CobiT Security Baseline Applied to Business Web Applications
A Practical Approach for All Sizes of Organisations

Web applications are application software programs delivered using a web browser over a network such as the Internet or an intranet. They do not include simple static web sites, but comprise interactional, transactional and business-integrated web sites and systems.

Many are developed in a relatively ad hoc manner. They may be built in-house, by specialist developers, or by other bodies such as design agencies that may subcontract development work. Compliance with the Payment Card Industry (PCI) Data Security Standard (DSS)\(^1\) is driving e-commerce operators to consider their application development processes, configuration and monitoring, but many other web applications do not have sufficient controls applied.\(^2\)

Security is not always defined adequately in specifications, proposals and contracts with design agencies and developers—unlike matters of functionality, visual design, usability and accessibility. The large number of web applications recently compromised by Structured Query Language (SQL) injection attacks has raised the importance of application security in some operators’ priorities, but often the response is to patch the application for a particular attack against one resource without addressing the fundamental flaws. This is akin to strengthening one door in a house against a burglar, but leaving the other doors and windows unprotected. Malicious users do not necessarily use the same entry point and method all the time.

For many organisations, applying security controls to web applications appears to be an impossible task, since they may lack experience with the types of controls more common in highly regulated environments and are unfamiliar with the standards and methods available. Software development companies may use an approach like that outlined in the Software Assurance Maturity Model,\(^3\) but many are small and medium enterprises/businesses (SMEs/SMBs) that may operate only a single web application, such as a transactional web site. CobiT\(^4\) Security Baseline can be used by such organisations to make a significant—methodical and verifiable—difference in the security of their web applications.

**CobiT SECURITY BASELINE**

CobiT Security Baseline defines 44 minimum control steps as the most important security-related objectives on which organisations should focus. The steps are cross-referenced to the 34 processes and control objectives found in CobiT and grouped into four domains: Plan and Organise (PO), Acquire and Implement (AI), Deliver and Support (DS), and Monitor and Evaluate (ME).

Figure 1 shows the steps classified against a typical software development life cycle (SDLC) model; web-application-specific issues are discussed in the following sections.

**PLAN AND ORGANISE STEPS**
The following are the steps related to the PO domain:

- **Step 1—Define the security strategy and the information architecture.** The first step for mapping an existing or proposed web application is to understand the data being processed and stored. Applications should have a data classification model defining what is being collected, processed and transferred, and how and when it should be protected and destroyed. The policies, regulations and other constraints to be complied with should be defined. Web application authorisation, which is often a role-based authorisation approach, must be applied consistently. The authentication and verification processes for every type of transaction by users (e.g., people, processes, remote systems) need to be considered. A

---

Colin Watson, CISA, CITP, CISSP, is the technical director and principal consultant at web security consultancy Watson Hall Ltd. (London, UK). With an engineering background, he has worked in the IT field for 13 years, concentrating exclusively on web application development, security and compliance.

---

Colin Watson, CISA, CITP, CISSP, is the technical director and principal consultant at web security consultancy Watson Hall Ltd. (London, UK). With an engineering background, he has worked in the IT field for 13 years, concentrating exclusively on web application development, security and compliance.
A multilayered security approach is often used with web, application and database servers protected in a number of ways. Special attention to defining the trust boundaries and control points can be of assistance in this step. Most web applications are not stand-alone systems, and all required services should be identified. The authenticity of information, as well as its confidentiality, integrity and availability, should be considered.

- **Step 2—Define the IT organisation and relationships.**
  Ownership of web applications is often unclear. Responsibilities need to be assigned to appropriate staff, documented and explained to everyone in the organisation. Delegation of responsibility to other organisations must not be allowed, but is often found where services have been supplied by a third party (e.g., hosting company) or the system has been supplied by another organisation (e.g., design agency).

- **Step 3—Communicate management aims and direction.**
  Security awareness should be built into the initial web application training. The type of awareness will depend on the individual roles (e.g., help desk, supervisor, application administrator, system owner, database administrator). Clear processes should exist for handling security incidents, such as performance degradation, defacement, denial-of-service (DOS) attacks, data contamination, data loss by SQL injection, etc. Security should form part of regular reviews, and staff should be given periodic refresher training on security issues and their responsibilities.

