Flexible Enforcement of Multi-factor Authentication with SSH via Linux-PAM for Federated Identity Users

Derek Simmel
Pittsburgh Supercomputing Center
dsimmel@psc.edu

Shane Filus
Pittsburgh Supercomputing Center
filus@psc.edu

ABSTRACT

A computational science project with restricted-access data was awarded an allocation by XSEDE in 2016 to use the Bridges supercomputer at the Pittsburgh Supercomputing Center (PSC). As a condition of the license agreement for access to the data, multi-factor authentication (MFA) with XSEDE's Duo MFA service is required for users of this project to login to Bridges via SSH, in addition to filesystem access controls. Since not all Bridges users are required to authenticate to Bridges in this manner, a solution was implemented via Linux-PAM to require XSEDE Duo MFA for SSH login access by specific users, as identified by their local account name or membership in a local group. This paper describes the implementation on Bridges and its extensibility to other systems and environments with similar needs.

CCS CONCEPTS

• **Security and privacy → Multi-factor authentication**

KEYWORDS

authentication, Duo, federated identity, GSI-OpenSSH, Kerberos, LDAP, Linux, multi-factor authentication, MFA, MyProxy, OpenSSH, PAM, SSH, X.509, XSEDE

ACM Reference format:

Derek Simmel and Shane Filus. 2017. Flexible Enforcement of Multi-factor Authentication with SSH via Linux-PAM for Federated Identity Users. In *Proceedings of Practice & Experience in Advanced Research Computing 2017 (PEARC17), New Orleans, Louisiana, U.S.A., July 9-13, 2017*, 9 pages.
DOI: http://dx.doi.org/10.1145/3093338.3093392.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

PEARC17, July 09-13, 2017, New Orleans, LA, USA
© 2017 Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-5272-7/17/07...$15.00
http://dx.doi.org/10.1145/3093338.3093392

1 Introduction

The *Bridges*[1] supercomputer at the Pittsburgh Supercomputing Center (PSC) is one of several high performance computing (HPC) systems funded by the U.S. National Science Foundation (NSF), with allocations for use granted to eligible computational science projects on a peer-reviewed basis via the XSEDE[2] Resource Allocation System (XRAS). XSEDE is an NSF-funded cyberinfrastructure program for coordination and development of interoperable infrastructure and scientific user support. PSC is one of several XSEDE service provider (SP) institutions operating HPC resources for computational science users in the XSEDE community.

The principal investigators (PIs) for projects who are awarded allocations on XSEDE resources identify and take responsibility for users who may login and access XSEDE resources on behalf of their project and spend allocation units for resource usage against the project's balance. Resource usage may include computing service hours for jobs executed on various types of computing hardware, including multi-core CPUs and GPUs, storage volume occupied for the duration of the project, and charges for hosting virtual machines and databases on behalf of the project.

As an XSEDE-allocated HPC resource, the Bridges system accepts authentication for users with their XSEDE username and password, a federated identity separate from PSC's local site user account and PSC credentials. The federated XSEDE username and password facilitates login to the XSEDE User Portal[13] website, as well as single sign-on access to multiple XSEDE resources, relieving XSEDE users from having to recall and use separate usernames, passwords or other unique credentials for each XSEDE resource. Upon successful authentication to an XSEDE SP resource system using XSEDE credentials, the login procedure maps users from their XSEDE identity to the local user account on that SP resource.

For stronger authentication needs, XSEDE subscribes to the Duo[3] MFA service. Duo provides flexible means for supplemental authentication, including smart-phone apps, hardware tokens, and other out-of-band authentication methods. Duo provides open source software to integrate the service with a variety of web applications and Internet services, including Secure Shell (SSH) login and file transfer services on Linux systems[4].

During 2016, a computational science project with restricted-access data was awarded an allocation by XSEDE to use the Bridges supercomputer. The licensing agreement for a large data collection that the project uses for its computational applications requires strong, multi-factor authentication (MFA) of users authorized to access the data, in addition to filesystem access controls, to prevent unauthorized access to the data by other users on Bridges.

Utilizing the XSEDE Duo service subscription, PSC implemented MFA with SSH for these users, while keeping other authentication means for unrelated users intact and unaffected. This paper describes the requirements, design and implementation of flexible controls to enforce the MFA requirement for user authentication to SSH services for authorized users on Bridges login hosts. Modifications made to the Duo MFA client software are described, as well as details of SSH service authentication behavior and the Linux Pluggable Authentication Modules (PAM)[5] configuration files used to carry out the authentication enforcement sequence. The implementation described is adaptable to other similar flexible authentication needs on other systems utilizing PAM for authentication management.

