Ensuring Resource Trust and Integrity in Web Browsers using Blockchain Technology

Benjamin Leiding[1] Clemens H. Cap[2]

[1]University of Göttingen, Germany
benjamin.leiding@cs.uni-goettingen.de

[2]University of Rostock, Germany
clemens.cap@uni-rostock.de

June 11, 2018

About Me

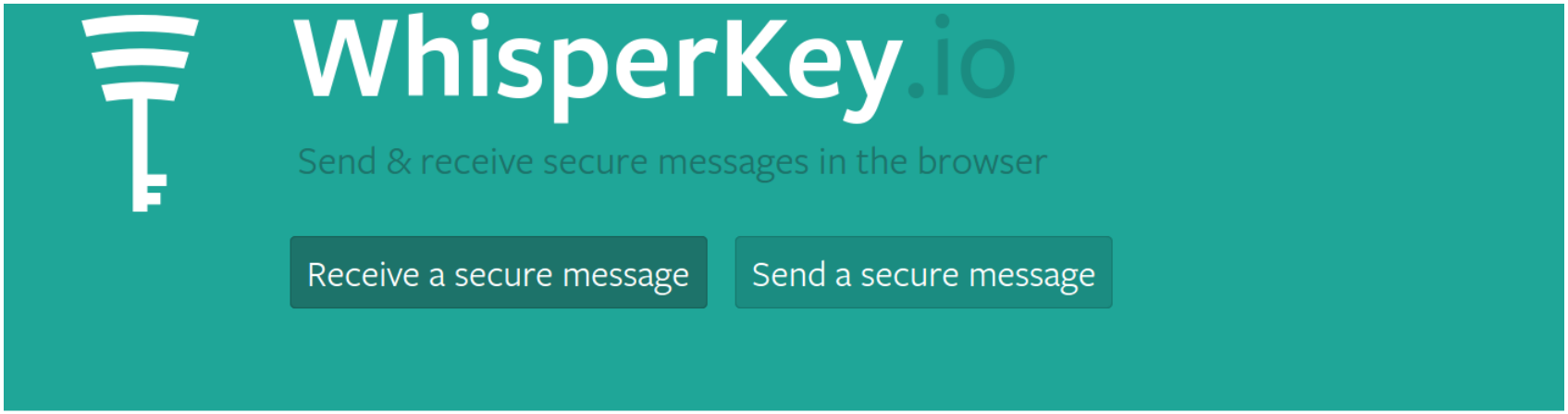
Academic

- PhD student (University of Göttingen, Germany)
- Security/Privacy background
- Current research areas:
  - (Self-Sovereign) identity systems and authentication protocols $\rightarrow$ Authcoin protocol.
  - Architectures and designs of blockchain systems and applications.
  - Application of blockchain technology, e.g. Blockchain-based academic peer-review systems.
  - M2M economy among autonomous agents