- **Steps 4-7—Manage IT human resources.**
  Taking references and undertaking background checks are vital for all those involved with the application throughout its life cycle. Therefore, this needs to include all parties who have any form of privileged access, whether they are permanent

---

**Figure 1—C T Security Baseline Steps Classified Against SDLC Phases**

<table>
<thead>
<tr>
<th>Domain/Step</th>
<th>SDLC Phase</th>
<th>Control Objective</th>
</tr>
</thead>
<tbody>
<tr>
<td>PO</td>
<td>Initiation</td>
<td>Define the security strategy and the information architecture.</td>
</tr>
<tr>
<td>1</td>
<td>Requirements</td>
<td>Define the IT organisation and relationships.</td>
</tr>
<tr>
<td>2</td>
<td>Design</td>
<td>Assess and manage IT risks.</td>
</tr>
<tr>
<td>3</td>
<td>Development</td>
<td>Communicate management aims and direction.</td>
</tr>
<tr>
<td>4-7</td>
<td>Implementation</td>
<td>Manage IT human resources.</td>
</tr>
<tr>
<td>8-10</td>
<td>Operation</td>
<td></td>
</tr>
<tr>
<td>Al</td>
<td>Design</td>
<td>Acquire and maintain application and technology infrastructure.</td>
</tr>
<tr>
<td>11 and 12</td>
<td>Implementation</td>
<td>Identify automated solutions.</td>
</tr>
<tr>
<td>13-15</td>
<td>Operation</td>
<td>Manage the physical environment.</td>
</tr>
<tr>
<td>16</td>
<td>Acquire</td>
<td>Manage changes.</td>
</tr>
<tr>
<td>17 and 18</td>
<td>Maintain</td>
<td>Install and accredit solutions and changes.</td>
</tr>
<tr>
<td>19 and 20</td>
<td>Manage</td>
<td></td>
</tr>
<tr>
<td>DS</td>
<td>Design</td>
<td>Define and manage service levels.</td>
</tr>
<tr>
<td>21</td>
<td>Implement</td>
<td>Manage third-party services.</td>
</tr>
<tr>
<td>22-24</td>
<td>Use</td>
<td>Enable operation and use.</td>
</tr>
<tr>
<td>25-33</td>
<td>Enable</td>
<td>Manage the configuration.</td>
</tr>
<tr>
<td>34 and 35</td>
<td>Use</td>
<td>Manage the data.</td>
</tr>
<tr>
<td>36-39</td>
<td>Manage</td>
<td></td>
</tr>
<tr>
<td>40 and 41</td>
<td>Maintain</td>
<td></td>
</tr>
<tr>
<td>ME</td>
<td>Acquire</td>
<td>Monitor and evaluate IT performance.</td>
</tr>
<tr>
<td>42</td>
<td>Maintain</td>
<td>Obtain independent assurance.</td>
</tr>
<tr>
<td>43</td>
<td>Manage</td>
<td>Ensure regulatory compliance.</td>
</tr>
<tr>
<td>44</td>
<td>Implement</td>
<td></td>
</tr>
</tbody>
</table>
full-time employees or not (e.g., developers, systems administrators, Internet service provider [ISP] staff, content administrators). Organisations need to ensure that the continued availability of the application and the ability to update and alter its functions do not depend on undocumented knowledge, especially if this is held by a third party.

• Steps 8-10—Assess and manage IT risks. Threat analysis techniques are available for applications. These can be used together with attack trees to identify possible scenarios that management can use to assess risk. As business objectives change, the web application must be reviewed to ensure that it meets the needs and that risk management controls are suitably selected.

