Securing Hyperledger Fabric

Monitoring and analyzing blockchain deployments using Splunk

Christopher Cordi
Chris Cordi
Software Engineer
Splunk Blockchain Team
Why Splunk Blockchain?

Blockchain solutions hold a lot of promise but add new and difficult infrastructure, application and security requirements.

• Hard to gain **end to end visibility** between all components
• Many options to choose for infrastructure including **on-prem, cloud, hybrid, unmanaged, managed and multi-organization**.
• **Diverse set of data sources** both with differing formats and velocities.
• Different blockchain technologies and platforms
• **Interoperability** and collaboration between consortiums
• **Disparate tools** for logging, metrics, tracing, transaction analytics and security.
• **Different tools (if any)** for load testing and development performance vs production monitoring and investigation.
Hyperledger Fabric Overview

Hyperledger Fabric is designed to enable secure collaboration between multiple organizations operating with limited trust.

Despite the security improvements Hyperledger Fabric provides, deployments still require careful configuration and monitoring to ensure that they are operating securely.

We’re going to review different types of threats that Hyperledger Fabric operators should consider and discuss how to mitigate them.
Assets to Protect

- Data
  - IP or otherwise sensitive data
- Hardware
- Software/Applications
Threats to Hyperledger Fabric Networks

As a permissioned blockchain, Hyperledger Fabric’s network threats can differ from popular permissionless chains.

Some attacks are common to all distributed systems like Denial of Service (DoS) or consensus manipulation.

Other attacks target specific components in a Hyperledger Fabric network, such as the Membership Service Provider (MSP).
Threats: Denial of Service (DoS)

Denial of Service attacks disrupt the network’s availability and are a threat to any distributed system.

Many different attacks can result in denial of service, which makes it difficult to proactively prevent.

Mitigation:

- collecting performance metrics, such as transaction throughput and latency, to detect compromised availability early on
Attacks on the network consensus include DoS and transaction reordering attacks.

Mitigation:

- Logging threat indicators, such as leadership elections and transaction latencies, can speed up detection.
Threats: MSP Compromise

The Membership Service Provider (MSP) is able to modify access control to the network and, if malicious, could deny service and perform Sybil attacks.

The MSP may be compromised by a rogue insider or through private key theft, which may only be detectable after exploitation.

Mitigation:

- Follow Key Management and Access Control best practices
- Logging and alerting MSP actions, such as certificate creation and revocation
Threats: Smart Contract Exploitation

Smart contract exploits can compromise business logic and network performance.

In addition to ordinary programming logic bugs, common errors can also stem from inappropriately handling concurrency or nondeterminism.

Mitigation:

- Design smart contracts with security in mind
- Smart contract analysis tools like the Hyperledger Lab Chaincode Analyzer
- External Security Audits
- Monitor performance and usage of the smart contract to detect anomalous behavior.
## Key Data Sources

<table>
<thead>
<tr>
<th>Threat</th>
<th>Indicators</th>
<th>Data Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Denial of Service</td>
<td>Tx throughput &amp; latency</td>
<td>Block headers, Node metrics</td>
</tr>
<tr>
<td></td>
<td>Block latency</td>
<td></td>
</tr>
<tr>
<td></td>
<td># Senders</td>
<td></td>
</tr>
<tr>
<td></td>
<td># Open connections</td>
<td></td>
</tr>
<tr>
<td>Consensus Manipulation</td>
<td>Changes in chain config</td>
<td>Blocks</td>
</tr>
<tr>
<td></td>
<td>Leadership Elections</td>
<td>Node logs</td>
</tr>
<tr>
<td>Ledger Manipulation</td>
<td>Orphaned blocks</td>
<td>Block headers</td>
</tr>
<tr>
<td>Smart Contract Exploitation</td>
<td>Scanned Vulnerabilities</td>
<td>Vulnerability Scanner</td>
</tr>
</tbody>
</table>
Additional Threats & Data Sources

Effective Monitoring Requires Both On Chain & Off Chain Data

Threats

- Front End / Application Vulnerability
- Key Theft
- Smart Contract Vulnerability
- Denial of Service
- Network Partitioning
- Malicious Consensus Behavior

Off Chain

- Threat Intelligence
- P2P
- Node, OS Logs & Metrics
- External APIs
- Networking
- Key Management

On Chain

- Blocks
- Accounts
- Events
- Transactions
- Addresses
- Smart Contracts
Data-to-Everything Platform

IT
Security
Observability

Logs
Metrics
Traces
Ledger
Splunk App for Hyperledger Fabric

Gain full observability around Hyperledger Fabric environments

INFRASTRUCTURE HEALTH & MONITORING  |  ANALYZE LEDGER DATA  |  ACT ON CHAINCODE EVENTS

Gain observability into the Consortium
Unify monitoring, troubleshooting, investigation, and take action on Hyperledger Fabric components across organizations and multi-cloud environments.

Reduce MTTR by combining logs, metrics and traces
Remove data silos and the need for multiple tools when troubleshooting and optimizing the network.

Get to production faster and securely
Understand the performance of your development and monitor configuration changes.

https://splunkbase.splunk.com/app/4605
Getting Started with Splunk App for Fabric

We can easily analyze Hyperledger Fabric’s ledger, log, and metric data with the following tools.

- Splunk Connect for Hyperledger Fabric grabs ledger and metric data from a Hyperledger Fabric deployment.
- The Splunk Docker logging driver or Splunk Connect for Kubernetes can be used to send container logs to Splunk.
- Finally, Splunk App for Hyperledger Fabric visualizes this data in Splunk.

See https://github.com/splunk/fabric-logger for an example docker-compose incorporating these tools.
Getting Data In
Example Security Dashboard

- **Transaction Latency**: 43 ms
- **Transactions Per Second**: 1,300
- **Unique Transaction Vendors**: 2
- **Net gRPC Connections Open by Node**
- **Orphaned Blocks (Drilldown)**: 0
- **Consensus Leader Changes**: 0
Detecting DoS Attacks

What would detect during a DoS attack?

In this scenario, an authorized user has their keys compromised and begins spamming the network with transactions.

We will be paying particular attention to the transaction latency, throughput, and number of open connections.

First, we’ll look at the normal case where a single client is sending 10 transactions per second.
Baseline Measurements

Normal case: a single client is sending 10 transactions per second.
Adversary Scenario

Next, we’ll have a single client open up persistent 1000 connections each performing 1 query per minute.

We expect to see:
- transaction latency starts to increase
- transactions per second decreases
- the number of open connections increases.
Adversary Scenario Results

At this point, it may be difficult to determine if this is reflective of a high period of load, misconfiguration, or a denial of service attack.
Adversary Scenario Results

Drilling Down on Errors

We can investigate further in the Infrastructure Health and Monitoring Dashboard, where we see connection and I/O timeout errors.
Adversary Scenario Results

Because we noticed a large number of open connections, we should query Splunk to see the distribution of gRPC message subjects and addresses.

When we perform this search we see a large discrepancy in message count -- indicating that “User1@buttercup.example.com,L=San Francisco,ST=California,C=US” is likely compromised or misconfigured and should be investigated further.
More Integrations

Splunk Application Performance Monitoring (APM) real-time observability

Splunk On-Call intelligent outage alerting

Splunk Enterprise Security enrich and prioritize notables based on risk

Splunk SOAR automate parts of the incident response process.
S&P Case Study

Case Study
Splunk correlates data across all datasets—including Hyperledger Fabric

https://www.hyperledger.org/learn/publications/splunk-sp-case-study
Thank You

splunkdlt.com