BLOCKCHAIN REALLABOR RHEINISCHES REVIER

Anforderungen für sichere Smart-Contracts / Chaincode / DApps

K. Wittek[1], T. Spielmann[1]
[1] Institut für Internet-Sicherheit – if(is), Westfälische Hochschule, Gelsenkirchen, Deutschland
kontakt_reallabor@fit.fraunhofer.de

Ausgangssituation und Problemstellung

Der Begriff "Smart-Contract" wurde durch Ethereum geprägt. Im Kontext von anderen Blockchain-Implementierungen tauchen auch die Begriffe "Chaincode" oder "DApps" auf. Zur besseren Übersicht und Lesbarkeit soll hier unter dem Begriff "Smart-Contract" jeglicher ausführbarer, in der Blockchain gespeicherter, Code gemeint sein, unabhängig von einer konkreten Blockchain-Implementierung.

Bei einem Smart-Contract handelt es sich um ausführbaren Programmcode in einer Blockchain. Für diesen bedarf es einer Reihe von strengen Anforderungen, um diesen sicher und vertrauenswürdig umsetzen zu können. Einerseits bestehen die bereits bekannten Anforderung, welche für jede (verteilte) Anwendung gelten. Doch auf Grund seiner speziellen Eigenschaften gelten darüber hinaus weitere spezifische Anforderungen für Smart-Contracts. Diese ergeben sich aus zwei besonderen Merkmalen:

- Wurde ein Smart-Contract einmal als Transaction in die Blockchain aufgenommen und validiert, kann dieser nachträglich nicht mehr verändert werden. In manchen Blockchain-Implementierungen existieren jedoch definierte Pattern und Workarounds, um Smart-Contract-Code zu aktualisieren, z.B. per Delegation.
- Der Programmcode ist für jeden Teilnehmer der Blockchain einsehbar und somit auch evtl. vorhandene Fehler.
- Der Aufruf eines Smart-Contracts ist ebenso transparent und unwiderruflich wie der Smart-Contract selbst.
- Die erfolgreiche Ausführung eines Smart-Contract Aufrufs ist nicht garantiert und von verschiedenen Parametern abhängig. Gleichzeitig können unter Umständen keine starken zeitlichen Garantien für den Ausführungszeitpunkt gegeben werden.

Anwendungskontext

Die Betrachtung der Sicherheit von Smart-Contracts lässt sich in drei Phasen unterteilen: Programmierung, Veröffentlichung und Aufruf. In jeder Phase können die Anforderungen wiederum in neue und bereits bekannte Anforderungen aufgeteilt werden.

Programmierung:

Diese ist die essentielle Phase. Zu dieser Zeit hat der Entwickler alle Möglichkeiten seinen Smart-Contract sicher und robust zu gestalten. Daher stellen sich hier unter anderem folgende sicherheitskritische Frage:

- Hat die gewählte Programmiersprache Auswirkungen auf die Sicherheit?
- Welche Gefahren von unberechtigten Lese- und Schreibzugriffen auf Variablen bestehen bei der parallelen Ausführung verschiedener Smart-Contracts auf derselben Node?
- Wie riskant sind Aufrufe von Blockchain-Oracles, Code der off-chain gespeichert und ausgeführt wird (Stichwort: Abgabe der Kontrolle über den Ablauf des Prozesses)? Wie können Vertrauensketten geschaffen werden?
- Wie wichtig ist die Unabhängigkeit des Smart-Contracts von seiner Position in der Blockchain für den Ablauf und die Sicherheit?
- Wie wichtig ist die Vertrauenswürdigkeit des Compilers und der ausführenden Umgebung für die Sicherheit des Smart-Contracts und der involvierten Daten?

Veröffentlichung:

Die Veröffentlichung des Smart-Contracts ist der kritischste Zeitpunkt in der Entwicklung. Es stellt einen "Point of no return" dar. Dazu wird der Programmcode in einer Transaktion

gespeichert und entsprechend des jeweiligen Algorithmus in einem Block in der Blockchain verstetigt. Ab diesem Zeitpunkt ist er für alle berechtigten Teilnehmer sichtbar und ausführbar.

