Global Information Assurance Certification Paper

Auditing Mac OS X Compliance with the Center for Internet Security Benchmark Using Nessus

GSNA Gold Certification

Author: Ricky D. Smith, rdsmith@mac.com

Adviser: Don C. Weber

Accepted: November 7th 2008

Outline

1. Abstract

This paper describes the auditing of a Mac OS X v10.5 system using the compliance checks in Nessus 3.2.x from Tenable Network Security. The creation process for the audit compliance policy file for use with Nessus will be covered. Example compliance checks will be used to demonstrate the process. The basis for the checks performed against the Mac OS X system will be the Mac OS X Benchmark from the Center for Internet Security. A complete listing of the checks implemented will be provided.

2. Auditing Capability of Nessus v3.2

Originally started in 1998 by Renaud Deraison as an open source project, Nessus became one of most well known network vulnerability scanners. (Wikipedia.org, 2008a) In 2002, Renaud Deraison and others founded Tenable Network Security to create a commercial product from Nessus. Tenable continues as the sole sponsor of the open source Nessus project. (Tenable, 2008a) Nessus 2.2.11 is latest currently available the open source version. (Tenable, 2008a)

With the release of Nessus 3 in 2005, Tenable changed Nessus to a closed-source product of Tenable Security. (Wikipedia.org, 2008a) Tenable has two types of subscription licenses for Nessus 3. (Tenable, 2008f) Both provide "real-time Vulnerability Updates" for the plugins. For personal or non-commercial use, they provide free HomeFeed subscriptions with licensing that prohibit its use in a commercial environment or installation on a computer owned by your employer or for your employer benefit. And for professional use, Tenable sells ProfessionalFeed subscription with licensing that allows the use of Nessus by commercial organizations. ProfessionalFeed subscriptions include other benefits such as the capability to do compliance checks. For this

paper, a ProfessionalFeed scanner was used for to provide the compliance checks.

Nessus uses a client-server architecture. A Nessus client is used to configure the network vulnerability scans performed by the Nessus daemon. The client also presents the vulnerability scan results to the user. Tenable provides Nessus client but there are several open source projects that provide the Nessus client functionality. The Nessus daemon, nessusd, is the server part of the architecture. It takes the configuration from the client and performs the requested vulnerability checks against the target computers. (Tenable, 2008b)

The Nessus daemon is the engine to perform the actions that are specified in the Nessus "plugins." Generally, all of the checks to determine if a target machine is susceptible to a specific vulnerability are implemented in one plugin. But not all plugins are checks for vulnerabilities instead they determine information about the target machine that is used by other plugins. The plugins are normally written in the Nessus Attack Scripting Language (NASL), but plugins may be written in other languages and used as compiled binaries. For example, the check for the installation of Mac OX X 10.5.5 update, macosx_10_5_5.nasl is a NASL script but the Unix compliance checks plugin, unix_compliance_check.nbin, is a compiled plugin. (Tenable, 2008c)

Nessus 3 Local Security Checks

Nessus 3 introduced the capability to log in to the target machines and perform local security checks as an addition to being a network vulnerability scanner. The benefits of local checks are the ability to run commands on the system and examine the file system and the configuration of all services of the target machine. With access to the file system, Nessus can

directly check all software packages installed for vulnerable versions not just the operating system and network services. In addition, file system access provides the ability to determine the status of the installation of patches. Also with access to the configuration files, the security stance of the machine can be more completely assessed than with a scan across the network.

Nessus 3 uses secure shell (SSH) to log in to Unix machines to perform the local checks. For local checks on Windows machines, Nessus can use local or domain credentials to connect via network logins. It can be configured to use SSH if a SSH server installed. Nessus can be configured to use either a user name and password pair or a public and private key pair to login via secure shell. For local checks, the account used must have administrative privileges on the machine. For Unix machines, that can be the root account, an account with `sudo` privileges to run all commands as root, or an account that can `su` to root. On Windows machines, the account must be in the local Administrators group either directly or as part of a group.

Nessus 3 Compliance Checks

A ProfessionalFeed license enables the compliance check plugins on the Nessus server. (Tenable, 2008d) Compliance checks allow an auditor to write or modify checks to meet local policy without have to write a complete Nessus plugin. They are essentially an advanced capability of the local security checks. The compliance checks are specified in an audit compliance policy file. The compliance checks plugin interprets the audit compliance policy file and then conducts the vulnerability checks on the target systems via the Nessus daemon. The audit compliance policy files are sometimes called ".audit" files since they have a ".audit" extension.

The audit compliance check items can either be "built-in" items or custom items. For Unix machines, all compliance items can be specified in a single file but may be split into multiple files. However, for Windows audit compliance policy files, the compliance checks are specified in two types of audit policy files: a configuration type audit compliance file or a content type audit compliance file.

The format of an audit compliance file is similar to XML but the only header information required is a <check_type: "type"> tag at the beginning and a corresponding </check_type> tag at the end. The types that can be specified for check_type are "Windows" for configuration compliance policy files, "WindowsFiles" for content compliance policy files, or "Unix" for Unix compliance policy files.

Built-in Compliance Items

Built-in compliance items are specified in an audit compliance file with "<item> tags and end with a "</item>" tag. The compliance items are specified by set of keyword-setting pairs. For example, to specify that a check is applicable to a machine running Mac OS X, the keyword-setting pair used would be:

System: "Darwin"

Note that for Mac OS X, the system is specified as "Darwin," which is the result received when running uname on a Mac OS X machine. (Apple, 1998)

Typically, the name and description keywords are the minimum required for built-in items. The built-in items may be may be used without any additional keyword-setting pairs or they may require the value keyword with the local configuration settings specified. For example, the passwd_zero_uid built-in item, which checks for only one account in /etc/passwd with a UID

of 0, does not require an additional keyword-setting pair and would be written in a .audit file as:

Table 1
Example Nessus Built-in Item

```
<item>
   name: "passwd_zero_uid"
   description : "Check zero UID account in /etc/passwd"
</item>
```

An example where a range of values is required is the **min_password_length** built-in item that checks for a minimum password length of 8 characters. In this built-in item, the **value** keyword-setting pair is required to change the default minimum length from six to eight characters. The built-in check would be specified with the **value** key-setting pair as shown in Table 2.

Table 2
Example Nessus Built-in Item Requiring a Value

```
<item>
   name: "min_password_length"
   description: "Make sure that each password has a
   minimum length of 8 chars or more"
   value: "8..Max"
</item>
```

For all compliance items, the severity of a failed check will be "HIGH" unless the **severity** keyword is set to specify the severity of the failure as "MEDIUM" or "LOW."

Some of the types of built-in local security checks that Nessus 3 can perform on Unix machines are:

- Password Policy and File Management Checks – various checks to verify the password policy on the machine and the proper permissions and password file formats, user and groups ids;

- Permission Management Checks – various checks to verify the permissions on user home directories and various other files on the machine; and

- Suspicious File Checks – various checks to look for SUID or SGID permissions set on files, world-writable directories and other suspicious file on the machine.

The built-in checks are designed to function correctly on most Unix variants. However, several are not available on Mac OS X due to the unique configuration mechanisms used by Apple for some settings and services in Mac OS X.

Custom Compliance Items

A custom item starts with a <custom_item> tag and ends with a corresponding </custom_item> tag. Inside each compliance item, the type of item must be specified and the values required for that type of check. For example, a custom check to verify the ownership and permission of the /var/tmp directory would be specified as

Table 3
Example Nessus Custom Compliance Item

```
<custom_item>
   System: “Darwin”
   type: FILE_CHECK
   description: "Permission and ownership check /var/tmp"
   file: "/var/tmp"
   owner: "root"
   mode: "1777"
</custom_item>
```

A more complete explanation for the compliance item keywords will be given in the next sections.

Some of the types of custom local security checks that can be specified are:

- FILE_CHECK – a check of the existence of a file and the ownership or the file permissions of the specified file

- FILE_CHECK_NOT – a check of the existence of the specified file and the ownership and the file permissions are not those specified in the check

- FILE_CONTENT_CHECK – a check of the existence of a file and the content of the file for specific content using a regular expression

- FILE_CONTENT_CHECK_NOT – a check of the existence of a file and the content of the file for specific content using regular expressions but the check fails if the content matches the regular expression

- CMD_EXEC – a check that allows the execution a single command line and the examination of the results using regular expressions to find the desired section of the results and the expected results

- GRAMMAR_CHECK – a check of the contents of the file for a loosely defined "grammar" based on one or more regular expressions

- Conditionals – if-then-else logic that uses the output of a compliance check or combination of compliance checks to determine if other compliance checks will be performed.

There are four other types of custom compliance items available for Unix machines but they are not applicable to machines running Mac OS X 10.5 Leopard. Three of them, RPM_CHECK, PROCESS_CHECK, and CHKCONFIG, only apply to Red Hat operating systems or operating systems derived from Red Hat. The other check, XINETD_SVC, is for operating systems that use the extended internet daemon, `xinetd`, for controlling services.

(Wikipedia.org, 2008b) **Xinetd** was replaced by the system-wide and per-user daemon/agent manager daemon, **launchd**, in Mac OS X 10.4. (Apple, 2004c; Wikipedia.org, 2008b)

3. Center for Internet Security Mac OS X 10.5 Leopard Level 1 & 2 Benchmark

The Center for Internet Security (CIS) is a not-for-profit organization that creates a number of security benchmarks. (Center for Internet Security [CIS], 2002) The benchmarks are intended to help organizations reduce the information technology risk by providing them with a "best practice" or minimum acceptable configuration for a particular technology. The benchmarks are consensus documents in which the member organizations, and generally the vendor, agree that the recommended security settings are the minimum required. For Level 1 benchmarks the recommended settings are meant to increase the security of the technology from the default installation without breaking any functionality. Level 2 benchmarks provide an increase in the security over the Level 1 benchmarks but may break functionality. (CIS, 2008a)

The Mac OS X 10.5 Leopard Level 1 and 2 Benchmark document (CIS, 2008b) provides the benchmark items for the Level 1 and Level 2 benchmarks interspersed the appropriate sections. There are seven sections of benchmark elements. Each benchmark elements section has a number of benchmark items. Most benchmark items consist of multiple sub items. There are roughly 120 items to be checked with some of those requiring multiple compliance checks. Some of the benchmark elements cannot be checked using an automated tool since they are operational policy not technical settings.

The seven sections of the benchmark are:

- Section 2.1 Installation Action Items – steps to be taken before and during the operating system installation;
- Section 2.2 Hardware and Core Mac OS X Action Items – steps to be performed for hardware security and basic OS functionality security;
- Section 2.3 Account Configuration Items – steps for setting password policy, account management, account directory settings and other account security items;
- Section 2.4 Securing System Software Action Items – steps for setting system preferences related to security;
- Section 2.5 Data Maintenance and Encryption Action Items – steps for providing confidentiality, integrity and availability of user data;
- Section 2.6 Network Services Configuration Action Items – steps to turn off unnecessary services and monitoring for activation of other services; and
- Section 2.7 System Integrity Validation Action Items – steps to increase the log retention for certain system logs.

Some of the benchmark elements cannot be checked using an automated tool since they are operational policy not technical settings.

4. Writing the Mac OS X 10.5 Leopard audit compliance file

The first step was to review the checks list in the CIS Benchmark for those items that were not technical in nature, i.e., items that happen before or during installation or items

that are time based. For example, a network-based scanner cannot check the recommendation to "use the Password Assistant to help generate complex password" or the recommendation to wipe the hard drive before installing the operating system. Those items were categorized as "Policy Items." In the final compliance audit files, the CIS Benchmark items designated as Policy Items were listed in the comments for completeness.

The next step was to review the Scoring Status and Audit sections of the remaining benchmark item to determine if an auditing method had been suggested in the benchmark for that item. From that review, the benchmark items were separated into three categories of benchmark items:

- Definite compliance check – items marked with a Scoring Status of "Scorable" and a command line method of auditing the item are in this category. The compliance checks were implemented using the most appropriate type of compliance check.

- Compliance check possible - items marked with a Scoring Status of "Scorable" but the Audit did not contain a command line method of auditing the item. Some testing was conducted to determine if there was a method to check the benchmark item that was not listed in the benchmark. If no method was found, these items were marked as manual checks. The items in the Manual checks category are listed in the comments of the final compliance audit files.

- No compliance check possible – items marked with a Scoring Status as "Not Scorable" and the Audit as "None." Limited testing was conducted to determine if there was a method to check the benchmark item that was not listed in the benchmark. Most of these were categorized as manual checks. However, for a few of these items, a method to conduct the compliance check was found and implemented.

Implementing Compliance Checks

General Notes

This section will use example Nessus compliance checks to show how to implement a representative CIS Benchmark audit check. The examples will briefly discuss the Benchmark audit check and then show how the Nessus compliance item was implemented. The final Nessus compliance item will be listed. The compliance item will be used in a Nessus vulnerability scan against two targets. Screen captures of the results in the Nessus Client will be presented to show both passing and failing results.

The Nessus compliance items all require a description keyword-setting pair. For this paper, the description keyword setting will be the title of the CIC Benchmark item. In cases where there are multiple checks that comprise the Benchmark item, a short description of the check will be inserted at the end of the title.

In compliance checks, the use of the `info` keywords is optional. However, in the compliance items that were implemented, they are used to provide a reference to the appropriate section of the Benchmark. Additional information about some compliance items is also provided to aid understanding of the results. Compliance items that should be customized for the local policy also have additional `info` keywords to alert the auditor. Additional information is also provided for some built-in compliance items. In these cases, the item fails because the default configuration Mac OS X 10.5 Leopard does not satisfy the check but the CIS Benchmark does not require changing the default setting.

As the number of compliance items increased as they were implemented, the size of the audit policy file became unwieldy.

After sorting the compliance items by section, the audit compliance policy file was split into four parts based on the CIS Benchmark sections. As further checks were written, the compliance items were placed in the appropriate part of the overall audit compliance policy file section. In Appendix A, the four parts of the complete Nessus audit compliance policy is listed.

CMD_EXEC Compliance check example

The CMD_EXEC permits a compliance item be based on a single command line. This type of compliance item is used where the configurations settings are not set by a single line in a text based configuration file or are set in a binary property list (plist) file. Table 4 shows a listing of an example use of this type of compliance item. The compliance item is a check that the correct warning banner set for the GUI login window. The text for the warning banner is set as the value of the LoginwindowText key in the com.apple.loginwindow.plist file. This is a binary plist file that is manipulated with the defaults command. To examine the value of a key, the command "`read`", and the plist name and the key name are passed to `defaults`. (Apple, 2003) In this case, the command would be:

```
$ defaults read /Library/preferences/com.apple.loginwindow \
  LoginwindowTxt
```

This would print out the warning banner configured. The test banner used for the development of the audit compliance file was:

```
        !!Authorized Uses ONLY!!
All activity may be monitored and reported.
```

This test value causes a problem when specifying the expect keyword. The setting for the `expect` keyword is basically the regular expression for searching of the output of the command,

similar to passing the output through **grep**. (Apple, 2002a) The issue arises because the "grepping" only examines one line of text and the output of the defaults command has multiple lines. To solve the issue, the output of the defaults command must be piped into another command create a single line. The solution implemented is the PERL one-liner:

```
perl -0777 -nle 's/\n//g;s/\r//g;print;'
```

which takes the input from the pipe and removes all line feed and carriage return characters, "\n" and "\r" respectively, and prints the rest of the input to the standard output. (Mates, 2008) The compliance check then examines that output.

For CMD_EXEC compliance items, the **type**, **description** and **cmd** keyword-setting pairs must be included followed by a **expect** keyword-setting pair as the content check. Figures 1 and 2 show the compliance item succeeding and failing.

Table 4
Example CMD_EXEC Custom Item

```
<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Create an access warning for the login
   window "
   info: "Section 2.2.2, CIS Mac OS X 10.5 Leopard Level 1
   & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 2"
   info: "Context: System"
   info: " "
   info: "NOTE: This check must be configured for site
   specific warning banner text."
   info: " "
   cmd: "defaults read
   /Library/Preferences/com.apple.loginwindow
   LoginwindowText | perl -0777 -nle
   's/\\n//g;s/\\r//g;print;'"
   expect: ".*?Authorized Uses ONLY.*?All activity may be
   monitored and reported.*"
</custom_item>
```

Note for the **cmd** keyword-setting pair, all double quotes and backslashes in the regular expression must be escaped by a preceding backslash. In the example above, the backslashes for the linefeed and carriage return characters are escaped properly.

