Checkmarx

AppSec-Awareness- und Entwickler-Trainings: Ein moderner Ansatz

Der Checkmarx Guide zum Secure-Coding-Training

Enter

INHALT

EINFÜHRUNG

Security wird mehr und mehr zum integrierten und automatisierten Bestandteil der Software-Entwicklung, und die Application-Security-Experten der Unternehmen arbeiten heute oft eng mit abteilungsübergreifenden Entwicklungsteams zusammen, um die von den Unternehmen entwickelte Software sicherer zu machen. Unklar ist allerdings noch, welche Rolle den Entwicklern bei der Umsetzung der Software-Security-Programme zukommt – und in welchem Maße die Devs heute Verantwortung für die Security übernehmen müssen.

Wer eine Antwort auf diese Fragen sucht und die Einhaltung von Secure-Coding-Practices sicherstellen möchte, muss sich zunächst eingestehen, dass es in diesem Bereich Defizite gibt – und dann nach modernen Ansätzen Ausschau halten, um seine Entwickler rund um AppSec und Awareness zu schulen. Bevor wir auf den bewährten Ansatz von Checkmarx zu sprechen kommen, schauen wir uns aber zuerst an, wo viele Unternehmen heute stehen.

In einer ESG-Studie gaben **79 %** der Befragten an, dass ihr Unternehmen regelmäßig oder gelegentlich Code mit bekannten organischen Schwachstellen in die Produktivumgebung überführt. Devs, AppSec- und Ops-Teams stehen offenbar unter hohem Druck, wenn sie bekanntermaßen fehlerhaften Code ausrollen. **54 %** begründen dies mit kritischen Deadlines.

Mehr lesen

In einer Studie von DarkReading gaben **78 %** der befragten IT- und Security-Manager an, Security sei wichtig genug, um deshalb das Deployment von Anwendungen zu verzögern. Aber nur **26 %** der Unternehmen finden, dass ihre Entwickler sehr gut mit Secure-Coding-Practices vertraut sind.

Mehr lesen

Eine Untersuchung von Checkmarx belegt, dass sich die Verantwortung für AppSec immer mehr von der IT auf die Entwickler verlagert. **46 %** der Devs geben an, dass sie in den letzten 12 Monaten den Fokus auf den Erwerb oder die Verbesseung von Secure-Coding-Skills gelegt haben. **36 %** geben an, AppSec-Trainings stünden ganz oben auf ihrer Wunschliste.

Mehr lesen

Gibt es da einen Zusammenhang?

Diese Daten legen eine inhärente Korrelation nahe:

- Angreifbare Software gelangt in Produktivumgebungen.
- Schwachstellen führen zu Verzögerungen und langsameren Releases.
- Entwickler wollen sich fortbilden, um diese Situation zu entschärfen.

Es gibt eindeutig eine Diskrepanz zwischen dem Wunsch von Unternehmen, sichere Software zu produzieren, und dem Level der AppSec-Awareness und der AppSec-Trainings sowie den Secure-Coding-Skills der Entwickler.

Dieser Guide soll diese Lücke schließen. Er zeigt Ihnen, wie Sie Ihren Entwicklern helfen, ihr Bewusstsein für die AppSec-Awareness zu schärfen und ihre Secure-Coding-Skills im Rahmen ihrer täglichen Arbeit nachhaltig zu verbessern. So stellen Sie die Weichen, um schneller sicherere Software bereitzustellen.

Im Folgenden erfahren Sie alles, was Sie wissen müssen, um Ihren Entwicklern die bestmögliche Plattform für AppSec- und Awareness-Trainings an die Hand zu geben: eine, die sie tatsächlich nutzen werden.

KAPITEL 1

DIE ENTWICKLER-PERSPEKTIVE VERSTEHEN

Alle Indikatoren sprechen dafür, dass Entwickler sichereren Code entwickeln wollen. Und sie sind auch durchaus offen für Security-Trainings – sei es, weil sie sich gerne mit Kollegen messen, weil sie ihre Security-Awareness schärfen oder einfach nur, weil sie besser coden wollen. Was sie aber nicht möchten, ist, dass die Security auf ihre eigentliche Aufgabe ausstrahlt: die Entwicklung und Bereitstellung von Software.

