INFORMATION SYSTEMS & TECHNOLOGY

APPLICATION SECURITY[1]

BACKGROUND

Credit unions rely on applications to ensure accurate, timely, and confidential processing of data. Vulnerabilities,[2] particularly those associated with web-based applications,[3] are increasingly the focus of attacks from external and internal sources for the purpose of committing fraud and identity theft.

Web-based applications are most frequently associated with Internet banking, bill payment, credit cards, electronic loan and member applications, and cash management services, but they do provide the mechanism for many other web services. Web-based applications are being targeted for several reasons including:

- Easy Internet access by all (members, non-members, employees, servicers, and vendors);
- Traditional network defenses (firewalls, intrusion detection/prevention systems) may not detect or prohibit unauthorized activity;
- Breached applications may provide perpetrators unauthorized access to sensitive data that supports the application (member nonpublic personal information,[4] financial records); and
- Known vulnerabilities due to weaknesses in application development, testing, and quality assurance[5] processes.

Some companies which develop applications mitigate application security risks by incorporating security into the development and quality assurance processes. Those processes include rigorous testing and well-defined recurring processes to identify and monitor vulnerabilities, notify users of vulnerabilities, and provide remediation or corrective measures (e.g., patches, etc.). Applications developed internally by the credit

[1] The Office of the Comptroller of the Currency issued OCC Bulletin 2008-16 on May 8, 2008 on Application Security. NCUA has expanded on the guidance contained in the OCC Bulletin in this Risk Alert and Enclosure.
[2] See Appendix A – Common Vulnerabilities in Web-Based Applications.
[3] A web-based application is an application that is accessed via a web browser over a network such as the Internet or an intranet.
[4] NCUA Rules and Regulations, Part 716.3(q)(1).
[5] Quality assurance processes refers to planned and systematic production processes that provide confidence in a product's suitability for its intended purpose. It is a set of activities intended to ensure the product satisfies customer requirements in a systematic, reliable fashion.

union, vendor-acquired,[6] or contracted [7] by the credit union, may not be subject to similar risk mitigation measures and may expose a credit union to increased risks.

Two FFIEC Information Technology Examination Handbooks, *Information Security* and *Development and Acquisition,* [8] provide basic guidance regarding application security. This letter expands on existing guidance and emphasizes the need for comprehensive application development and quality assurance processes integrating security for all applications.

RISK MANAGEMENT CONSIDERATIONS

As part of the information security program, management should determine all applications are developed and maintained in a manner that appropriately addresses risks to the confidentiality, availability, and integrity of data. Management should include application security in the risk assessment which is required by the National Credit Union Administration Rules and Regulations, Part 748, Appendix A.[9] The scope of management's application security efforts may vary depending on its size and complexity and the nature of its software applications.

Key factors to consider in the risk assessment of an application include:

- Internet accessibility of the application;
- Ability to process or have access to sensitive data;
- Source of the application's development (e.g., in-house, vendor-acquired, or contracted);
- Extent of the secure practices used in the application's development process;
- Existence of an effective, recurring process to monitor, identify, and remediate or correct vulnerabilities (e.g., patch program, etc.); and
- Existence of a periodic assurance process (e.g., independent audit, code review, etc.) to independently validate the security of the applications.

Credit unions that purchase applications typically rely upon the vendors to provide secure applications. However, management remains responsible for ensuring the application meets the credit union's security requirements at acquisition and thereafter.

---

[6] Vendor-acquired software includes purchased "commercial off-the-shelf" credit union applications and those developed by a technology service provider, corporate credit union, credit union service organization (CUSO), or other vendor.

[7] Contract for software is defined as that which is written for a credit union based on a defined contractual arrangement.

[8] The FFIEC IT Examination Handbooks can be found at http://www.ffiec.gov/ffiecinfobase/html_pages/it_01.html.