Overview

Introduction

Introduction



Source: `https://www.whisperkey.io`

Problem Statement

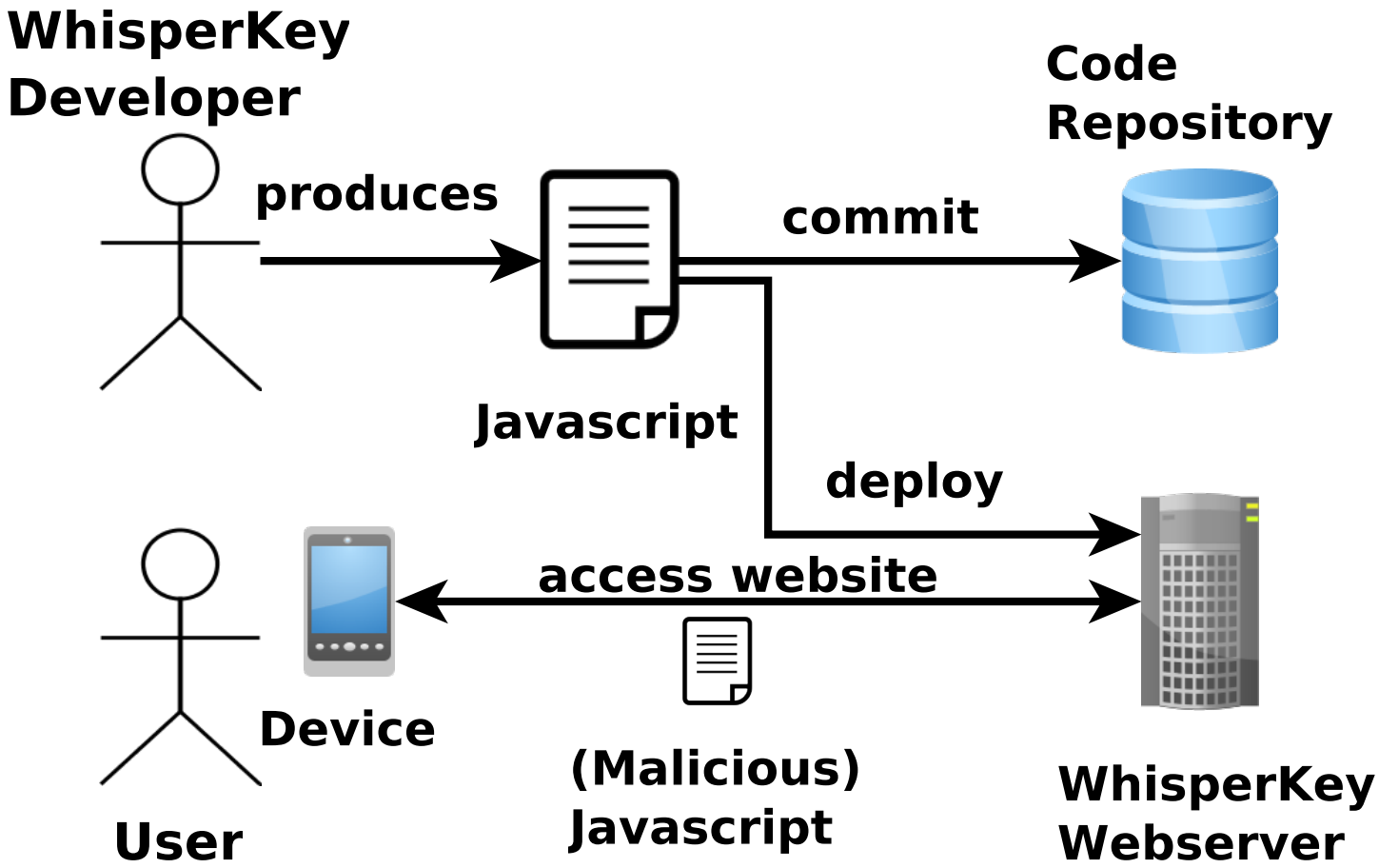


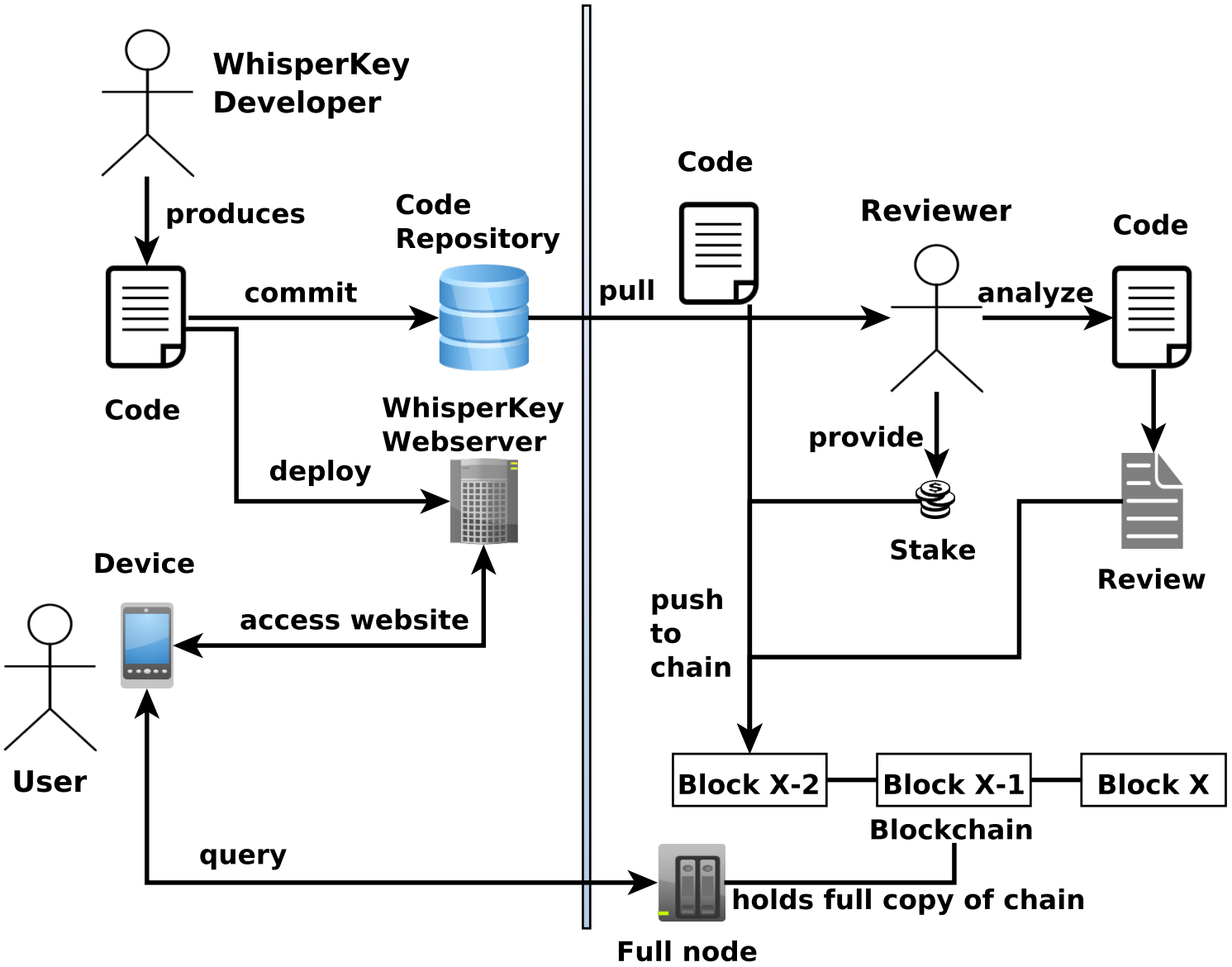
Figure: Server-side code poising attack.

