# Security Tools - A “Try Before You Buy” Web-Based Approach

Sheila Frankel
National Institute of Standards and Technology
NIST North, Room 426
Gaithersburg, MD 20899

sfrankel@nist.gov

## ABSTRACT

For users who peruse the World Wide Web to locate potentially useful security-related tools, the typical information contained in most sites - a pointer to the tool and, optionally, a short abstract - is not enough to determine whether this tool will fulfill the required criteria. NIST has initiated a Web site that enables users to view more detailed tool descriptions, sample input and output, and, in some cases, to conduct a real or simulated tool demonstration.

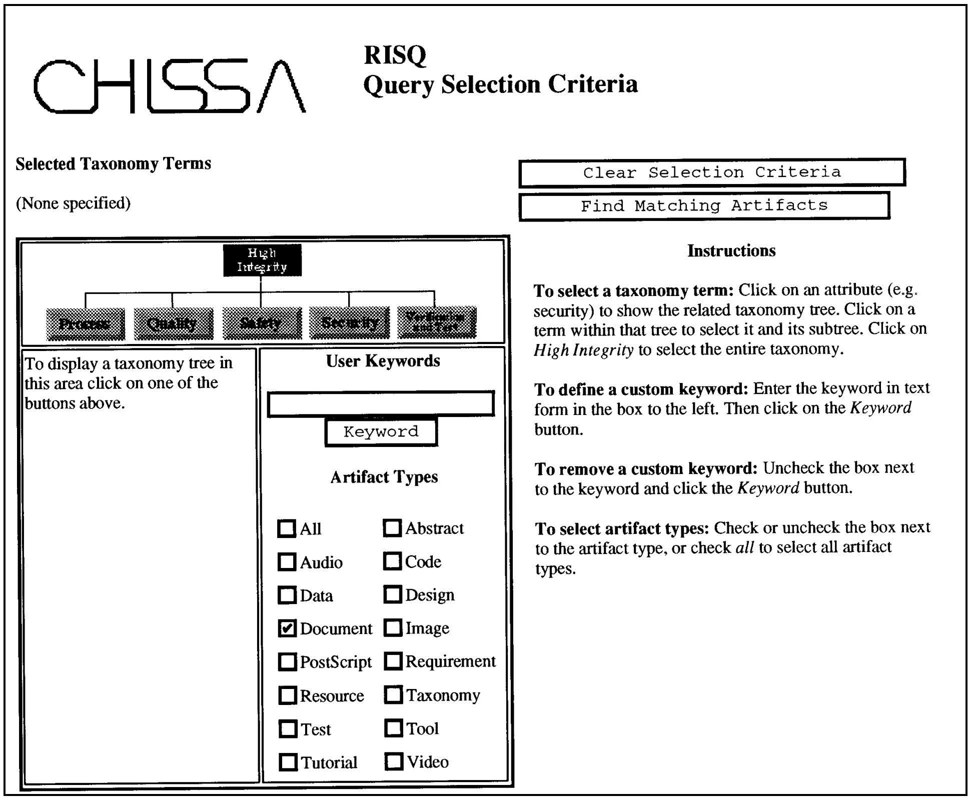
KEYWORDS: Security tools.

## Introduction

Many Web sites currently contain lists of security-related tools. Most of these sites consist of an undifferentiated list of tools, where each tool entry contains a pointer enabling the user to download the tool and, possibly, a short abstract. This is sufficiently useful for a user who knows exactly what tools he/she wants and is simply looking for a site from which to download the tool. However, it can be somewhat overwhelming for the user who is trying to discover what is available in this arena. Some sites go one step further, and sort the tools list according to functional capabilities. This does add some organization to the chaos, but it still requires the user to choose one tool, out of perhaps 20 or more, on the basis of a terse abstract.

Of course, the user can just download the tool and try it out. However, this is not always so simple for Unix tools. Most of these tools require root-level access for the installation process, and many of them demand a fair amount of tinkering and tweaking before they will run. If a user is shopping for a variety of security tools, it can take a substantial amount of time and effort to unearth an assortment of tools that turn out, after all that work, not to be exactly what he/she had in mind.

The Center for High Integrity Software Systems Assurance (CHISSA) at NIST recently initiated a Web Site that has the capacity to address this problem. The site contains a database of “artifacts” that includes documents (papers and abstracts), audio/video performances, and interactive tools demonstrations. Figure 1 shows the RISQ (Reference Information for Software Quality)[1] introductory screen, displayed when the user points his/her browser to the URL **http://hissa.ncsl.nist.gov/cgi-bin/risq.pl**. As its name might indicate, the domain of interest is broader than security, covering the whole landscape of high integrity software, but one branch of the high integrity “tree” is that of security. The Computer Security Division of NIST plans to add a large variety of security tools to the RISQ database.