2 SSH, GSI-OpenSSH & XSEDE Single Sign-On

Although the flexible MFA implementation could be extended to all services that use PAM for authentication, the scope of this implementation is limited to the two SSH services operated on the PSC Bridges login nodes. The first SSH service operates on the default SSH service port (22/TCP) as a standard OpenSSH[6] service with native Kerberos authentication enabled. This SSH service permits users to authenticate using their PSC Kerberos credentials, or using SSH public key authentication. The second SSH service operates on an alternate service port (2222/TCP) as a GSI-OpenSSH[7] service.

The GSI-OpenSSH service permits users to authenticate using X.509 credentials issued to them by XSEDE-approved certificate authorities (CAs), which are a subset of the IGTF[8] distribution of accredited CAs. Included among the XSEDE-approved CAs are the XSEDE MyProxy CA and the PSC MyProxy CA, both of which issue short-lived X.509 credentials to authorized XSEDE users, authenticated using their XSEDE username and password. Users who authenticate to GSI-OpenSSH using an X.509 credential need to be mapped to their corresponding local username on the system. To facilitate this, part of the GSI-OpenSSH configuration requires creation and maintenance of a *grid-mapfile*, a text file containing one-to-one mappings of X.509 certificate subject (a.k.a. distinguished name, or DN) strings to a local username; when a user authenticates successfully with their X.509 certificate, and a valid mapping of the certificate's subject string is found in the local grid-mapfile, the login process proceeds with the local username corresponding to that certificate subject in the grid-mapfile. These mappings are many-to-one; users may have several different X.509 certificate subjects mapped to their local username, but each unique certificate subject can only be mapped to one local username. In other words, users may have several certificates to authenticate with, but each certificate subject must be uniquely mapped to a single user account on the system. Figure 1 shows examples of some certificate subject mappings from a grid-mapfile:

```
$ grep janeuser$ /etc/grid-security/grid-mapfile

"/C=US/O=National Center for Supercomputing
Applications/CN=Jane User 1" janeuser
"/C=US/O=Pittsburgh Supercomputing Center/CN=Jane User 1"
janeuser
"/DC=org/DC=cilogon/C=US/O=Carnegie Mellon
University/CN=Jane User A9876" janeuser
```

Figure 1: grid-mapfile entries for the *janeuser* account (long lines appear wrapped)

Upon successful login using an X.509 credential, the GSI-OpenSSH service generates a new, short term X.509 proxy credential on the system on behalf of the user, so that they may continue to authenticate to other GSI-authenticated services without further interaction (for as long as that proxy is valid).

Users may also authenticate interactively to the GSI-OpenSSH service on Bridges using their federated XSEDE username and password. To permit mapping of federated identities to local usernames during authentication processing, GSI-OpenSSH adds a new configuration parameter to the OpenSSH service, named *PermitPAMUserChange*, to allow a PAM module to change the current USERNAME value during PAM processing.

The XSEDE Single Sign-on (SSO) configuration of GSI-OpenSSH uses a PAM module, *pam_remapuser.so*, provided by the MyProxy-Enhanced GSI-OpenSSH (MEG)[9] package, to add federated authentication and single sign-on functionality. During PAM authentication processing, pam_remapuser.so is used to call a script, in which a MyProxy[12] client is executed to connect via a TLS-secured session to one of the XSEDE MyProxy CA services, authenticating with the current username and password. If authentication succeeds, an X.509 credential is retrieved, and is used to look up the user in the GSI-OpenSSH grid-mapfile. If a valid mapping is found, then the current PAM USERNAME value is switched to the local username that corresponds with the retrieved X.509 certificate subject in the grid-mapfile, and further PAM processing continues.

As part of the XSEDE user account provisioning process for users on XSEDE-allocated projects, the mappings of certificate subjects for credentials issued by the XSEDE MyProxy CA and PSC MyProxy CA are automatically added to XSEDE SP system grid-mapfiles. Users may also add certificate subject mappings for authentication credentials issued to them from other CAs that are accepted by XSEDE.

3 XSEDE MFA for SSH via pam_duo

In this section, the specific needs for employing Duo for MFA are discussed, including some modifications that were applied to make it function as required for our mandatory MFA enforcement needs. For implementations using a different MFA service and technology, the functionality requirements defined in Section 4 should be sufficient to guide similar implementation.

Duo provides the source code for a PAM module and related Linux command-line utilities for download from its website and GitHub repository[10]. The code is open-source, and must be built, installed and configured on a host for use with a valid Duo service account. To complete the configuration, the administrator for the Duo service account must generate a set of unique keys for that host, which the host administrator copies into configuration files on that host. The Duo configuration files must be restricted for access by only the *root* user via filesystem access controls. The host administrator must also adjust SSH service configuration parameters to enable PAM processing, disable SSH native password processing, enable keyboard-interactive authentication, enable challenge-response authentication, and disable DNS lookups for IP addresses of user client hosts [4].

Every user authorized to use a Duo service account for MFA must first enroll in that Duo service account and configure an available Duo app (on smart phones or tablets), hardware token, or other available authentication method with their Duo user account. XSEDE users enroll in the XSEDE Duo service using a Duo enrollment user interface provided on the XSEDE User Portal.

