Item 65

git_comments:

1. * Licensed to the Apache Software Foundation (ASF) under one or more * contributor license agreements. See the NOTICE file distributed with * this work for additional information regarding copyright ownership. * The ASF licenses this file to You under the Apache License, Version 2.0 * (the "License"); you may not use this file except in compliance with * the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License.
2. * * @return redaction string to be used instead of the value.
3. * * Returns if the given system property should be redacted. * * @param name The system property that is being checked. * @return true if property should be redacted.
4. * Licensed to the Apache Software Foundation (ASF) under one or more * contributor license agreements. See the NOTICE file distributed with * this work for additional information regarding copyright ownership. * The ASF licenses this file to You under the Apache License, Version 2.0 * (the "License"); you may not use this file except in compliance with * the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License.

git_commits:

1. **summary:** SOLR-10076: Hide keystore and truststore passwords from /admin/info/* outputs.
   message: SOLR-10076: Hide keystore and truststore passwords from /admin/info/* outputs.

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** Hiding keystore and truststore passwords from /admin/info/* outputs
   description: Passing keystore and truststore password is done by system properties, via cmd line parameter. As result, {{/admin/info/properties}} and {{/admin/info/system}} will print out the received password. Proposing solution to automatically redact value of any system property before output, containing the word {{password}}, and replacing its value with {{******}}.

jira_issues_comments:

1. Is there any objection about the approach? Does this count as API change (assuming that somebody depends on the exposed passwords)? If there is no objection, I will start working on the patch.
2. Been thinking about the same, but perhaps instead of a generic rule about containing password, we could have a property somewhere for what paths to hide. I would also like to hide the content of some ZK nodes such as security.json, and there may also be other places where passwords are exposed through props or APIs... Ideal would be if this could be coupled with Authorization, so that certain info could be controlled through group membership in AuthorizationPlugin?
3. bq. Been thinking about the same, but perhaps instead of a generic rule about containing password, we could have a property somewhere for what paths to hide. Thanks [~janhoy] for the feedback! I was also thinking of a pattern-based parameter masking for input password. I prepared a patch with a RedactionUtils that I will extend with external parameters and upload it shortly. bq. I would also like to hide the content of some ZK nodes such as security.json, and there may also be other places where

passwords are exposed through props or APIs... I was not aware of the security.json exposing password, I created a separate jira for that as well (SOLR-10100). bq. Ideal would be if this could be coupled with Authorization, so that certain info could be controlled through group membership in AuthorizationPlugin? In general, I would not add password visibility based on privileges. I think passwords should not be revertible, as that would expose them to the reliability of the authorization plugin and the admin users' cautiousness. For me it would somewhat beat the purpose of this jira: reducing the exposure of the security credentials. Do you see any business-case when you would grant certain roles to view these passwords?

4. Attaching patch.
5. bq. I was not aware of the security.json exposing password Passwords are not exposed. Salted hashes of the passwords are, though. bq. In general, I would not add password visibility based on privileges. I think passwords should not be revertible, as that would expose them to the reliability of the authorization plugin and the admin users' cautiousness. In the case of security.json, we should encourage and try to ensure that proper authorization is in place while starting a Solr cluster. To an authorized admin user, I don't see why we shouldn't show salted hashes of passwords. Anyway, we can deal with that issue on SOLR-7890/SOLR-10100.
6. Thank you for the feedback, [~ichattopadhyaya]. Do you think the redaction of command line password could be handled as the first patch contains?
7. **body:** This looks okay to me. We probably want to push users towards configuring this in a way it's not on the command line though, right? It's nice not to expose it via the web UI when we see it, but you also don't really want it on the command line as that stuff is pretty easy to introspect via people that should not. Our doc should probably encourage people to use system property on the command line alternatives or we should look at disabling / warning when it's done. I know our start scripts recently still set some of this ssl stuff via the command line, but if that is still the case, we should fix that too.
   label: code-design
8. Thank you [~markrmiller@gmail.com] for your comment. bq. We probably want to push users towards configuring this in a way it's not on the command line though, right? I agree that this is more like a workaround in the current state. It could also work as a second layer of protection if passwords being passed in command line. I would assume that getting the list of running processes on a server would require higher privileges than accessing the admin-ui, which suggests that the passwords should not be exposed there. Also, the {{/admin/info/properties}} API would expose password were set differently. bq. I know our start scripts recently still set some of this ssl stuff via the command line, but if that is still the case, we should fix that too. Is there a jira for that? I would be happy looking into it.
9. **body:** So I think we want to make sure the search for 'password' is case insensitive due to things like javax.net.ssl.trustStorePassword. Could use a test for that too. We should move RedactionUtils.java to org.apache.solr.util probably. Greg did something similar in Cloudera Search lucene-solr repo as a temporary hack, but used '--REDACTED--' I think that is more clear than the ****** redaction string. Given the affect this could have on tools/scripts that read output, I think it's not a huge deal if we changed it, but I don't see a strong reason to do it and that should usually favour back compat, even if we would guess those affected might be very few. We can do it by default in 7 and anyone looking for this in 6.5 and beyond will know they need it and it didn't exist in 6.4 and < and can turn it on. Seems like the least friction.
   label: code-design
10. [~markrmiller@gmail.com], thank you for the review and comments! - I added test for case-sensitive property name, it in fact was not properly working. - I changed the redaction text to the one that Greg added. Actually this patch is the generalization of his original intent. - I made the system property redaction configurable with default true. 6.x backport only need to vary by the default value of that configuration to have it turned off.
11. Commit 91c3f78f8fafbd95cd375bb114e80831ba50d525 in lucene-solr's branch refs/heads/master from markrmiller [https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=91c3f78] SOLR-10076: Hide keystore and truststore passwords from /admin/info/* outputs.
12. Thanks [~manokovacs]! I fixed the testDisabledRedaction tests (was still the same as when enabled), everything else looks good. Can you put up a backport with it defaulting to off?
13. Commit ddda27e4deab45b9a6bfec8d61319b00f88e27f6 in lucene-solr's branch refs/heads/master from [~cpoerschke] [https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=ddda27e] SOLR-10076: 'String.format(Locale.ROOT,...' instead of (forbidden API) 'String.format(...)'
14. Thanks [~markrmiller@gmail.com]! Sorry for the non-finished test, I will be more careful next time. I attached the 6x backport with default false configuration. It includes [~cpoerschke]'s patch about forbidden API. (Did not know about that, thank you!)

15. Commit 81e85993bd32f6475e762086bbd4f32dec76ca53 in lucene-solr's branch refs/heads/branch_6x from markrmiller [https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=81e8599] SOLR-10076: Hide keystore and truststore passwords from /admin/info/* outputs.
16. Thanks [~manokovacs]! We missed 6.5, so I have to move the changes entry on master and then I'll close.
17. Commit e11c86f6e4f85fc4ea561283cf6d2fa8c8df2208 in lucene-solr's branch refs/heads/master from markrmiller [https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=e11c86f] SOLR-10076: Move changes entry to 6.6 release.