Minimum Requirements for Integrating Services with Central Authentication  
Version 1.0 – December 2008

To better safeguard the University’s data and resources, the IT Security Office requires the following practices. These requirements apply to any service which takes advantage of electronic IDs issued through CIT’s central identity management system based on Kerberos—currently NetID, ApplicantID and GuestID. CIT also issues VMIDs for access to legacy applications. Those applications are not within scope of these requirements. Requirements and best practices for implementing federated access to services are out of scope.

A. Exceptions

For all services that are not able to meet these security requirements the following exception process must be followed:

1. Services that cannot meet the following requirements must be identified to the appropriate Unit Security Liaison. The Unit Security Liaison will look for alternate methods to address the risk that is being addressed. If an alternate security solution can be found to address the specific risk, an exception is not required.

2. The Unit Security Liaison may determine a local solution or consult with the IT Security Office for assistance.

3. The Unit Security Liaison will maintain a list of all IT resources that require an exception and review them on an annual basis.

B. All Services

1. All services must adhere to Baseline IT Security Requirements and any additional security requirements based on the level of data being protected:  
   http://www.cit.cornell.edu/security/requirements/

2. The developer will subscribe to the CIT discussion list cuwa-announce-l.

3. Best practices for securing any key material \(^1\) such as keytabs\(^2\) and certs must be followed.

---

\(^1\) The term key material refers to the sequence of electronic values, stored in a file, which could be used to decipher encrypted communications or impersonate a network entity such as a server or application. The MIT Kerberos protocol refers to this file as a srvtab (Kerberos 4) or keytab (Kerberos 5).

\(^2\) A keytab is a file which contains a Kerberos Version 5 principal and the mathematical hash of that principal's password. It has two uses: (1) to allow a service (application or web server) to identify users on campus via CUWebLogin or other CIT authentication infrastructure, and (2) to allow a service (application or web server) to identify itself to other servers on campus.
4. A named Cornell staff member must be responsible for key material associated with any externally-hosted service using CIT authentication services and must sign a formal agreement. The staff member will subscribe to the CIT discussion list cuwa-announce-l and will be responsible for ensuring that the integration is kept current.

5. Any application developer using native Kerberos libraries is responsible for understanding how Kerberos works and how to implement the protocol securely. Sources of information about the Kerberos protocol include the MIT site: http://www.kerberos.org/software/tutorial.html and a book in the O’Reilly series entitled Kerberos: the Definitive Guide by Jason Garman.

6. All externally hosted services must be reviewed on a case by case basis by the IT Security Office before CIT authentication services are extended for its use.

C. Further information and assistance

1. Additional information about integrating with CIT-supported authentication services is available at:  http://identity.cit.cornell.edu/

2. Address specific questions about these requirements or about taking advantage of CIT’s authentication services to: idmgmt@cornell.edu

D. Specific to web-based authentication

CIT-supported solutions are CUWebAuth, which will meet the needs of most campus developers for deploying web-based services which require authentication: https://identity.cit.cornell.edu/authn/index.html and Shibboleth, which may be the preferred option for services hosted externally: https://identity.cit.cornell.edu/authn/shibbolethintro.html

1. Use SSL in most cases.
   - Production services should use SSL certs signed by a recognized authority, where user is prompted for allow/deny.
   - CIT’s SSL cert service is recommended: http://identity.cit.cornell.edu/ssl/

2. The integrity of the keytab must be maintained
   - Use a different keytab for each application or security domain.
   - Never use a developer keytab for a production service.
• Never use the developer keytab in development and test environments housing production data which is either sensitive or confidential.

3. Never use the basic web authentication prompt supplied with the browser

4. Write application in such a way as to require authorization either within the application or within CUWebAuth.

5. If CUWebAuth is implemented for an external service, a named Cornell staff member must be responsible for the key material and the integration. An agreement to this effect must be signed by the staff member. Link to agreement here.

E. Specific to native Kerberos

MIT provides two API’s for developing native Kerberos applications: GSSAPI and KRB5 API. GSSAPI is recommended, but in some circumstances KRB5 API is a better choice. You can download documentation, libraries, and source code at http://web.mit.edu/Kerberos/

1. The application should not directly prompt the user for his or her Kerberos password unless it is an approved Kerberos proxy (see section H).

