Certified Tester
Advanced Level Syllabus

Sicherheitstester
(Security Tester)

Fassung 2016
Deutsche Übersetzung 2018, V1.0

International Software Testing Qualifications Board

Deutschsprachige Ausgabe
Herausgegeben durch German Testing Board e.V.

Änderungsübersicht

Version	Datum	Anmerkungen
0.1	24. April 2015	Erstellung der Basisversion auf der Grundlage des bestehenden Expert Security Tester Draft Syllabus in Version 3.9
0.2	15. Juni 2015	Einarbeitung der Anmerkungen der Autoren nach Treffen der Autoren in Oslo
1.0 – Beta	20. September 2015	Beta-Fassung – Kommentare der Alpha-Fassung eingepflegt
1.0 – Freigabeversion für GA	4. März 2016	Prüfung durch Arbeitsgruppe nach Durchsicht, LO 4.1.2 aus K2 und K3 geändert und entsprechend neu formuliert. Text unterstützt eine K3 LO bereits angemessen.
1.0 – GA	18. März 2016	GA-Fassung – Kommentare der Beta-Fassung eingepflegt
D0.1	19. Juni 2017	Erstellung einer deutschen Basisversion auf der Grundlage des Expert Security Tester Syllabus von 2016
D0.2	5. Oktober 2017	Überprüfung der inhaltlichen Deckungsgleichheit zwischen Basisversion und Originalversion. QS
D0.3	15. November 2017	Lokalisierung auf Deutschland (BSI, IT-SIG, ISO27000, u.ä.)
D0.9	3.Januar 2018	Anpassung an deutsches Glossar, Erweiterung des Glossars um spezifische Sicherheits-Begriffe
D0.9.1	18. Januar 2018	Vorbereitung für Layout-professionalisierung in Word
D0.9.2	19. März 2018	Formatvorlagenverwendung, Beautification
D0.9.3	2. Mai 2018	Finalisierung für Beta-Review
1.0	1.-3.Okt.2018	Einarbeitung der abgestimmten Beta-Review-Findings

Inhaltsverzeichnis

0 Einführung in den Lehrplan

0.1 Zweck dieses Dokuments

Dieser Lehrplan bildet die Grundlage für das Softwaretest-Qualifizierungsprogramm der Aufbaustufe (Advanced Level) für das Spezialmodul „Sicherheitstester“. Das ISTQB® stellt den Lehrplan folgenden Adressaten zur Verfügung:

1. Nationalen Boards zur Übersetzung in die jeweilige Landessprache und zur Akkreditierung von Ausbildungsanbietern. Die nationalen Boards können den Lehrplan an die eigenen sprachlichen Anforderungen anpassen sowie die Querverweise ändern und an die bei ihnen vorliegenden Veröffentlichungen angleichen.
2. Prüfungsinstitutionen zur Erarbeitung von Prüfungsfragen in der jeweiligen Landessprache, die sich an den Lernzielen der jeweiligen Lehrpläne orientieren.
3. Ausbildungsanbietern zur Erstellung ihrer Kursunterlagen und zur Bestimmung einer geeigneten Unterrichtsmethodik.
4. Prüfungskandidaten zur Vorbereitung auf die Prüfung.
5. Allen Personen, die im Bereich Software- und Systementwicklung tätig sind und ihre fachliche Kompetenz beim Testen von Software verbessern möchten, sowie als Grundlage für Bücher und Fachartikel.

Das ISTQB® kann die Nutzung dieses Lehrplans auch anderen Personenkreisen oder Institutionen für andere Zwecke genehmigen, sofern diese vorab eine entsprechende schriftliche Genehmigung einholen.

0.2 Überblick

Zielgruppe des Aufbaukurses zur Ausbildung zum Sicherheitstester (Advanced Level Sicherheitstester) sind Personen, die bereits berufliche Erfahrungen beim Testen von Software vorweisen können und ihr Wissen im Bereich der Sicherheitstests vertiefen wollen. Die im Advanced Level angebotenen Module decken ein breites Spektrum von testbezogenen Themen ab.

Für den Erhalt der Advanced Level-Zertifizierung im Modul „Sicherheitststester“ benötigen Kandidaten das „Certified Tester Foundation Level“-Zertifikat. Zudem müssen sie vor der Prüfstelle (Exam Board) nachweisen, dass sie über ausreichende praktische Erfahrungen verfügen, um auf Advanced Level zertifiziert zu werden. Das heißt, sie müssen mindestens drei Jahre einschlägiger Erfahrung im akademischen, praktischen oder beratenden Bereich vorweisen können. Die konkreten Kriterien für die praktische Erfahrung erfragen Sie bitte bei der zuständigen Prüfungsstelle.

0.3 Prüfung

Sämtliche Prüfungen, die auf Advanced Level für dieses Modul durchgeführt werden, basieren auf dem vorliegenden „Advanced Level Sicherheitstester“ Lehrplan.

Das Prüfungsformat ist in den „Advanced Exam Guidelines“ des ISTQB festgelegt.

Die Prüfungen können im Rahmen eines akkreditierten Ausbildungskurses oder unabhängig (z. B. bei einer Prüfstelle) abgelegt werden. Die Prüfungen können auf Papier oder elektronisch abgelegt werden. Dies muss jedoch in jedem Fall beaufsichtigt werden (durch einen Bevollmächtigten einer nationalen Zertifizierungsstelle).

0.4 Aufbau des Lehrplans

Der Lehrplan gliedert sich in zehn Kapitel. In der Kapitelüberschrift ist jeweils die Zeit für die Vermittlung des Kapitels angegeben. Zum Beispiel:

1. Die Grundlagen des Testens der Sicherheit 105 min

gibt an, dass für Kapitel 1 eine Zeit von 105 Minuten für die Vermittlung des Lernstoffs veranschlagt wird. Am Anfang der einzelnen Kapitel sind spezielle Lernziele angegeben.

Der Lehrplan enthält naturgemäß viele Aufzählungen. Im Gegensatz um englischen Original sind diese in der deutschen Übersetzung nicht als „dotted list“, sondern als nummerierte Liste aufgeführt. Dies vereinfacht das Referenzieren auf einzelne Punkte durch die Nummer. Der Charakter der Liste als unpriorisierte Aufzählung bleibt daher allerdings unberührt, d.h. die Nummern dienen nur für die Referenzierung und nicht einer ordinalen Reihenfolge.

0.5 Definitionen

Das offizielle Glossar des GTBs definiert die deutschen Übersetzungen der Begriffe, die im ISTQB Standardglossar [ISTQB_GLOSSARY] enthalten sind. Eine Version des Glossars ist erhältlich vom ISTQB® und von der GTB Website. Kandidaten der Prüfungen für Foundation und Advanced Level werden Fragen auf der Basis des ISTQB-Standard-Begriffsglossars gestellt. Es wird jedoch auch erwartet, dass die Kandidaten auf diesem Level abweichende Definitionen kennen und mit ihnen arbeiten können.

Im Arbeitsumfeld der Sicherheitstester werden häufig einige englische Begriffe verwendet. In diesem Lehrplan wurden deshalb einige diese Begriffe verwendet, um die Lesbarkeit und Relevanz für den Leser zu gewährleisten. Die folgende Tabelle listet die entsprechenden Begriffe.

Englischer Begriff	**Begriff aus dem ISQTB Glossar**
Salting	Salzen
Skriptkiddie	Skript kiddie
Social Engineering	soziale Manipulation
Malware-Scanner	Schadprogramm-Scanner
Cross-Site Scripting	Webseitenübergreifendes Skripten
Session	Sitzung

HINWEIS: „Informationsschutz“ (Information Assurance: IA) wird ausschließlich in Abschnitt 2.4 behandelt. Der Begriff wird im deutschsprachigem Raum selten verwendet, kann aber grundsätzlich als eine Reihe von Verfahren und Tätigkeiten verstanden, die analog zu einem ISMS [ISO 27001], die Schaffung, Wahrung und Auditierung von Informationssicherheit unterstützen. Einem Zitat unter 2.4.3 folgt der Hinweis, dass IA ein breiter gefasstes Konzept als „Sicherheitstests“ darstellt – ähnlich wie die Qualitätssicherung (QA), die breiter als das Testen von Software gefasst ist.

Der Begriff „Informationssicherheit“ wird in den Abschnitten 2.2, 2.3.1, 2.7.2, 6 (Hintergrund), 6.1.3 und im gesamten Kapitel 9 verwendet.

Die Begriffe „Cyber-Sicherheit“ oder „IT-Sicherheit“ werden gar nicht verwendet; in einigen Abschnitten wird stattdessen mit der Begriff IA bzw. Informationssicherheit verwendet.

Die am Anfang jedes Kapitels in diesem Advanced Level Lehrplan aufgeführten Schlüsselbegriffe sind entweder im Standard-Begriffsglossar definiert, das beim Testen von Software zum Einsatz kommt und vom ISTQB veröffentlicht wurde, oder stammen aus der angegebenen Literatur.

0.6 Detailtiefe

Die Detailtiefe in diesem Lehrplan ermöglicht ein international einheitliches Lehren und Prüfen. Zum Erreichen dieses Ziels setzt sich der Lehrplan aus den folgenden Elementen zusammen:

1. Allgemeine Lernziele, die den Zweck des Advanced Level beschreiben.
2. Lernziele für jeden Wissensbereich, die das Ergebnis des kognitiven Lernens und die Denkweise beschreiben, die erreicht werden sollen.
3. Eine Liste mit zu vermittelnden Informationen, einschließlich einer Beschreibung und ggf. Verweise auf zusätzliche Quellen.
4. Eine Beschreibung der zu vermittelnden Kernkonzepte, einschließlich der Quellen wie akzeptierte Literatur oder Standards.
5. In diesem Lehrplan können bestimmte Werkzeuge, Methoden und Marken genannt werden. Das erfolgt nicht mit der Absicht, eine bestimmte Sicherheitslösung zu bewerben oder zu empfehlen.

Der Inhalt des Lehrplans ist kein Abriss des gesamten Wissens für den Advanced Sicherheitstester; er spiegelt die Detailtiefe wider, die in einem „Advanced Sicherheitstester"-Ausbildungskurs abgedeckt wird.

0.7 Lernziele / Wissensstand

Der Inhalt dieses Lehrplans, die Begriffe und die zentralen Elemente (d.h. die Zwecke) aller aufgelisteten Standards müssen zumindest verinnerlicht (K1) und verstanden (K2) werden, auch wenn sie in den Lernzielen nicht explizit genannt werden.

Die folgenden Lernziele haben nach unserer Definition Geltung für diesen Lehrplan. Jedes Thema im Lehrplan wird anhand der Lernziele für dieses Thema abgeprüft.

Kognitive Stufe 1: Wiedererkennen (K1)

Der Prüfungskandidat kann einen Begriff bzw. ein Konzept erkennen, sich an ihn/es erinnern und ihn/es wiedergeben.

Stichwörter: Erinnern, wiedergeben, erkennen, wissen.

Beispiel

Kann die Definition von „Risiko" erkennen als:

1. „ein Faktor, der zukünftige negative Konsequenzen bewirken kann; in der Regel ausgedrückt in Form von Schadensausmaß und Eintrittswahrscheinlichkeit."

Kognitive Stufe 2: Verstehen (K2)

Der Prüfungskandidat kann die Gründe oder Erläuterungen für die themenbezogenen Aussagen nennen sowie Fakten, Testkonzepte, Testverfahren (die Abfolge der Aufgaben erläutern) zusammenfassen, differenzieren, klassifizieren und mit Beispielen untermauern (z. B. Begriffe vergleichen).

Stichwörter: Zusammenfassen, klassifizieren, vergleichen, zuordnen, gegenüberstellen, (mit Beispielen) veranschaulichen, interpretieren, übersetzen, darstellen, ableiten, schlussfolgern, kategorisieren.

Beispiele

Erläutern, aus welchem Grund Sicherheitstests so früh wie möglich definiert werden sollten:

1. Um Sicherheitsmängel und Schwachstellen zu finden, wenn sie noch aufwandsärmer zu beseitigen sind.
2. Um zu vermeiden, ein System oder eine Anwendung zu entwickeln, der die ständige Beseitigung von Schwachstellen mittels korrigierender Softwareaktualisierungen droht.

Kognitive Stufe 3: Anwenden (K3)

Der Prüfungskandidat kann die richtige Anwendung eines Konzepts oder einer Technik benennen und sie auf einen gegebenen Kontext anwenden. K3 ist in der Regel auf prozedurales Wissen anwendbar. Es ist kein schöpferischer Akt wie beispielsweise die Analyse einer Softwareanwendung oder die Erzeugung eines Modells für ein gegebenes Softwareprogramm. Liegt ein Modell vor, erläutert der Lehrplan die prozeduralen Schritte, die für die Erzeugung von Testfällen auf der Basis dieses Modells erforderlich sind. Dann handelt es sich um K3.

Stichwörter: Implementieren, ausführen, nutzen, ein Verfahren befolgen, ein Verfahren anwenden.

Beispiel

1. Nutzen des allgemeinen Verfahrens für die Erzeugung von Sicherheitstestfällen durch Auswahl der Testfälle auf der Basis eines Zustandsübergangsdiagramms mit dem Ziel, alle Zustandsübergänge zu erfassen.

Kognitive Stufe 4: Analysieren (K4)

Der Prüfungskandidat kann Informationen mit Bezug auf ein Verfahren oder eine Technik zum besseren Verständnis in ihre einzelnen Bestandteile aufgliedern sowie zwischen Fakten und Rückschlüssen unterscheiden. Eine typische Anwendung ist die Analyse eines Dokuments, einer Software oder einer Projektsituation und das Vorschlagen geeigneter Maßnahmen zur Lösung eines Problems oder einer Aufgabe.

Stichwörter: Analysieren, differenzieren, auswählen, strukturieren, fokussieren, zuordnen, dekonstruieren, evaluieren, bewerten, überwachen, koordinieren, erstellen, synthetisieren, generieren, vermuten, planen, konzipieren, konstruieren, produzieren.

Beispiel

1. Analysieren der Sicherheitsrisiken eines Produkts sowie Vorschlagen präventiver und korrigierender Maßnahmen zur Risikominderung.
2. Auswählen von Sicherheitstestwerkzeugen, die einer gegebenen Situation mit Kenntnis über vergangene Sicherheitslücken am besten Rechnung tragen.

Literatur (zu den kognitiven Stufen von Lernzielen)

Bloom, B. S. (1956). Taxonomy of Educational Objectives, Handbook I: The Cognitive Domain, David McKay, Co. Inc.

Anderson, L. W. und Krathwohl, D. R. (Hrsg.) (2001). A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives, Allyn & Bacon.

1 Grundlagen des Testens der Sicherheit – 105 min

Schlüsselbegriffe

Datenschutz, ethischer Hacker, Informationssicherheit, Penetrationstest, Risikobewertung, Risikogefährdung, Risikominderung, Sicherheitsangriff, Sicherheitsaudit, Sicherheitsrichtlinie, Sicherheitsverfahren, Sicherheitsrisiko

Lernziele für das Thema „Grundlagen des Testens der Sicherheit“

1.1 Sicherheitsrisiken

AS[1]-1.1.1 (K2) die Bedeutung der Risikobewertung als Informationsquelle für die Planung von Sicherheitstests und deren Ausrichtung an geschäftlichen Erfordernissen verstehen

AS-1.1.2 (K4) die wichtigen zu schützenden Assets identifizieren und den Wert der einzelnen Assets und der benötigten Daten für die Ermittlung ihrer Sicherheitsstufe ermitteln können

AS-1.1.3 (K4) Den wirksamen Einsatz der Verfahren zur Risikobewertung in einer gegebenen Situation zur Ermittlung aktueller und zukünftiger Sicherheitsgefährdungen analysieren können

1.2 Informationssicherheitsrichtlinien und -verfahren

AS-1.2.1 (K2) Das Konzept der Sicherheitsrichtlinien und -verfahren sowie deren Anwendung in Informationssystemen verstehen

AS-1.2.2 (K4) Einen gegebenen Satz von Sicherheitsrichtlinien und -verfahren analysieren sowie die Ergebnisse von Sicherheitstests zur Ermittlung der Wirksamkeit bewerten können

1.3 Sicherheitsaudits und ihre Rolle beim IT-Sicherheitstest

AS-1.3.1 (K2) Den Zweck eines Sicherheitsaudits kennen

AS-1.3.2[2] Kein neues Lernziel! Wiederholung von: (K4) Den wirksamen Einsatz der Verfahren zur Risikobewertung in einer gegebenen Situation zur Ermittlung aktueller und zukünftiger Sicherheitsgefährdungen analysieren können

AS-1.3.3[3] Kein neues Lernziel! Wiederholung von: (K2) Den Zweck eines Sicherheitsaudits verstehen

[1] AS steht für Advanced Level Sicherheitstester
[2] Der Abschnitt 1.3.2 ergänzt das unter 1.1.3 bereits genannte Lernziel
[3] Der Abschnitt 1.3.3 ergänzt das unter 1.3.1 bereits genannte Lernziel

Funktionales Testen basiert auf einer Vielzahl von Elementen wie Risiken, Anforderungen, Anwendungsfälle und Modelle. Sicherheitstests basieren auf den Sicherheitsaspekten dieser Spezifikationen, dienen aber auch dazu, Sicherheitsrisiken, -verfahren und -richtlinien, das Verhalten von Angreifern und bekannte Sicherheitsschwachstellen zu verifizieren und zu validieren.

1.1 Sicherheitsrisiken

1.1.1 Die Rolle der Risikobewertung beim IT-Sicherheitstest

Die Ziele von Sicherheitstests richten sich nach bestehenden Sicherheitsrisiken. Diese Risiken werden im Rahmen einer Sicherheitsrisikobewertung ermittelt. Allgemeine Risikomanagement-Techniken werden in [ISTQB_FL_SYL] und [ISTQB_ATM_SYL] beschrieben sowie durch die Standards ISO 31000 [ISO 31000], ISO 27005 [ISO 27005] und NIST 800-30 [NIST 800-30] international verbindlich definiert. Während die ISO 31000 den allgemeinen Ablauf des Risikomanagements beschreibt, fokussiert die ISO 27005 speziell Informationssicherheitsrisiken. NIST 800-30 hat insbesondere für den nordamerikanischen Markt eine hohe Bedeutung.

Das Risiko gibt an, in welchem Maß eine Einheit von einem potenziellen Umstand oder Ereignis gefährdet ist und ist in der Regel eine Funktion folgender Aspekte:

1. Schäden, die entstehen würden, wenn der Umstand oder das Ereignis eintritt.
2. Eintrittswahrscheinlichkeit

Informationssicherheitsrisiken sind Risiken, die Folge des Verlusts von Vertraulichkeit, Integrität oder Verfügbarkeit von Informationen oder Informationssystemen sind. Sie beschreiben die potenziellen schädlichen Auswirkungen auf betriebliche Vorgänge (z. B. Ziele, Funktionen, Image oder Reputation), betriebliche Assets, Individuen, andere Unternehmen sowie ein gesamtes Land. [NIST 800-30]

Im Rahmen der Sicherheitsrisikobewertung kann ein Unternehmen ermitteln, welche Bereiche und Assets einem Risiko ausgesetzt sind und welchen Schweregrad die einzelnen Risiken haben. Für Sicherheitstester kann eine Sicherheitsrisikobewertung eine ergiebige Informationsquelle sein, auf deren Grundlage sich Sicherheitstests planen und konzipieren lassen. Darüber hinaus lassen sich mit einer Sicherheitsrisikobewertung Sicherheitstests so priorisieren, dass die am stärksten gefährdeten Bereiche mit der größten Stringenz und Überdeckung getestet werden können.

Durch Priorisierung von Sicherheitstests auf der Basis einer Sicherheitsrisikobewertung werden die Tests auf die Unternehmenssicherheitsziele abgestimmt. Damit das der Fall ist, muss die Sicherheitsrisikobewertung jedoch präzise die Sicherheitsgefährdungen des Unternehmens, die betroffenen Stakeholder und die zu schützenden Assets widerspiegeln. Einen guten Überblick über die verschiedenen Möglichkeiten, wie Sicherheitsrisikobewertung und Sicherheitstesten sich im Rahmen einer umfassenden Sicherheitsbewertung ergänzen, erlauben ETSI 203251 [ETSI203251] sowie ETSI 101182 [ETSI101182].

Dabei muss unbedingt klar sein, dass jede Risikobewertung (ob sicherheitsbezogen oder nicht) nur eine Momentaufnahme ist und auf limitierten Informationen basiert, was ungültige Annahmen und Schlussfolgerungen zur Folge haben kann. Sicherheitsrisiken und -projekte ändern sich innerhalb eines Unternehmens kontinuierlich, weil ständig neue Gefährdungen auftauchen. Daher müssen Sicherheitsrisikobewertungen in regelmäßigen Abständen durchgeführt werden. Das genaue Zeitintervall für die Durchführung von Sicherheitsrisikobewertungen variiert in Abhängigkeit vom Unternehmen und vom Grad der Veränderungen, die sich in ihm vollziehen. Manche Unternehmen führen alle drei bis sechs Monate Sicherheitsrisikobewertungen durch, andere einmal jährlich.

Grundsätzlich ist davon auszugehen, dass im Rahmen einer nicht-systematischen Risikobewertung Risikofaktoren übersehen und Zusammenhänge zwischen Risiken nicht richtig verstanden werden, sodass die Gefahr besteht, dass

auch erhebliche Risiken in der Risikobewertung nicht auftauchen und entsprechend nicht behandelt werden können. Die Gefahr fehlender oder unvollständiger Risikoinformationen und -zusammenhänge hängt auch vom Wissenstand der Person ab, die die Risikobewertung durchführen. Sie lässt sich durch Verwendung etablierter Techniken zur Risikobewertung jedoch verringern. Zu diesen Techniken zählen u.a. die Verwendung von Checklisten, die Durchführung von Brainstormings, die Beteiligung möglichst vieler Experten, sowie die bedarfsgerechte Verwendung formalisierter und formaler Techniken wie Fehlerbäume, Ursache-Wirkungs-Analysen oder Monte-Carlo-Simulationen. Eine ausführliche Liste von Techniken zur Risikobewertung findet sich in der ISO 31010 [ISO 31010].