State of the Art

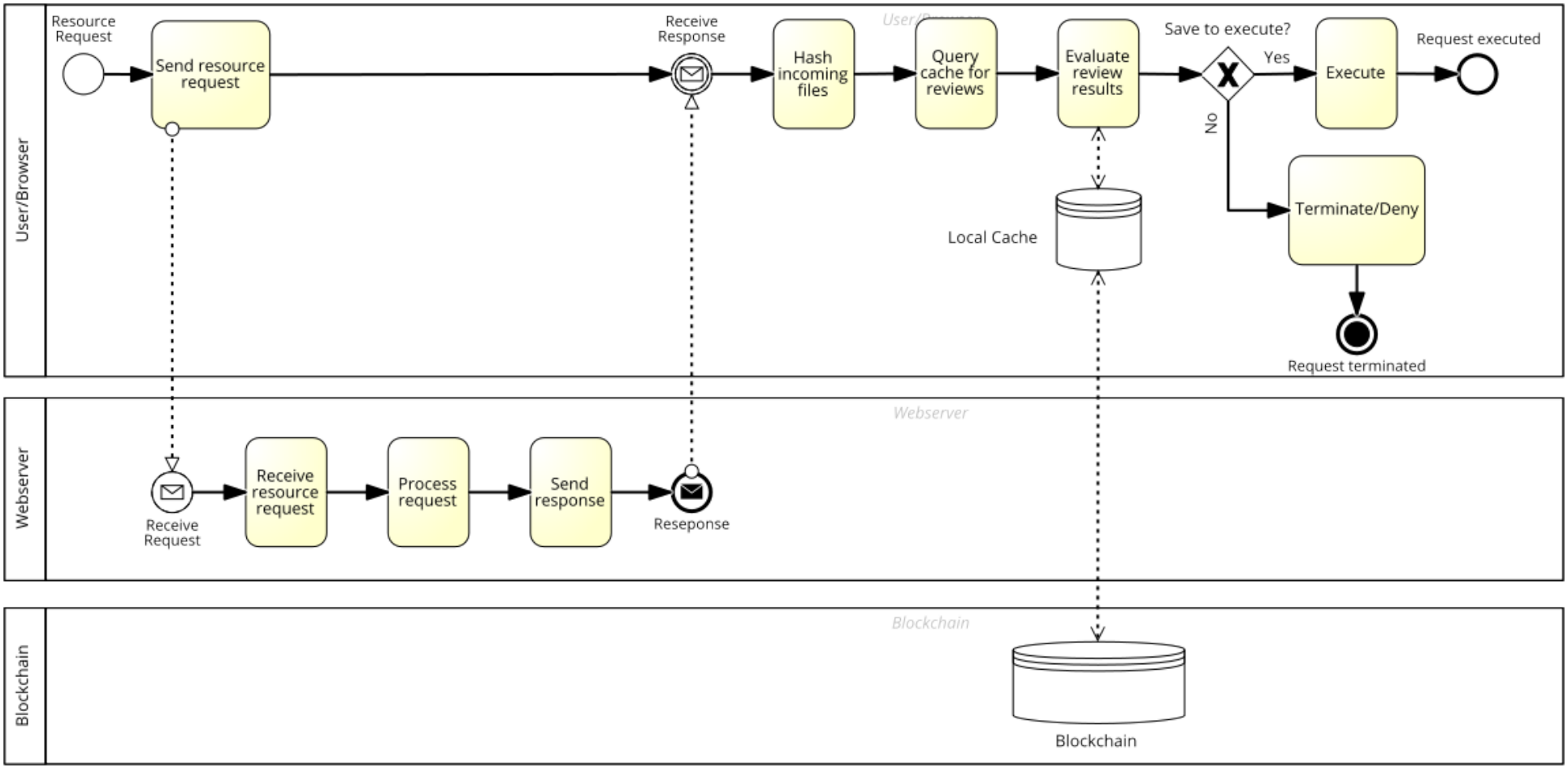
- Checking incoming JS manually?
- Disabling JS?
- CDN subresource integrity via hash-codes $\rightarrow$ Only protects against attacks from the CDN, not the server/programmer.

