TÜV AUSTRIA GRUPPE

Desktop-App → Web-App: Was kann schon schiefgehen?

Lejla Sarcevic, BSc, MSc, Ing. Simon Schönegger, BSc, MSc, DI Rainer Seyer, BSc

16.06.2023

Vorstellung

TÜV Trust IT

Pentest-Team

15 Pentester an 3 Standorten

- Brunn am Gebirge
- Graz
- (Klagenfurt)

Andere Teams:

- ISMS
- IT-Sec
- OT
- EIM

Vorstellung

Lejla Sarcevic, BSc, MSc

Studium Software Design (BSc) und IT&Mobile Security (MSc) FH Joanneum

Ehemalige Software Entwicklerin (Mobile)

Penetration Testerin bei TÜV Austria TÜV TRUST IT GmbH

Ing. Simon Schönegger, BSc, MSc

Studium Software Design (BSc) und IT&Mobile Security (MSc) FH Joanneum

Ehemaliger Software Entwickler (Desktop)

Penetration Tester bei TÜV Austria TÜV TRUST IT GmbH

DI Rainer Seyer, BSc

Studium IT Security (BSc) und Information Security (DI) FH St. Pölten

Background in Elektrotechnik

Teamleiter Penetration Testing / Senior Penetration Tester TÜV Austria TÜV TRUST IT GmbH

Ziele

- ✓ Sensibilisieren für die Thematik
- ✓ Kein Hersteller-Bashing
  - Responsible Disclosure-Prozess wurde eingehalten
  - Kunden- und Herstellernamen werden nicht genannt
  - Hersteller der Software hat schnell reagiert

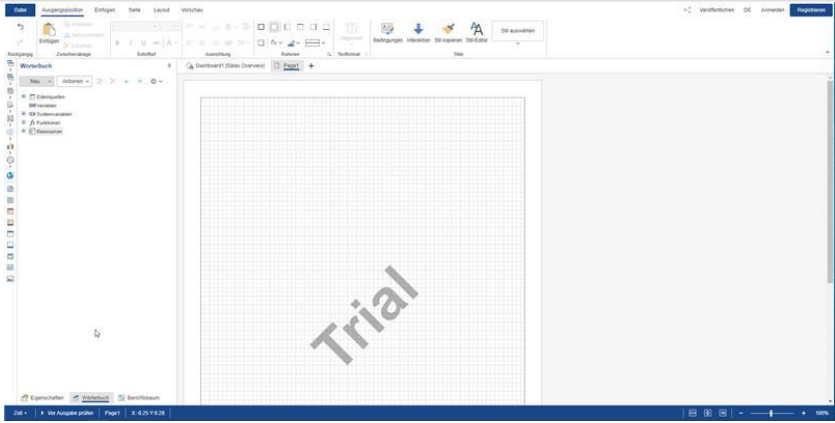
Ausgangslage

- ✔ Grey-Box Penetration Test
- ✔ Einsatz externes Produktes zur Generierung von Reports
- ✔ Externes Produkt explizit im Scope
- ✔ Produkt besteht aus:
  - Reporting Designer (Read & Write)
  - Reporting Viewer („readonly“ 😉)
  - Beides basiert auf .NET