1.1.2 Ermittlung der Assets

Nicht alle zu schützenden Informationen liegen in digitaler Form vor – z. B. kopierte Unterlagen (Verträge, Pläne, schriftliche Notizen, notierte Anmeldedaten und Passwörter). Dessen ungeachtet können diese Informationen einen hohen Wert haben. Daher muss die Frage gestellt werden, welche Informationen in digitaler Form vorliegen und welche nicht. Möglicherweise liegt das zu schützende Asset in digitaler und physischer Form vor. Bei der Ermittlung der zu schützenden Assets müssen folgende Fragen gestellt werden:

Welche Assets sind für das Unternehmen wertvoll?

Beispiele für sensible Informationen von hohem Wert:

1. Kundendaten
2. Business Pläne
3. Proprietäre Software, die vom Unternehmen entwickelt wurde
4. Systemdokumentation
5. Bilder und Diagramme, die Eigentum des Unternehmens sind
6. Geistiges Eigentum (z. B. Prozesse, Geschäftsgeheimnisse)
7. Kalkulationstabellen mit Finanzdaten
8. Präsentationen und Schulungskurse
9. Unterlagen
10. E-Mails
11. Mitarbeiterdatensätze
12. Steuererklärungen

Viele Assets sind informationsbasiert. Einige Assets in einem Unternehmen können aber physisch oder immateriell sein. Zu diesen Assets zählen beispielsweise:

1. physische Prototypen neuer und in Entwicklung befindlicher Geräte
2. die Fähigkeit, Dienstleistungen zu liefern
3. Reputation und Vertrauenswürdigkeit des Unternehmens

Wie wertvoll ist das Asset?

Viele sensible Assets haben einen materiellen Wert. Bei anderen bemisst sich der Wert eher an den Kosten und Folgen ihres Verlustes. Was würde beispielsweise ein Mitbewerber mit dem Business-Plan eines Konkurrenten tun?

Der Wert lässt sich nur schwer mit Sicherheit bestimmen; es gibt aber einige Methoden zur Bestimmung des Wertes digitaler Assets:

1. Der zukünftige Umsatz, den das Asset generieren soll.
2. Der Wert für einen Mitbewerber, der die Informationen vielleicht erhält.
3. Die Zeit und der Aufwand, die/der nötig ist, um das Asset neu zu schaffen.
4. Strafen für das Unvermögen, die Informationen zur benötigten Zeit bereitzustellen, z. B. für ein Audit oder ein Gerichtsverfahren.
5. Strafen für den Verlust von Kundendaten

Wo befinden sich die digitalen Assets?

Früher waren digitale Assets auf Servern, Computern oder Speichermedien wie externen Festplatten oder CDs gespeichert. Das ist ein alter und planloser Ansatz. Dennoch kann es noch viele sensible Daten auf alten CDs, DVDs und USB-Laufwerken geben. Deutlich sicherere Speicherorte für digitale Assets sind gesicherte Unternehmensserver mit starker Verschlüsselung für alle sensiblen Daten. Für den Zugriff auf sensible Daten, die auf sicheren Servern liegen, sollte eine Authentifizierung und Autorisierung erforderlich sein. Darüber hinaus kann es sein, dass weitere Schutzmaßnahmen wie digitale Zertifikate für den Zugriff auf sensible Informationen über das Internet erforderlich sind.

Der Speicherort von Daten ändert sich heute fortwährend. Heutzutage befinden sich große Mengen an Geschäftsdaten auf mobilen Geräten wie Smartphones, Tablets und USB-Sticks. Wenn digitale Informationen in einen Cloud-Speicher migriert wurden, gibt es neue Sicherheitsaspekte zu bedenken, was den Zugriff auf diese Daten angeht.

Die Bedeutung der Frage der Datenspeicherung sowie ihrer Absicherung wird deutlich an Fällen aus der Vergangenheit, in denen Menschen, denen sensible Daten anvertraut worden waren, mit einer Festplatte, CD oder DVD voller vertraulicher Kunden- und Geschäftsdaten einfach aus der Firma spazierten. Ein Fall dieser Art trug sich in den USA zu. Dort wurde eine Festplatte aus einem gesicherten Bereich einer staatlichen Sicherheitsbehörde gestohlen, auf der sich die Lohnabrechnungs- und Bankdaten von mehr als 100.000 ehemaligen und aktuellen Angestellten befanden. [Washington Post, 2007].

Wie erfolgt der Zugriff auf die digitalen Assets?

Folgende Methoden des Zugriffs auf digitale Assets sind weit verbreitet:

1. Zugriff per Computer über ein LAN- (Local Area Network) oder Wi-Fi-Netze
2. Zugriff aus der Ferne über ein VPN- (Virtual Private Network) oder ein Cloud-Laufwerk
3. Weitergabe physischer Datenspeicher (CDs, DVDs, USB-Laufwerke) von Person zu Person – eine technisch simple, aber durchaus weit verbreitete Praxis
4. Verschicken von Dateien per E-Mail
5. Nutzung anderer Austauschplattformen wie z. B. WhatsApp, Snapchat.

Wie sind die digitalen Assets geschützt?

Es gibt eine Reihe von Möglichkeiten, digitale Assets zu schützen, darunter:

1. Verschlüsselung (Welcher Typ, welche Stärke, wer hat Zugriff auf die Schlüssel?)
2. Authentifizierung und Token (Werden digitale Zertifikate benötigt? Gibt es adäquate Passwortrichtlinien und werden diese eingehalten?)
3. Autorisierung (Welche Zugriffsrechte wurden Benutzern gewährt, die mit digitalen Assets zu tun haben?)

1.1.3 Analyse von Verfahren der Risikobewertung

Die Bewertung von Sicherheitsrisiken ähnelt der standardmäßigen Risikobewertung sehr stark. Der wichtigste Unterschied ist, dass sicherheitsbezogene Fragen im Mittelpunkt stehen.

Eine Sicherheitsrisikobewertung sollte die Perspektiven externer Stakeholder einschließen (d.h. firmen-externe Personen oder Parteien, die in das Projekt oder Produkt involviert sind und Interesse an der Sicherheit des Projekts/Produkts haben). Zu diesen Stakeholdern gehören:

1. Kunden und Benutzer: Hilfreich für das Verständnis der Kundenperspektive, das Sammeln von Input für Sicherheitstests und die Etablierung einer guten Kommunikation zwecks Sicherheitskultur.
2. Öffentlichkeit und Gesellschaft: Es muss vermittelt werden, dass die Informationssicherheit eine gemeinschaftliche Aufgabe ist.
3. Aufsichtsbehörden: Notwendig für die Gewährleistung der Konformität mit geltenden Gesetzen bezüglich der Informationssicherheit.

Die Vorbereitung einer Risikobewertung umfasst folgende Aufgaben [NIST 800-30]:

1. Ermittlung des Zwecks der Bewertung
2. Ermittlung des Umfangs der Bewertung
3. Ermittlung der Annahmen und Randbedingungen im Zusammenhang mit der Bewertung
4. Ermittlung der Informationsquellen, die als Input für die Bewertung genutzt werden
5. Ermittlung des Risikomodells und der analytischen Konzepte (d.h. Bewertungs- und Analysekonzepte), die bei der Bewertung zu nutzen sind

Die Durchführung von Risikobewertungen umfasst folgende spezifische Aufgaben [NIST 800-30]:

1. Ermittlung der Gefährdungsquellen, die für das Unternehmen relevant sind
2. Ermittlung der Gefährdungsereignisse, die von diesen Quellen ausgehen können
3. Ermittlung von Schwachstellen im Unternehmen, die von Gefährdungsquellen mittels konkreter Gefährdungsereignisse ausgenutzt werden könnten, sowie der auslösenden Bedingungen für eine erfolgreiche Ausnutzung
4. Ermittlung der Wahrscheinlichkeit, mit der ermittelte Gefährdungsquellen spezielle Gefährdungsereignisse initiieren, sowie der Erfolgswahrscheinlichkeit von Gefährdungsereignissen

5. Ermittlung von schädlichen Folgen für betriebliche Vorgänge und Assets, Einzelpersonen, andere Unternehmen/Organisationen und ggf. das gesamte Land, die aus der Ausnutzung von Schwachstellen durch die Gefährdung herrühren

Kommunikation und Informationsaustausch umfassen folgende spezifische Aufgaben [NIST 800-30]:

1. Kommunizieren der Ergebnisse der Risikobewertung
2. Austausch von Informationen, die bei der Risikobewertung entwickelt wurden, um andere Maßnahmen des Risikomanagements zu unterstützen

1.2 Informationssicherheitsrichtlinien und -verfahren

1.2.1 Verstehen von Informationssicherheitsrichtlinien und -verfahren

Bedingt durch das Geschäftsmodell, die spezifische Branche und die einzigartigen Sicherheitsrisiken, mit denen ein Unternehmen konfrontiert ist, variieren Informationssicherheitsrichtlinien üblicherweise von Unternehmen zu Unternehmen. Trotz der großen Schwankungsbreite sind die Ziele von Sicherheitsrichtlinien sehr ähnlich. Die Grundlage aller Sicherheitsrichtlinien sollte eine Sicherheitsrisikobewertung sein, bei der spezielle Sicherheitsgefährdungen und deren Auswirkungen auf das Unternehmen untersucht werden. [Jackson, 2010]

Folgende Beispiele für Sicherheitsrichtlinien wären u.a. zu nennen [Jackson, 2010]:

Akzeptable Nutzung: Diese Richtlinie definiert Praktiken, an die sich ein Benutzer eines Computersystems halten muss, um nicht gegen die Sicherheitsrichtlinien und -verfahren des Unternehmens zu verstoßen. Sie regelt, was bei der Nutzung digitaler Ressourcen wie Netzwerken, Websites und Daten akzeptabel und nicht akzeptabel ist. Zudem kann die Richtlinie sowohl für interne als auch für externe Benutzer der Systeme eines Unternehmens gelten. Die Benutzer des Systems müssen die Richtlinie kennen und jederzeit befolgen. Um Missverständnisse und versehentliche Verstöße gegen die Richtlinie zu vermeiden, sollte diese konkrete Regeln für akzeptables Verhalten, nicht akzeptables Verhalten und erforderliches Verhalten definieren.

Mindestmaß an Zugriffsrechten: Diese Richtlinie definiert das Mindestmaß an Zugang, das für die Ausführung bestimmter Aufgaben benötigt wird. Diese Richtlinie soll verhindern, dass Personen Zugriffsrechte erhalten, die über die für die Erfüllung ihrer Aufgaben benötigten Rechte hinausgehen. Umfangreichere Zugriffsrechte als nötig würden Möglichkeiten für den versehentlichen oder bewussten Missbrauch von Benutzerrechten schaffen.

Netzwerkzugriff: Diese Richtlinie definiert Kriterien für den Zugriff auf verschiedene Arten von Netzwerken wie z. B. LANs oder W-LANs. Zudem kann diese Richtlinie definieren, was während des Aufenthalts in einem Netzwerk zulässig ist und was nicht. Häufig verbietet diese Richtlinie, dass Benutzer nicht genehmigte Geräte wie Router und Hotspots in das Netzwerk einbinden.

Fernzugriff: Diese Richtlinie regelt, welche Voraussetzungen für die Gewährung des Fernzugriffs auf Netzwerke für interne Mitarbeiter und externe (unternehmensfremde) Benutzer gegeben sein müssen. Häufig wird in dieser Richtlinie die Nutzung von VPNs geregelt.

Internetzugang: Diese Richtlinie definiert die zulässige Nutzung des Internets durch Mitarbeiter und Gäste eines Unternehmens. Der Geltungsbereich dieser Richtlinie schließt ein, auf welche Arten von Websites zugegriffen werden kann – z. B. keine Glücksspielseiten oder pornografischen Seiten – und regelt auch, ob die private Nutzung des Internets zulässig ist. Einige Aspekte dieser Richtlinie können zwar in der Richtlinie zur akzeptablen Nutzung geregelt

werden, häufig wird diese Richtlinie aufgrund der Vielzahl von Personen, die über das Internet Geschäfte betreiben, jedoch gesondert definiert.

Benutzerkontenverwaltung: Diese Richtlinie definiert die Erstellung, Pflege und Löschung von Benutzerkonten. Die regelmäßige Überprüfung von Benutzerkonten wird in dieser Richtlinie ebenfalls vorgeschrieben. Sie ist Voraussetzung für die Richtlinienkonformität.

Datenklassifizierung: Es gibt viele Arten Daten aus Sicherheitsperspektive zu klassifizieren. In diesem Lehrplan sprechen wir von „sensiblen Daten“ als Oberbegriff für sämtliche Daten, die vor Verlust geschützt werden müssen. Eine Richtlinie für die Datenklassifizierung definiert die verschiedenen Datentypen, die als sensibel eingestuft werden und daher geschützt werden müssen. Mit einer Datenklassifizierungsrichtlinie lässt sich steuern, dass Daten auf der Basis ihres Wertes, für das Unternehmen und für seine Kunden, geschützt werden. In der Regel ist der Geschäftsbereich, in dem die Daten erzeugt werden für ihre Klassifizierung auf Basis einer Standard-Klassifizierungsstruktur zuständig.

Nachstehend wird eine Klassifizierungsstruktur (aus einem Unternehmenskontext) mit einem Beispiel veranschaulicht:

1. Öffentlich: Jeder, ob innerhalb oder außerhalb des Unternehmens, kann diese Daten einsehen (z. B. Dokumente und Webseiten ohne Zugriffsbeschränkungen).

2. Vertraulich: Das ist normalerweise die Standardklassifizierung für intern erstellte Dokumente. Das können für den internen Gebrauch bestimmte E-Mails, Berichte und Präsentationen sein. Ein Beispiel dafür wäre ein Umsatzbericht. Nur autorisierte Benutzer dieser Daten dürfen in der Lage sein, mit Informationen dieser Ebene zu arbeiten. Vor der Weitergabe dieser Art von Daten an Dritte oder Berater müssen häufig Vertraulichkeitsvereinbarungen unterzeichnet werden.

3. Streng vertraulich: Dies ist ein höherer Vertraulichkeitsgrad für sensible Informationen, die nur bestimmten Personen in einem Unternehmen zugänglich sein dürfen. Das schließt Informationen wie Geschäftsgeheimnisse, Strategiepläne, Produktentwürfe und interne Finanzdaten ein. Die Weitergabe dieser Art von Daten ist nur mit ausdrücklicher Genehmigung des Inhabers der Daten erlaubt.

4. Geheim: Das sind Informationen, die häufig auf leitende Angestellte des Unternehmens beschränkt sind, die eine besondere Autorisierung für den Zugriff auf diese Daten benötigen. Die Offenlegung dieser Informationen kann dem Unternehmen erheblich schaden, z. B. in finanzieller Hinsicht. Aufgrund des hohen Risikos, das mit ihrem Verlust einhergeht, müssen private Informationen mit besonderer Sorgfalt geschützt werden. Zu diesen Daten zählen Forschungs- und Entwicklungsdaten, Fusions- und Übernahmepläne sowie Kundendaten, z. B. Kreditkarten- und Kontodaten.

5. Streng geheim: Im Unternehmenskontext sind dies Informationen, die ein Unternehmen von einem Dritten erhält, um Änderungen an ihnen vorzunehmen. Sie dürfen jedoch weder innerhalb noch außerhalb des Unternehmens bekannt werden. Ein Beispiel aus dem Wirtschaftskontext wäre ein Dokument eines Beraters, der an einer neuen Art von Technologie arbeitet, was die Kooperation anderer Unternehmen erfordert. Jedes dieser Unternehmen muss die Informationen geheim halten, bis die Technologie präsentationsreif ist. Diese Klassifizierung ist vergleichbar mit „streng vertraulich“, unterscheidet sich jedoch dahingehend, dass die Informationen keinen materiellen Wert für das Unternehmen selbst haben. In dieser Hinsicht unterscheidet sie sich auch vom Geschäftsgeheimnis. Die Offenlegung geheimer Informationen kann für das Unternehmen, andere Unternehmen bzw. das Land jedoch negative Konsequenzen haben. Bei Militär und Staat sind dies selbst entwickelte oder bezogene Informationen, die nur Geheimnisträgern bekannt sein dürfen. Beim Militär wären das Details von wissenschaftlichen

Projekten oder Forschungsprojekten, die neue Technologien zum Gegenstand haben bzw. Techniken mit direkter militärischer Anwendung, die von entscheidender Bedeutung für die Landesverteidigung sind.

Für den europäischen Raum sei hier zusätzlich und explizit auf die Bestimmung der Europäischen Datenschutzgrundverordnung [EUDSGVO] sowie der nationalen Datenschutzgesetze wie dem Bundesdatenschutzgesetz [BDSG] verweisen, die jeweils explizit den Umgang mit personenbezogenen Daten regeln. Als personenbezogene Daten gelten Einzelangaben über persönliche oder sachliche Verhältnisse einer bestimmten oder bestimmbaren natürlichen Person. Hierzu zählen u.a. Name, Alter, Familienstand, Anschrift, Kreditkarteninformationen, Gesundheitsdaten. In diesem Sinne sind Kundendaten und Personaldaten personenbezogene Daten. Diese sind gesondert zu schützen und unterliegen in Europa einer strengen Regulierung hinsichtlich Weitergabe und Verarbeitung.

Konfigurations- und Änderungsmanagement: Diese Richtlinie kann einen normalen betrieblichen Kontext haben. So kann sie z. B. beschreiben, wie Änderungen an Systemen zu managen und zu konfigurieren sind, um Ausfälle durch unerwartete Schäden zu verhindern.

Aus der Sicherheitsperspektive steuert das Konfigurationsmanagement wie Sicherheitseinstellungen auf sichere Geräte und Anwendungen angewendet werden. Es besteht das Risiko, dass eine nicht autorisierte Änderung an einem sicheren Gerät eine Schwachstelle zur Folge haben kann, die u.U. unerkannt bleibt.

Eine weitere Gefahr ist, dass eine nicht autorisierte Änderung am Programmcode oder an der Konfiguration einer Anwendung eine Sicherheitsschwachstelle erzeugen kann. Diese Richtlinie definiert zu verwendende Standardkonfigurationen, einen Genehmigungsprozess für alle Änderungen und einen Rollback-Prozess bei Problemen. Diese Richtlinie kann für alle IT-Services, Anwendungen und Geräte in einem Unternehmen gelten.

Server-Sicherheit: Diese Richtlinie definiert die Verantwortung von Server-Betreibern, sich beim Installieren, Konfigurieren und Betreiben von Servern und IT-Systemen an die im Unternehmen geltenden Sicherheitspraktiken sowie die Best Practices der Branche zu halten. Darüber hinaus müssen Basiskonfigurationen verpflichtend definiert und gepflegt werden. Als Beispiele für die in dieser Richtlinie beschriebenen Praktiken wären Sicherheitsanforderungen, Sicherung und Wiederherstellung sowie die Beschränkung der aktiven Dienste auf die für die Ausführung von Anwendungen notwendigen Dienste zu nennen. Zudem kann diese Richtlinie die Überwachungs- und Überprüfungsvorgaben regeln, um sicherzustellen, dass der Server richtig konfiguriert und aktualisiert wird.

Mobile Geräte: Für mobile Geräte gelten besondere Sicherheitsbelange. Daher ist für sie u.U. eine gesonderte Richtlinie nötig. So können Laptops und Smartphones beispielsweise leicht verloren gehen oder gestohlen werden, was mit dem Verlust von privaten Daten und Unternehmensdaten einhergeht. Zudem besteht für diese Geräte ein hohes Risiko mit Schadsoftware in Kontakt zu kommen. Diese Risiken erfordern spezielle Regeln und Vorkehrungen, die einzuhalten bzw. zu treffen sind, um die Risiken zu mindern und die Gefährdung der Sicherheit des Unternehmens zu minimieren. Diese Richtlinie kann vorschreiben, welche Daten verschlüsselt werden müssen, dass aktuelle Versionen von Schadsoftware-Scannern installiert und auf dem neuesten Stand gehalten werden und dass das Gerät mit Passwörtern zu schützen ist. Zudem sind in dieser Richtlinie die Arten von Unternehmensdaten definiert, die sich auf mobilen Geräten befinden dürfen. Auch die physische Sicherheit kann darin ein Thema sein: beispielsweise Kabelschlösser für Laptops und Verfahren für das Melden verloren gegangener oder gestohlener Geräte.

Gastzugang: Diese Richtlinie definiert die Praktiken, die für den Schutz des Unternehmens zu befolgen sind, wenn Gäste und andere sich in den Netzwerken des Unternehmens bewegen. Ein Aspekt dieser Richtlinie ist es von Gästen zu verlangen, dass sie die Richtlinien zur akzeptablen Nutzung lesen und akzeptieren, bevor sie Zugang zu Netzwerken erhalten. Diese Richtlinie lässt sich auf verschiedene Arten implementieren – z. B. indem man Gäste eine Richtlinie über die akzeptable Nutzung unterschreiben lässt und ihnen dann einen Zugangscode für den temporären Zugang aushändigt. Hauptzweck dieser Richtlinie ist es, die Sicherheitsstandards des Unternehmens durchzusetzen, ohne Gästen den Zugang zum Netzwerk oder Internet gänzlich verweigern zu müssen.

Physische Sicherheit: Diese Richtlinie definiert, welche Mittel der Zutrittskontrolle zu Räumlichkeiten nötig sind, weil die physische Nähe zu sicheren Geräten das Risiko eines Sicherheitsverstoßes erhöhen kann. Die Richtlinie kann aber

auch andere Risiken wie Stromausfälle, Diebstahl, Brand und Naturkatastrophen abdecken. Ebenfalls Gegenstand dieser Richtlinie ist, welche Geräte aus dem Unternehmen heraus- oder in das Unternehmen hineingebracht werden dürfen. Das gilt vor allem für Bereiche, in denen mit sensiblen Daten gearbeitet wird.

