Purpose

This document provides the processes and necessary guidelines to:

1. Maintain the integrity of network systems and university data by applying the latest OS and application security patches and/or updates in a timely manner.
2. Establish a baseline methodology and timeframe for patching and confirming appropriate patch-management compliance.

University owned servers represent access points to sensitive and confidential University data. Ensuring that security updates and patches are distributed and implemented in a timely manner is essential for mitigating malware, exploitation, and other threats.

Scope

The processes addressed in this document affect all servers on campus connected to our network and managed by Tech Support Services teams. Server definition in this context being “A computer connected to the network that provides files and services to other computers on or off of the campus network.”

Responsibility

Tech Support Team

1. Patch installation for server systems, including operating system and application patches*.
2. Work with customers when problems arise from patching test systems, or application vendors do not support the latest patches for their applications.
3. Direct customers to server exception form when patches cannot be applied.
4. Maintain a working knowledge of patches that are deployed in the respective areas.

Enterprise Security

4. Apply appropriate security measures to systems that cannot be patched.
5. Work directly with systems owners with systems on the exception list.
6. Maintain a working knowledge of patches that are deployed in the respective areas.

Application Administrators

1. Patching/updating applications that are running on university owned servers. Full responsibility for application patches.
2. Working with vendors to obtain application patches when vulnerabilities are discovered.
3. Responsible for all patching that is not maintained by the Tech Support Team.
*Application Patches: Any patches that do not fall in to the following categories – Operating System, .NET, Microsoft Service Packs. Critical Java, Adobe, and Web Browser exploits may be addressed directly by tech support personnel depending on severity.

Process

1. As soon as patches are made available, download patches from a trusted source.
2. Test patches where applicable to identify any adverse effects.
3. Be sure that any patches being applied are brought to infrastructure and/or change management meetings and the date at which they are applied is made known through the applicable communication channels.
4. Communicate changes to affected parties. Change management is an open forum, application administrators should be encouraged to attend.
5. Deploy patches
   a. Windows Servers
      i. Monthly patches-test: Patches are to be applied as soon as they are released to all test systems. Any problems encountered must be communicated to appropriate parties as soon as possible.
      ii. Monthly patches: Patches are to be applied no later than the second weekly maintenance window of each month or the next maintenance window after “Patch Tuesday” for production systems.
      iii. Out-of-Band security patches: Deploy no later than two weeks following release to production systems. Test systems must be patched as soon as possible.
   b. UNIX/Linux Servers
      i. Patches that correct a known critical security vulnerability need to be deployed as soon as possible, and no later than two weeks following release.
      ii. Important and Moderate vulnerabilities must be applied at least once per month.
      iii. Deploy patches at least once per month to production systems.
      iv. IBM AIX patches will be applied based on IBM best practices.
      v. Oracle database patches will be applied within a 30 day window after quarterly patch release.
   c. Applications
      i. Patches that correct a known critical security vulnerability need to be deployed as soon as possible, and no later than two weeks following release.
      ii. Patches that correct non-critical vulnerabilities must be applied no later than thirty days after a patch is released.

1: Critical Impact Vulnerability – Flaws that could easily be exploited by a remote unauthenticated attacker and lead to system compromise (arbitrary code execution) without requiring user interaction. These are the types of vulnerabilities that can be exploited by worms. Flaws that require an authenticated remote user or local user are not considered critical.

2: Important Impact Vulnerability – Flaws that can easily compromise the confidentiality, integrity, or availability of resources. Vulnerabilities that allow local users to gain privileges, allow unauthenticated users to view resources that should be protected by authentication, allow authenticated remote users to execute arbitrary code, or allow local or remote users to cause a DoS.
Moderate Impact – Increased difficulty to exploit. Some instances could be considered critical or moderate, but are more difficult to exploit, or the vulnerability only affects unlikely software configurations.

Exceptions

1. Systems or applications that cannot be patched to resolve a known vulnerability will have the justification documented by the device/application owner, and necessary security controls will be implemented to mitigate the vulnerability until the system can be patched. Required in instances where patches are unable to be applied in a timely manner.
   a. Justification
      i. No patches from application vendor are available.
      ii. Patches create instability within the system, and the instability outweighs the risk.
   b. Security Controls
      i. Network segmentation
      ii. Access Control Lists
      iii. Host-Based Intrusion Prevention System

2. Systems that transmit or store protected data and cannot be patched to resolve a known vulnerability will be brought to the attention of the data owner. This information will be given to the Enterprise Security Team and necessary security controls to compensate for the vulnerability will be implemented.

Patch-Compliance

Enterprise Security Group will analyze various reporting tools when determining patch compliance and items that require action of another group.

1. System Center Configuration Manager will be used to determine patch levels of production and test Windows servers.
2. Red Hat Satellite Server will be used to determine patch levels of production and test Red Hat servers.
3. Rapid7 Nexpose will be used in conjunction with other tools to determine patch levels of production and test systems.
4. Symantec Control Compliance Suite can be leveraged to determine patch levels of systems that are not managed by mechanisms in this list.

Periodic Vulnerability Assessment

Periodic vulnerability assessments will be performed per the below time frames.

1. Weekly internal vulnerability scan via Nexpose
2. Quarterly external vulnerability scan via Nexpose
3. Monthly patch compliance review for Windows and Linux/Unix systems utilizing methods outlined in “Patch-Compliance” section of this policy.