2. Care should be taken to insure message privacy and integrity.
   • We recommend that you use SSL when transmitting credentials and application data.
   • You can also use a combination of Kerberos or GSSAPI function calls to build message privacy and integrity into your application.

3. Prior to implementing in production consult with IT Security and have your design and code reviewed. Submit your request to security-services@cornell.edu

4. The requirements relating to the use of developer keytabs outlined in section D apply.

F. Specific to CUWebAuth or Kerberos authenticated password reset

If an application requires a separate password database, one can build a self-service password reset application that uses CUWebAuth to authenticate a user through Kerberos as a prerequisite for creating or resetting the password for an application account.

1. Your page for entering application passwords MUST contain strong wording discouraging users from using their Kerberos password for their application password.
This is covered in University policy 5.8, Authentication of IT Resources: http://www.policy.cornell.edu/vol5_8.cfm

2. Before accepting the password change check the password to ensure it doesn’t match the NetID password. Do not capture the password for this purpose. Need either reference web service or documented method for doing this.

3. If you accept user passwords log “non-existent” account notation only. Security Engr needs to weigh in. Need to address logging of failed attempts. Open question about whether technical solution creates a greater risk.

4. Prior to implementing in production you must consult with IT Security. Submit your request to security-services@cornell.edu

5. The requirements in section D apply.

G. Active Directory and other cross-realm Kerberos authentication

Cross-realm authentication can be established between an Active Directory or other Kerberos instances and CIT’s Kerberos database. Many Microsoft applications and functions like domain login can then use the CIT- maintained NetID and password for authentication. Some set of operations still requires the internal Active Directory password.

1. Treat the password used for cross-realm trust as key material.

2. If the internal AD password is needed the domain administrator should advise end users to keep it separate from the NetID password.

3. If the internal password is not needed we strongly recommended that you generate random strong passwords. Do not actively synch them with the NetID password.


H. Kerberos proxy

If an application is not written to take advantage of Kerberos, another server can be set up to ask the user for the password (in encrypted form) and authenticate to Kerberos on behalf of the user. The server is called a Kerberos proxy. This approach to authentication is considered a last resort. Direct access to NetID passwords has University policy implications. Refer to University Policy 5.8: Authentication of IT Resources http://www.policy.cornell.edu/vol5_8.cfm.

CIT maintains a small number of Kerberos proxies: CUWebLogin, Radius, central email
via TLS, NetID activation, NetID password reset, NetID Administrator, Network News.

1. A Kerberos proxy must be approved by a governing body. *IT Security Council?*

2. The confidentiality of the password must be protected.
   - Passwords must be transmitted in encrypted form only. Use of transport layer encryption such as SSL is strongly advised.
   - Passwords must never be stored on a server or captured in audit logs (including any debug output produced by the application) including any encrypted form. Note that passing a password to another process as a command line parameter can expose the password.
   - Use the password to obtain a Kerberos (time limited) credential and then destroyi the password immediately.

3. The server(s) must be housed in the CIT Server Farm or comparable facility

4. The service must be hosted at Cornell, not at an external site

5. All system administration must be performed by a permanent Cornell staff member in an IT job title. Others should not have login privileges to the system.

6. Multi-factor authentication, such as SecurID, must be used for all administrative interactive login to the server.

7. Audit reports detailing admin access to the server must be available; these should be tamper evident where possible.

8. Log files detailing user authentication events (includes event time, NetID, and IP Address) must be available to the IT Security Office.

9. Prior to implementing in production consult with IT Security and have your design and code reviewed. Submit your request to security-services@cornell.edu

**I. Specific to shared machine/lab machine**

Lab machines that authenticate to another server using NetID and password need risk mitigation measures. If a small group of users requires access to the machine, local authentication is preferred.

1. When the system is administered by a student
   - The unit should require a signed agreement on file.
• A named Cornell staff member must be responsible for key material associated with the system. The staff member will subscribe to the CIT discussion list cuwa-announce-l and will be responsible for ensuring that the integration is kept current.

• A separate administrative account should be used and changed out each semester.

2. The confidentiality of the password must be maintained.

  • To log in to the machine the password or its equivalent must not be transmitted over the network. Use Kerberos or equivalent mechanisms.

