Keep Your Data Secure by Staying Updated with Toad® by Quest®

A technical brief on the software security standards built into Toad for Oracle

By Julie Hyman, Senior Product Manager, and Ryan Crochet,
Sr. Product Marketing Manager, Quest Software

In today's climate of increased cybersecurity threats, it is crucial for IT administrators to draw a clear line when it comes to the software they allow on their desktops and network, regardless of existing security protocols or architecture in place. The unique combination of increased exotic ransomware attacks and malicious activities combined with the reliance of organizations to utilize a wider array of software and applications to achieve business outcomes requires a proactive, unyielding commitment to software management. When it comes to protecting your organization's digital assets, top priorities require (at a minimum) that:

1. Only secure software is installed
2. Software is continually up to date to keep systems secure

In this technical brief, you will find support for both imperatives offering reassurance towards security in your supply chain software.

Keep reading for more detail about:

- **What Software Supply Chain Security is and why it matters to you**
- **The importance of keeping software up to date**
- **Crucial security controls that help Toad keep you secure**

SOFTWARE SUPPLY CHAIN SECURITY

Security in software supply chain refers to the integrity, confidentiality and availability of software components and their dependencies throughout the software development lifecycle, from creation to deployment. It is important because compromised software components can harm organizations and their customers. Effective software supply chain security measures help to minimize these risks by providing visibility, control and assurance throughout the entire process.

DATA SECURITY AND TOAD

Toad by Quest is an Integrated Development Environment (IDE) that enables developers and database administrators to create, manage and maintain databases, including roles and users. So, if an attacker gains access to a user's Toad session, they may be able to execute unauthorized database commands, steal sensitive data or even bring down the entire database. Therefore, it's crucial to ensure that Toad and its associated components are regularly updated, properly configured and protected with appropriate security controls to mitigate these risks. Note: Irrespective of architecture and security protocols in place, the critical components to keep up to date are Operating Systems and the Database Platforms themselves. By proximity, this also includes Database IDE's, as any security vulnerabilities within an IDE or its associated components can potentially compromise the security of the entire database environment.

APPLICATION SECURITY AND UPDATING TOAD

Next, think about your role in keeping the network secure by continually updating your software. The ransomware attacks allowed by the WannaCry/Petya vulnerability demonstrated that hundreds of thousands of computers worldwide were running outdated software. (Many of them probably still are.) Although the fix for WannaCry was distributed as part of a normal update to Windows, too many companies did not apply the update. Despite months of warnings, the Petya attack was a second wave against the same vulnerability, affecting computers that had not yet been updated.

Granted, some IT groups delay updates that force a system restart and interrupt productivity. Some worry about how updates will affect applications upon which thousands of their users depend. Others give priority only to necessary updates and make their own call as to what "necessary" means. Still others have an "ain't-broke-don't-fix-it" attitude towards updating Toad and other development software.

Identified vulnerabilities

Below is an example of why users need to be mindful and use the latest supported version. A number of years ago, as part of performing regular checks during the build/release process, Quest identified and addressed a vulnerability in Toad for Oracle v13.1.1, along with the following notification:

"Quest recently patched one of these [third-party] libraries, which contained a critical (CVSS 9.3) vulnerability that was discovered. The vulnerability affected the Microsoft Visual C++ library and could lead to a privilege escalation. This vulnerability has been remediated with our latest release of Toad."

If you're reading this and thinking, "It works with the version of Oracle I need it to work with… what's the problem?"

It works with the version of Oracle I need to work with… what's the problem?"

Firstly, isn't it great that a piece of software released ten years ago works with a database that was released nine years later? Forward compatibility is rare today.

Secondly however, this is not best practice from a cybersecurity perspective, no matter what architecture is employed or what security protocols or firewalls are in place. Using software with discontinued support can leave the software vulnerable to security threats without any updates or patches, which can compromise the integrity and confidentiality of the data it interacts with. The sources of cyber threats are increasing as the growth in employing open-source libraries in proprietary software increases.

In 2018, a study by Fossa found that 92% of software projects use open-source components, and that the average application contains 57% open-source code. Similarly, in 2019, Synopsys found that open-source code accounted for an average of 70% of the code in the proprietary applications it analyzed.

Then, take the Log4j crisis of 2021. The Log4j cyberthreat was a result of a vulnerability in the Apache Log4j library, a widely used open-source logging tool for Java applications. This vulnerability allowed attackers to remotely execute malicious code on affected systems, potentially leading to data theft, ransomware attacks and other security breaches.

