Idaho Technology Authority (ITA)

ENTERPRISE GUIDELINES G500 – SECURITY PROCEDURES

Category: G590C – PUBLIC-FACING WEB SERVER SETUP

CONTENTS:

I. DEFINITIONS

A. Web Server – System that hosts content published for the world-wide web.

B. Public-Facing (or DMZ) – Area of the state network that separates the public outside network from the internal private network.

II. RATIONALE

The purpose of this guideline is to provide a security baseline for State of Idaho server administrators to use in hardening their web servers. The parameters in this guideline are widely accepted by the global security community as prudent and effective.

III. GUIDELINE

This guideline is part of the G590 series and it addresses hardening of the Web server environments. Implementing this guideline will better secure all state-used web servers in accordance with *ITA Enterprise Standard S3230 – Server Security Requirements*. This addresses the third phase in Standard S3230; each procedure references that by stating it addresses "a Phase 3 finding."

IV. PROCEDURE REFERENCE

The following pages will address these procedures:

A. Remove, or do not install, unneeded services.

B. Remove, or do not install, tutorial, documentation, or development software/files.

C. Run web server process under restricted account.

D. Keep web content outside of default installation locations.

E. Disable debugging.

F. Limit use of CGI or other executable programs.

G. Define and limit access by web services.

H. Keep log files outside of default installation locations.

I. Keep temporary files outside of default installation locations.

J. Allow only Secure FTP connections.

K. Use HTTPS for all web form submissions.

L. Use robots.txt file.

A. Remove, or do not install, unneeded services.

1. **Details:** For any public-facing web servers, disable WebDAV. Unless absolutely required, remove or disable the file and print sharing, DNS, SQL Server, and SMTP (email) services. Use web services to format web-based email and send them to one of the State's enterprise IronMail units for relay.

2. **Description:** This is a Phase 3 finding because certain server-level services are installed by default when a web server application is installed on a server.

   Some services, such as WebDAV, are specifically for non-public, non-production systems and should never be installed or enabled on a public-access, and/or, production-level web server.

   Others, like file and print sharing, DNS, SQL Server, and/or SMTP services are not necessary for web server operation. Having these installed and enabled open the server to non-web-specific attack vectors.

3. **Solution:** When installing services on the server, install only those services that are absolutely necessary. If services are installed without express consent (for example, WebDAV on older Windows platforms or SendMail on Linux systems), use the server's services management application to disable them and prevent them from starting automatically on system boot-up.

4. **References:**

   a. NIST Special Publication 800-44
   http://csrc.nist.gov/publications/nistpubs/800-44-ver2/SP800-44v2.pdf

   b. IIS 6 Best Practices
   http://technet.microsoft.com/en-us/library/cc782762(WS.10).aspx

   c. IIS 7 Best Practices
   http://technet.microsoft.com/en-us/library/cc721652.aspx

   d. Best Practices In Managing World Wide Web Server Security
   http://www.boran.ch/security/webserver_practices.html

   e. Microsoft IIS7 Security Best Practices
   http://searchsecurity.techtarget.com/tip/0,289483,sid14_gci1521981,00.html?track=NL-422&ad=792838USCA&asrc=EM_NLT_12703904&uid=7993407

B. Remove, or do not install, tutorial, documentation, or development software/files.

1. **Details**: Remove any and all documentation files and/or tutorials, development environment software/tools, and/or any other tutorial software and files from the server.

2. **Description**: This is a Phase 3 finding because most web server software installs, by default, server documentation, tutorial, and additional development programs. Although useful in a testing/development environment, these are not acceptable on production-level systems.

3. **Solution:** If possible, do not allow these items to install on the server. If installed by default without express action by the server administrator, physically remove all files and software of this sort from the server.

4. **References**:

   a. NIST Special Publication 800-44
   http://csrc.nist.gov/publications/nistpubs/800-44-ver2/SP800-44v2.pdf

   b. IIS 6 Best Practices
   http://technet.microsoft.com/en-us/library/cc782762(WS.10).aspx

   c. IIS 7 Best Practices
   http://technet.microsoft.com/en-us/library/cc721652.aspx

   d. Best Practices In Managing World Wide Web Server Security
   http://www.boran.ch/security/webserver_practices.html

   e. Microsoft IIS7 Security Best Practices
   http://searchsecurity.techtarget.com/tip/0,289483,sid14_gci1521981,00.html?track=NL-422&ad=792838USCA&asrc=EM_NLT_12703904&uid=7993407

C. Run web server process under restricted account.

1. **Details:** Configure the web server process to operate under an account with limited rights on the server.

