SECURITY PATROL

by Vincent LeVeque

Self-validation

QUESTION: We are setting up an internal Web server (i.e., an intranet) on our AS/400. While researching how to validate our users, I came across something called a "validation list." Can you explain this? Can we use it in our other applications?

ANSWER: Validation lists are a specific type of AS/400 object (type *VLDL) and are new with OS/400 V4R1. Validation lists were intended to aid in authenticating Internet users accessing Web sites but are versatile objects with much broader use.

Validation lists validate Web users who require authentication to access private Web sites, where issuing a user profile is not desired or possible. Specific examples of Web sites that might use validation lists include subscription-only news services, online catalogs, or Web-based email services. You want to allow these users access to certain Web pages, but you do not want to provide them with more general access to AS/400 functions, which a user profile implies. A user authenticated via a validation list cannot use this ID and password to gain Telnet or FTP access, for example.

AS/400 Web serving software enables the use of validation lists through server protection directives. When a Web user attempts to access a protected page, a prompt for an ID and a password is presented. Server directives are specialized instructions for the IBM HTTP server; they are added to the HTTP server's configuration file, which governs server functioning. The following are specific protection directives relevant to validation-list user authentication:

- Protection setup—defines protection
- Server ID—identifies scope of protection
- Password file—specifies exactly how users are to be identified (via user profile or validation list)

Directives are stored in a configuration object. The actual directives contained in this object are maintained through the AS/400 Web server's browser management interface or, for us old-timers, via the green-screen command Work with HTTP Configuration (WRKHTTPCFG). Figure 1 shows a WRKHTTPCFG screen (unfortunately without protection directives). Other AS/400 TCP/IP functions—specifically, Serial Line Internet Protocol (SLIP) and Point-to-Point Protocol (PPP)—use validation lists for authentication for remote access.

Use of validation lists is not restricted to AS/400 TCP/IP functions. Validation lists are a very good way to authenticate users to your own applications beyond the sign-on authentication provided by the user profile. Sometimes, programs such as kiosk-based information software run under a generic identity and require a more specific identity to perform certain functions.

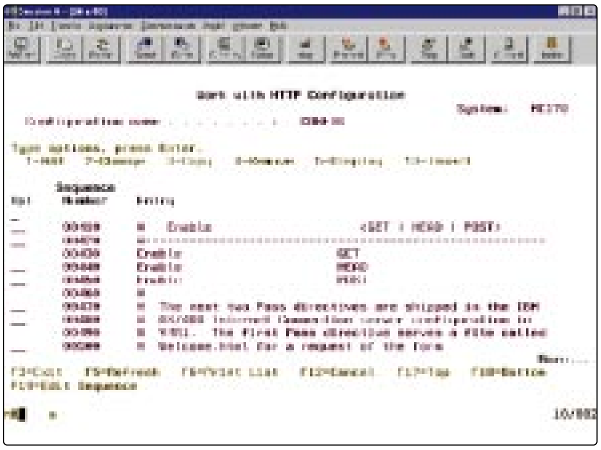
Many programs currently handle user authentication by setting up a database file of user names and passwords, against which the validating program checks. Validation lists can perform these same functions, but with several added advantages:

- The passwords are stored in encrypted format. Should someone "dump" the validation list data, that person would not obtain the original passwords. Only the original, clear-text passwords can be used to authenticate users; the encrypted versions are useless for this purpose.
- Validation lists are stored in protected system domain storage. They can be accessed only via authorized OS/400 commands and supported APIs. The internal content of a validation list cannot be otherwise modified or tampered with. Database files of passwords, by contrast, can be directly accessed by someone with the appropriate authority to view or modify their contents. Even if the passwords were encrypted, you could defeat the security by arbitrarily changing passwords (by, say, cutting and pasting a known password into a powerful user ID).
- Validation lists are "index" objects. Looking up a user as an entry in an index is much faster than with the flat files used in some systems (such as the UNIX /etc/passwd file).