**RISQ**
**Query Selection Criteria**

**Selected Taxonomy Terms**

(None specified)

To display a taxonomy tree in this area click on one of the buttons above.

**User Keywords**

**Artifact Types**

□ All
□ Audio
□ Data
☑ Document
□ PostScript
□ Resource
□ Test
□ Tutorial
□ Abstract
□ Code
□ Design
□ Image
□ Requirement
□ Taxonomy
□ Tool
□ Video

**Instructions**

**To select a taxonomy term:** Click on an attribute (e.g. security) to show the related taxonomy tree. Click on a term within that tree to select it and its subtree. Click on *High Integrity* to select the entire taxonomy.

**To define a custom keyword:** Enter the keyword in text form in the box to the left. Then click on the *Keyword* button.

**To remove a custom keyword:** Uncheck the box next to the keyword and click the *Keyword* button.

**To select artifact types:** Check or uncheck the box next to the artifact type, or check *all* to select all artifact types.

**Figure 1 - RISQ Introductory Screen**

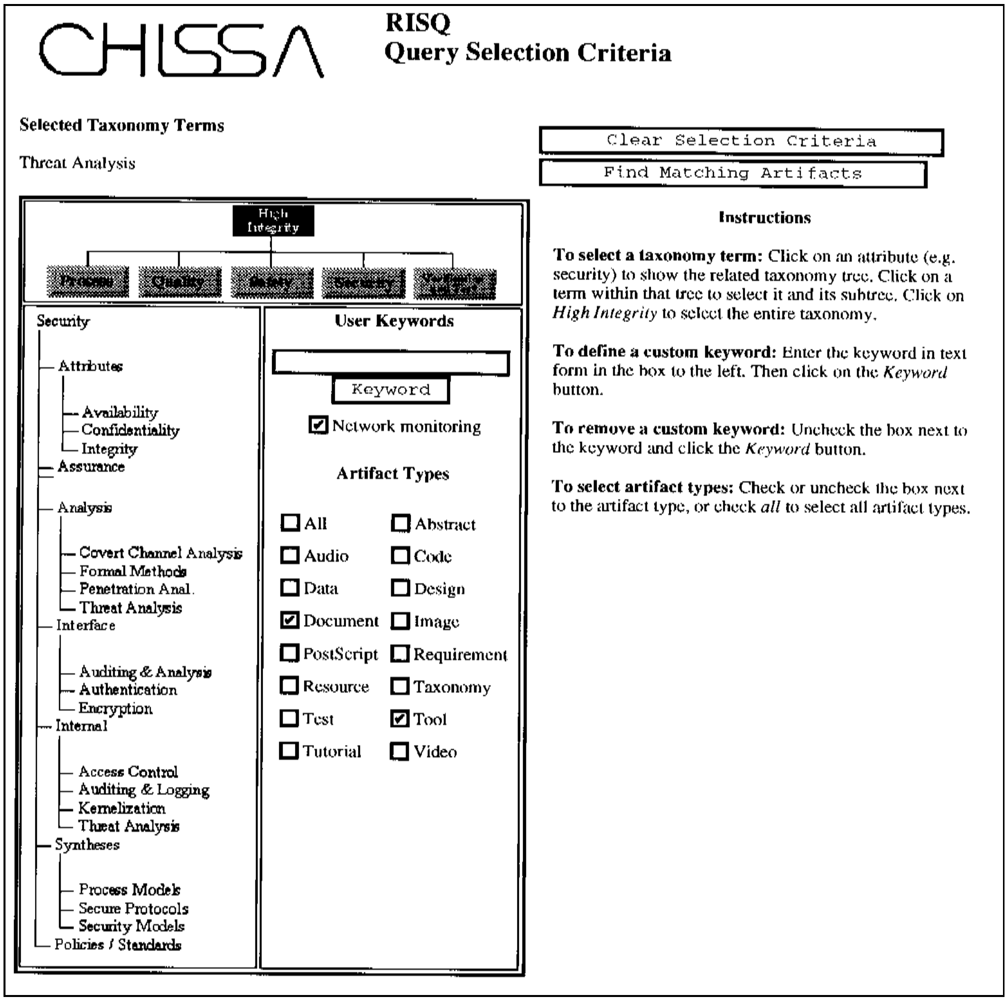
## What Types of Security Tools Are Available?

The array of public-domain, freely-available security tools covers a wide variety of functionality. Some security tools aid in the prevention of potential security problems by detecting, in advance, problematic system or network configurations. In particular, there are tools that scan all of a system's user passwords, reporting on those that can be easily guessed, or "cracked"; other tools offer alternatives to password-related user authentication, supplying advanced authentication methods such as one-time passwords. Other preventive tools analyze user and system files, reporting on dangerous permissions, insecure file relationships, and the use of system features, files, or protocols that can be easily subverted. A particularly aggressive form of preventive security tool is the "real-time penetration" class of tools, that actively probe a system for known vulnerabilities.