Passwort-Richtlinie: Diese Richtlinie definiert die Mindestanforderungen für starke Passwörter und andere sichere Passwort-Praktiken wie z. B. die Zeitintervalle zwischen der obligatorischen Änderung des Passwortes4 und die Art und Weise der Geheimhaltung der Passwörter (z. B. Nichtverwendung der „Passwort speichern“-Funktion von Browsern, Verbot der gemeinsamen Nutzung von Passwörtern sowie Verbot der Weitergabe von Passwörtern per E-Mail). Diese Richtlinie kann für Anwendungen, Benutzerkonten und andere passwortgeschützte Bereiche gelten.

Schutz gegen Schadsoftware: Diese Richtlinie definiert ein System von Abwehrmaßnahmen und Verhaltensweisen, um die Ausführung von Schadsoftware zu verhindern, zu erkennen und Schadsoftware zu entfernen. Weil die Infizierung mit Schadsoftware und Spyware bei einer Vielzahl von Quellen erfolgen kann, ist dies eine wichtige Richtlinie, die jeder im Unternehmen kennen und befolgen muss. So kann diese Richtlinie beispielsweise die Verwendung von USB-Laufwerken einschränken.

Reaktion auf sicherheitsrelevante Störfälle (Incident Response): Diese Richtlinie beschreibt, wie bei einem sicherheitsrelevanten Störfall reagiert werden muss. Dabei kann es sich um die Entdeckung von Schadprogrammen (Malware) und Verstößen gegen die Richtlinie über die akzeptable Nutzung bis hin zum unbefugten Zugriff auf sensible Daten handeln. Diese Richtlinie muss vorliegen, bevor sich ein Störfall ereignet, damit die geeigneten Gegenmaßnahmen nicht jedes Mal neu ermittelt werden müssen. Gegenstand dieser Richtlinie ist zudem die Kommunikation, darunter der Umgang mit Medien und die Benachrichtigung von Strafverfolgungsbehörden.

Audit-Richtlinie: Diese Richtlinie autorisiert Prüfer den Zugriff auf Systeme anzufordern, um ein Audit durchführen zu können. Das Audit-Team benötigt u.U. Zugriff auf Protokolldaten, Aufzeichnungen des Netzwerkverkehrs und andere forensische Daten.

Software-Lizensierung: Diese Richtlinie regelt, wie das Unternehmen die von ihm genutzte Software erwirbt und lizenziert. Bei Verletzung von Lizenzen für gewerblich-genutzte Software muss das Unternehmen mit Strafen und juristischen Schritten rechnen. Daher müssen Lizenzen unbedingt ermittelt und überwacht werden. Häufig wird in dieser Richtlinie ausdrücklich untersagt, nicht genehmigte Software herunterzuladen und zu installieren.

Elektronische Überwachung und Datenschutz: Unternehmen haben u.U. das Recht und die Pflicht, die elektronische Kommunikation zu überwachen, die über die Hardware und die Ressourcen des Unternehmens abgewickelt wird. Das schließt ggf. auch die E-Mail-Korrespondenz und soziale Medien ein. Diese Richtlinie schreibt vor, welche Überwachungsmaßnahmen das Unternehmen ergreift und welche Daten dabei erfasst werden. In Deutschland sind in diesem Kontext insbesondere die arbeitsrechtlichen Regelungen sowie der Datenschutz [EUDSGVO, BDSG] zu berücksichtigen. Da die Rechtslage von Land zu Land variiert, wird beim Verfassen dieser Richtlinie juristische Hilfe benötigt. [Jackson, 2010]

Sicherheitsverfahren

Sicherheitsverfahren definieren die einzelnen Schritte, die bei der Implementierung einer bestimmten Richtlinie oder eines Kontrollmechanismus sowie in Reaktion auf einen bestimmten Sicherheitsstörfall zu ergreifen sind. Offizielle, dokumentierte Verfahren erleichtern die Implementierung der Sicherheitsrichtlinien und vorgeschriebenen Kontrollmechanismen.

Richtlinien, Standards und Leitlinien beschreiben Sicherheitsmechanismen, die vorhanden sein müssen. Ein Verfahren hingegen beschreibt die Einzelheiten und erläutert, wie Sicherheitsmechanismen Schritt für Schritt zu implementieren sind. So kann beispielsweise ein Verfahren beschrieben werden, das erläutert, wie Benutzerzugriffsebenen zugewiesen

[4] Inzwischen empfiehlt die NIST (Special Publication 800-63B Digital Identity Guidelines) nicht mehr vorzugeben, dass Passwörter regelmäßig geändert werden müssen.

werden. In diesem Verfahren ist jeder Schritt, der zu ergreifen ist, detailliert aufgeführt, um sicherzustellen, dass die richtige Zugriffsebene gewährt wird, so dass die Benutzerrechte die geltenden Richtlinien und Standards erfüllen.

1.2.2 Analyse von Sicherheitsrichtlinien und -verfahren

Vor der Prüfung von Sicherheitsrichtlinien und -verfahren müssen die Ziele der Prüfung ermittelt und Kriterien definiert werden, nach denen ihre Angemessenheit bewertet wird. In bestimmten Fällen können die Kriterien von Standards wie COBIT [COBIT], ISO27001 [ISO27001], BSI IT-Grundschutz [BSIITG, BSI200-1, BSI200-2, BSI200-3] oder PCI [PCI] definiert werden.

Darüber hinaus muss Folgendes definiert werden:

1. Welche Ressourcen in Bezug auf Kompetenzen und Wissen in bestimmten zu prüfenden Bereichen benötigt werden.
2. Wie die Angemessenheit der Richtlinien und Verfahren bestimmt wird.
3. Was gemessen und bewertet werden soll (z. B. Wirksamkeit, Benutzbarkeit, Anpassung).
4. Wo sich die Richtlinien und Verfahren im Unternehmen befinden.
5. Eine Checkliste als Leitfaden für die Bewertung und Vereinheitlichung.

Die Checkliste dient als Leitfaden für den Prüfer. Ihr kann er entnehmen, wo er kontrollieren muss und was ihn erwartet. Werkzeuge wie z. B. für Passwort-Audits können das Testen bestimmter Kontrollmechanismen erleichtern. Mit ihnen lässt sich ermitteln, ob sie ihre Aufgabe erfüllen. Die erzeugten Daten können später bei der Risikobewertung genutzt werden. Der Prüfer prüft auf „wasserdichte“ Konformität mit Richtlinien, Vorschriften und Standards. Einige der Aufgaben in der folgenden Liste sind von ihrem Charakter her statisch, andere wie die Überwachungsprozesse sind dynamisch. Der Prüfer führt Folgendes durch:

1. Er prüft die Systemdokumentation.
2. Er befragt Beteiligte bezüglich ihrer Wahrnehmung im Hinblick auf die Wirksamkeit von Richtlinien und Verfahren.
3. Er spricht mit wichtigen Mitarbeitern, die in die zu prüfenden Prozesse eingebunden sind.
4. Er begutachtet Systeme und auszuführende Prozesse.
5. Er analysiert frühere Prüfergebnisse, um Trends zu erkennen.
6. Er analysiert Protokolle und Berichte.
7. Er prüft die Konfiguration der technischen Kontrollmechanismen wie die Konfiguration der Firewall und des Angriffserkennungssystems.
8. Er prüft stichprobenartig Datentransaktionen auf Anomalien oder verdächtige Vorgänge [Jackson, 2010].

Kontrollmechanismen

Sicherheitsrelevante Kontrollmechanismen sind technische oder administrative Schutz- oder Gegenmaßnahmen, die den Verlust oder die Nichtverfügbarkeit von Daten oder Systemen aufgrund von Angriffen auf mögliche Schwachstellen (d.h. Sicherheitsrisiken) verhindern oder minimieren sollen. [Northcutt, 2014] Ein Kontrollmechanismus in einem Lohnabrechnungssystem könnte z. B. darin bestehen, dass eine Änderung an den Gehaltsdaten eines

Gehaltsempfängers von zwei Mitarbeitern gesondert genehmigt werden muss. Sicherheitstester müssen die konkreten Kontrollmechanismen in ihrem Unternehmen kennen und im Rahmen der Sicherheitstests entsprechend testen.

Die wichtigsten sicherheitsbezogenen Kontrollmechanismen sind administrativer, technischer und physischer Art. In den einzelnen Kategorien lassen sich folgende Kontrollmechanismen implementieren: präventive, aufdeckende und wiederherstellende. Sie agieren im Zusammenspiel. Im Allgemeinen muss es Mechanismen aus jeder Kategorie geben, um ein Asset wirksam schützen zu können. [Jackson, 2010]

Für den deutschen Raum stellt das Bundesamts für Sicherheit in der Informationstechnik (BSI) mit den IT-Grundschutz-Katalogen [BSIITG] eine Menge vordefinierter und systematisch aufbereiteter Kontrollmechanismen zur Verfügung, mit denen sich eine Absicherung für Unternehmen mit niedrigem bis mittlerem Sicherheitsbedarf möglichst einfach realisieren und auditieren lässt. Die Kataloge definieren neben technischen Sicherheitsmaßnahmen insbesondere auch infrastrukturelle, organisatorische und personelle Schutzmaßnahmen. Die IT-Grundschutzmethodik und die IT-Grundschutzkataloge dienen Organisationen, Unternehmen und Behörden zur Zertifizierung. In Kombination mit der ISO 27001 ist darüber hinaus eine kombinierte Zertifizierung nach IT-Grundschutz und ISO 27001 möglich.

Sicherheitstests

Der wichtigste Unterschied von Sicherheitstests im Vergleich zur statischen Analyse von Sicherheitsrichtlinien und -verfahren ist die Verwendung der Testergebnisse aus Tests, die speziell dafür entwickelt wurden, die Wirksamkeit von Sicherheitsrichtlinien und -verfahren zu prüfen bzw. nachzuweisen. Der Schwerpunkt dieser Tests adressiert das Risiko, dass es zwar eine Sicherheitsrichtlinie gibt, diese auch umgesetzt wird, sie aber dennoch keine Wirksamkeit im Hinblick auf den Schutz des Assets entfaltet.

Bei der Durchführung der Bewertungen von Sicherheitsrichtlinien und -verfahren kann es vorkommen, dass dem Tester bestimmte Aufgaben, die ausgeführt werden sollen, vorgegeben werden. Ein Sicherheitstest dieser Aufgaben kann helfen zu ermitteln, wie wirksam Sicherheitsrichtlinien und -verfahren in der Praxis tatsächlich sind. Beispiel: Eine Passwortrichtlinie und das zugehörige Verfahren können auf dem Papier logisch und zweckerfüllend wirken, bei Verwendung eines Tools zum Knacken von Passwörtern jedoch versagen.

Sicherheitsrichtlinien und -verfahren können der Ausgangspunkt für Sicherheitstests sein; der Sicherheitstester muss dabei jedoch beachten, dass sich die Angriffe stets verändern und verfeinern. Neue Angriffsarten entstehen und wie bei jeder anderen Softwareanwendung können neue Fehler zutage treten – all dies spricht für die Durchführung von Sicherheitstests aus der Perspektive des Angreifers.

1.3 Sicherheitsaudits und ihre Rolle beim IT-Sicherheitstest

Ein Sicherheitsaudit ist eine manuelle Prüfung und Evaluierung, bei der Schwächen in den Sicherheitsprozessen und der Sicherheitsinfrastruktur eines Unternehmens aufgedeckt werden. Sicherheitsaudits auf prozeduraler Ebene (z. B. zum Prüfen interner Kontrollmechanismen) können manuell durchgeführt werden. Sicherheitsaudits auf architektonischer Ebene werden häufig mit Sicherheitsaudit-Werkzeugen durchgeführt, die auf eine bestimmte Anbieterlösung für Netzwerke, Serverarchitektur und Workstations zugeschnitten sein können.

Wie ein Sicherheitstest garantiert auch ein Sicherheitsaudit nicht, dass alle Schwachstellen gefunden werden. Das Audit ist jedoch eine weitere Aktivität im Sicherheitsprozess zur Ermittlung von Problemfeldern und Optimierungsbedarf.

Bei einigen Sicherheitsauditansätzen erfolgt im Rahmen des Audits auch das Testen. Das Sicherheitsaudit ist im Umfang jedoch wesentlich größer als ein Sicherheitstest. Bei Sicherheitsaudits werden häufig Bereiche wie Verfahren, Richtlinien und Kontrollmechanismen untersucht, die sich auf direktem Weg nur schwer testen lassen. Bei Sicherheitstests geht es zur Erhöhung der Sicherheit mehr um die Technik: z. B. die Firewall-Konfiguration, die richtige Anwendung von Authentifizierung und Verschlüsselung und die Arbeit mit Benutzerrechten.

Sicherheitsaudits werden von fünf Säulen getragen [Jackson, 2010]:

Bewertung: Im Rahmen von Bewertungen werden potenzielle Gefährdungen, wichtige Assets, Richtlinien und Verfahren sowie die Akzeptanz von Risiken durch das Management ermittelt und dokumentiert. Bewertungen sind keine einmaligen Ereignisse. Weil sich das Umfeld und das Unternehmen ständig verändern, müssen Bewertungen regelmäßig stattfinden. So lässt sich auch ermitteln, ob Sicherheitsrichtlinien noch relevant und wirksam sind.

Prävention: Präventive Maßnahmen gehen über die reine Technologie hinaus und erstrecken sich auch auf administrative, operative und technische Kontrollmechanismen. Prävention lässt sich nicht nur durch Technik, sondern auch über Richtlinien, Verfahren und Aufklärung erreichen. Es ist unrealistisch alle Angriffe verhindern zu können. Die Kombination aus Abwehrmaßnahmen kann jedoch dazu beitragen erfolgreiche Angriffe so schwer wie möglich zu machen.

Erkennung: Die Ermittlung einer Verletzung der Sicherheit oder eines Eindringens. Ohne adäquate Erkennungsmechanismen besteht das Risiko einer unerkannten Kompromittierung des Netzwerks. Aufdeckende Kontrollmechanismen können helfen Sicherheitsstörfälle zu erkennen und die Vorgänge im Netzwerk transparent zu machen. Die frühe Erkennung von Störfällen ermöglicht eine angemessene Reaktion zur schnellen Wiederaufnahme des Betriebs.

Reaktion: Bei guten Abwehr- und Erkennungsmechanismen verkürzt sich die Reaktionszeit enorm. Sicherheitsverletzungen sind zwar schlechte Nachrichten, trotzdem muss man wissen, wenn sie stattgefunden haben. Eine kurze Reaktionszeit ist entscheidend für die Minimierung der Folgen des Störfalls. Ein schnelles Reagieren setzt gute präventive Abwehr- und Erkennungsmechanismen voraus, die die nötigen Daten und den Kontext für die Reaktion liefern. Die Geschwindigkeit und Wirksamkeit der Reaktion auf Vorfälle ist ein wichtiger Indikator für die Wirksamkeit der Sicherheitsbemühungen eines Unternehmens.

Wiederherstellung: Die Wiederherstellung beginnt damit, zu ermitteln was passiert ist, damit Systeme wiederhergestellt werden können, ohne dieselbe Schwachstelle oder dasselbe Problem, das den Störfall ursprünglich auslöste, zu reproduzieren. Die Wiederherstellungsphase endet nicht mit der Wiederherstellung des Systems. Weiterführend ist eine Ursachenanalyse nötig, bei der ermittelt wird, welche Änderungen an Prozessen, Verfahren und Technologien vorgenommen werden müssen, um die Wahrscheinlichkeit für dieselbe Schwachstelle zukünftig zu verringern. Ein Prüfer muss sicherstellen, dass das Unternehmen einen Plan für die Wiederherstellung hat, der Schritte zur Verhinderung ähnlicher Störfälle in der Zukunft enthält.

1.3.1 Zweck eines Sicherheitsaudits

Nachstehend sind Punkte aufgelistet, die bei einem Sicherheitsaudit zutage treten können:

1. Inadäquate physische Sicherheit. Eine Sicherheitsrichtlinie kann die Verschlüsselung sämtlicher Kundendaten – sowohl bei der Speicherung als auch bei der Übertragung – vorschreiben. Beim Audit könnte sich herausstellen, dass einmal pro Woche eine Kundendatenakte an alle leitenden Angestellten geschickt wird. Diese Akte wird anschließend entsorgt, einige Angestellte werfen sie jedoch fahrlässiger Weise einfach in den Papierkorb. Dort könnte sie von jemandem gefunden werden, der den Abfall gezielt durchsucht („Dumpster Diving“).

2. Inadäquates Passwort-Management. Eine Sicherheitsrichtlinie kann vorschreiben, dass die Benutzer alle 30 Tage ihr Passwort ändern müssen. Bei einem Sicherheitsaudit stellt sich heraus, dass die Passwörter zwar geändert werden, viele Benutzer aber der Einfachheit halber jeweils zwischen „Passwort A“ und „Passwort B“ wechseln. (Die Passwort-Historie ist eine gängige Funktion von Passwort-Audit-Werkzeuge.)

3. Inadäquate Steuermechanismen für Benutzerrechte und die gemeinsame Nutzung von Zugriffsrechten. Ein negatives Prüfergebnis wäre z. B., wenn Benutzer mehr Zugriffsrechte haben, als sie eigentlich für die Erledigung ihrer Arbeit benötigen. Oder wenn die Dateien eines Benutzers, die eigentlich privat sein sollen, im Netzwerk für alle zugänglich sind. Das gilt vor allem für Benutzer mit Notebooks und besonders für jene, die über Wi-Fi-Verbindungen zuhause oder an öffentlichen Hotspots auf das Intranet zugreifen.

4. Inadäquate Sicherheit auf Serverebene. Zu den konkret zu prüfenden Punkten zählen:
   - Port-Zuweisung und –Sicherheit
   - Schutz von Daten
   - Schutz von Benutzerkonten (Anmeldedaten und sonstige schutzwürdige Informationen)

5. Inadäquate Anwendung von Software-Updates von Sicherheitslösungen Dritter

6. Inadäquate Mechanismen zur Angriffserkennung

7. Inadäquate Maßnahmenpläne für die Reaktion bei einer Sicherheitsverletzung

1.3.2 Ermittlung, Bewertung und Minderung von Risiken

Sobald die Problemfelder im Rahmen des Audits ermittelt wurden, muss das Risiko bewertet und ein Optimierungsplan aufgestellt werden. Der Bericht des Prüfers kann Empfehlungen sowie Verweise auf weitere Risikobereiche enthalten. Davon ausgehend kann die Ermittlung, Bewertung und Minderung der Risiken[5] geplant werden.

Die Risikoermittlung ist der Prozess der Dokumentierung eines Risikos oder Risikobereichs. Im Kontext der Informationssicherheit sind die Risiken sicherheitsbezogen. Die Risikobewertung ist die Gewichtung der ermittelten Risiken. Es muss klar sein, dass traditionelle Risikobewertungsmodelle für Informationssicherheitsrisiken nicht ausreichen. Ein Bewertungsmodell oder -konzept für Informationssicherheitsrisiken muss speziell auf Informationssicherheitsrisikoprofile ausgerichtet sein.

Sicherheitsrisiken werden häufig nach ihrer Risikogefährdung gewichtet. Die Risikogefährdung ist das rechnerische Produkt aus dem potenziellen Schaden oder Verlust und der Eintrittswahrscheinlichkeit des Schadens/Verlusts. Ein

[5] Die Risikominderung ist eine Form der Risikobehandlung. Im Original ISTQB-Glossar wird als Übersetzung von Risk Mitigation die Risikobegrenzung angegeben, was weniger den reduzierenden Charakter (minderung) betont als solche Aktivitäten, die das Risiko unterhalb einer Grenze halten. Im diesem Syllabus wird die Risikominderung in Anlehnung an die ISO 31000 verwendet.

Beispiel: Welche Folgen hätte es, wenn die Kontodaten eines Kunden kompromittiert sind? Was, wenn dieser Kunde $100 Millionen an Vermögenswerten hinterlegt hat?

Die Eintrittswahrscheinlichkeit lässt sich durch Anwendung eines Sicherheitsrisiko-Bewertungsmodells ermitteln. Normen wie die ISO 31010 beschreiben verschiedene generische Bewertungsmodelle für Risiken. Für den Bereich der Sicherheitsrisikobewertung finden sich Spezialisierungen in der ISO 27005, in BSI 200-3, im NIST-Dokument 800-30 sowie mit der OWASP-Risiko-Rating-Methodik [OWASP2] bei der OWASP. Die folgenden Angaben sind beispielhaft und stammen aus [NIST 800-30].

Risikomodelle definieren die zu bewertenden Risikofaktoren und die Zusammenhänge zwischen diesen Faktoren. Risikofaktoren sind Kenndaten, die als Input für Risikomodelle genutzt werden, um bei Risikobewertungen die Risikograde zu bestimmen. Zudem werden Risikofaktoren umfassend in der Risikokommunikation genutzt, um zu unterstreichen, was in bestimmten Situationen oder Kontexten starken Einfluss auf die Risikograde hat.

Typische Risikofaktoren sind: Gefährdung, Schwachstelle, Schadensausmaß, Wahrscheinlichkeit und auslösende Bedingung. Risikofaktoren können auf noch feinere Kenndaten aufgeschlüsselt werden (z. B. lassen sich Gefährdungen in Gefährdungsquellen und Gefährdungsereignisse aufschlüsseln). Diese Definitionen müssen von Unternehmen vor der Durchführung von Risikobewertungen dokumentiert werden, weil sich diese Bewertungen zur wirksamen Ermittlung von Risiken auf die klar definierten Attribute Gefährdungen, Schwachstellen, Schadensausmaß und andere Risikofaktoren stützt.

Gefährdungen

Eine Gefährdung ist ein Umstand oder Ereignis mit dem Potenzial, die operativen Vorgänge in einem Unternehmen sowie Assets, Personen, andere Unternehmen oder ein Land zu beeinträchtigen – über ein Informationssystem mittels unbefugtem Zugriff, Zerstörung, Offenlegung oder Modifikation von Informationen und/oder Dienstblockade (Denial of Service).

