind frei verfügbar.
Mittel	Die Schwachstelle ist in angemessener Zeit zu entdecken, Exploits müssen eventuell angepasst werden.
Gering	Die Schwachstelle ist aufwendig zu finden und es werden eventuell Privilegien benötigt oder die Ausnutzung setzt weitere Schwachstellen voraus.

Schadensausmaß

Kritisch	Vollständige Kompromittierung des Systems.
Hoch	Verletzung der Sicherheitsziele in Bezug auf Informationen oder IT-Systeme.
Mittel	Umgehung von Schutzmechanismen.
Gering	Vorbereitung der Ausnutzung weiterer Schwachstellen als Teil einer Langzeitattacke.
Warnung	Informationslücke.

Aus der Einstufung des Risikos wird eine Handlungspriorität abgeleitet.



5.4 Über die binsec GmbH

Wir sind ein auf IT-Penetrationstests spezialisiertes Dienstleistungsunternehmen aus Frankfurt am Main. Seit 2013 ist die Durchführung technischer Sicherheitsanalysen von IT-Infrastrukturen, Web-Anwendungen, APIs, Mobilen APPs (Android / iOS) usw. der Kernbestandteil unserer täglichen Arbeit. Als inhabergeführtes Unternehmen legen wir hohen Wert auf die langfristige Zufriedenheit unserer Kunden. Die Zertifizierungen unserer Mitarbeiter, die Lehrtätigkeiten an Hochschulen sowie unsere Praxiserfahrung sprechen für sich.

binsec GmbH
Solmsstraße 41
60486 Frankfurt am Main
Germany

✉ info@binsec.com
☎ +49 69 2475607-0

Geschäftsführer: Patrick Sauer
Prokurist: Dominik Sauer, Florian Zavatzki

Handelsregister: Frankfurt a.M. HRB 97277
USt-IdNr.: DE290966808