TECHNISCHE HOCHSCHULE DEGGENDORF

Bachelor Cyber Security

Kryptologie 2 Projektbericht

Ausarbeitung Cryptochallenge: CurveBall

Autor:

Manuel Friedl, Matrikel-Nr.: 1236626
Christof Renner, Matrikel-Nr.: 22301943

Betreuer:

Prof. Dr. Martin Schramm

Deggendorf – July 20, 2025

Contents

1 Einleitung und Hintergrund

CVE-2020-0601, bekannt als **CurveBall**, ist eine kritische Schwachstelle in der Windows CryptoAPI. Die Sicherheitslücke erlaubt Angreifern, gefälschte ECC-Zertifikate zu erzeugen, indem sie elliptische Kurvenparameter manipulieren, speziell den Generatorpunkt G.

2 Projektübersicht und Zielsetzung

Das Projekt 'SS25-Kryptologie2' bietet eine interaktive Web-basierte Simulation der CurveBall-Schwachstelle. Ziel ist es, diese Sicherheitslücke eigenständig zu simulieren, zu verstehen und praxisnah zu analysieren.

3 Beschreibung der Webinterface-Simulation

Das Webinterface ermöglicht eine intuitive Bedienung der Challenge, unterteilt in die folgenden Schritte:

3.1 Aufbau und Übersicht

Das Interface gliedert sich in drei zentrale Bereiche:

1. Schlüsselgenerierung
2. Zertifikatsgenerierung
3. Validierungssimulation

3.2 Schlüsselgenerierung über das Webinterface

Im ersten Schritt wird über die Oberfläche das Python-Skript `gen-key.py` ausgeführt, welches aus einem gültigen CA-Zertifikat einen manipulierten ECC-Schlüssel erzeugt.

Auflistung 1: Schlüsselgenerierung

```
python gen-key.py USERTrustECCCertificationAuthority.crt
```

3.3 Zertifikatsgenerierung

Im nächsten Schritt erfolgt die automatische Zertifikatsgenerierung via OpenSSL mithilfe des generierten Schlüssels und vordefinierter Konfigurationsdateien.

3.4 Validierungssimulation

Abschließend prüft das Webinterface mit `simulate_vuln_check.py`, ob die erstellten Zertifikate von der verwundbaren Implementierung akzeptiert werden.

Auflistung 2: Simulationsprüfung

```
signature = rogue_key.sign_digest(digest)
if rogue_pub.verify_digest(signature, digest):
    print("Vulnerable validation: Signature accepted.")
```

