



# 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

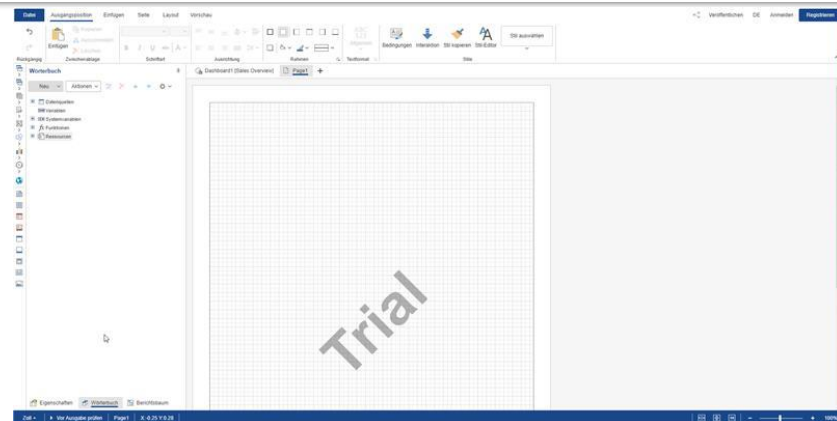
# Ausgangslage

- ✓ 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

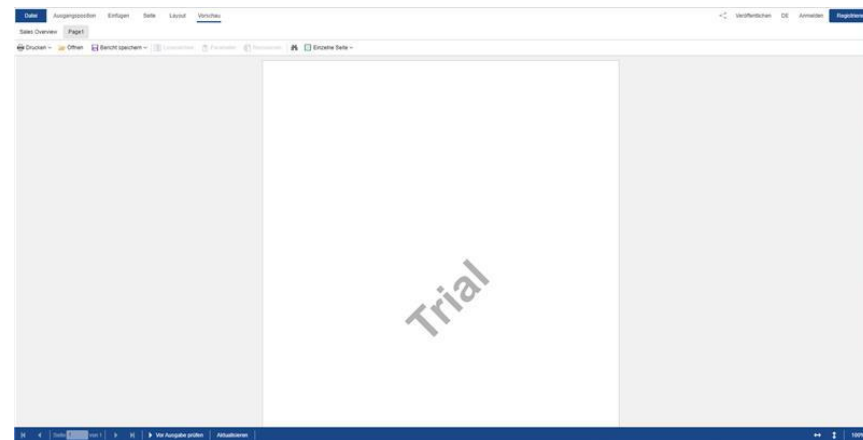
# Reporting Designer

- ✓ 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

```

1 <?xml version="1.0" encoding="utf-8" standalone="yes"?>
2 <StiSerializer version="1.02" type="Net" application="StiReport">
3   <Dictionary Ref="1" type="Dictionary" isKey="true">
4     <BusinessObjects isList="true" count="1">
5       <CorporateAction Ref="2" type="CorporateAction" isKey="true">
6         <Alias>CorporateAction</Alias>
7         <BusinessObjects isList="true" count="0" />
8         <Category>CorporateAction</Category>
9         <Columns isList="true" count="1">
10          <value>notes,System.String</value>
11        </Columns>
12        <Dictionary isRef="1" />
13        <Guid>10368e9dd69843768cd2648d92814cc8</Guid>
14        <Name>CorporateAction</Name>
15      </CorporateAction>
16    </BusinessObjects>
17    <Databases isList="true" count="0" />
18    <DataSources isList="true" count="0" />
19    <Relations isList="true" count="0" />
20    <Report isRef="0" />
21    <Resources isList="true" count="0" />
22    <Variables isList="true" count="1">
23      <value>,Data,Data,,System.String,,False,False,False,False</value>
24    </Variables>
25  </Dictionary>
26  <EngineVersion>EngineV2</EngineVersion>

