CGI

Five points for securing legacy applications “in place”

VIEWPOINT SERIES

By Michael J. Corby

Rarely was security considered in the application suite of years ago. The computer room required a “secret” keypad code to enter it, exposing the sounds of rushing air and the sights of spinning tapes and paper flowing from line printers. The green-screen terminals had a standard complement of user IDs and passwords, but that was pretty much all the security needed. Users only worked onsite, and applications were geared primarily toward entering data for past transactions. There was no obsession about the Internet, real-time transactions, malware, privacy laws or compliance regulations.

Many legacy systems implemented back in those “simpler” days are still in operation today. While they may not be broken, they do need to meet more stringent security requirements because they may now have connectivity with newer web-based, cloud or other service platforms. Most legacy developers have retired, and who wants to write programs in COBOL or PL/1 or FORTRAN anyway?

So how can organizations sufficiently secure their legacy applications without throwing them out or rewriting them? Here are five points to consider:

1. It’s all about the data, so inventory it.

Since the dawn of commercial computing, the focus has been on the data. Processing speed was important, but the key product was using and creating accurate data records. Despite the changing environment, this is still a root requirement. A key step in securing applications is to inventory the databases and files, identify particularly sensitive data items, and document whether each item has a high, moderate or low need for the three components of security—confidentiality, integrity and availability—as shown here in Figure 1. Another step is to ensure legacy systems are noted on data flow diagrams and that data on legacy systems is encrypted properly.

Figure 1 – Sample Data Security Attributes Inventory

Data Item Security Inventory

Data Name: ______________________________

Database/File Name: ______________________________

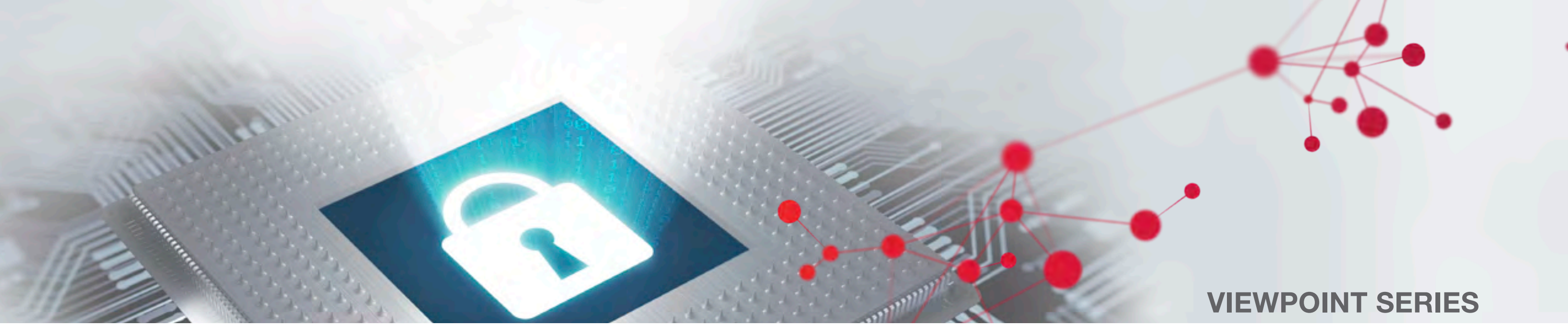
Size: ______________________________

Format: ______________________________

Security Needs: (Check each)	**High**	**Moderate**	**Low**
Confidentiality			
Integrity			
Availability			

2. Availability is a part of security.

Most people think of security as protecting private information from unauthorized use. But security also includes data integrity (the data is accurate) and availability (the data is not lost or unacceptably delayed). Some mechanics of legacy systems present an obstacle to availability. Sequential files, for example, do not support rapid record retrieval. However, archiving sequential files can be an efficient way to re-load corrupted databases or compare data changes. A reliable set of backup files, kept offsite and properly protected, is still the best way to make sure critical data assets are not lost or rendered unusable. Even more recent applications can be expanded to include fast recovery from backup storage, with the chance to “roll back” to earlier versions and create a working copy of key master files to be used in the event of a business disruption or system failure.



3. Require regular security awareness training.

I’m no longer surprised at the number of breaches caused by people failing to do the right thing. Well over half of reported incidents are caused by clicking on infected attachments, browsing dangerous websites, leaving computers logged in and, worst of all, choosing pathetically simple passwords. The best security technology in the world is rendered useless if a person leaves the system open to “visitors.” Security awareness training is not just for new employees. All employees should be required to take refreshers at least annually. Best practice is to put security awareness in performance reviews. Managers will be asked to explain their workers’ security responsibilities and employees will seek ways to become more security savvy.

4. Use single sign-on.

Although strong identity and access management (IAM) controls didn’t exist for many legacy batch applications, today’s host systems have implemented single sign-on (SSO) to ease the pain of frequent and diverse password change rules. SSO also enables control over who can use what. Legacy systems can be part of the SSO environment just as successfully as web-based applications. The rules of good behavior built into SSO and IAM front-end services can provide a controlled portal that enforces session limits, allows for application maintenance windows, and prevents brute force attacks by unauthorized people.

5. Be ready for an audit.

The auditor actually can be a good friend. Many principles that come under scrutiny during financial, operational or compliance audits extend to legacy as well as state-of-the-art applications. Often legacy systems are running finances, capturing customer orders and planning production floor activities. Audit principles like separation of duties, maintaining current copies of program source code, and providing for file backup and recovery, are time-tested security practices. Data owners, system stakeholders and business users demand compliance with good business practices, and that includes security.

My philosophy regarding audits is to do what the auditors expect every day: make sure all systems are documented and changes are approved; verify that protected data is being treated properly (see Figure 1 again); and, above all, conduct a good risk assessment. Know what your legacy apps do and whether they are as secure as they can be, and craft a plan to conduct more frequent reviews or implement process surveillance to confirm that all is happening according to plan.

In summary, legacy systems may not have all the technical facilities built in to their architectures, but good security practices can replace the technology that is lacking. The good news is that the risk inherent in mobile and web-based applications usually are not the same in older systems, and we can take heart in knowing the legacy infrastructure is probably too big for someone to just carry off.