

# SECURITY MANAGEMENT IN SOFTWARE PRODUCTION

Well-thought-out security management in software production generates genuine added value – both for the customer and in development.

*by Marcel Vinzens and René Rehmann*

They are called FREAK, Shellshock, Poodle, Heartbleed and BEAST – serious security vulnerabilities in familiar standards and products, which have made headlines regularly in recent years. The National Institute of Standards and Technology (NIST) has recorded over 68,000 software defects known as Common Vulnerabilities and Exposures (CVEs) of various severity in its database. The reported number of these vulnerabilities has increased from around 1000 to 5000 – 8000 annually since the year 2000.

So what do these vulnerabilities and numbers mean for a company such as AdNovum, which produces custom software and software products and frequently also uses open-source software (OSS) as well as licensed closed-source components for its solutions?

The preceding figures and facts clearly show that developing secure software requires a significant effort. In principle, all participants in the Software Development Lifecycle (SDLC) need to assume shared responsibility for the development of secure software. Therefore, it is crucial to approach each phase of the SDLC with the right security mindset. This also includes implementing quality assurance measures and security mechanisms for design, development and distribution as well as monitoring in order to minimize the likelihood of exposure and the effects in case of an exploit. In the following, we show how security management is implemented in our software production which processes more than 100 customer projects annually with over 2000 software deliveries to customers.

## Secure software development as the basis

Software security needs to be incorporated in the design and development phases as a matter of principle. Numerous measures contribute to this at AdNovum: Architecture and security sign-offs, quality assurance in the form of manual code reviews and automatic code analyses during the development phase along with training, directives and best practices for developers. In particular, these measures are intended to prevent a situation where securi-

ty needs to be «integrated» after the fact because the developers focused exclusively on features and usability during software development. Our Security Engineering team is responsible for the control and further development of the measures. It is supported by other engineering teams and continuously takes feedback from software production into account.

## ADNOVUM EXAMINES NEW COMPONENTS FOR THEIR ADDED VALUE AND RISK.

As a central element of secure software development, AdNovum operates strict technology and central dependency management as well as using managed repositories for storing software artifacts. What does this mean in concrete terms?

Before AdNovum uses new software components in projects, a critical evaluation of the added value and risk is performed in the course of what are known as technical investigations that form part of the technology approval process. We carefully select third-party components (OSS and proprietary libraries) and store them in centrally managed repositories – provided the evaluation result is positive. All components for every customer solution and product are subject to technology management. We also define a lifecycle status for each component of every version at the operational level. Here we primarily differentiate between components that have been released for use («approved»), are being examined («investigating»), will be replaced («deprecated») and those that are no longer approved for use («forbidden»). Which components are used with what version is reviewed in the course of application development. For components with the status «forbidden» or «restricted»/«investigating» that are not approved for a project, the developer immediately receives an error message. This tells the developer at a glance which components may no longer be used, for example because of



*Marcel Vinzens and René Rehmann: Monitoring and assuring the security of AdNovum software.*





a known vulnerability. Information specifying the newer version to be used which has been freed of the vulnerability is also provided.

Each internal component, every customer solution and every product is built every night by what is called our NightlyBuild. For projects in wait mode, for example customer solutions with no active further development, we thereby review which components and versions are being used (dependency management). Naturally, the NightlyBuild also performs all tests defined for a customer project or product automatically. We therefore know at all times when a project is no longer current and secure because of lifecycle status changes in third-party components.

### ADNOVUM ASSUMES RESPONSIBILITY FOR THE MAINTENANCE OF INTERMEDIATE PRODUCTS AND LIBRARIES ON BEHALF OF ITS CUSTOMERS.

With today's customer projects and products, the objective for reasons of efficiency is to always use existing intermediate products and libraries when possible and sensible, and to «only» develop those components in-house that are either very specific (technical functionality) or result in a competitive advantage for products. AdNovum assumes responsibility for the maintenance of these intermediate products and libraries on behalf of the customer. This also includes for example that we inform customers of the OSS we use in a customer solution. We also notify customers when AdNovum software or OSS used in the same is affected by a security vulnerability. Our security management is responsible for monitoring these components and launching an alerting pro-

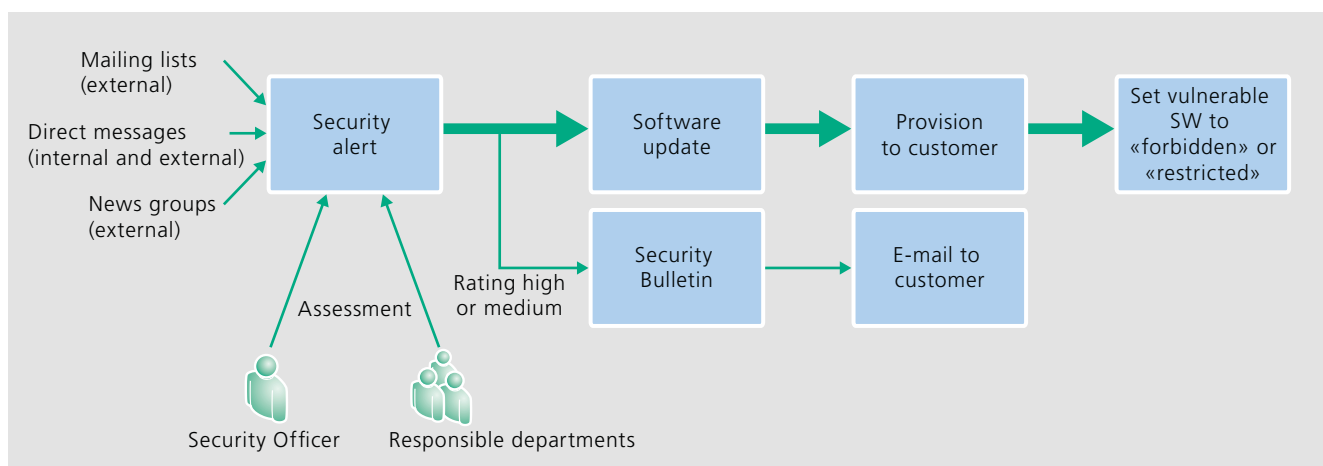
cess as needed. This alerting process regulates how vulnerabilities that are discovered need to be handled and ensures compliance with internal directives.