```

# Report Files

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

```

107 <ReferencedAssemblies isList="true" count="11">
108   <value>System.Drawing.dll</value>
109   <value>System.Drawing.Common.dll</value>
110   <value>System.Windows.Forms.dll</value>
111   <value>System.Windows.Forms.DataVisualization.dll</value>
112   <value>System.Windows.Input.dll</value>
113   <value>System.Windows.Media.dll</value>
114   <value>System.Windows.Media.Imaging.dll</value>
115   <value>System.Windows.Navigation.dll</value>
116   <value>System.Windows.Shell.dll</value>
117   <value>System.Windows.Threading.dll</value>
118 </ReferencedAssemblies>
119 <ReportAlias>Report</ReportAlias>
120 <ReportChanged>1/19/2023 8:28:37 AM</ReportChanged>
121 <ReportCreated>10/8/2018 1:28:08 PM</ReportCreated>
122 <ReportFile <Path>C:\Users\johannes\AppData\Local\Temp\Report_10000000000000000000\ReportFile>
123 <ReportGuid>b888e51bd74743d48d31089bd05c7004</ReportGuid>
124 <ReportName>Report</ReportName>
125 <ReportUnit>Centimeters</ReportUnit>
126 <ReportVersion>2023.1.4.0</ReportVersion>
127 <Script>using System;
128     using System.Drawing;
129     using System.Drawing.Drawing2D;
130     using System.Drawing.Imaging;
131     using System.Drawing.Printing;
132     using System.Windows.Forms;
133     using System.Windows.Forms.DataVisualization;
134     using System.Windows.Input;
135     using System.Windows.Media;
136     using System.Windows.Media.Imaging;
137     using System.Windows.Navigation;
138     using System.Windows.Shell;
139     using System.Windows.Threading;
140
141     namespace StiReports
142     {
143         public class FoundStuff
144         {
145             public FoundStuff()
146             {
147                 // ...
148             }
149         }

```

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)

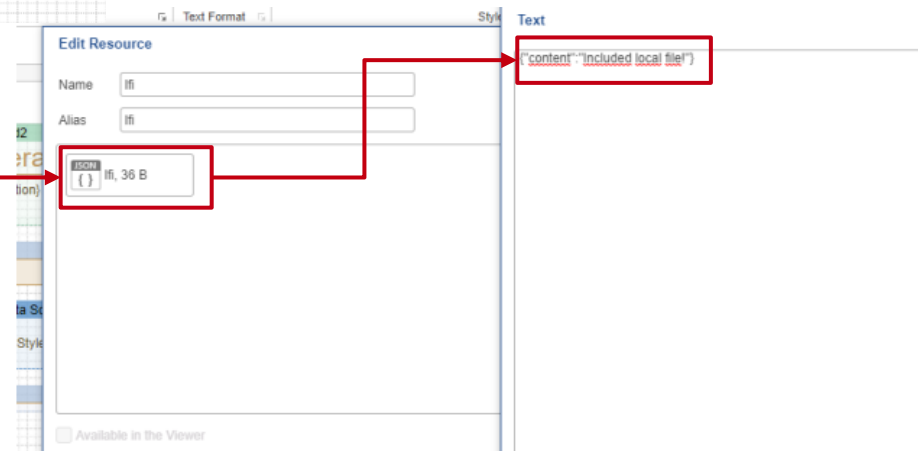
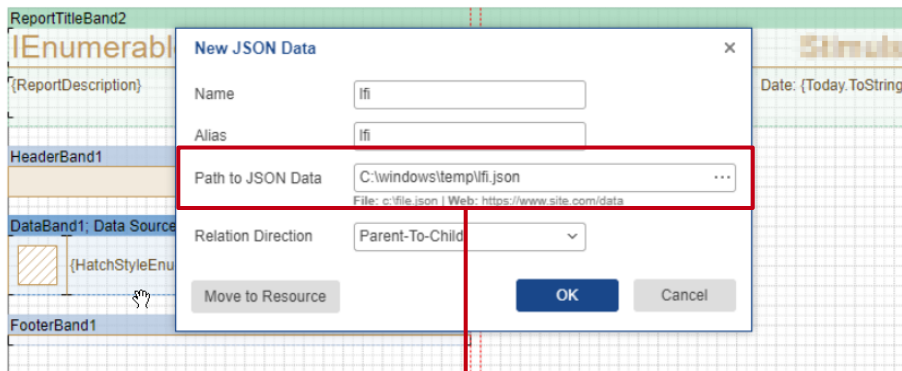
# Grundlegende Probleme