The log4j vulnerability affected a vast number of organizations worldwide, and the speed and scope of its exploitation raised concerns about the security of

open source software and the need for more robust cybersecurity measures.

In short, without a software bill of materials, you can never truly know the dependencies the software has, and you can never truly know where the next threat comes from. The only assurance you have is using supported software with patches that become available when the next threat arises.

So, to ask again, do you know what version of Toad you use and if it is supported? Quest has continually released newer, more secure versions. As an example. Table 1 lists the partial release history for Toad for Oracle. Anything less than version 15.0 has discontinued support.

Newer versions of Toad products, like Toad for Oracle, extend far beyond functional improvements to include substantial security improvements. Toad's evolution through our release process closely follows the ever-changing landscape of vulnerabilities and threats in database development, including:

- Insecure (unencrypted) connections to a database
- Unintended privilege escalation (e.g., low-privilege user to superuser)
- Incorrectly low barriers to accessing personally identifiable information (PII) and sensitive data stored in the cloud
- PII stored (and forgotten) in non-production copies of databases

At least twice a year, Quest's customers have the opportunity of updating Toad to the current version and adding security improvements that mitigate threats like those.

Toad for Oracle Versions and Support

Release	Support	General Availability Date
16.3	**Full Support**	07-Apr-2023
16.2	**Full Support**	30-Sep-2022
16.1	**Limited Support**	29-Jun-2022
16.0	**Limited Support**	26-Apr-2022
15.1	**Limited Support**	27-Jan-2022
15.0	**Limited Support**	18-Oct-2021
14.2	**Support Discontinued**	13-Jul-2021
14.1	**Support Discontinued**	30-Mar-2021
14.0	**Support Discontinued**	23-Oct-2020

Table 1. Earlier product versions not listed are considered discontinued. Please refer to Toad for Oracle's version support table for the most up to date information. (HERE)

SECURITY CONTROLS FOR PREVENTING SUPPLY CHAIN ATTACKS

In today's complex software environment, IT administrators must contend with a range of security challenges, including the need to address vulnerabilities across multiple applications and platforms. As the use of open source libraries becomes increasingly widespread in software development, there is a growing imperative to identify and manage dependencies within the software supply chain.

Effective management of software dependencies is essential for ensuring system security and mitigating the risk of cyberattacks or legal complications related to licensing issues. To achieve this, organizations must prioritize supply chain security and implement strategies for managing and monitoring dependencies throughout the software development lifecycle.

Toad's evolution closely follows the everchanging landscape of vulnerabilities and threats in database development.

Quest sees growing concern about supply chain attacks, in which malware is introduced before products are released to customers. A series of scanning controls is in place to follow software security standards for identifying and eliminating malware in the build process of Quest's Toad products, including Toad for Oracle. The controls ensure that products remain free of vulnerabilities and malware, do not contain hidden backdoors and are developed by employees and contractors acting with integrity. Table 2 shows the security controls which every release of Toad undergoes.

Security Control	Description
1. Security Training	Developers, managers and directors are required to take 2.5 hours of assigned Security Training.
2. Secure Software Development Lifecycle	Development lifecycle is based on best practices of Quest's Information and Systems Management (ISM) business unit.
3. Third-Party Software	Bundled third-party software is checked for vulnerabilities in a standard process before application release.
4. Vulnerability Scanning	All software is scanned for vulnerabilities with an industry-standard SAST/DAST product.
5. Penetration Testing (Third-Party)	Products undergo third-party penetration testing annually.
6. Malware Scanning	All products are scanned for malware before release with two independent, industry-standard anti-malware scanners.
7. Code Signing	Software distributed to customers is cryptologically signed using Quest's official signing key to validate authenticity.
8. Software Integrity	Checksums for software installers are published so customers can ensure integrity of distributed software.
9. FIPS Compliance	Using cryptographic algorithms approved by FIPS, sensitive data is protected at rest and in transit.