2. **Description:** This is a Phase 3 finding because a web server application must run in conjunction with the Operating System and a valid server account must be used to execute the program.

3. **Solution:** Create a user account, with limited server-level privileges, and configure the web server program to run under the authority of that account.

   Disable write access permissions for the user account running the web service.

   Do not assign write and script source access permissions or scripts and executables permissions.

   Deny execute permissions for anonymous users to all executables in Windows directories and subdirectories.

4. **References:**

   a. NIST Special Publication 800-44
   http://csrc.nist.gov/publications/nistpubs/800-44-ver2/SP800-44v2.pdf

   b. IIS 6 Best Practices
   http://technet.microsoft.com/en-us/library/cc782762(WS.10).aspx

   c. IIS 7 Best Practices
   http://technet.microsoft.com/en-us/library/cc721652.aspx

   d. Best Practices In Managing World Wide Web Server Security
   http://www.boran.ch/security/webserver_practices.html

   e. Microsoft IIS7 Security Best Practices
   http://searchsecurity.techtarget.com/tip/0,289483,sid14_gci1521981,00.html?track=NL-422&ad=792838USCA&asrc=EM_NLT_12703904&uid=7993407

D. Keep web content outside of default installation locations.

1. **Details:** Establish a web content area outside of the default settings used by your web server software's installation program. When possible, this web content area should be on a separate physical drive or virtual partition. Disable parent pathing in IIS.

2. **Description:** This is a Phase 3 finding because web server content files provide a ready attack vector to a web server. Most initial web server attacks are targeted to run against web content files and/or web applications located in the standard default web content location (for example, c:\inetpub\wwwroot or /etc/httpd/docs).

   Additional security concerns relate to the prevention of allowing a "directory traversal" attack from gaining unintended access to system-level executable code.

   Placing web content and/or web application code outside of the standard default installation location makes if far harder to successfully exploit these types of attacks.

3. **Solution:** Generate a directory tree outside of the standard default web content location and configure the web server program to use that content area only. If possible to do so, creating a separate partition and/or using a separate physical hard drive is preferred over simply establishing a separate directory structure on the same drive or partition on which the server's operating system resides.

4. **References:**

   a. NIST Special Publication 800-44
   http://csrc.nist.gov/publications/nistpubs/800-44-ver2/SP800-44v2.pdf

   b. IIS 6 Best Practices
   http://technet.microsoft.com/en-us/library/cc782762(WS.10).aspx

   c. IIS 7 Best Practices
   http://technet.microsoft.com/en-us/library/cc721652.aspx

   d. Best Practices In Managing World Wide Web Server Security
   http://www.boran.ch/security/webserver_practices.html

   e. Microsoft IIS7 Security Best Practices
   http://searchsecurity.techtarget.com/tip/0,289483,sid14_gci1521981,00.html?track=NL-422&ad=792838USCA&asrc=EM_NLT_12703904&uid=7993407

E. Disable debugging.

1. **Details:** On all production web servers, disable client and server-side script debugging (in the web server software) and debugging in individual applications (for example, DotNET [in the web.config], ColdFusion [in the ColdFusion Administrator], and PHP [in the php.conf]).

2. **Description:** This is a Phase 3 finding because allowing detailed debugging information to be accessible to the public visitors of a web site gives a malicious visitor a vast amount of information concerning the configuration of the web site as well as the server itself.

   Providing this type of information to a malicious visitor constitutes a major security risk to any production-level web server, especially one that is accessible to the Internet.

3. **Solution:** Set debugging on system-level software to disabled. Set debugging on all web site and application-level programs to disabled or false.