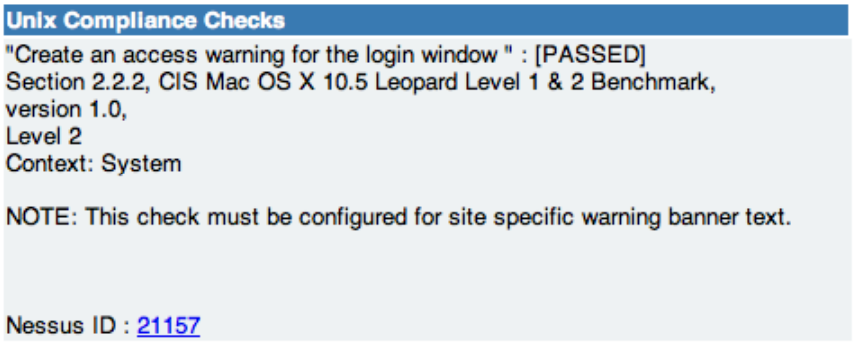

Figure 1. Results from the CMD_EXEC item with the correct access warning configured

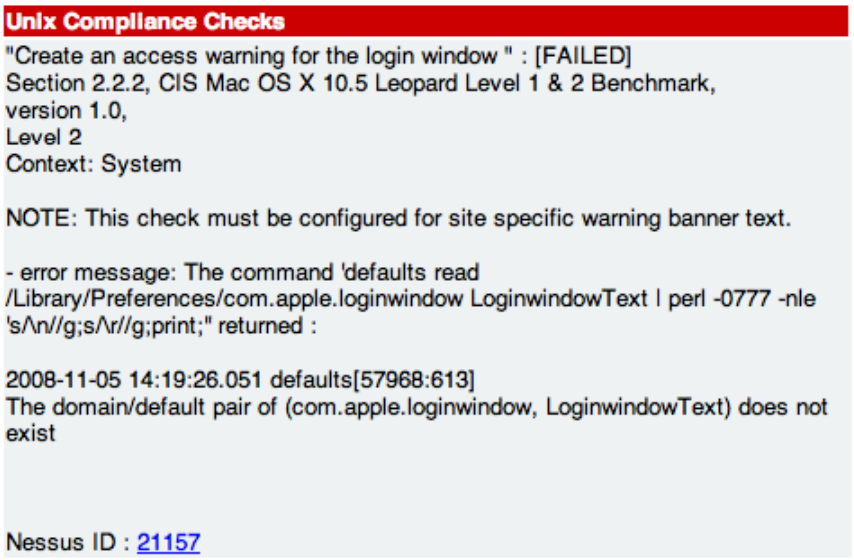

Figure 2. Results from the CMD_EXEC item without an access warning configured

Another example of a CMD_EXEC compliance item looking at the status of Bluetooth is listed in Table 5. The check recommended by the Benchmark produces no output if Bluetooth is disabled. The **expect** keyword regular expression search does not handle this well. A Perl one-liner is used again to provide output easily used by the expect keyword regular expression The one-liner concatenates it's input with a standard text that can easily be checked by the **expect** keyword check. The "**if**" statement generates the output that indicates the state of Bluetooth on the machine. Note the double quotes in this Perl one-liner are escaped with backslashes.

Table 5
Second Example CMD_EXEC Custom Item

```
<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Disable Bluetooth "
   info: "Section 2.2.4, CIS Mac OS X 10.5 Leopard Level 1
            & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 2"
   info: "Context: System"
   cmd: "system_profiler SPBluetoothDataType | \
```

```
      perl -0777 -ne '$x = \"No Bluetooth\" . $_; \
      if ($_ ne \"\") {print \"Bluetooth available\n\";}
      else {print $x . \"\n\";};'"
   expect: "No Bluetooth"
</custom_item>
```

Figure 3. Results from the Second CMD_EXEC Item with Bluetooth Disabled

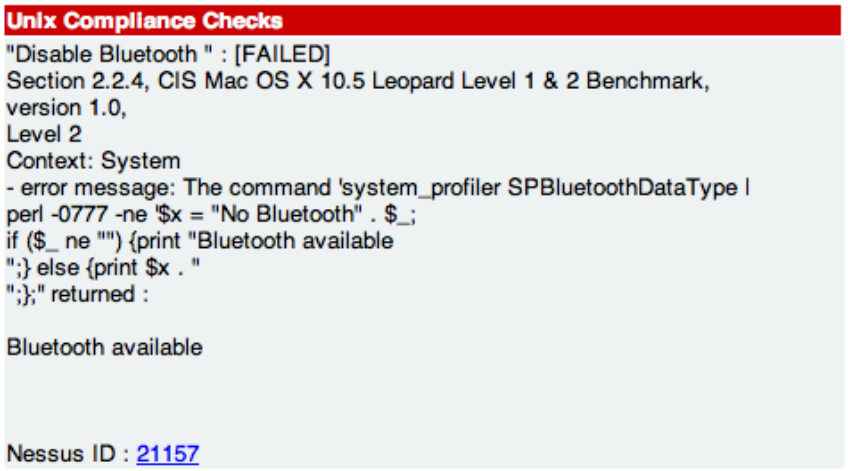

Figure 4. Results from the Second CMD_EXEC Item with Bluetooth Enabled

FILE_CHECK Compliance check example

The CIS Benchmark compliance item used for this example checks that the permissions on the users home folders. The permissions should be set so that no user can view the contents of another users home folder. Mac OS X by default sets the permissions to so others may view the top home directory contents. The default permissions are 0755, in absolute mode, which is the same as "drwxr-xr-x" in symbolic mode. (Apple,

2004a) For the Benchmark, we need to verify that the home directories have 0700 permissions set.

To do this a FILE_CHECK compliance item could be used. For a FILE_CHECK item, the **type**, **description** and **file** keyword-setting pairs must be included followed by at least one check. The checks can be for ownership, group ownership, or file permissions. The file is specified as "**/Users/***", using the normal location of user home directories and a glob (Apple, 2004b) for the usernames. The **file_type** keyword setting of "**d**" restricts the compliance check to directories in the /Users directory.

Table 6
Example FILE_EXEC Custom Item

```
<custom_item>
   system: "Darwin"
   type: FILE_CHECK
   description: "Secure Home Folders "
   info: "Section 2.5.2, CIS Mac OS X 10.5 Leopard Level 1
            & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 2"
   info: "Context: User"
   file_type: "d"
   file: "/Users/*"
   mode: "0700"
</custom_item>
```

This compliance item is in the audit compliance file as written but it is commented out. The file or directory globbling, i.e., the expansion of "**/Users/***", does not appear to function properly which prevents checking of the permissions on the home directories.

Another option for this benchmark item is the built-in **accounts_bad_home_permissions** item. This built-in item checks the permissions of the home directories specified in /etc/passwd. It checks the permissions are at least as stringent as mode 0755, the default value of the check, or those specified by the **mode**

keyword-setting pair. For the benchmark item, the mode keyword setting is 0700. As shown in Figures 5 and 6, this check identifies two default installation home directories, /var/empty and /var/root, as not meeting the benchmark item. The severity of this built-in check was reduced to MEDIUM by using the severity keyword-setting pair since the check will fail even if the machine is properly configured according to the Benchmark recommendations.

Table 7
Example Built-in Item for Users' Home Directory Permissions Benchmark Item

```
<item>
   name: "accounts_bad_home_permissions"
   description : "Account with bad home permissions"
   mode: "0700"
   info: "The normal result is: "
   info: "/var/empty mode: 0755 (should be 0700) owner:
      root"
   info: "/var/root mode: 0750 (should be 0700) owner:
      root"
   info: ""
   info: ""
   severity: MEDIUM
</item>
```

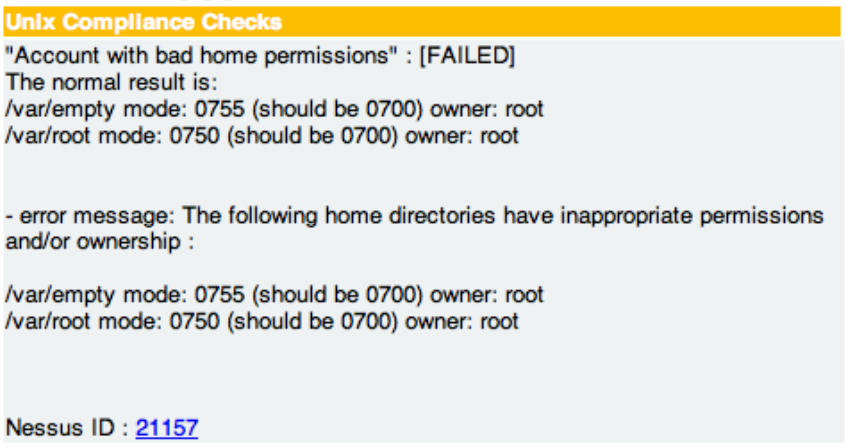

Figure 5. Built-in Item Results with the CIS Benchmark Machine Configuration

The actual check used in the audit compliance file is a CMD_EXEC item that is listed in Table 8. This check is a series

of commands piped together. Table 9 lists the commands used in this check with an explanation of the command.

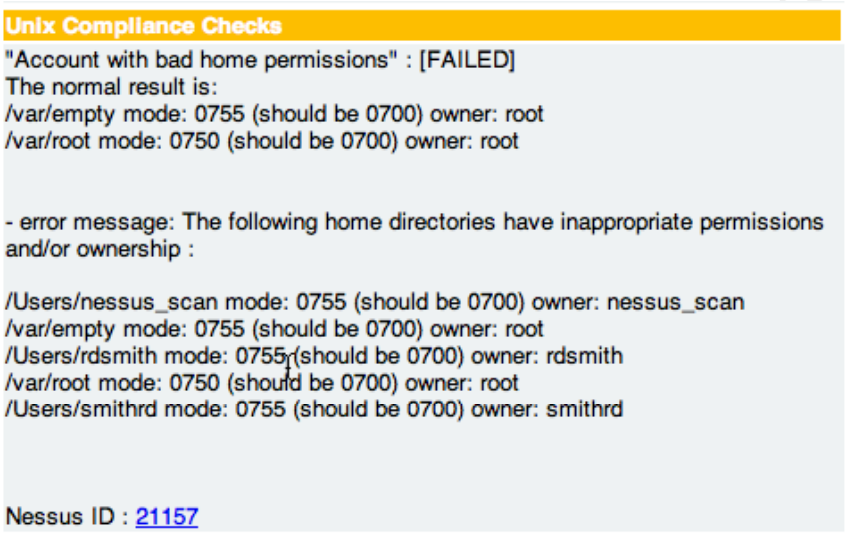

Figure 6. Built-in Item Results without the CIS Benchmark Machine Configuration

Table 8
CMD_EXEC Custom Item for Users' Home Directory Permissions Benchmark Item

```
<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Secure Home Folders "
   info: "Section 2.5.2, CIS Mac OS X 10.5 Leopard Level 1
      & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 2"
   info: "Context: System"
   info: " "
   info: "NOTE: This check assumes that all home
      directories are in /Users."
   info: " "
   cmd: "ls -l /Users | perl -nle 'print \"check \"; if
      (/^d/) {if (!/^drwx------/) {s/.*?\\s([a-zA-
      z]+$)/\\1/; if (!(/Shared/||/total/)) {print
      \"failed: \".$_.\" ;\"}};};' | perl -ne
      's/\\r//g;s/\\n//g;print'"
   expect: "^(check )+$"
</custom_item>
```

Table 9
Explanation of commands used in the CMD_EXEC Custom Item for Users' Home Directory Permissions Benchmark Item

`ls –l /Users` Gets the list of home directories in long format. (Apple, 2002b) This assumes all home directories are in /Users. Add any other directories where user home directories are located.

`perl -nle 'print \"check \"; \`
`if (/^d/) {if (!/^drwx------/) {s/.*?\\s([a-zA-z]+$)/\\1/;`
`if (!(/Shared/||/total/)) {print \"failed: \".$_.\" ;\"}};};'`
 	
Takes the output of the `ls` command one line at a time and produces the correct output for each directory. "check " is printed for each input line. If the line starts with "d", indicating it is a directory, then the symbolic permissions are checked. If the symbolic permissions are not "drwx------", the equivalent of mode 0700, then the directory name is extracted from the input line and printed out with "failed: " before it. The `(!(/Shared/||/total/))` check prevents a failed check caused by the Shared directory, which isn't required to have the mode 700 requirements, and the `total` line from the `ls` output.

`perl -ne 's/\\r//g;s/\\n//g;print'` Removes carriage return and line feed characters so that there is only one line of output.

The results of the test vulnerability scans are shown in Figures 7 and 8. Figure 8 shows an example of the command output with the home directories for three users, nessus_scan, rdsmith and smithrd, having incorrect permissions set.

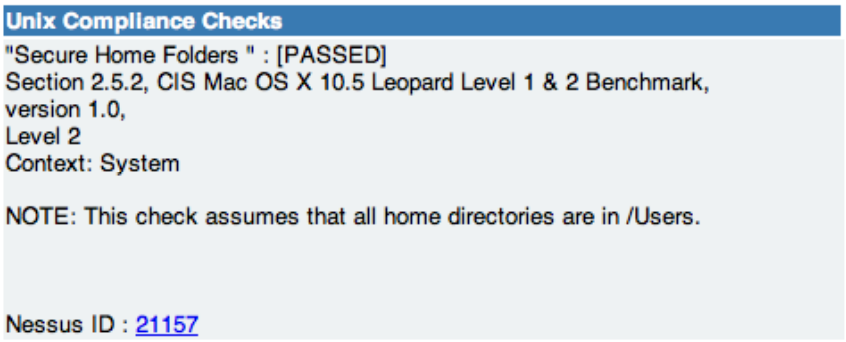

Figure 7. Example CMD_EXEC Item Results with the correct home directory permissions

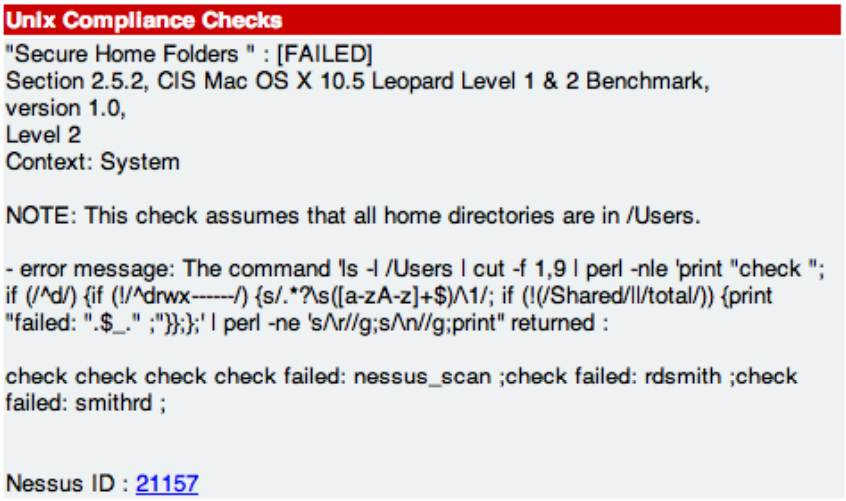

Figure 8. Example CMD_EXEC Item Results with the home directory permissions set incorrectly

FILE_CHECK_NOT Compliance check example

This compliance item will be used to check that X11 application is not installed on the Mac OS X machine. The X11.app is installed by default during operating system installation but should not be installed unless needed by the user. When during the software installation process, a receipt is created in /Library/Receipts to indicate that the software has been installed. For most Apple software, the receipts are stored as binary data in a bom file, where stands for "bill of

material." (Apple, 2006) The bom files are located in the "boms" directory under /Library/Receipts.

For this benchmark item, a FILE_CHECK_NOT compliance item will be used. Like the FILE_CHECK compliance item, the **type**, **description** and **file** keywords must be included followed by at least one check. The checks can be for ownership, group ownership, or file permissions. The file is specified as "**/Library/Receipts/boms/com.apple.pkg.X11User.bom**". The check is to see if the owner is "_installer", the default name used by the operating system installer. If the file exists and the check matches the specified value, the compliance item fails.

Table 10
Example FILE_CHECK_NOT Custom Item

```
<custom_item>
   system: "Darwin"
   type: FILE_CHECK_NOT
   description: "Do not install any unnecessary packages "
   info: "Section 2.1.4, CIS Mac OS X 10.5 Leopard Level 1
      & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 1"
   info: "Context: System"
   file:
      "/Library/Receipts/boms/com.apple.pkg.X11User.bom"
   owner: "_installer"
</custom_item>
```

This check could also be implemented a CMD_EXEC compliance item similar to compliance item use to check the existence of Bluetooth that was listed in Table 5. The "**system_profiler SPBluetoothDataType**" command would have to be replaced with an "**ls -1 /Library/Receipts/boms/com.apple.pkg.X11User.bom**" command. And, obviously, the text in the Perl one-liner and regular expression **setting of the expect** keyword would need to be replaced with something appropriate.