- ✓ 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

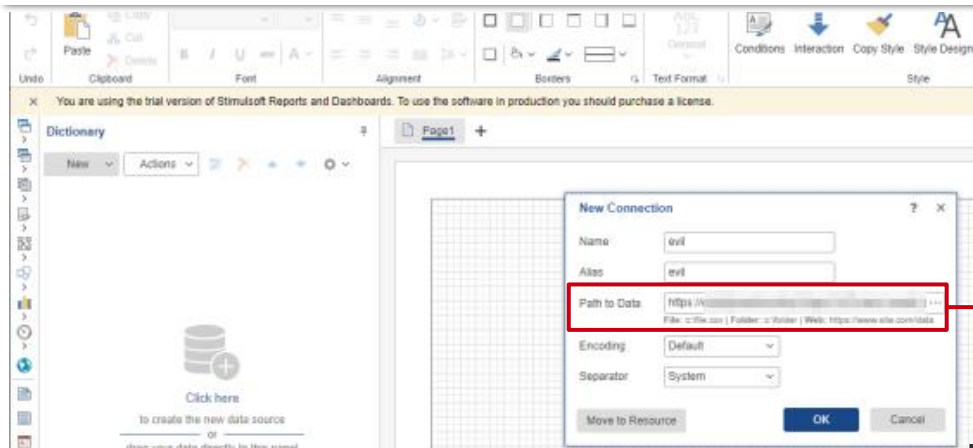




## Problem 2: Server Side Request Forgery

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

# Problem 2: Server Side Request Forgery



#	Time	Type	Payload	Source IP address
12	2023-Jan-19 16:07:59.157 UTC	DNS	clfydg2aoz9now6qv7qgnx21t7kvbszh	40.107.162.10
13	2023-Jan-19 16:07:59.245 UTC	DNS	clfydg2aoz9now6qv7qgnx21t7kvbszh	40.107.162.10
14	2023-Jan-19 16:07:59.573 UTC	HTTP	clfydg2aoz9now6qv7qgnx21t7kvbszh	40.107.162.10

Description	Request to Collaborator	Response from Collaborator
	<pre> Pretty Raw Hex 1 GET /evil.csv HTTP/1.1 2 User-Agent: Foo 3 Accept: */* 4 Host: oastify.com 5 Connection: Keep-Alive </pre>	

# 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="Microsoft SQL Server Database" is
  <Alias>MS SQL</Alias>
  <ConnectionStringEncrypted>
    9BTRSM1MowV5FXnUekzhJ9F40etUa5KF4VrtfPpcLRf/bK3rV0+ykXJ4Zb
    11h0Pn3SyNaExUQvV5EyH4Yz3gf8Ram8JNe508a3vdpXr4brv1EtnLXW3S+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)

```

[...]  
[...]  
public string ConnectToDB(Login login)  
{  
    get  
    {  
        return ConfigurationManager.Encrypt(this.ConnectionString,  
            "8pPFR0u4AQ0w7");  
    }  
    set  
    {  
        try  
        {  
            this.ConnectionString = ConfigurationManager.Decrypt(newValue,  
                "8pPFR0u4AQ0w7");  
        }  
        catch  
        {  
            this.ConnectionString = null;  
        }  
    }  
}

```

```

3 Verweis  
private void button3_Click(object sender, System.EventArgs e)  
{  
    var x = ConfigurationManager.ConnectionStrings["DB"].ConnectionString;  
    x = ConfigurationManager.Decrypt(x, "8pPFR0u4AQ0w7");  
    close();  
}