ACQUIRE AND IMPLEMENT

The following are the steps related to the AI domain:

• Steps 11-12—Identify automated solutions. The responsibility to define a data classification scheme and identify which data fall into each category should not be delegated to the supplier of automated/third-party solutions. Specifications should be as explicit as possible in their requirements. The security expectations should be defined and mapped to business objectives, and potential suppliers should be asked to explain how they will meet these requirements and provide examples of previous systems. For packaged solutions, security should be evaluated on a system configured in the same manner as the proposed live system. The tests and results should be documented to identify deviations in the delivered system. Comparisons with published good practice should be considered.

• Steps 13-15. Acquire and maintain application and technology infrastructure. The infrastructure requirements should be defined in the specification. These will require security controls and should not conflict with each other (e.g., a mis-configured web firewall that causes a denial of service). Suppliers should be able to explain the required configuration and operating measures needed to protect the application infrastructure. Suppliers must provide a schedule of all components necessary for the infrastructure and processes to identify and monitor upgrades and patches. This should extend to all services required (e.g., Domain Name System [DNS], third-party web services such as address lookups and mapping). The effect of and process to apply these should be identified, and their effect on the availability of the system (e.g., reboot required, payment module unavailable) and other controls (e.g., antivirus software needs to be disabled to install) documented.

• Step 16—Enable operation and use. Additional procedures are required for operating applications. Even the simplest of web sites should be checked daily for uptime, errors, unapproved changes and defacement.

• Steps 17-18—Manage changes. A full change control process is necessary for applications. This requires having separate development and test environments that should match the production environment as far as practically possible. Development and test environments must not use live data. The impact of any differences with the production environment should be assessed before the changes are applied. It should always be possible to rebuild a web application from scratch on a new host with the backup data, documentation and referenced software/components/licences—making an identical configuration of the production system.

• Steps 19-20. Install and accredit solutions and changes. Security testing as well as other acceptance testing of the application should be undertaken. This would normally include network penetration and application penetration testing. Where possible, the test environment should be identical to the production environment. It may be possible to perform a source code or binary code review. These processes should be defined and efficient enough to ensure that time-to-market is not impacted by delays in installation and accreditation.

DELIVER AND SUPPORT

The following are the steps related to the DS domain:

• Step 21—Define and manage service levels. Ensure that service level agreements (SLAs) with internal and external service providers are clearly defined. In particular, definitions for uptime should explain the testing frequency, testing locations, number of failures and time-outs to count as downtime, retest delay time, and what scheduled downtime is excluded from the calculations. SLAs must not be contrary to security requirements. Responsibilities must be defined, and procedures must be in place for handling SLA-compliance failures. Where compensation will not make up for potential losses, other methods for business continuity should be considered.
• Step 22-24—Manage third-party services. The security capabilities of every party required to develop and operate the application should be assessed. Where security of the application is dependent upon third parties, how the risks may be managed, reduced or removed should be considered. The organisation’s right to audit and undertake security testing of all networks and devices associated with the application must be ensured.

• Steps 25-33—Ensure continuous service. The application may be a key business service. For any part of the application that is identified as a critical business function, additional effort should be made to maintain its availability. It may be possible to provide alternative processes for a short duration, but the resources for recovering from a prolonged incident should be analysed. In a catastrophic incident affecting, say, the hosting location, organisations should have enough data and information available to rebuild the application at an alternative location. This plan should include a regularly updated list of contacts at related parties whose services interact with the web application or are necessary for its function. Consideration should be given to how to remap domain names in such an event. The location and access to offsite data backups should be reviewed, and methods to restore data to a recovery site tested. Any Internet Protocol (IP)-specific security measures (e.g., restrictions on submitting payment authentication requests) should be identified, and procedures should be created for their change at short notice and out of normal hours. Network-, system- and application-level intrusion detection systems should be installed, configured and monitored, and documented procedures should be in place for incident reporting. Methods for the transfer of application code and data from the servers (e.g., backups, data extracts, file copying, screen reports, logs) must be defined, and access controls to limit who can undertake these tasks, what data they can contain and logging to identify when they occur must be implemented.

