# Bedrohungslage von SAP Systemen durch ABAP Eigenentwicklung

Andreas Wiegenstein

Virtual Forge GmbH
andreas.wiegenstein@virtualforge.com

## Zusammenfassung

Praktisch jedes Unternehmen, das SAP einsetzt, muss die SAP Standardsoftware an die eigenen, speziellen Arbeitsweisen im Unternehmen anpassen. Solche Eigenentwicklungen werden dabei – je nach Unternehmen – rein intern, rein extern oder als Mischform durchgeführt. Doch obwohl Unternehmen teilweise schon jahrelang eigenen ABAP Code schreiben, haben die wenigsten von ihnen einen Überblick über den tatsächlichen Umfang ihrer Eigenentwicklungen. Insbesondere haben Sie keinen Überblick über die Sicherheitsrisiken, die ihnen durch den selbst geschriebenen Code entstehen. Dieses Dokument stellt die kritischsten Sicherheitsrisiken vor, die durch ABAP Eigenentwicklungen entstehen und liefert eine statistische Auswertung der Analyse der Eigenentwicklungen aus 217 SAP Systemen von internationalen Industrieunternehmen und Organisationen. Abschließend werden auch Best Practices und Sicherheitsstandards für sichere ABAP Entwicklung vorgestellt.

# 1 Grundlagen und Motivation

Der folgende Abschnitt liefert eine Übersicht darüber, welchen Fokus die meistens Unternehmen in der Vergangenheit in Punkto SAP Sicherheit hatten und warum es wichtig ist, einen ganzheitlichen Ansatz zu verfolgen.

## 1.1 Risikobewusstsein SAP

SAP ist heute der de-facto Standard im Bereich Enterprise Ressource Planning. Laut SAP setzen weltweit mehr als 248.000 Unternehmen Software von SAP ein. Entsprechend bilden SAP Systeme das digitale Rückgrat der Industrie und bereits einzelne Systemausfälle verursachen rasch Kosten in Millionenhöhe. Unternehmen sind daher gut beraten, Ihre Systeme stabil und sicher zu betreiben.

Der Fokus von SAP Sicherheit liegt bei den meisten Unternehmen auf der korrekten Ausprägung der Rollen und Berechtigungen für die Anwender, insbesondere im Bereich "Segregation of Duties". Dieser Ansatz ist primär von Compliance Anforderungen getrieben und führt teilweise zu enormen Investitionen seitens der Unternehmen. Doch während dieser Fokus den Anforderungen von Audits und Wirtschaftsprüfungen genügen mag, so schafft er keine hinreichend Basis für einen sicheren Betrieb.

Der sichere Betrieb von SAP Anwendungen ist nämlich nur möglich, wenn Sicherheit ganzheitlich betrachtet wird. Berechtigungen sind dabei ein wichtiger Aspekt, der aber nur dann greifen kann, wenn die Integrität der Funktionalität der SAP Software im Ganzen gewährleistet ist.

P. Schartner · K. Lemke-Rust · M. Ullmann (Hrsg.) · D•A•CH Security 2015 · syssec (2015) 378-386.

Die ganzheitliche Sicherheit von SAP Systemen hängt von verschiedenen Faktoren ab. Diese sind z.B. Architektur, Rollen & Berechtigungen, Systemhärtung, Verschlüsselung, Konfiguration, Patch-Management und Applikationssicherheit.

Aber Applikationssicherheit wird hierbei oft vergessen bzw. nicht als Risiko gesehen. Insbesondere für Eigenentwicklungen in ABAP – mit denen Unternehmen den SAP Standard anpassen – gibt es häufig keine Richtlinien, welche Sicherheitsaspekte zu beachten sind. Fundamentale Fragen wie "Was helfen mir meine Rollen und Berechtigungen, wenn meine Eigenentwicklungen gar keine Berechtigungsprüfungen enthalten?" werden leider viel zu selten gestellt. Und andere, viel gefährlichere Sicherheitsrisiken wie z.B. *Command Injections* sind den meisten ABAP Entwicklern gänzlich unbekannt. In Konsequenz finden sich in den (teilweise jahrzehnte-alten) ABAP Eigenentwicklungen massenweise Sicherheitsfehler.