  • Passwords must never be stored on the workstation or lab machine (even in encrypted form) or captured in audit logs (including any debug output produced by the application). Note that passing a password to another process as a command line parameter can expose the password.

  • Use the password to obtain a Kerberos (time limited) credential and then destroy the password immediately.

5. Controlled access to the area where the lab machines are housed is strongly encouraged.

6. Audit reports detailing admin access to the server must be available; these should be tamper evident where possible.

7. Log files detailing user authentication events (includes event time, NetID, and IP Address) must be available to the IT Security Office.

**J. When using local authentication...**

Integrating some services with central authentication may not be feasible and the use of a Kerberos proxy not justified. These services will use local accounts and passwords for authentication. There are some standards and best practices which will contribute to security and the integrity of the central authentication service.

1. Avoid the term “NetID” to refer to the user account even when the same character string is used as an account.

2. Instruct the user to set a password which is separate from his Cornell NetID password.
Appendix A – Additional Information on Securing Your Application

Writing Secure Code, 2nd edition, by Howard et al. is recommended text on this topic.

More on SSL: You are assuming some risk by using CUWebAuth without SSL. An order form and best practices for SSL certificates are available at: http://www.cit.cornell.edu/services/identity/sslcert/

More on ketabs: Anyone with a copy of your keytab can bypass all authentication and impersonate any user to your system. The risk is exposure of data on the server and of rights that were granted to that server for access to other information. These practices will greatly reduce the risk of such exposure:
   a) Read access to the keytab must be carefully controlled. In general it should be readable only by the uid under which the application runs.
   b) The keytab file must never be transmitted over the network in the clear (such as through e-mail).
   c) Make sure that your backup system does not expose the keytab by excluding it from the backup.
   d) If you have reason to believe the keytab has been compromised, request a new one through ServiceID Manager: https://serviceidmanager.cit.cornell.edu/serviceidmanager
   f) If you retire the application expire the keytab via ServiceID Manager.

More on internal authorization and “Require valid –user”: Service entitlements are based on the affiliation (student, staff, alum) or role of the individual (e.g. student registered for CS101). The authorization step, not the authentication step, accomplishes this access decision. Blackboard is an example of where internal authorization tables allow access only to authenticated individuals whose NetIDs have been assigned access to specific courses. Without internal security tables, an application with this parameter set will allow access to anyone with an active NetID. Because that group is so large and diverse it begs the question of whether it’s even meaningful to require authentication at all.

More on developer keytabs: The standard of one keytab per application or service may constitute a serious bottleneck for developers. The reuse of a developer keytab in a development or test environment is acceptable if no production restricted or confidential data is at risk. Submit a request for a developer keytab to idmgmt@cornell.edu. The login dialogue box will display a disclaimer when a developer keytab is installed on the server.

More on password reset mechanisms: Passing a user supplied password or any other data in a string to exec or as part of a SQL query without escaping is asking for problems. Try to use other methods to interact with the native password reset mechanism.
   a. Exec(“app-passwd foo a & rm-rf /”) is not what you want but will happen if user foo supplies password “a & rm-rf /”
   b. Similarly, if the password is inserted into a SQL update query directly it is easy
for an attacker to choose a password that has side effects (like deleting a table or changing someone else’s password too).

**More on Kerberos proxies:** CIT maintains a small number of Kerberos proxies: CUWebLogin, Radius, central email via TLS, NetID activation, NetID password reset, NetID Administrator, Network News. A Kerberos proxy will be implemented only after approval by a governing body. Criteria to be used: 1) No other alternative available to meet business need, 2) application used campus-wide or deemed mission-critical.
Follow-up items

- Awareness/communication campaign around proactive measures CIT will take upon detection of service which does not adhere to these standards, including immediately disabling a service which is protecting confidential data. Unauthorized Kerberos proxies fall in to this category.
- Link to this document from CUWebAuth site and/or ServiceID Manager
- Identifying governing body for Kerberos proxy decision
- Need process for approving Kerberos proxy and registering authorized proxies
- CIT should be able to articulate to campus the justification for each Kerberos proxy it runs
- Need process for annual review/expiration of keytabs
- CIT should consider writing web service which would check to make sure a particular password choice is not the same as the NetID (Kerberos) password
- Consider whether there is a follow-up item requiring that named, regular staff are associated with all lab environments.