To use validation lists in your application, you must first create the *VLDS object with the OS/400 command Create Validation List (CRTVLDL). Deleting a validation list can be done with Delete Validation List (DLTVLDL). A number of APIs are available that allow you to use validation lists in your own program:

- Add Validation List Entry (QsysAddValidationLstEntry) allows your program to add a user and that user's associated password to the validation list.
- Change Validation List Entry (QsysAddValidationLstEntry) allows your program to change an entry in a validation list object.
- Find First Validation List Entry (QsysFindFirstValidationLstEntry) allows your program to obtain information on the validation list's first entry.
- Find Next Validation List Entry (QsysFindNextValidationLstEntry) allows your program to obtain the validation list entry after the specified Entry_ID parameter.
- Find Validation List Entry (QsysFindValidationLstEntry) allows your program to obtain information about a specific validation list entry.
- Remove Validation List Entry (QsysRemoveValidationLstEntry) allows your program to remove an entry from the validation list.
- Verify Validation List Entry (QsysVerifyValidationLstEntry) allows your program to test the validity of the ID and password entered by comparing these values to those in the appropriate validation list entry.

If you are programming in C, be sure to include the qsyvldl.h header file, as in this directive:



```
                    Work with HTTP Configuration
                                                     System:   [illegible]
 Configuration name . . . . . . . . :   [illegible]

Type options, press Enter.
  1=Add   2=Change   3=Copy   4=Remove   5=Display   13=Insert

     Sequence
Opt   Number    Entry
 _    00010     #  Enable                     <GET | HEAD | POST>
 _    00020     #-------------------------------------------------
 _    00030     Enable                   GET
 _    00040     Enable                   HEAD
 _    00050     Enable                   POST
 _    00060     #
 _    00070     #  The next two Pass directives are shipped in the IBM
 _    00080     #  AS/400 Internet Connection server configuration in
 _    00090     #  V3R1.  The first Pass directive serves a file called
 _    00100     #  Welcome.html for a request of the form
                                                                 More...
F3=Exit   F5=Refresh   F6=Print List   F12=Cancel   F17=Top   F18=Bottom
F19=Edit Sequence
```

Figure 1: **Work with HTTP Configuration (WRKHTTPCFG) maintains directives contained in the configuration object.**

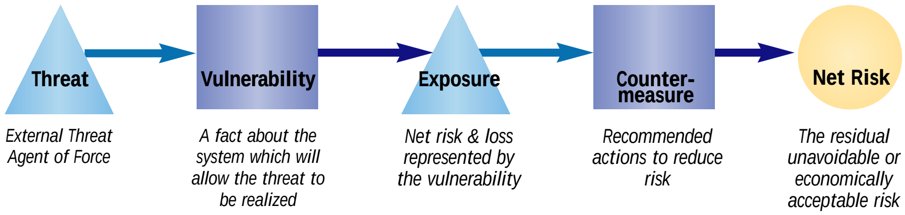

Threat
External Threat Agent of Force
Vulnerability
A fact about the system which will allow the threat to be realized
Exposure
Net risk & loss represented by the vulnerability
Counter-measure
Recommended actions to reduce risk
Net Risk
The residual unavoidable or economically acceptable risk

Figure 2: **Risk analysis examines threats, vulnerabilities, exposures, potential losses, and countermeasures.**

```
#include <qsyvldl.h>
```

Chapter 73 of the *AS/400 System API Reference V4R4* covers validation list APIs in detail.

If you are using validation lists, you need to be careful with system value QRETSVRSEC. If this value is 1, it permits storage of decryptable passwords in the validation list, which should be enabled only if there is a specific requirement for it. For example, SLIP and PPP will not function properly unless they can send decrypted passwords for authentication. Absent having one of these conditions, it is definitely safer to set QRETSVRSEC to 0.

Risky Business

QUESTION: I understand that a good security program should start with a risk analysis, but I'm having a hard time finding a good format for a risk analysis and an even harder time finding some solid numbers. Can you help?

