Continuous Authentication and Distributed Session Management

Mance Harmon
CEO and Co-founder
Swirlds Inc.
@ManceHarmon
@Swirlds
We need a session ‘kill switch’

...that works for all protocols and client types

WHEN...
• Employment Termination
• Lost or Stolen Devices
• Elevated Risk

...AND for Continuous Authentication
Session Types

Relying Parties:
(Applications / APIs)

Identity Provider:

Authentication Session

Application Session #1

Application Session #2

Application Session #3
Front Channel Solutions

Browser-based communication
• iFrame
• Form Post / Redirection
• Logout Images
Problems
• No guarantees
• Unknown state

Browser Cache
- App1 Session Cookie
- App2 Session Cookie
- App3 Session Cookie
- IdP Session Cookie

Web Browser

Application 1
Application 2
Application 3
Back Channel Solutions

Direct API: IdP → RP

Problem: Correlating users with application sessions
## The Ideal Solution

<table>
<thead>
<tr>
<th>Sharing State</th>
<th>Verification</th>
<th>Front Channel</th>
<th>Back Channel</th>
<th>Ideal Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### Notes
- Sharing state and verification are both necessary for an ideal solution.
- If either is not satisfied, the solution is not ideal.
IDP Polling (One proposal for OIDC)

- **JS Session Management Client**
  - IdP loads JS for each application
  - JS polls IdP to check status of authN session
  - Access Token only available if approved by IdP

- **Problems**
  - Only works with single-page applications
  - IdP must be able to scale polling service
  - IdP must ensure polling service is active
    - When not active, allow / deny access?
Distributed Session Management

- Advantages
  - Independent of identity protocol
  - Independent of client type
  - Status lookups are local
    - No additional load on IdP
- Disadvantages
  - Additional bandwidth needed to replicate session transactions
  - RPs must add check of local session database to workflow
Advanced Features

- Session Extension
- Session Suspension
- Fine-grained, Dynamic Attributes
  - Level of Assurance
- Continuous Authentication
  - Clients and Applications can pass signals to risk engine
  - “Kill Switch” needed for infinite sessions
Continuous Authentication = Logout By Exception

Single Logout
“Kill Switch”

Enables
Continuous Authentication
Sessions with no expiry

Requires
Use Case Requirements

- Fine-grained, trusted consensus timestamps
  - Cryptographic Proof of Receipt
  - Cryptographic Proof of Transmission
- Resilience to DoS attacks
- Immutable record for audit
- High throughput (transactions per second)
- High availability (no single point of failure)
- Low computation cost
- Scalable to large numbers of network members
Choosing a Consensus Algorithm

Categories of distributed consensus algorithms:

- Leader-based Systems
  - PBFT, Paxos, RAFT, TenderMint
  - Non-Proof of Work Blockchain
- Proof of Work Blockchain
  - Bitcoin, Ethereum
- Hashgraph
Leader-based Consensus

- Many variations
  - Includes Non-PoW Blockchain

- Advantages
  - Low computation requirements
  - Proof of Transmission
  - Immutable Audit
  - High Throughput
    - 1000s of tps, seconds latency
  - High Availability

- Disadvantages
  - Designed for ‘fault’ tolerance, not ‘attack’ tolerance
  - Susceptible to DoS attacks
  - No proof of receipt
  - No consensus timestamps
  - Moderate Scalability
    - Max Nodes: ~100
Proof of Work Blockchain

- **Advantages**
  - Proof of Transmission
  - DoS Resistance
  - Immutable audit trail
  - High throughput possible
  - High availability
  - Scalability

- **Disadvantages**
  - High computation requirements
  - Coarse-grained timestamps
  - No proof of receipt
Hashgraph Consensus

- Satisfies all requirements
  - Consensus Timestamps
  - Proof of Receipt
  - Proof of Transmission
  - DoS Resistance (Gossip Protocol)
  - Immutable Audit
  - High Throughput (>400,000 tps)
  - High Availability
  - Low computation (No PoW)
  - Scalable (1000 nodes)
## Summary: Analysis of Requirements

<table>
<thead>
<tr>
<th>Feature</th>
<th>Server</th>
<th>Leader</th>
<th>PoW Blockchain</th>
<th>Hashgraph</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consensus Timestamps</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>✔</td>
</tr>
<tr>
<td>Proof of Receipt</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>✔</td>
</tr>
<tr>
<td>Proof of Transmission</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>DoS Resistant</td>
<td>×</td>
<td>×</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Immutable Audit</td>
<td>×</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Throughput</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Fault Resistant</td>
<td>×</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Computation Cost</td>
<td>✔</td>
<td>✔</td>
<td>×</td>
<td>✔</td>
</tr>
<tr>
<td>Scalable</td>
<td>✔</td>
<td>×</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>
Summary

- A distributed session database moves complexity to a layer below identity protocols
- A mechanism for shared state opens opportunities beyond session logout
- Ping Identity proposed DSM to OIDF last year
- Open Source implementation of DSM to be released soon; DSM protocol continues to evolve
- When choosing consensus algorithm, start with application requirements, and plan for feature creep
Resources

- Contact Info: Mance@Swirlds.com
- DSM Software Download: dwaite@pingidentity.com
- Blog: Choosing a consensus algorithm
  - www.linkedin.com/in/manceharmon
- www.PingIdentity.com
- www.Swirlds.com