Maintaining PJM Tool Availability

Foluso Afelumo
Concurrent Sessions

- Limit the number of concurrent authenticated sessions (tokens)

Usage Rates

- Limit the request rates against the PJM tools
Concurrent Sessions - Implementation Options

1. Deny access
   - 26th attempt is unsuccessful

2. Invalidate next expiring session
   - Session with longest inactivity time is destroyed on 26th attempt

3. Invalidate oldest session
   - 1st created session is destroyed on 26th attempt
2 Scenarios

- Quick or single interaction transactions like downloading data
- Long running transactions – decision making, multiple interactions
Scenario – Quick transactions

Session 1
Authenticate
Perform work
Idle Time: 1 min

Session 2
Authenticate
Perform work
Idle Time: 2 min

Session 3
Authenticate
Perform work
Idle Time: 30 sec

Session 25
Authenticate
Perform work
Idle Time: 1 min
### Impact – Quick transactions

<table>
<thead>
<tr>
<th>Session 1</th>
<th>Session 2</th>
<th>Session 3</th>
<th>Session 25</th>
<th>Session 26</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authenticate</td>
<td>Authenticate</td>
<td>Authenticate</td>
<td>Authenticate</td>
<td>Authenticate</td>
</tr>
<tr>
<td>Perform work</td>
<td>Perform work</td>
<td>Perform work</td>
<td>Perform work</td>
<td>Perform work</td>
</tr>
<tr>
<td>Idle Time: 1 min</td>
<td>Idle Time: 2 min</td>
<td>Idle Time: 30 sec</td>
<td>Idle Time: 45 sec</td>
<td>Idle Time: 0 sec</td>
</tr>
</tbody>
</table>

Least recently used session (#2) is invalidated
Solution – Quick transactions

Session 1
Authenticate
Perform work
Log out

Session 2
Authenticate
Perform work
Log out

Session 3
Authenticate
Perform work
Log out

Session 25
Authenticate
Perform work
Log out
authenticate(username, password) {
    Get token from sso.pjm.com/access/authenticate
    Return Token
}

logout(token) {
    Log out token from sso.pjm.com/access/logout
}

transaction 1(){
    token = authenticate(username, password)
    do work
    logout(token)
}

transaction n(){
    token = authenticate(username, password)
    do work
    logout(token)
}

Main(){
    transaction 1(username, password)
    transaction n(username, password)
}
Scenario – Long running transactions

Session 1
Authenticate
Perform work
Idle Time: 1 min
Perform work

Session 2
Authenticate
Perform work
Idle Time: 2 min
Perform work

Session 3
Authenticate
Perform work
Idle Time: 30 sec
Perform work

Session 25
Authenticate
Perform work
Idle Time: 1 min
Perform work
Impact – Long running transactions

- Least recently used session (#2) is invalidated
- Perform work step will fail if it starts after invalidation occurs – re-authentication will be required
Solution – Long running transactions

Session 1
Authenticate

Pass session along

Transaction 1
Perform work

Transaction 2
Perform work

Transaction 3
Perform work

Transaction n
Perform work

Session 1
Logout
authenticate(username, password) {
    Get token from sso.pjm.com/access/authenticate
    Return Token
}
transaction 1(token){
    do work
}
transaction 2(token){
    do work
}
logout(token) {
    Log out token from sso.pjm.com/access/logout
}
Main(){
    token=authenticate(username, password)
    transaction 1(token)
    transaction 1(token)
    logout(token)
}
Concurrent Sessions - Implementation

- October
  - Begun connecting with users
  - 10/24 - Implemented in Train

- November
  - 11/29 - Implement in Production
• Corrections
  – Markets Gateway: 1600 requests/minute
  – Oasis.pjm.com: 1200 requests/minute