## 1.2 Datenbasis der Qualität von Eigenentwicklungen

Die Firma Virtual Forge GmbH führt kontinuierlich Qualitätsanalysen von Eigenentwicklungen durch: den "Business Code Quality Benchmark". Die Basis des vorliegenden Dokuments ist ein Auszug aus den Ergebnissen dieser statistischen Datenerhebung [ViFo15][1], der Codequalität und Sicherheitsfehler in SAP Systemen von 217 Unternehmen umfasst.

Die Informationen in diesem Dokument beziehen sich auf Analysen aus den Jahren 2013 und 2014, die über 450 Millionen Zeilen ABAP Code aus Kundenentwicklungen abdecken. Diese statistische Datenerhebung ist bislang einzigartig im SAP Umfeld und liefert damit der Industrie zum ersten Mal einen (belastbaren) Überblick über die zu erwartende Risikosituation ihrer eigenen ABAP Anwendungen. Es ist allerdings wichtig, die Zahlen zu kommentieren, um die damit verbundenen Risiken entsprechend deutlich zu machen.

# 2 Generelle ABAP-Metriken

Bevor ich auf konkrete Arten von Sicherheitsdefekten eingehe, stelle ich zunächst ein paar allgemeinere Statistiken vor.

## 2.1 Umfang und Relevanz von Eigenentwicklungen

Die meisten Unternehmen messen den Umfang ihrer Entwicklungen an der Anzahl der Pakete, die sie für ihre Entwicklungsobjekte erstellt haben. Dies ist natürlich keine belastbare Kennzahl, da Pakete nichts über den Umfang des eigentlichen Codes aussagen und je nach Programmierstil auch die Anzahl der Entwicklungsobjekte in Paketen stark variiert.

Üblicherweise misst man den Umfang von Programmen an der Menge des entwickelten Quellcodes, genauer gesagt an der Anzahl der Quellcodezeilen ("Lines of Code", kurz "LoC"), Kommentare und Leerzeilen nicht mitgerechnet.

Nach diesem Verfahren ergibt sich *pro SAP System* bei Kunden ein Volumen von etwa 2,1 Millionen Zeilen Quellcodes. Ein Unternehmen mit 25 produktiven SAP Systemen hat damit statistisch gesehen etwa so viel Code selbst entwickelt (oder entwickeln lassen), wie in Microsoft's Windows Server 2003 steckt.

[1] Quelle: https://www.virtualforge.com/de/labs/benchmark.html

Insbesondere lässt sich aus dieser Zahl ableiten, dass (praktisch) alle Kunden den SAP Standard im großen Umfang anpassen bzw. erweitern. In der Analyse gab es kein Unternehmen, das SAP Systeme (nahezu) in der reinen Standardauslieferung betreibt. Das bedeutet insbesondere, dass die Risiken durch unsicheren ABAP Code statistisch gesehen alle Unternehmen betreffen, die SAP (Netweaver) im Einsatz haben.

## 2.2 Angriffsoberfläche

Viele Sicherheitsrisiken entstehen, weil Daten, die ins System gelangen ("Input") nicht hinreichend auf Korrektheit geprüft werden, bevor sie an potentiell gefährliche Unterfunktionen oder Befehle weitergegeben werden. Anders ausgedrückt: je mehr Input von einem Programm verarbeitet wird, desto höher ist das Risiko, dass dadurch eine Schwachstelle entsteht.

Durch Eigenentwicklungen erhöht sich die Zahl der verarbeiteten Inputs um 14.508 pro System. Das sind etwa 3% der vorhandenen Inputs in einem SAP ECC 6 Standard System. Dabei können Entwickler, selbst wenn sie den Input überhaupt prüfen, nicht unbedingt entscheiden, welches der (aus Sicherheitsaspekten) erlaubte Wertebereich ist. Denn häufig wird Input nicht im Kundencode verarbeitet, sondern an einen (komplexen) SAP Standardbaustein weitergegeben, dessen Funktionsweise von Entwicklern nicht im Detail in Punkto Sicherheit analysiert werden kann.

Durch die große Menge an zusätzlichem Input erhöhen Eigenentwicklungen die Angriffsoberfläche eines SAP Systems signifikant.