Produkte

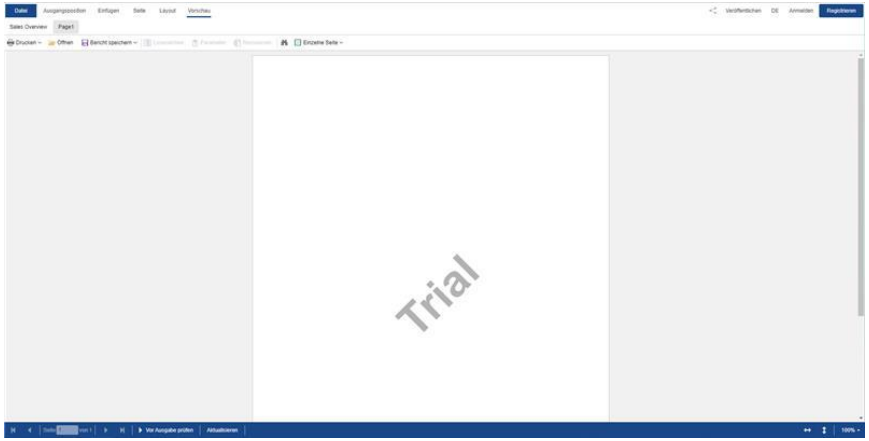
Reporting Designer

- Einbinden von externen Datenquellen
  - DB Connections, HTTP Responses, CSV, JSON, XML
- Generierung von Dashboards
- Exportieren von Report Files
  - Viewer Import möglich
- Zugriff auf **lokale** Report Files am Server
- Zugriff auf **lokale** Datenquellen am Server
- Ob man es glaubt oder nicht, der Screenshot zeigt eine **Webapplikation**



Reporting Viewer

- ✓ Report Files anzeigen, nicht editieren
- ✓ Interpretation von Files **am Server**
- ✓ Zeigt Reports mit geladenen Daten an
- ✓ „nur lesender“ Zugriff auf Report Files
- ✓ Auch dieser Screenshot zeigt eine **Webapplikation**



Report Files

- ✔ Custom XML ähnliches Konstrukt
- ✔ Beinhaltet Definition aller Datenquellen
- ✔ Beinhaltet Definition aller Felder des Reports und ihre Referenzen
- ✔ Dynamische Erweiterung durch C# Code im Report File möglich

Report Files

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<StiSerializer version="1.02" type="Net" application="StiReport">
  <Dictionary Ref="1" type="Dictionary" isKey="true">
    <BusinessObjects isList="true" count="1">
      <CorporateAction Ref="2" type="[illegible]" isKey="true">
        <Alias>CorporateAction</Alias>
        <BusinessObjects isList="true" count="0" />
        <Category>CorporateAction</Category>
        <Columns isList="true" count="1">
          <value>notes,System.String</value>
        </Columns>
        <Dictionary isRef="1" />
        <Guid>10368e9dd69843768cd2648d92814cc8</Guid>
        <Name>CorporateAction</Name>
      </CorporateAction>
    </BusinessObjects>
    <Databases isList="true" count="0" />
    <DataSources isList="true" count="0" />
    <Relations isList="true" count="0" />
    <Report isRef="0" />
    <Resources isList="true" count="0" />
    <Variables isList="true" count="1">
      <value>,Data,Data,,System.String,,False,False,False,False</value>
    </Variables>
  </Dictionary>
  <EngineVersion>EngineV2</EngineVersion>
```

Report Files

- ✔ Dynamische Erweiterung durch C# Code im Report File möglich

```
<ReferencedAssemblies isList="true" count="11">
  <value>System.Drawing.Dll</value>
  <value>System.Windows.Forms.Dll</value>
</ReferencedAssemblies>
<ReportAlias>Report</ReportAlias>
<ReportChanged>1/19/2023 8:28:37 AM</ReportChanged>
<ReportCreated>10/8/2018 1:28:08 PM</ReportCreated>
<ReportFile ...</ReportFile>
<ReportGuid>b888e51bd74743d48d31089bd05c7004</ReportGuid>
<ReportName>Report</ReportName>
<ReportUnit>Centimeters</ReportUnit>
<ReportVersion>2023.1.4.0</ReportVersion>
<Script>using System;
using System.Drawing;