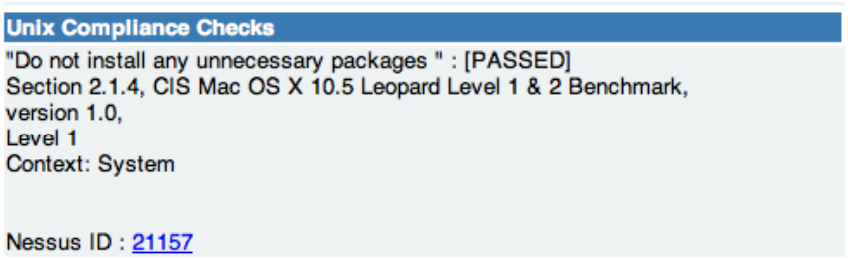

Figure 9. Example FILE_CHECK_NOT Item Results with the X11.app not installed on the machine

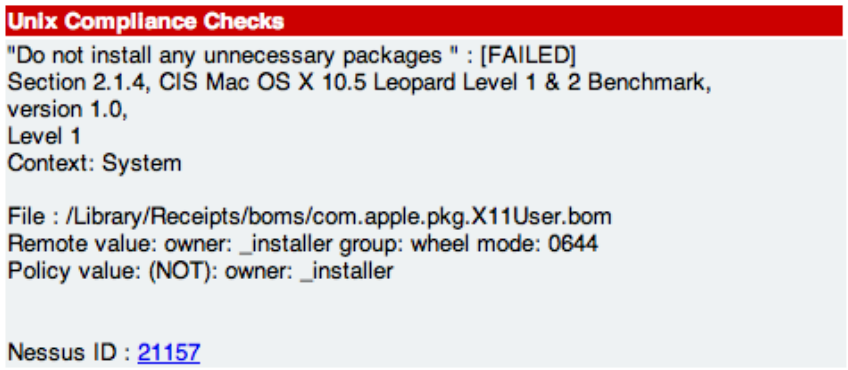

Figure 10. Example FILE_CHECK_NOT Item Results with the X11.app installed on the machine

FILE_CONTENT_CHECK Compliance check example

This compliance item will be used to check that the operating system is configured to synchronize time with an NTP server. This compliance items verifies that the /etc/hostconfig file has the correct entry, "TIMESYNC=-YES-", required to enable the time synchronization to occur. This compliance item partially completes the benchmark item since the benchmark requires that the correct NTP servers must also be configured in /etc/ntp.conf.

A FILE_CONTENT_CHECK compliance item will be used. Like the FILE_CHECK compliance item, the **type**, **description** and **file** keywords must be included followed by the content check. The checks can be for ownership, group ownership, or file permissions. Again, the description is set to the title of the

benchmark item. The file is specified as "/Library/Receipts/boms/com.apple.pkg.X11User.bom".

Table 11
Example FILE_CONTENT_CHECK Custom Item

```
<custom_item>
   system: "Darwin"
   type: FILE_CONTENT_CHECK
   description: "Enter correct time settings "
   info: "Section 2.4.5.1, CIS Mac OS X 10.5 Leopard Level
      1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 1"
   info: "Context: System"
   file: "/etc/hostconfig"
   regex: ".*TIMESYNC=.*$"
   expect: ".*TIMESYNC=-YES-"
</custom_item>
```

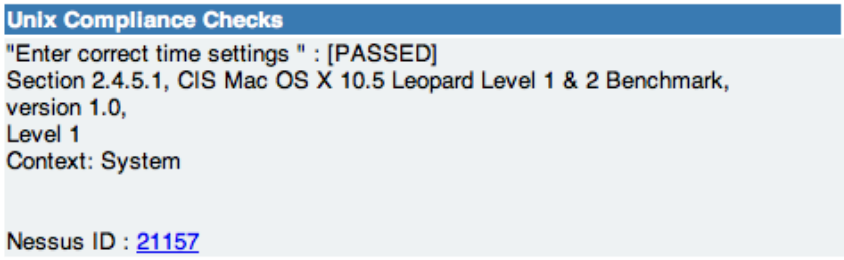

Figure 11. Example FILE_CONTENT_CHECK Item Results with the correct /etc/hostconfig contents

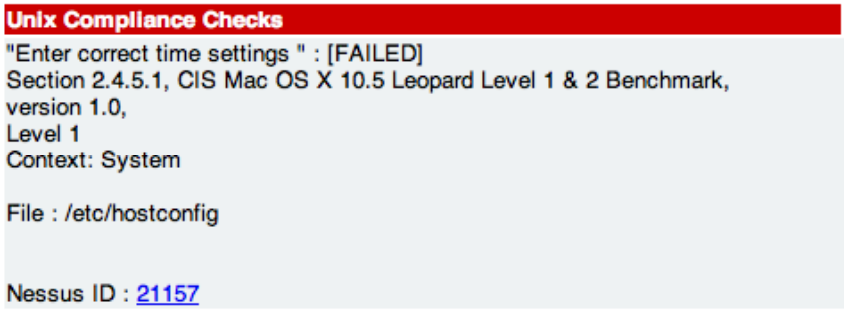

Figure 12. Example FILE_CONTENT_CHECK Item Results with TIMESYNC disabled in /etc/hostconfig

Conditionals example

The conditional example is a check that the .Mac preference pane has been disabled. The conditional compliance check in the "if" portion determines if the .Mac preference pane is owned by root. This check should succeed if the preference pane exists. The results of the conditional compliance check do not appear in the Nessus scan results only the results of the "then" section or "else" section will appear in the Nessus scan results.

Table 12
The "if" section of the example conditional custom item

```
<if>
   <condition type: "or">
      <custom_item>
         system: "Darwin"
         type: FILE_CHECK
         description: "Disable the .Mac preference pane
            from System Preferences (owned by root check)"
         info: "Section 2.4.1.7, CIS Mac OS X 10.5 Leopard
            Level 1 & 2 Benchmark,"
         info: "version 1.0,"
         info: "Level 2"
         info: "Context: System"
         file:
            "/System/Library/PreferencePanes/Mac.prefPane"
         owner: "root"
      </custom_item>
   </condition>
```

If the conditional compliance check succeeds, the FILE_CHECK compliance check in the "then" section, listed in Table 13, determines if the permissions for the preference pane are mode 0700. If permissions are mode 0700, the check passes and the .Mac preference pane exists but is not available to other users, including administrator accounts, which meets the intent of the Benchmark item. If the compliance check in the then section fails, then the .Mac preference pane is potentially usable by users. Only the results of the "then" section compliance check will appear in the Nessus scan results. Figures 13 shows the Nessus vulnerability scan results when the .Mac Preference Pane still exists and has permissions set to prevent its use. When

this check fails, the Nessus scan should produce results similar to those shown in Figure 14.

Table 13
The "then" section of the example conditional custom item

```
<then>
   <custom_item>
      system: "Darwin"
      type: FILE_CHECK
      description: "Disable the .Mac preference pane
         from System Preferences (owned by root and mode
         0700 check)"
      info: "Section 2.4.1.7, CIS Mac OS X 10.5 Leopard
         Level 1 & 2 Benchmark,"
      info: "version 1.0,"
      info: "Level 2"
      info: "Context: System"
      file:
         "/System/Library/PreferencePanes/Mac.prefPane"
      mode: "0700"
   </custom_item>
</then>
```

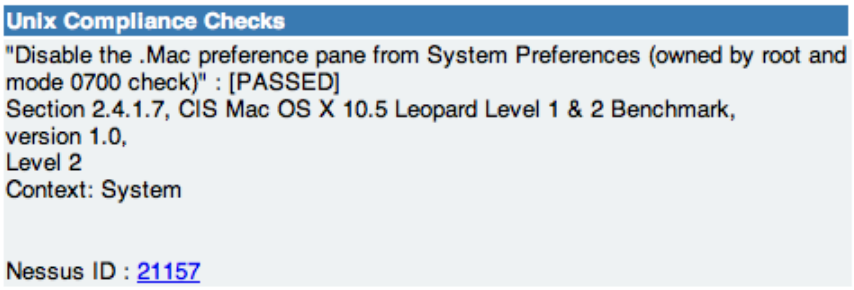

Figure 13. Example Conditional Check Item Results when the .Mac Preference Pane exists but the permissions are set correctly

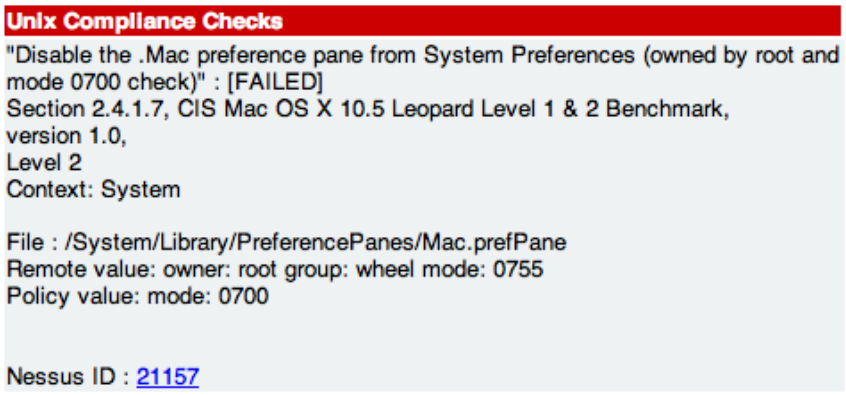

Figure 14. Example Conditional Check Item Results when the .Mac Preference Pane exists and the permissions allow it use

If the conditional compliance check fails, the FILE_CHECK_NOT compliance check in the "else" section will succeed if the group owner is NOT wheel or the file does not exist. The results of this compliance check will appear in the results. The Nessus scan results for this check with a machine is correctly configured are shown in Figure 15. An incorrectly configured machine will generate the results shown in Figure 16.

Table 14
The "else" section of the example conditional custom item

```
<else>
   <custom_item>
      system: "Darwin"
      type: FILE_CHECK_NOT
      description: "Disable the .Mac preference pane
         from System Preferences (check that it doesn't
         exist"
      info: "Section 2.4.1.7, CIS Mac OS X 10.5 Leopard
         Level 1 & 2 Benchmark,"
      info: "version 1.0,"
      info: "Level 2"
      info: "Context: System"
      file:
         "/System/Library/PreferencePanes/Mac.prefPane"
      owner: "root"
      mode: "0700"
      /custom_item>
 </else>
```

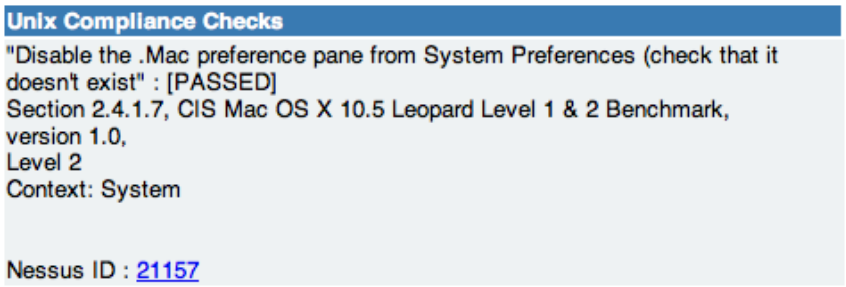

Figure 15. Example Conditional Check Item Results with the .Mac Preference Pane removed

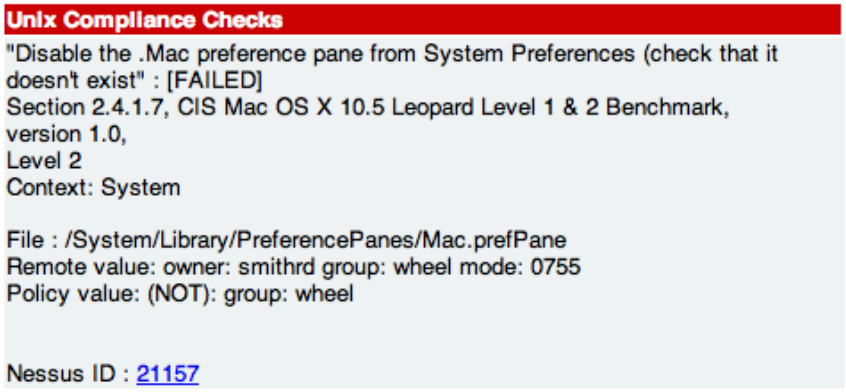

Figure 16. Example Conditional Check Item Results when the .Mac Preference Pane exists

Testing the audit compliance file

Testing Environment

The testing environment for creating and testing the Mac OS X 10.5 Leopard audit compliance file is a simple network with all machines on the same subnet.

- Nessus Server: The Nessus daemon, version 3.2.1, is running on a machine with a fully patched copy of openSUSE 11.0. The Nessus daemon machine has a ProfessionalFeed subscription license installed.

- Nessus Client: The majority of the editing and testing of the audit compliance file will be conducted on a Mac Book

Pro 17" running a fully patched copy of Mac OS X 10.5.5. Nessus was downloaded and installed on this machine but a HomeFeed license was installed since the intent was to only use the Nessus Client to connect the Nessus Server with the ProfessionalFeed subscription license.

- Targets: A Mac Book Pro 15 and a PowerMac G4 867Mhz were the target machines. This set of machines permitted testing both Intel and PowerPC architectures. The PowerMac G4 also is old enough to not have an iSight camera or Bluetooth adapter built-in. An external iSight camera and a USB Bluetooth adapter were available to facilitate testing. Both are running Mac OS 10.5.5 with all patches. Since, the focus of the testing is the audit compliance file the target machines have all patches installed to minimize the extraneous findings in the results of the Nessus scans. Both machines have Remote Login, Apple's name for secure shell, enabled to allow the Nessus server to conduct the local checks and compliance checks. An administrative account named "Nessus Scan Account," with short name of nessus_scan, was created on each machine for the Nessus local checks.

Two targets were available for used for testing to allow testing of two configurations at the same time: one target would be configured to pass the compliance item; the other configured to fail the compliance item. The "failure" configuration for almost all cases was the default configuration of Mac OS X 10.5.

Nessus Scan Policy Configuration

A scan policy, Mac OS X – CIS Benchmark v1.0, was created for testing the audit compliance policy file. On the Options tab, the scan policy was configured to only run the "Safe checks" and to "Log the details of the scan on the server." Safe Check

disables any plugins that is marked as having a negative impact on the target system. The "Ping the remote host" option configures Nessus to verify the target machine is answering on the network before starting the full scan. The Nessus TCP Scanner and the netstat portscanner (SSH) were selected in "Port scanners to use:" section. The default list of ports is sufficient for testing the audit compliance policy file.

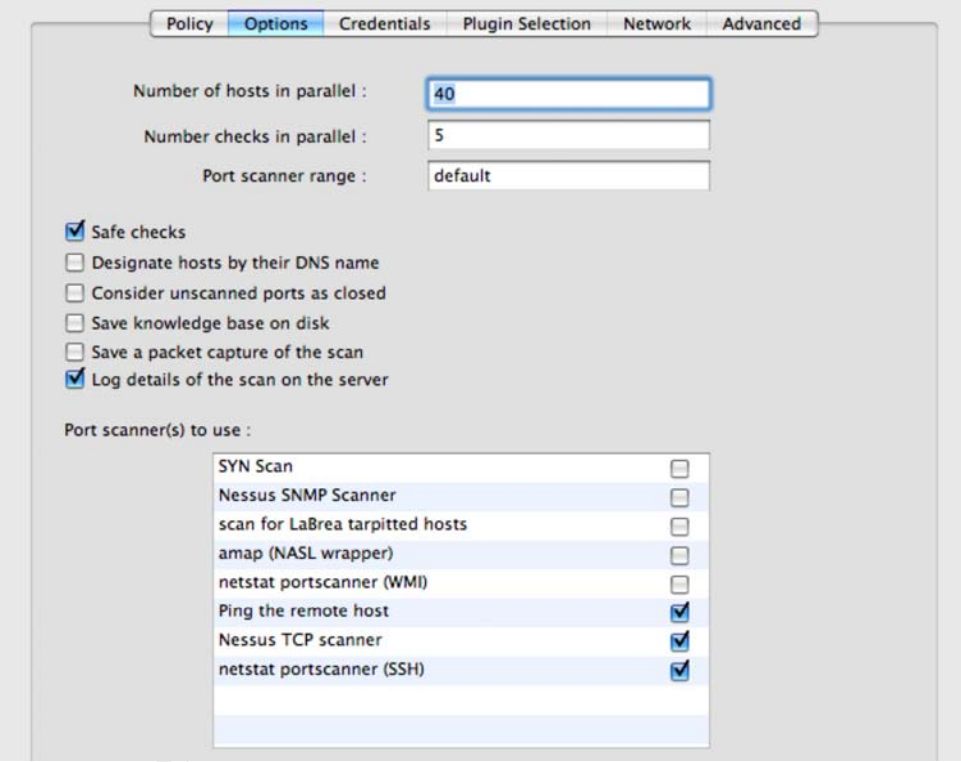

Figure 17. Options tab for the Mac OS X – CIS Benchmark v1.0 Nessus scan policy

Figure 18 shows the correct Remote Login credentials were configured on the Credentials tab on the SSH settings page including the `sudo` password. No credentials were supplied on any of the other pages of the Credentials tab.

