Canonicalization problems in iSeries FTP security

Insufficient default FTP access control

The IBM iSeries (AS/400) server supports an Integrated File System (IFS) that provides a unified access scheme to all of the files and to all of the database tables in all of the database libraries. For example, the path to a zipped XML file you exchange with a business partner may look like this:

```
/home/bp/outgoing/2004/jp-orders-2765.zip
```

The root file system is reminiscent of and fully compatible with the Windows and Unix file systems, and it can be used to store images, Windows executables, and basically any file you want. On the other hand, the QSYS file system is specific to the AS/400 platform and it holds all of the AS/400 library objects: programs, data areas, tables, index files etc. The QSYS file system is mapped into the integrated file system, so the path to the invoices table may look like this:

```
/QSYS.LIB/APLIBF.LIB/APINVP.FILE/APINVP.MBR
```

The iSeries has a built-in FTP server that is among the more popular ways to exchange data between legacy iSeries applications and the rest of the world. The built-in FTP server does not include support for an FTP document root that limits a user to a restricted set of folders. The FTP server relies only on ACLs (object authority) and on FTP verb permissions - a user can be blocked from specific FTP verbs, like CD, GET, PUT, and DIR. The practical meaning is that if a user has a valid account on an AS/400 server, and permission to GET files via FTP, then this user can retrieve any database table that he has read access to, and change any database table that he has update access to.

Because of the way most legacy applications are constructed, users that have access to the application via the terminal emulation usually have both read and update permissions to the application's database.

Even if a user is blocked from the CD verb, the user can still retrieve a database table by specifying the direct path to the table.

For example, an application user retrieves a daily summary table called DAILYFILE from library EXCELDATA for further spreadsheet processing, by running the following command:

```
GET EXCELDATA/DAILYFILE
```

By running the next command, this user, who has read authority to the application database, retrieves the entire chart of accounts:

```
GET APLIBF/APCACCP
```

If the daily file is a CSV file already transformed into a PC compatible form, and the command to get it is

```
get /home/user/dailyfile.csv
```

Then the following is the equivalent IFS path to the chart of accounts table:

```
GET /qsys.lib/APLIBF.lib/APCACCP.file/APCACCP.mbr
```

The user was supposed to get only the prepared CSV file, but due to lack of an FTP document root he can pillage the entire database at will, and so far there is nothing to stop him.

The attempted solution that fails to work

IBM did provide a set of API programs, also called exit programs, which enable enhancements like an FTP document root to be added to the basic FTP server. Some AS/400 administrators have written their own code to further secure FTP, others went out and purchased third party security applications to achieve elevated security. The FTP exit program actually must manage a secondary group of ACL and cross-check all FTP access requests with the application ACL in addition to the regular permissions the OS manages.

Why is that? Many legacy applications were not designed with network security in mind, and when a user has read access to a table so he can use an application program, this user has read access to the table regardless of the way he is connected. This means that he can read the application table files also by network protocols like FTP and ODBC. A hardening of OS level ACL may result in the legacy application failure, therefore a network security application must provide its own access lists in addition to those managed by the system. Some of these solutions are vulnerable to a particular canonicalization attack resulting in directory traversal exposure.

Imagine the following scenario: a business partner is to retrieve a zipped file from an IFS directory, like the file in the first example. There is an FTP exit program that is supposed to restrict the external user's access exclusively to the /home/bp/outgoing/ folder.

The external user should have read access to all of the sub-folders and files inside /home/bp/outgoing. The application database tables

have an ACL allowing public access, because that is how the vendor set it up 5 years ago. When the external user attempts to access a database library, by issuing

```
GET /qsys.lib/APLIBF.lib/APCACCP.file/APCACCP.mbr
```

he receives a "550 Request rejected" error message, because the specified path is outside the allowed folder. Apparently, there is no way for the external user to actually access the database tables, is there? In fact, there is. The external user can try the following:

```
Get /home/bp/outgoing/../../../qsys.lib/aplibf.lib/
apcaccp.file/apcaccp.mbr
```

Of course, that is not the only possible syntax. Try this:

```
Get /home/bp/outgoing/./../../../bp/../../qsys.lib/aplibf.lib/
apcaccp.file/apcaccp.mbr
```

The exit program did not parse the FTP string properly. The external user has gained unintentional access to sensitive data. If the public authority includes update as well as read, then a PUT will replace the table contents with modified data. Game over.

Summary

The iSeries FTP server has no built-in ftp document root, relying only on ACL and on verb control to manage access permissions via FTP. The integrated file system structure makes it very easy to access both files and database tables from FTP. Existing legacy applications usually do not have proper ACL permissions that stop network based access. Third party tools that attempt to secure access via FTP may be susceptible to directory traversal attacks. iSeries lacking FTP security tools are 100% vulnerable to this type of attack.

Several commercial iSeries security vendors were approached to see whether their products are vulnerable to this canonicalization attack.

Raz Lee: notified on Feb 15, fix is available.
Castlehill: notified on March 15, fix is available.
Powertech: notified on March 15, fix is available.
Bsafe: notified on March 2, vendor has no comment.
SafeStone: notified on March 15, no reply received.
NetIQ: notified on March 15, no reply received.

More information about AS/400 vulnerabilities and about AS/400 security issues can be found in the "Hacking iSeries" book, at www.venera.com