4. **References:**

   a. NIST Special Publication 800-44
   http://csrc.nist.gov/publications/nistpubs/800-44-ver2/SP800-44v2.pdf

   b. IIS 6 Best Practices
   http://technet.microsoft.com/en-us/library/cc782762(WS.10).aspx

   c. IIS 7 Best Practices
   http://technet.microsoft.com/en-us/library/cc721652.aspx

   d. Best Practices In Managing World Wide Web Server Security
   http://www.boran.ch/security/webserver_practices.html

   e. Microsoft IIS7 Security Best Practices
   http://searchsecurity.techtarget.com/tip/0,289483,sid14_gci1521981,00.html?track=NL-422&ad=792838USCA&asrc=EM_NLT_12703904&uid=7993407

F. Limit use of CGI or other executable programs.

1. **Details:** If possible, do not use Common Gateway Interface (CGI) or other executable programs. If use of CGI or other executable programs is required, keep all such programs in a separate directory tree.

2. **Description:** This is a Phase 3 finding because executable programs and/or subroutines within a web site are a major vector for security breaches and attacks.

   Most current web application development programs use a system-level service or interpreter to perform processing of web application code. These system-level services (such as DotNET and ColdFusion) or interpreters (such as PERL and PHP) are configured to allow only specific system-level functions to be executed. Limiting these programs is a key to preventing malicious visitors from gaining access to operating system level tools that would allow them to take control of a server.

3. **Solution:** Where possible, require the use of known, tested, and trusted application and interpretive software. If a custom-built executable program is required, ensure that it is separate from the operating system core programs and all basic web content files.

4. **References:**

   a. NIST Special Publication 800-44
   http://csrc.nist.gov/publications/nistpubs/800-44-ver2/SP800-44v2.pdf

   b. IIS 6 Best Practices
   http://technet.microsoft.com/en-us/library/cc782762(WS.10).aspx

   c. IIS 7 Best Practices
   http://technet.microsoft.com/en-us/library/cc721652.aspx

   d. Best Practices In Managing World Wide Web Server Security
   http://www.boran.ch/security/webserver_practices.html

   e. Microsoft IIS7 Security Best Practices
   http://searchsecurity.techtarget.com/tip/0,289483,sid14_gci1521981,00.html?track=NL-422&ad=792838USCA&asrc=EM_NLT_12703904&uid=7993407

G. Define and limit access by web services.

1. **Details:** Allow the web server service only read access for all web content directories and disable HTTP directory browsing. Remove all default page designators not being used. Configure each DotNET application to run in its own distinct application pool.

2. **Description:** This is a Phase 3 finding because allowing executable permissions to general web content directories can allow a malicious visitor to remotely run executable code against the web server using only the web service itself.

   HTTP directory browsing can allow a malicious visitor to see a listing of all files within a web site directory.

   Constraining all DotNET applications to their own individual application pool allows the server administrator to set limits to system resource usage by individual applications.

   This allows for far more granular control of overall system resources and allows system administrators to more easily determine which applications may be causing system level delays or other problems.

   Require all third-party vendors to provide standalone DLLs for web applications.

3. **Solution:** Disabling directory browsing, removing unused default page designators, and configuring DotNET applications to run in their own distinct application pools is specific to each web site. Limiting the web server service read only access for all web content directories may be a system level process on some servers.

4. **References:**

   a. NIST Special Publication 800-44
   http://csrc.nist.gov/publications/nistpubs/800-44-ver2/SP800-44v2.pdf

   b. IIS 6 Best Practices
   http://technet.microsoft.com/en-us/library/cc782762(WS.10).aspx

   c. IIS 7 Best Practices
   http://technet.microsoft.com/en-us/library/cc721652.aspx

   d. Best Practices In Managing World Wide Web Server Security
   http://www.boran.ch/security/webserver_practices.html

e. Microsoft IIS7 Security Best Practices
http://searchsecurity.techtarget.com/tip/0,289483,sid14_gci1521981,00.html?track=NL-422&ad=792838USCA&asrc=EM_NLT_12703904&uid=7993407

f. Microsoft DotNET Application Pool Best Practices
http://learn.iis.net/page.aspx/624/application-pool-identities/

H. Keep log files outside of default installation locations.

1. **Details:** If possible to do so without adversely affecting system operations, define a single hard drive or logical partition, separate from the Operating System, to use for log files, including system event, web site, and other software logs.

2. **Description:** This is a Phase 3 finding because some applications and services must have write permissions to create files, especially entries in log files. Most of this type of access is specific to machine level paths (such as c:\logs, l:\logfiles, and /var/log/weblogs) which are unusable by most web server services.