Gefährdungsereignisse werden von Gefährdungsquellen ausgelöst. Eine Gefährdungsquelle ist durch Folgendes charakterisiert:

1. Die Absicht und Methode, die auf die Ausnutzung einer Schwachstelle abzielt, oder
2. eine Situation und Methode, durch die eine Schwachstelle unbeabsichtigt ausgenutzt werden kann.

Zu diesen Gefährdungsquellen zählen im Allgemeinen:

1. Feindliche Cyberangriffe oder physische Angriffe
2. Menschliche Fehler durch Versäumnis oder fehlerhaftes Handeln
3. Strukturelles Versagen von unternehmenskontrollierten Ressourcen (z. B. Hardware, Software, Kontrollmechanismen für das Umfeld)
4. Naturkatastrophen und menschgemachte Katastrophen sowie Unfälle und Ausfälle, die sich der Kontrolle des Unternehmens entziehen.

Es wurden verschiedene Klassifizierungssysteme für Gefährdungsquellen entwickelt. Einige nutzen die Art der schädlichen Auswirkungen als Ordnungsprinzip. Mehrere Gefährdungsquellen können dasselbe Gefährdungsereignis auslösen oder bewirken. So kann z. B. ein Bereitstellungsserver durch einen DoS-Angriff, das geplante Handeln eines böswilligen Systemadministrators, einen administrativen Fehler, einen Hardwarefehler oder einen Stromausfall offline genommen werden.

Schwachstellen und auslösende Bedingungen

Eine Schwachstelle ist eine Sicherheitslücke in Informationssystemen, Sicherheitsverfahren, internen Kontrollmechanismen oder der Implementierung, die von einer Gefährdungsquelle ausgenutzt werden kann.

Die meisten Schwachstellen in Informationssystemen stehen mit sicherheitsbezogenen Kontrollmechanismen im Zusammenhang, die entweder (bewusst oder unbewusst) nicht angewendet wurden oder angewendet wurden, aber mangelhaft sind. Nicht zu vergessen sind jedoch auch Schwachstellen, die im Laufe der Weiterentwicklung von Unternehmenszielen/Geschäftsfunktionen, der Änderung von Betriebsumgebungen, der Verbreitung neuer Technologien und dem Auftauchen neuer Gefährdungen quasi von selbst entstehen können. Im Kontext derartiger Änderungen können bestehende sicherheitsbezogene Kontrollmechanismen unzureichend werden. Dann müssen sie erneut auf ihre Wirksamkeit überprüft werden. Aufgrund der Tendenz, dass diese Kontrollmechanismen mit der Zeit potenziell an Wirksamkeit verlieren, ist es umso wichtiger, im gesamten Softwareentwicklungslebenszyklus Risikobewertungen durchzuführen. Zudem müssen fortwährend Überwachungsprogramme laufen, um ständig über die Sicherheitslage des Unternehmens informiert zu sein.

Schwachstellen werden nicht nur innerhalb von Informationssystemen ermittelt. Betrachtet man Informationssysteme aus weiter gefasster Perspektive, lassen sich auch in der Leitungs-/Verwaltungsstruktur von Unternehmen Schwachstellen finden (z. B. fehlende wirksame Risikomanagement-Strategien und adäquater Risikorahmen, mangelnde innerbetriebliche Kommunikation, widersprüchliche Entscheidungen über relative Prioritäten von Unternehmenszielen/Geschäftsfunktionen oder falsche Ausrichtung der Unternehmensarchitektur in Bezug auf die Förderung von Zielen/Geschäftsaktivitäten).

Schwachstellen lassen sich auch in folgenden Bereichen finden: externe Beziehungen (z. B. Abhängigkeiten von bestimmten Energiequellen, Lieferketten, Informationstechnologien und TK-Anbietern), Ziele/Geschäftsprozesse (wie z. B. Prozesse mit ungenauer Definition bzw. mangelndem Risikobewusstsein) sowie Unternehmens-/Informationssicherheits-Architektur (z. B. schlechte architekturbezogene Entscheidungen, die fehlende Vielfalt oder Widerstandsfähigkeit von unternehmenseigenen Informationssystemen zur Folge haben).

Schadensausmaß

Das Schadensausmaß eines Gefährdungsereignisses beschreibt die Schwere des Schadens, der als Folge einer unbefugten Offenlegung, Veränderung oder Vernichtung von Informationen oder dem Verlust der Verfügbarkeit von Informationen oder Informationssystemen zu erwarten ist. Von diesem Schaden kann eine Vielzahl von Stakeholdern innerhalb und außerhalb des Unternehmens betroffen sein:

1. Stellen-/Behördenleiter
2. Verantwortliche für Unternehmensziele/Unternehmen
3. Inhaber/Verwalter von Informationen
4. Verantwortliche für Leitbild-/Geschäftsprozesse
5. Inhaber von Informationssystemen
6. Einzelne/Gruppen im öffentlichen oder privaten Sektor, die sich auf das Unternehmen verlassen – im Grunde jeder mit einem ureigenen Interesse an den Vorgängen, Assets oder Mitarbeitern des Unternehmens, einschließlich anderer Unternehmen, die mit dem Unternehmen kooperieren, oder ein ganzes Land

Folgende Informationen müssen vom Unternehmen ausdrücklich dokumentiert werden:

1. der für die Durchführung von Folgenabschätzungen verwendete Prozess

2. Annahmen in Bezug auf Folgenabschätzungen
3. Quellen und Methoden für den Erhalt von Folgeninformationen
4. die Logik hinter gezogenen Schlussfolgerungen im Hinblick auf Folgenabschätzungen

Unternehmen können explizit definieren, wie bestehende Prioritäten und Werte zur Ermittlung von hochwertigen Assets und dem Potenzial von feindlichen Auswirkungen auf Geschäftsinteressenten beitragen. Werden derartige Informationen nicht definiert, können Prioritäten und Werte in Bezug auf die Ermittlung von Zielen für Gefährdungsquellen und die entsprechenden Folgen für das Unternehmen in der Regel aus strategischen Plänen und Richtlinien abgeleitet werden. So signalisieren z. B. abgestufte Sicherheitskategorien die Folgen der Kompromittierung verschiedener Informationstypen für das Unternehmen.

Wahrscheinlichkeit

Die Eintrittswahrscheinlichkeit beschreibt die Wahrscheinlichkeit, mit der ein Gefährdungsereignis in einen Schaden mündet, unabhängig vom Ausmaß des zu erwartenden Schadens. Dies ist ein gewichteter Risikofaktor, der auf der Analyse der Wahrscheinlichkeit basiert, mit der eine gegebene Gefährdung in der Lage ist, eine gegebene Schwachstelle (oder mehrere Schwachstellen) auszunutzen. Der Wahrscheinlichkeitsrisiko-Faktor kombiniert eine Schätzung der Wahrscheinlichkeit, dass das Gefährdungsereignis ausgelöst wird, mit der Schätzung der Schadenswahrscheinlichkeit (d.h. der Wahrscheinlichkeit, mit der das Ereignis in einem Schaden mündet).

Bei feindlichen Gefährdungen basiert die Bewertung der Eintrittswahrscheinlichkeit in der Regel auf folgenden Aspekten:

1. Absicht des Angreifers
2. Fähigkeiten des Angreifers
3. Zielrichtung des Angreifers

Bei anderen Gefährdungsereignissen wird die Eintrittswahrscheinlichkeit mittels historischer Beweise, empirischer Daten und weiterer Faktoren abgeschätzt. Zu beachten ist dabei: Die Wahrscheinlichkeit, dass ein Gefährdungsereignis initiiert wird oder stattfindet, wird mit Bezug auf einen konkreten Zeitrahmen bewertet (z. B. die nächsten sechs Monate, das nächste Jahr oder die Zeitspanne bis zum Erreichen eines festgelegten Etappenziels).

Wenn ein Gefährdungsereignis im (festgelegten oder impliziten) Zeitraum mit annähernder Sicherheit initiiert wird oder stattfindet, kann bei der Risikobewertung die geschätzte Häufigkeit des Ereignisses berücksichtigt werden. Die Eintrittswahrscheinlichkeit eines Gefährdungsereignisses kann auch auf dem Status des Unternehmens basieren (wie z. B. seine Kernziele/-geschäftsprozesse, Unternehmensarchitektur, Informationssicherheits-Architektur, Informationssysteme und Umgebungen, in denen diese Systeme operieren). Zu berücksichtigen sind ferner auslösende Bedingungen sowie die Präsenz und Wirksamkeit der sicherheitsbezogenen Kontrollmechanismen, die zum Schutz gegen unbefugtes/unerwünschtes Verhalten, zur Erkennung und Begrenzung des Schadens und/oder zur Aufrechterhaltung bzw. Wiederherstellung von Missions-/Geschäftsfähigkeit eingerichtet wurden.

Ermitteln des Schweregrades eines Sicherheitsrisikos

Die Bewertung der Eintrittswahrscheinlichkeit und die Folgenabschätzung lassen sich zur Berechnung eines Gesamtschweregrads für das Risiko kombinieren. Spezielle Bewertungspunktzahlen können als Basis für die Komplettierung der Risikomatrix dienen. Andernfalls können Schätzwerte (niedrig, mittel oder hoch) genutzt werden.

Die Punktezahl für die Risikomatrix kann auf einer Skala von 0–9 basieren, bei der die Zahlenwerte anhand spezifischer Kriterien bestimmt werden. Risikowahrscheinlichkeitskriterien für den Datenschutz könnten z. B. wie folgt ermittelt werden:

1. 0 bis <3 (niedrig): Personenbezogene Daten werden nicht auf lokalen Geräten gespeichert und bei Speicherung auf sicheren Geräten verschlüsselt.
2. 3 bis <6 (mittel): Personenbezogene Daten können auf Geräten wie Notebooks gespeichert sein, sind allerdings verschlüsselt.
3. 6 bis 9 (hoch): Es ist nicht genau bekannt, ob sich personenbezogene Daten auf lokalen Geräten befinden. Die Verschlüsselung kann nicht garantiert werden.

Analog dazu lassen sich Risikofolgen-Kriterien, gestützt auf spezifische Kriterien, auf derselben Skala von 0 bis 9 definieren. Zum Beispiel:

1. 0 bis <3 (niedrig): Die Kompromittierung personenbezogener Daten würde weniger als 200 Menschen betreffen.
2. 3 bis <6 (mittel): Die Kompromittierung personenbezogener Daten würde zwischen 200 und 1000 Menschen betreffen.
3. 6 bis 9 (hoch): Die Kompromittierung personenbezogener Daten würde mehr als 1000 Menschen betreffen.

Ungeachtet des Weges, auf dem der Tester zu den Schätzwerten für Wahrscheinlichkeit und Schadenshöhe kommt, lassen sich die Schätzwerte zu einer finalen Schweregradeinstufung für die Risikoposition kombinieren. Wenn gute Informationen zu den Auswirkungen auf die Geschäftstätigkeit vorliegen, sollten diese statt der technischen Informationen genutzt werden. Liegen keine solchen Informationen vor, sind die errechneten Werte als nächstbessere Alternative zu verwenden.

Nachstehend finden Sie ein Beispiel für eine Risikomatrix, mit der sich der Schweregrad einzelner Risiken ermitteln lässt.

Gesamtschweregrad des Risikos				
Risikofolgen	Hoch	Mittel	Hoch	Kritisch
	Mittel	Mittel	Mittel	Hoch
	Niedrig	Niedrig	Niedrig	Mittel
		Niedrig	Mittel	Hoch
		Risikowahrscheinlichkeit		

Wenn die Wahrscheinlichkeit in der obigen Beispielmatrix mittelhoch und der zu erwartende Schaden hoch ist, ist auch der Gesamtschweregrad hoch.

Im Risikobewertungsbericht muss zudem vermerkt sein, ob es sich um ein permanentes Risiko handelt. Permanente Risiken erhöhen die Wahrscheinlichkeit, dass ein Verlust auch tatsächlich eintritt.

Der Schweregrad eines Risikos bestimmt die relative Dringlichkeit der Minderung dieses Risikos. Je höher der Schweregerad ist, desto zeitnaher muss reagiert werden. Der Detailgrad einer jeden Risikobewertung entspricht dem Zweck der Risikobewertung und dem Typ von Input, der nötig ist, um nachfolgende Wahrscheinlichkeits- und Folgenabschätzungen vornehmen zu können.

1.3.3 Mensch, Prozess und Technik

Bei den IT-Praktiken eines Unternehmens kommen zudem drei Faktoren zum Tragen: Mensch, Prozess und Technik. Alle drei Faktoren haben einen Einfluss auf die Sicherheit. Bei Chris Jackson heißt es in Network Security Auditing [Jackson, 2010] dazu: „Sämtliche Sicherheitsstörfälle von Einbrüchen bis hin zum Verlust von Kundendatensätzen

haben ihre Ursache meist in einer Unzulänglichkeit, die sich dem Menschen, dem Prozess oder der Technik zuschreiben lässt."

Menschen: Das können Endbenutzer, Systemadministratoren, Dateneigentümer und leitende Mitarbeiter des Unternehmens sein. Jeder Mensch hat einen anderen Kompetenzstand, andere Einstellungen und Pläne. Das hat einen Einfluss darauf, inwieweit Sicherheitsfragen für ihn relevant sind und welchen Einfluss er auf die Wirksamkeit von sicherheitsbezogenen Kontrollmechanismen hat. Vorhandene Sicherheitsrichtlinien und -verfahren sowie Sicherheitskontrollmechanismen werden unwirksam, wenn Menschen sie nicht einhalten. Befolgt der Mensch die Sicherheitsrichtlinien nicht, kann eine Abhilfe nötig sein – z. B. eine Sicherheitsschulung oder Bestrafung bei Nichtbefolgung. Unternehmensstrukturen und Sicherheitsrichtlinien werden häufig von Menschen von innerhalb und außerhalb des Unternehmens getragen.

Prozess: Prozesse definieren, wie IT-Dienstleistungen, darunter auch sicherheitsbezogene Dienstleistungen, bereitgestellt werden. Im Sicherheitskontext schließen Prozesse die Verfahren und Standards/Normen ein, die zum Schutz wertvoller Assets eingerichtet werden. Um wirksam zu sein, müssen Prozesse definiert werden. Sie müssen ferner aktuell und konsistent sein sowie den sicherheitsbezogenen Best Practices entsprechen. Prozesse definieren Rollen und Zuständigkeiten, Kontrollmechanismen, Werkzeuge und die spezifischen Schritte zur Durchführung einer Aufgabe.

Technik: Technik schließt die Anlagen/Einrichtungen, die Ausrüstung sowie die Computer-Hardware und -Software ein, die die technische Grundlage für die Geschäftstätigkeit bilden. Technik versetzt Menschen in die Lage sich wiederholende Tätigkeiten schneller und fehlerfreier als manuell auszuführen. Manche Aufgaben wie z. B. der Passwortschutz wären ohne die richtigen Werkzeuge sogar unmöglich. Die Gefahr liegt darin, dass Menschen die Technik falsch einsetzen und dadurch schneller Fehler machen.

Die genannten drei Aspekte kann man sich als „eisernes Dreieck" vorstellen, dessen drei Ecken im Zusammenspiel eine IT-Komplettlösung bilden. Wird einer der drei Aspekte ignoriert, leidet die gesamte IT-Leistungsbereitstellung und -Sicherheitsarbeit.

Bei der Evaluierung sicherheitsrelevanter Kontrollmechanismen muss sich der Prüfer das System aus Sicht eines Angreifers anschauen und antizipieren, wie Mensch, Prozess und Technik ausgenutzt werden könnten, um unbefugten Zugriff auf wertvolle Assets zu erlangen. Die Geschäftsführung ist häufig überrascht, dass die sicher geglaubten Sicherheitsmechanismen gar nicht sicher sind. Ob eine bestimmte Sicherheitsvorkehrung funktioniert und wirksam ist, lässt sich nur durch Testen des Systems aus Sicht des Angreifers herausfinden. Das wird häufig als ethisches Hacking oder Penetrationstest bezeichnet.

Hier wird die Beziehung zwischen Audit und Test am deutlichsten sichtbar. Beim Audit werden Unzulänglichkeiten und Bereiche aufgedeckt, die unbedingt getestet werden müssen. Mit Sicherheitstests lässt sich nachweisen oder widerlegen, dass die sicherheitsbezogenen Kontrollmechanismen vorhanden und wirksam sind.

Beispielszenario:

Die Steuerbehörde eines Landes ist Gegenstand eines Sicherheitsaudits. Eine der Erkenntnisse des Audits ist, dass es Angreifern möglich ist, eine gefälschte Steuerrückerstattung im System zu hinterlegen und auf Kosten des betrogenen Steuerzahlers eine Rückzahlung zu erhalten. Diese Erkenntnis wird durch Sicherheitstests untermauert, und das Risiko wird als „kritisch" eingestuft. Die Steuerbehörde erkennt die Möglichkeit eines solchen betrügerischen Vorgehens an, beschließt jedoch, die Schwachstelle erst im nächsten Jahr zu beseitigen.

Betrogene Steuerzahler, die alle vorgeschriebenen Sicherheitsverfahren eingehalten haben, können die Steuerbehörde verklagen, die ja über den Fehler beim Einreichen der Steuererklärung Bescheid wusste. In diesem Fall wäre die Steuerbehörde für den Betrug haftbar zu machen.

2 Zwecke, Ziele und Strategien von Sicherheitstests – 130 min

Schlüsselbegriffe

Cross-site Scripting, Datenmaskierung, Dienstblockade, Informationsschutz, Sicherheitsrichtlinie, Sicherheitstest, Sicherheitsschwachstelle, Softwareentwicklungslebenszyklus, Teststrategie

Lernziele für das Thema „Zwecke, Ziele und Strategien von Sicherheitstests“

2.1 Einleitung

Für diesen Abschnitt gibt es keine Lernziele.

2.2 Der Zweck von Sicherheitstests

AS-2.2.1 (K2) Die Notwendigkeit von Sicherheitstests in einem Unternehmen verstehen können – einschließlich der Vorteile für das Unternehmen, wie bspw. die Risikominderung sowie größeres Vertrauen

2.3 Der Unternehmenskontext

AS-2.3.1 (K2) Den Einfluss von Projektrealitäten, geschäftlichen Einschränkungen, Softwareentwicklungslebenszyklen und anderen Überlegungen auf die Aufgabe des Sicherheitsteststeams analysieren und verstehen können

2.4 Ziele von Sicherheitstests

AS-2.4.1 (K2) Erläutern können, warum Ziele von Sicherheitstests an der Sicherheitsrichtlinie des Unternehmens und anderen Testzielen im Unternehmen ausgerichtet sein müssen

AS-2.4.2 (K3) Für ein gegebenes Projektszenario die Fähigkeit beweisen können, Sicherheitsteststeziele auf der Grundlage von Funktionalität, Technologiemerkmalen und bekannten Schwachstellen auszuwählen

AS-2.4.3 (K2) Den Zusammenhang zwischen Informationsschutz und Sicherheitstests verstehen

2.5 Umfang und Überdeckungsgrad von Sicherheitsteststeziele

AS-2.5.1 (K3) Für ein gegebenes Projektszenario die Fähigkeit beweisen zu können, den Zusammenhang zwischen Sicherheitsteststeziele und der Notwendigkeit einer starken Integrität von sensiblen digitalen und physischen Assets zu definieren

2.6 Sicherheitstestvorgehensweisen

AS-2.6.1 (K4) In einer gegebenen Situation ermitteln können, welche Sicherheitstestvorgehensweisen wahrscheinlich am erfolgreichsten sind

AS-2.6.2 (K4) Eine Situation, in der ein gegebenes Sicherheitstestvorgehensweisen fehlgeschlagen ist, analysieren und die wahrscheinlichen Ursachen für das Fehlschlagen ermitteln können

AS-2.6.3 (K3) Für ein gegebenes Projektszenario die Fähigkeit beweisen können, die verschiedenen Stakeholder zu ermitteln und den Nutzen von Sicherheitstests für jede Stakeholdergruppe zu veranschaulichen

2.7 Optimierung der Sicherheitstestpraktiken

AS-2.7.1 (K4) Die Hauptleistungsindikatoren (KPIs) zur Ermittlung von Sicherheitstestpraktiken mit und ohne Optimierungsbedarf analysieren können

2.1 Einleitung

Vor Anwendung spezialisierter Sicherheitstest-Techniken muss der breiter gefasste Kontext von Sicherheitstest und ihre Rolle innerhalb des jeweiligen Unternehmens verstanden werden. Mit diesem Verständnis lassen sich folgende Fragen beantworten:

1. Warum sind Sicherheitstests notwendig?
2. Welchen Zweck haben Sicherheitstests?
3. Wie fügen sich Sicherheitstests in das Unternehmen ein?

Sicherheitstests unterscheiden sich in zwei wichtigen Aspekten von anderen Formen des funktionalen Tests [ISTQB_ATTA_SYL]:

1. Standardtechniken zur Auswahl von Eingabedaten für Tests können wichtige Fragen der Sicherheit unberücksichtigt lassen bzw. sind nicht in der Lage diese abzudecken.
2. Die Symptome von Sicherheitsfehlerzuständen unterscheiden sich stark von denen anderer Arten von funktionalen Tests.