Table 2: Toad security controls against supply chain attacks

- Software Security Threats
- Secure Software Design
- Secure Coding
- Testing for Security
- Security and Risk Overview
- Web Client Server Interaction Code Issues
- Thick App and Client-Server Interaction Issues
- Crypto and Security Misuse Issues
- Security in the Software Development Lifecycle

Following are details of how Quest Software implements each control for Toad Products:

1. SECURITY TRAINING

The goals of the training are to develop a baseline of security knowledge, improve secure coding practices and improve adherence to the Quest Secure Software Development Lifecycle (see below). Training topics currently covered include:

- Understanding software security
- Software security threats
- Secure software design
- Secure coding
- Testing for security
- Security and risk overview
- Web client server interaction code issues
- Thick app and client-server interaction issues
- Crypto and security misuse issues
- Security in the software development lifecycle

Quest tracks completion of each course and retains records for use during customer audits. The records demonstrate that Quest Engineering teams receive training in secure development practices.

2. SECURE SOFTWARE DEVELOPMENT LIFECYCLE (SSDLC)

The SSDLC for Toad products introduces security and privacy considerations to the process of developing the product itself. The SSDLC is designed to help Toad engineers write secure software, comply with software security standards and keep engineering costs low. The components and phases in the SSDLC include:

- Define security requirements
- Define metrics and compliance reporting
- Perform threat modeling
- Define and use cryptography standards
- Manage risk of using third-party components
- Establish a standard incident response process

3.THIRD-PARTY SOFTWARE AND COMPONENT PATCHING

Like many vendors of enterprise software, Quest often incorporates code written by other companies (“third parties”) so that Quest does not have to recreate it from scratch. But over time, vulnerabilities that require patching turn up in some third-party components. Besides relying on the NIST CVE Database for information on current vulnerabilities, Quest uses tools to identify third-party DLLs and any vulnerabilities that may have been identified in those DLLs. Quest has also developed a process for reducing the likelihood of releasing software with vulnerable third-party components. The process:

- Specifies criteria for no-ship vulnerabilities
- Requires an inventory of all third-party components, software and DLLs bundled into products
- Mandates regular checks for vulnerabilities in third-party components
- Establishes deadlines (30/60/180 days) for third parties to patch vulnerable components
- Mandates notices to customers if products are patched and a new release is issued

4. VULNERABILITY SCANNING (SAST AND DAST)

Quest has defined processes around SAST (Static Application Security Testing) and DAST (Dynamic Application Security Testing) scanning, using external tools.

SAST is a form of white-box testing. A tester using SAST examines the application from the inside, searching its source code for conditions that indicate potential security vulnerabilities. DAST is a form of black-box testing, from the outside as an attacker would see the application. A tester using DAST examines a web application when it is running and tries to hack it as an attacker would.

SAST tools assist in identifying weaknesses found on a list known as the common weakness enumeration (CWE). The tools are limited in their handling of issues of logical flow, authentication and authorization, which are better suited to penetration tests (see below) or manual source code reviews. DAST scanners, interacting with a web application from the outside, rely on HTTP and are technology-independent.

For Toad products, the process includes the following:

- Full scans of the product/web application are performed automatically where possible, at least twice per year and before all releases, using the currently designated SAST/DAST tool
- All product code developed by Quest is scanned by the SAST/DAST tool
- The Toad security advocate reviews the results of each scan and, when necessary, works with the InfoSec Principal Engineer to determine the severity of any issues found
- No products are released with critical or high vulnerabilities
- For medium and low vulnerabilities, the security advocate works with the product architect
- The security advocate and the product architect determine the best solution to any vulnerabilities uncovered by the scans

SAST examines the application from the inside, searching source code for potential security vulnerabilities. DAST is a form of black-box testing, from the outside as an attacker would see the web application

5. PENETRATION TESTING (THIRD-PARTY)

Penetration testing demonstrates real-world impact if a vulnerability or process weakness were to be exploited. It is designed to assess security before a bad actor strikes. A penetration test is not an automated scan of an application or its source code, but a next step after automated vulnerability scanning (see above).

For Toad products, penetration testing is conducted annually. The tests, a combination of manual and automated testing, are designed to uphold software security standards in the following areas of the product:

- Application logic
- Code injection
- Local storage
- Binary exploitation and reverse engineering
- Excessive privileges
- Unencrypted storage of sensitive information
- Unencrypted transmission of sensitive information
- Weak encryption implementations
- Weak assembly controls
- Weak GUI controls
- Weak or default passwords

In most cases, a penetration test follows a specific framework depending on the target application or infrastructure, and tactics vary depending on the adversary being mimicked.

The Case for Updating Toad

As a result of penetration testing, Quest identified and addressed a vulnerability in Toad for Oracle v13.3, then notified customers as follows: “During a recent penetration test an issue was discovered in which a user was not notified when connecting to a database insecurely. If a user unknowingly connects to a system over a non-encrypted link, an attacker can capture a user’s credentials or all data transferred across the connection. Quest has remediated this weakness.”

