Swauth Documentation

Release 1.0.1

OpenStack, LLC

December 08, 2016

Contents

An Auth Service for Swift as WSGI Middleware that uses Swift itself as a backing store. Sphinx-built docs at: http://gholt.github.com/swauth/ Source available at: https://github.com/gholt/swauth

See also https://github.com/khussein/keystone for the future standard OpenStack auth service.

This is currently a work in progress of pulling Swauth out of the Swift repo and here into its own project. See https://code.launchpad.net/~gholt/swift/deswauth/+merge/62392 for the Swift side of things.

CHAPTER 1

Quick Install

1. Install Swauth with `sudo python setup.py install` or `sudo python setup.py develop` or via whatever packaging system you may be using.
2. Alter your proxy-server.conf pipeline to have swauth instead of tempauth:

   Was:

```
[pipeline:main]
pipeline = catch_errors cache tempauth proxy-server
```

   Change To:

```
[pipeline:main]
pipeline = catch_errors cache swauth proxy-server
```

3. Add to your proxy-server.conf the section for the Swauth WSGI filter:

```
[filter:swauth]
use = egg:swauth#swauth
set log_name = swauth
super_admin_key = swauthkey
```

4. Restart your proxy server `swift-init proxy reload`.
5. Initialize the Swauth backing store in Swift `swauth-prep -K swauthkey`.
6. Add an account/user `swauth-add-user -A http://127.0.0.1:8080/auth/ -K swauthkey -a test tester testing`.
7. Ensure it works `st -A http://127.0.0.1:8080/auth/v1.0 -U test:tester -K testing stat -v`.