Another class of security tools can detect system intrusions by monitoring, logging, or auditing the progress and output of potentially problematic connections or processes. When normal behavior is documented and recorded, unexpected changes in such things as file sizes, program output, user profiles, or other system behavior can alert the system administrator that an intrusion has occurred.
]

Cryptographic tools enable users to ensure the confidentiality of files and transmissions through the use of encryption, and to authenticate the identity of one or both parties engaged in an electronic transaction.

In the past, systems and network security were the domain of professional systems administrators, and the majority of security tools ran on Unix machines. The largest number of freely-available, public domain security tools today still are Unix tools. However, with the proliferation of desktop PC's and the advent of Windows NT servers, more and more systems and networks are administered by individual users who are not expert in the domain of system administration. There are currently a number of freeware tools available for DOS, Windows, and the Macintosh, and undoubtedly more will be offered in the near future. The RISQ security tools database will undoubtedly be heavily weighted on the Unix side, but as non-proprietary public domain security tools for PC's become available, they will be added to the database.



**RISQ**
**Query Selection Criteria**

**Selected Taxonomy Terms**

Threat Analysis

**User Keywords**

☑ Network monitoring

**Artifact Types**

☐ All ☐ Abstract
☐ Audio ☐ Code
☐ Data ☐ Design
☑ Document ☐ Image
☐ PostScript ☐ Requirement
☐ Resource ☐ Taxonomy
☐ Test ☑ Tool
☐ Tutorial ☐ Video

**Instructions**

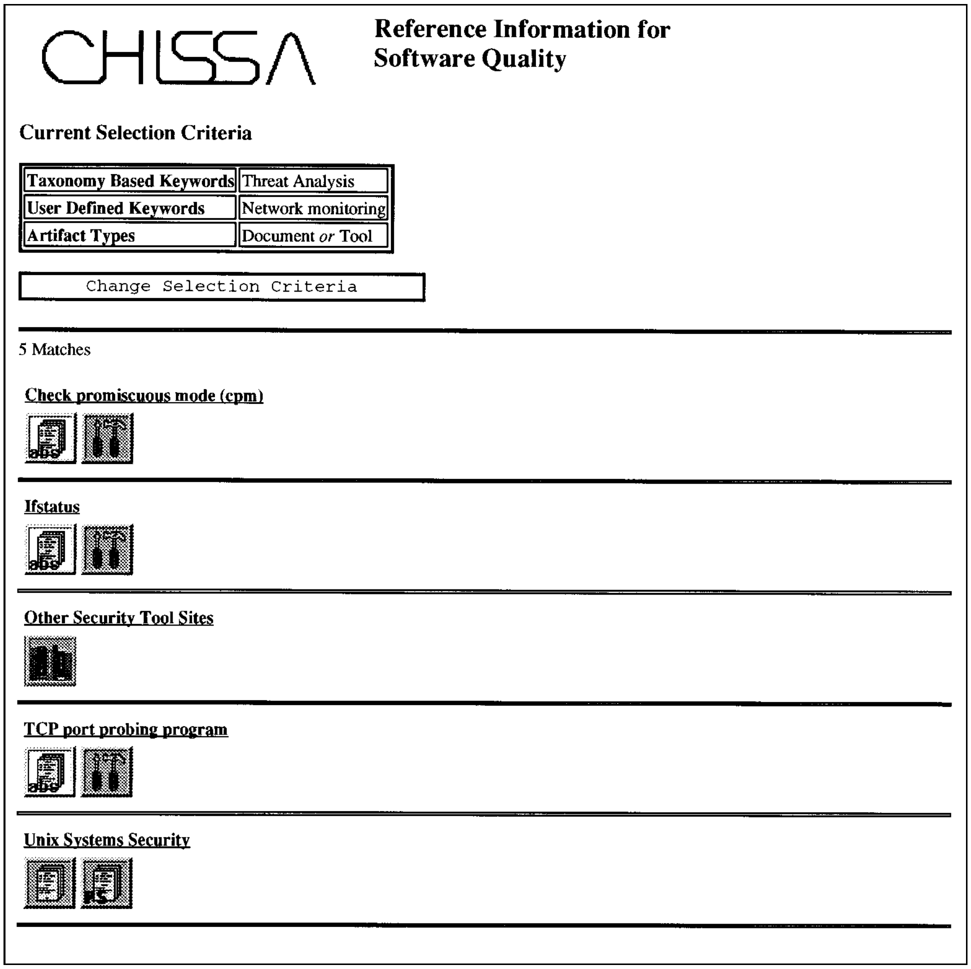
**To select a taxonomy term:** Click on an attribute (e.g. security) to show the related taxonomy tree. Click on a term within that tree to select it and its subtree. Click on *High Integrity* to select the entire taxonomy.