6. MALWARE SCANNING OF SOFTWARE BUILDS

When Toad product builds are packaged for release, they are first scanned for malware following a consistent process that includes these steps:

- Install packages are hashed using the SHA-256 algorithm before scanning, and the hash must match the final, published hash on the package.

All software builds are scanned for malware before they are released for install.

- All files packaged into an installer are first scanned for malware.
- Malware scanning is automated (command-line process or script).
- Independent malware scanners (two for Windows and two for Linux) are used, updated with the latest signatures/definitions before the scan.
- An evidence record including time, date and results is produced and permanently retained for each scan.
- The security and engineering directors must approve exceptions for any files in the package.

The process addresses multiple software security standards, including:

nstall packages are hashed using the SHA-256 algorithm before scanning, and the hash must match the final, published hash on the package

- All software builds are scanned for malware before they are released for install
- All files packaged into an installer are first scanned for malware
- Malware scanning is automated (command-line process or script)
- Independent malware scanners (two for Windows and two for Linux) are used, updated with the latest signatures/definitions before the scan
- An evidence record including time, date and results is produced and permanently retained for each scan
- The security and engineering directors must approve exceptions for any files in the package

7. & 8. CODE SIGNING AND SOFTWARE INTEGRITY

An application bearing Quest’s code signing certificate is a customer’s assurance that Quest created the application and that the software can be trusted. The code signing process serves the goal of ensuring authenticity by verifying the author of the software. It also ensures the integrity of the software by demonstrating that the code has not been altered

since it was signed. Code signing also plays a role in releasing updates and patches. When Quest signs an update to a Toad product with the same key used in the original application, it means that the update can be trusted; it couldn't have come from any source other than Quest. Finally, the checksums generated in code signing assure users that they have received the correct file, rather than a file that has been signed with a stolen key. All major operating systems and web browsers support code signing to prevent the distribution of malicious code.

Quest signs every .exe and .dll file included in the installer, along with binaries packaged with the application and installer files themselves.

9. PROTECTING SENSITIVE DATA THROUGH FIPS COMPLIANCE

The presence of sensitive data in your applications and databases imposes a burden of protection. Sensitive data extends to almost any kind of data you would want to prevent from falling into the wrong hands, including:

- Network credentials
- Passwords
- Social Security numbers
- Credit card information
- Personally identifiable information (PII) such as names, addresses and phone numbers
- Personal health information (PHI)
- Financial information
- Internal records
- Intellectual property

The Family of Toad products protect sensitive data through cryptographic algorithms that conform to the Federal Information Processing Standards (FIPS) of the United States Government. Furthermore, they apply that protection to sensitive data both in transit and at rest.

The FIPS standards specify the best practices and requirements for cryptography-based security systems, including methods for encryption and for generating encryption keys. FIPS compliance is mandatory for all computers used for U.S. government work and extends to testing outside applications (like Toad) that will run on U.S. government computers.

All Quest products comply with FIPS approved algorithms for encryption and hashing. The current status of FIPS compliance (currently FIPS 140-2) is validated prior to each release.

Toad products are built using one or more of the following cryptography service providers and libraries/classes:

- SHA-256 (.NET)
- DSA (.NET)
- RSA (.NET)
- ECDSA (.NET)
- AES (.NET)
- Java Cryptography Class

All Quest products comply with FIPS approved algorithms for encryption and hashing. The current status of FIPS compliance (currently FIPS 140-2) is validated prior to each release.

The Case for Updating Toad

As a result of FIPS 140-2 compliance testing, Quest identified and addressed a vulnerability in Toad for Oracle v13.2, then notified customers as follows:

"Quest was made aware of an active exploit in the wild that allowed an attacker to decrypt credentials saved in the Toad product. The exploit, which took advantage of the way Toad encrypted passwords, could be used to decrypt and use credentials to compromise affected databases, FTP servers or SSH servers. Quest has since implemented strong encryption, patched this vulnerability and released a nonvulnerable version of the software."

