IT-Sicherheit | Kryptographie

Alternative Prüfungsleistung

Belehrung

Die Nutzung der angewandten **Techniken** ist ausschließlich **auf** die Alternative **Prüfungsleistung** im Rahmen der Lehrveranstaltung **beschränkt.** Missbrauch ist ggf. strafrechtlich relevant.

S01 | Konsolen-Applikation und Digitale Erpressung | 80 Punkte

Für die Simulation im Kontext der Digitalen Erpressung ist der bereitgestellte Datensatz zu verwenden und eine Konsolen-Applikation mit Eingabezeile zu realisieren. Teilnehmer Clue Less (Opfer) verfügt über ein Bankkonto mit 5000 € Euro Guthaben. Die Teilnehmer Clue Less, Ed (Angreifer) sowie die Miner Bob, Eve und Sam verfügen über ein **Wallet**. Das Wallet ist grundsätzlich an eine Person und keine zentrale Institution gebunden. **Ein Euro entspricht 0,000019 BTC**.

In einem durch MCG signierten Java-Archive report.jar ist eine Ransomware integriert. Dieses Schadprogramm verschlüsselt alle 20 Dateien in einem dedizierten Verzeichnis mit 256-Bit AES. Die verschlüsselte Datei erhält zusätzlich die Dateiendung mcg, wie beispielsweise xy.txt.mcg. Die originären Dateien werden gelöscht. Benutzer Clue Less (CL) gibt in der Zeile den Befehl **launch http://www.trust-me.mcg/report.jar** ein und die Verschlüsselung startet automatisch. Danach erscheint die Meldung „Oops, your files have been encrypted. With a payment of 0.02755 BTC all files will be decrypted.“. CL gibt den Befehl **exchange 0.02755 BTC** ein, die 0.02755 BTC werden dem Wallet gutgeschrieben und das Bankkonto mit dem Äquivalent in Euro belastet. CL gibt den Befehl **show balance** ein und das Guthaben von Bankkonto und Wallet werden angezeigt. CL gibt den Befehl **show recipient** ein und erhält die (anonyme) Bitcoin-Adresse von Angreifer Ed. CL gibt den Befehl **pay 0.02755 BTC** to [address] ein und die Zahlung abgewickelt. CL gibt den Befehl **check payment** ein und erhält den Status „transaction [successful | unsuccessful] durch Auswertung der Blockchain. Bei dem Status „transaction successful“ entschlüsselt das korrespondierende Java Archive report.jar alle 20 Dateien in den originären Zustand im dedizierten Verzeichnis.

S02 | Blockchain | Bitcoin | 80 Punkte

Bitcoin-Adressen sind [i] vorzugw. 34-stellige Zeichenkombinationen[1] o. [ii] alternativ Public-Key. In dem Netzwerk sind drei Miner Bob, Eve und Sam registriert. Bei dem **PoW** wird ein Miner nach dem Zufallsprinzip ausgewählt, es ist **kein** (parallelisierter) **Wettbewerb** zu realisieren. Die Belohnung beträgt 0,025 BTC – in Form eines neu generierten Coins – die der Miner erhält. Durch Digitalen Signaturen ist ein Höchstmaß an Vertraulichkeit sicherzustellen. Der Difficulty-Level (mindestens 3) wird in einer zentralen Konfiguration – realisiert als Enum – definiert. Als Algorithmus für das **Hashing** ist **SHA256** zu nutzen. Für das einmalige Setup der Bitcoin-Blockchain mit der Genesis-Transaktion sowie einem BTC ist eine Person mit dem Pseudonym „Satoshi Nakamoto“ zuständig. **Regeln** des Netzwerkes sind [i] Korrekte Anwendung SHA256 und [ii] Echtheit der Transaktion. Miner fügt dem Block mit der korrespondierenden Transaktion eine weitere Transaktion mit dem definierten Reward 0,025 BTC in Form eines neuen Bitcoin zu.[2] In einem **Logfile** ist der Prozess [01] Transaktion, [02] TX Broadcast, [03] Verifikation, [04] Strukturierung, [05] Proof-of-Work (PoW), [06] Blockübertragung, [07] Verifikation und [08] Hinzufügen des Blocks detailliert zu protokollieren. Die **Blockchain** ist im **Format JSON** zu protokollieren.

1 https://www.novixys.com/blog/generate-bitcoin-addresses-java/

2 https://www.bitcoin.com/bitcoin.pdf | 6. Incentive

Komplexaufgabe S01 und S02 | Aufbau psychologischer Druck | 20 Punkte

Für den Aufbau von psychologischen Druck ist ein (parallelisierter) Timer-Task zu integrieren. Jede Minute seit Verschlüsselung erhöht sich das Lösegeld um 0,01 BTC. Nach fünf Minuten ist keine Entschlüsselung mehr möglich. Jede Minute bis zur vierten erfolgt die Ausgabe „Amount to pay increased by 0,01 to [x] BTC“. In der vierten Minute erfolgt die Ausgabe „Pay [x] BTC immediately or your files will be irrevocably deleted.“. Ignoriert der Benutzer diese Meldung werden nach fünf Minuten seit Verschlüsselung alle (verschlüsselten) Dateien in dem dediziert. Verzeichnis gelöscht.

Wichtige Hinweise für die Bearbeitung

- Durch die sorgfältige und qualitativ hochwertige Bearbeitung[3] der Trainings Playfair Chiffre (PCxx), Enigma (ENxx), S-DES (Otxx), RSA (PKxx), Hill Chiffre (HCxx) können maximal 10 Sonderpunkte zur Anrechnung auf die Komplexaufgabe S01 und S02 generiert werden. **Sonderpunkte** werden **individuell** je Studierenden und bearbeitetes Training angerechnet.
- Die **Bearbeitung**[4] dieser Aufgabenstellung erfolgt **im Team mit zwei Studierenden S01 / S02**.
- **Verwendung geeigneter englischer Begriffe**.
- **Implementierung einer Konsolen-Applikation**, keine Modellierung und keine JUnit-Tests.
- **Zugelassene Bibliotheken**: *[g]json*[5].
- Als **Entwicklung**sumgebung wird [i] **Java SE Development Kit 17.0.2**, [ii] **IntelliJ IDEA Community oder Ultimate 2021.3.2 und** [iii] **gradle 7.3.3** genutzt.
- **Erstellung** einer vollständigen (IntelliJ-Projekt) und unverschlüsselten **7-Zip-Datei** (Kompressionsstärke: Ultra) mit der Bezeichnung **its_ap_2022_[mnr01_mnr02].**
- **Abgabetermin**: So., 06.03.2022 | **Bewertung:** 100 Punkte / Studierenden im Team

3 Vorgehensweise der Ver-/Entschlüsselung ist nachvollziehbar zu dokumentieren.
4 Empfohlener Zeitansatz: maximal 50h / Studierenden
5 https://mvnrepository.com/artifact/org.json/json/20211205 oder https://mvnrepository.com/artifact/com.google.code.gson/gson/2.8.9