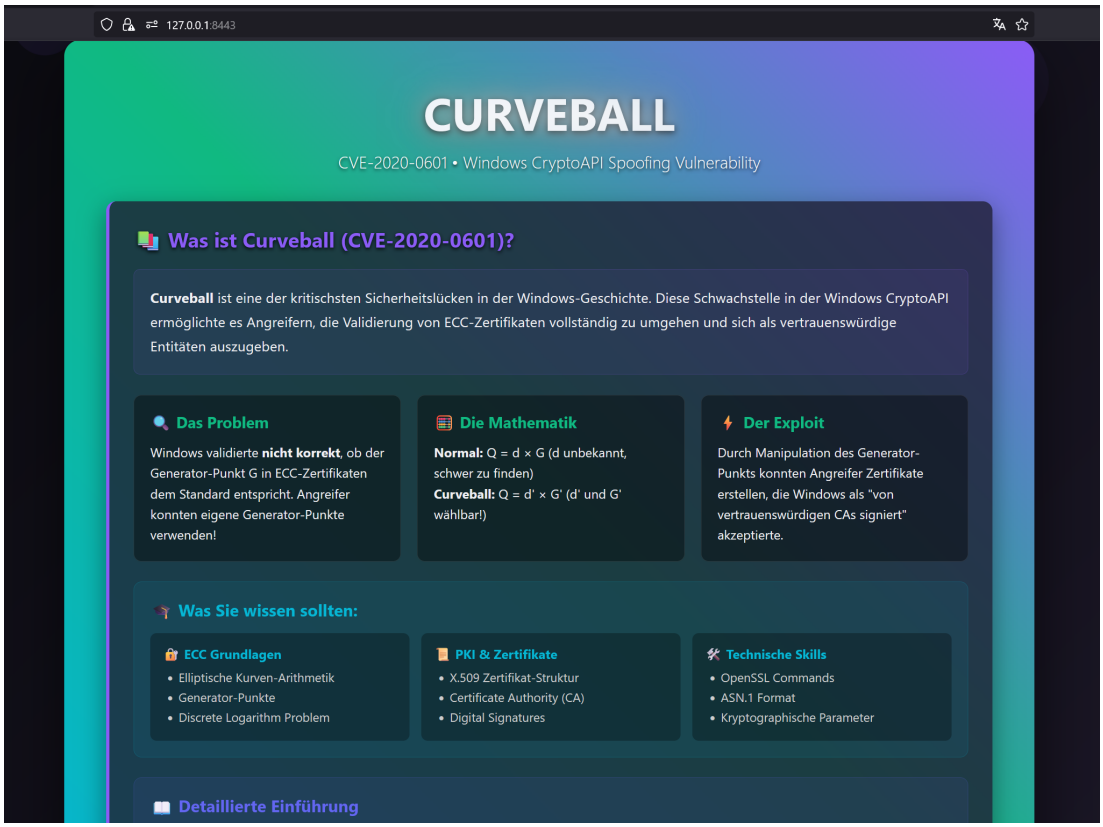

Figure 1: Übersicht des Webinterfaces (Platzhalter)

4 Technische Hintergründe der Schwachstelle

Durch gezielte Manipulation des ECC-Generatorpunkts (G') kann ein Angreifer einen privaten Schlüssel erzeugen, der vermeintlich korrekte Signaturen liefert. Windows CryptoAPI validierte die Herkunft des Schlüssels von G nicht, was die Attacke ermöglicht.

5 Ergebnisse der Simulation

Die Web-basierte Simulation bestätigte eindrucksvoll, wie einfach manipulierte Zertifikate erzeugt und validiert werden konnten. Die erfolgreiche Validierung im simulierten Szenario zeigt die Gefahr solcher unzureichenden Validierungen.

6 Fazit

Das Projekt bietet eine didaktisch wertvolle Möglichkeit, die CurveBall-Schwachstelle praxisnah nachzuvollziehen. Die Webschnittstelle vereinfacht den Lernprozess und verdeutlicht eindrücklich die Gefahren der Schwachstelle.