## 2.3 Datenquellen

Für eine Sicherheitsbetrachtung ist es wichtig zu verstehen, woher die Daten stammen, die im System verarbeitet werden. Insbesondere ist dabei wichtig, welchen Grad des Systemzugangs ein Angreifer benötigt. Muss er beispielsweise an einem User Interface angemeldet sein, oder genügt eine einfache HTTP Verbindung die jeder von irgendwo auf der Welt aufrufen kann?

Je weniger Rechte ein Angreifer benötigt und je weiter er vom LAN des Unternehmens entfernt arbeiten kann, desto höher ist das Risiko eines Angriffs von außen.

Die Statistik zeigt, dass nur 0,3% aller Programme im Kundencode via HTTP gerufen werden können. Insbesondere stammen 78% aller verarbeiteten Daten (Input) von SAP GUI* Anwendungen, wohingegen immer noch 18% per RFC* gesendet werden.

Daraus lassen sich mehrere Schlüsse ziehen.

- Sicherheitslösungen, die gegen Angriffe aus dem Internet schützen (wie z.B. Web Application Firewalls) liefern so gut wie keinen Schutz vor Sicherheitsrisiken, die durch Eigenentwicklungen im ABAP entstehen.
- Die Sicherheitsrisiken durch Eigenentwicklungen bieten insbesondere Innentätern ein erhöhtes Angriffspotential auf Unternehmensdaten.

* SAP GUI ist ein proprietärer Client und RFC ein Protokoll für (automatisierten) Datenaustausch zwischen SAP Systemen.

## 2.4 Datenbankzugriffe

Auch die Anzahl der Datenbankzugriffe ist (für Sicherheitsbetrachtungen) statistisch interessant. Werden beispielsweise nur sehr wenige Datenbankzugriffe ausgeführt, ist das Potential

für Datendiebstahl und Manipulation viel geringer, als bei Programmen die massiv auf die Datenbank zugreifen.

In Eigenentwicklungen lassen sich im Schnitt 24.635 Open SQL Zugriffe pro System verzeichnen. Anders ausgedrückt findet pro 85 Zeilen Quellcode ein Datenbankzugriff statt. Von diesen Open SQL Zugriffen sind etwa 86% lesend und 14% schreibend. Das bedeutet, dass Eigenentwicklungen im großen Umfang auf Unternehmensdaten zugreifen. Entsprechend führen also Sicherheitsfehler in den Eigenentwicklungen mit hoher Wahrscheinlichkeit auch zu unerlaubtem Datenzugriff.

Interessant ist auch die Beobachtung, dass in den Eigenentwicklungen im Schnitt 26.147 Programmierfehler beim Umgang mit Open SQL Zugriffen gemacht werden. Diese Fehler beziehen sich auf Qualitätsaspekte aus den Bereichen Sicherheit, Compliance, Performance, Stabilität und Wartbarkeit. Anders formuliert: in Eigenentwicklungen gibt es mehr Fehler im Zusammenhang mit Datenbankzugriffen, als es Datenbankzugriffe gibt. Das ist eine ziemlich ernüchternde Feststellung in Punkto Code Qualität.

# 3 Sicherheitsrisiken in ABAP

Dieser Abschnitt befasst sich mit Testergebnissen, die sich auf konkrete Sicherheitsrisiken im Code beziehen. Diese Risiken setzen sich aus Sicherheitsfehlern und Complianceverletzungen zusammen.

Als Sicherheitsfehler werden solche Programmierpraktiken eingestuft, die es einem Angreifer erlauben, die Funktionsweise eines Programms in einer vom Unternehmen unerwünschten und vom Entwickler nicht erwarteten Weise schadhaft zu manipulieren. Darunter fallen beispielsweise sogenannte *Command Injection* Schwachstellen.

Als Complianceverletzung werden solche Programmierpraktiken eingestuft, bei denen der Entwickler absichtlich Sicherheitsmechanismen (des SAP Standards) umgeht. Darunter fallen beispielsweise proprietäre Berechtigungsprüfungen. Complianceverletzungen sind insbesondere bei Audits und Wirtschaftsprüfungen problematisch, da solche Programmierpraktiken sich negativ auf die IT General Controls (ITGC) auswirken. Praktisch jeder Compliance-Standard basiert auf den ITGC, weshalb hier ein zwingender Handlungsbedarf seitens der Unternehmen besteht.