[9] Appendix A to Part 748 of the NCUA Rules and Regulations can be found at http://www.ncua.gov/RegulationsOpinionsLaws/rules_and_regs/NCUA%20R%20&%20R%20Book%201%20May%208%202008.pdf.

Management should expand the vendor management due diligence program for purchased software to include application security considerations in the request for information (RFI) or request for proposal (RFP) process. An attestation in writing from the vendor that the software development process follows secure development practices and is periodically tested may suffice for some applications. For applications that present higher risks,[10] management should require vendor evidence of adhering to sound processes and validation through third-party testing and/or audits. All applications purchased should be supported by appropriate vulnerability identification and remediation processes, including appropriate vendor support.[11] Additionally, credit union management should ensure the ongoing testing process (e.g., penetration tests, vulnerability assessments) includes in-house developed, vendor-acquired, and contracted for applications. Management should consider the following when evaluating applications for purchase:

- What are the vendor's risk-based processes for development and validation of the application security before, during, and after it is purchased?
- What are the vendor's notification processes whenever security vulnerabilities are identified by the vendor, reported by credit unions, other customers, or reported in media by others?
- Will the vendor provide timely mitigation or remediation solutions to identified security vulnerabilities?
- Does the vendor have an industry-recognized third party who conducts application vulnerability assessments on the applications, including security? If so, credit union management should before purchase or during the RFI/RFP process:
    - obtain the third party's name,
    - determine how often the assessment is conducted,
    - determine the date of the last assessment,
    - secure a copy of the most recent assessment, if possible,
    - determine whether the application has any known open vulnerabilities,
    - determine the nature of the vulnerabilities, and
    - determine if the vendor is willing to share its secure coding processes and practices.

[10] NCUA Letter to Credit Unions 05-CU-18, Guidance on Authentication in an Internet Banking Environment (November 2005), http://www.ncua.gov/letters/2005/CU/05-CU-18.pdf.

[11] Vendor support should include any changes to applications necessitated by security updates to other required software, such as operating systems.

- If the vendor does not have a third party who conducts application vulnerability assessments, including security, can the vendor describe their internal methodology?[12]
- Is the vendor willing to conduct, or contract for, an assessment to provide assurance to the credit union regarding the security of the application?
- Where appropriate, management should include in the contract language the need for current and ongoing application vulnerability assessments, including security, and who will conduct the assessments. Depending on the risk profile of the application, management may request the full vulnerability assessment report or a summary.

To ensure maximum effectiveness, credit unions that develop applications in-house should consider following an enterprise-wide effort coordinated across business lines and include the following elements:

- Incorporate appropriate attack model/threats[13] in the risk assessment to assist in determining the security and assurance requirements for the application;
- Analyze the environment in which the application will reside. As the environment changes, the security requirements and assurance needs for the application may also change;
- Ensure any open source application[14] are also subject to appropriate development and assurance processes;
- Ensure appropriate personnel (i.e., management, developers, security, and auditors) are trained sufficiently to understand and be aware of risks associated with the credit union's technology environment;
- Engage in periodic application testing or validation based on a current risk assessment to ensure the ongoing and appropriate protection of transactions and member data. Testing consideration may include:
    - Static, dynamic, and functional[15] evaluations, depending on the type and criticality of the application;

[12] Credit unions can utilize the guidance contained in the FFIEC IT Handbook Development and Acquisition (http://www.ffiec.gov/ffiecinfobase/html_pages/d_a_book_frame.htm) as their guide when assessing the vendor's methodology.

[13] The attack models/threats should address potential attacks on the entire system, including endpoints and end users who enter and/or retrieve information.

[14] Issues related to open source software are addressed in NCUA Letter to Credit Unions 04-CU-14, Risk Management of Free and Open Source Software, November 2004. (http://www.ncua.gov/letters/2004/04-CU-14.pdf and enclosure http://www.ncua.gov/letters/2004/04-CU-14-Encl-1.pdf)

[15] Static testing is typically associated with code review during the development process, also known as "white box" testing. Dynamic review is performed while the code is being executed in either a lab or production environment, also known as "black box" testing. Functional testing validates what should occur given an input or action by the user.