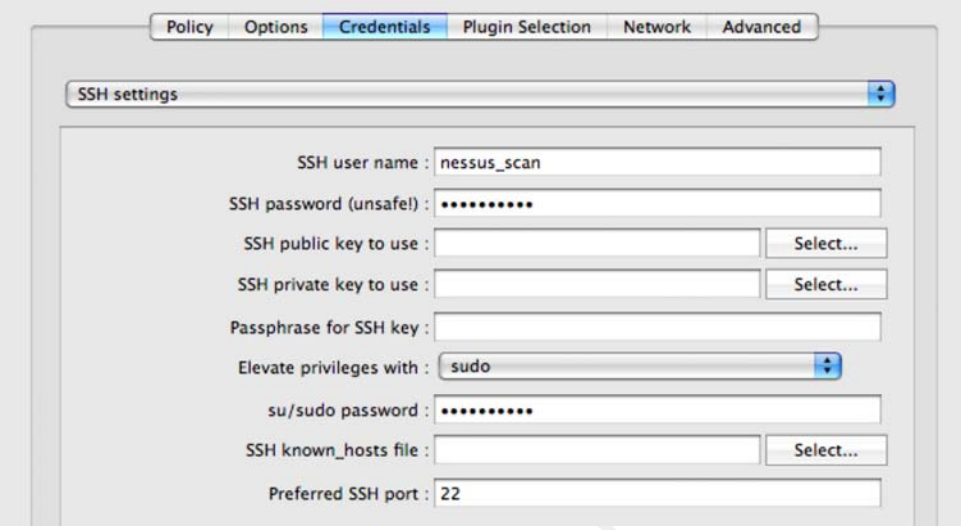

Figure 18. SSH settings page of the Credentials tab for the Mac OS X – CIS Benchmark v1.0 Nessus scan policy

On the Plugin Selection tab, the "Enable All" button was selected to enable all plugins. To limit the number of extraneous entries in the nessusd.messages file on the Nessus server, all Local Security Checks Groups except the Mac OS X Local Security Checks were disabled. In the Policy Compliance group, only the Unix Compliance Checks were enabled.

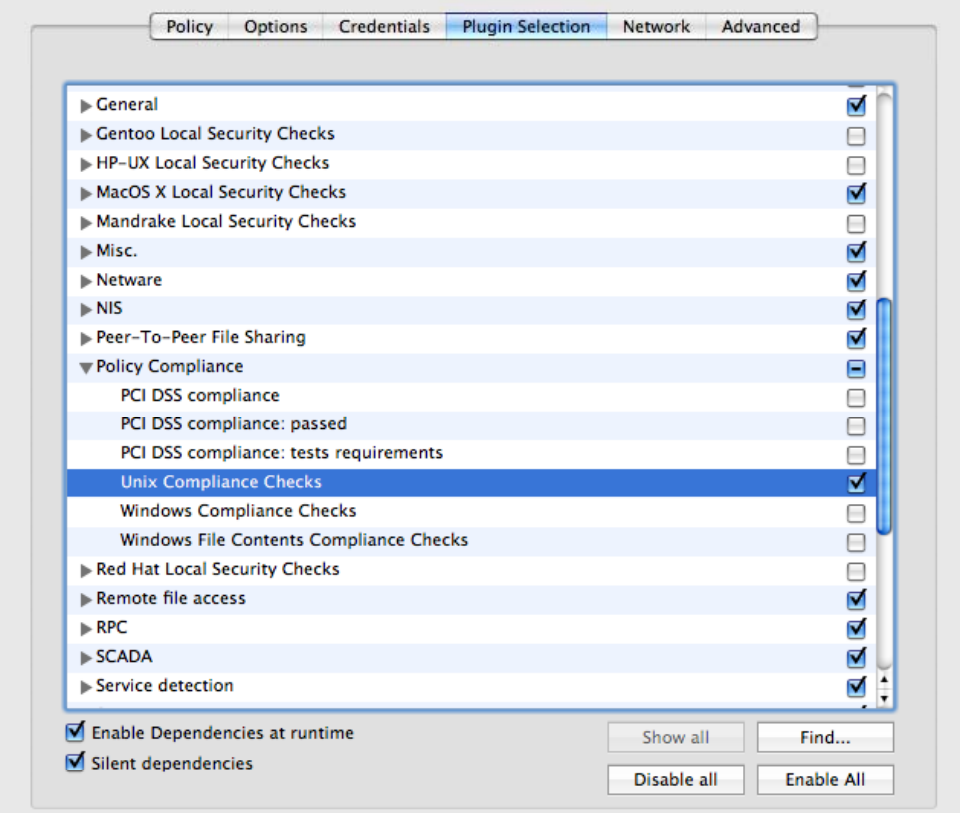

Figure 19. Plugin Selection tab for the Mac OS X - CIS Benchmark v1.0 Nessus scan policy

No changes were made to the settings on the Network tab.

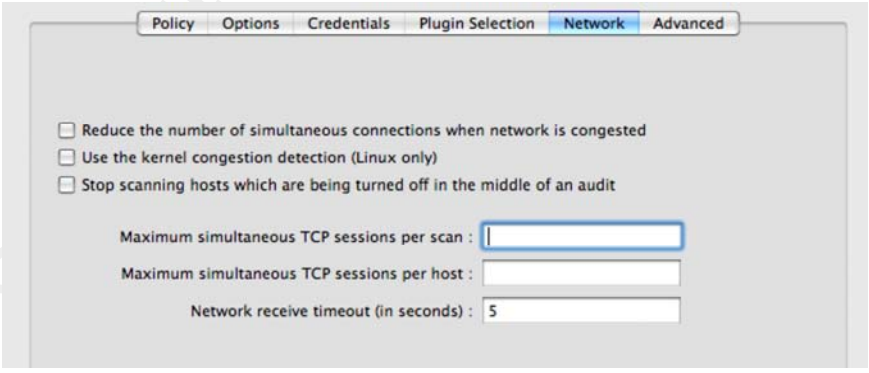

Figure 20. Network tab for the Mac OS X - CIS Benchmark v1.0 Nessus scan policy

On the Advanced tab, the Unix compliance page was selected. The Policy file fields were populated with the four parts of the

Mac OS X CIS Benchmark audit compliance file. No other settings were changed on any of the other pages on the Advance tab.

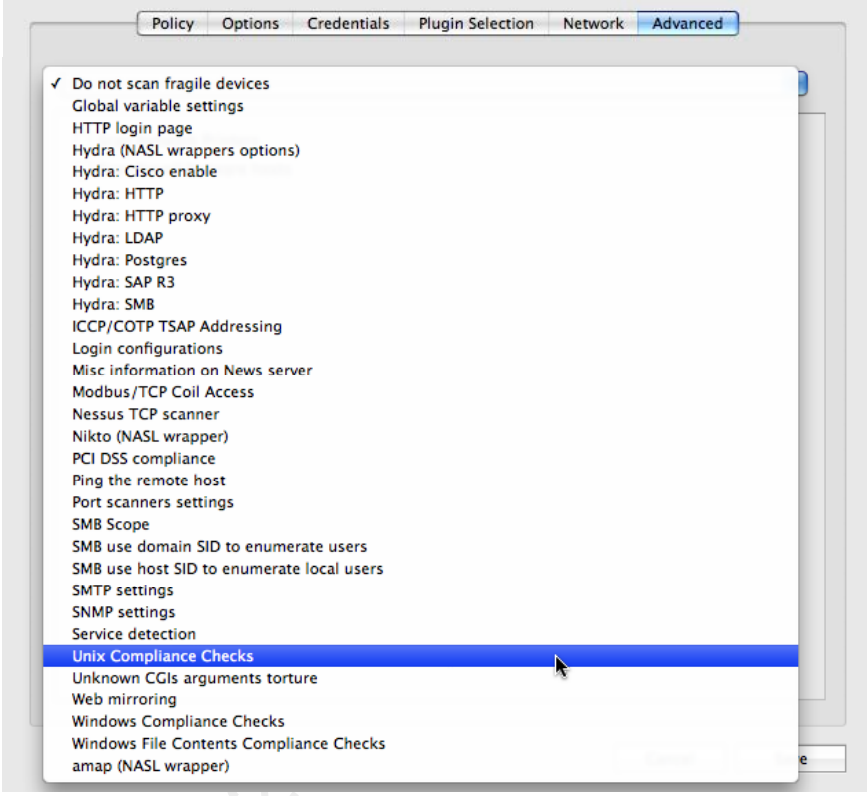

Figure 21. Page selection list on the Advanced tab for the Mac OS X – CIS Benchmark v1.0 Nessus scan policy

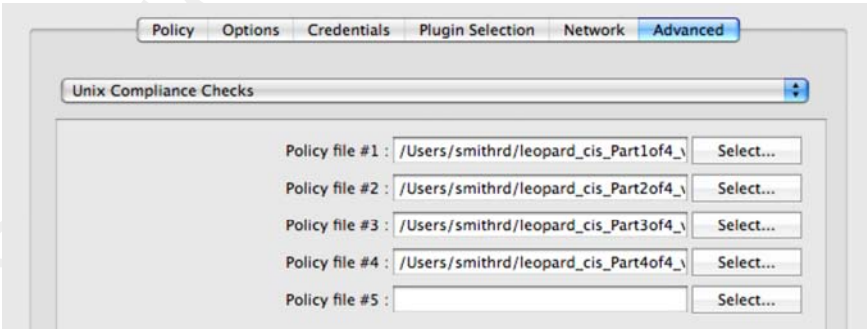

Figure 22. Unix Compliance Checks page on the Advanced tab for the Mac OS X – CIS Benchmark v1.0 Nessus scan policy

Test Methodology

Testing of the audit compliance file was done iteratively. As a category of compliance items was completed, the checks were copied to a test audit compliance policy file. A Nessus scan of the target machines was used to validate that the compliance items were written correctly and could be properly parsed by the Unix compliance check plugin, unix_compliance_check.nbin. Once the compliance item parsed correctly, then the output, PASSED or FAILED, was verified correct for the configuration of each target. If the compliance item output was not correct, the compliance item structure was reviewed and troubleshot. Most compliance check were corrected and then moved to the final audit compliance policy file.

Some CIS Benchmark items could not be checked using the recommended audit procedure. Most of these were configuration file content checks where the recommended configuration settings were spread across multiple lines. The plist file content checks in /System/Library/LaunchDaemons in Section 2.4.14.3 of the CIS Benchmark are excellent examples of this problem. In these cases, another type of Nessus compliance items was written and tested until the output was correct.

Testing Results

The Mac OS X – CIS Benchmark v1.0 scan policy was configure to use the complete set of CIS Benchmark compliance items in the four audit compliance policy files. A Nessus vulnerability scan with the Mac OS X – CIS Benchmark v1.0 scan policy was the last verification of the compliance checks. This scan was run against both targets.

The results of local security checks and the compliance checks appear in results for the "general/tcp" port. Figure 23

shows the Nessus Client with the results of a scan with the final Mac OS X – CIS Benchmark v1.0 scan policy.

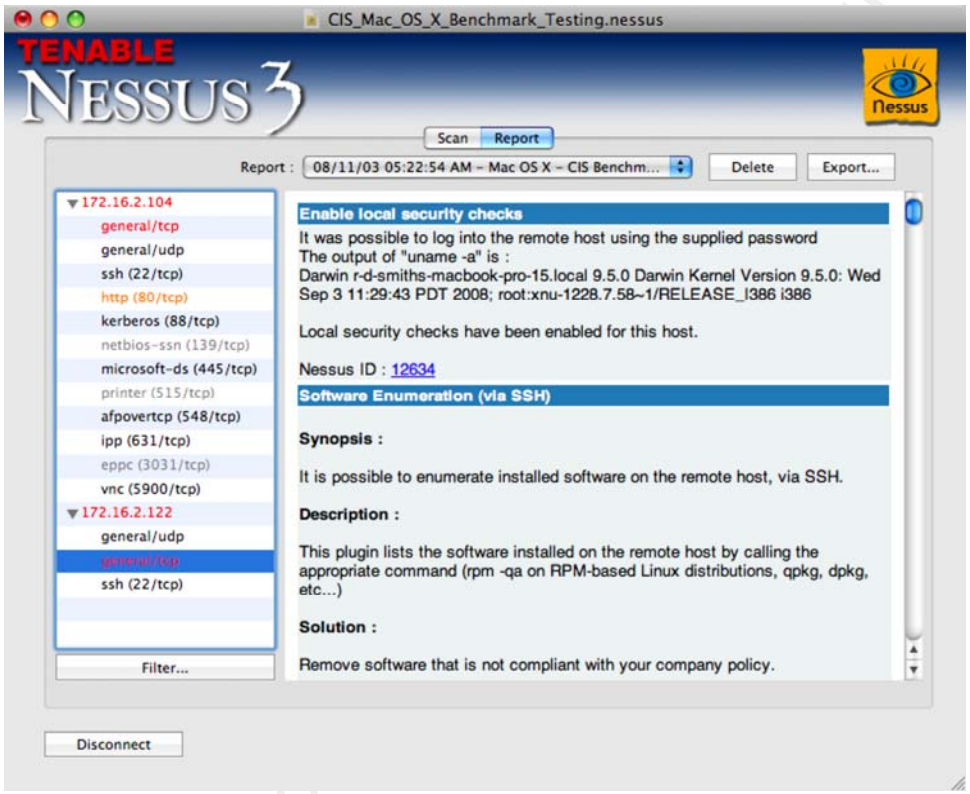

Figure 23. Results from a Nessus scan using the Mac OS X – CIS Benchmark v1.0 policy

Using the filtering capability of the Nessus Client reduces the number of scan results to only those generated by the Unix compliance check plugin. The results from the Unix compliance plugin have a Nessus ID or Plugin ID of 21157. The report filter shown in Figure 24 also filters the results so that only the failed compliance checks are shown in the client as shown in Figure 25.

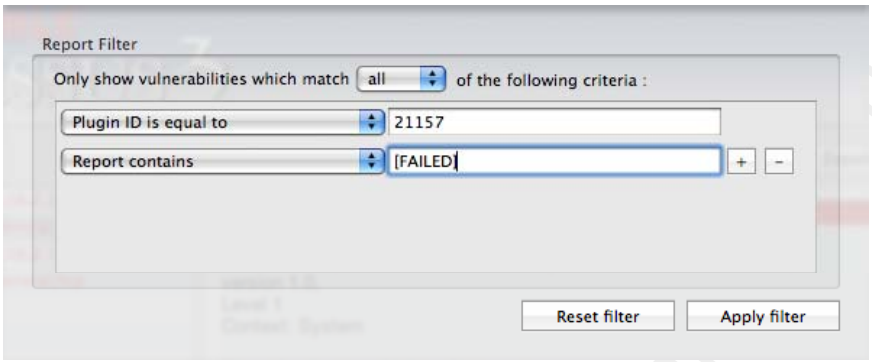

Figure 24. Results from a Nessus scan using the Mac OS X - CIS Benchmark v1.0 policy

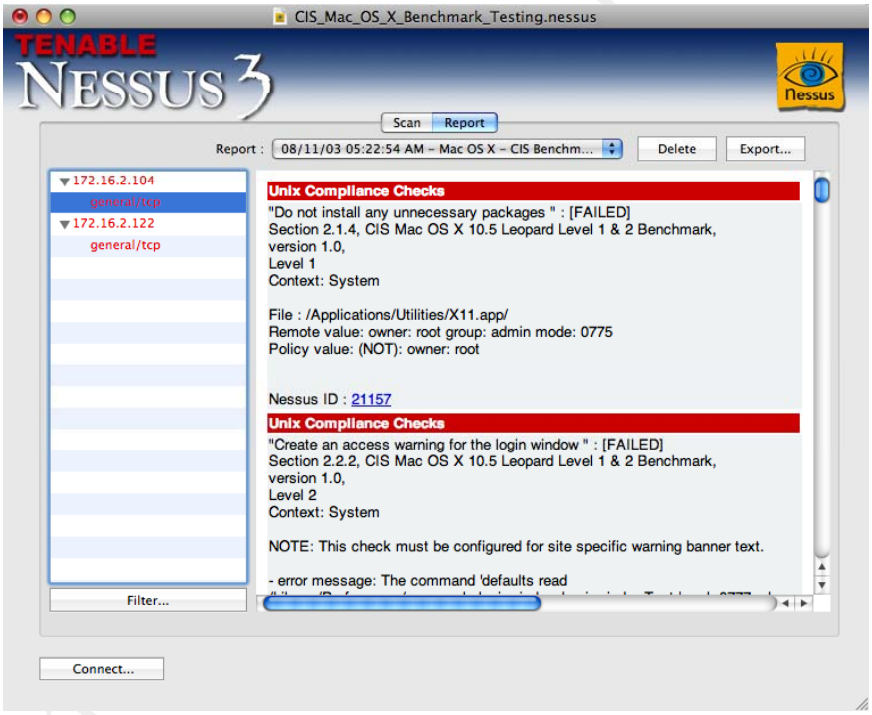

Figure 25. Filtered results from a Nessus scan using the Mac OS X - CIS Benchmark v1.0 policy

5. Conclusions

The compliance check functionality of Nessus 3 provides a useful way to check the configuration of a machine running Mac OS X Leopard against over half of the compliance items in the Center for Internet Security Benchmark. The remaining compliance items are almost evenly split between Policy Items and Manual Checks.

