From businesses to government agencies, nearly every entity contracts some aspect of software development, system integration and hosting services—creating an emerging crisis in accountability.

How does an organization that has an IT department with average skills implement a large, complex, far-from-average new technology, such as electronic health records or asset management systems? In this age of specialized skill sets, it seems perfectly sensible to outsource such a deployment. Managing how to secure the confidential data contained within the new technology—and the welter of regulatory requirements that must be met to do so—is one of the most important and underappreciated challenges of this decade.

With hundreds of frequently overlapping security requirements, it can seem deceptively simple to contractually require that the vendor be compliant with all the appropriate regulations. What cannot be overlooked, however, is that the contracting organization must have sufficient resources to provide adequate oversight of vendor compliance activities.

RESPONSIBILITY CANNOT BE OUTSOURCED

Whether the vendor is developing and integrating new technology that the organization will maintain or the vendor is also hosting the new technology, the compliance requirements for securing confidential data are the same.

In the US, for example, federal regulations require that even if a vendor agrees to provide security services, the owner of the data be responsible for ensuring that the vendor protects the data.

Though a vendor may be the source of a data breach, in the court of public opinion, the negligent party is the entity that has contracted the services of an inadequate vendor.

For example, in the case of the Target breach, the name of the third-party vendor that was the source of the breach was eventually identified, but the breach itself was publicized as, “Target has been hacked.” At Target’s highest management levels, heads rolled and the company’s bottom line took a major hit.

The accountability and compliance crisis goes far beyond the retail world, touching all industries: commercial, not-for-profit and, perhaps most urgently, government.

FEDERAL FUNDING TRIGGERS FEDERAL COMPLIANCE STANDARDS FAR AND WIDE

Although US state, city and town agencies are not federal entities, by accepting federal funding, they must meet federal standards to connect to federal sources of information, such as the US Internal Revenue Service (IRS), the US Social Security Administration, and the US Department of Health and Human Services (HHS). The funding of these systems has fueled the implementation of these standards.

Correspondingly, many business and nonprofit entities that provide services to cities and states based upon confidential information are finding that they are contractually required to become compliant with such standards in order to continue doing business with these government entities.

Over the past five years, the US National Institute of Standards and Technology (NIST), Special Publication 800-53, Security and Privacy Controls for Federal Information Systems and Organizations,1 has emerged as an information security standard for compliance among US state and local government entities.

Regulations such as the US Health Insurance Portability and Accountability Act (HIPAA), the US Affordable Care Act (ACA), and the US Federal Information Security Act (FISMA) have had additional impact on IT security controls for personal health information (PHI). IRS Publication 1075 is a complementary set of standards for federal tax information (FTI).

In the rollout of the new health insurance exchanges across the US, the Center for Medicaid and Medicare Services (CMS) has mandated the
use of NIST SP 800-53 for those state entities choosing to accept funding. The latest set of compliance requirements (the CMS Minimum Security Requirements, or MARS-E, Minimum Acceptable Risk Safeguards for Exchanges) maps directly to NIST SP 800-53. These standards are now being attached to funding to update or implement new Medicaid management information systems (MMIS) and eligibility systems run by states across the country.

Commercial and nonprofit support services for these new and updated health systems are feeling the trickle-down effect of these mandates when contracting entities require periodic inspection of their controls to determine if they are compliant.

One of the requirements specifically called out by the CMS and the IRS has been for those entities to have periodic independent third-party security assessments. These and other assessments have revealed critical and persistent challenges involved in managing the complexity of third-party contracts for services.

THIRD-PARTY ASSESSMENTS REVEAL GAPS IN THE GOVERNANCE PROCESS
Governance problems become visible when mandated independent security assessments examine vendor practices. The most frequent findings appear in these NIST-designated areas:

- Secure software development (SA)
- Access controls (AC)
- Configuration management (CM)
- Logging and monitoring (AU)

These areas map to the following gaps in governance activities by the contracting organization:

- Lack of resource planning for sufficient technical oversight
- Limited in-house knowledge of the security requirements for the new technology
- Over-reliance on generic contract language for technical compliance requirements

A new technology compounds existing problems. Layers of technology continue to increase, creating more layers of security risk. Virtual technologies, for instance, have added the ability to build out incredibly powerful operating systems in a far smaller physical space. These technologies make possible a security breach much bigger than the compromise of one server. Compromise of the hypervisor (the virtual machine host managing the virtual operating systems) can mean that the hacker has access to all the servers and data inside that virtual system.

As new products are deployed, there is more chance for documentation of security features to be minimal or rushed, and existing documentation can quickly become outdated. For example, service-oriented architecture (SOA), with its certificate architecture for authentication, can become a black hole for compliance analysis. For a contracting organization, lack of documentation can mean being held captive to a vendor and expensive consulting fees.