**To define a custom keyword:** Enter the keyword in text form in the box to the left. Then click on the *Keyword* button.

**To remove a custom keyword:** Uncheck the box next to the keyword and click the *Keyword* button.

**To select artifact types:** Check or uncheck the box next to the artifact type, or check *all* to select all artifact types.

**Figure 2 - RISQ Search**



CHISSA

Reference Information for Software Quality

Current Selection Criteria

| | |
|---|---|
| Taxonomy Based Keywords | Threat Analysis |
| User Defined Keywords | Network monitoring |
| Artifact Types | Document *or* Tool |

5 Matches

Check promiscuous mode (cpm)

Ifstatus

Other Security Tool Sites

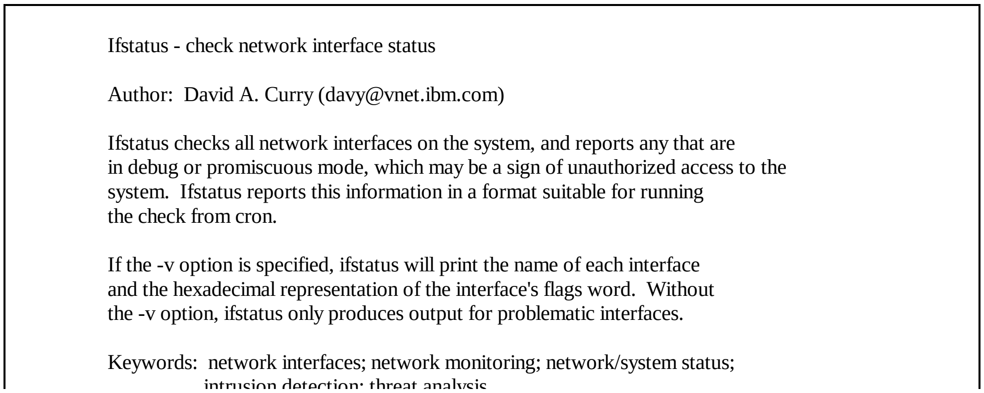
TCP port probing program

Unix Systems Security

**Figure 3 - RISQ Search Result**

## Using the RISQ Database

When using the RISQ database, the user can search either the whole high integrity infrastructure or one of its branches (process, quality, safety, security, and verification/test) by specifying one or more of the three classes of searches. He/she can click on any tree or sub-tree in the taxonomy, enter user-specified keywords, and/or specify which type of artifacts to search for. Thus, if a user is searching for information and tools related to security threat analysis, with the specific goal of monitoring network connections for the purpose of intrusion detection, he/she could click on the "threat analysis" branch of the security tree, enter the user-specified keyword "network monitoring," and click on the artifact types "document" and "tool." Figure 2 shows the RISQ screen set up to perform this search, and Figure 3 illustrates the possible outcome of such a search. In this case, five matching database entries were found: three of them (cpm, ifstatus, and the TCP port probing program) are tools, each having 2 types of associated database artifacts: an abstract and a tool. Figure 4 illustrates the format of a sample tool abstract, and the types of

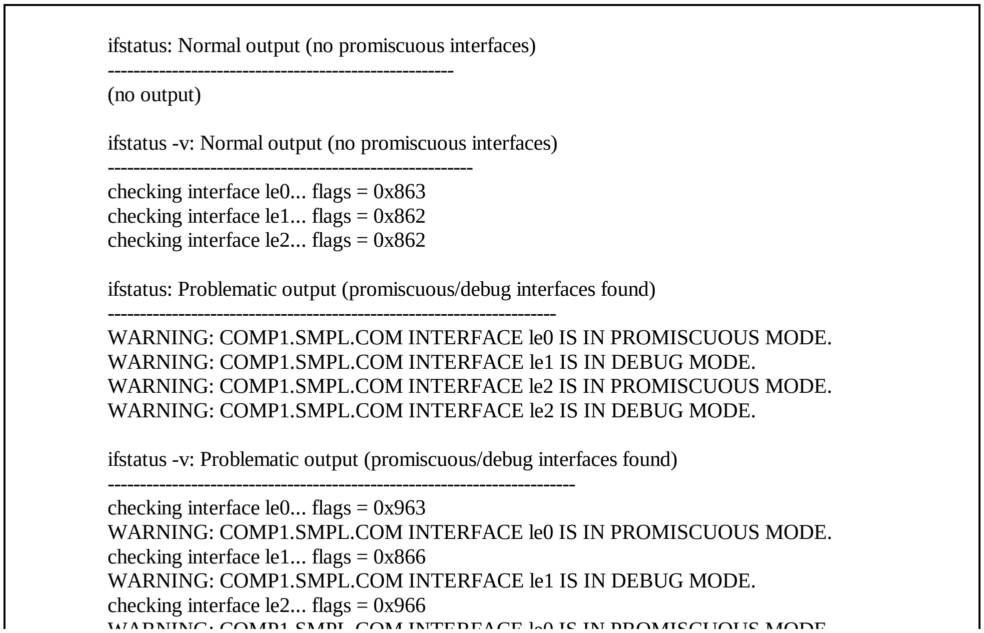


```
Ifstatus - check network interface status

Author:  David A. Curry (davy@vnet.ibm.com)

Ifstatus checks all network interfaces on the system, and reports any that are
in debug or promiscuous mode, which may be a sign of unauthorized access to the
system.  Ifstatus reports this information in a format suitable for running
the check from cron.

If the -v option is specified, ifstatus will print the name of each interface
and the hexadecimal representation of the interface's flags word.  Without
the -v option, ifstatus only produces output for problematic interfaces.

Keywords:  network interfaces; network monitoring; network/system status;
           intrusion detection; threat analysis
```