namespace StiReports
{
  public class FoundStuff
  {
  }
}
```

Also, was kann schon schief gehen?

Spoiler: Endresultat

- ✓ Es konnten „geringe“ Probleme in der Architektur der Software gefunden werden
  - CVE-2023-25260 (CVSS: 8.6)
  - CVE-2023-25261 (CVSS: 10.0)
  - CVE-2023-25262 (CVSS: 5.4)
  - CVE-2023-25263 (CVSS: 7.9)

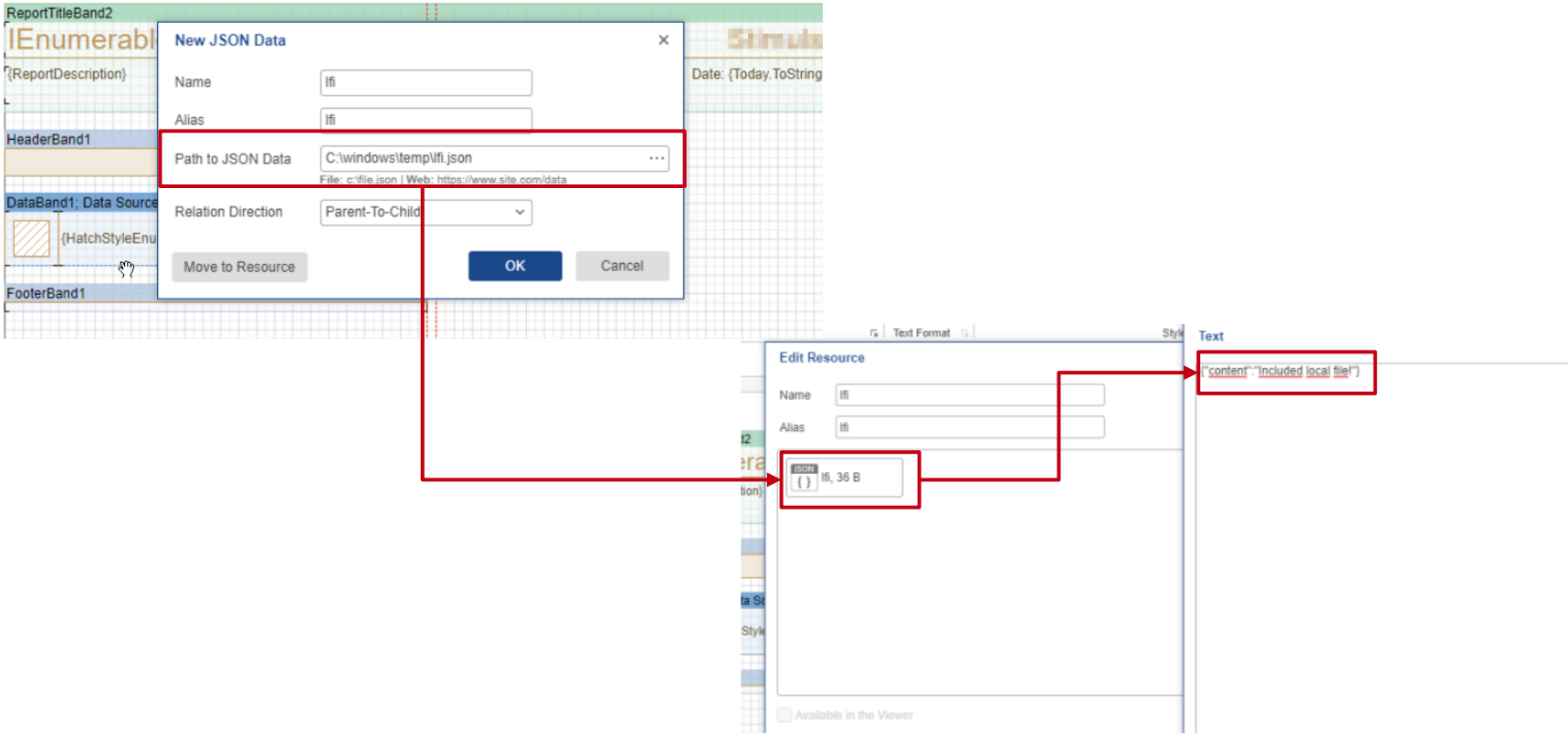
Grundlegende Probleme

- ✓ Mit höher privilegiertem Benutzer ausgeführt
- ✓ Offensichtliche Konvertierung einer Desktop-Applikation in eine Web-Applikation
  - Großteil der Operationen am Server ausgeführt anstatt am Client
  - Auslagerung von Security-Themen mittels „Responsible Use“
- ✓ Per default keine Authentication für Komponenten
- ✓ Fragwürdige Features

Problem 1: Local file Inclusion

- ✓ Betrifft Reporting Designer
- ✓ Einbindung von CVS, JSON, XML, etc. Dateien als Datenquellen