The compliance checks of the CIS Benchmark items in the System context settings are reasonably easy to implement. However, checks in the user context are more difficult implement. The checks for the user context require checking the setting for each user. User context file checks and file content checks have difficulties due to the file permissions. User context checks that require command execution by the user have two problems: the command must be run with the effective uid of that user; and the limitation that a CMD_EXEC command and output are limited to one line each.

When attempting to iteratively check a setting for a number of users, the one line command restriction can be worked around using Perl one-liners. However, the more difficult restriction to overcome is the one line limitation on the command output. It's difficult to create one line of output the `expect` keyword regular expression can match against without a high false positive rate.

The problems with user context checks could potentially be solved with a suid root binary or shell script that conducts the required compliance check and provides a single line output. The suid check would then be used in a CMD_EXEC Nessus audit compliance item to allow the results to be included in the Nessus vulnerability scan results. There are a number issues with this approach including: each check would require a different binary or shell script; every machine would have to have a copy of the binary or script installed; and the security implications of adding a number of suid root binaries or scripts. Most of these user context compliance checks must still be done manually for each user account. Fortunately, there are less than a quarter of the CIS Benchmark compliance items are manual checks.

Tables 15 and 16 summarize the numbers of CIS Benchmark compliance items and the implemented Nessus compliance check items. All of the implemented Nessus compliance checks were

tested successfully using the methodology discussed in the previous section.

Table 15
Number of CIS Benchmark compliance items

CIS Benchmark compliance items		
Total Number of compliance items		118
Number of items at Benchmark Level	Level 1	66
	Level 2	52
Number of items with Scoring Status:	Scorable	51
	Not Scorable	66
	Not designated	1
Number of items in Context:	System	71
	User	41
	System and User	6

Table 16
Numbers of implemented Nessus compliance checks

Nessus Compliance Checks		
Number of implemented checks:		62
Number of Policy Items:		27
Number of Manual Checks:		15
Number of Policy Items/Manual Checks:		9
Number of Custom Checks:	Total	95
	Level 1	47
	Level 2	48
Number of Conditional Checks:		6

6. References

Apple, Inc. (2006, September 28). *Mac OS X Manual Page For bom(5).* Retrieved November 4 2008 from Apple web site: http://developer.apple.com/documentation/Darwin/Reference/ManPages/man5/bom.5.html

Apple, Inc. (2004a, July 8). *Mac OS X Manual Page For chmod(1).* Retrieved November 4 2008 from Apple web site: http://developer.apple.com/documentation/Darwin/Reference/ManPages/man1/chmod.1.html

Apple, Inc. (2003, November 3). *Mac OS X Manual Page For defaults(1).* Retrieved November 4 2008 from Apple web site: http://developer.apple.com/documentation/Darwin/Reference/ManPages/man1/defaults.1.html

Apple, Inc. (2004b, September 1). *Mac OS X Manual Page For glob(3).* Retrieved November 4 2008 from Apple web site: http://developer.apple.com/documentation/Darwin/Reference/ManPages/man3/glob.3.html

Apple, Inc. (2002a, January 22). *Mac OS X Manual Page For grep(1).* Retrieved November 4 2008 from Apple web site: http://developer.apple.com/documentation/Darwin/Reference/ManPages/man1/grep.1.html

Apple, Inc. (2004c, September 30). *Mac OS X Manual Page For launchd(8).* Retrieved November 4 2008 from Apple web site: http://developer.apple.com/documentation/Darwin/Reference/ManPages/man8/launchd.8.html

Apple, Inc. (2002b, May 19). *Mac OS X Manual Page For ls(1).* Retrieved November 4 2008 from Apple web site: http://developer.apple.com/documentation/Darwin/Reference/ManPag

es/man1/ls.1.html

Apple, Inc. (1998, November 9). *Mac OS X Manual Page For uname(1).* Retrieved November 4 2008 from Apple web site: http://developer.apple.com/documentation/Darwin/Reference/ManPages/man1/uname.1.html

Arboi, M. (2005, April 29). *The NASL2 reference manual.* Retrieved November 4, 2008 from Nessus.org web site: http://nessus.org/doc/nasl2_reference.pdf

Center for Internet Security. (2008). *Center for Internet Security – Benchmarks and Tools*. Retrieved November 4, 2008 from Center for Internet Security web site: http://www.cisecurity.org/bench.html

Center for Internet Security. (2008, May). *CIS Level 1 & 2 Benchmark for Mac OS X.* Retrieved November 4, 2008 from Center for Internet Security web site: http://www.cisecurity.org/bench_macosx.html (registration required)

Center for Internet Security. (2002, April 1). *CIS Charter.* Retrieved November 4, 2008 from Center for Internet Security web site: http://www.cisecurity.org/charter.html

Gula, R. (2006, September 7). *Understanding the Nessus "Safe Checks" Option.* Retrieved November 3, 2008 from Tenable Network Security web site: http://blog.tenablesecurity.com/2006/09/understanding_t.html

Mates, J. (2008, October 24). *Perl One Liners.* Retrieved October 31, 2008, from Jeremy Mates's Domain web site: http://sial.org/howto/perl/one-liner/index.xml

Newham, C. & Rosenblatt, B (1998). *Learning the bash Shell, 2nd*

ed. Sebastopol, CA: O'Reilly & Associates

Siever, E., Spainhour, S., & Patwardhan, N. (1999). *Perl in a Nutshell.* Sebastopol, CA: O'Reilly & Associates

Tenable Network Security. (2008). *About Tenable.* Retrieved November 5, 2008, from Tenable Network Security web site: http://nessus.org/about/

Tenable Network Security. (2008a). *Available download information.* Retrieved October 30, 2008, from Tenable Network Security web site: http://nessus.org/download/

Tenable Network Security. (2008b, August 1). *Nessus 3.2 Advanced User Guide.* Retrieved October 30, 2008, from Tenable Network Security web site: http://nessus.org/documentation/nessus_3.2_advanced_user_guide.pdf

Tenable Network Security. (2008c, August 1). *Nessus 3.2 Installation Guide.* Retrieved October 30, 2008, from Tenable Network Security web site: http://www.tenablesecurity.com/documentation/nessus_3.2_installation_guide.pdf

Tenable Network Security. (2008d, August 1). *Nessus Compliance Checks – Auditing UNIX and Windows Device Configurations.* Retrieved October 30, 2008, from Tenable Network Security web site: https://plugins-customers.nessus.org/support-center/nessus_compliance_checks.pdf

Tenable Network Security. (2008e). *Nessus Frequently Asked Questions (FAQ).* Retrieved October 30, 2008, from Tenable Network Security web site: http://www.tenablesecurity.com/documentation/index.php?doc=faq#anchor59

Wikipedia.org, (2008a, October 21). *Nessus (software) - Wikipedia, the free encyclopedia.* Retrieved November 3, 2008 from Wikipedia.org Web site: http://en.wikipedia.org/wiki/Nessus_(software)

Wikipedia.org, (2008b, January 8). xinetd - *Wikipedia, the free encyclopedia.* Retrieved November 3, 2008 from Wikipedia.org Web site: http://en.wikipedia.org/wiki/Xinetd

7. Appendix A Listing of the Center for Internet Security Level 1 & Benchmark for Mac OS X audit file

PART 1

```
# Copyright 2008, R.D. Smith
#
# Name           : Mac OS X (10.5) CIS Benchmark Level 1 & 2, v1.0
#
#
# Description    : Covers Sections 2.1, 2.2, and 2.3
#
#
# Notes          : 1. Policy Items are not auditable from Nessus.
#                  2. Manual Check are technical checks that can not be done with
#                     Nessus and must be performed manually.
#
#
# Author         : R.D. Smith, GIAC Certified (GSE, GSNA), CISSP-ISSEP, CISA
# Date           : 20081105
#
# Version        : v0.9
#


<check_type:"Unix">

########################################
#                                      #
# Section 2.1 Installation Action Items #
#                                      #
########################################

#
# Policy Item
#  description: "Securely erase the Mac OS X partition before installation "
#  info: "Section 2.1.1, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
#  info: "version 1.0,"
#  info: "Level 2"
#  info: "Context: System"
#

#
# Policy Item
#  description: "Do not connect to the Internet when setting up a Mac "
#  info: "Section 2.1.2, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
#  info: "version 1.0,"
#  info: "Level 2"
#  info: "Context: System"
#

<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Install Mac OS X using Mac OS Extended Journaled disk format "
   info: "Section 2.1.3, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 1"
   info: "Context: System"
   cmd: "diskutil info /"
   expect: "File System:             Journaled HFS+"
</custom_item>
```

```
<custom_item>
   system: "Darwin"
   type: FILE_CHECK_NOT
   description: "Do not install any unnecessary packages "
   info: "Section 2.1.4, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 1"
   info: "Context: System"
   file: "/Library/Receipts/boms/com.apple.pkg.X11User.bom"
   owner: "_installer"
</custom_item>

#
# Policy Item
#  description: "Do not transfer confidential information in Setup Assistant "
#  info: "Section 2.1.5, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark, "
#  info: "version 1.0,"
#  info: "Level 1"
#  info: "Context: System"
#

<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Create administrator accounts with difficult-to-guess names "
   info: "Section 2.1.6, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 1"
   info: " "
   info: "NOTE: This check must be customized for site-specific names."
   info: " "
   info: "Context: System"
    cmd: "id -p 501 | grep -i -E
       'uid.*admin|uid.*administrator|uid.*supervisor|uid.*root'"
   expect: ""
</custom_item>

# Policy Item
#  description: "Create complex passwords for administrator accounts "
#  info: "Section 2.1.7, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
#  info: "version 1.0,"
#  info: "Level 1"
#  info: "Context: User"
#

#
# Policy Item
#  description: "Do not enter a password-related hint "
#  info: "Section 2.1.8, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
#  info: "version 1.0,"
#  info: "Level 1"
#  info: "Context: User"
#

#
# Policy Item
#  description: "Update system software using verified packages "
#  info: "Section 2.1.9, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
#  info: "version 1.0,"
#  info: "Level 2"
#  info: "Context: System"
#

##########################################################
#                                                        #
# Section 2.2 Hardware and Core Mac OS X Action Items  #
#                                                        #
##########################################################
```

```
<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Use an Open Firmware or EFI password "
   info: "Section 2.2.1, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 2"
   info: "Context: System"
   cmd: "sudo nvram -p | grep security-mode"
   expect: "security-mode.*command"
</custom_item>

<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Create an access warning for the login window "
   info: "Section 2.2.2, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 2"
   info: "Context: System"
   info: " "
   info: "NOTE: This check must be configured for site specific warning banner text."
   info: " "
   cmd: "defaults read /Library/Preferences/com.apple.loginwindow LoginwindowText |
      perl -0777 -nle 's/\\n//g;s/\\r//g;print;'"
   expect: ".*?Authorized Uses ONLY.*?All activity may be monitored and reported.*"
</custom_item>

<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Create an access warning for the command line "
   info: "Section 2.2.3, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 2"
   info: "Context: System"
   info: " "
   info: "NOTE: This check must be configured for site specific warning banner text."
   info: " "
   cmd: "grep -E '^Banner'  /etc/sshd_config | cut -d \" \" -f 2 | perl -nl -
      e'system(\"cat\", $_);' | perl -0777 -nle 's/\\n//g;s/\\r//g;print;'"
   expect: ".*?Authorized Uses ONLY.*?All activity may be monitored and reported.*"
</custom_item>

<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Disable Bluetooth check one"
   info: "Section 2.2.4, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 2"
   info: "Context: System"
   cmd: "system_profiler SPBluetoothDataType | perl -0777 -ne '$x = \"No Bluetooth\"
      . $_; if ($_ ne \"\") {print \"Bluetooth available\n\";} else {print $x .
      \"\n\";};'"
   expect: "No Bluetooth"
</custom_item>

<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Disable Bluetooth "
   info: "Section 2.2.4, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 2"
   info: "Context: System"
   cmd: "system_profiler SPBluetoothDataType | perl -0777 -ne '$x = \"No Bluetooth\"
      . $_; if ($_ ne \"\") {print \"Bluetooth available\n\";} else {print $x .
      \"\n\";};'"
```

```
   expect: "No Bluetooth"
</custom_item>

<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Disable the iSight camera "
   info: "Section 2.2.5, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 2"
   info: "Context: System"
   cmd: "system_profiler | grep iSight | perl -0777 -ne '$x = \"No iSight\" . $_; if
      ($_ ne \"\") {print \"$_\n\";} else {print $x . \"\n\";};'"
   expect: "No iSight"
</custom_item>

<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Reduce the sudo timeout period "
   info: "Section 2.2.6, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 2"
   info: "Context: System"
   cmd: "sudo cat /etc/sudoers | grep timestamp"
   expect: "Defaults.*timestamp_timeout=0"
</custom_item>

<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Remove unneeded QuickTime components "
   info: "Section 2.2.7, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 2"
   info: "Context: System and User"
   info: " "
   info: "NOTE: This check must be customized for site specific QuickTime
      components."
   info: " "
   cmd: "ls /Library/QuickTime | perl -0777 -ne '$x = \"X\" . $_; if ($_ ne \"\")
      {print \"$_\n\";} else {print $x . \"\n\";};'"
   expect: ".*"
   severity: MEDIUM
</custom_item>

<custom_item>
   system: "Darwin"
   type: CMD_EXEC     description: "Remove unneeded QuickTime components "
   info: "Section 2.2.7, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 2"
   info: "Context: System and User"
   info: " "
   info: "NOTE: This check must be customized for site specific QuickTime
      components."
   info: " "
   cmd: "ls /Library/Internet\\ Plug-Ins/"
   expect: ".*"
   severity: MEDIUM
</custom_item>

<custom_item>
   system: "Darwin"
   type: CMD_EXEC     description: "Disable Core Dumps "
   info: "Section 2.2.8, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 2"
   info: "Context: System and User"
```

```
   info: " "
   info: " "
   info: "Note: Executing 'launchctl limit core 0' will not survive a reboot."
   cmd: "launchctl limit core"
   expect: "core *0 *0"
   severity: MEDIUM
</custom_item>

##############################################
#                                            #
# Section 2.3 Account Configuration Items  #
#                                            #
##############################################

#
# Policy Item
#  description: "Create an administrator account and a standard account for each
      administrator "
#  info: "Section 2.3.1, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark, "
#  info: "version 1.0,"
#  info: "Level 1"
#  info: "Context: User"
#

#
# Policy Item
#  description: "Create a standard or managed account for each non-administrator "
#  info: "Section 2.3.2, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark, "
#  info: "version 1.0,"
#  info: "Level 1"
#  info: "Context: User"
#

#
# Policy Item
#  description: "Set appropriate parental controls for managed accounts "
#  info: "Section 2.3.3, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark, "
#  info: "version 1.0,"
#  info: "Level 1"
#  info: "Context: User"
#

#
# Manual Check
#  description: "Restrict sudo users to being able to access only required commands "
#  info: "Section 2.3.4, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark, "
#  info: "version 1.0,"
#  info: "Level 2"
#  info: "Context: System and User"
#

#
# Policy Item
#  description: "Securely configure LDAPv3 access "
#  info: "Section 2.3.5, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark, "
#  info: "version 1.0,"
#  info: "Level 2"
#  info: "Context: System"
#

#
# Policy Item
#  description: "Securely configure Active Directory access "
#  info: "Section 2.3.6, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark, "
#  info: "version 1.0,"
#  info: "Level 2"
#  info: "Context: System"
#
```

```
#
# Policy Item
#  description: "Use Password Assistant to help generate complex passwords "
#  info: "Section 2.3.7, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark, "
#  info: "version 1.0,"
#  info: "Level 1"
#  info: "Context: User"
#

#
# Policy Item
#  description: "Set a strong password policy "
#  info: "Section 2.3.8, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark, "
#  info: "version 1.0,"
#  info: "Level 2"
#  info: "Context: System"
#

#
# Policy Item
#  description: "Secure the login keychain "
#  info: "Section 2.3.9, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark, "
#  info: "version 1.0,"
#  info: "Level 2"
#  info: "Context: User"
#

#
# Policy Item
#  description: "Secure individual keychain items "
#  info: "Section 2.3.10, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark, "
#  info: "version 1.0,"
#  info: "Level 2"
#  info: "Context: User"
#

#
# Policy Item
#  description: "Create specialized keychains for different purposes "
#  info: "Section 2.3.11, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark, "
#  info: "version 1.0,"
#  info: "Level 2"
#  info: "Context: User"
#

#
# Policy Item
#  description: "Use a portable drive to store keychains "
#  info: "Section 2.3.12, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark, "
#  info: "version 1.0,"
#  info: "Level 2"
#  info: "Context: User"
#

<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Do not enable the 'root' account "
   info: "Section 2.3.13, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 1"
   info: "Context: System"
   cmd: "dscl . -read /Users/root AuthenticationAuthority"
   expect: "No such key|DisabledUser"
</custom_item>

</check_type>
```