**Figure 4 - Sample Tool Abstract**

sample tool output that are available will be demonstrated in subsequent parts of this article. The third match (Other Security Tool Sites) is a list of pointers to sites with further information, and the fifth match (Unix Systems Security) is an article that can be retrieved in either text or postscript format.

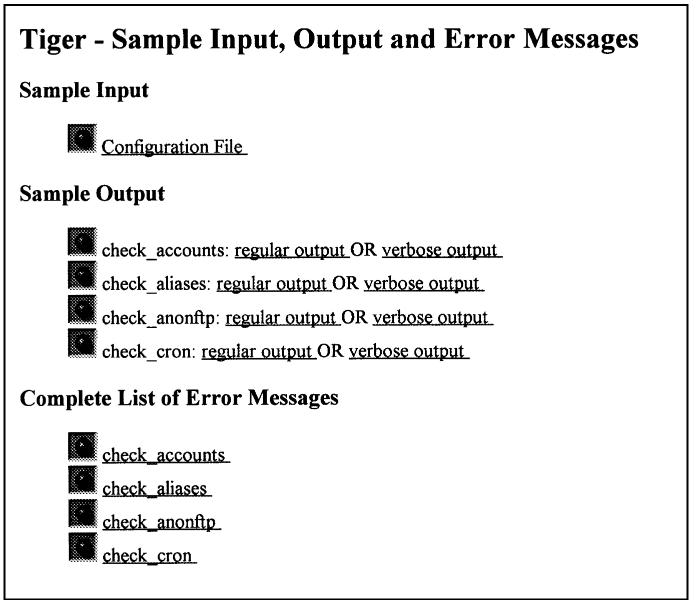
Most security tools are designed to be run repeatedly, often in the background at pre-specified times of the day or week. Thus, most do not have an elaborate graphical user interface; quite the opposite, the tools are usually launched through an unadorned Unix command-line interface, with variations in the configuration or output of the tool being specified either through a configuration file or via Unix command-line options.



```
ifstatus: Normal output (no promiscuous interfaces)
-----------------------------------------------------
(no output)

ifstatus -v: Normal output (no promiscuous interfaces)
--------------------------------------------------------
checking interface le0... flags = 0x863
checking interface le1... flags = 0x862
checking interface le2... flags = 0x862

ifstatus: Problematic output (promiscuous/debug interfaces found)
----------------------------------------------------------------------
WARNING: COMP1.SMPL.COM INTERFACE le0 IS IN PROMISCUOUS MODE.
WARNING: COMP1.SMPL.COM INTERFACE le1 IS IN DEBUG MODE.
WARNING: COMP1.SMPL.COM INTERFACE le2 IS IN PROMISCUOUS MODE.
WARNING: COMP1.SMPL.COM INTERFACE le2 IS IN DEBUG MODE.

ifstatus -v: Problematic output (promiscuous/debug interfaces found)
-------------------------------------------------------------------------
checking interface le0... flags = 0x963
WARNING: COMP1.SMPL.COM INTERFACE le0 IS IN PROMISCUOUS MODE.
checking interface le1... flags = 0x866
WARNING: COMP1.SMPL.COM INTERFACE le1 IS IN DEBUG MODE.
checking interface le2... flags = 0x966
WARNING: COMP1.SMPL.COM INTERFACE le0 IS IN PROMISCUOUS MODE.
```

**Figure 5 - Sample Tool Output (ifstatus)**

For those security tools that have a straightforward command-line interface with few options and a limited number of output modes, the RISQ system will offer the user the option of viewing sample output for most or all of the cases. For example, ifstatus[2] is a tool that tests whether one or more of a system's interfaces is running in promiscuous mode. Normally, applications that are running on a system can only view network packets that are destined for that particular system. If one of a system's interfaces is running in promiscuous mode, it allows applications running on the system to view all packets passing over that interface, not just those packets that were sent to that particular system. Turning on the "promiscuous mode" flag is a common device used by intruders who want to monitor passwords and other sensitive information that is being sent over the network. Ifstatus has 2 modes, regular and verbose, and 2 possible outcomes for each system interface: promiscuous or non-promiscuous. For a tool such as ifstatus, the RISQ system will offer the user the opportunity to view sample output for all possible cases. Figure 5 displays this sample output.