Bei Sicherheitstests wird die Anfälligkeit eines Systems für Gefährdungen getestet. Dazu wird versucht, die Sicherheitsrichtlinie des Systems aktiv auszuhebeln. Nachstehend sind potenzielle Gefährdungen aufgelistet, die bei Sicherheitstests adressiert werden müssen [ISTQB_ATTA_SYL]:

1. Unbefugtes Kopieren von Anwendungen oder Daten
2. Kontrolle des nicht autorisierten Zugangs (z. B. die Möglichkeit, Aufgaben auszuführen, für die keine Berechtigung vorliegt). Im Mittelpunkt dieser Tests stehen Benutzerrechte, Zugang und Zugangsrechte. Diese Informationen müssen in den Spezifikationen für das System zu finden sein.
3. Software, die bei der Ausführung ihrer vorgesehenen Funktion unbeabsichtigte Nebeneffekte aufweist. Ein Beispiel dafür wäre ein Media Player, der Audiosignale korrekt abspielt, aber dazu unverschlüsselte, temporär gespeicherte Dateien schreibt, was im Nebeneffekt eine Schwachstelle erzeugt, die von Softwarepiraten genutzt werden kann.
4. In eine Webseite eingefügter Programmcode, der bei nachfolgenden Benutzern u.U. ausgeführt wird (Cross-site Scripting oder XSS). Dieser Programmcode kann bösartig sein.
5. Pufferüberlauf, der durch die Eingabe von langen Datenstrings (länger als vom Programm handhabbar) über die Schnittstellen eines Systems ausgelöst werden kann. Eine Pufferüberlauf-Schwachstelle bietet eine Möglichkeit für die Ausführung bösartiger Codebefehle.
6. Dienstblockade (DoS), die verhindert, dass Benutzer mit einer Anwendung interagieren können (z. B. durch Überfrachtung eines Webservers mit sinnlosen Anfragen)
7. Das Abfangen, Imitieren und/oder Ändern und anschließende Weiterleiten von Kommunikationsinhalten (wie z. B. Kreditkartentransaktionen) durch einen Dritten, so dass dessen Präsenz für den Benutzer unbemerkt bleibt („Man in the Middle“-Angriff)
8. Knacken von Verschlüsselungscodes, die dem Schutz sensibler Daten dienen

9. Logische Bomben, die mit böser Absicht in Programmcode eingefügt werden können und nur unter bestimmten Bedingungen aktiviert werden (z. B. an einem bestimmten Tag). Wenn logische Bomben zünden, können sie böswillige Akte wie die Löschung von Dateien oder das Formatieren von Festplatten auslösen.
10. Code-Injection, durch das Senden nicht vertrauenswürdiger Daten an einen Interpreter, der unautorisiert Befehle ausführt. Code-Injection wird häufig durch Fehler in der Eingabevalidierung eines Programms möglich. Betroffen sind unter anderem SQL-, LDAP-, XPath-Abfragen, abhängig vom Interpreter aber auch Betriebssystembefehle oder Programmargumente.

Sicherheitstests müssen mit allen anderen Entwicklungs- und Testaktivitäten verzahnt werden. Das erfordert die Berücksichtigung der einzigartigen Erfordernisse des Unternehmens, möglicher bestehender Sicherheitsrichtlinien, aktueller Sicherheitstest-Kompetenzen sowie bestehender Teststrategien.

2.2 Der Zweck von Sicherheitstests

Wie Softwaretests im Allgemeinen können auch Sicherheitstests nicht garantieren, dass ein System oder Unternehmen völlig sicher vor Angriffen ist. Sicherheitstests können jedoch dazu beitragen, Schwachstellen zu identifizieren, Risiken zu ermitteln und die Wirksamkeit bestehender Schutzmaßnahmen nachzuweisen.

Audits und die Prüfung von Sicherheitspraktiken sind ergänzende Maßnahmen für Sicherheitstests.

Sicherheitstests offenbaren zudem, ob beim Schutz digitaler Assets die nötige Sorgfalt eingehalten wurde. Bei einer Verletzung der Sicherheit können rechtliche Schritte drohen. Wenn ein Unternehmen nachweisen kann, dass es angemessene Maßnahmen zum Schutz digitaler Assets ergriffen hat – z. B. das Testen auf Schwachstellen –, kann das bei einem Verfahren entlastend wirken. Zudem können Sicherheitstests Kunden die Gewissheit geben, dass ein Unternehmen angemessene Maßnahmen zum Schutz sensibler Informationen ergreift.

2.3 Der Unternehmenskontext

Sicherheitstests bilden neben weiteren Testarten häufig eine Art von Funktionstests. Bei begrenzter Zeit für Tests muss ein Testmanager entscheiden, wie viel getestet werden kann. Das schließt auch Sicherheitstests ein. Es ist nicht unüblich, dass Sicherheitstests als Sonderfall gesehen und daher an ein entsprechend spezialisiertes Unternehmen ausgelagert werden. Der Umfang der Sicherheitstests wird letztlich von den geschäftlichen oder unternehmerischen Risiken bestimmt, die sicherheitsbasiert sind. Gibt es in einem Unternehmen zahlreiche Sicherheitsrisiken, müssen umfangreichere Sicherheitstests durchgeführt werden.

Wie Softwaretesten ist die Informationssicherheit eine Lebenszyklus-Aktivität. Sicherheitserfordernisse müssen in den Anforderungen definiert, im Entwurf ausgedrückt und im Programmcode implementiert werden. Dann lässt sich mit Sicherheitstests die Richtigkeit und Zweckmäßigkeit der Sicherheitsimplementierung überprüfen und nachweisen. Sicherheit lässt sich nicht nachträglich in den Programmcode einbinden noch wird Software allein durch Testen sicherer. Nur wenn Sicherheit mittels sicherer Programmier- und Entwurfstechniken in Software integriert wird, kann Software sicher sein.

Neben Risikostufen, Sicherheitstestkompetenzen und Lebenszyklusansätzen haben praktische Beschränkungen wie begrenzte Zeit, Ressourcen und Umfang einen großen Einfluss auf den Erfolg des Sicherheitstest-Teams in einem Unternehmen.

2.4 Ziele von Sicherheitstests

2.4.1 Die Ausrichtung von Sicherheitstestzielen

Die Sicherheitstest-Richtlinie kann geschrieben werden, sobald die Sicherheitsrichtlinie des Unternehmens von der Unternehmensleitung gebilligt wurde. Es ist wichtig, dass die Ziele der Sicherheitstests, die in der Sicherheitstest-Richtlinie aufgeführt sind, im Einklang mit der Gesamtsicherheitsrichtlinie des Unternehmens stehen. Andernfalls werden entweder nicht autorisierte Sicherheitstests durchgeführt oder diese Tests erfüllen nicht die gewünschten Zwecke.

2.4.2 Ermittlung von Zielen von Sicherheitstests

Ziele von Sicherheitstests sind mit Zielen von funktionalen Tests vergleichbar, nur dass bei ihnen die Sicherheit im Mittelpunkt steht. Für jeden Sicherheitsaspekt des Systems oder der Anwendung sollte es mindestens ein Sicherheitstestziel geben.

Zudem müssen Sicherheitstestziele auf die Merkmale einer Technologie (z. B. Internet, Mobil, Cloud, LAN) und bekannte Schwachstellen – sowohl in der Anwendung als auch allgemeine Schwachstellen – ausgerichtet sein. Sicherheitstestziele können beispielsweise folgende sein:

1. Nachweis, dass bei der Passwort-Authentifizierung die richtige Regel für die Passwortstärke angewendet wird
2. Nachweis, dass alle Dateneingabefelder eingabevalidiert sind, um SQL-Injection-Angriffe zu verhindern
3. Nachweis, dass die Kundendatendateien mit der richtigen Stärke verschlüsselt sind

2.4.3 Der Unterschied zwischen Informationsschutz und Sicherheitstests

Informationsschutz (Information Assurance: IA) ist wie folgt definiert: „Maßnahmen, die Informationen und Informationssysteme durch Gewährleistung ihrer Verfügbarkeit, Integrität, Authentifizierung, Vertraulichkeit und Nichtabstreitbarkeit schützen. Zu diesen Maßnahmen gehört die Gewährleistung der Wiederherstellbarkeit von Informationssystemen durch Einbindung von Schutz-, Erkennungs- und Reaktionsfähigkeiten.“ [NISTIR 7298]

In Deutschland wird der Begriff Informationsschutz selten verwendet. Bekannter ist der Begriff ISMS als Beschreibung des Managementsystems zur Herstellung, Aufrechterhaltung, und kontinuierlicher Verbesserung der Informationssicherheit in Organisationen. Aufbau, Struktur und Verfahren eines ISMS werden durch [ISO27001] und [BSIITG] definiert.

Der Zweck von Sicherheitstestens besteht darin, festzustellen, ob ein System die spezifizierten Sicherheitsanforderungen erfüllt. Die Anforderungen sollten Aussagen zu Sicherheitsfunktionen, Leistungs-einschränkungen und Softwarezuverlässigkeit enthalten. [ETSI101583]

Vergleicht man die Begriffe Informationsschutz (Information Assurance: IA) und Sicherheitstest, dann ist IA der weiter gefasste Begriff. Die Beziehung zwischen beiden ähnelt der zwischen Quality Assurance (QA: Qualitätssicherung) und Softwareprüfung.

2.5 Umfang und Überdeckungsgrad von Sicherheitstestzielen

Je wichtiger die Integrität sensibler digitaler und physischer Assets ist, desto größer muss der Umfang sein, den die Sicherheitstestziele abdecken. Sicherheitstestziele beschreiben im Wesentlichen den Umfang der Sicherheitstests. Ist er zu klein, entsteht kein Vertrauen, dass die Sicherheit angemessen ist. Ist er zu groß, sind möglicherweise die Ressourcen erschöpft, bevor der Test abgeschlossen werden kann.

Sicherheitstestziele sollten beschreiben, was die Sicherheitstests in Bezug auf die Verifizierung und Validierung der bestehenden Schutzmaßnahmen für sensible digitale und physische Assets erreichen sollen. Sicherheitstestziele müssen sich direkt auf konkrete Assets, Schutzmaßnahmen, Risiken und die Ermittlung von Sicherheitsschwachstellen beziehen.

2.6 Sicherheitstestvorgehensweise

Die Sicherheitsteststrategie wird definiert, um die allgemeine Ausrichtung eines Unternehmens im Hinblick auf das Testen der Sicherheit zu formalisieren und zu kommunizieren. Dann werden Vorgehensweisen oder Konzepte zur Umsetzung der Teststrategie definiert.

2.6.1 Analyse der Sicherheitstestvorgehensweise

Jedes Unternehmen hat eigene Interessen, bezogen auf seine Geschäfte und Aufträge, die ihrerseits bestimmte Sicherheitsteststrategien und -konzepte zur Ermittlung und Minderung von Sicherheitsrisiken erfordern. Es gibt jedoch auch Sicherheitsinteressen, die vielen Unternehmen gemein sind.

Eine Sicherheitstestvorgehensweise wird auf Projektebene definiert und sollte im Einklang mit der Testrichtlinie und -strategie des Unternehmens stehen. Die Sicherheitstestvorgehensweise eines Projekts ist eine individuell zusammengestellte Mischung aus Techniken, Werkzeugen und Kompetenzen, die den Sicherheitstestzielen für dieses Projekt Rechnung trägt.

Bei der Analyse einer Situation zum Zweck der Erstellung einer Sicherheitstestvorgehensweise sind folgende Punkte zu berücksichtigen:

1. die Quelle/Herkunft der zu testenden Systeme oder Anwendungen
2. etwaige vorherige Sicherheitstests
3. die Sicherheitsrichtlinie
4. die Sicherheitstestrichtlinie
5. Sicherheitsrisikobewertungen, die im Unternehmen bereits vorgenommen wurden
6. die technische Umgebung (z. B. Softwaretyp und -version, Frameworks, Programmiersprachen, Betriebssysteme)
7. Sicherheitstest-Kompetenzen im Testteam
8. allgemeine Sicherheitsrisiken
9. die Struktur der Testorganisation
10. die Struktur des Projektteams

11. die Erfahrung des Testteams mit verschiedenen Sicherheitstestwerkzeugen
12. Randbedingungen (z. B. limitierte Ressourcen, limitierte Zeit, fehlender Zugriff auf Umgebungen)
13. Annahmen (z. B. über frühere Arten von durchgeführten Sicherheitstests)

Unterschiedliche technische Umgebungen und Anwendungstypen (z. B. Client/Server, Internet, Mainframe) erfordern häufig unterschiedliche Sicherheitstestvorgehensweisen und -strategien. Im Rahmen der Softwareentwicklung kann beispielsweise ein Code-Review auf sicherheitsbezogene Schwachstellen nötig sein; beim Testen von Software kann wiederum die Pseudonymisierung/Anonymisierung der Testdaten erforderlich sein. Webbasierte Anwendungen haben andere Schwachstellen als Mainframe-Systeme und müssen daher auch anders auf Sicherheit getestet werden.

Manche Schwachstellen gibt es auch bei mehreren Technologien. So können beispielsweise Pufferüberläufe bei mobilen Anwendungen, sowie bei Client-Server- und Webanwendungen auftreten. Unterschiede bestehen lediglich darin, wie die einzelnen Technologien das Speichermanagement handhaben. Das Ergebnis ist in allen Umgebungen dasselbe: Unvorhersehbares Verhalten der Software, welches es einem Angreifer ermöglicht, auf eine Anwendung zuzugreifen und Aufgaben auszuführen, die normalerweise nicht zulässig sind.

Inadäquate Absicherung der Daten kann bei jeder Technologie und in jeder Umgebung auftreten. Die Verschlüsselung von Daten in webbasierten und mobilen Umgebungen erfolgt jedoch anders als in einer Mainframe-Umgebung. Die Verschlüsselungsalgorithmen mögen zwar dieselben (oder ähnliche) sein, der Unterschied ist jedoch, dass die Daten bei webbasierten und mobilen Anwendungen zusätzlich auch während der Übermittlung über das Internet geschützt werden müssen. Bei allen Technologien sollten sensible Daten in verschlüsselter Form gespeichert werden. Es gab Störfälle, bei denen sensible Mainframe-Daten physisch (auf Magnetband) und unverschlüsselt an andere geschickt wurden. „Die auf Privatkredite und das Forderungsmanagement spezialisierte Cattles Group musste einräumen, zwei Backup-Bänder mit den Daten von rund 1,4 Millionen Kunden verloren zu haben.“ [ComputerWeekly]

2.6.2 Analyse des Fehlschlagens von Sicherheitstestvorgehensweisen

In diesem Zusammenhang sollte man unbedingt wissen, dass es verschiedene Grade des Fehlschlagens gibt. Nur weil eine sicherheitsbezogene Schwachstelle nicht erkannt und beseitigt wird, heißt das nicht zwangsläufig, dass die Sicherheitstestvorgehensweise fehlgeschlagen ist. Es gibt schlicht zu viele mögliche sicherheitsbezogene Schwachstellen, und jeden Tag werden neue entdeckt. Es kann allerdings auch vorkommen, dass die Sicherheitstestvorgehensweise nicht angemessen genug sind, um Sicherheitsrisiken wirksam zu erkennen, und dadurch sensible Daten oder andere digitale Assets kompromittiert werden.

Mit der Ursachenanalyse lässt sich u.U. ermitteln, warum eine Sicherheitstestvorgehensweise gescheitert ist. Mögliche Ursachen sind:

1. mangelndes Engagement der Führungsebene bei der Etablierung der Sicherheitstests
2. fehlende Bereitstellung der für die Umsetzung der Sicherheitsteststrategie benötigten Ressourcen seitens der Unternehmensführung (Fehlen von Mitteln, Zeit, Personal)
3. fehlende wirksame Implementierung der Sicherheitstestvorgehensweise (z. B. fehlende Kompetenzen, die für die Ausführung der erforderlichen Aufgaben benötigt werden)
4. fehlendes Verständnis und Unterstützung der Sicherheitstestvorgehensweise seitens des Unternehmens
5. fehlendes Verständnis und Unterstützung der Sicherheitstestvorgehensweise seitens der Stakeholder
6. fehlendes Verständnis für die Sicherheitsrisiken

7. mangelnde Abstimmung zwischen Testvorgehensweise und Sicherheitsrichtlinie des Unternehmens
8. mangelnde Abstimmung zwischen Testvorgehensweise und Sicherheitstestrichtlinie und -strategie des Unternehmens
9. fehlendes Verständnis für den Zweck des Systems
10. fehlende technische Informationen über das System (mit falschen Annahmen als Folge)
11. fehlende wirksame Sicherheitstestwerkzeuge
12. fehlende Kompetenzen im Hinblick auf Sicherheitstests

2.6.3 Ermittlung der Stakeholder

Damit die Notwendigkeit von Sicherheitstests auch durch das Management erkannt wird, hilft die Entwicklung eines Business Case. In diesem Business Case müssen die Risiken von Sicherheitslücken und die Vorteile einer wirksamen Sicherheitstestvorgehensweise für ein konkretes Projekt klar definiert werden.

Die verschiedenen Stakeholder sehen unterschiedliche Nutzeffekte in einer Sicherheitstestvorgehensweise:

1. Die Geschäftsleitung sieht den Schutz des Unternehmens als Nutzen.
2. Das obere Management sieht u.U. Sorgfaltspflicht.
3. Geschäftskunden sehen u.U. den Schutz vor Missbrauch.
4. Compliance-Beauftragte (für interne Sicherheitsrichtlinien des Unternehmens) sehen u.U. die Gewährleistung der Einhaltung der gesetzlichen Verpflichtungen durch das Unternehmen.
5. Zuständige im regulatorischen Bereich (externe Sicherheitsgesetze) sehen u.U. den Nutzen, dass die Sicherheitsvorschriften befolgt werden.
6. Datenschutzbeauftragte sehen u.U. den Nutzen, dass personenbezogene Daten sicher sind und beim Schutz digitaler Assets mit der gebührenden Sorgfalt gearbeitet wird.

2.7 Optimierung der Sicherheitstestpraktiken

Voraussetzung für die Optimierung der Sicherheitstestpraktiken ist zunächst eine Prüfung der bestehenden Praktiken. Die Sicherheitstestpraktiken müssen auf objektive Art und Weise evaluiert werden. Diese Evaluierung basiert auf den Schlüsselmetriken für Sicherheitstestziele. Auf ihrer Grundlage kann der Erfolgsgrad der wichtigen Strategieelemente ermittelt werden.

Diese Praktiken müssen wie folgt evaluiert werden:

1. aus kurz- und langfristiger Perspektive
2. unter Berücksichtigung von Prozess und Unternehmen
3. unter Berücksichtigung von Mensch, Werkzeugen, Systemen und Techniken

Die Schlüsselmetriken umfassen u.a.:

1. Überdeckungsgrade von Sicherheitsrisiken durch Tests

2. Überdeckungsgrade von Sicherheitsrichtlinien und -strategien durch Tests
3. Überdeckungsgrade von Sicherheitsanforderungen durch Tests
4. Wirksamkeitsgrade früherer Sicherheitstests – basierend darauf, wann und wo Sicherheitsschwachstellen gefunden wurden. Das schließt Schwachstellen vor und nach der Freigabe ein.

3 Sicherheitstestprozesse – 140 min

Schlüsselbegriffe

Account Harvesting, Passwort-Knacken, Social Engineering, Soziale Manipulation, Testvorgehensweise, Testkonzept, Testprozess

Lernziele für das Thema „Sicherheitstestprozesse“

3.1 Definition des Sicherheitstestprozesses

AS-3.1.1 (K3) Für ein gegebenes Projekt die Elemente eines wirksamen Sicherheitstestprozesses definieren zu können

3.2 Planung von Sicherheitstests

AS-3.2.1 (K4) Eine Sicherheitstestvorgehensweise sowie dessen Stärken und Schwächen analysieren und erläutern zu können

3.3 Entwurf von Sicherheitstests

AS-3.3.1 (K3) Für ein gegebenes Projekt auf der Grundlage einer gegebenen Sicherheitstestvorgehensweise und unter Berücksichtigung funktionaler und struktureller Sicherheitsrisiken konzeptuelle (abstrakte) Sicherheitstests realisieren zu können

AS-3.3.2 (K3) Testfälle zur Validierung von Sicherheitsrichtlinien und -verfahren realisieren können

3.4 Durchführung von Sicherheitstests

AS-3.4.1 (K2) Schlüsselelemente und Merkmale einer effektiven Sicherheitstestumgebung verstehen

AS-3.4.2 (K2) Die Bedeutung der Planung und Einholung von Genehmigungen vor der Durchführung jedes Sicherheitstests verstehen

3.5 Auswertung von Sicherheitstests

AS-3.5.1 (K4) Sicherheitstestergebnissen für die Ermittlung folgender Punkte analysieren können:

1. Art der Sicherheitsschwachstelle
2. Ausmaß der Sicherheitsschwachstelle
3. Potenzielles Schadensausmaß der Sicherheitsschwachstelle
4. Vorgeschlagene Abhilfemaßnahmen
5. Optimale Testberichtsmethoden

3.6 Wartung von Sicherheitstests

AS-3.6.1 (K2) Die Bedeutung der Wartung von Sicherheitstestprozessen vor dem Hintergrund der Weiterentwicklung von Technologie und Gefährdungen verstehen

3.1 Sicherheitstestprozesse - Definition

Wie Softwaretests im Allgemeinen sind Sicherheitstests auch eine Lebenszyklus-Aktivität. Werden die Sicherheitsmaßnahmen nicht im gesamten Projektverlauf implementiert und getestet, können schwere Sicherheitslücken auftreten, die u.U. nie richtig geschlossen werden. Der Sicherheitstestprozess muss auf den Entwicklungsprozess abgestimmt sein, damit bei Bedarf entsprechende Testaktivitäten durchgeführt werden können.

Die Risiken und Erfordernisse der Sicherheitstests sind bei jedem Unternehmen anders – bedingt durch den Charakter des Unternehmens, den technischen Umgebungen, dem Softwareentwicklungsprozess und den Geschäftsrisiken. Daher muss der Sicherheitstestprozess im Kontext dieser Faktoren definiert werden.

3.1.1 ISTQB-Sicherheitstestprozess

Tabelle 3.1 zeigt den Zusammenhang zwischen dem allgemeinen ISTQB-Testprozess gemäß Beschreibung in den ISTQB-Lehrplänen für Foundation und Advanced Level sowie dem ISTQB-Sicherheitstestprozess. Für jeden Schritt im Prozess werden Sicherheitstestaufgaben als Beispiele gezeigt.