## 3.1 Die Anzahl der Schwachstellen in ABAP

Die Anzahl der durch Eigenentwicklungen verursachten kritischen Sicherheitsrisiken ist eine wichtige Kennzahl für Unternehmen, um eine Ressourcen- und Zeitplanung für die Behebung der Risiken erstellen zu können.

Die Analyse zeigt: Es gibt 1 kritisches Sicherheitsrisiko pro 1.000 Zeilen ABAP Code.

Diese Zahl ist alarmierend. Denn sie bedeutet, dass Unternehmen *pro System* im Schnitt 2.151 als kritisch zu betrachtende Sicherheitsrisiken haben. Das bedeutet zunächst, dass Unternehmen ein deutlich erhöhtes Risiko hinsichtlich der Vertraulichkeit und Integrität Ihrer Unternehmensdaten haben. Es bedeutet auch, dass der ABAP Code bei einer Wirtschaftsprüfung mit sehr hoher Wahrscheinlichkeit eine rote Ampel erzeugen wird. Es bedeutet aber vor allem, dass Unternehmen nach dieser Statistik mit einem erheblichen Aufwand rechnen müssen, wenn sie die vorhandenen Sicherheitsrisiken manuell beheben wollen.

## 3.2 Absolut toxische Schwachstellen

In SAP Systemen geht das höchste Risiko von 3 Arten von Schwachstellen aus.

- ABAP Command Injections
- OS Command Injections
- Native SQL Injections

Ein einziges Vorkommen einer dieser Schwachstellen bedeutet, dass ein Angreifer die totale Kontrolle über das betroffene SAP System erlangen kann.

Totale Kontrolle bedeutet hierbei unter anderem, dass sämtliche Geschäftsdaten (z.B. Personal-, Finanz-, Kunden- und Produktionsdaten) ausgelesen und/oder geändert werden können. Totale Kontrolle bedeutet ferner, dass der Angreifer seine Rechte im System erweitern und sich neue Benutzer anlegen kann. Totale Kontrolle bedeutet außerdem, dass der Angreifer die Sitzungen (aller) anderen Benutzer beenden und deren Zugangskonten sperren kann. Totale Kontrolle beinhaltet auch die Möglichkeit den SAP Rechner abzuschalten, dessen Festplatte zu formatieren oder beliebigen Schadcode zu installieren. Totale Kontrolle bedeutet nicht zuletzt auch, dass der SAP Standardcode nachhaltig schadhaft verändert werden kann und damit beliebige Hintertüren ins System eingebaut werden können.

Sicherheitsrisiken dieser Art müssen unbedingt sofort behoben werden.

Die statistische Analyse zeigt, dass durch Eigenentwicklungen *pro System* **10** dieser hochkritischen Sicherheitsdefekte vorhanden sind.

## 3.3 Die häufigsten kritischen Schwachstellen in ABAP

Es gibt jede Menge Programmierpraktiken, die Sicherheitsrisiken in sich bergen. Meistens sind die Auswirkungen jedoch begrenzt auf die Anwendung, in der der Fehler auftritt bzw. auf bestimmte Daten oder Datenbanktabellen, die von dem Programm verarbeitet werden.

Bei den für diese Analyse verwendeten Schwachstellen handelt es sich nur um die Teilmenge aller erkannten Probleme, die wirklich kritische Auswirkungen haben. Die tatsächliche Anzahl von Sicherheitsrisiken ist deutlich höher.

Die Top 5 Arten von Fehlern finden sich in Tabelle 1. Dort ist insbesondere auch ausgewiesen auf wie vielen der untersuchten Systeme mindestens ein solcher Fehler auftrat und wie viele dieser Fehler im Schnitt pro System zu erwarten sind.

Im Voraus eine kurze Erklärung, welche Effekte die Schwachstellen haben.

