CODE REVIEW

SECURE CODING

Developers are faced with constant pressure to produce new or modified code on a daily basis for organizations. The reality is, no code is **100% bug-free**. Organizations must ask themselves, what kinds of bugs are within their code? Are individuals' jobs on the line when they are expected to identify all bugs in the code before pushing to production? What if a value has been hardcoded, something that if the code was decompiled would reveal the SA (System Administrator) password to your SQL Server?

These are a select few our CyberArq experts look for when conducting a comprehensive code review. Our industry leading experts take a hybrid approach utilizing a combination of automated and manual assessments. To name a few our team inspects your organizational code for logic, security issues and any other areas where a vulnerability may exist if discovered and abused.

WHEN TO PERFORM A SECURE CODE REVIEW

Security should be a focus throughout the entire development life cycle. Creating threat models during the design phase, educating developers on secure coding practices, and performing frequent peer reviews of code with security personnel involved will all help increase the overall quality of the code and reduce the number of issues reported (and hence that need to be fixed) by the secure code review.

However, a secure code review is best used toward the end of the source code development, when most or all functionality has been implemented. The reason for waiting until late in the development phase is that a secure code review is expensive and time consuming. Performing it once toward the end of the development process helps mitigate cost.

SECURE CODING MANUAL & AUTOMATED

There are two primary limiting factors that can make a secure code review tricky: humans and automation. For a human, the limiting factor is the relatively limited lines of code that an expert individual can review in a work day. A human may be able to review several hundred lines of code in a day. Considering that modern software is often comprised of tens or even hundreds of thousands of lines of code, it is highly unlikely for a human to manually review every line of code. It would require nearly as many reviewers as developers to approach the process using manual methods alone.

Automated tools can review code much faster than humans. The trade-off, however, is that automation is far more prone to missing security implications (false negatives) as well as falsely identifying them (false positives). In addition, automated tools often don't understand the context in which code is written.

To overcome these limitations, a review should be performed through a combination of manual and automated efforts. Automated tools can quickly scan the code base to identify areas of interest and potential vulnerabilities. Triaging automated findings guides the manual investigation into those potential vulnerabilities. Manual reviews are also useful when reviewing the code for certain classes of flaws such as authentication and cryptography.

The best approach for a secure code review is to understand the advantages and disadvantages of each method and to incorporate both as appropriate.

SECURE CODING OBJECTIVES

- *Understand the developer's approach*
- *Do not assess level of risk*
- *Follow up on review points*
- *Use multiple techniques*
- *Focus on the big picture*
- *Stick to the intent of the review*