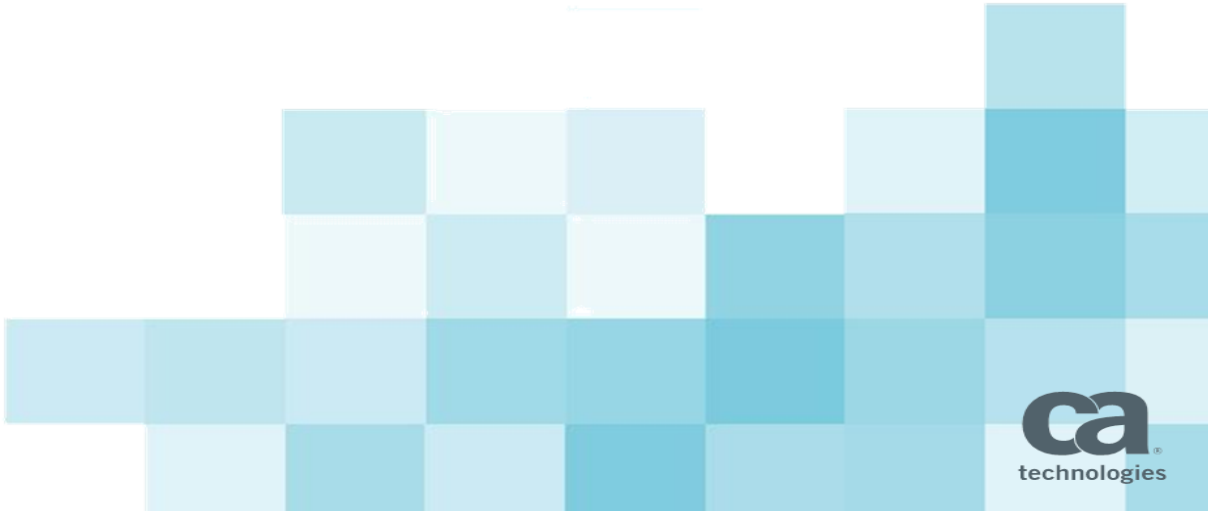
CA Single Sign-On 12.52 SP1 CR5 Enhancements

Stephan McQuiggan, Sr. Principal Support Engineer
Aaron Berman, Sr. Advisor

May 2016



ca technologies

New Enhancements in a CR?

- CR releases do contain a large number of bug fixes, but also contain new features
- CR4 contained:
  - Multiple ACO for IIS Agent
  - Turn off Authorization calls
  - IWA for Office 365 Thick Clients
- CR5 contains new enhancements around session store, performance metrics and visibility of slow transactions

Why use a Session Store?

- Additional Security
  - Mark and remember a session
  - Once user logs out session cannot be used – even from same device
- Session Assurance
  - Remember the device fingerprint
- Store Inbound Federation data to present to applications
  - Data not in the user store
  - CA SSO operating entirely in “claims” mode
- Administratively log a user out (with Global Delivery module)
  - Search for users that have active sessions

Session Store Notes

- CA SSO supports Oracle, SQL and CA Directory for session stores
- Session store objects are created when the user logs in and rechecked every time the user accesses a persistent realm
- For large sites this can create additional overhead and cause slowdowns
- Session store is still optional but more and more features require one

- 12.52 SP1 CR5 has several enhancements to reduce session store overhead and make session store easier to use

Reduce Round Trips to Session Store, both ODBC and LDAP

- In prior releases, each and every Authentication, Validation, and Authorization sent to the policy server required a round trip to the session store in order to validate timeouts and update the last access time.
- In the current release, we reuse existing session data in policy server memory if a subsequent request falls within a given 'grace period' interval and session record has not changed, avoiding round trips to the session store. This grace period is called **SessionUpdateGracePeriod**, and defaults to **1 second**.
- This grace period is similar to the existing 'Validation Period' setting but is server side, affecting all policy server requests. The Validation Period affects the cache on agent side, and is only configured on a per Realm basis. *SDK clients, and client side components that do not have agent caches can take advantage of the new SessionUpdateGracePeriod setting.*

Caution: As with validation period, this setting **lengthens the idle timeout**

HKEY_LOCAL_MACHINE\SOFTWARE\Netegrity\SiteMinder\CurrentVersion\SessionServer\SessionUpdateGracePeriod

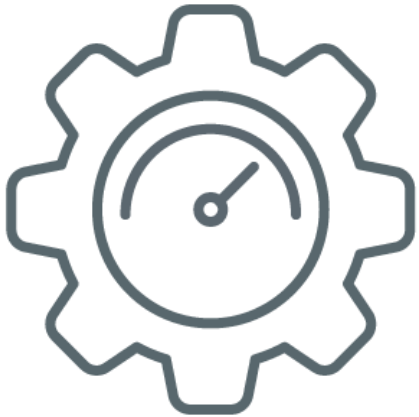
Improve performance of session deletion thread, both ODBC and LDAP

- In prior releases when deleting expired sessions a single query was used to fetch the complete list of all expired sessions. Only when all results for this query were returned did the deletion process begin.
- Now we only scan a first block of **MaintenanceQueryRowLimit** records, delete those, and then scan a second block of records and so on until all expired records are deleted. This setting defaults to 100
- This change reduces access contention between runtime threads writing to session store indexes (login, logout, last access time update), and deletion thread updating the same session store indexes.