**Tiger - Sample Input, Output and Error Messages**

**Sample Input**

● Configuration File

**Sample Output**

● check_accounts: regular output OR verbose output
● check_aliases: regular output OR verbose output
● check_anonftp: regular output OR verbose output
● check_cron: regular output OR verbose output

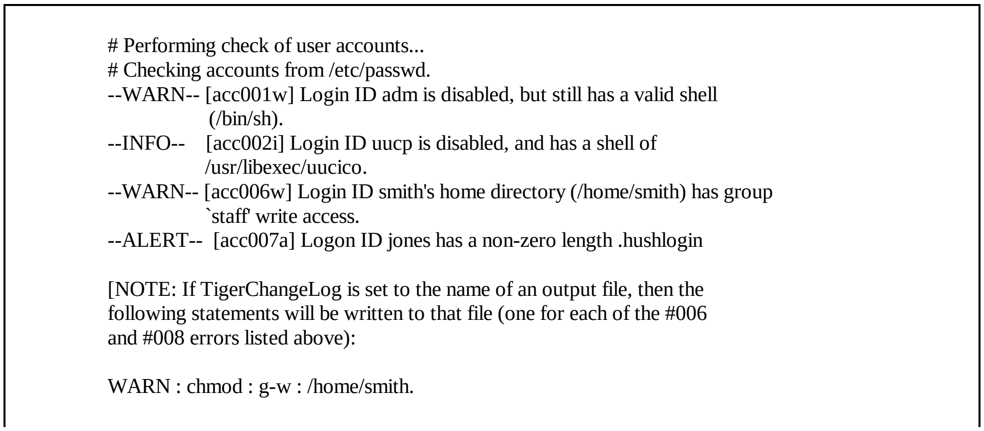
**Complete List of Error Messages**

● check_accounts
● check_aliases
● check_anonftp
● check_cron

**Figure 6 - Sample Menu (Tiger)**

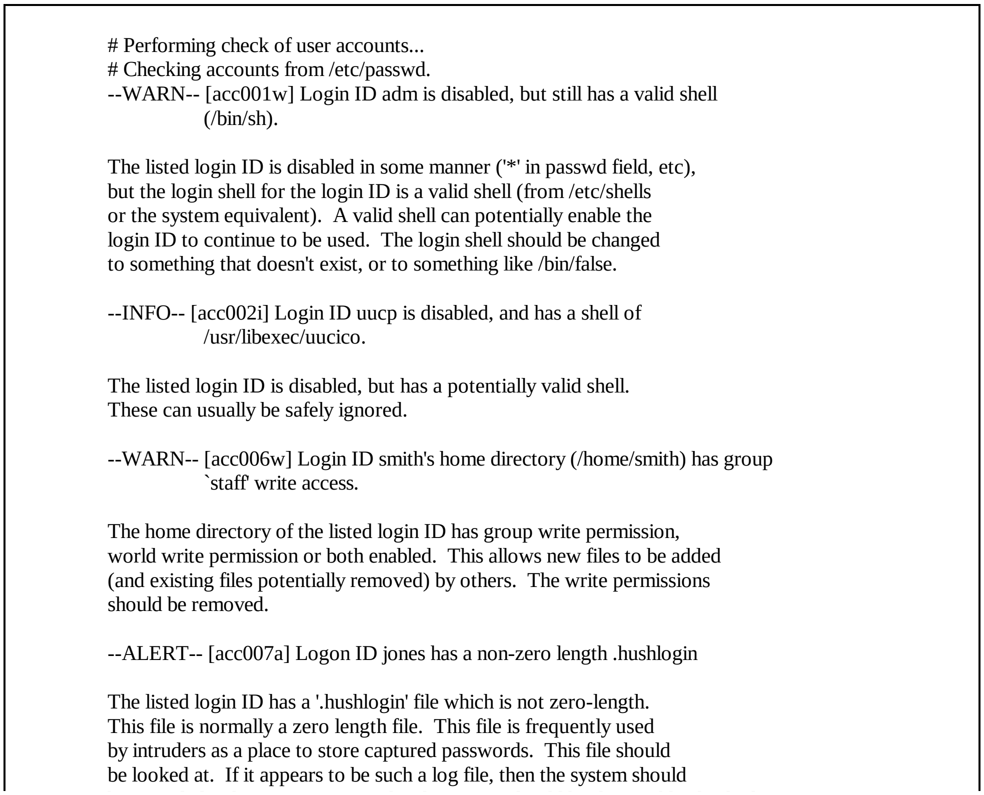
An example of a security tool with multiple input configurations and a prodigious amount of potential output is tiger[3] . Tiger consists of a set of scripts that scan a Unix system looking for potential security problems and vulnerabilities. The scripts, which check for such problems as incorrect file and directory access permissions, can be run either individually or all at once, and can also be run either immediately or on pre-specified dates and times. The operation of the tiger scripts is controlled by a small set of options, some of which determine the level of verbosity of the error messages, and by a configuration file. The configuration file sets environment variables which dictate such details as: the depth of search within the file tree, the types of files to be checked, which checks should be performed, whether informational warnings should be issued, which users and groups are entitled to exercise "root" privileges, etc. The RISQ system will enable the user to peruse a sample configuration file, sample error messages issued by each of the scripts, and/or a complete list of all possible error messages that can be issued by each script. Figure 6 shows (for 4 representative tiger scripts) the menu of tiger-related information presented to the user.