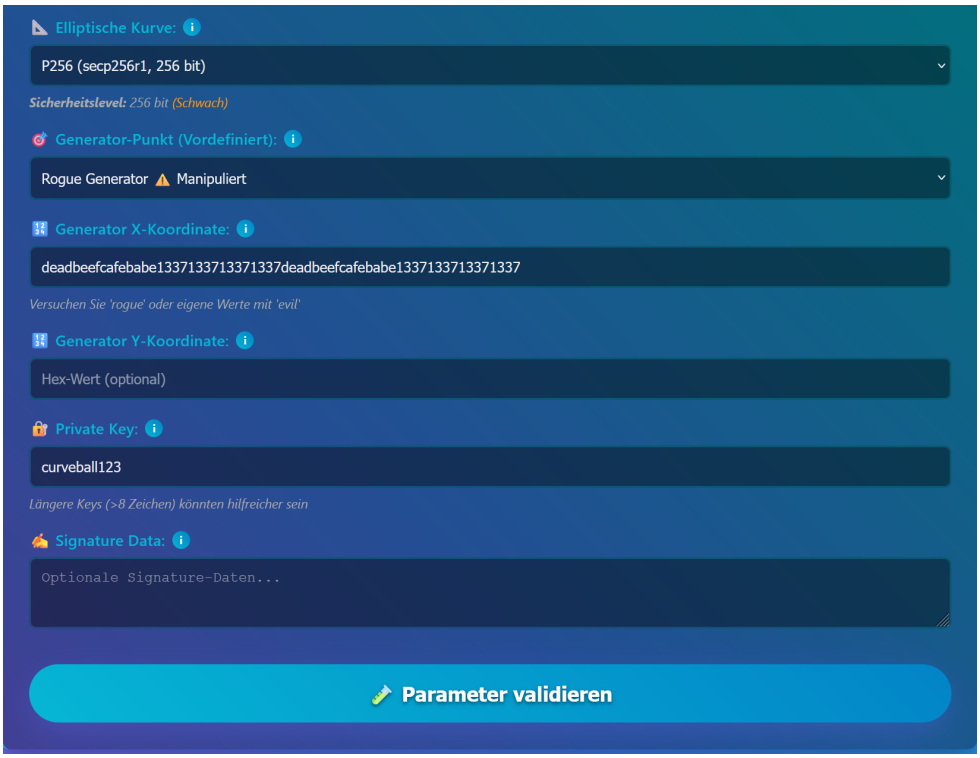

Figure 2: Webinterface Schlüsselgenerierung

Eigenständigkeitserklärung

Hiermit bestätigen wir, dass wir die vorliegende Arbeit selbstständig verfasst und keine anderen Publikationen, Vorlagen und Hilfsmittel als die angegebenen benutzt haben. Alle Teile unserer Arbeit, die wortwörtlich oder dem Sinn nach anderen Werken entnommen sind, wurden unter Angabe der Quelle kenntlich gemacht.

Deggendorf, _____

(Datum) (Unterschrift)

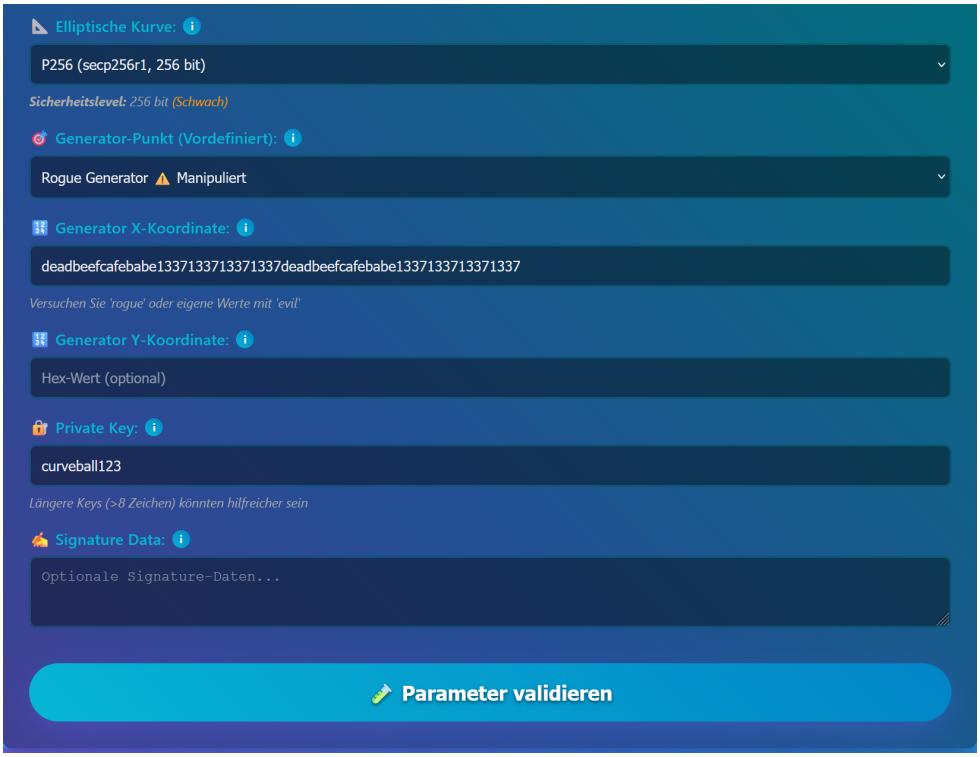

Figure 3: Webinterface Zertifikatsgenerierung (Platzhalter)

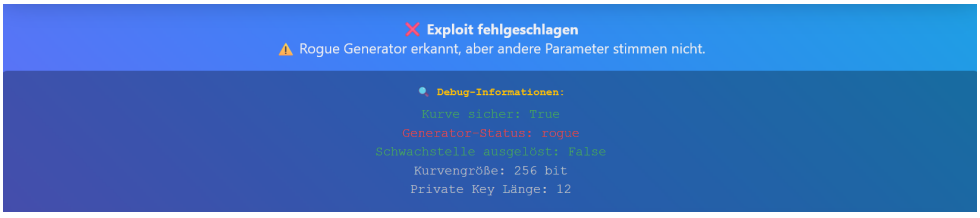

Figure 4: Validierungsergebnis über das Webinterface