Die wertvollste Ressource des Entwicklers ist Zeit

Die schnell getaktete Entwicklung von heute setzt auf kurze Bereitstellungszyklen mit minimaler Fehlerzahl, um eine nahtlose Integration in die CI-/CD-Pipelines zu ermöglichen. Trainings im Bereich Secure Coding werden von den Entwicklern nur dann angenommen, wenn sie sie nicht verlangsamen, einen konkreten Bezug zu täglichen Aufgaben aufweisen und die Abläufe nicht stören.

Unternehmen müssen bei der Auswahl von AppSec-Awareness-Lösungen und Secure-Coding-Trainings darauf achten, dass sich diese nahtlos in die Umgebung einfügen, in der der Entwickler seinen Code schreibt, und ihm genau die Werkzeuge an die Hand geben, die er benötigt.

Entwickler sollen in erster Linie entwickeln

Den Punkt „Entwicklung von sicherem Code" werden Sie in den meisten Jobbeschreibungen, Arbeitsverträgen und Onboarding-Guides vergeblich suchen. Sicher zu entwickeln, gilt oft als „Nice-to-have, wenn es die Zeit zulässt". Und Zeit ist vor dem Hintergrund der Digitalisierung eine knappe Ressource.

Entwickler werden heute daran gemessen, wie schnell sie funktionalen Code bereitstellen – nicht an der Zahl der darin enthaltenen Schwachstellen. Sie wissen zwar, dass sie fehlerfreien Code abliefern sollen, und bemühen sich, das zu tun. In aller Regel konzentieren sie sich aber auf Fehler, die die Funktionsweise der Software beeinträchtigen. Sicherheitsrelevante Schwachstellen stehen für sie weniger im Fokus.

Wenn Entwickler sicheren Code bereitstellen sollen, müssen die Team-Leads die Security-Schwachstellen mit der gleichen Priorität behandeln wie Coding-Fehler. So steigt der Stellenwert des Secure Codings, und sie können im nächsten Schritt systematische AppSec- und Awareness-Trainings auf die Agenda nehmen.

KAPITEL 2

WARUM APPSEC-TRAININGS SO OFT SCHEITERN

Videotutorials, Vorlesungen, Powerpoint-Slides, regelmäßiger Frontalunterricht und obligatorische Online-Kurse sind fester Bestandteil vieler AppSec-Initiativen. Aber es gelingt damit nur selten, das Bewusstsein der Entwickler für die Security zu schärfen und ihnen die Skills zu vermitteln, die sie brauchen, um sicheren Code zu erstellen. Das liegt in erster Linie daran, dass all diese Ansätze nur als Checkliste wahrgenommen werden, die es abzuhaken gilt – nicht als wertvolles Tool zum Schutz der eigenen Anwendungen.

Solange die Entwickler ganz andere Prioritäten setzen, werden solche veralteten Schulungsansätze nicht den gewünschten Nutzen bringen. Gibt es vielleicht einen besseren Weg?

Warum Checklisten nicht funktionieren

Studien zur Wirksamkeit von Schulungen (etwa Arthur et al., 2003) belegen:

- Lernen gelingt am besten, wenn das Training auf eine konkrete Verhaltensweise oder Fähigkeit ausgerichtet ist, in einem für den Lernenden relevanten Kontext vermittelt wird und zeitnah umsetzbar ist.
- Trainings und Fortbildungen müssen zugänglich, relevant und sofort anwendbar sein, statt lediglich Informationen bereitzustellen, um Wissen aufzubauen.
- Simulationen können das prozedurale, erklärende und erfahrungsbasierte Verständnis der Cybersecurity erweitern und das Lernen effektiver machen (wobei bislang nur wenige Studien diese Behauptung systematisch geprüft haben).

Auch wenn es verführerisch ist, technische Themen in einer simplen Checkliste zusammenzufassen, ist dies der falsche Weg, um das Bewusstsein für die Gefahren in der Application Security zu schärfen.

Wenn sie lediglich als informative Liste präsentiert werden, bewirken Themen wie die OWASP Top 10 Risiken, die SANS Top 25 Software-Fehler, gängige Secure-Coding-Guides oder Best-Practice-Tutorials erfahrungsgemäß keine nachhaltige Verhaltensänderung auf Seiten der Zuhörer.