One of the tiger scripts, check_accounts, checks all user accounts, both those that are currently active and those that have been disabled, for certain types of anomalies. It checks the account's home directory, the presence of several types of special-purpose files that can easily



```
# Performing check of user accounts...
# Checking accounts from /etc/passwd.
--WARN-- [acc001w] Login ID adm is disabled, but still has a valid shell
          (/bin/sh).
--INFO--   [acc002i] Login ID uucp is disabled, and has a shell of
          /usr/libexec/uucico.
--WARN-- [acc006w] Login ID smith's home directory (/home/smith) has group
          `staff' write access.
--ALERT--  [acc007a] Logon ID jones has a non-zero length .hushlogin

[NOTE: If TigerChangeLog is set to the name of an output file, then the
following statements will be written to that file (one for each of the #006
and #008 errors listed above):

WARN : chmod : g-w : /home/smith.
```

**Figure 7 - Sample Tool Output (Tiger - normal mode)**

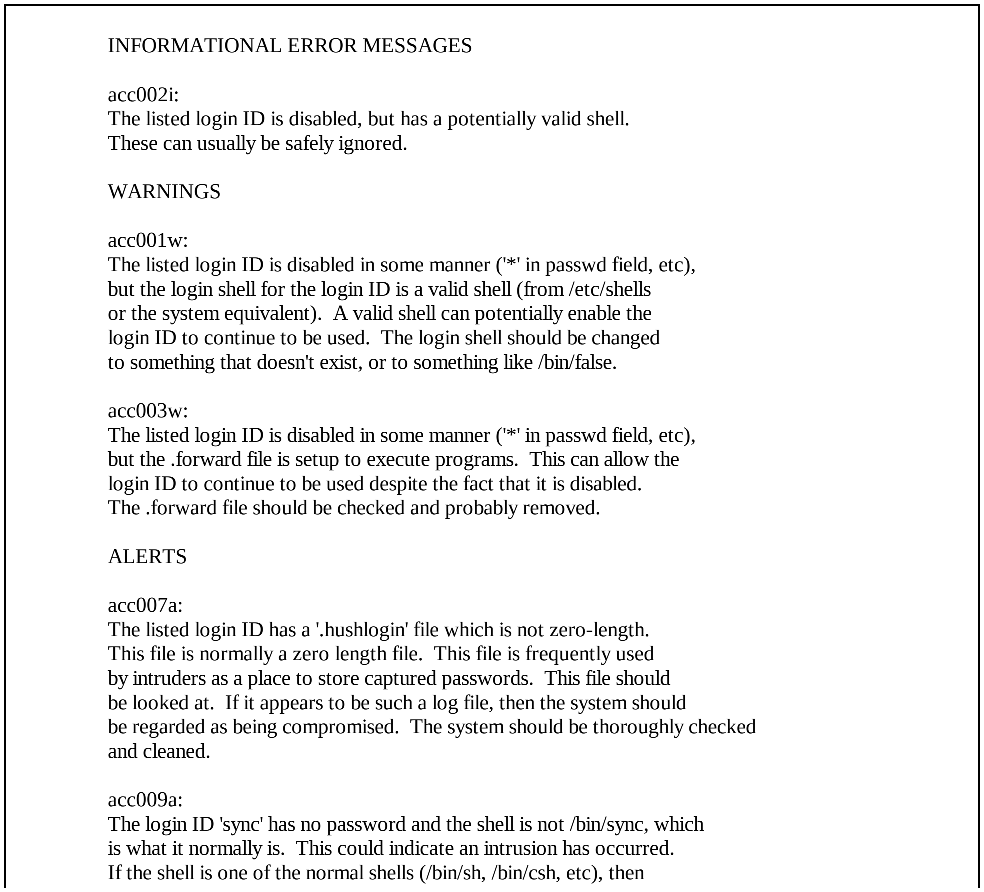


```
# Performing check of user accounts...
# Checking accounts from /etc/passwd.
--WARN-- [acc001w] Login ID adm is disabled, but still has a valid shell
         (/bin/sh).

The listed login ID is disabled in some manner ('*' in passwd field, etc),
but the login shell for the login ID is a valid shell (from /etc/shells
or the system equivalent).  A valid shell can potentially enable the
login ID to continue to be used.  The login shell should be changed
to something that doesn't exist, or to something like /bin/false.

--INFO-- [acc002i] Login ID uucp is disabled, and has a shell of
         /usr/libexec/uucico.

The listed login ID is disabled, but has a potentially valid shell.
These can usually be safely ignored.

--WARN-- [acc006w] Login ID smith's home directory (/home/smith) has group
         `staff' write access.

The home directory of the listed login ID has group write permission,
world write permission or both enabled.  This allows new files to be added
(and existing files potentially removed) by others.  The write permissions
should be removed.

--ALERT-- [acc007a] Logon ID jones has a non-zero length .hushlogin

The listed login ID has a '.hushlogin' file which is not zero-length.
This file is normally a zero length file.  This file is frequently used
by intruders as a place to store captured passwords.  This file should
be looked at.  If it appears to be such a log file, then the system should
```