```

## 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? 😞

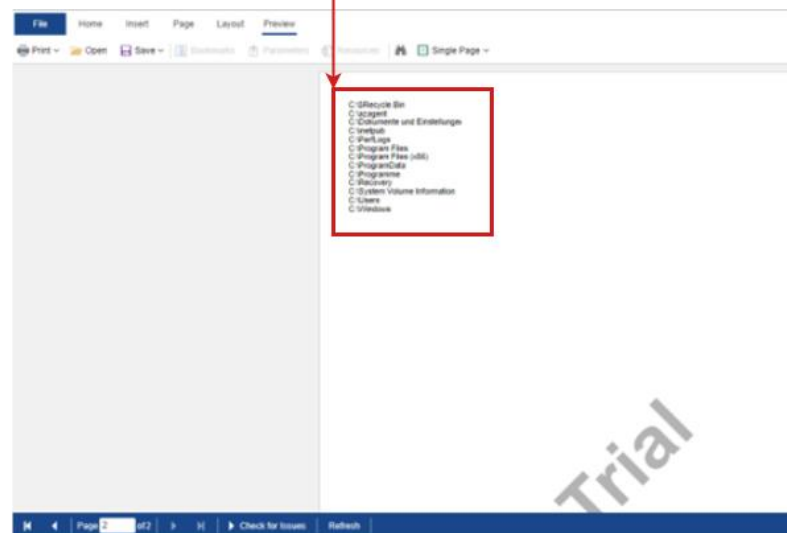
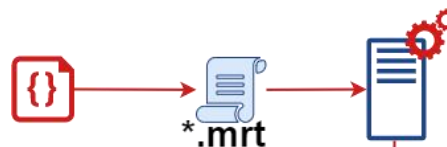
## Problem 4: Remote Code Execution

- ✓ 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





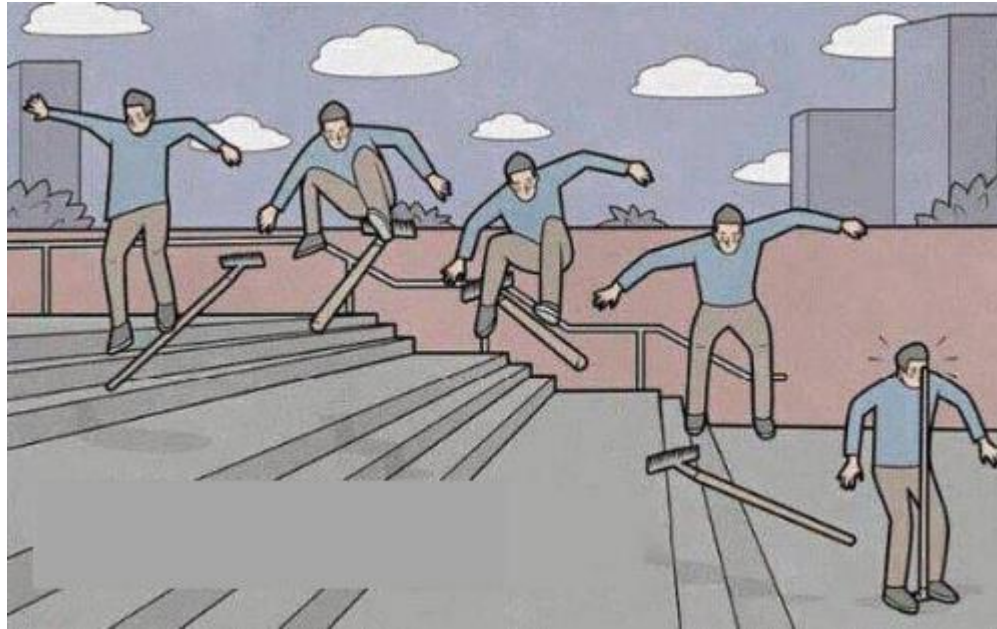
# RCE in der Vergangenheit

- ✓ 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!