Der Erfolg von Schulungsmaßnahmen hängt davon ab, ob die Teilnehmer das Gelernte selbst anwenden können – etwa im Rahmen von immersiven Simulationen und Live-Aktivitäten. Für Trainigs im Bereich AppSec-Awareness empfehlen wir gamifizierte und kontextbasierte Schulungen.

KAPITEL 3

WIRKSAME APPSEC- UND AWARENESS-TRAININGS

Wirksame AppSec- und Awareness-Trainings sollten die Möglichkeiten und die Werkzeuge ausschöpfen, die uns moderne Technologie heute bietet. Innovativen Mobile Apps gelingt es ja auch immer wieder, das Verhalten der User nachhaltig zu prägen. Moderne Secure-Coding-Lösungen sollten die gleichen Gaming-Prinzipien und technologischen Features nutzen, um die Anwender langfristig zu binden.

Die Lehren der Gamifizierung

Gamifizierung ist die Anwendung von Game-Design-Elementen und Gaming-Prinzipien außerhalb des Kontexts eines Spiels. Die Vorteile von gamifizierten Trainings sind bekannt, werden bei AppSec- und Awareness-Trainings bislang aber selten genutzt. Dabei eignet sich Gamifizierung sehr gut für Simulationen, etwa von Szenarien, bei denen Angreifer Schwachstellen ausnutzen und Verteidiger die Lücken schließen müssen.

Es ist erwiesen, dass wir besser lernen und länger aufnahmefähig bleiben, wenn wir beim Lernen Spaß haben und eine aktive Rolle einnehmen. Da Entwickler die meiste Zeit ihres Tages vor einem Bildschirm verbringen und auf Codezeilen gucken, schätzen sie lockere und spielerische Trainings weitaus mehr als langweilige Lektionen, die von ihnen tiefe Konzentration erfordern – und lernen dabei erfahrungsgemäß auch mehr.

Ausgehend von dieser Prämisse lassen sich vier Schritte für eine effiziente Gamifizierung der AppSec-Trainings ableiten.

Wie Sie in **vier Schritten** Ihre AppSec-Trainings erfolgreich gamifizieren

1

Schulen Sie interaktiv

Der Chief Learning Officer® hätte es nicht besser ausdrücken können:

„Wenn ein Teilnehmer fleißig klickt, bedeutet das nicht, dass er aufmerksam folgt. Vielleicht will er einfach nur schnell ans Ende der Lektion kommen."

Kennen Sie das? Wir sehen das oft, wenn Mitarbeiter online Schulungen zu einem Thema abslovieren müssen, deren Sinnhaftigkeit ihnen nicht einleuchtet, etwa bei obligatorischen IT-Security-Schulungen. Sie haben schon genug auf dem Schreibtisch, also wollen sie die Schulung so schnell wie möglich hinter sich bringen und sich wieder ihrem Arbeitspensum widmen. Also klicken sie sich schnell durch den Online-Kurs oder das Quiz, ohne die Bedeutung des Contents zu erfassen – eine wirkungslose Pflichtübung.

Der Chief Learning Officer führt dann aus, dass Stories und Beispiele, die interaktiv vermittelt werden (z. B. bei Angriffs- und Verteidigungssimulationen) ein integraler Bestandteil moderner Lernkonzepte sind. Sie ermöglichen es Teilnehmern, Inhalte unmittelbar und emotional aufzunehmen und machen es ihnen wesentlich leichter, das vermittelte Wissen zu behalten.

Die Interaktivität sorgt dafür, dass die Entwickler dem Inhalt ihre volle Aufmerksamkeit widmen, was die Chance erhöht, dass sie etwas lernen und behalten. Dies ist besonders wichtig, wenn man bedenkt, dass viele Menschen durch Handeln und Erleben effektiver lernen als durch Hören und Sehen.

Interaktivität kommt den Bedürfnissen der Zielgruppe entgegen, zumal dann, wenn das interaktive Schulungsmaterial in kleinen Dosen sofort konsumiert werden kann – etwa, wenn man lernt, einen soeben aufgetretenen Fehler zu beheben (statt dies heute in der Theorie zu lernen, aber erst in einer Woche praktisch anwenden zu können). Diese Unmittelbarkeit schließt die Lücke zwischen Lernen und Tun und macht es ungleich leichter, dass Gelernte zu behalten.