- Automated evaluations using commercial or freeware tools, as well as manual interaction to supplement application tools;
- Authenticated and non-authenticated user scenarios;
- Comprehensive testing in a simulated production environment including appropriate operating systems and associated databases. The weakest link in several connected components may expose the entire system to compromise;
- Implementation of lessons learned iteratively[16] throughout the development and periodic testing process; and
- Identification and monitoring of developed applications for vulnerabilities through an ongoing and defined process that includes appropriate communications and remediation.

Credit unions are encouraged to leverage upon available resources to assist in risk identification and improve application security practices. Software tools, industry resources, specific certifications, and education courses are available to provide assistance to enhance the credit union and technology service providers' application development architecture (e.g., software development lifecycle and assurance processes). The National Institute of Standards and Technology (NIST) Special Publication 800-64,[17] Security Considerations in the System Development Life Cycle, provide resources to assist in applications security efforts.

Additionally, management should establish a mechanism to receive and respond appropriately to vulnerability reports from public and private sources, such as, US-CERT[18], and any other groups that detect, analyze, and report vulnerabilities.

[16] Iteratively in computing is the repetition of a process within a computer program.
[17] NIST SP 800-64, Security Considerations in the System Development Life Cycle (http://csrc.nist.gov/publications/nistpubs/800-64-Rev2/SP800-64-Revision2.pdf).
[18] United States Computer Emergency Readiness Team (http://www.us-cert.gov/).

APPENDIX A
Common Vulnerabilities in Web-Based Applications

VULNERABILITY	DEFINITION
Cross Site Scripting (XSS)	XSS is a type of computer security vulnerability typically found in web applications which allow code injection by malicious web users into the web pages viewed by other users. XSS allows attackers to execute script in the victim's browser that can hijack a users session, deface a web site, and possible introduce worms, Trojans, and other malware.
Injection Flaws	Injection flaws, particularly Structure Query Language (SQL), allow attackers to relay malicious code through a web application to another system. Injection occurs when user-supplied data is sent to an interpreter as part of a command or query. The attacker's hostile data tricks the interpreter into executing unintended commands or changing data.
Malicious File Execution (MFE)	Code which is vulnerable to remote file inclusion, a technique often used to attack Internet websites from a remote computer, which allows attackers to include hostile code and data. This could result in attacks that result in total server compromise. MFE attacks affect PHP, XML, and any framework which accepts filenames or files form users.
Insecure Direct Object Reference	A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, database record, or key, as a URL or form parameter (i.e., form parameters store retrieved information that is included in the HTTP request for a web page). Attackers can manipulate those references to access other objects without authorization.
Cross Site Request Forgery (CSRF)	A CSRF attack forces a logged-on victim's browser to send a pre-authenticated request to a vulnerable web application, which then forces the victim's browser to perform a hostile action to the benefit of the attacker. CSRF can be as powerful as the web application that it attacks.
Information Leakage and Improper Error Handling	Applications can unintentionally leak information about their configuration, internal workings, or violate privacy through a variety of application problems. Attackers use this weakness to steal sensitive data, or conduct more serious attacks.
Broken Authentication and Session Management	Account credentials and session tokens are often not properly protected. Attackers compromise passwords, keys, or authentication tokens to assume other users' identities.
Insecure Cryptographic Storage	Web applications rarely use cryptographic functions properly to protect data and credentials. Attackers use weakly protected data to conduct identity theft and other crimes, such as credit card fraud
Insecure Communications	Applications frequently fail to encrypt network traffic when it is necessary to protect sensitive communications.
Failure to Restrict URL Access	An application only protects sensitive functionality by preventing the display of links or URLs to unauthorized users. Attackers can use this weakness to access and perform unauthorized operations by accessing those URLs directly.