CHAPTER 2

Contents

2.1 LICENSE

```
 Copyright (c) 2010-2011 OpenStack, LLC

 Licensed 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.


                              Apache License
                        Version 2.0, January 2004
                     http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

   "License" shall mean the terms and conditions for use, reproduction,
   and distribution as defined by Sections 1 through 9 of this document.

   "Licensor" shall mean the copyright owner or entity authorized by
   the copyright owner that is granting the License.

   "Legal Entity" shall mean the union of the acting entity and all
   other entities that control, are controlled by, or are under common
   control with that entity. For the purposes of this definition,
   "control" means (i) the power, direct or indirect, to cause the
   direction or management of such entity, whether by contract or
   otherwise, or (ii) ownership of fifty percent (50%) or more of the
   outstanding shares, or (iii) beneficial ownership of such entity.

   "You" (or "Your") shall mean an individual or Legal Entity
```

```
   exercising permissions granted by this License.

   "Source" form shall mean the preferred form for making modifications,
   including but not limited to software source code, documentation
   source, and configuration files.

   "Object" form shall mean any form resulting from mechanical
   transformation or translation of a Source form, including but
   not limited to compiled object code, generated documentation,
   and conversions to other media types.

   "Work" shall mean the work of authorship, whether in Source or
   Object form, made available under the License, as indicated by a
   copyright notice that is included in or attached to the work
   (an example is provided in the Appendix below).

   "Derivative Works" shall mean any work, whether in Source or Object
   form, that is based on (or derived from) the Work and for which the
   editorial revisions, annotations, elaborations, or other modifications
   represent, as a whole, an original work of authorship. For the purposes
   of this License, Derivative Works shall not include works that remain
   separable from, or merely link (or bind by name) to the interfaces of,
   the Work and Derivative Works thereof.

   "Contribution" shall mean any work of authorship, including
   the original version of the Work and any modifications or additions
   to that Work or Derivative Works thereof, that is intentionally
   submitted to Licensor for inclusion in the Work by the copyright owner
   or by an individual or Legal Entity authorized to submit on behalf of
   the copyright owner. For the purposes of this definition, "submitted"
   means any form of electronic, verbal, or written communication sent
   to the Licensor or its representatives, including but not limited to
   communication on electronic mailing lists, source code control systems,
   and issue tracking systems that are managed by, or on behalf of, the
   Licensor for the purpose of discussing and improving the Work, but
   excluding communication that is conspicuously marked or otherwise
   designated in writing by the copyright owner as "Not a Contribution."

   "Contributor" shall mean Licensor and any individual or Legal Entity
   on behalf of whom a Contribution has been received by Licensor and
   subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
```

```
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
       that You distribute, all copyright, patent, trademark, and
       attribution notices from the Source form of the Work,
       excluding those notices that do not pertain to any part of
       the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its
       distribution, then any Derivative Works that You distribute must
       include a readable copy of the attribution notices contained
       within such NOTICE file, excluding those notices that do not
       pertain to any part of the Derivative Works, in at least one
       of the following places: within a NOTICE text file distributed
       as part of the Derivative Works; within the Source form or
       documentation, if provided along with the Derivative Works; or,
       within a display generated by the Derivative Works, if and
       wherever such third-party notices normally appear. The contents
       of the NOTICE file are for informational purposes only and
       do not modify the License. You may add Your own attribution
       notices within Derivative Works that You distribute, alongside
       or as an addendum to the NOTICE text from the Work, provided
       that such additional attribution notices cannot be construed
       as modifying the License.

   You may add Your own copyright statement to Your modifications and
   may provide additional or different license terms and conditions
   for use, reproduction, or distribution of Your modifications, or
   for any such Derivative Works as a whole, provided Your use,
   reproduction, and distribution of the Work otherwise complies with
   the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
   any Contribution intentionally submitted for inclusion in the Work
   by You to the Licensor shall be under the terms and conditions of
   this License, without any additional terms or conditions.
   Notwithstanding the above, nothing herein shall supersede or modify
   the terms of any separate license agreement you may have executed
   with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
   names, trademarks, service marks, or product names of the Licensor,
```

```
   except as required for reasonable and customary use in describing the
   origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
   agreed to in writing, Licensor provides the Work (and each
   Contributor provides its Contributions) on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
   implied, including, without limitation, any warranties or conditions
   of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
   PARTICULAR PURPOSE. You are solely responsible for determining the
   appropriateness of using or redistributing the Work and assume any
   risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
   whether in tort (including negligence), contract, or otherwise,
   unless required by applicable law (such as deliberate and grossly
   negligent acts) or agreed to in writing, shall any Contributor be
   liable to You for damages, including any direct, indirect, special,
   incidental, or consequential damages of any character arising as a
   result of this License or out of the use or inability to use the
   Work (including but not limited to damages for loss of goodwill,
   work stoppage, computer failure or malfunction, or any and all
   other commercial damages or losses), even if such Contributor
   has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
   incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

   To apply the Apache License to your work, attach the following
   boilerplate notice, with the fields enclosed by brackets "[]"
   replaced with your own identifying information. (Don't include
   the brackets!)  The text should be enclosed in the appropriate
   comment syntax for the file format. We also recommend that a
   file or class name and description of purpose be included on the
   same "printed page" as the copyright notice for easier
   identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed 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.
```

2.2 Implementation Details

The Swauth system is a scalable authentication and authorization system that uses Swift itself as its backing store. This section will describe how it stores its data.

At the topmost level, the auth system has its own Swift account it stores its own account information within. This Swift account is known as self.auth_account in the code and its name is in the format self.reseller_prefix + ".auth". In this text, we'll refer to this account as <auth_account>.

The containers whose names do not begin with a period represent the accounts within the auth service. For example, the <auth_account>/test container would represent the "test" account.

The objects within each container represent the users for that auth service account. For example, the <auth_account>/test/bob object would represent the user "bob" within the auth service account of "test". Each of these user objects contain a JSON dictionary of the format:

```
{"auth": "<auth_type>:<auth_value>", "groups": <groups_array>}
```

The *<auth_type>* can only be *plaintext* at this time, and the *<auth_value>* is the plain text password itself.

The *<groups_array>* contains at least two groups. The first is a unique group identifying that user and it's name is of the format *<user>:<account>*. The second group is the *<account>* itself. Additional groups of *.admin* for account administrators and *.reseller_admin* for reseller administrators may exist. Here's an example user JSON dictionary:

```
{"auth": "plaintext:testing",
 "groups": ["name": "test:tester", "name": "test", "name": ".admin"]}
```

To map an auth service account to a Swift storage account, the Service Account Id string is stored in the *X-Container-Meta-Account-Id* header for the <auth_account>/<account> container. To map back the other way, an <auth_account>/.account_id/<account_id> object is created with the contents of the corresponding auth service's account name.