Und wenn Sie all das noch nicht überzeugt hat: Bedenken Sie, wie schwer es ist, sich im Eiltempo durch eine interaktive Session zu klicken.

2

Erzählen Sie eine Geschichte

Charaktere, Rollenspiele und ein roter Faden helfen Ihnen, Informationen in einen Kontext einzubetten: Stories stimulieren und binden uns ein.

AppSec-Trainings in Form von Bullet-Points, Fragen & Antworten und öden Textwüsten werden Ihre Entwickler langweilen. Wirksame Schulungsprogramme leben von Stories, Charakteren und Problemen, die es zu lösen gilt.

Wenn Sie Ihren Entwicklern Charaktere präsentieren, in deren Schuhe sie schlüpfen können, und eine Storyline, der sie folgen können (in unserem Fall eine Schwachstelle, die behoben werden muss), werden Sie es ihnen leicht machen, das Gelernte zu behalten. Storytelling steht ja auch bei vielen Spielen im Fokus und ist ein wichiger Spaß-Katalysator – ein wichtiger Brückenschlag zur Gamifizierung.

3

Fassen Sie sich kurz

Man kann streiten, ob unsere Aufmerksamkeitsspanne immer kürzer wird. Wer Informationen präsentiert, sollte sich auf jeden Fall möglichst kurz fassen.

Kurze Lektionen sind meist präzise und auf den Punkt und enthalten keine irrelevanten Informationen. Damit steigt die Wahrscheinlichkeit, dass sich die Teilnehmer darauf einlassen werden. Zeit ist für Ihre Entwickler eine kostbare Ressource, also sollten Ihre Schulungen möglichst kompakt sein.

4

Spaß am Wettbewerb und an der Sieger-mentalität

Laut Dr. Ian Robertsons Studie „The Winner Effect" steigert es unsere Gehirnleistung, wenn wir uns Herausforderungen stellen und sie meistern. Wer siegt, lernt besser – denn Erfolge im Wettbewerb belohnen uns mit der Erhöhung des Dopaminlevels im Gehirn.

Kurze und interaktive Geschichten sind wichtig, um AppSec- und Awareness-Trainings zu etablieren. Aber auch dem „Gewinn" einer Training-Session, also etwa der erfolgreichen Behebung einer Schwachstelle, kommt eine Schlüsselrolle zu. Positive Lernerfahrungen fühlen sich gut an und führen dazu, dass die Entwickler gerne zu Ihrem Programm zurückkehren, um weitere Schwachstellen zu entdecken.

Die Freude an individuellen, persönlichen Siegen ist also ein wichtiger Anreiz, um immer wieder in eine gamifizierte Lernumgebung zurückzukehren. Ein anderer ist der Wettbewerb mit Kollegen: Sich mit anderen in einem größeren Rahmen zu messen, ist ein bewährtes Motivationsinstrument. Und hier kommen Turniere ins Spiel.

KAPITEL 4

DIE MACHT DER TOURNIERE

Viele Unternehmen stellen mit jährlichen Schulungen sicher, dass die Entwickler ihre Kenntnisse und Skills im Bereich der sicheren Programmierung stetig verbessern. Wenn Sie diese als Turniere abhalten, macht das Lernen plötzlich allen Spaß!

Ein Turnier ist ein Live-Wettbewerb, bei dem sich Entwickler mit ihren Kollegen im Lösen von AppSec-Aufgaben messen. Es vermittelt wichtige Skills rund um das sichere Coding und die Behebung von Schwachstellen – und das in einer energiegeladenen Umgebung, wo die Entwickler freundschaftliche Rivalitäten auf einem dynamischen Leaderboard austragen und spannende Challenges innerhalb festgelegter Zeitlimits knacken.

Warum Turniere sinnvoll sind

Sie schärfen die Awareness für die Application Security und sind dabei **eine unterhaltsame und lohnende Erfahrung.**

Sie machen Spaß – und **vermitteln den Entwicklern wichtige neue AppSec-Skills.**

Sie geben Aufschluss über die **AppSec-Awareness im Team** und zeigen Problemfelder auf.