- ✓ Abruf mittels HTTP oder Angabe eines Full-Qualified-Path
- ✓ Welch Wunder: Full-Qualified-Path bindet **Serverressourcen** ein.
- ✓ .NET Applikation, d.h. appsettings.json und Web.config vorhanden

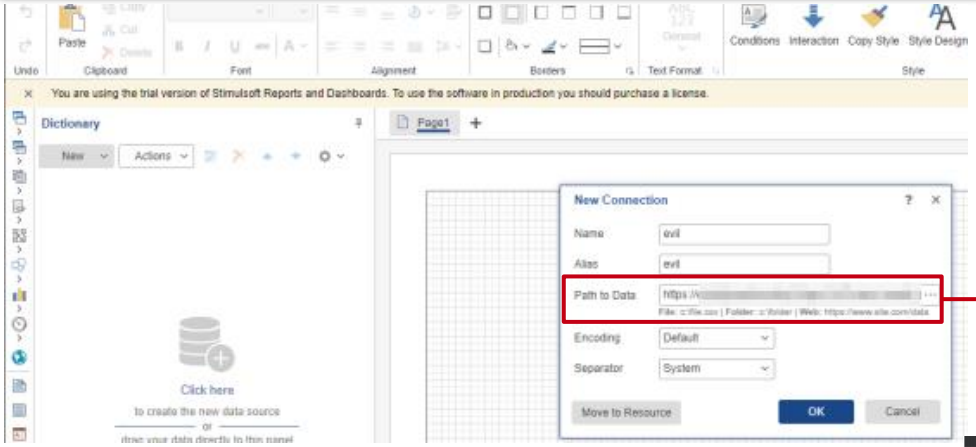
Problem 1: Local file Inclusion

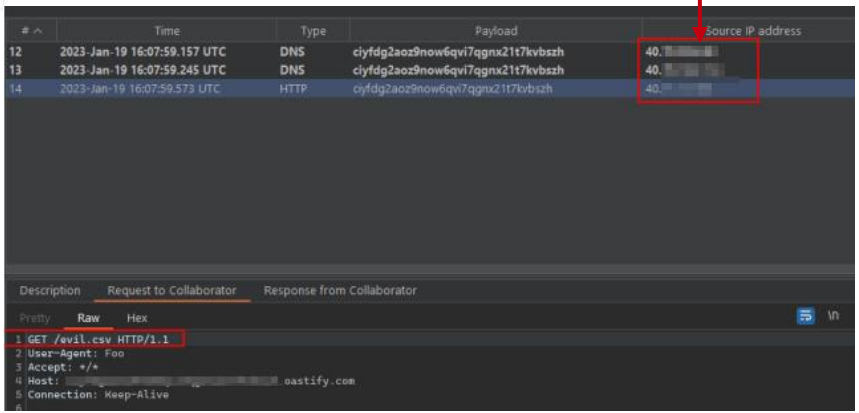


Problem 2: Server Side Request Forgery

- ✓ Betrifft Reporting Designer
- ✓ Einbindung von HTTP-Responses als Datenquellen
- ✓ Daten werden **serverseitig** abgerufen

Problem 2: Server Side Request Forgery





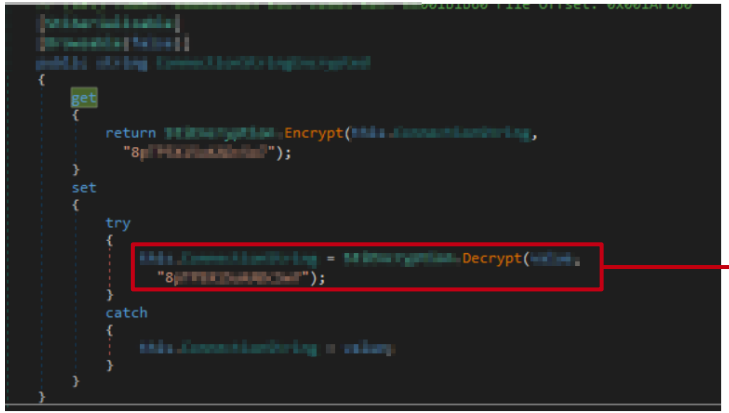
Problem 3: Verschlüsselung von Datenquellen

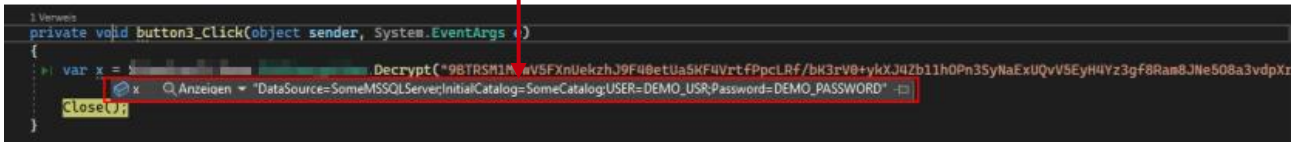
- ✔ Betrifft Reporting Designer
- ✔ Datenquellen in Report Files gespeichert
- ✔ Reminder: Datenbankverbindungen können als Datenquelle angegeben werden!
- ✔ Schlechte Nachricht:
  Diese Datenquellen sind im Report File „verschlüsselt“ ☹

