# Global Information Assurance Certification Paper

# Brief overview of AS/400 Security fundamentals
# March 2001

### Introduction

For years, the AS/400 has been acknowledged as one of the most secure computing systems in the world, while Microsoft and Unix systems have struggled with one security catastrophe after another. Why has every major computer virus devastated Microsoft and Unix systems and not touched the AS/400? One answer lies in how the OS/400 operating system does its job. Unlike NT or Unix, every object in OS/400 is self described and monitored by the operating system. In contrast, NT and Unix treat files, programs, drivers and viruses with indifference. In an AS/400 a program will not look like a file because of different attributes for each object defined on the AS/400. AS/400 is object-oriented, and evolved from the System38 architecture of the early eighties. This paper will attempt to discuss the very basic questions of how security on an AS/400 can be strengthened. In particular, attention will be given to Securing AS/400 FTP, Special Authorities, Default Passwords and the AS/400 logging subsystem. It should be noted that these areas outlined for discussions are not the only key areas in AS/400 security. They are chosen because they are probably the most easily neglected areas.

### Securing AS/400 FTP

File Transfer Protocol is one the most popular and highly utilised TCP/IP service, despite its security risks. Because FTP can be used to upload data or execute CL commands on an AS/400, it should be secured. Users and attackers alike may upload illegal material to the AS/400. This uploading of data to the AS/400 may take up system resources. System jobs may be initiated by submitting CL commands through or within an FTP session. These jobs may consume system resources e.g. CPU. For these reasons, it is of fundamental importance that FTP on the As/400 system, just like in any other system, is secured.

One of security concerns with the AS/400 is the manner in which the system handles FTP sessions. The AS/400 handles FTP requests in a different manner from UNIX based FTP requests. Despite the differences, the standard FTP commands like PUT, DIR and GET apply on the AS/400 in the exact manner as in a UNIX or Windows system. There are several other AS/400-specific FTP commands that should be treated with caution. Such commands can allow a user or an attacker to create and delete libraries, physical, logical, and source files; and AS/400 file members. Also, AS/400 CL jobs can be run from the local AS/400 FTP client through the use of the Pass an AS/400 CL command (SYSCMD) inside an FTP session. In addition, one can execute a system command on a remote AS/400 FTP server by executing the OS/400 command (RCMD) inside an FTP session. Just as in a UNIX system, a user would create a file with all FTP commands together with a .netrc file with user name and password in plain text to be executed in a non-interactive mode; the same can be done in an AS/400. The process works just the same, a file containing user profile and password in plain text and another file containing FTP commands can be created. For these reasons FTP to the AS/400 should be controlled and be secured. Only authorized profiles should have the privilege of using FTP to the AS/400, and these

profiles should be secured. Two ways of securing FTP in an AS/400, the OS/400 security features and the use of exit programs will be briefly discussed.

#### OS/400 Security

It is important to watch FTP session default naming format (NAMEFMT) value when exchanging files with the AS/400 Integrated File System (AS/400 IFS). When NAMEFMT is set to 0, OS/400 limits FTP transfers to the QSYS.LIB file system in the AS/400 IFS. When NAMEFMT is set to 1, AS/400 FTP allows you to exchange files with a number of other AS/400 IFSs, including the Root (/), QDLS, QOpenSys, QSYS.LIB, and QfileSrv.400 systems. The first option i.e. that of restricting downloading and /or uploading of files through FTP access to specific library, QSYS.LIB, is probably more secure and preferred. When NAMEFMT is set to 0 only the QSYS.LIB filing system is dealt with through an FTP session. Using the NAMFMT system value offers some security in that it restricts FTP session to specific libraries. However, it does not prevent users from executing CL commands or placing illegal data on the system. A lot of other system security features may be employed. Such features were not explored during the compilation of the paper.

#### Using exit programs

Exit programs are another excellent way of ensuring that security within the FTP session is maintained. Exit programs will, once a certain exit point already defined is reached, pass control of the session to an exit program. This program will check registration information to see if an exit program is attached to that exit point. If a program has been attached, control will be handed over to the program. Based on parameters assigned to the exit program, the security officer may perform additional logging, allow or disallow certain transactions within the FTP session. Because an exit program is called before requests are processed (depending on which exit points are provided in an application), the security officer may be able to control or see actions of users using FTP. Furthermore, the exit program may be set up such that it will be able to detect suspicious activities, allowing security officer to create a log, send a security officer a message, notify the user that they are performing actions that are not allowed, or even completely disconnect the user. Some of the activities outlined will of course require a higher level of programming. Exit programs maintain a control point for FTP sessions.