Conceptual Overview

General Overview



Client Request Processing



BPMN representation of the local client requesting and processing an incoming file.

Peer Review Process

Peer Review

Review Report

- ID
- Project name
- Project description
- Link to resource/repository
- Hash of the reviewed committed version
- Resource itself
- Reviewer information (ID, etc.)
- Detailed report on review results
- Boolean value $\rightarrow$ secure vs. insecure

Similar to academic peer-review process.

Conflict Resolution

What is a good and objective criteria for insecure code?

How to settle disputes on what constitutes a vulnerability?

Conflict Resolution - CVEs

The Common Vulnerabilities and Exposures (CVE) system provides a reference-method for publicly known information-security vulnerabilities and exposures.

CVE-ID

CVE-2018-7747 Learn more at National Vulnerability Database (NVD)
• CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information

Description

Multiple cross-site scripting (XSS) vulnerabilities in the Caldera Forms plugin before 1.6.0-rc.1 for WordPress allow remote attackers to inject arbitrary web script or HTML via vectors involving (1) a greeting message, (2) the email transaction log, or (3) an imported form.

References

Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.

- EXPLOIT-DB:44489
- URL:https://www.exploit-db.com/exploits/44489/
- MISC:http://packetstormsecurity.com/files/147257/WordPress-Caldera-Forms-1.5.9.1-Cross-Site-Scripting.html
- CONFIRM:https://wordpress.org/plugins/caldera-forms/#developers
- CONFIRM:https://calderaforms.com/2018/03/caldera-forms-1-6-is-here/
- CONFIRM:https://calderaforms.com/updates/caldera-forms-1-6-0/#security