```
<Databases isList="true" count="1">
    <MS_x0020_SQL Ref="2" type="[redacted]" is
    <Alias>MS SQL</Alias>
    <ConnectionStringEncrypted>
        9BTRSM1MowV5FXnUekzhJ9F40etUa5KF4VrtfPpcLRf/bK3rV0+ykXJ4Zb
        11hOPn3SyNaExUQvV5EyH4Yz3gf8Ram8JNe5O8a3vdpXr4brvlEtnLXW3S+Wxa
    </ConnectionStringEncrypted>
    <Name>MS SQL</Name>
    </MS_x0020_SQL>
</Databases>
```

Problem 3: Verschlüsselung von Datenquellen

- ✔ Gute Nachrichten:
  Die Software verwendet in **allen** Versionen einen **statischen** Schlüssel zum Ver- und Entschlüsseln dieser Datenquellen! ☺
  (Source Code ist nicht obfuscated)





Problem 4: Remote Code Execution

- ✔ Betrifft Reporting Designer und Reporting Viewer
- ✔ Import von Report Files mittels Designer und Viewer möglich
  - Serverseitige Interpretation der Files
- ✔ Dynamische Erweiterung des Reports durch eingebetteten C# Code
- ✔ Aber keine Angst! Es werden Sicherheitsmaßnahmen getroffen!
  - Zugriff auf diverse Libraries wie z.B. System.Net.Sockets, System.Net.HTTP und System.Diagnostics.Process unterbunden!
- ✔ Aufbau einer Reverse-Shell problematisch ☹

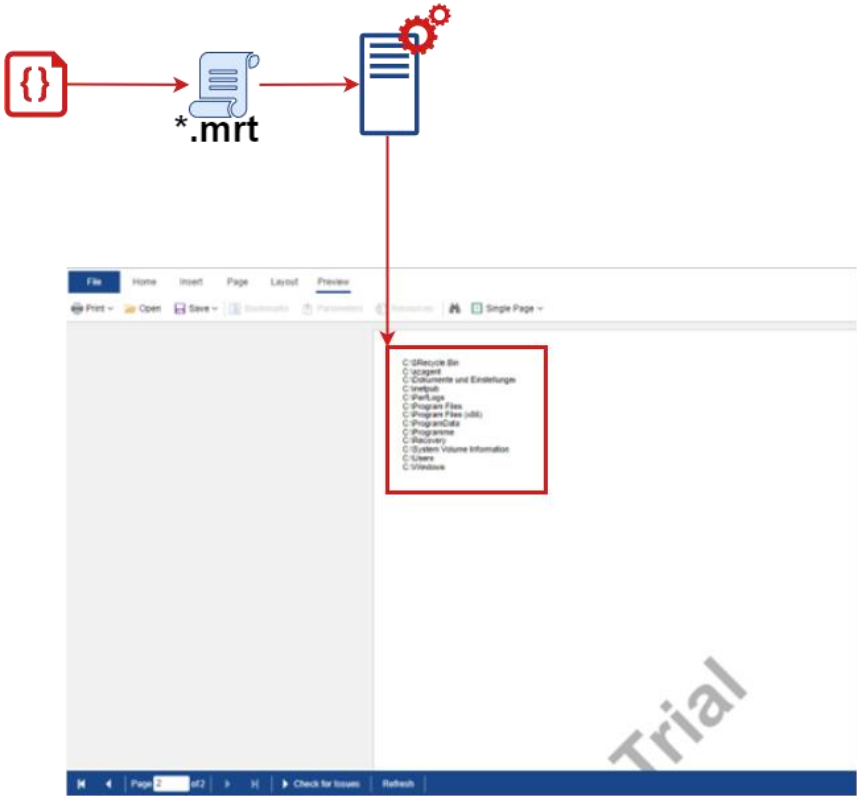
Was jetzt? 😒

Problem 4: Remote Code Execution

- ✔ Remote Code Execution über verschiedene Wege ausnutzbar