Wie Sie von Tournieren profitieren

Beim Launch eines neuen Trainingsprogramms

Flankieren Sie die Einführung neuer Trainingsprogramme mit einem Turnier, um Ihren Entwicklern einen schnellen Einstieg in das Thema zu bieten und erste KPI-Daten zu erfassen. So erhalten Sie ein klares Bild von den Skills Ihres Teams und eine erste Baseline, um Fortschritte zu messen und Ziele zu setzen. Und: Sie können sich vom ersten Tag an auf die Problemfelder konzentrieren, in denen der Schulungsbedarf am höchsten ist.

Zum Abschluss eines Schulungsprogramms

Als dynamisches und aufregendes Event ist ein Turnier der perfekte Schlusspunkt am Ende eines Trainingsprogramms – zumal die Aussicht auf den Sieg Ihren Entwicklern einen zusätzlichen Motivationsschub geben wird, eventuell noch ausstehende Lektionen rechtzeitig abzuschließen.

Im Rahmen regelmäßiger AppSec-Awareness-Events

Turniere eignen sich hervorragend als Highlight auf der Agenda wiederkehrender AppSec-Awareness-Events – zum Beispiel, indem das Turnier den Rahmen bildet und zwischen den Runden andere Aktivitäten wie Vortragsveranstaltungen, Podiumsdiskussionen und Award-Verleihungen stattfinden.

KAPITEL 5

DIE VORTEILE VON KONTEXT-BASIERTEM LERNEN

Ein modernes, spielerisches AppSec-Training aufzusetzen, ist wichtig, damit die Entwickler es tatsächlich nutzen und damit trainieren. Vergessen Sie aber nicht, die Trainingseinheiten im richtigen Kontext bereitzustellen – sonst könnten Sie Ihre harte Arbeit zunichte machen.

Klassischer Frontalunterricht ist nicht nur deswegen ineffizient, weil es an spielerischen Elementen fehlt. Es fehlt auch an Kontext: Sobald Sie Ihre Devs aus der Entwicklungsumgebung reißen, durchbrechen Sie die täglichen Abläufe und machen es ihnen schwer, sich konkrete Probleme ins Gedächtnis zu rufen. Ihr Input sollte deshalb genau da bereitgestellt werden, wo er gebraucht wird: beim Coden.

Das Geheimnis erfolgreicher AppSec-Trainings ist es, sie so in die täglichen Abläufe der Entwickler einzubinden, dass sie stets nur einen Klick entfernt sind.

Wie spannend und spielerisch Ihr AppSec-Content auch ist, Ihre Entwickler werden ihn nicht auf einmal durcharbeiten. Können sie bei der Arbeit aber jederzeit auf die Inhalte zugreifen – und zwar direkt aus der Entwicklungsumgebung heraus – werden sie die Lektionen zuverlässig aufrufen, sobald eine Schwachstelle auftaucht.

Der zweite wichtige Aspekt beim kontext-basierten Lernen ist es, sicherzustellen, dass die Lösung jederzeit Ihre aktuellen Programmiersprachen abdeckt. Die Best Practices für das Secure Coding sind von Sprache zu Sprache unterschiedlich, und es gibt keine einheitliche Lösung. Mehr dazu in unseren Checkmarx-Guides:

The JavaScript Guide:
Web Application Secure Coding Practices

The Go Language Guide:
Web Application Secure Coding Practices

Kotlin Guide:
Mobile Application Secure Coding Practices

KAPITEL 6

REGELMÄSSIGE ASSESSMENTS

Behalten Sie Ihre AppSec-Awareness-Kennzahlen im Blick! Wenn Sie in AppSec- und Awareness-Trainings investieren, müssen Sie auch jederzeit darüber im Bilde sein, ob sich dieses Investment auszahlt – und ob das Risikopotenzial im Bereich Software-Security auch tatsächlich abnimmt. Um eine kontinuierliche Verbessung sicherzustellen, gilt es, die Fortschritte Ihrer Entwicklungsteams genau zu dokumentieren und die AppSec-Awareness regelmäßig neu zu bewerten.

Um schnell und einfach zu bewerten, wie es um die Secure-Coding-Skills Ihrer Entwickler bestellt ist, empfehlen wir kurze, 10- bis 15-minütige Assessments, die einzelnen Mitarbeitern oder Teams zugewiesen werden können. Die Assessments ermöglichen es Ihnen:

- Eine Baseline zu definieren, um später die Erfolge und Fortschritte zu dokumentieren.

