SSH with Globus Auth

Summary

As the community moves away from GSI X.509 certificates, we need a replacement for GSI-OpenSSH that uses Globus Auth (see https://docs.globus.org/api/auth/) for authentication. The solution would support both interactive and programmatic use, for example, to run remote jobs.

This document describes a seamless drop-in solution for current SSH use cases that integrates Globus Auth without modification to the SSH client or server. Proposed enhancements to the Globus CLI to obtain tokens from Globus Auth for SSH servers provides an end to end solution for users. The system-installed SSH is leveraged verbatim for all other operations, only repurposing the password authentication to use the tokens.

Overview

The solution has the SSH server as a registered resource server with Globus Auth, and client sends a OAuth token from Globus Auth in place of the SSH user password. The server is configured with a PAM module that talks to Globus Auth, and authorizes the user's access using the presented token.

In order to simplify client-side token management for the end user, the Globus CLI provides functionality to obtain and manage the necessary tokens from Globus Auth, and inject the token into the SSH client password request thereby avoiding manual input of tokens.

The SSH server-side Globus OAuth PAM module receives the OAuth token via standard password authentication mechanisms and calls out to Globus Auth to perform introspection on the token (See https://docs.globus.org/api/auth/reference/#token_introspection_post_v2_oauth2_token_introspect). The validated token can be used for authorization and mapping.

OAuth token authentication can be used in addition to, or in lieu of, system password authentication. The SSH service administrator can configure the PAM module to perform site-specific account mapping, require specific identity providers, and apply limitations on token use for the SSH service including time-since-retrieval of the initial token.

Globus CLI

The Globus CLI makes use of the system-installed SSH client which is not modified for this new authentication flow. The following Globus CLI option wraps the SSH client in order to perform the authentication flow but then passes the options verbatim to the local SSH client:

```
globus ssh <standard ssh client options including fqdn>
```

```
    Checks for valid credentials for the target fqdn, initiates the
    'globus ssh login <fqdn>` sequence if valid credentials are not
    found, then executes the local client SSH with the supplied
    client options.
```

The following command syntax is provided by the Globus CLI to aid in non interactive sessions, namely, to allow separation of login/logout authentication operations from session creation (above) and provide a way to check the validity of existing credentials.

```
globus login [--check] [--copy] [--ssh <fqdn>]

    Forces Globus CLI to check for valid credentials for the SSH
    service on <fqdn>. If the credentials are invalid or do not
    exist, initiates the sequence to authenticate and retrieve a
    token from Globus Auth.

    --check: Check for valid credentials for <fqdn> if given,
    otherwise check for valid transfer credentials. Does not
    initiate the login flow; useful for scripting.

    --copy: copy to clipboard if possible, else print to stdout

globus logout [--ssh <fqdn>]
    Forces Globus CLI to invalidate local SSH credentials for
    <fqdn>.
```

DNS round robin'ed servers, or any configuration of multiple A records in a CNAME, will share login credentials allowing users to log into each SSH service with a single authentication.

Password Injection

After the user has performed the authentication flow with Globus Auth and has received an access token, the token must be used as the password for the SSH connection.

The token can be passed to the SSH client in any of the following ways:

1) The user calls '`globus ssh <options>`' or globus-ssh which launches the SSH client, monitors the process for the password prompt and enters the bearer token. This option is transparent to the user.
2) The user calls '`globus login --copy --ssh <fqdn>`' which will place the token in the user's clipboard if possible, otherwise the token is printed to stdout. Then it is up to the user to paste this token into the SSH password prompt. This options supports graphical and non OpenSSH clients.

In either method, the user can specify the local username using standard SSH client options.

Supported Client Platforms

Since Globus SSH makes use of standard SSH password authentication, it should be compatible with most SSH clients. Transparent password inject via ‘`globus ssh <options>`’ or globus-ssh is only supported with OpenSSH clients on Unix, Linux and BSD distributions. ‘`globus login --copy --ssh <fqdn>`’ allows for simple cut-n-paste for graphical and other non OpenSSH clients.

Token Validity/Lifetime

Once authenticated, the user will be permitted to log into the SSH server at <fqdn> without being prompted for authentication until the access tokens issued for the SSH server (specific FQDN) expire or are invalidated.

The access tokens are issued with a lifetime of 48 hours. Refresh tokens can be used to obtain new access tokens, and are valid for 6 months from last use.

Using the Globus CLI, the following will result in prompts for the user to reauthenticate:

- the access tokens issued by Globus Auth expires. Since CLI uses refresh tokens to obtain user tokens, this will be 6 months after last use.
- the user performs ‘`globus logout`’ to invalidate access tokens to all SSH servers or a particular server
- The user rescinds consent given to the CLI for a particular SSH server (<fqdn>)

SSH server as Globus Auth resource server

The SSH service administrator registers the SSH service with Globus Auth (currently https://developers.globus.org) and is given a Globus Auth client ID and secret for use in the configuration of the SSH service. The admin installs the Globus SSH PAM module, configures it with the Globus Auth client ID and secret, and configures the service to allow or disallow system passwords per site security policy. The administrator will be able to specify the identity provider required for use with this SSH service.

Longer term, the administrator will be able to limit the capabilities of users who use this form of authentication with specific ‘levels of assurance’ (LOA) guarantees. These include, for example, requirements that the user must have performed ‘`globus login`’ within a recent given timeframe or that the credentials were retrieved via two factor authentication.

Supported Server Platforms
Currently, only supported with OpenSSH on Unix, Linux and BSD platforms.

Note: CLI command syntax here is a proposal and is still under review and consideration.