   Log files also tend to grow in disk space usage extremely fast and can cause a server to run out of drive space on whichever physical drive or drive partition in which they are being kept. If this partition or drive is the same as the operating system, this could cause the server to lock and/or suffer a serious system fault.

3. **Solution:** Create a separate drive or drive partition and reset the service-level settings to write logs to that drive or drive partition.

4. **References:**

   a. NIST Special Publication 800-44
   http://csrc.nist.gov/publications/nistpubs/800-44-ver2/SP800-44v2.pdf

   b. IIS 6 Best Practices
   http://technet.microsoft.com/en-us/library/cc782762(WS.10).aspx

   c. IIS 7 Best Practices
   http://technet.microsoft.com/en-us/library/cc721652.aspx

   d. Best Practices In Managing World Wide Web Server Security
   http://www.boran.ch/security/webserver_practices.html

   e. Microsoft IIS7 Security Best Practices
   http://searchsecurity.techtarget.com/tip/0,289483,sid14_gci1521981,00.html?track=NL-422&ad=792838USCA&asrc=EM_NLT_12703904&uid=7993407

I. Keep temporary files outside of default installation locations.

1. **Details:** If possible to do so without adversely affecting system operations, define a single hard drive or logical partition, separate from the Operating System, web content, and log files, as a system “temp” or “tmp” location for all applications to use for creating any necessary temporary files.

2. **Description:** This is a Phase 3 finding because numerous software programs and/or web applications, in the course of their natural processing, use disk space to write files or snippets of files while they are building.

   Although these temporary files should be deleted when a program is finished processing whatever required it to use temporary drive space, this function is not always programmatically done.

   As with log files, these temporary files can quickly take up excessive physical drive space which could cause a system to become unstable. If these temporary files are being written to the same partition as the server operating system uses, this could cause the entire server to fail.

3. **Solution:** Set your operating system temporary file usage area (such as c:\tmp, c:\temp, or /tmp) to a non-default location or a separate drive partition.

4. **References:**

   a. NIST Special Publication 800-44
   http://csrc.nist.gov/publications/nistpubs/800-44-ver2/SP800-44v2.pdf

   b. IIS 6 Best Practices
   http://technet.microsoft.com/en-us/library/cc782762(WS.10).aspx

   c. IIS 7 Best Practices
   http://technet.microsoft.com/en-us/library/cc721652.aspx

   d. Best Practices In Managing World Wide Web Server Security
   http://www.boran.ch/security/webserver_practices.html

   e. Microsoft IIS7 Security Best Practices
   http://searchsecurity.techtarget.com/tip/0,289483,sid14_gci1521981,00.html?track=NL-422&ad=792838USCA&asrc=EM_NLT_12703904&uid=7993407

J. Allow only Secure FTP connections.

1. **Details:** Allow only secured FTP connections (SFTP or FTPS) from outside systems. Restrict external connections by IP address.

2. **Description:** This is a Phase 3 finding because most web servers need to be accessible remotely via an FTP service. The FTP service allows web developers to write files to the server.

   Under normal operating procedures, basic FTP server services send the user name, password, and transmitted data in open text which is readable by malicious programs.

   Use of a secured FTP connection (such as SFTP or FTPS) encrypts the login and password so that they are unreadable by any but the FTP-specific client and server. It further encrypts all FTP commands passed during a connection session.

   Some, but not all, secure FTP services can also encrypt the actual data as it passes from the FTP client to the FTP server.

   Some firewalls have difficulty allowing these types of encrypted FTP transactions.

3. **Solution:** Use a secure FTP server program capable of encrypting all traffic between a customer's system and the web server (both command and data channels).

4. **References:**

   a. NIST Special Publication 800-44
   http://csrc.nist.gov/publications/nistpubs/800-44-ver2/SP800-44v2.pdf

   b. IIS 6 Best Practices
   http://technet.microsoft.com/en-us/library/cc782762(WS.10).aspx

   c. IIS 7 Best Practices
   http://technet.microsoft.com/en-us/library/cc721652.aspx

   d. Best Practices In Managing World Wide Web Server Security
   http://www.boran.ch/security/webserver_practices.html

   e. Microsoft IIS7 Security Best Practices
   http://searchsecurity.techtarget.com/tip/0,289483,sid14_gci1521981,00.html?track=NL-422&ad=792838USCA&asrc=EM_NLT_12703904&uid=7993407

K. Use HTTPS for all web form submissions.

1. **Details:** Obtain an SSL certificate for all web sites and use HTTPS for all web form submissions that write data to the web server or to a Structured Query Language (SQL) database server. Do not use Basic or Digest authentication.