ANSWER: Risk analysis typically looks at threats, vulnerabilities, exposures, potential losses, and countermeasures. You try to estimate how at risk your business's essential information is to various threats, how much you stand to lose, and, given that, what countermeasures are cost-effective. Figure 2 illustrates this basic model.

Threats can be outside hackers out for the thrill of cracking your system, professional criminals set to steal from you, dishonest or disgruntled employees, or even unethical vendors or customers. A vulnerability is some problem with your security that allows the intruder in (e.g., an unlocked door, a default password, or a misconfigured Web server). Exposure is how much you stand to lose if a threat is realized. What does it cost your business to have a defaced Web site or a mail server taken over as a Spam relay? What is the impact of financial fraud caused by tampering with data and programs? These are scenarios that you should present to your management. Finally, countermeasures are the steps you should take to mitigate or eliminate potential loss. With luck, your management will be convinced of your logic and provide you resources to implement the countermeasures. Countermeasures are things such as new locks on doors; a program to change all default passwords; or a standard, secure Web site configuration. Countermeasures can reduce your risk but rarely eliminate it completely. There is generally some residual risk that you should estimate and whose acceptability your management should determine.

A number of published surveys attempt to determine risk levels by polling security managers on threats they observe in their own organizations. The two most accessible surveys are the Computer Security Institute/FBI study and the Price Waterhouse Coopers/*Information Week* study. These both involve well-designed surveys that attempt to judge risks and common security countermeasures as they exist in a variety of organizations. The latest Computer Security Institute/FBI study can be found at *www.gocsi.com/prelea990301.htm*; the latest Price Waterhouse Cooper/Information Week study, at *www.informationweek.com/743/security.htm*.

The best way to ascertain your risk, however, is to start keeping your own numbers. Review firewall audit logs for outside scans or other hacking attempts. Keep good records of losses or inconveniences caused by inadequate security. Remember that the most credible numbers to your management are those of their own business.

Vulnerabilities can be assessed through an audit conducted internally by either your own staff or an outside consultant. A good vulnera-

bility study looks at the technical configuration of all significant systems, administrative practices, organizationwide policies, and general management controls.

Exposure or possible losses due to inadequate security require two estimates regarding your information's data and systems. The first estimate is of the value of your existing data and systems to the organization. The second is the extent to which this value would be affected by a security breach. What if information were improperly disclosed? What if a critical system were damaged beyond repair? What is some large-scale theft resulted from the manipulation of information? To find these answers, you need a very good understanding of your business and how information systems create value for it.

If you are interested in more information on risk and vulnerability analysis, I highly recommend two Web sites. The first site hosts research papers by Katherine Morse, Ph.D., that develop the notion of process-oriented risk analysis. (Katherine works with me at **Science Applications International Corporation [SAIC]**, and I adapted her diagram for Figure 2.) You can find this site at *www.ics.uci.edu/~kmorse/NCSC_94.html*. The second Web site is the Computer Emergency Response Team (CERT) site at *www.cert.org/research/JHThesis/Start.html*, which hosts a research paper based on six years of data on reported security incidents.

Vincent LeVeque is a senior security engineer for SAIC. He can be reached at *vleveque@earthlink.net.*

REFERENCES AND RELATED MATERIALS

- "An Analysis of Security Incidents on the Internet 1989-1995," John D. Howard, CERT Coordination Center Research, 1997 (*www.cert.org/research/JHThesis/Start.html*)
- *AS/400 System API Reference V4R4* (SC41-5801-03)
- Computer Security Institute Press Release: *www.gocsi.com/prelea990301.htm*
- TechWeb's "Global Security Survey:Virus Attack": *www.informationweek.com/743/security.htm*
- "The Security-Specific Eight Stage Risk Assessment Methodology," David L. Drake and Katherine L. Morse, Science Applications International Corporation, 1994 (*www.ics.uci.edu/~kmorse/NCSC_94.html*)