In the simplest implementation, one adds Duo MFA to an SSH service PAM configuration file by adding a line that calls the *pam_duo.so* module. This initiates a secured connection to the Duo service, passing along the current username for MFA authentication. If the user has enrolled in the Duo service configured for that host, then the Duo service will initiate MFA to the user's configured MFA app, device or other method. If the user completes that MFA sequence correctly, the Duo session ends and the pam_duo.so module exits with a PAM_SUCCESS return value. If an enrolled user fails to authenticate correctly, or the MFA prompt times out, then the Duo session ends and the pam_duo.so module exits with a PAM_SERVICE_ERR return value. PAM processing proceeds accordingly.

3.1 XSEDE customization of pam_duo.so

The pam_duo.so module source code includes a query to check whether or not the username supplied is a valid username on the local system on which the PAM processing is taking place. This is reasonable when the local username matches the Duo user account name. For authentication with federated identities, however, such as with XSEDE usernames and passwords, and specifically in the case where the federated identity username is used to identify the user to the Duo MFA service, the username supplied to the pam_duo.so module should not be checked against local usernames on the local system, since the federated identity and local username may differ. A patched edition of to the pam_duo.c source code provided by XSEDE[11] adds a configuration parameter to the pam_duo.so module, *nosshpwcheck*, to disable the local username check in the pam_duo.so module accordingly.

3.2 PSC customization of pam_duo.so

In preparation for flexible MFA implementation, PSC added two additional patches to the Duo source code. The first PSC patch addresses default behavior of the pam_duo.so module when a service error occurs during an attempt by the pam_duo.so module to contact the Duo service. As defined in the original source code, the pam_duo.so module will return a PAM_SUCCESS value if the Duo PAM configuration is defined with the failure mode set to DUO_FAIL_SAFE. This is the default value in the source code if the host administrator leaves the failure mode undefined in the Duo PAM configuration file; as a result, when an unexpected error occurs during communication between the pam_duo.so module and the Duo service, it returns with a PAM_SUCCESS value. In other words, a user will be permitted to authenticate successfully by default if the Duo service is unable to execute the MFA challenge correctly.

While this may be appropriate to prevent unintended lockout from a service in debugging, we at PSC prefer for the default behavior to deny user access when Duo service errors arise, to avoid the possibility of a deliberate attempt to bypass Duo MFA using a denial of service attack. This PSC patch also adds a new function, *duo_local_hostname()*, to return the locally-defined hostname for use when Duo MFA is employed by authorized administrators for authentication with the Linux *su* or *sudo* commands.

The second PSC patch addresses default behavior of the pam_duo.so module when a user has not yet enrolled in the Duo service. In the original source code, if a user is unknown to the Duo service when the pam_duo.so module calls, a PAM_SUCCESS value is returned. In this interpretation, an unknown user is assumed to be someone who is *not* required to authenticate with Duo MFA for that Duo service account, and therefore for whom there is no need to deny access via DUO authentication in PAM. For our mandatory MFA requirements, however, we need to identify users who have not yet enrolled in Duo, but who are required by our local policy to have done so. These users must therefore be denied access until they have enrolled in the XSEDE Duo account and can authenticate themselves successfully to the XSEDE Duo service using their federated XSEDE username and password.

To identify not-yet-enrolled users when the pam_duo.so module calls the Duo service, two modifications were made: first, an additional parameter for the pam_duo.so module was added,

enrolledreqd, to define a flag which, if enabled, causes the pam_duo.so module to return the value of PAM_AUTHINFO_UNAVAIL instead of PAM_SUCCESS when a username is unknown to the Duo service. Although there is a PAM_USER_UNKNOWN value that could also have been used, other unrelated parts of the pam_duo.so code use that value in a different context, and we thought it better to avoid misinterpretation by using an otherwise unused return value to signal the lack of enrollment by a user in the Duo service.

The second PSC patch also adds definition of a temporary environment variable, PAM_DUO_AUTHN, to which the PAM return value is assigned before the pam_duo.so module exits. This environment variable provides a mechanism to communicate the result of the pam_duo.so module execution between PAM processing stacks, which is helpful in the case of PAM processing for SSH, as we will clarify later in the Implementation section below.

4 Requirements for Flexible MFA

The implementation of controls to enforce MFA for user authentication to SSH services must take into account the following requirements:

1. **Limited MFA Requirement**: Only users in designated groups authorized to access the restricted-access data are required to authenticate to SSH services with MFA; all other unrelated users may authenticate as they have done before.

2. **Varying MFA Requirement**: The membership of users in groups required to authenticate with MFA will vary over time. Individuals may be authorized to access restricted data for limited periods. The implementation must therefore be flexible to permit various role- and project-defined groups and individual users to be added or removed from the MFA authentication requirement.