Tabelle 3.1: ISTQB-Sicherheitstestprozess

ISTQB-Testprozess	ISTQB-Sicherheitstest-prozess	Beispiele für Sicherheitstestaufgaben
Testplanung und -steuerung	**Planung und Steuerung von Sicherheitstests: Ziel ist es, einen angemessenen Umfang für das Testen zu definieren, der den Sicherheitsrisiken entspricht.**	• **Sicherheitstestziele definieren** • **Umfang der Sicherheitstests definieren** • **Ressourcen für die Sicherheitstests ermitteln** • **Schätzung und Zeitpläne für die Sicherheitstests definieren**

ISTQB-Testprozess	ISTQB-Sicherheitstest-prozess	Beispiele für Sicherheitstestaufgaben

Testanalyse und -entwurf	**Analyse und Entwurf von Sicherheitstests: Ziel ist es, auf der Basis von Risikobewertungen, Audits und bekannten Schwachstellen, Erkenntnisse über konkrete Sicherheitsgefährdungen zu gewinnen.**	• **Elemente prüfen, die als Basis für die Sicherheitstests dienen: Sicherheitsrisikobewertungen, Sicherheitsanforderungen und Sicherheitsrichtlinien**
		• **Sicherheitstestbedingungen auf Basis folgender Aspekte definieren:**
		  • **Testziele**
		  • **Sicherheitsrisiken**
		  • **Sicherheitsstandards und bekannte Schwachstellen**

Realisierung und Ausführung der Tests	**Realisierung und Ausführung der Sicherheitstests: Ziel ist es, konzeptuelle Tests in manuelle oder automatisierte Tests umzusetzen. Weiteres Ziel ist es, diese Tests unter Rückgriff auf eine Vielzahl von Perspektiven durchzuführen: interner Benutzer, externer Benutzer, böswilliger Benutzer usw.**	• **Sicherheitstestfälle, Testszenarien, Testskripte oder andere Testspezifikationen erstellen** • **Funktionale Sicherheitstests auf der Basis definierter Sicherheitstestspezifikationen durchführen** • **Funktionale Sicherheits- und Penetrationstests auf Basis der Kenntnisse und Intuition des Testers durchführen**

ISTQB-Testprozess	ISTQB-Sicherheitstest-prozess	Beispiele für Sicherheitstestaufgaben
Bewertung der Endekriterien und Abschlussberichterstattung	**Evaluierung und Abschlussberichterstattung über Ergebnisse der Sicherheitstests: Dies erfolgt zur schnellstmöglichen Auswertung einzelner Tests und zur Dokumentation neuer Gefährdungen häufig parallel zur Ausführung der Tests.**	• **Konkrete Sicherheitsschwachstellen auf der Basis der Testergebnisse ermitteln** • **Auf der Basis der durchgeführten Sicherheitstests Sicherheitsrisikostufen evaluieren** • **Der Geschäftsführung und anderen befugten Parteien Zwischen- und Abschlussergebnisse der Tests melden**
Abschluss der Testaktivitäten	**Abschluss der Testaktivitäten: Ziel ist es, die Testaktivitäten zum Abschluss zu bringen, damit die Tests gewartet und regelmäßig durchgeführt werden können, um neue Sicherheitsanforderungen zu unterstützen und/oder neue Gefährdungen zu entdecken. Darüber hinaus werden sämtliche Sicherheitstestmittel und -ergebnisse sicher gespeichert, damit sie verfügbar sind, wenn sie für spätere Sicherheitstests benötigt werden.**	• **Sicherstellen, dass alle geplanten Sicherheitstests durchgeführt wurden** • **Ermitteln, ob Sicherheitstest-Arbeitsergebnisse (Berichte) geliefert wurden** • **Testergebnisse, Testdaten und andere sensible Informationen an sicheren Orten archivieren** • **Sicherheitstestergebnisse analysieren, um die System- und Anwendungsentwicklung in Bezug auf Sicherheit zu optimieren**

Es muss klar sein, dass der ISTQB-Sicherheitstestprozess nicht zwangsläufig sequenziell sein muss. Der Sicherheitstestprozess muss auf den Softwareentwicklungslebenszyklusprozess des Unternehmens abgestimmt sein. Wichtig bei dem in diesem Abschnitt beschriebenen Prozess ist, dass die Sicherheitstest-Aktivitäten parallel zu anderen Projektphasenaktivitäten und -tests durchgeführt werden.

Zudem sind die in Tabelle 3.1 aufgeführten Sicherheitstestaufgaben nicht als vorgeschriebene Anforderungen für Sicherheitstestaufgaben, sondern als Beispiele gedacht. Die konkreten Sicherheitstestaufgaben für ein Unternehmen hängen von der Sicherheitsteststrategie und -vorgehensweise ab, für die sich das Unternehmen entschieden hat (siehe auch Abb. 3.1).

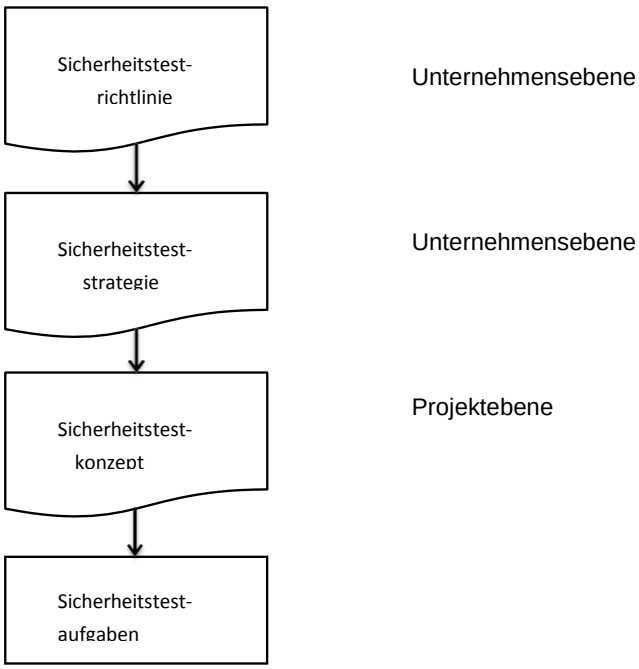

Abb. 3.1: Hierarchie der Sicherheitstestplanung

3.1.2 Ausrichtung des Sicherheitstestprozesses an einem bestimmten Anwendungsentwicklungslebenszyklusmodell

Jeder der folgenden Typen von Lebenszyklusprozessen hat eigene Sicherheitstestbelange. Die Sicherheitstests müssen unbedingt auf den Lebenszyklus abgestimmt werden.

Sequenzielle Lebenszyklen

In diesen Projekten muss der Sicherheitstester Folgendes beachten:

1. Sicherheitserfordernisse und -risiken werden früh im Projekt definiert und sollten in Software-Lastenheften dokumentiert werden.
2. Sicherheitserfordernisse können sich während des Projekts ändern, was sich in den aktualisierten Softwareanforderungen aber u.U. nicht widerspiegelt.
3. Sicherheitstests können sehr spezifisch und umfassend wirken, sind aber aufgrund von Spätrisiken im Projekt u.U. nicht umfassend oder aktuell.
4. Sicherheitstests können jederzeit durchgeführt werden. Üblicherweise werden sie jedoch spät im Projekt durchgeführt.
5. Es kann schwierig sein, den Ergebnissen von Sicherheitstests am Ende eines Projekts mit sequenziellem Lebenszyklus noch Rechnung zu tragen.

Iterative/inkrementelle Lebenszyklen

Inkrementelle Projekte liefern kleine und häufige Releases einer Anwendung. Agile Methoden sind ein Beispiel für dieses Konzept. In diesen Projekten muss der Tester Folgendes beachten:

1. Sicherheitserfordernisse und -risiken treten während des gesamten Projekts auf (normalerweise im Kontext einer Iteration oder eines Sprints) und können in Lastenheften, User-Storys, Modellen, Akzeptanzkriterien und/oder Prototypen definiert werden.
2. Sicherheitserfordernisse und -risiken können sich während des Projekts ändern und können (sollten) in der Iteration, in der sie gefunden wurden, adressiert werden.
3. Sicherheitstests können während des gesamten Projekts permanent durchgeführt werden.
4. Je nach Art des Sicherheitsrisikos lässt sich das Risiko u.U. nicht im Verlauf eines kurzen Release-Zyklus komplett beseitigen und testen.

Standardsoftware (COTS)

Standardsoftware ist ihrem Wesen nach meist eine Black-Box und kann – muss aber nicht – individualisiert sein. Häufig enthält Standardsoftware viele Sicherheitsschwachstellen, sodass häufige Sicherheitsupdates und -patches nötig sind. Auf den Programmcode kann nicht zugegriffen werden. Eine strukturelle Analyse und strukturelles Testen sind daher nicht möglich.

Open-Source-Software

Open-Source-Software eine Variante von Standardsoftware, jedoch mit einem wichtigen Unterschied: Der Programmcode ist für jedermann einsehbar. Diese Produkte haben ebenfalls Sicherheitsschwachstellen. Daher ist es enorm wichtig, dass die Software durch Sicherheitsupdates auf dem neuesten Stand gehalten wird. Sobald eine Sicherheitsschwachstelle für eine Softwareversion öffentlich gemacht wurde, sind die Benutzer dieser Version (und älterer Versionen) der Software dem Risiko eines Angriffs besonders ausgesetzt.

Beispiel: Sicherheitstestprozess in einem sequenziellen Lebenszyklus

Zu beachten ist, dass Sicherheitstests nicht auf eine Phase oder Aktivität in einem Projekt beschränkt werden dürfen. Unbedingt zu vermeiden ist die Situation, dass erst in der Abnahmephase des Projekts Sicherheitstests (und andere Tests) durchgeführt werden. Am Ende des Projekts ist es besonders teuer und riskant, entdeckte Fehler beheben zu müssen. Nachstehend sind die Sicherheitstestaufgaben aufgeführt, die in jeder Phase des sequenziellen Lebenszyklus erledigt werden sollten:

1. **Anforderungen:** Sicherheitsanforderungen werden im Rahmen der Formulierung sämtlicher Anforderungen gemäß den Erfordernissen des Unternehmens definiert. In dieser Phase können auch die Anwendungsfälle geschrieben werden. An dieser Stelle muss die Sicherheitstestvorgehensweise entwickelt werden.

2. **Analyse und Entwurf:** Normalerweise schaut sich ein Businessanalyst die erste Fassung der Anforderungen an und verfeinert diese, um Lücken zu füllen. Ein Systemanalyst und/oder Systemarchitekt analysiert die Anforderungen dann und schlägt vor, wie sich auf optimale Weise eine Lösung liefern lässt, die den Erfordernissen des Kunden genügt. Sicherheit wäre in diesem Fall – neben anderen wie Benutzerbarkeit und Effizienz –eine der funktionalen oder nicht-funktionalen Anforderungen an ein System. Verfasser der Sicherheitstests können zu diesem Zeitpunkt eine Vorstellung – sowohl aus struktureller wie aus funktionaler Sicherheitssicht – von der Architektur und der zu testenden Aspekte des Systems bekommen. Zu diesem Zeitpunkt sollten wichtige Sicherheitstestziele definiert werden.

3. **Detaillierter Entwurf:** An diesem Punkt werden Benutzeroberflächen und Datenbanken entwickelt. Funktionsregeln werden verfeinert und der Sicherheitstestentwurf wird detaillierter. Die ersten Sicherheitstests können auf der Basis von Modellen erfolgen.

4. **Kodierung/Implementierung:** Das ist der Punkt, an dem die Designspezifikationen als Programmcode implementiert werden. Das ist die erste Möglichkeit, die Struktur der Anwendung zu testen. Das schließt das Prüfen auf Sicherheitsschwachstellen wie Pufferüberläufe und unzureichender Eingabevalidierung von Feldeingaben ein, über die eine Code-Injection möglich wäre.

5. Statische Analyse und Code-Reviews sind in dieser Phase äußerst hilfreich und sollten das Prüfen des Programmcodes aus der Sicherheitsperspektive einschließen. Das Testen von Komponenten ist auch eine zentrale Aktivität für die Überprüfung, ob der Programmcode wie vorgegeben funktioniert. Integrationstests zwischen Komponenten können beginnen, sobald Komponenten, die Schnittstellen miteinander haben, für das Testen in kleinen Gruppen verfügbar werden.

6. **Systemtests:** Das ist das Testen von Systemen und Subsystemen. Der Systemtest schließt Software, Hardware, Daten, Prozeduren und die Interaktion der Benutzer mit dem System ein. Diese Tests sind häufig transaktional, weil Geschäftsprozesse getestet werden. Die Grundlage für die Systemtests können Anforderungen, Designmodelle, Anwendungsfälle und andere Spezifikationen sein, die die Systemperspektive vermitteln. Darüber hinaus müssen u.U. Systemintegrationstests durchgeführt werden,

um zu testen, wie verschiedene Systeme bzw. Subsysteme kommunizieren und Daten austauschen. Sicherheitstests haben in dieser Phase eine weiter gefasste Perspektive, weil Hardware und der Austausch von Daten involviert sind. Die Transaktionssicherheit kann getestet werden. Das schließt Authentifizierung, Datenspeicherung, Firewall-Implementierung sowie prozedurale Sicherheitskontrollmechanismen ein.

7. **Benutzer-Abnahmetests:** Mit diesen Tests wird geprüft, ob ein System die in der Praxis auftretenden Geschäftsprozesse unterstützt und über mehrere Systeme in mehreren Unternehmen hinweg funktioniert. Ziel dieser Phase ist es weniger, Fehler zu finden, sondern vielmehr zu prüfen, ob das System die Erfordernisse der Benutzer unter Praxisbedingungen erfüllt. Dabei soll auch sichergestellt werden, dass die Sicherheitsanforderungen richtig implementiert und erfüllt wurden. In dieser Phase sind die Sicherheitstests schon zum großen Teil erfolgt. Es gibt jedoch noch Möglichkeiten, Sicherheitsszenarien zu testen, die sich auf Geschäftsprozessebene ergeben.

8. **Bereitstellung:** Der Zeitpunkt, zu dem das fertige und getestete System für die Benutzer bereitgestellt wird. Dies kann auf viele Arten erfolgen: Pilot-Bereitstellung für ausgewählte Gruppen oder eine Komplettbereitstellung für alle Benutzer. Eine weitere Möglichkeit ist eine parallele Bereitstellung, bei der das alte und das neue System für begrenzte Zeit gleichzeitig laufen. Die Entscheidung, ob die Bereitstellung als abrupter Wechsel (direct cut-over) erfolgen kann, hängt davon ab, welches Risiko damit einhergeht und ob die Abnahmetests zufriedenstellend verlaufen sind. Sicherheit ist auch bei der Systembereitstellung ein Thema, weil alle Systembestandteile so ausgeliefert und konfiguriert werden müssen, dass keine neuen Schwachstellen entstehen. Das kann unter anderem passieren, wenn die Sicherheitskonfigurationen in der Zielumgebung falsch sind. Das wäre z. B. dann der Fall, wenn die Zugriffsrechte für Datenbanken in der Live-Umgebung falsch konfiguriert sind.

9. **Wartung:** Wenn nach der Bereitstellung neue Erfordernisse entstehen oder Fehler entdeckt werden, werden Wartungsmaßnahmen durchgeführt. Das Testen hat jetzt eine andere Dimension, weil sein Fokus auf dem Testen von Änderungen und der Durchführung von Regressionstests liegt. Sicherheitstests sollten auch erfolgen, um sicherzustellen, dass mit den Änderungen keine neuen Schwachstellen Einzug halten. Bestandteil des Wartungsprozesses ist es unter anderem, Firewalls und sonstige Sicherheitstechnik auf dem neuesten Stand zu halten. Durch kontinuierliche Überwachung des Systems lassen sich verdächtigte Aktivitäten entdecken, auf die u.U. sofort reagiert werden muss. Darüber hinaus ist davon auszugehen, dass kontinuierlich neue Schwachstellen bekannt und neue Angriffstechniken entwickelt werden. Die sich dadurch kontinuierlich verändernde Risikolandschaft muss im Rahmen der Wartung durch systematische Penetrationstest adressiert werden.

Beispiel: Sicherheitstestprozess in einem iterativen/inkrementellen Lebenszyklus

In den vergangenen 20 Jahren wurde eine Vielzahl von Methoden eingeführt, mit denen sich die Entwicklung von Software in kleineren Schritten (Inkrementen) oder Iterationen definieren lässt. Im vorliegenden Beispiel werden alle vier Wochen Releases der Software ausgeliefert. Die Grundlage der Arbeit (und des Testens) bilden User-Storys – jeweils mit definierten Abnahmekriterien.

Die Entscheidung, welche Features entwickelt und ausgeliefert werden, wird anhand eines priorisierten Arbeitsrückstands getroffen. Die ausgewählten Features sollten die Elemente widerspiegeln, die den größten Wert liefern und innerhalb des Sprint-Zeitrahmens realisierbar sind. Die passenden Sicherheitsanforderungen erarbeitet der Sicherheitstester in Zusammenarbeit mit dem Geschäfts- und/oder Produktverantwortlichen.

In diesem Beispiel werden vier wichtige Sicherheitsfunktionen für die erste Iteration ausgewählt, weil sie für die Entwicklung vieler weiterer Features benötigt werden. Dabei handelt es sich um folgende Features:

1. Benutzeranmeldung
2. Nutzbarkeit von SSL (Secure Socket Layer)
3. Zurücksetzen vergessener Passwörter
4. Sperrung eines Kontos nach drei erfolglosen Anmeldeversuchen

Jedes dieser Features wird als User-Story geschrieben und in detailliertere Anforderungen aufgeschlüsselt – jeweils mit Abnahmekriterien.

Aus Perspektive der Sicherheitstests arbeitet der Sicherheitstester mit dem Entwickler zusammen, um sicherzustellen, dass sich die richtigen Sicherheitsrichtlinien und Sicherheitsprotokolle im Programmcode widerspiegeln. Zudem arbeitet der Sicherheitstester parallel zum Entwickler, um Funktionen während ihrer Entwicklung zu testen.

In diesem Beispiel könnte das erste Release nur in der Anmeldeseite und den zugehörigen Funktionen für die Anmeldung bestehen: Zurücksetzen vergessener Passwörter und Kontosperrung bei Eingabe falscher Passwörter. In der nächsten Iteration werden weitere Funktionen entwickelt – basierend auf der Priorität für die Stakeholder. In jeder Iteration testet der Sicherheitstester, dass die Sicherheitsmechanismen richtig funktionieren und keine sicherheitsrelevanten Schwachstellen erzeugt wurden. Die Iterationen setzen sich fort, bis alle Aufgaben aus dem Backlog abgearbeitet wurde.

In beiden Beispielen (iterativ/inkrementell und sequenziell) können die Sicherheitstest-Prozessschritte als integrale Aufgaben zur Gewährleistung einer sicheren Anwendung gesehen werden.

3.2 Planung von Sicherheitstests

3.2.1 Ziele der Sicherheitstestplanung

Bei Sicherheitstests sollten im Allgemeinen zwei Aspekte im Mittelpunkt stehen:

1. Überprüfen, ob die entworfenen Sicherheitsvorkehrungen implementiert wurden und wie vorgesehen funktionieren
2. Sicherstellen, dass bei der Entwicklung der Anwendung keine Schwachstellen erzeugt wurden

Wie eingangs in diesem Lehrplan erwähnt, sollten alle zu implementierenden Sicherheitsvorkehrungen auf einer Risikoanalyse basieren. Diese dient als Ausgangspunkt für die Planung der Sicherheitstests für ein Projekt.

Viele unbeabsichtigte Schwachstellen lassen sich durch Maßnahmen der Qualitätssicherung und Rückgriff auf Best Practices bei Konzeption der Architektur, im Entwurf und während der Programmierung vermeiden. Das Testen auf in der Entwicklung erzeugte Schwachstellen beginnt mit einer Bewertung der vom Entwicklungsteam genutzten Praktiken. Auf der Basis des Ergebnisses müssen u.U. zusätzliche Sicherheitstests ausgewählt und durchgeführt werden.

3.2.2 Schlüsselelemente der Sicherheitstestvorgehensweise

Nachstehend sind die Schlüsselelemente einer Sicherheitstestvorgehensweise aufgeführt. Jedes dieser Elemente lässt sich durch Stellen der angegebenen Fragen für ein gegebenes Projekt ermitteln.

1. Ermitteln des Geltungsbereichs der Sicherheitstests
2. Was liegt inner- und außerhalb des Geltungsbereichs?
3. Was ist angesichts der gegebenen Projektressourcen, Sicherheitsrisiken und zeitlichen Beschränkungen erreichbar?
4. Ermitteln, wer die Sicherheitstests durchführen soll
5. Gibt es intern Mitarbeiter mit den entsprechenden Kompetenzen?
6. Kann sich das Unternehmen eine Fremdvergabe der Sicherheitstests vorstellen?
7. Für welche Sicherheitstests ist bei kommerzieller bzw. durch Zulieferer entwickelter Software der Anbieter und für welche der Auftraggeber zuständig?
8. Müssen Sicherheitstester für die Verwendung bestimmter Sicherheitstestwerkzeuge geschult werden?
9. Aufstellung eines geeigneten Zeitplans für die Sicherheitstests unter Berücksichtigung anderer zeitlicher Anforderungen für das Projekt
10. Welche sicherheitsbezogenen Elemente müssen implementiert und getestet werden, bevor andere Tests stattfinden? (z. B. Zugangsrechte und Anmeldungen)
11. Wann sind die Sicherheitsfunktionen für das Testen verfügbar?
12. Wie lange dauert es, die Sicherheitstests mit den geplanten Ressourcen und im geplanten Umfang durchzuführen?
13. Definieren der durchzuführenden Aufgaben und der für sie jeweils benötigten Zeit
14. Wie viel Zeit wird für den Entwurf der entsprechenden Sicherheitstests basierend auf den geplanten Ressourcen und dem geplanten Umfang benötigt?
15. Wie viel Zeit wird gebraucht, um die Ergebnisse der Sicherheitstest auszuwerten und zu dokumentieren?
16. Wie viel Zeit wird gebraucht, um sicherheitsbezogene Regressionstests durchzuführen?
17. Wie viel Zeit wird gebraucht, um die Sicherheitstestumgebung einzurichten?
18. Definieren der Sicherheitstestumgebung(en)
19. Welchen Umfang hat die Umgebung? (Plattform, Technologie, Größe, Ort)
20. Ist es eine neue Umgebung?