#### Special Authorities

Special authorities may allow users unrestricted execution of programs like DELETE, MODIFY etc. It is important to ensure that users are not granted special authorities or unnecessary privileges. The principle of least privilege should apply. Different AS/400 security levels will control the granting of special authorities and privileges.

If system security level is set to 20, the system will not control access to system resources. All profiles will be authenticated via a username/password combination. All profiles will be granted equal access, as they will all have access to the *ALLOBJ privilege. It is recommended that all profiles be reviewed for the *LMTCPB and *ALLOBJ values under security level 20. The LMTCPB value can either be set to NO or YES for each profile. If it is set to YES, then that profile will be restricted to menu options, it will not have command line access. This is an effective way of ensuring that users do not run commands on the AS/400. If the AS/400 security level is set to 20 LMTCPB is set to NO for all new profiles by default. Whenever there have been new profiles, these values for *LMTCPB and *ALLOBJ should be checked to ensure that users are not granted excessive privileges, as this can be a dangerous combination.

Security levels 30 and above will, by default, apply the principle of denying all. This is advantageous because only the security officer will be able to grant to users special authorities as required.

In general access to the following authorities should be granted to profiles or programs that specifically need them for their job functions: *ALLOBJ, *AUDIT, *SECADM and *SAVSYS.

All latest AS/400 releases no longer ship with security level 20 since that level was found to be inadequate when it comes to security. However, it is still worth auditing for profiles with special authorities. The following command, obtained from an audit tool that we use on our AS/400 audit work, may be used to carry out this function:

```
DSPUSRPRF USRPRF (*ALL) OUTPUT (*OUTFILE) +
                OUTFILE (SANS/LEVELONE)
```

The above command will dump all user profiles in a file called LEVELONE residing on the SANS library in the AS/400. The information it provides might not be useful unless filtering is applied to obtain only relevant information, e.g. dump all profiles with *SECADM privileges or dump all profiles with *LMTCPB=NO. For more information on how to perform these tasks, it is recommended that the AS/400 technical programming tips newsletter archives be consulted. A link to the archives may be obtained through the ***http://www.as400network.com*** address.

#### The AS/400 and default passwords

Default passwords are one thorny issue when it comes to the security of systems. Most systems are shipped with system accounts (e.g. guest on UNIX and NT) that have default names and passwords. These accounts make the systems more vulnerable, if they are not disabled. The AS/400 is no different. The AS/400 comes with default IBM-supplied profiles. These profiles sometimes have their passwords set to *NONE, meaning that they cannot be used to sign on to the AS/400, just like the password field in an /etc/shadow file of Solaris has NP (sometimes LK) for accounts like bin, sys, etc showing that such accounts are locked. However, it is worth checking if these profiles do not contain default passwords. According to Joe Hertvik, "The way many AS/400 administrators traditionally assign passwords is to set up the Password parameter (PASSWORD) as a default password (where the user's password

is equal to the user profile name) and then set the Set Password to Expired parameter (PWDEXP) to equal *YES." Mistakes may happen however. Assume that the administrator accidentally set the PWDEXP to *NO for a specific profile. That profile may be used indefinitely without it having to change its password, which was initially set equal to profile name. It is therefore worth checking for default passwords in the system. IBM provides the ANZDFTPWD command in order to help in alleviating the problem of default passwords. To use this command, the profile should have *SECADM and *ALLOBJ privileges. Because of the sensitivity of these privileges, only authorized users and security officer should be allowed to run the ANZDFTPWD command. This command will search for all profiles with default passwords. An action may also be specified in the ACTION field of the command. The action may be to disable the profile, force it to change the password or do nothing. This is a useful command because it produces a report detailing all profiles with default passwords and their status, whether still active or disabled. It is recommended that ANZDFTPWD be run with a *NONE action specified first so that there will be no disruptions to normal business processes. This can be accomplished by issuing the command:

ANZDFTPWD ACTION (*NONE)

The above will provide a report of all profiles with default passwords. This report can be sent to a print queue or can be viewed on screen. After analyzing the report produced by ANZDFTPWD all profiles with default passwords should be forced to change their passwords in their next sign on session. This can be accomplished by issuing the command:

ANZDFTPWD ACTION (*PWDEXP)

The command above will change the *PWEDXP value from NO to YES. This will effectively force all profiles with default passwords to change them when they sign on to the system after the ANZDFTPWD command was issued with ACTION (PWDEXP) specified. Having changed all default passwords, the next discussion point is that of logging and audit trail.