- ✔ Verwendung aller System Libraries die mit dem Dateisystem interagieren
  - Lese- & Schreibzugriff auf alle Dateien
- ✔ Newtonsoft.Json und LFI um Ergebnisse auszulesen
- ✔ Anzeigen von Ergebnissen direkt über ein Custom-Feld im Report

Problem 4: Remote Code Execution

- ✓ Beispiel: Auslesen schreibbarer Verzeichnisse



RCE in der Vergangenheit

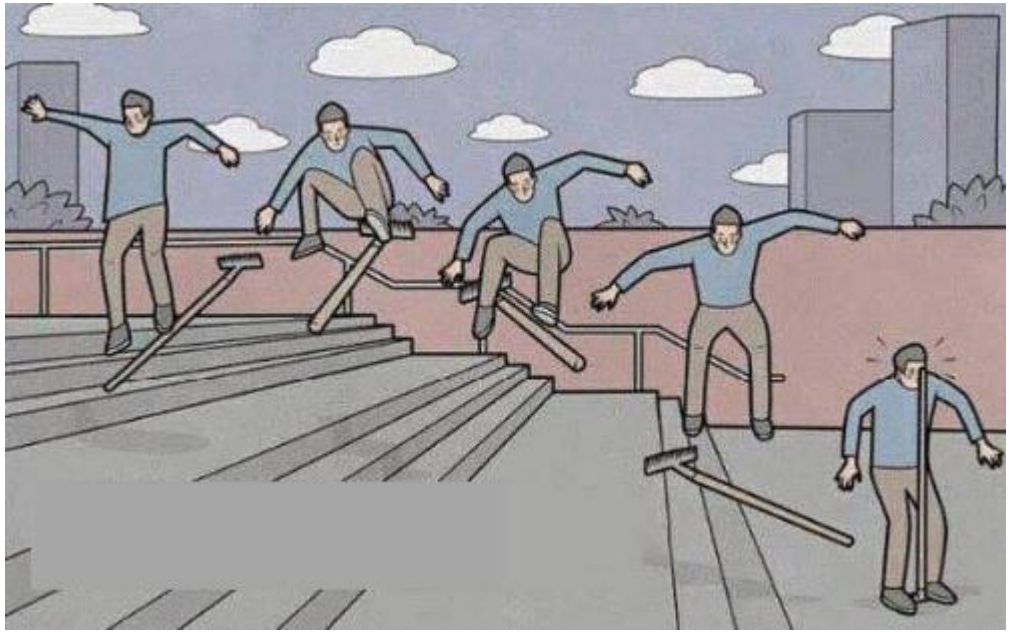
- CVE-2021-42777
  - Nutzt System.Diagnostics.Process (Exakt diese Library wurde unterbunden)
- CVE-2020-15865
  - Keine genauen Informationen über Exploit vorhanden, nutzt jedoch ebenfalls den eingebetteten C# Code

Aktueller Status

- ✓ CVE-2023-2560 (LFI)
  - Zusätzliche Konfigurationsmöglichkeit, die das Laden von lokalen Ressourcen unterbindet (Keine Information ob diese Einstellung per default gesetzt ist)
- ✓ CVE-2023-25261 (RCE)
  - Per default wird die Kompilierung von eingebettetem Code unterbunden, kann aber aktiviert werden
- ✓ CVE-2023-25262 (SSRF)
  - Wird nicht vom Hersteller behoben
- ✓ CVE-2023-25263 (Statische Secrets)
  - Wird nicht vom Hersteller behoben

- ✓ Generelle Advisory: Report Files nicht mit anderen teilen! (Warum kann man sie dann überhaupt exportieren?)

Fazit



- ✓ Diese und andere CVEs von uns findet ihr hier:
- ✓ https://cves.at/
- ✓ Besucht uns bei unserem Stand!