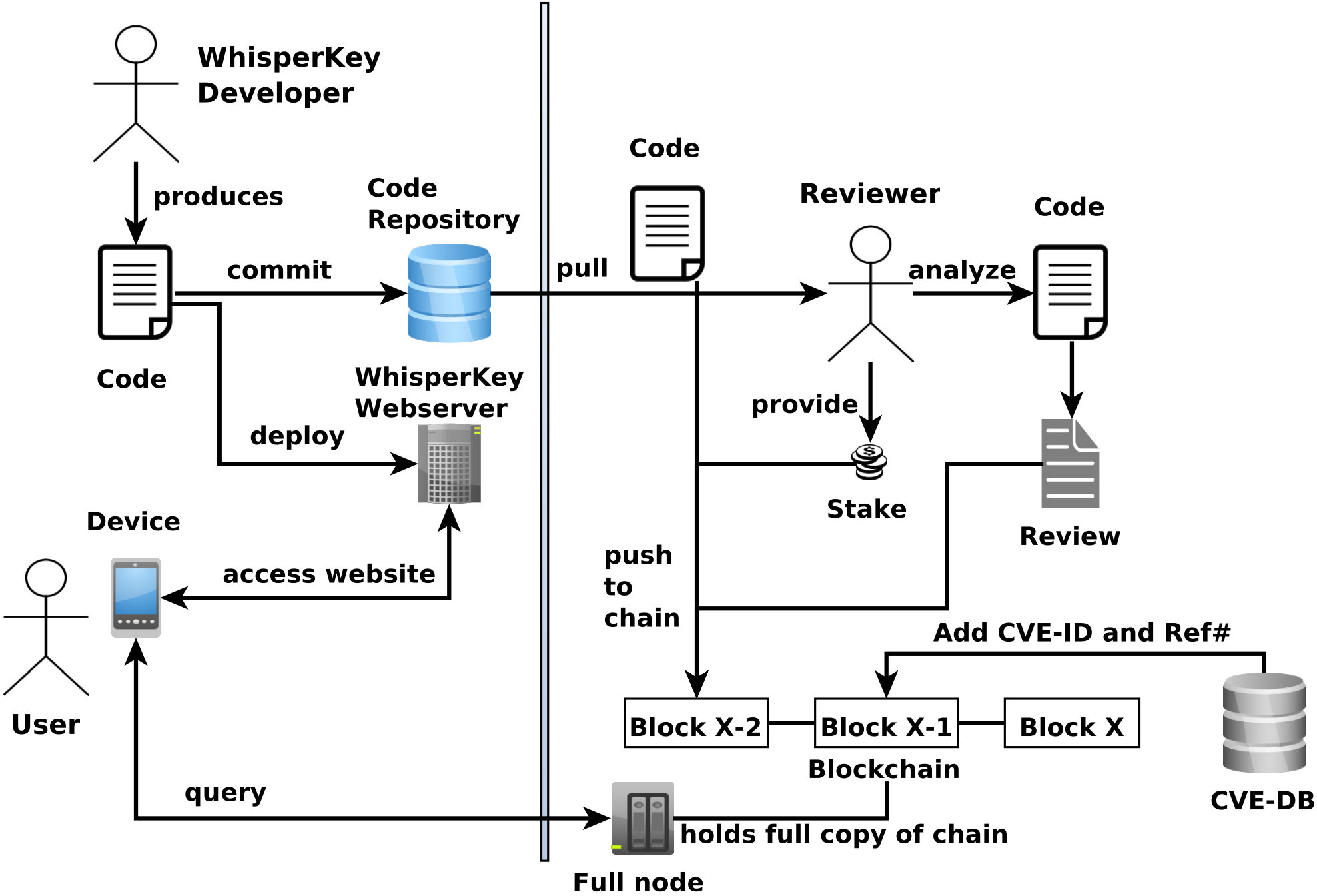
Assigning CNA

MITRE Corporation

Date Entry Created

20180307 Disclaimer: The entry creation date may reflect when the CVE ID was allocated or reserved, and does not necessarily indicate when this vulnerability was discovered, shared with the affected vendor, publicly disclosed, or updated in CVE.

General Overview



Incentive Mechanism

- **Incentive for user:**
  - Enhanced security
- **Incentive for reviewer:**
  - Rewards (Steem[1]-like token system)
  - Reputation
  - Bounties by developers/users
- **Incentive for programmer/software provider:**
  - Enhanced security of product based on external reviews.
  - Trustworthiness

[1] https://steem.io/

Issues and Disadvantages

- Depending on the stake-size, it might be still worth loosing the stake to launch a successful attack.
- Incentivize reviewers.
- Not all vulnerabilities are listed as CVEs.
- Definition of vulnerability or insecure code.
- Small and unknown projects might not be reviewed at all.

Conclusion and Future Work

Conclusion

Take Home Message

- Enable secure delivery and execution of code.
- Prevent code manipulation by binding code to a review via a hash.
- Browser validates review status, hash and code $\rightarrow$ Insecure code is not executed.
- Concept is versatile and can be used for all kind of documents and software.

Future Work

Research Tasks

- Prototype implementation starts in July (Browser extension + IOTA/Ethereum).
- How to ensure that a reviewer invests sufficient time to produce a quality review? (Proof-of-X?)
- Apply the same methodology in a more general way $\rightarrow$ Resource trust and integrity of files.
- Reputation-driven distributed autonomous organization (DAO) for resource reviews based on an abstract review protocol.
- Dispute resolution using a Semada[2]-like betting pool.

[2] `http://semada.io/`

Questions?