3. **MFA Service Enrollment**: The XSEDE Duo authentication service is operated on an opt-in basis. This means that XSEDE users who are required to use the service must first enroll into the XSEDE Duo service and configure their multi-factor authentication app or device correctly before they can use the XSEDE Duo service for authentication. However, not all XSEDE users are required to use XSEDE Duo. As a result, services configured to authenticate with Duo may permit un-enrolled users to pass unchallenged, i.e., that the Duo service cannot assume that its use is mandatory, and is configured to skip authentication if a user is unknown. Enforcement of enrollment by a user required to employ MFA for authentication therefore falls to the service (in our case, SSH) on the target system calling the Duo service for MFA, rather than the Duo service itself. To do so, the target service operator must be able to identify whether or not a user is currently enrolled in the Duo MFA service.

4. **MFA Integration Interface to SSH**: Duo authentication for SSH services is implemented via PAM. SSH services permit a variety of authentication methods, including built-in SSH public key authentication, Kerberos and other GSSAPI-mediated methods, in addition to external methods for authentication and authorization via PAM. This means that the SSH configuration must include and enable PAM, and that the implementation of enforcement of MFA must prevent the other built-in SSH authentication methods from bypassing it.

5. **Mapping of Federated Identity to Site-Local Identity**: XSEDE users employ their federated XSEDE username and password to authenticate to XSEDE services, including the XSEDE Duo MFA service. Following initial XSEDE username/password authentication, XSEDE usernames must be mapped to site-local user account names before SSH sessions can be initiated on site-local systems.

6. **Consistency of Enforcement**: Other services operated on the same system must not permit users to bypass the MFA authentication requirement. For example, if additional SSH services are operating on alternate ports to facilitate different user access methods, these services must be configured to identify and deny access to users in who are required to use MFA.

5 Design

The requirements described above impose a sequence of authentication and authorization decisions for both the PSC-local OpenSSH service and the XSEDE SSO GSI-OpenSSH service on Bridges login nodes. The XSEDE SSO GSI-OpenSSH service already handles authentication for XSEDE users using their federated XSEDE username and password, and the XSEDE Duo MFA service is expecting the same XSEDE username and password for authentication via PAM. XSEDE Duo MFA enforcement is therefore integrated with the XSEDE SSO GSI-OpenSSH service. Even so, the PSC-local OpenSSH service must also participate in enforcement, to the extent that it must prevent users who are required to authenticate using XSEDE Duo MFA from logging in without it. When such users attempt to login incorrectly, the user interface should advise them to enroll in the MFA service and direct them to the correct SSH service to login.

The design of operational authentication workflow for the flexible MFA implementation on the PSC Bridges system is depicted in Figure 2 below.

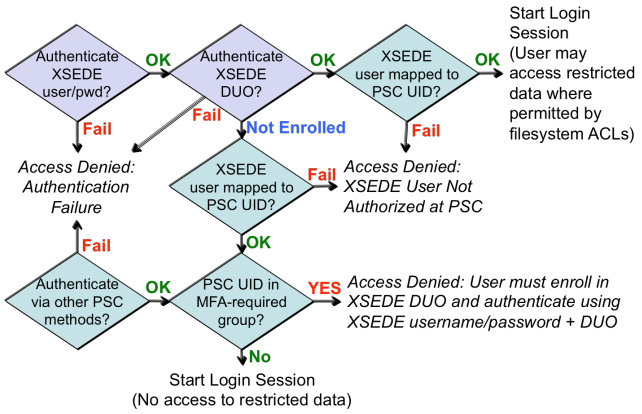

Figure 2: MFA authentication workflow

The top row of decision points in Figure 2 reflects the normal authentication sequence for an XSEDE user via the XSEDE SSO GSI-OpenSSH service with XSEDE Duo MFA. This is the required sequence to be followed by users in a group with access to restricted data. Note that other users not required to use XSEDE Duo MFA but who are enrolled in the XSEDE Duo MFA service account may also login via the top row sequence, but will not be able to access the restricted data because the filesystem access controls will prevent them from doing so.

The bottom row of decision points reflects authentication for users logging in and authenticating via the PSC-local OpenSSH service. If a user required to use XSEDE Duo MFA attempts to follow this sequence, they will be denied access and advised to login via the XSEDE SSO GSI-OpenSSH service plus XSEDE Duo MFA. Other users not required to use XSEDE Duo MFA are allowed to login to Bridges as they did before, with no access to the restricted data.

The central column of decision points represents the path taken by users not yet enrolled in the XSEDE Duo service account.