ADDRESSING SOFTWARE SECURITY STANDARDS

	Security Training	3rd Party Software and Component Patching	Vulnerability Scanning	Malware Scanning of Software Builds	Code Signing and Software Integrity
NIST SP 800-53 R4	✓	✓	✓	✓	✓
ISO 27001	✓	✓	✓	✓	
PCI DSS v3.0	✓	✓	✓		
PCI 1.4				✓	
AICPA TSC 2014		✓	✓		
AICPA TSC (SOC-2)				✓	
HIPAA 45 C.F.R.		✓	✓	✓	

CONCLUSION

The security controls described above are designed and applied to mitigate the risk of supply chain security in Toad products. Figure 2 illustrates their flow.

Toad products are valued and trusted, and they have unlocked millions of hours of productivity gains for database professionals. By updating Toad products, you ensure a continual flow of new features and adherence to software security standards. You also receive full technical support and keep your organization's security profile tight.

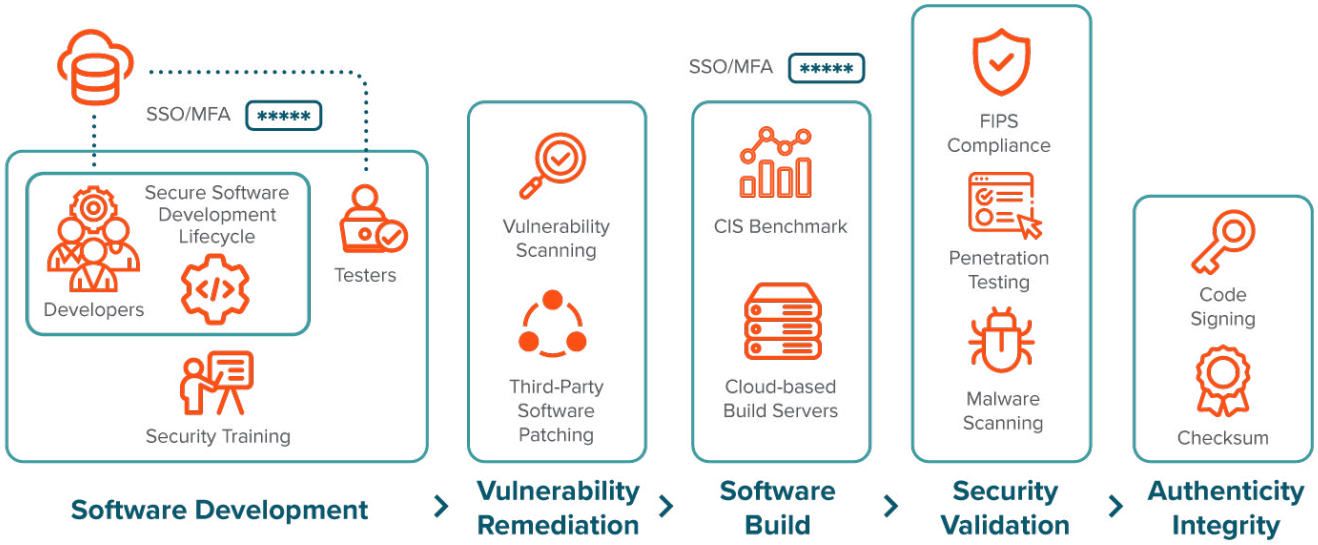

Figure 2: Flow of security controls in developing Toad products

ABOUT THE AUTHORS

Julie Hyman is a senior product manager for the Database tools portfolio at Quest Software. She is a seasoned software product manager with 25 years of experience in creating and improving software products at startups and Fortune 500 firms. Julie has worked closely for years with DBAs, developers and analysts in all industries and many leading corporations to ensure Quest continues to provide world-class solutions.

Ryan Crochet is a Senior Product Marketing Manager in the Information and Systems Management division of Quest Software. Passionate about the market he serves and the problems that it faces regularly, Ryan continues to seek ways to address them while showcasing and evangelizing the power of data-driven decision making to align technical expertise and business outcomes.

About Quest Software

Quest Software creates technology and solutions that build the foundation for enterprise AI. Focused on data management and governance, cybersecurity and platform modernization, Quest helps organizations address their most pressing challenges and make the promise of AI a reality. Around the globe, more than 45,000 companies including over 90% of the Fortune 500 count on Quest Software. For more information, visit www.quest.com or follow Quest Software on X (formerly Twitter) and LinkedIn.

If you have any questions regarding your potential use of this material, contact:

Quest Software Inc.
Attn: LEGAL Dept
4 Polaris Way
Aliso Viejo, CA 92656

Refer to our website (www.quest.com) for regional and international office information.