21. Welche Sicherheitstestwerkzeuge und andere Testwerkzeuge müssen in der Umgebung installiert werden?
22. Einholung von Autorisierung und Genehmigungen für Sicherheitstestaktivitäten
23. Wer muss die Sicherheitstests autorisieren und genehmigen?
24. Wann wird diese Autorisierung benötigt?
25. Sind Budget und Finanzierung ausreichend?

Wie jedes Projektarbeitsergebnis muss auch die Sicherheitstestvorgehensweise auf Vollständigkeit und Richtigkeit geprüft werden. Weil Sicherheitstests oft technischer Natur sind, ist eine technische Review-Sitzung u.U. die geeignetste Methode. Aber auch Walkthroughs und Inspektionen können geeignet sein.

Eine Standard-Checkliste kann die Basis dessen bilden, was in einer Review-Sitzung abzuarbeiten ist. Wie bei jedem anderen Review muss das Feedback konstruktiv sein und darf nicht auf den Entwickler der Sicherheitstestvorgehensweise abzielen. Dem Review-Team sollten sachkundige Personen aus allen Bereichen angehören, die von den in der Sicherheitstestvorgehensweise besprochenen Sicherheitsaspekten betroffen sind. Die Mitglieder des Review-Teams müssen nicht zwangsläufig Sicherheitstester sein oder über Fachwissen aus diesem Bereich verfügen. So kann z. B. der Leiter eines Geschäftsbereichs Informationen über Sicherheitsrisiken haben, die in der Sicherheitstestvorgehensweise erfasst werden müssen. IT-Auditoren und Sicherheitsadministratoren sind bei Reviews der Sicherheitstestvorgehensweise aufgrund ihres Wissens über Sicherheitsrichtlinien und -verfahren besonders hilfreich.

3.3 Entwurf von Sicherheitstests

Es gibt verschiedene Wege, mit dem Entwerfen von Sicherheitstests zu beginnen:

1. auf der Basis einer durchgeführten Risikoanalyse
2. auf der Basis eines verfügbaren Gefährdungsmodells
3. auf der Basis einer Ad-hoc-Quellenkategorisierung der Sicherheitsrisiken (siehe [ISTQB_ATTA_SYL]).

All dies können brauchbare Ausgangspunkte sein.

Je nach Art des Projekts muss gewährleistet werden, dass es in jeder relevanten Entwicklungsphase Sicherheitstests gibt.

3.3.1 Entwurf von Sicherheitstests

Detaillierte Sicherheitstests basieren auf den Sicherheitsrisiken, einer Sicherheitsteststrategie und anderen Quellen wie z. B. Gefährdungsmodellen. Sicherheitstests können von ihrem Wesen her auch als funktional (Black-Box) oder strukturell (White-Box/Grey-Box) gesehen werden. Beim Testen der Sicherheit einer eCommerce-Website können die funktionalen Sicherheitsrisiken z. B. in SQL-Injection, Account Harvesting und Passwort-Knacken bestehen. Ein Beispiel für ein strukturelles Sicherheitsrisiko ist ein Pufferüberlauf, der es einem Angreifer ermöglicht, sich über ein Speicherleck Zugriff zu verschaffen.

Detaillierte Sicherheitstests haben die folgenden entscheidenden Eigenschaften. Sie sind:

1. Priorisiert nach ermittelten Sicherheitsrisiken und Gefährdungsmodellen
2. Zurückgeführt auf definierte Sicherheitsanforderungen
3. Definiert auf Basis der und in Hinblick auf die Zielgruppe (Entwickler, Funktionstester, Sicherheitstester)
4. Definiert auf Basis von Profilen unter Berücksichtigung pot. Sicherheitsmängel.
5. Ausgelegt auf Automatisierung; sofern geeignet

Der elementare Ablauf des Entwurfs von Sicherheitstests sieht wie folgt aus:

1. Die Sicherheitstestvorgehensweise (Projektebene)
2. Sicherheitstestrisiken, Gefährdungsmodelle und Anforderungen (Projektebene)
3. Techniken zum Entwerfen von Sicherheitstests (basierend auf Risiken, Anforderungen und Anwendung)
4. Sicherheitstestfälle und -szenarien

Im Rest des Kapitels werden neben der zugehörigen Sicherheitstest-Entwurfstechnik allgemeine Sicherheitsrisiken und Schwachstellen vorgestellt. Neue Sicherheitsrisiken und Schwachstellen entstehen so schnell, dass es für Planer von Sicherheitstests ratsam ist, sich im Hinblick auf Sicherheitsstandards und Gefährdungslisten (siehe Kapitel 9) stets auf dem neuesten Stand zu halten.

Als grundlegendes Prinzip gilt, dass ein Sicherheitstest-Entwurfsprozess die Möglichkeit bieten muss, Tests auf der Basis von ermittelten Sicherheitsrisiken, Anforderungen oder Gefährdungen zu entwickeln und zu realisieren.

Funktionale Sicherheitskontrollmechanismen (z. B. Transaktionskontrolle)

Mit diesen Tests soll nachgewiesen werden, dass Kontrollmechanismen vorhanden sind, richtig funktionieren sowie unbefugte Handlungen wirksam erkennen und verhindern.

Beispiel: Ein Bankangestellter darf eine Barabhebung, die einen bestimmten Betrag übersteigt, nicht autorisieren, ohne dass sein Vorgesetzter eine Genehmigung ins System eingibt.

Funktionale Zugangskontrollmechanismen (z. B. Anmeldedaten, Passwörter, Tokens)

Diese Tests fallen den meisten Menschen im Zusammenhang mit Sicherheitstest wohl am ehesten ein. Folgende Tests gibt es:

1. Richtige Anwendung von Benutzernamen- und Passwortrichtlinie
2. Die Zugangskontrollmechanismen und die Zugangskontrollebene sind ausreichend für das Risiko
3. Die Zugangskontrolle widersteht Software zum Knacken von Passwörtern

Beispiel: Account Harvesting ist eine Technik zur Ermittlung eines Benutzernamens. Sobald der Benutzername erraten oder ermittelt wurde, wird nur noch das Passwort benötigt, um Zugang zum System zu erhalten. Ein gängiger Test besteht darin, nachzuweisen, dass bei Eingabe eines richtigen Benutzernamens mit einem falschen Passwort aus der angezeigten Fehlermeldung nicht hervorgeht, welches von beiden falsch ist.

Strukturelle Zugangskontrollmechanismen (z. B. Benutzerzugriffsrechte, Verschlüsselungsgrade, Authentifizierung)

Tests für diese Kontrollmechanismen basieren darauf, welche Benutzerrechte für den Zugriff auf Daten und Funktionen und welche Datenschutzstufen eingerichtet wurden. Strukturelle Zugangskontrollen werden in der Regel von einem Systemadministrator, Sicherheitsadministrator oder Datenbankadministrator eingerichtet. In manchen Anwendungen sind Zugriffsrechte eine Konfigurationsoption. In anderen Fällen greifen Zugriffsrechte auf der Systeminfrastrukturebene.

Tests von strukturellen Zugangskontrollen schließen Folgendes ein: das Erstellen von Testbenutzerkonten für die einzelnen Ebenen des Sicherheitszugangs sowie die Überprüfung, dass die einzelnen Zugangsebenen über keine Zugriffsrechte verfügen, die für die jeweilige Ebene beschränkt sind. Ein Beispiel: Benutzerkonten werden mit minimalem Zugang, Manager-Zugang oder Administratorzugang erstellt. Beim Testen muss nachgewiesen werden, dass ein Benutzer mit minimalem Zugang keine Aktivitäten der Manager-Zugangsebene ausführen kann.

Sichere Programmierpraktiken

Das ist in erster Linie eine statische Testmethode, mit der ermittelt wird, ob Software- und Systementwickler beim Programmieren von Anwendungen etablierte Sicherheitsverfahren einhalten.

Als Grundprinzip gilt dabei: Viele sicherheitsrelevante Angriffe gelingen durch Ausnutzung von Softwarefehlern, die ein unerwartetes Systemverhalten bewirken.

Hier eine kurze Liste mit sicheren Programmierpraktiken:

1. Bewährte Algorithmen und Mechanismen zum Session-Management werden eingesetzt und zur Erzeugung zufälliger Session-IDs verwendet.
2. Autorisierungsentscheidungen werden nur von vertrauenswürdigen Systemkomponenten gefällt, die unter Kontrolle des Unternehmens steht, das Autorisierung bereitstellt (z. B. sollte die Autorisierung serverseitig erfolgen).
3. Sichere Informationen dürfen nicht in Fehlermeldungen erscheinen. Das schließt Systeminformationen, Session-IDs und Zugangsdaten ein.
4. Anwendungsfehler sollten innerhalb der Anwendung gehandhabt werden, statt sich auf die Serverkonfiguration zu verlassen.
5. HTTP GET-Anfragen dürfen keine sensiblen Daten enthalten.
6. Error-Handler dürfen keine Stack-Trace- oder anderen Debugging-Informationen anzeigen.
7. Alle Dateneingabe-Validierungsfehler sind zu protokollieren.
8. Sensible Daten, die temporär auf dem Server gespeichert sein können, müssen geschützt werden (z. B. durch Verschlüsselung). Diese temporären sicheren Daten müssen gelöscht werden, wenn sie nicht mehr gebraucht werden.
9. Eine Anwendung sollte keine direkten Befehle an das Betriebssystem ausgeben können. Stattdessen sind für die Ausführung von Betriebssystemaufgaben integrierte APIs zu nutzen.

10. Passwörter, Verbindungszeichenfolgen bzw. andere sensible Daten dürfen nicht in Klartext auf Client-Rechnern gespeichert werden (z. B. in Cookies). Das Einbetten derartiger Daten in nicht sichere Formate wie Adobe Flash, kompilierten Programmcode und MS-Viewstate-Parameter muss untersagt sein.
11. Sensible Daten dürfen grundsätzlich nur verschlüsselt übertragen werden. Mit TLS (Transport Layer Security) lassen sich Daten bei der Übermittlung über HTTP-Verbindungen schützen. Bei Nicht-HTTP-Verbindungen ist beim Übertragen sensibler Daten mit Verschlüsselung zu arbeiten.
12. Von Benutzern gelieferte Daten dürfen nicht direkt an eine dynamische „include“-Funktion übergeben werden.
13. Alle von Benutzern gelieferten Daten müssen vor der Nutzung durch die Anwendung von potenziell schadhaften Anteilen bereinigt (input sanitization) und validiert werden.
14. Variablen müssen stark typisiert werden in Sprachen, die eine Typprüfung unterstützen. Das heißt, bei den Variablen muss der Eingabetyp definiert sein. So sollte z. B. ein numerisches Feld keine Buchstaben als Eingabe akzeptieren. Diese Einschränkung muss vorzugsweise in der Typdefinition der Variablen sowie in der Datenbank definiert sein. Es ist möglich, in JavaScript (Node JS) oder anderen Sprachen, die keine vom Compiler durchgesetzte Typprüfung unterstützen, sicheren Programmcode zu schreiben.
15. Statt für allgemeine Aufgaben neuen „unmanaged“ Code zu verwenden, ist getesteter, vertrauenswürdiger und genehmigter Programmcode zu nutzen, für den es ein Konfigurationsmanagement gibt.
16. Dienste müssen mit geringstmöglichen Zugriffsrechten ausgeführt werden (niemals unter Root-Rechten), und jeder Dienst sollte im Betriebssystem sein eigenes Benutzerkonto haben.

Eine Liste mit sicheren Programmierpraktiken finden Sie im OWASP Secure Coding Practices Quick Reference Guide [OWASP1] und in Top 10 Secure Coding Practices [CERT1]. Darüber hinaus gibt es von SANS unter [SANS1] eine Liste mit den 25 gefahrenträchtigsten Softwarefehlern.

Mit dynamischen Tests lässt sich ermitteln, ob Praktiken wie Datenvalidierung und Fehlermeldungsausgabe von den Entwicklern eingehalten wurden. Eine der gängigsten Sicherheitsschwachstellen ist zudem der Speicherpufferüberlauf. Er lässt sich mit dynamischen Speichertestwerkzeugen ermitteln.

Zugriff auf das Betriebssystem

Sobald ein Angreifer Zugriff auf das Betriebssystem hat, kann er auf Daten und das Netzwerk zugreifen und Schadsoftware platzieren. Mit Tests, die diese Schwachstelle adressiert, können die Möglichkeiten reduziert werden, Rootkits und anderen bösartigen Programmcode in ein System einzuschleusen.

Schwachstellen von Programmiersprachen (z. B. Java)

Laut den Analysten von WhiteHat Security, einem Anbieter von Sicherheitslösungen für Anwendungen, gab es im Hinblick auf Schwachstellen generell keine erheblichen Unterschiede zwischen den einzelnen Programmiersprachen. [WhiteHat Security, 2014] Im April 2014 gab WhiteHat Security einen Website Security Statistics Report heraus. Dazu hatte man 30.000 Kunden-Websites mit einem hauseigenen Scanner auf Schwachstellen abgeklopft. Die Unterschiede in der relativen Sicherheit von Sprachen wie .NET, Java, PHP, ASP, ColdFusion und Perl waren laut den Ergebnissen vernachlässigbar. Diese sechs Sprachen haben eine relativ ähnliche durchschnittliche Zahl von Schwachstellen und Probleme wie SQL-Injection oder Cross-site Scripting sind weiterhin weit verbreitet. [WhiteHat Security, 2014] Dazu ist festzuhalten, dass sich sicherer Programmcode mit vielen Sprachen schreiben lässt – aber eben auch unsicherer Programmcode. Ungeachtet der verwendeten Sprache ist entscheidend, wie eine Anwendung codiert (implementiert) wird.

Die CERT Division des Software Engineering Institute bietet Publikationen [CERT2] und Werkzeuge [CERT3], die sich mit sprachspezifischen Sicherheitsproblemen befassen. Darüber hinaus finden sich in der Vulnerability Notes Database [CERT4] aktuelle Informationen über Software-Schwachstellen. Dazu zählen Zusammenfassungen, technische Details, Angaben zu Abhilfen und eine Liste mit betroffenen Anbietern. Im deutschsprachigem Raum betreibt das Hasso-Plattner-Institut eine Datenbank mit Schwachstelleninformationen. [HPI1]

Schwachstellen von Plattformen (z. B. Windows, Linux, Mac OS, iOS, Android)

Jede Softwareplattform hat ihre eigenen Schwachstellen. Sicherheitstester müssen dafür sorgen, dass die Sicherheitsupdates für die Plattform sofort nach Erscheinen und auf allen Geräten der betroffenen Plattform installiert werden.

Gefährdungen von außen

Wenn es um Cyber-Angriffe geht, denken die meisten Menschen an Sicherheitsgefahren von außen. Manche dieser Gefährdungen wie die Ausnutzung von Schwachstellen in Anwendungen oder Programmiersprachen lassen sich erkennen, testen und verhindern.

Eine weitere Form dieser Gefährdung sind Dienstblockaden (DoS). Bei diesen Angriffen werden in der Regel die System- oder Anwendungsressourcen so überlastet, dass reguläre Benutzer nicht mehr auf das System oder die Anwendung zugreifen können. DoS-Angriffe lassen sich auf die Netzwerkbandbreite, die Konnektivität eines Systems oder einer Anwendung sowie bestimmte Dienste oder Funktionen ausrichten.

Ein DDoS-Angriff (Distributed Denial of Service) ist eine Form des DoS-Angriffs. In diesem Fall wird er dezentral von vielen Computern mit Netzanbindung geführt. Mögliche Techniken sind die Verstärkung oder der Einsatz von Botnetzen. Ein Botnetz besteht aus einer Vielzahl vorher gekaperter Rechner, die sich unter Kontrolle oder Befehlsgewalt des Angreifers befinden. Der Angreifer erhält die Kontrolle über die Infizierung des Rechners mit Viren oder das Platzieren von Trojanern. Die infizierten Computer können dann genutzt werden, um das Opfer des Angreifers (Netzwerk) mit Datenverkehr zu fluten.

Bei Verstärkungs- oder Reflection-Angriffen nutzt der Angreifer eine Schwachstelle (oder sogar eine reguläre Funktion) in bestimmten Protokollen (z. B. DNS oder NTP). Er sendet große Datenverkehrsmengen an IP-Broadcast-Adressen (mehrere Hosts), die die vorgetäuschte Quelladresse des Opfersystems enthalten. Das hat zur Folge, dass der Broadcast-Service seinen Verkehr als Echo in Richtung der Adresse des Opfers schickt und damit das ursprüngliche Verkehrsaufkommen als Funktion der Anzahl der Hosts vervielfacht. Wenn der Angreifer diese Anfragen mehrmals pro Sekunde schickt, sieht sich das Opfer mit einer hohen Anzahl zu sendender Antworten konfrontiert.

Beispiel: Angreifer A schickt getarnt als Opfer C eine Anfrage nach einer vollständigen Liste aller bekannten DNS-Datensätze an System B – häufig mit vorgetäuschter IP-Adresse. System B schickt daraufhin die vollständige Liste an Opfer C. Das überschwemmt den Server von Opfer C mit einer riesigen Datenmenge.

Eine weitere Form von DoS-Angriffen sind Angriffe die die Ressourcen eines Systems gezielt überlasten. Dabei wird i.d.R. eine reguläre Funktion dahingehend missbraucht, sodass die für die Bereitstellung der Funktion benötigten Verarbeitungsressourcen (Prozessor, Arbeitsspeicher, Festplattenspeicher usw.) so stark belastet werde, dass entweder die Funktion oder das gesamte System nicht mehr oder nur noch eingeschränkt benutzbar sind.

Beispiel: Eine der Funktionen im SSL-Protokoll ist die Option, in einer bestehenden Session neue Schlüssel zu erzeugen, wenn der Client oder Server vermutet, dass die Session kompromittiert ist. Das Erzeugen von Schlüsseln ist ein ressourcenintensiver Vorgang. Wenn ein Angreifer mehrmals pro Sekunde eine Anfrage auf Erzeugung neuer Schlüssel schickt, kann ein schlecht konfiguriertes oder ungeschütztes System in einen Zustand geraten, indem es nur noch neue Schlüssel erzeugt und keine Ressourcen für andere Aufgaben mehr verfügbar hat.

Und zuletzt gibt es noch so genannte logische DoS-Angriffe. Bei ihnen nutzt ein Angreifer vorgesehene Funktionen aus, um andere Benutzer daran zu hindern, auf das System zuzugreifen.

Beispiel: Eine Anwendung nutzt vorhersagbare Benutzernamen und sperrt einen Benutzer nach drei fehlgeschlagenen Anmeldeversuchen dauerhaft. Ein Angreifer kann die Benutzernamen erraten, viele Konten im System sperren lassen und damit den Zugriff dieser Benutzer auf das System verhindern (und damit indirekt einen DoS-Angriff auf den Helpdesk durchführen).

Tests auf DDoS-Schwachstellen gibt es in vier Stufen.

1. Test zur Gewährleistung, dass die Computer nicht mit bekannter Schadsoftware infiziert sind
2. Testen der Fähigkeit der Angriffserkennungssysteme auf schnelle Erkennung vieler Anfragen von einem Computer innerhalb kurzer Zeit
3. Ermitteln von Konfigurationen, die den Missbrauch von Funktionen durch einen Angreifer ermöglichen (wie z. B. SSL, Webserver, DNS)
4. Ermitteln von Logikfehlern, die DoS-Angriffe ermöglichen

Intrusionen sind eine weitere Form des Angriffs von außen. Für das Eindringen von außen in ein System gibt es viele Wege. Die Gemeinsamkeit dieser Angriffe ist der „Einbruch“ in ein System zum Erhalt oder zur Manipulation von Informationen. Einige der Methoden sind in der nachstehenden Liste aufgeführt:

1. Social Engineering
2. Injection-Angriffe (SQL, bösartiger Programmcode)
3. Kompromitierte Konten (Harvesting, Passwort-Rücksetzung)
4. Ausnutzung bekannter Schwachstellen (Firewall, Betriebssystem, Framework, Anwendung)
5. Angriffe durch Malware (Schadprogramme)
6. Angriffe durch unsichere Konfiguration
7. Mängel in der Autorisierung
8. Angriffe auf die Anwendungslogik (Ausnutzung von Fehlerzuständen in Anwendungen, vor allem in webbasierten Anwendungen, um die Funktionalität zu missbrauchen – z. B. das Ausführen von Schritten in einer eCommerce-Shopping-Anwendung in falscher Reihenfolge, um einen Rabatt oder ein Guthaben zu erhalten)

Das Abfangen von Daten, die über das Netz von einem Unternehmen zu einem anderen übertragen werden, gilt nicht als Intrusion.

Interne Gefährdungen

Die größten Gefährdungen können von innen kommen. Für interne Angriffe sind die folgenden Quellen zu berücksichtigen:

1. Wirtschaftsspionage – ein Mitarbeiter, dem man vertraut, verkauft Unternehmensdaten, darunter Kontodaten von Kunden, Geschäftsgeheimnisse, Zugangsdaten von Mitarbeitern usw.
2. Informationen, die über beauftragte externe Entwickler, Tester und sonstiges Personal nach außen dringen (z. B. Kundendienstvertreter). Wenn Mitarbeiter beim Outsourcing-Unternehmen kündigen, können sie wertvolle Daten mitnehmen.

3. Diebstahl von Festplatten und anderen physischen Datenspeichern
4. Frustrierte Mitarbeiter, die dem Unternehmen schaden wollen, indem sie vertrauliche Informationen nach außen dringen lassen, oder Diebstahl begehen, indem Sie sich unter dem Deckmantel gefälschter Rechnungen Geld an sich selbst auszahlen

Format und Aufbau von Sicherheitstests

Jedes Unternehmen, das Sicherheitstests durchführt, hat seine eigene Methode, ausführliche Tests zu strukturieren. Häufig kann beim Entwurf von Sicherheitstests dasselbe Format wie bei anderen Arten von Tests genutzt werden. Der einzige Unterschied besteht im Gegenstand des Tests und der Testumgebung.