6 Implementation

The MFA authentication workflow is implemented on the PSC Bridges system in the PAM configuration files for the *sshd* (PSC local OpenSSH) and *gsisshd* (XSEDE SSO GSI-OpenSSH plus XSEDE Duo MFA) services depicted below in Figure 3 and Figure 5, respectively:

Note in Figure 3 the PAM rule added to the *account* stack for sshd, to include additional rules defined in a file called *xsede-tfa-check.acct*. The PAM *account* stack is normally associated with *authorization* decisions to be made after the sequence of authentication steps represented in the PAM *auth* stack. However, since this SSH service permits SSH-native Kerberos authentication and SSH public key authentication, users who employ one of these authentication means will bypass the PAM auth stack. This means that to catch users who are required to use XSEDE Duo MFA to authenticate via SSH to the system, we have to do so in the PAM account authorization stack, which is processed after initial SSH authentication of the user has completed.

```
[root@br006:~]# cat /etc/pam.d/sshd

#%PAM-1.0
auth    required   pam_sepermit.so
auth    substack   password-auth
auth    include    postlogin
...
account required   pam_nologin.so
account include  /usr/xsede/etc/pam.d/xsede-tfa-check.acct
account include    password-auth
...
```

Figure 3: /etc/pam.d/sshd *auth* and *account* stacks

Figure 4 shows the PAM account authorization steps executed to determine whether or not a user should have logged in using the XSEDE SSO GSI-OpenSSH and XSEDE Duo MFA service. The decision is based on whether or not the user's local account is in a local group listed in a file named *duo-required-groups*, or the user's local account is listed in a file named *duo-required-users*. If either is true, a message is displayed, indicating that they were denied access and that XSEDE Duo authentication is required for them to access the system:

```
[root@br006:~]# cat /usr/xsede/etc/pam.d/xsede-tfa-
check.acct

account [success=ok auth_err=2 default=bad]
 pam_listfile.so onerr=succeed item=group sense=deny
 file=/usr/xsede/etc/duo-required-groups
account [success=4 auth_err=bad default=bad]
 pam_listfile.so onerr=succeed item=user sense=deny
 file=/usr/xsede/etc/duo-required-users
account [default=1] pam_echo.so
 file=/usr/xsede/etc/denied-user-msg.txt
account optional  pam_echo.so
 file=/usr/xsede/etc/denied-group-msg.txt
account optional  pam_echo.so
 file=/usr/xsede/etc/xsede-duo-reqd-msg.txt
account requisite pam_deny.so

[root@br006:~]# cat /usr/xsede/etc/denied-user-msg.txt

Access denied: user

[root@br006:~]# cat /usr/xsede/etc/denied-group-msg.txt

Access denied: group

[root@br006:~]# cat /usr/xsede/etc/xsede-duo-reqd-msg.txt

XSEDE DUO authentication is required for login access to
this system. SSH publickey and gsissh authentication are
not permitted when XSEDE DUO is required. For additional
help, send e-mail to help@xsede.org.
```

Figure 4: /usr/xsede/etc/pam.d/xsede-tfa-check.acct (long lines appear wrapped)

The PAM rules for the *gsisshd* service, representing XSEDE SSO GSI-OpenSSH with XSEDE Duo MFA, are shown in Figure 5:

```
[root@br006:~]# cat /etc/pam.d/gsisshd

#%PAM-1.0
auth     required pam_sepermit.so
auth     include  /usr/xsede/etc/pam.d/xsede-tfa.auth
auth     include  password-auth
account  required pam_nologin.so
account  include  /usr/xsede/etc/pam.d/xsede-tfa.acct
account  include  password-auth
password include  password-auth
# pam_selinux.so close should be the first session rule
session  required pam_selinux.so close
session  required pam_loginuid.so
# pam_selinux.so open should only be followed by sessions
to be executed in the user context
session  required pam_selinux.so open env_params
session  optional pam_keyinit.so force revoke
session  include  password-auth
session  include  /usr/xsede/etc/pam.d/xsede-tfa.sess
```

Figure 5: /etc/pam.d/gsisshd (long lines appear wrapped)

```
[root@br006:~]# cat /usr/xsede/etc/pam.d/xsede-tfa.auth

auth optional pam_echo.so
 file=/usr/xsede/etc/xsede-tfa-msg.txt
auth [success=ok new_authtok_reqd=ok default=die]
 /usr/xsede/lib64/security/pam_remapuser.so
 /usr/xsede/sbin/auth_myproxy_user.sh
auth [success=ok new_authtok_reqd=ok authinfo_unavail=1
 service_err=die default=die]
 /usr/xsede/lib64/security/pam_duo.so
 conf=/usr/xsede/etc/duo/pam_duo-xsede.conf nosshpwcheck
 enrolledreqd debug
auth sufficient /usr/xsede/lib64/security/pam_remapuser.so
 /usr/xsede/sbin/auth_map_xsede_user.sh
# Do the following if the user was not enrolled in DUO
(and deny access if they are required to be enrolled).
auth requisite /usr/xsede/lib64/security/pam_remapuser.so
 /usr/xsede/sbin/auth_map_xsede_user.sh
auth [success=ok auth_err=2 default=bad] pam_listfile.so
 onerr=succeed item=group sense=deny
 file=/usr/xsede/etc/duo-required-groups
auth [success=done auth_err=bad default=bad]
 pam_listfile.so onerr=succeed item=user sense=deny
 file=/usr/xsede/etc/duo-required-users
auth [default=1] pam_echo.so
 file=/usr/xsede/etc/denied-user-msg.txt
auth optional pam_echo.so
 file=/usr/xsede/etc/denied-group-msg.txt
auth optional pam_echo.so
 file=/usr/xsede/etc/xsede-duo-enroll-msg.txt
auth requisite pam_deny.so
```