2. **Description:** This is a Phase 3 finding because, as a normal operation, HTTP traffic is not being encrypted as it traverses a network (either Internet or intranet).

   Use of HTTPS encrypts all traffic from an end user to the web server, including the data being submitted to a web form.

   During a submission from a web server to a SQL server, most web servers are then passing that data to the database server in open text. If possible, this data should also be encrypted between the web server and database server. However, in most cases, access to the SQL server's operational port is protected from outside scrutiny by firewall rules.

3. **Solution:** Obtain an SSL certificate and configure the web server to require and use HTTPS when accessing any web forms, especially when those web forms submit the input data to the database server.

   OCIO has obtained a wild card certificate that can be used for all third-level (and deeper) web site names (such as cio.idaho.gov, apps.cio.idaho.gov, labor.idaho.gov, and project.sto.idaho.gov) at no additional cost to an agency.

4. **References:**

   a. NIST Special Publication 800-44
      http://csrc.nist.gov/publications/nistpubs/800-44-ver2/SP800-44v2.pdf

   b. IIS 6 Best Practices
      http://technet.microsoft.com/en-us/library/cc782762(WS.10).aspx

   c. IIS 7 Best Practices
      http://technet.microsoft.com/en-us/library/cc721652.aspx

   d. Best Practices In Managing World Wide Web Server Security
      http://www.boran.ch/security/webserver_practices.html

   e. Microsoft IIS7 Security Best Practices
      http://searchsecurity.techtarget.com/tip/0,289483,sid14_gci1521981,00.html?track=NL-422&ad=792838USCA&asrc=EM_NLT_12703904&uid=7993407

L. Use robots.txt file.

1. **Details:** Establish, use, and keep current a robots.txt file in the web root of all web sites being hosted on a public-access web server.

2. **Description:** This is a Phase 3 finding because web sites on public-access web servers are constantly being indexed by “web crawlers”. Most of these are from reliable sources (such as Google, Yahoo, MSN, and so forth). However, there are many, many web crawler programs being run against public-access web sites that are attempting to gather information for use by malicious entities.

   Use of the robots.txt file can cause the web server to prevent access by malicious web crawling agents that have been defined. This is useful to prevent known malicious indexing agents from harvesting information that web server administrators and/or web content managers do not wish to make accessible for indexing purposes.

3. **Solution:** Create and place a robots.txt file at the root of all web sites. Keep the file current.

4. **References:**

   a. NIST Special Publication 800-44
   http://csrc.nist.gov/publications/nistpubs/800-44-ver2/SP800-44v2.pdf

   b. IIS 6 Best Practices
   http://technet.microsoft.com/en-us/library/cc782762(WS.10).aspx

   c. IIS 7 Best Practices
   http://technet.microsoft.com/en-us/library/cc721652.aspx

   d. Best Practices In Managing World Wide Web Server Security
   http://www.boran.ch/security/webserver_practices.html

   e. Microsoft IIS7 Security Best Practices
   http://searchsecurity.techtarget.com/tip/0,289483,sid14_gci1521981,00.html?track=NL-422&ad=792838USCA&asrc=EM_NLT_12703904&uid=7993407

   f. Robots.txt Generator
   http://www.mcanerin.com/EN/search-engine/robots-txt.asp

   g. The Web Robots Page
   http://www.robotstxt.org/

V. REFERENCE DOCUMENTS

In addition to this guideline, the following documents apply:

A. *ITA Enterprise Standard S3230 – Server Security Requirements*

B. *ITA Enterprise Guideline G590A – Server Operating System*

C. *ITA Enterprise Guideline G590B – Public-Facing SQL Server Setup*

VI. CONTACT INFORMATION

For more information, contact the ITA Staff at (208) 332-1876.

VII. REVIEW CYCLE

Twelve (12) months

VIII. TIMELINE

Date Established: April 27, 2011
Last Reviewed:
Last Revised:
Last ITRMC Approval: April 27, 2011

IX. REVISION HISTORY

07/01/13 – Changed “ITRMC” to “ITA”.