- Anhand des Skill-Levels unnötige Entwickler-Trainings zu vermeiden.

- Probleme und Mitarbeiter zu identifizieren, die besondere Aufmerksamkeit benötigen.

- Die Wirksamkeit des Security-Awareness-Programms zu bewerten.

- Angemessene Ziele für erfahrene und weniger erfahrene Entwickler zu definieren.

- Die Secure-Coding-Skills von Bewerbern und neuen Team-Mitgliedern einzuschätzen.

Ziel der Assessments ist es, festzustellen, ob Entwickler weitere Trainings benötigen, Problemfelder zu identifizieren, Fortschritte zu tracken und zu dokumentieren und Wiederholungsfehler zu vermeiden.

KAPITEL 7

WIE SIE DIE RICHTIGE APPSEC-TRAINING-LÖSUNG FINDEN

Die AppSec- und Awareness-Training-Plattform von Checkmarx – CxCodebashing – wurde von Entwicklern für Entwickler **designt.**

Hier sind einige Fragen, die Sie den zur Wahl stehenden Anbietern vor Ihrer Entscheidung stellen sollten, sowie unsere Antworten.

Q Sind für unsere Entwickler neue, innovative Trainingsansätze verfügbar?

Ja.

CxCodebashing ist eine AppSec- und Awareness-Training-Lösung einer neuen Generation. Sie hilft Ihren Entwicklern, schneller zu lernen und ihr Skill-Set stetig zu erweitern, und gibt ihnen den wertvollen Input an die Hand, den sie von einem der Marktführer im AST-Markt erwarten.

Q Können wir unsere Entwickler up-to-date halten, ohne die Entwicklung zu bremsen?

Ja.

CxCodebashing gliedert sich nahtlos in die Prozesse und die Umgebung der Entwickler ein. Die Lösung lenkt sie nicht ab, und zwingt sie auch nicht, die gewohnte Oberfläche zu verlassen. Nahtlos in die täglichen Abläufe integriert, steht sie Ihren Mitarbeitern jederzeit zur Verfügung, ohne den Rhythmus zu unterbrechen. Auf diese Weise können sie schneller und fokussierter lernen.

Q Passt die Lösung zum täglichen Workflow unserer Entwickler?

Ja.

Im Zusammenspiel mit CxSAST weist CxCodebashing Ihre Entwickler automatisch auf die konkreten Schwachstellen hin, die in ihrem Alltag auftreten. Die Lösung zeigt dabei nicht nur auf, welche Probleme bei einem CxSAST-Scan erkannt wurden, sondern erklärt auch, wie es zu der Schwachstelle kam und wie sie korrigiert werden kann.

Q Werden unsere Entwickler die Lösung auch langfristig nutzen?

Ja.

Studien belegen, dass wir mit kontext-basierten Inhalten und Simulationen am besten lernen – so schnell, dass Ihre Entwickler ihre Fortschritte beim Secure Coding selbst merken werden.

Und es wird ihnen Spaß machen! Mit CxCodebashing lernen sie spielerisch – ganz anders als mit herkömmlichen Tools.

Q Ist die Lösung für Enterprise-Umgebungen ausgelegt?

Ja.

Codebashing bietet Ihnen standardmäßig anspruchsvolle, für Enterprise-Umgebungen optimierte Management-, Kommunikations- und Analyse-Tools, die es Ihnen leicht machen, die Lösung für 10 bis über 10.000 User zu skalieren.

Die Lösung setzt dabei einen klaren Fokus auf Compliance, und wird Ihre Entwickler für die Compliance-Vorgaben von GDPR, PCI DSS, NIST 800-53 und HIPAA sensibilisieren.

KAPITEL 8

KALKULIEREN SIE DEN ROI IHRES APPSEC-TRAININGS

AppSec- und Awareness-Trainings systematisch anzugehen und über eine ganzheitliche Plattform wie Checkmarx Codebashing abzubilden, wird sich für Ihr Business auch wirtschaftlich lohnen.

Die nebenstehende Tabelle zeigt am Beispiel eines mittelständischen Unternehmens mit 350 Software-Entwicklern (und branchenüblichen Produktivitäts- und Kostenparametern), wie Sie den RoI Ihrer AppSec-Training-Lösung kalkulieren können.