**Berechtigungsfehler** decken eine ganze Gruppe von Fehlern ab, die mit fehlenden oder falsch programmierten Berechtigungsprüfungen zu tun haben. In jedem Fall führen diese Fehler dazu, dass Geschäftslogik ohne (hinreichende) Berechtigungen ausgeführt werden kann. Weitere Details zu Berechtigungsfehlern werden auch im nachfolgenden Abschnitt geliefert.

**Directory Traversal** Schwachstellen sind ein gängiger Fehler beim Zugriff auf Dateien, die auf dem SAP Server angelegt sind. Hierbei kann ein Benutzer durch Verwendung spezifischer Steuerzeichen unerlaubt auf Dateien zugreifen, die von dem Programm eigentlich gar nicht verarbeitet werden sollen. Je nach Dateizugriffsmodus kann der Benutzer diese Dateien auslesen oder ändern. Dies ist kritisch, da es sich um wichtige Dateien handeln kann, wie z.B. das Security Audit Log, Konfigurationsdateien des Betriebssystems, SAP Servers oder der Datenbank, Batch-Input-Mappen uvm.

**Direkte Modifikation von Standardtabellen** bedeutet, dass ein nicht von SAP entwickeltes Programm direkt Daten in SAP Standardtabellen modifiziert. Das ist deshalb gefährlich, weil die SAP Standardprozesse häufig Änderungsbelege anlegen, wenn wichtige Daten modifiziert werden und auch spezielle Berechtigungen abgefragt werden können. Direkte Datenmodifikation aus dem Kundencode birgt daher das Risiko der fehlenden Nachvollziehbarkeit und eines unberechtigten Zugriffs.

**Mandantenübergreifender Datenzugriff** ist eine Programmiertechnik mit der Daten aus einem anderen SAP Mandanten ausgelesen oder geändert werden können, ohne dass der Benutzer Zugriff auf diesem Mandanten hat. Es handelt sich dabei um einen vom Entwickler willentlich erstellen Bypass der von SAP vorgegebenen Mandantentrennung.

**Open SQL Injection** Probleme entstehen, wenn Entwickler Input ungeprüft in dynamische Open SQL Abfragen verarbeiten. Das primäre Risiko ist die schadhafte Veränderung der Datenbankabfrage, so dass Daten gelesen oder geändert werden könnten, auf die der Anwender überhaupt nicht zugreifen darf.

**Tab. 1:** Die 5 häufigsten kritischen Sicherheitsfehler in ABAP.

| Art der Schwachstelle | Wahrscheinlichkeit * | Anzahl ** |
|---|---|---|
| Berechtigungsfehler | 100% | 1.056 |
| Directory Traversal | 91% | 308 |
| Direkte Modifikation von Standardtabellen | 86% | 39 |
| Mandantenübergreifender Datenzugriff | 83% | 122 |
| Open SQL Injection | 70% | 15 |

* Wahrscheinlichkeit gibt in wie vielen untersuchten Systemen mindestens ein Fehler des jeweiligen Typs auftrat

** Anzahl gibt die absolute Zahl der Fehler pro System an, statistisch gemittelt.

## 3.4 Berechtigungsfehler

Die meisten SAP Kunden haben schon lange einen Fokus auf die Pflege von Rollen und Berechtigungen. In diesem Zusammenhang ist es verblüffend wie viele Programmierfehler in Zusammenhang mit Berechtigungen im ABAP zu finden sind.

Insbesondere gibt es pro System mehr Fehler in Zusammenhang mit Berechtigungsprüfungen, als korrekt ausprogrammierte Berechtigungsprüfungen. Die häufigsten Fehler sind:

- Fehlende Berechtigungsprüfung(en) in Reports
- Fehlende Berechtigungsprüfung(en) in RFC Bausteinen
- Fehlende Berechtigungsprüfung(en) bei Transaktionsstarts
- Fehlende Berechtigungsprüfung(en) in PAI Ereignissen (Dynpro Programmierung)
- Formal falsch programmierte Berechtigungsprüfungen
- Berechtigungsprüfungen ohne Ergebnisprüfung
- Proprietäre Berechtigungsprüfungen

Aus dieser Liste stellen insbesondere die proprietären Berechtigungsprüfungen einen Complianceverstoß dar. Denn hier wurde vorsätzlich eine Zugriffslogik programmiert, die am Standard vorbei arbeitet.