Project risk assessments have not adequately captured many aspects of vendor oversight, including managing the development, test and production system rollouts. It is not uncommon for vendors to have unfettered control over all aspects of the new development, test and production systems, often denying the contracting entity any access.

“Vendors scramble to get code to work on a deadline or to fix emergencies, too often at the expense of security.”

This allows code to be created in undocumented systems that will be more likely to have problems in a secure production environment. Vendors scramble to get code to work on a deadline or to fix emergencies, too often at the expense of security. The risk brought about by this deeply engrained pattern in this outsourcing culture cannot be overestimated.

MORE OUTSOURCING MAY HELP SOLVE OUTSOURCING PROBLEMS
Contracting organizations often struggle with the question of how to better monitor their vendors. Many are not prepared to assign in-house engineers who already have significant duties to provide oversight of vendor activities. Often, the reason organizations turn to outsourcing in the first place is that their employees have insufficient expertise to understand all aspects of the technology. Even organizations trying to monitor their vendors may not be set up to handle the necessary level of reporting duties. Front-line technical personnel often do not have sufficient access to higher-level project managers to report problems.

Ironically, the solution to outsourcing problems may be more outsourcing. In the same way an organization outsources for technology development and deployment expertise, it may need to consider whether to outsource technical compliance from an independent party that has no relationship with the vendor.
This establishes segregation of duties (SoD) so that the secure development and implementation of the systems and software underlying new technology are adequately protected. Rather than waiting for a security assessment just prior to, or just after, rollout into production, contracting organizations would be better served by implementing continuous monitoring throughout the project.

**IMPROVING VENDOR GOVERNANCE**

Improving vendor governance may require a shift in priorities or culture for the contracting organization. The security challenges discussed previously generally manifest themselves in five distinct areas where the contracting organization can take steps for better oversight:

1. **Recalculate the risk and cost of secure software development.** For many, especially cash-strapped government agencies, cost has been the limiting factor for providing sufficient vendor oversight. Today’s rising incident rates for data breaches, coupled with increased regulations, call for a fresh look at the cost-benefit analysis of putting more resources into vendor oversight.

   Both the NIST and the US National Aeronautics and Space Administration (NASA)\(^3\) have completed studies on the differences in cost for remediating code errors during the different phases of software development. The studies revealed that it can cost up to 30 times more to resolve code errors once the product is in production status.

   The Ponemon Institute’s ninth annual report, *2014 Cost of Data Breach Study: Global Analysis,*\(^4\) highlights the fact that the average cost for each record lost or stolen increased from US $136 to $145 (9 percent) from the previous year. The longer the delay in implementing and overseeing secure software development, the higher the cost when the breach occurs.

   In addition to data breach record costs, there is significant compliance risk\(^3\) in not providing sufficient oversight of vendor activities, as is required in CMS’ MARS-E, FISMA and IRS 1075\(^6\) regulatory documents. For example, one of the requirements from SP 800-53 is SA-10 Developer Configuration Management:

   The organization [meaning the contract holder] requires the developer of the information system, system component, or information system service to:
   
   a. Perform configuration management during system, component, or service development, implementation, and operation;
   
   b. Document, manage, and control the integrity of changes to configuration items under configuration management;
   
   c. Implement only organization-approved changes to the system, component, or service;
   
   d. Document approved changes to the system, component, or service and the potential security impacts of such changes; and
   
   e. Track security flaws and flaw resolution within the system, component, or service and report findings to defined personnel or roles (defined in the applicable security plan).\(^7\)

2. **Mandate secure software development.** Security controls should be built into every phase of software development, regardless of which software development model the vendor uses. NIST provides an excellent template in its Special Publication 800-64, *Security Considerations in the System Development Life Cycle.*\(^8\)

   Although a system integrator may take on the task of building out the infrastructure (e.g., servers, databases, virtual hosts, routers) to support the new application, this type of vendor’s primary focus is to develop a software product that meets the requirements of the client in the most cost-effective way possible.

---

**Enjoying this article?**

- Read *Configuration Management: Using COBIT® 5.*  
  [www.isaca.org/configuration-management](http://www.isaca.org/configuration-management)

- Learn more about, discuss and collaborate on governance of enterprise IT (GEIT) and information security in the Knowledge Center.  
  [www.isaca.org/knowledgecenter](http://www.isaca.org/knowledgecenter)
Unfortunately, cost-effective does not necessarily translate into secure. In many third-party environments, security is a much-delayed add-on, and documentation is focused primarily on application development and meeting business requirements.