#### Logging

Audit trails and violation attempts should be logged to determine the effectiveness of logical security and to establish if there were any unauthorized logon attempts or actions performed against the system. Logging and audit trails should form an integral part of a security policy. It is therefore important that logging on any critical system be enabled.

QAUDCTL serves as the on/off switch for AS/400 security auditing. By default, the value of QAUDCTL is set to *NONE which tells system not to perform any security auditing. This should be changed to the value *AUDLVL and/or *OBJAUD which tell AS/400 to perform auditing. *AUDLVL activates event auditing by system or by user. *OBJAUD activates object auditing. QAUDLVL value will help the security officer audit security by logging data concerning security-related events to a journal named QAUDJRN. QAUDLVL determines the level of auditing the system performs. The security officer can specify one or more values (unless one of the values is *NONE, which causes other values to be ignored). The most important thing to

remember when using QAUDLVL to set up auditing is to begin with only few values. If many values are activated at once, much time may be spent managing the size of the journal receiver attached to the audit journal (it will grow quickly). Some of these options are helpful to log and review on an ongoing basis, *SECURITY, *PGMFAIL, and *SERVICE, to maintain tight security. However, other values such as PGMADP, *CREATE, and *DELETE may prove more helpful for spot-checking or to solve specific security-related problems that may be encountered.

It is recommended that QAUDLVL be set to *SECURITY, *SAVRST, *SERVICE and *AUTFAIL. Others values should be added one at a time after the security officer is comfortable with the size of logging required and the growth of the audit journal. The size of the journal should be constantly monitored as it might result in unexpected size problems.

Most AS/400 administrators believe that what IBM has disabled is final. As a result they rely on application running on top of the AS/400 to provide logging. This may prove calamitous because the application will not log attempts to access the system, but failed transactions and other activities on the application.



### Conclusion

Although an attempt was made to cover AS/400 critical functions, the discussion in this paper is very limited. AS/400 is very versatile in terms of security settings. The topics were covered in a brief overview manner. It should be noted that a number of potential security areas were not covered. Focus was narrowed to the mostly problematic areas. They are problematic because it is easy to make a mistake and grant excessive authorities to users, as an example.

It is vital to secure FTP as threat areas were discussed. More importantly, it is recommended that, if possible, distinct profiles be used for FTP access. This will minimize the threat posed by sniffing. If an attacker can sniff FTP logon credentials, s/he will not be able to use them outside the FTP session as such credentials will be different from those used to normally sign on to the AS/400.

Special values regarding special authorities were highlighted. Although these values are important, they do not cover all aspects of the AS/400 special authorities. This is big area.

Logging should be enabled, as it forms a very important area in managing and assessing the overall security of the AS/400, and any other system.

As the AS/400 becomes more exposed to the Internet, security values should be tightened. Area like HTTP and trust relationship with other systems and the much talked about plan to deliver Linux in a logical partition (LPAR) in OS/400 sometime this year after the initial release of V5R1, should be watched. The AS/400 is no longer the "most secure system" as most AS/400 disciples would believe.

**References**

Madden, Wayne "Starter kit for the AS/400, 2nd Edition 1994" URL: http://www.as400network.com/resources/starterkit/toc.htm

Jones, L. Wendy "AS400 Internet Security: Keeping the wolf away from the back door" **Update from Partner World for Developers, AS/400 March 2000**

Dambrine, Thibault "Practical Programming for AS/400 FTP Automated Interfaces" URL: http://www.tugon.ca/AS400FTP.htm

Putla, Joe "AS/400: Open for e-business" **Update from IBM AS/400 Partners in Development July 1999**

International Technical Support Organization, Rochester Center "An Implementation Guide for AS/400 Security and Auditing: Including C2, Cryptography, communications, and PC Connectivity" **Document Number GG24-4200-00**

Hevik, Joe "Do you have default passwords?" **AS/400 Network Expert, November/December1999**