Also, to support a future where the auth service will support multiple Swift clusters or even multiple services for the same auth service account, an <auth_account>/<account>/.services object is created with its contents having a JSON dictionary of the format:

```
{"storage": {"default": "local", "local": <url>}}
```

The "default" is always "local" right now, and "local" is always the single Swift cluster URL; but in the future there can be more than one cluster with various names instead of just "local", and the "default" key's value will contain the primary cluster to use for that account. Also, there may be more services in addition to the current "storage" service right now.

Here's an example .services dictionary at the moment:

```
{"storage":
    {"default": "local",
     "local": "http://127.0.0.1:8080/v1/AUTH_8980f74b1cda41e483cbe0a925f448a9"}}
```

But, here's an example of what the dictionary may look like in the future:

```
{"storage":
    {"default": "dfw",
     "dfw": "http://dfw.storage.com:8080/v1/AUTH_8980f74b1cda41e483cbe0a925f448a9",
     "ord": "http://ord.storage.com:8080/v1/AUTH_8980f74b1cda41e483cbe0a925f448a9",
     "sat": "http://ord.storage.com:8080/v1/AUTH_8980f74b1cda41e483cbe0a925f448a9"},
 "servers":
    {"default": "dfw",
     "dfw": "http://dfw.servers.com:8080/v1/AUTH_8980f74b1cda41e483cbe0a925f448a9",
     "ord": "http://ord.servers.com:8080/v1/AUTH_8980f74b1cda41e483cbe0a925f448a9",
     "sat": "http://ord.servers.com:8080/v1/AUTH_8980f74b1cda41e483cbe0a925f448a9"}}
```

Lastly, the tokens themselves are stored as objects in the *<auth_account>/.token_[0-f]* containers. The names of the objects are the token strings themselves, such as *AUTH_tked86bbd01864458aa2bd746879438d5a*. The exact *.token_[0-f]* container chosen is based on the final digit of the token name, such as *.token_a* for the token *AUTH_tked86bbd01864458aa2bd746879438d5a*. The contents of the token objects are JSON dictionaries of the format:

```
{"account": <account>,
 "user": <user>,
 "account_id": <account_id>,
 "groups": <groups_array>,
 "expires": <time.time() value>}
```

The *<account>* is the auth service account's name for that token. The *<user>* is the user within the account for that token. The *<account_id>* is the same as the *X-Container-Meta-Account-Id* for the auth service's account, as described above. The *<groups_array>* is the user's groups, as described above with the user object. The "expires" value indicates when the token is no longer valid, as compared to Python's time.time() value.

Here's an example token object's JSON dictionary:

```
{"account": "test",
 "user": "tester",
 "account_id": "AUTH_8980f74b1cda41e483cbe0a925f448a9",
 "groups": ["name": "test:tester", "name": "test", "name": ".admin"],
 "expires": 1291273147.1624689}
```

To easily map a user to an already issued token, the token name is stored in the user object's *X-Object-Meta-Auth-Token* header.

Here is an example full listing of an <auth_account>:

```
.account_id
    AUTH_2282f516-559f-4966-b239-b5c88829e927
    AUTH_f6f57a3c-33b5-4e85-95a5-a801e67505c8
    AUTH_fea96a36-c177-4ca4-8c7e-b8c715d9d37b
.token_0
.token_1
.token_2
.token_3
.token_4
.token_5
.token_6
    AUTH_tk9d2941b13d524b268367116ef956dee6
.token_7
.token_8
    AUTH_tk93627c6324c64f78be746f1e6a4e3f98
.token_9
.token_a
.token_b
```

```
.token_c
.token_d
.token_e
    AUTH_tk0d37d286af2c43ffad06e99112b3ec4e
.token_f
    AUTH_tk766bbde93771489982d8dc76979d11cf
reseller
    .services
    reseller
test
    .services
    tester
    tester3
test2
    .services
    tester2
```

2.3 swauth

2.4 swauth.middleware

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

S

S