• Steps 34-35—Manage the configuration. The inventory of IT hardware should include all network devices such as traffic balancers, network firewalls and other firewalls (e.g., Extensible Markup Language [XML], web application, database) and connected systems (e.g., servers in a different DMZ or internally). The software inventory should include all scripts, components and settings, including those for the operating system itself. Where possible, a standard build configuration or image should be available for the base operating system load. Some devices may be shared (e.g., firewalls) with others and it may not be possible to have full details of the configuration, but any settings specific to the hosts/application should be managed. Changes to the configuration (including changes to scripts) should be automatically detected with the appropriate individual alerted.

• Steps 36-39—Manage the data. Data integrity should be ensured through database design, user identification, good access control and complete validation of data before use. Only complete transactions should be committed to databases, and any errors should cause a rollback to the previous state. Different database connections should be used for unauthenticated, authenticated and administrative users, and permissions should be restricted to what is necessary. The application should maintain data in compliance with suitable retention and disposal requirements—these may be internal, regulatory or legal. In particular, data flowing to/from another country should be checked—is the transfer permitted and what controls are required?

• Steps 40-41—Manage the physical environment. For applications hosted by third parties, the premises should be visited to ensure adequate physical security and protection from damage, theft and accidental loss. Who has physical access to the equipment and how their access is controlled and logged should be examined.

MONITOR AND EVALUATE

The following are the steps related to the ME domain:

• Step 42—Monitor and evaluate IT performance. Knowledge of current vulnerabilities in software and hardware included in the application should be kept up to date. The suppliers of any custom software must provide security and other updates as required. Traffic going to and being sent by the application should be monitored and assessed, and monitoring and security event handling should be reassessed as a result. Regular and intermittent (upon major changes) vulnerability assessments of the application should be undertaken.

• Step 43—Obtain independent assurance. Periodically, the web application should be assessed independently by security and legal specialists to ensure that the control mechanisms are up to date for the current threats and exploits, the application is compliant with current legislation.
(e.g., corporate, data protection), and contracts with third parties are operating as intended and are effective.

- **Step 44—Ensure regulatory compliance.** Whether the controls meet the regulatory requirements identified in step 1 (e.g., are the site licence/terms of use consistent with the business policies, privacy statement and copyright infringement notice?) must be checked. Apart from legal requirements, there may be contractual or insurance requirements that need to be checked. Existing guidance on assessments and audits relevant to web applications should be examined."11, 12, 13, 14 Intellectual property associated with visual design (e.g., photographs, fonts, logos, copyright of the design), content (text, other media, user-uploaded content), services consumed and software (open source licences, components) are additional issues for most web applications.

**CONCLUSIONS**

Web applications can be complex when they interact with multiple systems that have varying levels of trust and there are very few established practices for their successful governance. For many organisations, web-enabled systems are their first foray into commissioning and operating bespoke software, and the suppliers are often not information-security-minded. CoBIT is an internationally accepted standard for IT governance and control practices. CoBIT Security Baseline provides a thorough, low-barrier entry point for SMEs/SMBs to assess the current risks, identify improvement goals aligned with business objectives and implement controls for their web applications. A practical approach using CoBIT Security Baseline will help prioritise the implementation of controls."15

Once begun, step-by-step improvements will reduce business risk and the work may also encourage organisations to extend the approach to other systems. Organisations may also consider proceeding further with the full CoBIT framework or use available mappings"16 to integrate the controls discussed here with existing frameworks and standards, such as PRINCE2, ISO/IEC 27000 series and IT Infrastructure Library (ITIL).

**ENDNOTES**

5. Open Web Application Security Project (OWASP), A Guide to Building Secure Web Applications and Web Services 2.0
7. Ibid., OWASP, A Guide to Building Secure Web Applications and Web Services 2.0; and National Infrastructure Security Co-Ordination Centre
8. Tracy, Miles; Wayne Jansen; Karen Scarfone; Theodore Winograd; NIST SP 800-44 V2, Guidelines on Securing Public Web Servers, National Institute of Standards and Technology, USA
9. Ibid., National Infrastructure Security Co-Ordination Centre
17. IT Governance Institute, CoBIT Mapping series, www.isaca.org/cobitmappings