Configuring SSL Securely

With recent revelations about state sponsored intelligence gathering efforts I wanted to capitalize on our attention to them and suggest a way we, as IT professionals, can make a positive impact on privacy and the security of business our students and employees conduct with the University. Some previously unknown capabilities of our government to intercept and process protected Internet traffic has come to light [1]. If our government can do it you can bet the those in the cyber underworld can as well.

In the wake of all this snooping what can we do? We can make sure web services and applications are running over encrypted transport mechanisms, like HTTPS! True, but don't get too excited there a second important piece to this commonly missed in secure transport mechanism, in this discussion HTTPS, administration.

We can make sure web services and applications are running over encrypted transport mechanisms configured to default to secure encrypted configurations, like HTTPS using Perfect Forward Secrecy supporting ciphers and disabling vulnerabilities! [2] Encryption ciphers that employ Perfect Forward Secrecy (PFS) protect privacy be creating and using temporary encryption keys unique to a session and preserve privacy even if the private key becomes compromised. [3] Below are three critical steps to making the effort to secure HTTP traffic with SSL highly effective.

1. Remove weak cipher suites, like DES & 3DES, from use.
2. Disable SSLv2 support.
2. Make cipher suites that support PFS, like ECDHE-*, prefered.

Every HTTP server has different capabilities and ways to leverage them but below are links to some I know are in use at University of Alaska.
Apache:
http://www.lorrin.org/blog/2013/07/03/configuring-apache-for-perfect-forward-secrecy/

Microsoft IIS:
http://support.microsoft.com/kb/187498 Disable SSLv2
http://www.waynezim.com/2011/03/how-to-disable-weak-ssl-protocols-and-ciphers-in-iis/
Disable weak SSL ciphers
To date I have no references that demonstrate IIS supports PFS cipher suites. If you learn of any please let me know.

Oracle Weblogic:
http://docs.oracle.com/cd/E23943_01/web.1111/e13707/ssl.htm#i1194557 Manage SSL protocols
http://docs.oracle.com/cd/E23943_01/web.1111/e13707/ssl.htm#BABBDACC Manage cipher

suites
Support for cipher suites providing PFS do not seem to be documented here.

Apache Tomcat:
http://www.techstacks.com/howto/secure-ssl-in-tomcat.html

Now that you have put valuable time and effort into making improvements checking your work is positive final step. Qualys has a free SSLLab service that can do that. It probes and ranks the strenght of HTTPS options enabled on a server https://www.ssllabs.com/ssltest/. This is also a great place to start your assessment.

At the end of the day the individual using a web service or application also plays a role in how secure an interaction is but by configuring to not allow insecure options and default to highly secure choices gives us the best opportunity to have high confidence in the security of the business conducted with the University of Alaska.

[1] http://www.nytimes.com/2013/09/06/us/nsa-foils-much-internet-encryption.html?hp&_r=1&

[2]
https://www.eff.org/deeplinks/2013/08/pushing-perfect-forward-secrecy-important-web-privacy-protection

[3] http://vincent.bernat.im/en/blog/2011-ssl-perfect-forward-secrecy.html