Durch proprietäre Berechtigungsprüfungen entstehen folgende Risiken:

- Bestehende Rollen und Berechtigungskonzepte werden umgangen

- Die Prüfungen werden nicht protokolliert
- Es entstehen Benutzer mit intransparenten Privilegien
- Die Funktionalität dieser Programme ist auf QA Systemen nicht testbar, da sie nicht ausgeführt werden kann.

Ein typisches Beispiel von proprietären Berechtigungsprüfungen sind Zugriffsabfragen, die auf dem Benutzernamen (sy-uname) basieren. Solche Praktiken finden sich auf 91% der untersuchten Systeme. Im Schnitt hat ein System 185 solcher proprietären Prüfungen im eigen entwickelten Code.

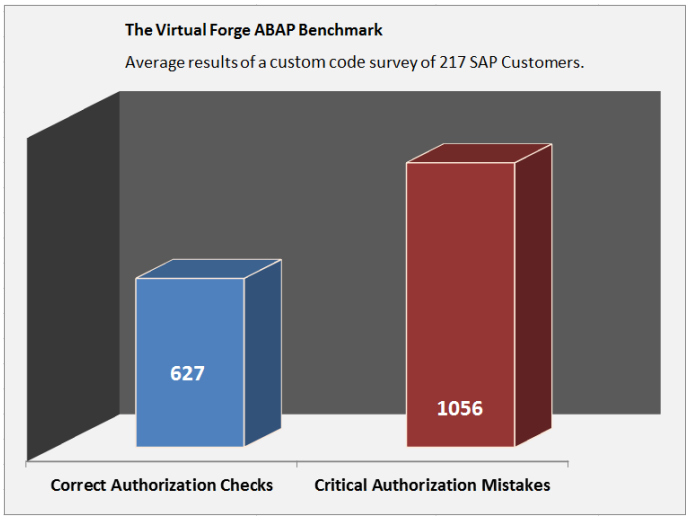


**Abb. 1:** Vergleich der korrekten Berechtigungsprüfungen und der kritischen Berechtigungsfehler

# 4 Empfehlungen

Nach den bisherigen Ausführungen ist klar, dass alle Unternehmen, die SAP einsetzen mit hoher Wahrscheinlichkeit signifikante Sicherheitsprobleme in ihren Eigenentwicklungen haben. Es ist daher sinnvoll Handlungsempfehlungen abzugeben, wie Unternehmen mit dieser Erkenntnis am besten umgehen.

Aus meinen Erfahrungen mit vielen großen ABAP Sicherheitsprojekten möchte ich die folgenden Empfehlungen weitergeben.

1. Machen Sie einen Awareness Workshop für Ihr Management
   Ohne einen verpflichtenden Prozess mit Rückendeckung des Managements können Sie keine substantiellen Änderungen an der Qualität Ihres ABAP Codes erwirken.
2. Setzen Sie sich erreichbare Ziele
   Rollen Sie neue Standards und Programmierrichtlinien in mehreren Phasen aus. Kein Entwickler lernt 300 optimierte Programmierpraktiken über Nacht.
3. Entwerfen Sie einen Plan, wie Sie mit Altlasten umgehen
   Die notwendigen Korrekturen brauchen Zeit. Darüber hinaus ist es nicht wirtschaftlich alle Fehler zu beheben. Priorisieren Sie die Sicherheitsrisiken anhand der möglichen Auswirkungen auf Ihr Unternehmen.
4. Verwenden Sie Entwicklungsrichtlinien mit dem richtigen Fokus
   Wenn Sie Verbesserungsmaßnahmen am Code durchführen, bringt das umfangreiche

funktionale Nachtests mit sich. Überlegen Sie daher vorher, ob Sie gleichzeitig auch Verbesserungen in anderen wichtigen Qualitätsbereichen (z.B. Performance) implementieren können. In jedem Fall sollten die Entwicklungsrichtlinien für interne und externe Entwicklungen verpflichtend sein. Idealerweise sind die Entwicklungsrichtlinien Teil des Pflichtenheftes bei künftigen Ausschreibungen.