Figure 6: *xsede-tfa.auth* PAM auth stack (long lines appear wrapped)

For the gsisshd PAM configuration shown in Figure 5, the *auth* stack includes a set of rules, in a file called *xsede-tfa.auth* (see Figure 6), to authenticate the user using their XSEDE username and password, and to execute Duo MFA. Note that as with the *sshd* PAM configuration, a set of rules is also included in the PAM *account* authorization stack. We need to do this because the GSI-OpenSSH service can accept SSH-native GSI-authentication using X.509 credentials, which would also bypass the PAM *auth* stack.

Figure 6 shows the details of the PAM auth stack steps executed to authenticate XSEDE users to GSI-OpenSSH using their federated XSEDE username and password, plus Duo MFA. Note that mapping of a user from their XSEDE username to their PSC local username must take place *after* Duo MFA is attempted, since the XSEDE username is required for Duo MFA user lookup. Users who are not enrolled in XSEDE Duo but who are required to be (as represented by one of their local groups being listed in the *duo-required-groups* file or their local account being listed in the *duo-required-users* file) will be denied access and advised in a message that XSEDE Duo authentication is required and where they should go on the XSEDE User Portal to enroll in Duo.

The auth stack line for mapping an XSEDE username to its PSC username appears twice in this stack. This is necessary because of the two different paths that the PAM processing can take: The return value of the pam_duo.so rule determines which of the subsequent auth rules are executed - if Duo MFA succeeds, then the first mapping line will exit the stack immediately, after attempting to map the XSEDE federated username to a local account name - and if that mapping fails, access is denied immediately too.

If the return value from the pam_duo.so PAM auth rule was PAM_AUTHINFO_UNAVAIL, then the user was not enrolled in XSEDE Duo and we need to determine whether the user is among those required to be enrolled and to use XSEDE Duo MFA authentication. The *authinfo_unavail=1* entry in the bracketed part of the pam_duo.so auth rule tells PAM to skip over the next auth rule in the stack if the return value was PAM_AUTHINFO_UNAVAIL, which lands us on the second mapping line. The mapping still needs to be done so that the *duo-required-groups* and *duo-required-users* files can be checked to see if that local user is required to use XSEDE Duo MFA. If that user is *not* required to use XSEDE Duo MFA, then they are permitted to login without it. Otherwise, appropriate user messages are presented before denying access.

Referring back to Figure 5, the PAM account authorization stack for gsissh includes a file, named *xsede-tfa.acct*, with additional PAM account authorization steps to check whether or not an authenticated user should have authenticated using XSEDE Duo MFA. The contents of this file and a utility script that it calls are depicted in Figure 7. This is where PAM processing checks that the user attempting to login via GSI-OpenSSH has a valid local account or one in the PSC's LDAP directory. Checking the value of the PAM_DUO_AUTHN environment variable set by the patched pam_duo.so module, it determines whether or not the user has already authenticated successfully using XSEDE Duo MFA in the PAM *auth* stack. If PAM_DUO_AUTHN is defined and has a value of zero, then XSEDE Duo MFA completed

successfully and PAM processing can resume after all the remaining account authorization lines in this file.

If the PAM_DUO_AUTHN environment variable is not defined, then pam_duo.so was not previously executed. If the PAM_DUO_AUTHN environment variable is defined, but the value is non-zero, then an error occurred during XSEDE Duo MFA. For both of these cases, the rules in *xsede-tfa-check.acct* (Figure 4) are executed to determine whether the user is required to authenticate using XSEDE Duo MFA.

At the bottom of the gsissh PAM configuration file shown in Figure 5 is a PAM *session* stack line used to include a PAM session rule for removing the PAM_DUO_AUTHN temporary environment variable from the user's shell environment. This PAM session stack rule and its associated configuration file are listed at the bottom of Figure 7 below:

```
[root@br006:~]# cat /usr/xsede/etc/pam.d/xsede-tfa.acct

account [success=1 default=ignore] pam_localuser.so
account [success=ok default=die] pam_ldap.so
account [success=6 default=ignore] pam_exec.so quiet
 /usr/xsede/sbin/pam_duo_authn_test.sh
account include /usr/xsede/etc/pam.d/xsede-tfa-check.acct


[root@br006:~]# cat /usr/xsede/sbin/pam_duo_authn_test.sh

#!/bin/bash
if [ -n "${PAM_DUO_AUTHN}" ]; then
         if [ "${PAM_DUO_AUTHN}" = "0" ]; then
                  exit 0
         fi
fi
exit 1


[root@br006:~]# cat /usr/xsede/etc/pam.d/xsede-tfa.sess

session  optional pam_env.so
 conffile=/usr/xsede/etc/pam_duo_authn_unset.env
 readenv=0 user_readenv=0


[root@br006:~]# cat /usr/xsede/etc/pam_duo_authn_unset.env

PAM_DUO_AUTHN
```