### AN ALERTING PROCESS REGULATES HOW VULNERABILITIES ARE HANDLED.

#### Active vulnerability monitoring and alerting

Since AdNovum software as described above builds on existing intermediate products and libraries wherever this is possible and sensible, in particular OSS, it is important that any corrections of vulnerabilities in these components are incorporated into the applications delivered by us as well. This is the responsibility of the AdNovum Security Officer, the Security Engineering team and other engineering and product teams.

To identify vulnerabilities, we perform daily active monitoring of the available and known mailing lists and security alerts. When a security alert is issued for an OS component that is used (such as OpenSSL or Apache HTTP Server), or an OS version for appliances, the responsible engineers in a first step analyze whether AdNovum software is directly or indirectly affected by this vulnerability. They estimate the risk and criticality of the vulnerability based on this analysis. If AdNovum software is actually affected and the provider of the affected component has already issued a patch or new release to correct the vulnerability, we provide it immediately or for the next release. If the criticality is high and no fix is available yet, we develop our own or at least find a workaround so the vulnerability cannot be exploited. We can for example deactivate specific elements of the affected components that are not necessarily needed by the AdNovum software. The



*Whenever vulnerabilities arise, AdNovum follows an established process.*

new release is automatically tested in the NightlyBuild or in a continuous/daily build depending on the criticality. With supplementary manual tests we ensure that the new release is no longer affected by the vulnerability so that the quality expectations of our customers are met. It is especially important to ensure that the fix is not limited to the vulnerability as such but also eliminates its cause.

For security alerts of high or moderate criticality, we send a Security Bulletin to our customers. It includes our risk assessment along with recommendations for the update priorities. We also provide the release dates for the individual affected software products.

In case of highly exposed components such as OpenSSL or Apache HTTP Server that are often used in the access zone, we send out a Security Bulletin even in case of lower or no criticality.

### THE FIX MUST NOT LIMIT ITSELF TO THE VULNERABILITY, BUT ALSO HAS TO ELIMINATE ITS CAUSE.

Customers often monitor the more common OSS products themselves using the same sources. Questions on the evaluation of criticality often come up. That is why our Security Officer coordinates communication with customers to ensure consistency. The respective project managers at AdNovum are responsible for project-specific communication.

Once the new releases of the affected components have been delivered, we internally set the components containing the vulnerability to «forbidden». This ensures that these components are not unintentionally used again in any customer project or product. In exceptional cases we set the affected versions of the components for certain projects and products to «approved restricted» when it is clear that they are not affected by the vulnerability directly or indirectly. This may be the case when the defective element of the affected component is not used in these projects and products, and if it is not opportune from a customer perspective to update the component at the time. Customers with components of AdNovum's Security Suite Nevis are not only informed with the Security Bulletin in case of vulnerabilities, but also through the Nevis blog.

### Conclusion

While software that is free of defects and security vulnerabilities is something to strive for, it cannot be guaranteed in the real world. It needs to be the goal of any reputable software provider

to minimize these vulnerabilities and defects through the principles, tools and processes of secure software development. A software company also needs to know on the one hand which components are installed in what customer solutions and, on the other hand, which weaknesses currently affect these components.

When security problems arise or become known, professional conduct and clearly defined directives for handling vulnerabilities offer the best protection. That means companies need to clarify who is responsible for security problems, for example a Security Officer. This is especially important for software companies. Security monitoring and a defined, established alerting process with consistent external communication are needed as well. In order to accomplish this, the processes for preparing new releases of customer projects and products as well as the (automatic) testing of these releases have to work effectively and efficiently. Products such as the Nevis Suite on the one hand and all customer projects on the other hand benefit from these core elements of security management at AdNovum. That is why security management in software production not only means costs for customers with a software maintenance agreement, but also genuine added value. ■

---

### René Rehmann

*René Rehmann, Dr. phil. nat., has been working for AdNovum as Business Project Manager and Security Officer since 2012. He focuses on security and identity management in projects. In his free time he likes to frequent the golf courses of this world.*

### Marcel Vinzens

*Marcel Vinzens, with AdNovum since 2002, holds an MSc ETH in Computer Science and is CISSP-certified. As Technical Project Manager and Solution Architect, he was dedicated to the conceptual design, engineering and integration of security and middleware systems for several years. He has been monitoring the lifecycle of the products and frameworks used by AdNovum internally and in customer projects as Deputy CTO since 2013, advises and supports customers in security and architecture matters and has been heading the Security Engineering team since the beginning of 2015. He spends his free time with his family, enjoying culinary highlights and leisure activities.*