5. Für die sichere Entwicklung sind Schulungen erforderlich
   Applikationssicherheit ist für viele Entwickler schwer zu verstehen. Für ein gutes Sicherheitsverständnis sind *regelmäßige* Schulungen nötig.
6. Verwenden Sie Tools um die Einhaltung Ihrer Entwicklungsrichtlinien zu überprüfen
   Nur überprüfbare Richtlinien lassen sich umsetzen. Nur Tools liefern Transparenz, wo Ihr Unternehmen in Punkto SAP Code-Qualität steht. Eine regelmäßige manuelle Sicherheitsanalyse Ihrer Eigenentwicklung ist zu teuer und dauert zu lange.
7. Messen Sie den Erfolg Ihrer Maßnahmen
   Erfolg sichert die langfristige Akzeptanz des neuen Prozesses. Dokumentieren Sie deshalb Ihre Fortschritte und geben Sie diese regelmäßig an Ihr Management weiter.

# 5 Standards

Abschließend stelle ich Ihnen noch 2 Standards vor, die Empfehlungen geben, welche Sicherheitsrisiken in ABAP mit hoher Priorität adressiert werden sollten.

Beiden Standards liegt die statistische Analyse von Virtual Forge zu Grunde. Sie unterscheiden sich primär in Umfang und Detailtiefe.

## 5.1 BIZEC APP/11

BIZEC ist ein Konsortium führender unabhängiger Sicherheitsfirmen, die auf SAP spezialisiert sind. Das Ziel von BIZEC ist es Unternehmen Empfehlungen zu geben, welches die kritischsten SAP Sicherheitsprobleme sind, um einen sinnvollen Fokus in Sicherheitsprojekten setzen zu können.

Der BIZEC APP/11 Standard stellt dabei die 11 kritischsten und häufigsten Sicherheitsprobleme in ABAP Anwendungen zusammen. Es werden jedoch aktuell keine Anleitungen gegeben, welche konkreten Programmiertechniken die Probleme lösen können.

BIZEC APP/11 dient damit lediglich als erste Orientierung für die Priorisierung von Risiken.

Weitere Informationen sind unter http://bizec.org in englischer Sprache verfügbar.

## 5.2 BSI Top 20 Risiken in ABAP Anwendungen

Viele Unternehmen setzen auf Empfehlungen und Richtlinien offiziell anerkannter Sicherheitsinstitutionen. In diesem Fall sind die "Top 20 Risiken in ABAP Anwendungen" die vom Bundesamt für Sicherheit in der Informationstechnik veröffentlicht wurden die erste Wahl. Es handelt sich dabei aber aktuell um ein sogenanntes Hilfsmittel, welches nicht Teil des Grundschutzes ist.

Diese Top 20 Risiken beinhalten den BIZEC APP/11 Standard und liefern detaillierte Beschreibungen der Probleme und empfohlener Gegenmaßnahmen.

Weitergehende Informationen sind unter [BSI14 ] oder dem QR Code in Abbildung 2 zu finden.

**Abb. 2:** QR Code für den Link zum BSI Dokument

# 6 Der Business Code Quality Benchmark

Die Inhalte dieses Dokuments basieren auf einer andauernden Qualitätsstudie von Eigenentwicklungen, dem "Business Code Quality Benchmark". Hierbei können Unternehmen kostenlos den gesamten Quellcode ihrer Eigenentwicklungen untersuchen lassen, wenn Sie die Kennzahlen des Ergebnisses anonymisiert für die Studie zur Verfügung stellen.

Wenn Sie SAP einsetzen und die Informationen in diesem Dokument als wichtig / hilfreich einstufen, dann bitte ich Sie diese Studie zu unterstützen und eines Ihrer Systeme vermessen zu lassen. Weitere Informationen sind unter [ViFo15] zu finden.

## Literatur

[BSI14] BSI: Top 20 Sicherheitsrisiken in ABAP Anwendungen – Detail-Level Ausarbeitung, Version 0.5, Oktober 2014.
https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/Hilfsmittel/ Extern/TOP-20_Sicherheitsrisiken-in-ABAP-Anwendungen.html

[ViFo15] VirtualForge: Fakten zum Business Code Quality Benchmark, 2015.
https://www.virtualforge.com/de/labs/benchmark.html