**Figure 8 - Sample Tool Output (Tiger - verbose mode)**

be compromised, and unsafe user ID/password combinations. Figures 7 and 8 illustrate sample output (regular and verbose, respectively) for check_accounts; and Figure 9 shows the format used to display the complete list of error messages, also using the check_accounts script.

For tools that have a more elaborate interface, and more numerous execution paths and/or options, the RISQ database can, under certain circumstances, provide an actual demonstration of the tool. This is only possible for users who access the database from an X terminal; for other users, a simulated, canned demonstration is provided. The less interactive mode of demonstration described in the preceding paragraphs will suffice for the security tools that are currently intended to be added to the RISQ database; no interactive security tool demonstrations are planned, although a number of other types of tools can be interactively executed in this manner.



```
INFORMATIONAL ERROR MESSAGES

acc002i:
The listed login ID is disabled, but has a potentially valid shell.
These can usually be safely ignored.

WARNINGS

acc001w:
The listed login ID is disabled in some manner ('*' in passwd field, etc),
but the login shell for the login ID is a valid shell (from /etc/shells
or the system equivalent).  A valid shell can potentially enable the
login ID to continue to be used.  The login shell should be changed
to something that doesn't exist, or to something like /bin/false.

acc003w:
The listed login ID is disabled in some manner ('*' in passwd field, etc),
but the .forward file is setup to execute programs.  This can allow the
login ID to continue to be used despite the fact that it is disabled.
The .forward file should be checked and probably removed.

ALERTS

acc007a:
The listed login ID has a '.hushlogin' file which is not zero-length.
This file is normally a zero length file.  This file is frequently used
by intruders as a place to store captured passwords.  This file should
be looked at.  If it appears to be such a log file, then the system should
be regarded as being compromised.  The system should be thoroughly checked
and cleaned.

acc009a:
The login ID 'sync' has no password and the shell is not /bin/sync, which
is what it normally is.  This could indicate an intrusion has occurred.
If the shell is one of the normal shells (/bin/sh, /bin/csh, etc), then
```

**Figure 9 - Sample Tool Output (Tiger - error messages**)

## Conclusion

The Security tools area of the RISQ database is currently in the planning stage, and user input is actively solicited. We welcome comments on all aspects of the database's format and content, including the following:

- Types of tools (or specific tools) that should be added to the database
- Types of information and data to include in the database
- Additional keywords and taxonomy terms
- Most and least useful aspects of the database

The RISQ database should be an excellent vehicle for the "Try Before You Buy" approach to investigating security tools. Hopefully, the combination of tool abstracts and other useful information; sample tool input, output, and error messages; and, where useful, interactive or simulated tool demonstrations will be of value to users. The goal is to enable users to pre-screen security tools, in order to better differentiate between those tools that can meet the user's pre-determined criteria and those that will not.

# References

1. Weinstock, Charles B. and Dolores R. Wallace, RISQ: A Web-Based Tool for Referencing Information on Software Quality, NISTIR 5954, January 1997.

2. Curry, David A. (IBM Internet Emergency Response Service), Ifstatus, http://www.ers.ibm.com/~davy/software/ifstatus.html.

3. Schales, Douglas Lee et. al. (Texas A&M University), Tiger, ftp://net.tamu.edu/pub/security/TAMU.