PART 2

```
# Copyright 2008, R.D. Smith
#
# Name           : Mac OS X (10.5) CIS Benchmark Level 1 & 2, v1.0
#
#
# Description    : Covers Section 2.4.1-2.4.12
#
#
# Notes          : 1. Policy Items are not auditable from Nessus.
#                  2. Manual Check are technical checks that can not be done with
#                     Nessus and must be performed manually.
#
#
# Author         : R.D. Smith, GIAC Certified (GSE, GSNA), CISSP-ISSEP, CISA
# Date           : 20081105
#
# Version        : v0.9
#


<check_type:"Unix">

########################################################
#                                                      #
# Section 2.4 Securing System Software Action Items    #
#                                                      #
########################################################

#
# Manual Check (potential perl or shell script)
#  description: "Do not enable .Mac for administrative accounts "
#  info: "Section 2.4.1.1, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
#  info: "version 1.0,"
#  info: "Level 1"
#  info: "Context: User"
#

#
# Manual Check (the file is now in
      ~/Library/Preferences/ByHost/com.apple.DotMacSync.XX where XX is unique for
      each machine.
#  description: "Disable all Sync options "
#  info: "Section 2.4.1.2, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark, "
#  info: "version 1.0,"
#  info: "Level 2"
#  info: "Context: User"
#
<if>
   <condition type: "or">
      <custom_item>
          system: "Darwin"
          type: FILE_CHECK_NOT
          description: "Disable all Sync options (file check)"
          info: "Section 2.4.1.2, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
          info: "version 1.0,"
          info: "Level 2"
          info: "Context: User"
          file: "~/Library/Preferences/ByHost/com.apple.DotMacSync.*"
          mode: "0600"
      </custom_item>
   </condition>
<then>
   # do nothing
</then>
<else>
      <custom_item>
```

```
            system: "Darwin"
            type: CMD_EXEC
            description: "Disable all Sync options (ShouldSyncWithServer check"
            info: "Section 2.4.1.2, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
              info: "version 1.0,"
            info: "Level 2"
            info: "Context: User"
            cmd: "ifconfig en0 | grep -E 'ether' | cut -d \" \" -f 2 |perl -nl -e
        's/://g;s/(.*)/defaults read
        ~\\/Library\\/Preferences\\/ByHost\\/com.apple.DotMacSyc.\\1
        ShouldSyncWithServer/; system($_);'"
            expect: "0"
        </custom_item>
</else>

#
# Manual Check (potential perl or shell script)
#  description: "Disable iDisk Syncing "
#  info: "Section 2.4.1.3, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark, "
#  info: "version 1.0,"
#  info: "Level 2"
#  info: "Context: User"
#

#
# Policy Item/Manual Check
#  description: "Enable Public Folder password protection "
#  info: "Section 2.4.1.4, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark, "
#  info: "version 1.0,"
#  info: "Level 1"
#  info: "Context: User"
#

#
# Policy Item/Manual Check
#  description: "Do not register computers for synchronization "
#  info: "Section 2.4.1.5, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark, "
#  info: "version 1.0,"
#  info: "Level 2"
#  info: "Context: User"
#

#
# Policy Item/Manual Check
#  description: "Sign out of .Mac if signed in "
#  info: "Section 2.4.1.6, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark, "
#  info: "version 1.0,"
#  info: "Level 1"
#  info: "Context: User"
#

<if>
   <condition type: "or">
      <custom_item>
         system: "Darwin"
         type: FILE_CHECK
         description: "Disable the .Mac preference pane from System Preferences
      (owned by root check)"
         info: "Section 2.4.1.7, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
         info: "Level 2"
         info: "Context: System"
         file: "/System/Library/PreferencePanes/Mac.prefPane"
         owner: "root"
      </custom_item>
   </condition>
<then>
      <custom_item>
         system: "Darwin"
```

```
            type: FILE_CHECK
            description: "Disable the .Mac preference pane from System Preferences
        (owned by root and mode 0700 check)"
            info: "Section 2.4.1.7, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
    info: "version 1.0,"
            info: "Level 2"
            info: "Context: System"
            file: "/System/Library/PreferencePanes/Mac.prefPane"
            mode: "0700"
        </custom_item>
</then>
<else>
        <custom_item>
            system: "Darwin"
            type: FILE_CHECK_NOT
            description: "Disable the .Mac preference pane from System Preferences
        (check that it doesn't exist"
            info: "Section 2.4.1.7, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
    info: "version 1.0,"
            info: "Level 2"
            info: "Context: System"
            file: "/System/Library/PreferencePanes/Mac.prefPane"
            group: "wheel"
        </custom_item>
</else>

#
# Policy Item
#  description: "Change initial password for the system administrator account "
#  info: "Section 2.4.2.1, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
#  info: "version 1.0,"
#  info: "Level 1"
#  info: "Context: User"
#

<custom_item>
    system: "Darwin"
    type: CMD_EXEC     description: "Disable automatic login "
    info: "Section 2.4.2.2, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
    info: "version 1.0,"
    info: "Level 1"
    info: "Context: System"
    cmd: "defaults read /Library/Preferences/.GlobalPreferences
        com.apple.userspref.DisableAutoLogin"
    expect: "1"
</custom_item>

<custom_item>
    system: "Darwin"
    type: CMD_EXEC     description: "Display login window as name and password "
    info: "Section 2.4.2.3, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
    info: "version 1.0,"
    info: "Level 1"
    info: "Context: System"
    cmd: "defaults read /Library/Preferences/com.apple.loginwindow SHOWFULLNAME"
    expect: "1"
</custom_item>

<custom_item>
    system: "Darwin"
    type: CMD_EXEC
    description: "Disable 'Show password hints' "
    info: "Section 2.4.2.4, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
    info: "version 1.0,"
    info: "Level 1"
    info: "Context: System"
    cmd: "defaults read /Library/Preferences/com.apple.loginwindow RetriesUntilHint"
    expect: "0"
</custom_item>
```

```
#
# Policy Item/Manual Check
#  description: "Configure 'Allow network users to login to this computer' "
#  info: "Section 2.4.2.5, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
#  info: "version 1.0,"
#  info: "Level 2"
#  info: "Context: System"
#

<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Disable 'Enable fast user switching' "
   info: "Section 2.4.2.6, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 2"
   info: "Context: System"
   cmd: "defaults read /Library/Preferences/.GlobalPreferences
      MultipleSessionEnabled"
   expect: "0"
</custom_item>

<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Disable 'Allow guest to log into this computer' "
   info: "Section 2.4.2.7, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 1"
   info: "Context: System"
   cmd: "dscl . -read /Users/Guest AuthenticationAuthority"
   expect: "AuthenticationAuthority: ;basic;|.*\(eDSRecordNotFound\).*"
</custom_item>

<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Disable 'Allow guests to connect to shared folders' (AFP check) "
   info: "Section 2.4.2.8, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 1"
   info: "Context: System"
   cmd: "defaults read /Library/Preferences/com.apple.AppleFileServer guestAccess"
   expect: "0"
</custom_item>

<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Disable 'Allow guests to connect to shared folders' (SMB check) "
   info: "Section 2.4.2.8, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 1"
   info: "Context: System"
   cmd: "defaults read /Library/Preferences/SystemConfiguration/com.apple.smb.server
      AllowGuestAccess"
   expect: "0"
</custom_item>

<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Disable Bluetooth by using System Preferences for each user account
      "
   info: "Section 2.4.3.1, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 1"
   info: "Context: User"
```

```
   cmd: "defaults read /Library/Preferences/com.apple.Bluetooth ControllerPowerState"
   expect: "0"
</custom_item>

<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Disable Bluetooth internet connection sharing "
   info: "Section 2.4.3.2, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 1"
   info: "Context: System"
   cmd: "defaults read /Library/Preferences/com.apple.Bluetooth PANServices"
   expect: "0|The domain/default pair of (com.apple.Bluetooth, PANServices) does not
      exist"
</custom_item>

#
# Policy Item/Manual Check
#  description: "If Bluetooth is used, turn off 'Discoverable' when not needed "
#  info: "Section 2.4.3.3, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
#  info: "version 1.0,"
#  info: "Level 1"
#  info: "Context: User"
#
<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Disable all Sync options "
   info: "Section 2.4.1.2, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 2"
   info: "Context: User"
   info: " "
   info: "NOTE: This check must be configured for site specific warning banner text."
   info: " "
   cmd: "ifconfig en0 | grep -E 'ether' | cut -d \" \" -f 2 |perl -nl -e
      's/://g;s/(.*)/defaults read
      ~\\/Library\\/Preferences\\/ByHost\\/com.apple.Bluetooth.\\1
      DiscoverableState/; system($_);'"
   expect: "0"
</custom_item>

#
# Policy Item/Manual Check
#  description: "Show Bluetooth status in menu bar "
#  info: "Section 2.4.3.4, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
#  info: "version 1.0,"
#  info: "Level 1"
#  info: "Context: User"
#

<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Disable automatic actions for blank CDs for each user account "
   info: "Section 2.4.4.1, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 2"
   info: "Context: User"
   cmd: "defaults read com.apple.digihub com.apple.digihub.blank.cd.appeared"
   expect: ".*action = 1.*"
</custom_item>

<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Disable automatic actions for blank DVDs for each user account "
   info: "Section 2.4.4.2, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
```

```
    info: "version 1.0,"
    info: "Level 2"
    info: "Context: User"
    cmd: "defaults read com.apple.digihub com.apple.digihub.blank.dvd.appeared"
    expect: ".*action = 1.*"
</custom_item>

<custom_item>
    system: "Darwin"
    type: CMD_EXEC
    description: "Disable automatic actions for music CDs for each user account "
    info: "Section 2.4.4.3, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
    info: "version 1.0,"
    info: "Level 2"
    info: "Context: User"
    cmd: "defaults read com.apple.digihub com.apple.digihub.cd.music.appeared"
    expect: ".*action = 1.*"
</custom_item>

<custom_item>
    system: "Darwin"
    type: CMD_EXEC
    description: "Disable automatic actions for picture CDs for each user account "
    info: "Section 2.4.4.4, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
    info: "version 1.0,"
    info: "Level 2"
    info: "Context: User"
    cmd: "defaults read com.apple.digihub com.apple.digihub.cd.picture.appeared"
    expect: ".*action = 1.*"
</custom_item>

<custom_item>
    system: "Darwin"
    type: CMD_EXEC
    description: "Disable automatic actions for video DVDs for each user account "
    info: "Section 2.4.4.5, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
    info: "version 1.0,"
    info: "Level 2"
    info: "Context: User"
    cmd: "defaults read com.apple.digihub com.apple.digihub.dvd.video.appeared"
    expect: ".*action = 1.*"
</custom_item>

<custom_item>
    system: "Darwin"
    type: FILE_CONTENT_CHECK
    description: "Enter correct time settings "
    info: "Section 2.4.5.1, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
    info: "version 1.0,"
    info: "Level 1"
    info: "Context: System"
    file: "/etc/hostconfig"
    regex: ".*TIMESYNC=.*$"
    expect: ".*TIMESYNC=-YES-"
</custom_item>

<custom_item>
    system: "Darwin"
    type: FILE_CONTENT_CHECK
    description: "Enter correct time settings "
    info: "Section 2.4.5.1, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
    info: "version 1.0,"
    info: "Level 1"
    info: "Context: System"
    info: " "
    info: "Change check to contain an approved NTP server.  Repeat check if multiple
       NTP servers are approved."
    info: " "
    file: "/etc/ntp.conf"
```

```
   regex: ".*server.*$"
   expect: ".*server time-a.nist.gov"
</custom_item>

<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Use an internal Software Update server "
   info: "Section 2.4.5.2, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 2"
   info: "Context: System"
   cmd: "defaults read /Library/Preferences/com.apple.SoftwareUpdate CatalogURL"
   expect: "<URL>"
   info: "This check must be customized for the site's software update server URL."
</custom_item>

<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Set a short inactivity interval for the screen saver "
   info: "Section 2.4.6.1, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 1"
   info: "Context: User"
   cmd: "defaults -currentHost read com.apple.screensaver idleTime"
   expect: ".*idleTime = 900.*"
</custom_item>

<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Disable sleeping the computer when connected to power "
   info: "Section 2.4.7.1, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 2"
   info: "Context: System"
   cmd: "pmset -g | grep sleep"
   expect: " sleep[[:space:]]*0"
</custom_item>

<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Verify Display Sleep is set to a value larger than the Screen Saver
      "
   info: "Section 2.4.7.2, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 1"
   info: "Context: System"
   cmd: "pmset -g | grep displaysleep"
   expect: " displaysleep[[:space:]]*20"
   info: "This value is given in minutes. Screen Saver idleTime is in seconds."
</custom_item>

<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Disable 'Wake when the modem detects a ring' for all power settings
      "
   info: "Section 2.4.7.3, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 1"
   info: "Context: System"
   cmd: "pmset -g | grep ring"
   expect: " ring[[:space:]]*20"
   info: "This requires an internal modem installed for this check to succeed."
</custom_item>
```

```
<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Disable 'Wake for Ethernet network administrator access' for power
      adapter settings "
   info: "Section 2.4.7.4, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 1"
   info: "Context: System"
   cmd: "pmset -g | grep womp"
   expect: " womp[[:space:]]*0"
   info: "This requires an internal modem installed for this check to succeed."
</custom_item>

<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Do not set any screen corner to Disable Screen Saver "
   info: "Section 2.4.8.1, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 1"
   info: "Context: User"
   cmd: "defaults read com.apple.dock | grep -E 'wvous-..-
      corner.[[:space:]]*=[[:space:]]*1'"
   expect: ""
   info: "This should cover all *-corner keys."
</custom_item>

<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Set a screen corner to Start Screen Saver "
   info: "Section 2.4.8.2, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 1"
   info: "Context: User"
   cmd: "defaults read com.apple.dock | grep -E 'wvous-..-
      corner.[[:space:]]*=[[:space:]]*5'"
   expect: ".*wvous-..-corner = 5.*"
   info: "This should cover all *-corner keys."
</custom_item>

<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Do not set any screen corner to Sleep Display "
   info: "Section 2.4.8.3, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 1"
   info: "Context: User"
   cmd: "defaults read com.apple.dock | grep -E 'wvous-..-
      corner.[[:space:]]*=[[:space:]]*10'"
   expect: ""
   info: "This should cover all *-corner keys."
</custom_item>

#
# Manual Check (the file is now in ~/Library/Preferences/ByHost/com.apple.Bluetooth.XX
      where XX is unique for each machine.
#  description: "Disable 'Allow Bluetooth devices to wake this computer' "
#  info: "Section 2.4.9.1, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
#  info: "version 1.0,"
#  info: "Level 1"
#  info: "Context: User"
#

#
# Manual Check
#  description: "Create network specific locations "
```

```
#  info: "Section 2.4.10.1, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
#  info: "version 1.0,"
#  info: "Level 2"
#  info: "Context: System"
#

#
# Manual Check
#  description: "Disable AirPort "
#  info: "Section 2.4.10.2, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
#  info: "version 1.0,"
#  info: "Level 1"
#  info: "Context: System"
#

#
# Manual Check
#  description: "Enable Show AirPort Status in Menu Bar "
#  info: "Section 2.4.10.3, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
#  info: "version 1.0,"
#  info: "Level 1"
#  info: "Context: System"
#

#
# Manual Check
#  description: "Disable Bluetooth "
#  info: "Section 2.4.10.4, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
#  info: "version 1.0,"
#  info: "Level 1"
#  info: "Context: System"
#

#
# Manual Check
#  description: "Disable IPv6 "
#  info: "Section 2.4.10.5, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
#  info: "version 1.0,"
#  info: "Level 2"
#  info: "Context: System"
#

#
# Policy Item
#  description: "Only use known printers "
#  info: "Section 2.4.11.1, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
#  info: "version 1.0,"
#  info: "Level 1"
#  info: "Context: User"
#

#
# Manual Check
#  description: "Disable receiving faxes "
#  info: "Section 2.4.11.2, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
#  info: "version 1.0,"
#  info: "Level 1"
#  info: "Context: System"
#

#
# Manual Check
#  description: "Disable 'Save movies in disk cache' "
#  info: "Section 2.4.12.1, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
#  info: "version 1.0,"
#  info: "Level 2"
#  info: "Context: User"
#
```

```
<custom_item>
   system: "Darwin"
   type: FILE_CHECK
   description: "Do not install third-party QuickTime software "
   info: "Section 2.4.12.2, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 1"
   info: "Context: System and User"
   file: "/Library/QuickTime/AppleIntermediateCodec.component"
   group: "admin"
   info: "Must be repeated for known good components."
</custom_item>

#
# Manual Check
#  description: "Disable 'Play Movies automatically' "
#  info: "Section 2.4.12.3, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
#  info: "version 1.0,"
#  info: "Level 1"
#  info: "Context: User"
#

</check_type>
```

