

Microsoft IIS Hardening Checklist

Introduction:

This checklist contains web server hardening procedures that balance industry best practices with the unique requirements of UCD's environment. Since Microsoft IIS does not come configured securely out of the box it is necessary to follow these steps to prevent attacks from exploiting known vulnerabilities. This document assumes that IIS 6.0 is in use – if the server is running IIS 5 or earlier, every effort should be made to upgrade to IIS 6. While this checklist is relatively short, an IIS 5 checklist would be many pages in length. If IIS 5 must be used ongoing, special care should be followed to implement the detailed IIS 5 hardening steps provided on Microsoft Technet.

These steps should be followed to secure a typical UCD IIS server, but may not be appropriate in all cases. In cases where an exception must be made, documentation should be retained on this worksheet describing the reason for the exception and any mitigating actions. In all cases, this worksheet should be retained for future reference.

Procedure:

- Complete Windows server hardening checklist. Ideally, run IIS on Windows 2003 servers.
- On installation, choose custom install and DO NOT install the following management protocols: FrontPage 2000 Server Extensions, Internet Service Manager (HTML), and Visual InterDev Remote Support. If FrontPage Extensions are in use, special steps (described on TechNet) are required to secure them.
- Install the appropriate IIS security hotfixes from Microsoft. This is particularly important, since some patches may be missed by Windows Update.
- Limit the exposure of the IIS service – if it's a development box, only allow connections from trusted IPs. If it's a campus box, ensure it's not visible to the Internet. If it is going to be Internet visible, ensure it's on a DMZ.
- Disable bundled network services that aren't explicitly necessary, including NNTP, FTP, and SMTP. Each of these services can put the server at significant additional risk.
- Assign the default web site to the localhost (127.0.0.1) address. This ensures that any site that is served has been intentionally configured. This approach is preferable to completely removing the default site (as recommended by Microsoft), as it allows for better flexibility in the future (should the default site become necessary). This avoids a ton of hardening steps, like removing default site files and changing default site settings.
- Review each ISAPI application in the WWW Service Master Properties. Any extension that's not actually in use on the site should be removed from this mapping. This is a terribly important step and the source of most IIS vulnerabilities in the past.
- Review access and error log settings and ensure each web server access is being recorded. Ideally, configure centralized logging for this data.
- Sites should not be configured to NOT allow "script source access", which permits loading of executable code across the web.
- Directory browsing should be disabled, so if an index page does not exist at least a directory listing isn't presented.
- If ASP scripts are required for the site, the "application protection" option should be set to High (or at least Medium) security.



- ❑ Review each ISAPI application in the WWW Service Master Properties. Any extension that's not actually in use on the site should be removed from this mapping. This is a terribly important step and the source of most IIS vulnerabilities in the past.
- ❑ If any sensitive or Personally Identifiable Information is to be stored on the web server (or available from it), all communications should be over HTTPS and protected with "real" SSL certificates (not self-signed ones).
- ❑ Once IIS is installed and configured, perform a manual backup of the IIS configuration. Store this file in a central repository so the configuration can be easily restored after a server failure.