One could say that it is an occupational hazard that IT vendors want to implement infrastructure in a way that is most conducive to software development. The fastest approach for software development is when the applications have complete access rights to all data. Fortunately, regulatory requirements mandate better controls, but if the contracting entity does not mandate secure development systems and detailed access control documentation of the systems, it risks a disaster. The application could break in a locked-down production environment or be hacked due to lack of controls in an open one.

These requirements should be put into place upon commencement of the contract and not applied in the final deployment into production, where it is far more costly to resolve.

How a software developer builds the development environment is critical to the delivery of a secure application and infrastructure.

3. Maintain access controls. With adequate resources, a contracting entity can better ensure that vendors implement compliant controls and develop secure software that meets business requirements. Vendors ought not to be allowed to develop in a security vacuum where use of generic administrator identifications (IDs) is the norm and password controls are minimal.

When new systems are first booted up for the initial development environment, the vendor should have a documented server build ready for deployment. The contracting entity should provide oversight for the standard build to confirm that security engineering principles form the backbone of the development environment.

For example, the NIST SP 800-53 Control SA-8 Security Engineering Principles offers the following guidance:

- Developing layered protections;
- Establishing sound security policy, architecture, and controls as the foundation for design;
- Incorporating security requirements into the system development life cycle;
- Delineating physical and logical security boundaries;
- Ensuring that system developers are trained on how to build secure software;
- Tailoring security controls to meet organizational and operational needs;
- Performing threat modeling to identify use cases, threat agents, attack vectors, and attack patterns as well as compensating controls and design patterns needed to mitigate risk; and
- Reducing risk to acceptable levels, thus enabling informed risk management decisions.

The contracting entity should require and maintain administrative access to all development, test and production systems. If the vendor has implemented proper logging and monitoring of access, any unauthorized changes should be easily tracked to the source of that activity.

It cannot be overstated that the contracting entity, not the vendor, is the owner of those systems and must maintain control. The vendor must never control the systems to the exclusion of the data owner. The simplest way to achieve this is to always have administrative access to the systems from the very beginning of the project.

Equally, the contracting entity must maintain ownership of the code that is being developed because it is a form of intellectual property for which the entity is paying. Therefore, consistent and complete access to the vendor’s code repository provides for continued possession and allows the entity to monitor the vendor’s controls over the code.

Using PHI, personally identifiable information (PII) and federal tax information (FTI) data in development environments often helps to develop code that will eventually use these data. However, maintaining access
controls over who sees the data is the responsibility of the data owner, not the software developer.

Products in the marketplace can obfuscate confidential data for testing purposes, but many organizations find them cost-prohibitive. With adequate controls over access, waivers from federal entities (the IRS, in particular) can be obtained.

SoD is often nonexistent in development environments and, quite frequently, in production environments. Software developers should not have any more than read-only access to production environments. Database administrators should not have server administrator rights and vice versa. Implementing these controls in the development environments means that systems are managed more securely from the beginning of the project.

Some vendors resist this approach, claiming that it could create security problems when the code is moved into production, but the reverse is actually true if the development systems are configured securely.

4. Start configuration management from the beginning. In the eyes of NIST, the IRS and FISMA, “configuration management” has become an umbrella term that incorporates a range of activities, including:

- Documented baseline configurations based on national standards
- Implementation of least functionality
- Change control management
- Information systems component inventory
- Testing of changes prior to deployment
- Security impact analysis of changes
- Access restrictions for changes
- Software usage restrictions

Generally, these requirements have not been addressed until much later in a project. As a result, undocumented changes, unpatched systems and a lack of standardization lead to the contracting organization not having firm control over the security architecture.

Patching and updating critical system components that work in layers can lead to expensive crashes and downtime when systems are not configured to a single standard across the architecture. A patch may work perfectly on one Linux server and fail on the next because someone made a change to the server that was not documented. When this is replicated across more than 200 servers, the cost to managing updates can become prohibitive and lead to insecure systems.

Monitoring changes on systems is much easier when a common standard is implemented. Small changes can also be the first alert of a data breach in progress.

Monitoring system changes is a core element in meeting, for example, the NIST requirement in AU-2 Audit Events:

*Generate audit records for the following events in addition to those specified in other controls:
  a) All successful and unsuccessful authorization attempts.
  b) All changes to logical access control authorities (e.g., rights, permissions).
  c) All system changes with the potential to compromise the integrity of audit policy configurations, security policy configurations and audit record generation services.
  d) The audit trail shall capture the enabling or disabling of audit report generation services.
  e) The audit trail shall capture command line changes, batch file changes and queries made to the system (e.g., operating system, application, and database).*

5. Control logging and monitoring. Possibly the largest security gaps exist in the areas of logging and monitoring. In implementing security controls, contracting organizations often focus on product performance and delivery to the detriment of security controls.