HKEY_LOCAL_MACHINE\SOFTWARE\Netegrity\SiteMinder\CurrentVersion\SessionServer\MaintenanceQueryRowLimit

LDAP Session Store Performance Optimizations

- Session deletion uses dedicated connection which does not compete with the other threads performing create/search/update
- Asynchronous delete model with ability to control the number of concurrent delete requests pending with session server at any given time. This value is called **MaxConcurrentDeletes** and defaults to 25.
- We now use a delete control to delete the session and all of its child objects at the same time.



HKEY_LOCAL_MACHINE\SOFTWARE\Netegrity\SiteMinder\CurrentVersion\SessionServer\MaxConcurrentDeletes

Other CR5 enhancements

Transaction Monitoring

- Numerous ways to monitor transactions
- What if I just want to get a list of transactions that took longer than X Seconds to complete?
  - Have to enable tracelog and run trace logs through the log analyzer that is on the community site
- Goal: What if I could just have a log of transactions that took longer than X seconds?

Logging of Slow transactions

- CA SSO now allows an administrator to define a specific threshold for slow transactions in milliseconds
- All transactions that take greater than this threshold will be logged to the smps.log
- Does not require tracing is enabled
- Default threshold is 5000 ms
- Threshold can be changed by editing :
  HKEY_LOCAL_MACHINE\SOFTWARE\Netegrity\SiteMinder\CurrentVersion\LogConfig\ ExecutionTimeThreshold

This log is a great way to log all transactions that do not meet a defined Service Level Agreement

Logging of Slow Transactions - Example

Exceeded threshold events are printed to smps.log:

```
[20979/2969324432][Mon Feb 08 2016
14:14:53][SmAuthUser.cpp:5385][INFO][sm-log-00000]
Execution time exceeded threshold.
(CSmAuthUser::Authenticate, 7500, 5000, agent=
client=127.0.0.1 server=
resource=/realm1/private.html24 action=GET user=user1)
```

CSmAuthUser::Authenticate exceeded the 5000 millisecond threshold, took 7500 milliseconds to complete

Visibility into Policy Server Statistics

- The smpolicysrv –stats command has been around for a long time but has some key problems
  - Stats accumulate from the time the policy server is started
  - No way to reset
  - The Msgs are tied to agent poll attempts not the number of transactions executed

- The stats command can now be reset and run for a specific interval
  - To restart collecting data: *spolicysrv –resetstats*
  - To generate stats for last collection interval: *smpolicysrv –stats*

Visibility into Policy Server Statistics

Sample output:

6241/2681244560][Wed Feb 03 2016 15:03:07][CServer.cpp:4729][INFO][sm-Server-02030] Thread pool: **Msgs**=58844 **Throughput**=238.235/sec **Response Time**=4.820ms **Wait Time In Queue**=16.886ms **Max HP Msg**=1 **Max NP Msg**=50 **Current Depth**=0 **Max Depth**=50 **Current High Depth**=0 **Current Norm Depth**=0 **Current Threads**=8 **Max Threads**=8 **Busy Threads**=0

- Msgs = total number of requests and not number of *poll attempts*.
- Throughput = number of requests divided by total run time of test
- Response Time = time spent processing transactions divided by number of transactions processed
- Wait Time in Queue = Avg time spent between 'Enqueue' and 'Dequeue'
- Max HP Msg = Maxumum number of queued new agent requests
- Max NP Msg = Maximum number of queued agent transactions (authentication, authorization, validation)
- Currently Depth= Current queue depth for agent transactions

Visibility into Policy Server Statistics

Sample output:

6241/2681244560][Wed Feb 03 2016 15:03:07][CServer.cpp:4729][INFO][sm-Server-02030] Thread pool: **Msgs**=58844 **Throughput**=238.235/sec **Response Time**=4.820ms **Wait Time In Queue**=16.886ms **Max HP Msg**=1 **Max NP Msg**=50 **Current Depth**=0 **Max Depth**=50 **Current High Depth**=0 **Current Norm Depth**=0 **Current Threads**=8 **Max Threads**=8 **Busy Threads**=0

- Max Depth= maximum depth of either the Normal Priority or the High Priority queues
- Current High Depth= Current depth of new agent connection queue
- Current Norm Depth= Current queue depth for agent transactions
- Current threads= Current threads running in the policy server
- Max threads= maximum number of configured agent transaction processing threads (from smconsole)
- Busy Threads= snapshot of number of normal priority threads

CA Security SaaS Validation Program

Accelerating the secure connection to Cloud-based services

WHAT IS IT?

- A formal program to validate secure single sign-on to SaaS solutions with CA Single Sign-On

WHY?

- Faster, proven integration = enabling the business
- Backed by CA Support

RESULT

- Runbooks that map out your steps
- View current runbooks on the CA Support site: http://bit.ly/1mZyWwJ

Benefits

- Faster connection to cloud apps
- Improved user experience
- Scalable
- Requisite security

ca technologies

Some of the Partners to Date…

ca technologies

Stephan McQuiggan
Sr. Principal Support Engineer

Aaron Berman
VP, Sr. Advisor Single Sign-On and Directory

@CASecurity
slideshare.net/CAinc
linkedin.com/company/ca-technologies

communities.ca.com