Figure 7: xsede-tfa.acct, pam_duo_authn_test.sh, xsede-tfa.sess, and pam_duo_authn_unset.env (long lines appear wrapped)

The contents of the *duo-required-groups* file and *duo-required-users* files are simply text lists with a single left-justified entry per line. The *duo-required-groups* file lists local or LDAP groups of users who are required to use XSEDE Duo MFA to authenticate to Bridges. Similarly, the *duo-required-users* file lists local or LDAP users who are required to use XSEDE Duo MFA to authenticate to Bridges.

7. Results

The flexible MFA enforcement implementation for SSH services via PAM described above has been working well in production on Bridges at PSC since October 2016. The error messages displayed for users who have not enrolled in XSEDE Duo MFA but who are required to use it for authorized access to Bridges have generally been sufficient to direct these users to take the steps that they need to enroll in XSEDE Duo and to authenticate correctly without additional support from XSEDE or PSC user consultants.

Figure 8 below shows an interaction for a user who attempts first to login via the PSC-local OpenSSH service (PSC Kerberos or SSH public key authentication) on port 22/TCP. He then tries to login via the XSEDE SSO GSI-OpenSSH service on port 2222/TCP:

```
$ ssh joeuser@bridges.psc.xsede.org

Access denied: user
XSEDE DUO authentication is required for login access to
this system. SSH publickey and gsissh authentication are
not permitted when XSEDE DUO is required.
For additional help, send e-mail to help@xsede.org.
Connection closed by 128.182.108.56


$ ssh -p 2222 -l joeXsede bridges.psc.xsede.org

XSEDE Authentication
password:

Access denied: group
XSEDE DUO authentication is required for login access to
this system. Please log in to the XSEDE User Portal and
enroll in DUO under My XSEDE->Profile:
https://portal.xsede.org/group/xup/profile#/
For additional help, send e-mail to help@xsede.org.
```

Figure 8: User denied SSH access without XSEDE DUO MFA

In his first attempt, the user is denied because his local PSC username (joeuser) is listed in the */usr/xsede/etc/duo-required-users* file, and he has attempted to login using SSH public key authentication. In the second attempt, the user is denied because his local PSC username (joeuser) is in a group listed in the *duo-required-groups* file, and the attempt to use XSEDE Duo MFA has failed since the user's XSEDE username (joeXsede) is not recognized as an enrolled in the XSEDE Duo service account.

Figure 9 depicts a correctly authenticated, XSEDE SSO GSI-OpenSSH login session including XSEDE Duo MFA on Bridges at PSC. Figure 10 depicts the user interface of the Duo client app for an Apple IOS device with an XSEDE Duo MFA authentication transaction in progress.

```
$ ssh -p 2222 -l joeXsede bridges.psc.xsede.org

XSEDE Authentication
password:

Duo two-factor login for joeXsede

Enter a passcode or select one of the following options:

 1. Duo Push to XXX-XXX-1234
 2. Phone call to XXX-XXX-1234

Passcode or option (1-2): 1

Pushed a login request to your device...
Success. Logging you in...

Last login: Thu Jun  8 16:59:12 2017 from example.psc.edu

****************** W A R N I N G **********************
You have connected to br006.pvt.bridges.psc.edu

This computing resource is the property of the Pittsburgh
Supercomputing Center. It is for authorized use only.  By
using this system, all users acknowledge notice of, and
agree to comply with, PSC polices including the Resource
Use Policy, available at
http://www.psc.edu/index.php/policies. Unauthorized or
improper use of this system may result in administrative
disciplinary action, civil charges/criminal penalties,
and/or other sanctions as set forth in PSC policies. By
continuing to use this system you indicate your awareness
of and consent to these terms and conditions of use.

LOG OFF IMMEDIATELY if you do not agree to the conditions
stated in this warning

Please contact remarks@psc.edu with any comments/concerns.

****************** W A R N I N G **********************

[joeuser@br005:~]$
```

Figure 9: Successful XSEDE user authentication to PSC Bridges via XSEDE SSO GSI-OpenSSH plus XSEDE Duo MFA

Maintenance of the group and username entries in the *duo-required-groups* and *duo-required-users* files on Bridges login nodes has been automated to ensure consistency of enforcement on all Bridges login nodes. PSC account management personnel add and remove user accounts from designated MFA-required groups as needed to enforce XSEDE Duo MFA use immediately for SSH login access, as required for projects requiring strong user authentication.