It is common practice for the contracting organization to require the vendor to perform all monitoring of the new systems. Unfortunately, this usually means that the organization expects the vendor to monitor vendor administrative activities. This is an obvious conflict of interest for the vendor and is in direct conflict with security best practices.

Consider the NIST SP 800-53 Audit and Accountability (AU) control AU-9(4) Access by a Subset of Privileged Users:
The organization [the contracting entity] authorizes access to management of audit functionality to only those individuals or roles who are not subject to audit by that system, and is defined in the applicable security plan.”

Reasonably, the organization would want to retain control of audit logs that may contain confidential information, in order to determine whether the vendor is performing activities that are compliant with federal requirements. Logs can be stored at a vendor location without the vendor having access, beyond read only, or can be transferred to another location.

Alternately, another third-party vendor could be engaged to perform monitoring activities as long as that party reports to the contracting organization, not to the vendor performing the administrative activities.

This is not to say that the vendor should be exempt from monitoring application and performance logs. It is simply a matter of SoD, so that system administrators and database administrators are not in charge of monitoring their own actions.

Should a contracting organization allow another group within the vendor organization to monitor their own administrative activities? There may not be sufficient SoD to be an effective practice. The vendor can overrule or delay security findings if kept to an internal group. Also, it is less likely that such decisions are transparent to the contracting organization.

Audit logs, when implemented according to requirements, are the backbone of security prevention, detection and response. An Incident Response Plan provides a process for responding to security incidents that are found in logs. Whether malware, failed logins or distributed denial of service (DDoS) attacks, there should be a process for performing an initial analysis, documentation, prioritization and notification. The contracting organization should ensure that a formal, detailed plan is in place for preparation, detection and analysis, containment, eradication, and recovery that is compliant with federal requirements and details actions, as well as reporting activities, for the vendor to incorporate.

Figure 1—Current Practices Related to Monitoring and Managing Third-party Risk

<table>
<thead>
<tr>
<th>Type of Information</th>
<th>How Often Does Your Organization Formally Track?</th>
<th>Base: 106 IT decision-makers at enterprises in the US, UK, France and Germany</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threat and vulnerability management</td>
<td>68%</td>
<td>36%</td>
</tr>
<tr>
<td>Encryption policies and procedures</td>
<td>67%</td>
<td>16%</td>
</tr>
<tr>
<td>Security incidence response processes</td>
<td>66%</td>
<td>33%</td>
</tr>
<tr>
<td>Threat intelligence practices</td>
<td>64%</td>
<td>23%</td>
</tr>
<tr>
<td>Security roles and responsibilities within the organization</td>
<td>58%</td>
<td>16%</td>
</tr>
<tr>
<td>Configuration standards for infrastructure</td>
<td>58%</td>
<td>13%</td>
</tr>
<tr>
<td>Patch management</td>
<td>55%</td>
<td>37%</td>
</tr>
<tr>
<td>Configuration standards for applications</td>
<td>50%</td>
<td>16%</td>
</tr>
<tr>
<td>Organizational management practices of their own third parties</td>
<td>42%</td>
<td>10%</td>
</tr>
</tbody>
</table>

Source: BitSight Technologies, a commissioned report conducted by Forrester Consulting, November 2014. Reprinted with permission.
CONCLUSION
In October 2014, BitSight Technologies commissioned Forrester Consulting to examine the current practices of IT decision makers as they relate to monitoring and managing third-party risk. The resulting report, released in January 2015, found that “there is significant appetite for monitoring various elements of third-party security, yet few firms have the resources to do so with adequate frequency or objectivity” (figure 1).

The challenge of applying good security practices is greater than ever. How resources are applied during major technology initiatives and improvements can be the difference between a secure system and one that is constantly subject to problems.

In addition to meeting security requirements, the core areas discussed in this article lead to reliable systems. Problems are more likely to appear when controls are not in place. They also become extremely difficult to track and resolve. Without solid security principles, undetected breaches are more likely to occur.

The time and effort taken during the beginning of a project to use secure standards will result in significant savings—of money, time and trouble—throughout the life of the technology.

ENDNOTES
5 Compliance risk is the risk of legal sanctions, material financial loss or loss to reputation that the organization may suffer as a result of its failure to comply with laws and regulations.
7 Op cit, NIST 2015
9 Op cit, NIST 2015
10 Ibid.
11 See NIST SP 800-61, Revision 2, Computer Security Incident Handling Guide for an excellent template.
13 Ibid., p. 3