Geschätzte Zahl der Codezeilen pro Entwickler pro Tag	67,5
Arbeitstage pro Jahr	230
Geschätzte Zahl der Codezeilen pro Entwickler pro Jahr	15.525
Geschätzte Zahl der Schwachstellen pro 1.000 Codezeilen	1
Geschätzte Zahl der Schwachstellen pro Entwickler pro Jahr	15,53
Geschätze Wirksamkeit des Trainings	10 %
Senkung der Zahl der Schwachstellen pro Entwickler pro Jahr	1,55
Geschätze Zahl der Mannstunden zur Behebung einer Schwachstelle	22
Zeitersparnis durch die Minimierung der Zahl der Schwachstellen (in Stunden)	34,16
Zeitaufwand für das Codebashing-Training (in Stunden)	3
Zahl der Entwickler	350

Zeitersparnis pro Entwickler pro Jahr (in Stunden)	31,16
Kosten pro Mannstunde eines Entwicklers	USD 35
Jährliche Gesamtersparnis pro Entwickler	**USD 1.090,43**

Zeitersparnis pro Jahr (in Stunden)	10.904,25
Kosten Codebashing (pro Jahr)	USD 80.000

Einsparunden durch Codebashing pro Jahr	**USD 301.648,75**

KAPITEL 9

BEST PRACTICES UND GRUNDLAGEN

Wollen Sie mehr Awareness für AppSec schaffen, Ihre Entwickler im sicheren Coden schulen und einen raschen Return-on-Investment erzielen? Mit den folgenden Tipps stellen Sie die Weichen für erfolgreiche AppSec- und Awareness-Trainings.

Implementierung und Launch

Vor dem Launch

Die Kommunikation mit den Entwicklern sollte vor dem Start beginnen und kontinuierlich fortgeführt werden. Kommunizieren Sie offen: Sagen Sie ihnen nicht nur, was Sie tun, sondern erklären Sie, warum, definieren Sie Ziele, und lassen Sie sie wissen, dass dies Spaß macht, produktiv ist und sowohl für sie als auch für Ihr Unternehmen von Nutzen ist.

Rollout

Wenn Sie den Rollout schrittweise angehen, wird die Einführung des neuen Programms einfacher sein. Dabei können Sie den Launch zum Beispiel nach Standort, nach Abteilung, nach Anwendung, nach Programmiersprache oder nach vielen anderen Kriterien splitten, um kleinere, leichter zu managende Gruppen zu erhalten.

Bewertung

Definieren Sie zum Start des Programms eine Baseline, um die Verbesserung der Entwickler-Skills messbar zu machen und Stärken und Problemfelder besser zu verstehen. So können Sie im Gespräch mit Stakeholdern jederzeit den Mehrwert des Programms darlegen – von der ersten Schulung an.

Dieser Ansatz ermöglicht es Ihnen, die Fortschritte Ihrer Entwickler und Ihren RoI auf der Basis der für Ihr Unternehmen relevanten KPIs zu verfolgen. So können Sie die Code-Qualität deutlich schneller verbessern – und das wird sich auch auf Ihren Umsatz positiv auswirken.

Einige wichtige Regeln

Trainieren, trainieren, trainieren

Reservieren Sie regelmäßig ausreichend Zeit für fokussierte Trainings zu festen Zeiten.

Belohnen Sie Erfolge

Awards und Kudos sind oft eine wertvolle Motivationshilfe. Überlegen Sie sich vorab attraktive Incentives für Ihr Team. Öffentliche Anerkennung zählt oft mehr als finanzielle Anreize.

Fördern Sie den freundlichen Wettbewerb im Team

Welcher Coder misst sich nicht gerne mit Kollegen? Veranstalten Sie spielerische Wettbewerbe und stellen Sie sicher, dass jeder mitmachen kann.

Bewerten, verfolgen, dokumentieren

Bewerten Sie durchgegehend Ihr Programm und die Erfolge Ihres Teams. Teilen Sie diese Daten mit allen relevanten Stakeholdern. Dokumentieren Sie Erfolge und Problemfelder und lassen Sie Teams und Mitarbeiter wissen, wo sie stehen.