PART 3

```
# Copyright 2008, R.D. Smith
#
# Name           : Mac OS X (10.5) CIS Benchmark Level 1 & 2, v1.0
#
#
# Description    : Covers Section 2.4.13-2.4.18
#
#
# Notes          : 1. Policy Items are not auditable from Nessus.
#                  2. Manual Check are technical checks that can not be done with
#                     Nessus and must be performed manually.
#
# Author         : R.D. Smith, GIAC Certified (GSE, GSNA), CISSP-ISSEP, CISA
# Date           : 20081105
#
# Version        : v0.9
#


<check_type:"Unix">

#######################################################
#                                                     #
# Section 2.4 Securing System Software Action Items #
#                                                     #
#######################################################


<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Require a password to wake the computer from sleep or screen saver "
   info: "Section 2.4.13.1, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 1"
   info: "Context: User"
   cmd: "defaults -currentHost read com.apple.screensaver askForPassword"
   expect: "1"
</custom_item>

<custom_item>
```

```
    system: "Darwin"
    type: CMD_EXEC
    description: "Disable automatic login "
    info: "Section 2.4.13.2, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
    info: "version 1.0,"
    info: "Level 1"
    info: "Context: System"
    cmd: "defaults read /Library/Preferences/.GlobalPreferences
       com.apple.userspref.DisableAutoLogin"
    expect: "1"
</custom_item>

<custom_item>
    system: "Darwin"
    type: CMD_EXEC
    description: "Require a password to unlock each System Preferences pane "
    info: "Section 2.4.13.3, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
    info: "version 1.0,"
    info: "Level 1"
    info: "Context: System"
    cmd: "perl -0777 -ne 'if
       (/<key>system.preferences<.*?>shared<\\/key>\\s*?<true\\/>.*?<key>system.prefer
       ences.access/s) {print \"shared prefs: true\\n\";} else {print \"shared prefs:
       false\\n\";}; ' /etc/authorization"
    expect: "shared prefs: false"
</custom_item>

<custom_item>
    system: "Darwin"
    type: CMD_EXEC
    description: "Disable 'automatic logout' after a period of inactivity "
    info: "Section 2.4.13.4, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
    info: "version 1.0,"
    info: "Level 1"
    info: "Context: System"
    cmd: "defaults read /Library/Preferences/.GlobalPreferences
       com.apple.autologout.AutoLogOutDelay"
    expect: "0|.*The domain/default pair of (.GlobalPreferences,
       com.apple.autologout.AutoLogOutDelay) does not exist.*"
</custom_item>

<custom_item>
    system: "Darwin"
    type: CMD_EXEC
    description: "Use secure virtual memory "
    info: "Section 2.4.13.5, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
    info: "version 1.0,"
    info: "Level 1"
    info: "Context: System"
    cmd: "defaults read /Library/Preferences/com.apple.virtualMemory UseEncryptedSwap"
    expect: "1"
</custom_item>

<if>
    <condition type: "or">
     <custom_item>
    system: "Darwin"
    type: CMD_EXEC
    description: "Disable remote control infrared receiver (no receiver check)"
    info: "Section 2.4.13.6, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
    info: "version 1.0,"
    info: "Level 1"
    info: "Context: System"
    cmd: "defaults read /Library/Preferences/com.apple.driver.AppleIRController"
    expect: ".*Domain com.apple.driver.AppleIRController does not exist.*"
     </custom_item>
    </condition>
<then>
    # do nothing
```

```
</then>
<else>
   <if>
   <condition type: "or">
      <custom_item>
         system: "Darwin"
         type: CMD_EXEC
         description: "Disable remote control infrared receiver (receiver disabled
      check)"
         info: "Section 2.4.13.6, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
         info: "version 1.0,"
         info: "Level 1"
         info: "Context: System"
         cmd: "defaults read /Library/Preferences/com.apple.driver.AppleIRController"
         expect: ".*DeviceEnabled.*=.*0|.*Domain com.apple.driver.AppleIRController
      does not exist.*"
      </custom_item>
   </condition>
   <then>
   <custom_item>
         system: "Darwin"
         type: CMD_EXEC
         description: "Disable remote control infrared receiver (receiver enabled but
      paired check)"
         info: "Section 2.4.13.6, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
         info: "version 1.0,"
         info: "Level 1"
         info: "Context: System"
         cmd: "defaults read /Library/Preferences/com.apple.driver.AppleIRController"
         expect: ".*UIDFilter.*=.*\d*.*"
   </custom_item>
   </then>
   <else>
   # do nothing
   </else>
</else>

<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Pair the remote control infrared receiver "
   info: "Section 2.4.13.7, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 1"
   info: "Context: System"
   cmd: "defaults read /Library/Preferences/com.apple.driver.AppleIRController"
   expect: ".*DeviceEnabled.*=.*0|.*UIDFilter.*=.*\d*.*|.*Domain
      com.apple.driver.AppleIRController does not exist.*"
</custom_item>

#
# Manual Check (potential perl or shell script)
#  description: "Enable FileVault for every account "
#  info: "Section 2.4.13.8, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark, "
#  info: "version 1.0,"
#  info: "Level 2"
#  info: "Context: User"
#

<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Enable firewall protection "
   info: "Section 2.4.13.9, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 1"
   info: "Context: System"
   cmd: "defaults read /Library/Preferences/com.apple.alf globalstate"
   expect: "1|2"
```

```
</custom_item>

<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Change the computer name "
   info: "Section 2.4.14.1, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 2"
   info: "Context: System"
   info: " "
   info: "Must be modified for the local machine."
   cmd: "systemsetup -getcomputername"
   expect: ".*Mac.*"
</custom_item>

<custom_item>
   system: "Darwin"
   type: FILE_CHECK_NOT
   description: "Configure Screen Sharing "
   info: "Section 2.4.14.2, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 1"
   info: "Context: System"
   file: "/Library/Preferences/com.apple.ScreenSharing.launchd"
   group: "admin"
</custom_item>

<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Configure File Sharing (AFS check)"
   info: "Section 2.4.14.3, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 1"
   info: "Context: System"
   cmd: "perl -0777 -nle 'if /<key>Disabled<.key\>.*<true.>/) {print
      \"Disabled\\n\";} else {print \"Service Enabled\\n\";};'
      /System/Library/LaunchDaemons/com.apple.AppleFileServer.plist"
   expect: "Disabled"
</custom_item>

<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Configure File Sharing (FTP check)"
   info: "Section 2.4.14.3, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 1"
   info: "Context: System"
   cmd: "perl -0777 -nle 'if /<key>Disabled<.key\>.*<true.>/) {print
      \"Disabled\\n\";} else {print \"Service Enabled\\n\";};'
      /System/Library/LaunchDaemons/ftp.plist"
   expect: "Disabled"
</custom_item>

<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Configure File Sharing  (NMB Check)"
   info: "Section 2.4.14.3, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 1"
   info: "Context: System"
   cmd: "perl -0777 -nle 'if /<key>Disabled<.key\>.*<true.>/) {print
      \"Disabled\\n\";} else {print \"Service Enabled\\n\";};'
      /System/Library/LaunchDaemons/nmbd.plist"
   expect: "Disabled"
</custom_item>
```

```
<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Configure File Sharing (SMB Check)"
   info: "Section 2.4.14.3, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 1"
   info: "Context: System"
   cmd: "perl -0777 -nle 'if /<key>Disabled<.key\>.*<true.>/) {print
      \"Disabled\\n\";} else {print \"Service Enabled\\n\";};'
      /System/Library/LaunchDaemons/smbd.plist"
   expect: "Disabled"
</custom_item>

<if>
   <condition type: "or">
      <custom_item>
         system: "Darwin"
         type: CMD_EXEC
         description: "Configure File Sharing (SMB Check)"
         info: "Section 2.4.14.3, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
         info: "version 1.0,"
         info: "Level 1"
         info: "Context: System"
         cmd: "perl -0777 -nle 'if /<key>Disabled<.key\>.*<true.>/) {print
      \"Disabled\\n\";} else {print \"Service Enabled\\n\";};'
      /System/Library/LaunchDaemons/smbd.plist"
         expect: "Disabled"
      </custom_item>
   </condition>
<then>
   # do nothing
</then>
<else>
   <custom_item>
         system: "Darwin"
         type: FILE_CONTENT_CHECK
         description: "Secure SMB (restrict anonymous check)"
         info: "Section 2.4.14.4, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
         info: "version 1.0,"
         info: "Level 2"
         info: "Context: System"
         info: " "
         info: "Note: This check is performed only if SMB File Sharing is enabled."
         info: " "
         file: "smb.conf"
         search_locations: "/etc"
         regex: ".*restrict anonymous.*=.*2.*$"
         expect: ".*restrict anonymous.*=.*2.*"
   </custom_item>
   <custom_item>
         system: "Darwin"
         type: FILE_CONTENT_CHECK
         description: "Secure SMB (disable guest access check)"
         info: "Section 2.4.14.4, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
         info: "version 1.0,"
         info: "Level 2"
         info: "Context: System"
         info: " "
         info: "Note: This check is performed only if SMB File Sharing is enabled."
         info: " "
         file: "smb.conf"
         search_locations: "/etc"
         regex: ".*guest OK.*=.*no.*$"
         expect: ".*guest OK.*=.*no.*"
   </custom_item>
   <custom_item>
         system: "Darwin"
```

```
         type: FILE_CONTENT_CHECK
         description: "Secure SMB (require NTLMv2 authentication check)"
         info: "Section 2.4.14.4, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
         info: "version 1.0,"
         info: "Level 2"
         info: "Context: System"
         info: " "
         info: "Note: This check is performed only if SMB File Sharing is enabled."
         info: " "
         file: "smb.conf"
         search_locations: "/etc"
         regex: ".*client ntlmv2 auth.*=.*yes.*$"
         expect: ".*client ntlmv2 auth.*=.*yes.*"
   </custom_item>
</else>

<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Configure Printer Sharing "
   info: "Section 2.4.14.5, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 1"
   info: "Context: System"
   cmd: "perl -0777 -nle 'if /<key>Disabled<.key\>.*<true.>/) {print
      \"Disabled\\n\";} else {print \"Service Enabled\\n\";};'
      /System/Library/LaunchDaemons/org.cups.cups-lpd.plist"
   expect: "Disabled"
</custom_item>

<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Configure Web Sharing "
   info: "Section 2.4.14.6, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"

    info: "version 1.0,"
   info: "Level 1"
   info: "Context: System"
   file: "org.apache.httpd.plist"
   cmd: "perl -0777 -nle 'if /<key>Disabled<.key\>.*<true.>/) {print
      \"Disabled\\n\";} else {print \"Service Enabled\\n\";};'
      /System/Library/LaunchDaemons/org.apache.httpd.plist"
   expect: "Disabled"
</custom_item>

<if>
   <condition type: "or">
      <custom_item>
         system: "Darwin"
         type: CMD_EXEC
         description: "Configure Web Sharing "
         info: "Section 2.4.14.6, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
         info: "version 1.0,"
         info: "Level 1"
         info: "Context: System"
         file: "org.apache.httpd.plist"
         cmd: "perl -0777 -nle 'if /<key>Disabled<.key\>.*<true.>/) {print
      \"Disabled\\n\";} else {print \"Service Enabled\\n\";};'
      /System/Library/LaunchDaemons/org.apache.httpd.plist"
         expect: "Disabled"
      </custom_item>
   </condition>
<then>
   # do nothing
</then>
<else>
      <custom_item>
         system: "Darwin"
```

```
            type: FILE_CONTENT_CHECK
            description: "Secure Web Sharing (ServerSignature check)"
            info: "Section 2.4.14.7, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
            info: "version 1.0,"
            info: "Level 2"
            info: "Context: System"
            info: " "
            info: "Note: This check is performed only if Web Sharing is enabled."
            info: " "
            file: "httpd.conf"
            search_locations: "/etc/apache2"
            regex: ".*ServerSignature.*Off.*$"
            expect: ".*ServerSignature.*Off.*"
        </custom_item>
        <custom_item>
            system: "Darwin"
            type: FILE_CONTENT_CHECK
            description: "Secure Web Sharing  (UserDir check)"
            info: "Section 2.4.14.7, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
            info: "version 1.0,"
            info: "Level 2"
            info: "Context: System"
            info: " "
            info: "Note: This check is performed only if Web Sharing is enabled."
            info: " "
            file: "httpd.conf"
            search_locations: "/etc/apache2"
            regex: ".*UserDir.*Disabled.*$"
            expect: ".*UserDir.*Disabled.*"
        </custom_item>
        <custom_item>
            system: "Darwin"
            type: FILE_CONTENT_CHECK
            description: "Secure Web Sharing (TraceEnable check)"
            info: "Section 2.4.14.7, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
            info: "version 1.0,"
            info: "Level 2"
            info: "Context: System"
            info: " "
            info: "Note: This check is performed only if Web Sharing is enabled."
            info: " "
            file: "httpd.conf"
            search_locations: "/etc/apache2"
            regex: ".*TraceEnable.*Off.*$"
            expect: ".*TraceEnable.*Off.*"
        </custom_item>
</else>

<custom_item>
    system: "Darwin"
    type: CMD_EXEC
    description: "Configure Remote Login "
    info: "Section 2.4.14.8, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
    info: "version 1.0,"
    info: "Level 1"
    info: "Context: System"
    cmd: "perl -0777 -nle 'if /<key>Disabled<.key\>.*<true.>/) {print
        \"Disabled\\n\";} else {print \"Service Enabled\\n\";};'
        /System/Library/LaunchDaemons/ssh.plist"
    expect: "Disabled"
</custom_item>

<if>
    <condition type: "or">
        <custom_item>
            system: "Darwin"
            type: CMD_EXEC
            description: "Configure Remote Login "
            info: "Section 2.4.14.8, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
```

```
         info: "version 1.0,"
         info: "Level 1"
         info: "Context: System"
         cmd: "perl -0777 -nle 'if /<key>Disabled<.key\>.*<true.>/) {print
      \"Disabled\\n\";} else {print \"Service Enabled\\n\";};'
      /System/Library/LaunchDaemons/ssh.plist"
         expect: "Disabled"
      </custom_item>
   </condition>
<then>
   # do nothing
</then>
<else>
      <custom_item>
         system: "Darwin"
         type: FILE_CONTENT_CHECK
         description: "Secure Remote Login (GSSAPIAuthentication check) "
         info: "Section 2.4.14.9, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
         info: "version 1.0,"
         info: "Level 2"
         info: "Context: System"
         info: " "
         info: "Note: This check is performed only if Remote Login is enabled."
         info: " "
         file: "sshd_config"
         search_locations: "/etc"
         regex: ".*GSSAPIAuthentication.*yes.*$"
         expect: ".*GSSAPIAuthentication.*yes.*"
      </custom_item>
      <custom_item>
         system: "Darwin"
         type: FILE_CONTENT_CHECK
         description: "Secure Remote Login (GSSAPICleanupCredentials check)"
         info: "Section 2.4.14.9, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
         info: "version 1.0,"
         info: "Level 2"
         info: "Context: System"
         info: " "
         info: "Note: This check is performed only if Remote Login is enabled."
         info: " "
         file: "sshd_config"
         search_locations: "/etc"
         regex: ".*GSSAPICleanupCredentials.*yes.*$"
         expect: "GSSAPICleanupCredentials.*yes.*"
      </custom_item>
      <custom_item>
         system: "Darwin"
         type: FILE_CONTENT_CHECK
         description: "Secure Remote Login (Protocol 2 Only Check)"
         info: "Section 2.4.14.9, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
         info: "version 1.0,"
         info: "Level 2"
         info: "Context: System"
         info: " "
         info: "Note: This check is performed only if Remote Login is enabled."
         info: " "
         file: "sshd_config"
         search_locations: "/etc"
         regex: "^Protocol *2$"
         expect: "Protocol *2"
      </custom_item>
</else>

<custom_item>
   system: "Darwin"
   type: FILE_CHECK_NOT
   description: "Configure Remote Management "
   info: "Section 2.4.14.10, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
```

```
    info: "Level 1"
    info: "Context: System"
    file: "/Library/Preferences/com.apple.RemoteManagement.launchd"
    group: "admin"
</custom_item>

<custom_item>
    system: "Darwin"
    type: CMD_EXEC
    description: "Configure Remote Apple Events "
    info: "Section 2.4.14.11, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
    info: "version 1.0,"
    info: "Level 1"
    info: "Context: System"
    cmd: "perl -0777 -nle 'if /<key>Disabled<.key\>.*<true.>/) {print
        \"Disabled\\n\";} else {print \"Service Enabled\\n\";};'
        /System/Library/LaunchDaemons/eppc.plist"
    expect: "Disabled"
</custom_item>

<custom_item>
    system: "Darwin"
    type: FILE_CONTENT_CHECK
    description: "Configure Xgrid Sharing "
    info: "Section 2.4.14.12, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
    info: "version 1.0,"
    info: "Level 1"
    info: "Context: System"
    cmd: "perl -0777 -nle 'if /<key>Disabled<.key\>.*<true.>/) {print
        \"Disabled\\n\";} else {print \"Service Enabled\\n\";};'
        /System/Library/LaunchDaemons/com.apple.xgridagentd.plist.plist"
    expect: "Disabled"
</custom_item>

<custom_item>
    system: "Darwin"
    type: CMD_EXEC
    description: "Configure Internet Sharing "
    info: "Section 2.4.14.13, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
    info: "version 1.0,"
    info: "Level 1"
    info: "Context: System"
    cmd: "perl -0777 -nle 'if /<key>Disabled<.key\>.*<true.>/) {print
        \"Disabled\\n\";} else {print \"Service Enabled\\n\";};'
        /System/Library/LaunchDaemons/com.apple.InternetSharing.plist"
    expect: "Disabled"
</custom_item>

<custom_item>
    system: "Darwin"
    type: FILE_CHECK_NOT
    description: "Completely disable sharing services (AFS check) "
    info: "Section 2.4.14.14, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
    info: "version 1.0,"
    info: "Level 2"
    info: "Context: System"
    file: "com.apple.AppleFileServer.plist"
    search_locations: "/System/Library/LaunchDaemons"
    group: "admin"
</custom_item>

<custom_item>
    system: "Darwin"
    type: FILE_CHECK_NOT
    description: "Completely disable sharing services (FTP check)"
    info: "Section 2.4.14.14, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
    info: "version 1.0,"
    info: "Level 2"
    info: "Context: System"
```

```
   file: "ftp.plist"
   search_locations: "/System/Library/LaunchDaemons"
   group: "admin"
</custom_item>

<custom_item>
   system: "Darwin"
   type: FILE_CHECK_NOT
   description: "Completely disable sharing services (SMB check)"
   info: "Section 2.4.14.14, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 2"
   info: "Context: System"
   file: "smbd.plist"
   search_locations: "/System/Library/LaunchDaemons"
   group: "admin"
</custom_item>

<custom_item>
   system: "Darwin"
   type: FILE_CHECK_NOT
   description: "Completely disable sharing services (Web check)"
   info: "Section 2.4.14.14, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 2"
   info: "Context: System"
   file: "org.apache.httpd.plist"
   search_locations: "/System/Library/LaunchDaemons"
   group: "admin"
</custom_item>

<custom_item>
   system: "Darwin"
   type: FILE_CHECK_NOT
   description: "Completely disable sharing services (Remote Apple Events check)"
   info: "Section 2.4.14.14, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 2"
   info: "Context: System"
   file: "eppc.plist"
   search_locations: "/System/Library/LaunchDaemons"
   group: "admin"
</custom_item>

<custom_item>
   system: "Darwin"
   type: FILE_CHECK_NOT
   description: "Completely disable sharing services (Bonjour check)"
   info: "Section 2.4.14.14, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 2"
   info: "Context: System"
   file: "com.apple.mDNSResponder.plist"
   search_locations: "/System/Library/LaunchDaemons"
   group: "admin"
</custom_item>

<custom_item>
   system: "Darwin"
   type: FILE_CHECK_NOT
   description: "Completely disable sharing services (Bonjour Helper check)"
   info: "Section 2.4.14.14, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 2"
   info: "Context: System"
   file: "com.apple.mDNSResponderHelper.plist"
   search_locations: "/System/Library/LaunchDaemons"
   group: "admin"
</custom_item>
```

```
<custom_item>
   system: "Darwin"
   type: FILE_CHECK_NOT
   description: "Completely disable sharing services (Xgrid agent check)"
   info: "Section 2.4.14.14, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 2"
   info: "Context: System"
   file: "com.apple.xgridagentd.plist"
   search_locations: "/System/Library/LaunchDaemons"
   group: "admin"
</custom_item>

<custom_item>
   system: "Darwin"
   type: FILE_CHECK_NOT
   description: "Completely disable sharing services (Xgrid controller check)"
   info: "Section 2.4.14.14, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 2"
   info: "Context: System"
   file: "com.apple.xgridcontrollerd.plist"
   search_locations: "/System/Library/LaunchDaemons"
   group: "admin"
</custom_item>

<custom_item>
   system: "Darwin"
   type: FILE_CHECK_NOT
   description: "Completely disable sharing services (Internet Sharing check)"
   info: "Section 2.4.14.14, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 2"
   info: "Context: System"
   file: "com.apple.InternetSharing.plist"
   search_locations: "/System/Library/LaunchDaemons"
   group: "admin"
</custom_item>

#
# Manual Check
#  description: "Disable 'Check for updates' for standard users "
#  info: "Section 2.4.15.1, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark, "
#  info: "version 1.0,"
#  info: "Level 1"
#  info: "Context: System"
#

#
# Manual Check
#  description: "Download important updates automatically "
#  info: "Section 2.4.15.2, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark, "
#  info: "version 1.0,"
#  info: "Level 2"
#  info: "Context: System"
#

#
# Policy Item
#  description: "Transfer installer packages from a test-bed computer "
#  info: "Section 2.4.15.3, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark, "
#  info: "version 1.0,"
#  info: "Level 2"
#  info: "Context: System"
#

#
# Manual Check
```

```
#  description: "Change sound input device to Line In "
#  info: "Section 2.4.16.1, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark, "
#  info: "version 1.0,"
#  info: "Level 2"
#  info: "Context: System"
#

#
# Manual Check
#  description: "Minimize input volume for all inputs "
#  info: "Section 2.4.16.2, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark, "
#  info: "version 1.0,"
#  info: "Level 2"
#  info: "Context: System"
#

#
# Manual Check
#  description: "Only enable speech recognition in a secure environment "
#  info: "Section 2.4.17.1, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark, "
#  info: "version 1.0,"
#  info: "Level 1"
#  info: "Context: User"
#

#
# Manual Check
#  description: "Configure Speech Recognition to use a Listening Key "
#  info: "Section 2.4.17.2, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark, "
#  info: "version 1.0,"
#  info: "Level 1"
#  info: "Context: User"
#

#
# Policy Item
#  description: "Use headphones if you enable text to speech, or turn text to speech
      off "
#  info: "Section 2.4.17.3, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark, "
#  info: "version 1.0,"
#  info: "Level 1"
#  info: "Context: User"
#

#
# Policy Item/Manual Check
#  description: "Prevent Spotlight from searching all confidential folders "
#  info: "Section 2.4.18.1, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark, "
#  info: "version 1.0,"
#  info: "Level 1"
#  info: "Context: User and System"
#

#
# Policy Item/Manual Check
#  description: "Prevent Spotlight from searching backup folders or volumes "
#  info: "Section 2.4.18.2, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark, "
#  info: "version 1.0,"
#  info: "Level 1"
#  info: "Context: User and System"
#

</check_type>
```