Ausführung:

Nicht nur der Smart-Contract selbst ist für berechtigte Teilnehmer voll einsehbar, sondern auch dessen Aufrufe durch andere Teilnehmer. Der Aufruf selbst und die ggf. übergebenen Parameter werden in einer Transaktion in der Blockchain gespeichert. Hieraus ergeben sich weitere Fragestellungen innerhalb dieser Forschungsfrage:

- Wie riskant ist eine öffentliche Sichtbarkeit des Aufrufs und der übergebenen Parameter?
- Wie kann sowohl der Aufruf als auch die Übergabe der Parameter sicher gestaltet werden?
- Wie riskant ist die öffentliche Sichtbarkeit der Inhalte privater Variablen eines Smart-Contracts?
- Wie lässt sich ein Update oder die Deaktivierung eines Smart-Contracts sicher umsetzen?

Lösungsansätze

Die meisten Lösungsansätze für die oben angeführten Probleme basieren auf der Beschränkung der Skripte oder auf Verbote bestimmter Konzepte. So ist je nach Blockchain-Technologie und Programmiersprache eine Begrenzung des Skriptlänge im Hinblick auf Speichergröße oder Anzahl der auszuführenden Befehle möglich. Auch potentiell verwundbare Konzepte wie Schleifen sind teilweise nicht erlaubt. Des Weiteren ist eine Beschränkung auf bereits bekannte und ausführlich getestete Programmiersprachen möglich. Somit sollen Neuentwicklungen mit "Kinderkrankheiten" nicht zu Schwachstellen führen. Solange die verwendeten Programmiersprachen und die ausführende Laufzeitumgebung allerdings Turing-Vollständig ist, sind eine große Menge an Fehlerklassen bei der Programmierung möglich.

Doch neben der genutzten Programmiersprache, sollten auch die unter Umständen genutzten Bibliotheken sicher sein. Des Weiteren kann es sehr empfehlenswert seit sorgfältig entwickelte und ausführlich getestet Pest-Practice-Methoden oder sichere Pattern zu verwenden.

Für die zur Laufzeit auftretenden Probleme ist es möglich sowohl den Code des Smart-Contracts, als auch die Übergabe der Parameter zu verschlüsseln. Dies schränkt ebenfalls die Anzahl der Teilnehmer ein, welche die Ausführung umsetzen können, da das entsprechende Schlüsselmaterial bei der ausführenden Node vorhanden sein muss.

Literatur

1. Arunkumar, S. & Muppidi, S., 2019. Secure your blockchain solutions. [Online]

2. Available at: https://developer.ibm.com/technologies/blockchain/articles/how-to-secure-blockchain-solutions/.

3. Bundesamt für Sicherheit in der Informationstechnik (BSI), 03/2019 "Blockchain sicher gestalten - Konzepte, Anforderungen, Bewertungen", https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Krypto/Blockchain_Analyse.pdf?__blob=publicationFile&v=5

4. Norbert Pohlmann, 2019, "Cyber-Sicherheit - Das Lehrbuch für Konzepte, Prinzipien, Mechanismen, Architekturen und Eigenschaften von Cyber-Sicherheitssystemen in der Digitalisierung", Kapitel 14.6 Sicherheit und Vertrauenswürdigkeit der Blockchain-Technologie

5. L. Luu, J. Teutsch, R. Kulkarni und P. Saxena, Demystifying Incentives in the Consensus Computer, Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, S. 706–719, 2015.

Forschungseinrichtungen

Zur Bearbeitung dieser Forschungsfrage sind Kompetenzen aus folgenden Bereichen erforderlich:

- Cyber-Security
- Informatik (Verteile-Systeme, Datenbankanken, Sprachdesign)