8. Summary

This paper describes the implementation of flexible enforcement of Duo multi-factor authentication with SSH via Linux-PAM for XSEDE federated identity users on PSC's Bridges supercomputer. The solution builds on XSEDE's efforts to provide single sign-on SSH access to XSEDE SP resource systems, as well as XSEDE Duo MFA for strong authentication needs. Modifications were made to the source code for Duo's pam_duo.so module to facilitate security and flexibility of the solution. The design and implementation take into account alternate OpenSSH-native authentication means that may bypass PAM-initiated Duo MFA in the PAM *auth* stack. Error messages reported to users inform them when they are required to use XSEDE Duo MFA for access, and if needed, where they need to go to enroll for XSEDE Duo MFA.

The implementation described has been in production operation to enforce strong authentication of users with access to restricted data on PSC's Bridges supercomputer since October 2016. We believe the design and implementation described here is readily extensible to similar deployments for MFA with SSH via Linux-PAM.

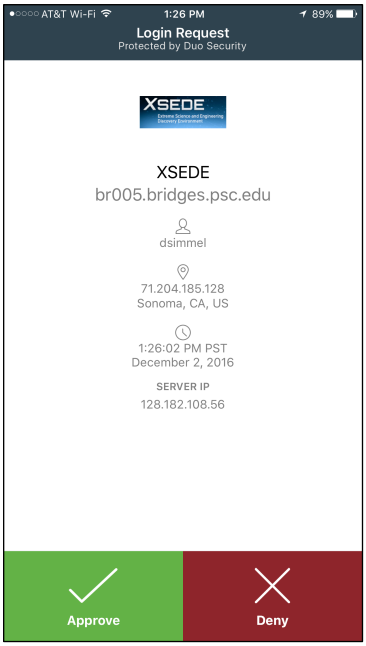

Figure 10: Duo IOS Client Authentication for XSEDE Duo MFA

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. ACI-1445606, "Bridges: From Communities and Data to Workflows and Insight" and Grant No. ACI-1548562, "XSEDE 2.0: Integrating, Enabling and Enhancing National Cyberinfrastructure with Expanding Community Involvement." We would like to acknowledge the contributions made by Venkatesh Sri Yekkirala at the National Center for Supercomputing Applications for his XSEDE-sponsored efforts to enable the use of federated identities with Duo multi-factor authentication.

REFERENCES

[1] Nystrom, N. A., Levine, M. J., Roskies, R. Z., and Scott, J. R.. 2015. Bridges: A Uniquely Flexible HPC Resource for New Communities and Data Analytics. In Proceedings of the 2015 Annual Conference on Extreme Science and Engineering Discovery Environment (St. Louis, MO, July 26-30, 2015). XSEDE15. ACM, New York, NY, USA., 2015. doi:http://dx.doi.org/10.1145/2792745.2792775

[2] The Extreme Science and Engineering Discovery Environment (XSEDE). Website accessed March 13, 2017: https://www.xsede.org/

[3] Duo Security, Inc. Website accessed March 13, 2017: https://duo.com

[4] Duo Unix - Two-Factor Authentication for SSH with PAM Support (pam_duo). 2017. Duo Security Inc. Website accessed March 13, 2017: https://duo.com/docs/duounix

[5] Pluggable Authentication Modules for Linux (Linux-PAM). Website accessed March 13, 2017: http://www.linux-pam.org

[6] OpenSSH. Website accessed March 13, 2017: https://www.openssh.com/

[7] GSI-OpenSSH: GSI-Enabled OpenSSH. 2017. Champaign-Urbana, Illinois: National Center for Supercomputing Applications. Website accessed March 13, 2017: http://grid.ncsa.illinois.edu/ssh/

[8] Interoperable Global Trust Federation (IGTF). Website accessed March 13, 2017: https://www.igtf.net

[9] SSO Hub with MEG PAM Module and OpenSSH. 2017. Champaign-Urbana, Illinois: National Center for Supercomputing Applications. Website accessed March 13, 2017: http://grid.ncsa.illinois.edu/myproxy/ssohub/

[10] GitHub duosecurity/duo_unix repository: Accessed March 13, 2017: https://github.com/duosecurity/duo_unix

[11] Multi-Factor Authentication (MFA) with Duo. XSEDE Community Software Repository (CSR). Website accessed March 13, 2017: https://software.xsede.org/software-and-service-component/multi-factor-authentication-mfa-duo

[12] MyProxy Credential Management Service. 2017. Champaign-Urbana, Illinois: National Center for Supercomputing Applications. Website accessed June 8, 2017: http://grid.ncsa.illinois.edu/myproxy/

[13] XSEDE User Portal. Website accessed June 8, 2017: https://portal.xsede.org/