PART 4

```
# Copyright 2008, R.D. Smith
```

```
#
# Name           : Mac OS X (10.5) CIS Benchmark Level 1 & 2, v1.0
#
#
# Description    : Covers Section 2.5-2.7 and Built-in Checks
#
#
# Notes          : 1. Policy Items are not auditable from Nessus.
#                  2. Manual Check are technical checks that can not be done with
#                     Nessus and must be performed manually.
#
#
# Author         : R.D. Smith, GIAC Certified (GSE, GSNA), CISSP-ISSEP, CISA
# Date           : 20081105
#
# Version        : v0.9
#


<check_type:"Unix">

###############################################################
#                                                             #
# Section 2.5 Data Maintenance and Encryption Action Items #
#                                                             #
###############################################################

#
# Policy Item
#  description: "Backup "
#  info: "Section 2.5.1, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark, "
#  info: "version 1.0,"
#  info: "Level 1"
#  info: "Context: System"
#

<custom_item>
   system: "Darwin"
   type: CMD_EXEC
   description: "Secure Home Folders "
   info: "Section 2.5.2, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 2"
   info: "Context: System"
   info: " "
   info: "NOTE: This check assumes that all home directories are in /Users."
   info: " "
   cmd: "ls -l /Users | perl -nle 'print \"check \"; if (/^d/) {if (!/^drwx------/)
      {s/.*?\\s([a-zA-z]+$)/\\1/; if (!(/Shared/||/total/)) {print \"failed: \".$_.\"
      ;\"}};};' | perl -ne 's/\\r//g;s/\\n//g;print'"
   expect: "^(check )+$"
</custom_item>

<item>
   name: "accounts_without_home_dir"
   description: "This check reports user accounts that do not have home
   directories."
   info: "The normal result is: "
   info: "_cyrus (/var/imap does not exist)"
   info: "_pcastagent (/var/pcast/agent does not exist)"
   info: "_pcastserver (/var/pcast/server does not exist)"
   info: "_teamsserver (/var/teamsserver does not exist)"
   info: ""
   info: ""
   severity: LOW
</item>


#
```

```
# Policy Item
#  description: "Encrypt sensitive files "
#  info: "Section 2.5.3, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark, "
#  info: "version 1.0,"
#  info: "Level 2"
#  info: "Context: User"
#

#
# Policy Item
#  description: "Securely erase files in the Finder "
#  info: "Section 2.5.4, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark, "
#  info: "version 1.0,"
#  info: "Level 1"
#  info: "Context: User"
#

#
# Policy Item
#  description: "Securely erase partitions "
#  info: "Section 2.5.5, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark, "
#  info: "version 1.0,"
#  info: "Level 2"
#  info: "Context: System"
#

#
# Policy Item
#  description: "Securely erase free space "
#  info: "Section 2.5.6, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark, "
#  info: "version 1.0,"
#  info: "Level 2"
#  info: "Context: System"
#

#
# Policy Item
#  description: "Repair disk permissions after installing software or software
      updates "
#  info: "Section 2.5.7, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark, "
#  info: "version 1.0,"
#  info: "Level 1"
#  info: "Context: System"
#

############################################################
#                                                          #
# Section 2.6 Network Services Configuration Action Items  #
#                                                          #
############################################################

<custom_item>
   system: "Darwin"
   type: FILE_CONTENT_CHECK
   description: "Secure Bonjour "
   info: "Section 2.6.1, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 2"
   info: "Context: System"
   file: "com.apple.mDNSResponder.plist"
   search_locations: "/System/Library/LaunchDaemons"
   regex: ".*\<key\>Disabled\<\/key\>.*$"
   expect: ".*\<key\>Disabled\<\/key\>.*"
   severity: MEDIUM
</custom_item>

#
# Policy Item/Manual Check
#  description: "Use an outbound network detection system "
```

```
#  info: "Section 2.6.2, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark, "
#  info: "version 1.0,"
#  info: "Level 2"
#  info: "Context: System"
#

##########################################################
#                                                        #
# Section 2.7 System Integrity Validation Action Items #
#                                                        #
##########################################################

<custom_item>
   system: "Darwin"
   type: FILE_CONTENT_CHECK
   description: "Increase the retention time for system.log and secure.log
      (secure.log check)"
   info: "Section 2.7.1.1, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 2"
   info: "Context: System"
   file: "newsyslog.conf"
   search_locations: "/etc"
   regex: "\/var\/log\/secure.log.*640.*$"
   expect: "\/var\/log\/secure.log.*640  30.*"
</custom_item>

<custom_item>
   system: "Darwin"
   type: FILE_CONTENT_CHECK
   description: "Increase the retention time for system.log and secure.log
      (system.log check)"
   info: "Section 2.7.1.1, CIS Mac OS X 10.5 Leopard Level 1 & 2 Benchmark,"
   info: "version 1.0,"
   info: "Level 2"
   info: "Context: System"
   file: "newsyslog.conf"
   search_locations: "/etc"
   regex: "\/var\/log\/system.log.*640.*$"
   expect: "\/var\/log\/system.log.*640  30.*"
</custom_item>


##################
#                #
# Builtin Checks #
#                #
##################

# Checks that are not customizable are buildin
# into the Unix compliance check module. Given below
# are the list of all the checks are the peformed
# using the biultin functions. Please refer to the
# the Unix cmpliance checks documentation for more
# details about each check.
#


# <item>
# name: "minimum_password_length"
# description : "Minimum password length"
# value : "8..MAX"
# </item>
#
# <item>
# name: "min_password_age"
# description : "Min password age"
# value: "6..21"
# </item>
```

```
<item>
   name: "accounts_bad_home_permissions"
   description : "Account with bad home permissions"
   mode: "0700"
   info: "The normal result is: "
   info: "/var/empty mode: 0755 (should be 0700) owner: root"
   info: "/var/root mode: 0750 (should be 0700) owner: root"
   info: ""
   info: ""
   severity: MEDIUM
</item>

<item>
   name: "invalid_login_shells"
   description: "Accounts with invalid login shells"
</item>

<item>
   name: "login_shells_with_suid"
   description : "Accounts with suid login shells"
   info: "The normal result is: "
   info: "_uucp has a SUID/SGID shell (/usr/sbin/uucico has permissions 4555 - it
      should be 0 "
   info: "755 or stricter)"
   info: ""
   info: ""
   severity: LOW
</item>

<item>
   name: "login_shells_writeable"
   description : "Accounts with writeable shells"
</item>

<item>
   name: "login_shells_bad_owner"
   description : "Shells with bad owner"
   info: "The normal result is: "
   info: "_uucp has a shell which is not owned by root/bin (/usr/sbin/uucico belongs
      to _uucp)"
   info: ""
   info: ""
   severity: LOW
</item>

<item>
   name: "passwd_file_consistency"
   description : "Check passwd file consistency"
</item>

<item>
   name: "passwd_zero_uid"
   description : "Check zero UID account in /etc/passwd"
</item>

<item>
   name : "passwd_duplicate_uid"
   description : "Check duplicate accouts in /etc/passwd"
</item>

<item>
   name : "passwd_duplicate_gid"
   description : "Check duplicate gid in /etc/passwd"
   info: "The normal result is: "
   info: "_installer _update_sharing nobody (sharing GID -2)"
   info: ""
   info: ""
   severity: LOW
```

```
</item>

<item>
   name : "passwd_duplicate_username"
   description : "Check duplicate username in /etc/passwd"
</item>

<item>
   name : "passwd_duplicate_home"
   description : "Check duplicate home in /etc/passwd"
</item>

<item>
   name : "passwd_shadowed"
   description : "Check every passwd is shadowed in /etc/passwd"
</item>

<item>
   name: "passwd_invalid_gid"
   description : "Check every GID in /etc/passwd resides in /etc/group"
   info: "The normal result is: "
   info: "_eppc (invalid GID of 71)"
   info: ""
   info: ""
   severity: LOW
</item>

<item>
   name : "group_file_consistency"
   description : "Check /etc/group file consistency"
</item>

<item>
   name: "group_zero_gid"
   description : "Check zero GUID in /etc/group"
</item>

<item>
   name: "group_duplicate_name"
   description : "Check duplicate group names in /etc/group"
</item>

<item>
   name: "group_duplicate_gid"
   description : "Check duplicate gid in /etc/group"
</item>

<item>
   name : "group_duplicate_members"
   description : "Check duplicate members in /etc/group"
</item>

<item>
   name: "group_nonexistant_users"
   description : "Check for non existent users in /etc/group"
</item>

</check_type>
```