

## Minimum Standards for Systems Integrating with Central Authentication Services

Integration Method	Description	Minimum Standards
CUWebAuth	CIT's solution for web-based applications, available for Unix, MacOS X and Windows server environments.	<ol style="list-style-type: none"> <li>1) Follow minimum standards established for protecting the category of data found on the server.</li> <li>2) SSL should be used in most cases and best practices followed for maintaining the security of the certificate. You are assuming some risk by using CUWebAuth without SSL. An order form and best practices for SSL certificates are available at: <a href="http://www.cit.cornell.edu/services/identity/sslcert/">http://www.cit.cornell.edu/services/identity/sslcert/</a></li> <li>3) The security of the srvtab (Kerberos 4) or keytab (Kerberos 5) must be maintained. Anyone with a copy of your srvtab or keytab can bypass all authentication and impersonate any user to your system.               <ol style="list-style-type: none"> <li>a) Read access to the srvtab must be carefully controlled. In general it should be readable only by the uid under which the application runs.</li> <li>b) The srvtab file must never be transmitted over the network in the clear (such as through e-mail).</li> <li>c) Make sure that your backup system does not expose the srvtab or keytab.</li> <li>d) If you have reason to believe the srvtab or keytab has been compromised, contact the Identity Management team to have a new one issued.</li> <li>e) In general, you should have a different srvtab or keytab for each application. Consult with the Identity Management staff for implementation assistance as needed.</li> <li>f) If you retire the application you must notify the Identity Management staff so the srvtab or keytab can be expired</li> </ol> </li> <li>4) The configuration parameter "Allow valid user" should be</li> </ol>

## Minimum Standards for Systems Integrating with Central Authentication Services

		used only if the application manages authorization internally.
Native Kerberos	Libraries and CIT-developed components such as CASAPI and SideCar are available to enable applications to communicate directly with Kerberos.	<ol style="list-style-type: none"> <li>1) Points 1 and 3 for CUWebAuth apply.</li> <li>2) The application developer is responsible for understanding how Kerberos works and how to implement the protocol securely</li> <li>3) The application should not directly prompt the user for his or her Kerberos credentials unless it is an approved Kerberos proxy (see rules below).</li> <li>4) Application content should be encrypted in transmission. The krb_priv or gss wrap calls do this correctly. The use of SSL is usually better than a custom cryptographic protocol.</li> </ol>
Kerberos-Aware 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.	<ol style="list-style-type: none"> <li>1) See CUWebAuth section above.</li> <li>2) Your page for entering application passwords <b>MUST</b> contain strong wording discouraging users from using their Kerberos password for their application password.</li> <li>3) Passing a user supplied password 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. <ol style="list-style-type: none"> <li>a. Exec("app-passwd foo a &amp; rm-rf /") is not what you want but will happen if user foo supplies password "a &amp; rm-rf /"</li> <li>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).</li> </ol> </li> </ol>
Active Directory	Cross-realm authentication can be established between an Active Directory instance and CIT's Kerberos database. Many Microsoft	<ol style="list-style-type: none"> <li>1) Points 1 and 3 for CUWebAuth apply; treat the password used for cross-realm trust as a keytab for the purposes of interpreting this point.</li> <li>2) The domain administrator should advise end users to keep the</li> </ol>

## Minimum Standards for Systems Integrating with Central Authentication Services

	<p>applications and functions like domain login can then use the CIT-maintained NetID and password for authentication. A relatively small set of operations still requires the internal Active Directory password.</p>	<p>internal Active Directory password separate from the NetID password associated with centrally authenticated services.</p>
<p>Kerberos Proxy</p>	<p>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. 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. <b>This approach to authentication is considered a last resort.</b></p>	<ol style="list-style-type: none"> <li>1) Direct access to NetID passwords has university policy implications. Do not take this step lightly.</li> <li>2) Passwords must be transmitted in encrypted form only. Use of transport layer encryption such as SSL is strongly advised.</li> <li>3) Passwords must never be stored on a server 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.</li> <li>4) Using the password to obtain a Kerberos (time limited) credential and then destroying the password immediately is strongly encouraged.</li> <li>5) The server(s) must be housed in the CIT Server Farm</li> <li>6) 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.</li> <li>7) Multi-factor authentication, such as SecurID, must be used for all interactive login to the server.</li> <li>8) Follow minimum standards established for protecting the category of data found on the server. Since the server handles the credentials that can be used to access other university data (including protected personal information such as SSN's) the minimum standards will include the minimum standards for</li> </ol>

## Minimum Standards for Systems Integrating with Central Authentication Services

		any other systems using NetID authentication. 9) Audit reports detailing admin access to the server must be available; these should be tamper evident where possible. 10) Log files detailing user authentication events (includes event time, NetID, and IP Address) must be available to the IT Security Office
--	--	---

Additional information about integrating with CIT-supported authentication services is available at:

<http://identity.cit.cornell.edu/>

Address specific questions about these standards or about taking advantage of CIT's authentication services to:

[aadssupport@cornell.edu](mailto:aadssupport@cornell.edu)