Addressieren Sie Probleme proaktiv

Wenn Sie neue Problemfelder identifizieren, sollten Sie diese proaktiv angehen, und alle Fortschritte und Erfolge, die Sie dabei erzielen, klar und in der Breite kommunizieren.

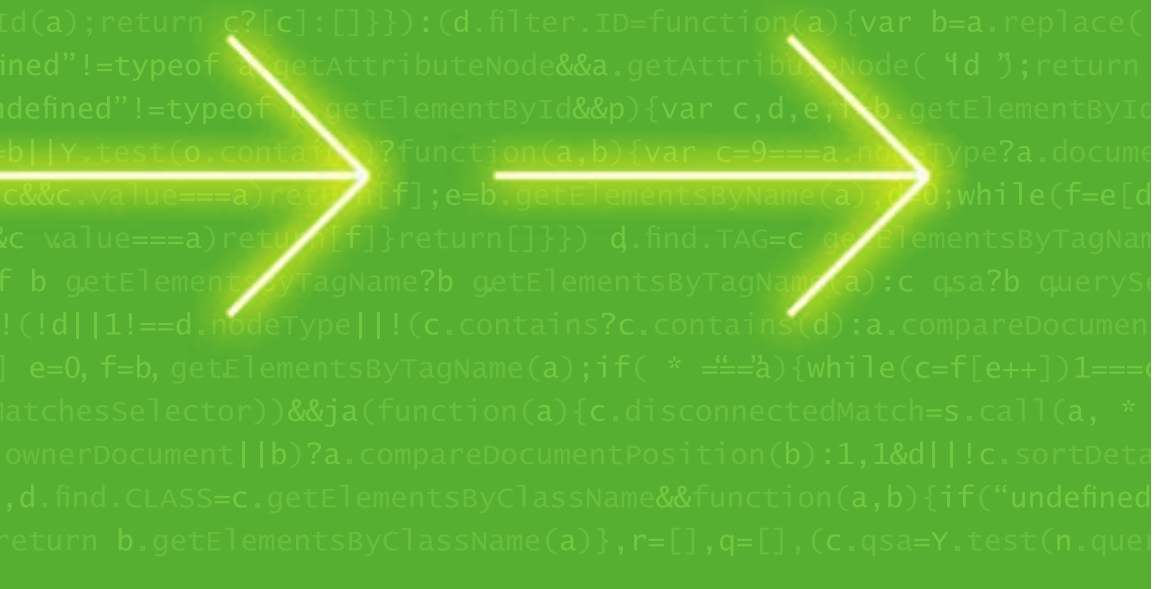
Bewährtes wiederholen

Wenn alles nach Plan funktioniert, müssen Sie daran auch nichts ändern.

KAPITEL 10

NÄCHSTE SCHRITTE

Stellen Sie Ihre AppSec- und Awareness-Trainings mit einer Plattform wie Codebashing auf ein tragfähiges Fundament – Ihr Business wird davon profitieren.



Fragen Sie Ihre Entwickler, ob sie sich unsere **kostenlosen Codebashing-Lektionen** ansehen möchten, die auf den in diesem Guide vorgestellten Prinzipien basieren.

Zu Codebashing

Wenn Ihre Entwickler ihre Hacking-Skills spielerisch testen möchten, ist unser **Game of Hacks** der perfekte Ausgangspunkt: In diesem Spiel präsentieren wir Ihren Entwicklern angreifbaren Code – und stellen sie vor die Aufgabe, die Schwachstellen darin so schnell wie möglich zu identifizieren.

Game of Hacks

Checkmarx

Über Checkmarx

Checkmarx entwickelt Software-Security für kritische Infrastrukturen, und setzt Maßstäbe, wenn es gilt, die Cyberrisiken von heute und morgen zu adressieren. Checkmarx bietet die branchenweit einzige vollumfängliche Software-Security-Plattform, die SAST, SCA, IAST und AppSec-Awareness-Trainings vereint, um Sicherheit in allen Phasen der CI/CD-Pipeline zu verankern und die Angriffsfläche zu minimieren. Über 1.400 Unternehmen weltweit setzen auf Checkmarx, um schneller sicherere Software bereitzustellen, darunter über 40 Prozent der Fortune 100 und zahlreiche große staatliche Einrichtungen.

Mehr unter Checkmarx.com