Selbst in Anlehnung an Normen wie IEEE 829-2008 und ISO 29119 [ISO/IEC/IEEE 29119-3] sollten Anpassungen an die Erfordernisse des Unternehmens vorgenommen werden. Diese Normen geben jedoch Grundregeln für den Inhalt der verschiedenen Testplanungsunterlagen vor. In vielen Fällen können Testfälle und Testverfahren (Skripte) in einem Testmanagementwerkzeug definiert und implementiert werden. Dieses Werkzeug gibt die Struktur häufig vor.

Testfälle sind die eigenständigste Form der Testbeschreibung. Sie erfordern keine sequenzielle Ausführung. Wenn für das Erreichen eines bestimmten Testziels die sequenzielle Ausführung nötig ist, werden Testfälle in einem Testverfahren oder -skript in einer bestimmten Abfolge kombiniert. Testfälle werden in der Regel für das Testen einzelner Testbedingungen genutzt. Bei Sicherheitstests kann das Testen der Anmeldefunktion z. B. aus Testfällen bestehen, mit denen validiert wird, dass die Anforderungen für das Passwortformat richtig durchgesetzt werden.

Während der Testimplementierung werden die Testfälle als Testabläufe entwickelt, priorisiert und organisiert. Ein Testablauf gibt die Ausführungsreihenfolge der Testfälle vor. Wenn Tests mit Hilfe eines Testausführungswerkzeugs durchgeführt werden, ist die Abfolge der Aktionen in einem Testskript (ein automatisierter Testablauf) festgelegt. Testabläufe kommen zum Einsatz, wenn die Abfolge wichtig ist. Ein Testablauf ist z. B. dann hilfreich, wenn der Vorgang „Verlorenes Passwort wiederherstellen“ getestet wird.

Wenn erfahrungsgestützte Tests wie explorative Tests nötig sind, werden die Testbedingungen und erwarteten Ergebnisse nicht vor dem Test definiert. Vielmehr müssen die getesteten Testbedingungen und tatsächlichen Ergebnisse vom Sicherheitstester für den Bericht erfasst werden.

3.3.2 Entwurf von Sicherheitstests gestützt auf Richtlinien und Verfahren

Beim Entwurf von Tests zur Validierung von Sicherheitsrichtlinien und -verfahren werden diese zur Grundlage der Tests. Aus dieser Perspektive sind Sicherheitstests fast ein Mittel des Sicherheits-Auditings.

Sicherheitsrichtlinien und -verfahren dürfen nicht die alleinige Basis der Tests sein, weil auch andere Perspektiven des Testens der Sicherheit gebraucht werden. Das Entwerfen von Tests zur Validierung von Sicherheitsrichtlinien und -verfahren hat folgende Ziele:

1. Verstehen des Zwecks und Geltungsbereichs der Richtlinie bzw. des Verfahrens
2. Bewertung der Testbarkeit der Richtlinie bzw. des Verfahrens
3. Entwicklung von Tests mit direktem Bezug zur Richtlinie bzw. zum Verfahren

Das folgende Beispiel soll das veranschaulichen. Ein Verfahren ist wie folgt definiert: *„In allen IT-Systemen von XYZ ist die Zahl der erfolglosen Anmeldeversuche auf drei beschränkt. Nach drei erfolglosen Anmeldeversuchen erfolgt eine zeitlich festgelegte Sperre. Ohne die entsprechenden lokalen Benutzerkontodaten ist kein Zugriff auf unsere IT-Systeme möglich. Wer ihn dennoch benötigt, muss sich zur Bestätigung der Identität und zum Erhalt eines temporären Passworts an unseren IT-Support wenden.“*

Das ist ein gut testbares Verfahren. Der Test umfasst folgende Schritte:

1. Drei erfolglose Anmeldeversuche bei einer Anwendung. Beim dritten erfolglosen Versuch müsste eine Sperrmeldung angezeigt werden. Bei jedem weiteren Anmeldeversuch beim Konto wird die Sperrmeldung angezeigt.
2. An den IT-Support wenden und Identität bestätigen. An eine bekannte E-Mail-Adresse wird ein temporäres Passwort geschickt.
3. Mit diesem temporären Passwort anmelden. Es müsste Zugriff gewährt werden.
4. Ein neues Passwort erstellen, das der Passwortrichtlinie entspricht. Das neue Passwort müsste akzeptiert werden.
5. Abmelden.
6. Mit dem neu erstellten Passwort anmelden. Es müsste Zugriff gewährt werden.

Beachten Sie, dass Schritt 4 auch die Möglichkeit bietet, die Passwortrichtlinie zu testen.

Nicht alle Sicherheitsrichtlinien sind so gut testbar. Ein Beispiel: „*Der Inhalt der Audit-Datensätze der XYZ GmbH enthält alle Audit-Events mit Datum-/Zeitstempel und ist zurückführbar auf konkrete Einzelpersonen. Herstellerspezifische Protokolle, die ausreichende Informationen zur Erfüllung dieser Anforderungen bieten, müssen als angemessen für Audit-Zwecke gelten.*“

Das ist zwar testbar, jedoch muss ein Test definiert und durchgeführt werden, bei dem alle relevanten Audit Ereignisse erfasst werden. Zu diesem Zweck kann ein Mustersatz mit den Ereignissen erzeugt werden, die in den Audit-Datensätzen erfasst werden sollen. Darüber hinaus muss die Genauigkeit der protokollierten Informationen, z. B. die Benutzer-ID sowie die Datums-/Zeitstempel, als richtig verifiziert werden.

3.4 Ausführung von Sicherheitstests

3.4.1 Schlüsselelemente und Merkmale einer effektiven Sicherheitstestumgebung

Bei vielen Arten von Tests kann eine Testumgebung auf demselben Server und im selben Netzwerk mit anderen Systemen genutzt werden. Mit Sicherheitstests gehen jedoch besondere Risiken einher, die eine gesonderte Vorgehensweise zur Schaffung der Testumgebung erfordern. Das gilt vor allem dann, wenn nicht vertrauenswürdige Anwendungen getestet werden (z. B. von einem Dritt- oder Open-Source-Anbieter).

Manche Sicherheitstests, wie das Testen von funktionalen Kontrollmechanismen und des Session-Managements, lassen sich ohne großes Risiko in einer typischen integrierten Testumgebung durchführen. Beim Testen von unbekanntem und nicht vertrauenswürdigem Programmcode besteht jedoch die Möglichkeit, dass Schadsoftware einen Server und/oder ein Netzwerk beschädigt. Daher ist es ratsam, den Test in einer isolierten oder virtuellen Testumgebung vorzunehmen.

Eine Umgebung für Sicherheitstests hat folgende Hauptmerkmale:

1. Isoliert – von anderen Systemen (je nach Risikograd der Schadsoftware)
2. Vollständig – die Gesamtumgebung muss die Zielumgebung (Produktionsumgebung) hinsichtlich folgender Aspekte abbilden:

3. Systeme und Anwendungen, die getestet werden
4. Betriebssysteme (exakte Version und Konfiguration)
5. Netzwerk-Anbindungen
6. Middleware
7. Desktop-Rechner (Hardwaremarke, Prozessor, Arbeitsspeicher)
8. Mobile Geräte (Hersteller, Prozessor, Arbeitsspeicher, Energieverwaltung)
9. Datenbanken
10. Zugriffsrechte
11. Browser und Plug-Ins
12. Koexistierende Anwendungen
13. Daten (erzeugte Testdaten oder Produktionsdaten, die maskiert wurden)
14. Reproduzierbar – zur Wiederholung von Tests und zur Wiederherstellung bei evtl. Beschädigung

3.4.2 Die Bedeutung von Planung und Genehmigungen für Sicherheitstests

Dafür, dass ein Sicherheitstester vor der Durchführung von Sicherheitstest eine Genehmigung einholen muss, gibt es mehrere Gründe:

1. In fast allen Ländern ist bereits der Versuch gesetzlich verboten, sich Zugriff auf Datensysteme und ihre Informationen zu verschaffen. In einigen Ländern verstößt es sogar gegen das Gesetz, Zugang zu Sicherheitstestwerkzeugen zu haben. Das heißt, dass Sie bei der ungenehmigten Durchführung der meisten Sicherheitstestaktivitäten Gefahr laufen gegen mindestens ein Gesetz zu verstoßen. Die Tests sollten erst durchgeführt werden, wenn vorher eine Verzichtserklärung des Besitzers des Systems und/oder der Daten vorliegt sowie eine Genehmigung der zuständigen Geschäftsführung eingeholt worden ist.
2. Sicherheitstests können Angriffserkennungsalarme auslösen und der Tester kann wie ein interner Angreifer mit bösen Absichten erscheinen. Penetrationstests sind ein Sonderfall, bei dem die Genehmigung besonders wichtig ist.
3. Sicherheitstests können zu größeren Systemausfällen und Blackouts führen. Das Risiko muss bekannt sein und mögliche Vorkehrungen sollten getroffen worden sein.

Ohne vorherige und gesonderte Genehmigung für Sicherheitstests kann ein Tester darüber hinaus gegen Sicherheitsrichtlinien und -verfahren verstoßen. Damit macht sich der Tester u.U. zum Gegenstand eines Kündigungsprozesses oder eines strafrechtlichen Verfahrens.

Ein Genehmigungsformular für Sicherheitstests sollte folgende Angaben enthalten:

1. Name der genehmigenden Stelle
2. Namen des Testpersonals und/oder der Prüfstelle
3. Leistungsbeschreibung
4. Gültigkeitszeitraum der Genehmigung (von/bis)
5. sonstige relevante Details wie die IP-Quelladressen, Benutzerkonten usw.
6. Bestätigungen:
7. Dem Kunden gehört das zu testende System.
8. Der Kunde hat die Vollmacht, die Sicherheitstests zu genehmigen.
9. Der Kunde hat ein Backup sämtlicher Systeme und Daten vorgenommen.
10. Der Kunde hat sich davon überzeugt, dass sich das System bei Bedarf aus den Backups wiederherstellen lässt.
11. Der Kunde kennt die Risiken, die mit den Sicherheitstests einhergehen.
12. Eine Haftungsfreistellungsklausel für die Prüfstelle.
13. Unterschriften eines Kundenvertreters, der bevollmächtigt ist, derartige Vereinbarungen einzugehen.

Ein Musterformular finden Sie unter [OWASP3].

Da in Deutschland besonders restriktive Regulierungen zum Umgang mit personenbezogenen Daten bestehen, sollte vor einen Sicherheitstest geklärt werden, ob durch den Sicherheitstest personenbezogene Daten betroffen sind. Ist diese der Fall, so muss der Datenschutzbeauftragte und gegebenenfalls auch die Personalvertretung der beauftragenden Institution vor den Tests einbezogen werden. Es sollte geklärt werden, in welcher Form personenbezogene Daten betroffen sind und vereinbart werden, wie diese Daten ggfs. anonymisiert werden können und wie die Daten nach Beendigung der Sicherheitstests gelöscht werden müssen. [BSIPT].

3.5 Bewertung von Sicherheitstests

Wie der Großteil der Testarbeit erfolgt die Bewertung der Sicherheitstests während der Durchführung der einzelnen Tests. Die Testbewertung ist die Bewertung des Ergebnisses eines Sicherheitstests. Wenn Sicherheitsmängel (Schwachstellen) festgestellt werden, muss ein Fehler- und Abweichungsbericht verfasst werden, in dem mindestens Folgendes aufgeführt ist:

1. Name des Testers, der die Schwachstelle gefunden hat
2. Testumgebung, in der die Schwachstelle gefunden wurde
3. Durchgeführte Testschritte (um die Reproduktion der Testergebnisse zu ermöglichen)
4. Art der Sicherheitsschwachstelle
5. Ausmaß der Sicherheitsschwachstelle
6. Potenzielles Schadensausmaß der Sicherheitsschwachstelle

7. Empfohlene Maßnahmen zur Abhilfe

Fehler- und Abweichungsberichte für Sicherheitstests lassen sich mit demselben Fehler- und Abweichungsmanagementsystem wie bei anderen Testarten erfassen. Sicherheitstestberichte müssen eine besondere Kategorie zugewiesen bekommen und geschützt werden, um zu verhindern, dass Unbefugte Zugriff auf sie haben. Solche Situationen können eintreten, wenn:

1. Sicherheitstests von einer unabhängigen Stelle durchgeführt werden und Fehler/Abweichungen in einem Werkzeug erfasst wurden, in dem die Anzeige der Fehler- und Abweichungsberichte nur geringen Einschränkungen unterliegt
2. Sicherheitsschwachstellen zwar ermittelt, aber nicht sofort behoben werden
3. Interne Mitarbeiter als potenzielle Gefährdung gesehen werden, weil sie die Sicherheitsschwachstellen ausnutzen könnten

Der für die IT-Sicherheit Verantwortliche (IT-Sicherheitsverantwortliche) muss entscheiden können, ob der Zugang zu den Sicherheitstestergebnissen beschränkt werden muss oder nicht.

Bei Abschluss einer großen Sicherheitstestreihe – z. B. dem Test eines Systems – kann ein Abschlussbericht ausgestellt werden. Dieser Bericht muss je nach Fortschritt bei der Beseitigung der gefundenen Schwachstellen u.U. ebenfalls vertraulich behandelt werden.

3.6 Wartung von Sicherheitstests

In vielen Fällen besteht die Modifizierung des Sicherheitstestprozesses u.U. nur im Hinzufügen neuer Arten von Tests in Reaktion auf neue Arten von Gefährdungen. Eines ist jedoch sicher: Die Ziele von Sicherheitstests und die Gefährdungen ändern sich täglich. Deshalb muss der Sicherheitstestprozess so ausgelegt sein, dass er leicht verändert werden kann.

Außerdem tauchen auf dem Markt neue Werkzeuge auf, die bei der Durchführung der Sicherheitstests hilfreich sein können. Sicherheitstester müssen über diese Entwicklungen informiert sein und beurteilen, welche Werkzeuge die Sicherheitstests leistungsfähiger und flexibler machen.

4 Sicherheitstesten im gesamten Softwarelebenszyklus – 225 min

Schlüsselbegriffe

Missbrauchsfall, Fuzz-Testen

Lernziele für das Thema „Sicherheitstesten im gesamten Softwarelebenszyklus“

4.1 Die Rolle des Sicherheitstestens im Softwareentwicklungslebenszyklus

AS-4.1.1 (K2) Erläutern können, warum sich Sicherheit am besten innerhalb eines Lebenszyklusprozesses erreichen lässt

AS-4.1.2 (K3) Die entsprechenden, sicherheitsbezogenen Aktivitäten für einen gegebenen Softwareentwicklungslebenszyklus (z. B. iterativ, sequenziell) realisieren können

4.2 Die Rolle des Sicherheitstestens in der Anforderungsermittlung

AS-4.2.1 (K4) Einen gegebenen Satz von Anforderungen aus der Sicherheitsperspektive analysieren können, um Unzulänglichkeiten ermitteln zu können

4.3 Die Rolle des Sicherheitstestens beim Entwurf

AS-4.3.1 (K4) Ein gegebenes Entwurfsdokument aus der Sicherheitsperspektive analysieren können, um Unzulänglichkeiten ermitteln zu können

4.4 Die Rolle des Sicherheitstestens bei der Implementierungsarbeit

AS-4.4.1 (K2) Die Rolle des Sicherheitstestens im Komponententesten verstehen

AS-4.4.2 (K3) Sicherheitstests auf Komponentenebene (abstrakt) bei einer definierten Spezifikation realisieren können

AS-4.4.3 (K4) Ergebnisse eines gegebenen Tests auf Komponentenebene analysieren können, um die Angemessenheit von Programmcode aus der Sicherheitsperspektive ermitteln zu können

AS-4.4.4 (K2) Die Rolle des Sicherheitstestens beim Komponentenintegrationstest verstehen

AS-4.4.5 (K3) Komponentenintegrations-Sicherheitstests (abstrakt) bei einer definierten Systemspezifikation realisieren können

4.5 Die Rolle des Sicherheitstestens in System- und Abnahmetest-Aktivitäten

AS-4.5.1 (K3) Ende-zu-Ende-Testszenarien für Sicherheitstests realisieren zu können, die eine oder mehrere Sicherheitsanforderungen verifizieren und einen beschriebenen funktionalen Ablauf testen

AS-4.5.2 (K3) Einen Satz von Abnahmekriterien für die Sicherheitsaspekte eines gegebenen Abnahmetests definieren können

4.6 Die Rolle des Sicherheitstestens bei der Wartung

AS-4.6.1 (K3) Eine durchgängige Vorgehensweise für Sicherheitswiederholungstests bzw. Regressionstests auf der Basis eines gegebenen Szenarios realisieren können

4.1 Die Rolle des Sicherheitstestens im Softwareentwicklungslebenszyklus

Sicherheit wird nicht bei einer bereits fertigen Anwendung getestet oder „eingefügt“. Vielmehr ist sie das Ergebnis einer sicherheitsorientierten Entwicklung und Verifikation im gesamten Entwicklungsprozess. Wie beim Testen von Software im Allgemeinen ist das Testen der Sicherheit ein Prozess, der innerhalb des Entwicklungslebenszyklus erfolgen muss.

4.1.1 Sicherheitstests und die Lebenszyklus-Perspektive

Ein Softwareentwicklungslebenszyklusprozess bietet einen Rahmen für die Durchführung bestimmter Aktivitäten zu Zeitpunkten, die auf andere Aktivitäten abgestimmt sind. So müssen beispielsweise die Erfordernisse der Benutzer ermittelt werden, bevor die Entwicklung einer Anwendung beginnt. Die Auswahl des Softwareentwicklungslebenszyklus hängt von der Art der Organisation, vom Projekt und von ähnlichen Faktoren ab [IEEE 12207]. Für den Zweck dieses Lehrplans und für das Sicherheitstesten können die Konzepte und Techniken auf jeden Lebenszyklusprozess – sequenziell oder iterativ – angewendet werden.

In Kapitel 3 dieses Lehrplans wurde ein Sicherheitstestprozess beschrieben, der an einem allgemeinen Muster-Softwareentwicklungslebenszyklus ausgerichtet ist. Die Gründe für das Einbinden der Sicherheitstests in den Softwarelebenszyklus werden in den folgenden Abschnitten erläutert.

Vorgabe eines festen Zeitpunkts im Entwicklungslebenszyklus, zu dem sicherheitsbezogene Aktivitäten stattfinden sollten

Beim Erfassen und Definieren von Benutzererfordernissen muss der Geschäfts- oder Systemanalyst beispielsweise folgende Fragen stellen:

1. Welche Stufen des Sicherheitszugriffs werden benötigt?
2. Gibt es digitale oder physische Assets, die spezielle Sicherheitsvorkehrungen benötigen?
3. Wie „offen“ soll die Anwendung sein?
4. Welche Sicherheitsrisiken gibt es?

Ein weiteres Beispiel während der Programmierung: Zu diesem Zeitpunkt hat der Entwickler die beste Möglichkeit, sichere Programmierpraktiken anzuwenden, um Angriffe wie SQL-Injection und Speicher-Pufferüberläufe zu verhindern. Schwachstellen dieser Art in späteren Phasen des Projekts zu finden, ist schwierig und teuer, weil u.U. viele weitere Softwarekomponenten ebenso geprüft und korrigiert werden müssten.

Vorgabe von Prüfpunkten für das Review

Beispielsweise müssen Sicherheitsanforderungen oder User-Storys geprüft werden, um sicherzustellen, dass die sicherheitsbezogenen Aspekte der Benutzererfordernisse angemessen untersucht und dokumentiert wurden. Programmcode-Änderungen müssen ebenfalls auf die Präsenz bösartigen Codes geprüft werden, der u.U. von internen Mitarbeitern oder Auftragnehmern eingeschleust wurde.

Vorgabe von Prüfpunkten für das Testen

In der Entwicklung sollten z. B. Komponententests durchgeführt und dokumentiert werden, um zu verifizieren, dass sichere Programmierpraktiken eingehalten und erfolgreich implementiert wurden.

Zur Vorgabe von Eingangs- und Endekriterien im gesamten Projekt

Ein Beispiel für diese Praxis wäre, dass keine Komponente für eine integrierte Testumgebung akzeptiert wird, solange nicht nachgewiesen werden kann, dass alle sicherheitsbezogenen Aktivitäten (Entwicklung und Tests) erfolgreich abgeschlossen wurden. Besonders wichtig ist das in späteren Projektphasen, in denen eine Sicherheitsschwachstelle ein Sicherheitsrisiko für das gesamte System oder die Anwendung entstehen lassen würde.

4.1.2 Sicherheitsbezogene Aktivitäten im Softwareentwicklungslebenszyklus

Die folgenden sicherheitsbezogenen Aktivitäten werden parallel zu anderen Projektaktivitäten durchgeführt – im Gegensatz zur Durchführung in einem eigenen gesonderten Projektlebenszyklus.

Anforderungen: Anforderungen werden in Abhängigkeit vom genutzten Softwareentwicklungslebenszyklus auf vielerlei Weise erfasst und definiert. Angemerkt sei dabei, dass Anforderungen über die Erfordernisse von Benutzern und Stakeholdern hinausgehen können. So kann es beispielsweise u.a. regulatorische, technische und geschäftliche Anforderungen geben.

Folgende anforderungsbezogene Ziele gibt es:

1. Verstehen und Ermitteln von Sicherheitserfordernissen aus allen Perspektiven innerhalb und außerhalb des Unternehmens. So ist beispielsweise der Kunde eines Unternehmens kein Teil des Unternehmens, hat aber ein berechtigtes Interesse sowie den gesetzlichen Rückhalt, dass seine personenbezogenen Daten sicher bleiben.
2. Dokumentieren der Sicherheitserfordernisse auf detaillierte und eindeutige Weise. Das macht es möglich, dass Implementierungen und Tests auf die Anforderungen zurückzuführen sind. So lassen sich die Anforderungen später verifizieren und validieren.

Folgende anforderungsbezogene Aktivitäten gibt es:

1. Ermitteln aller involvierten und kompetenten Personen, die einen Beitrag zum Definieren der Anforderungen leisten können.
2. Verwendung einer Vielzahl von Methoden – Gespräche, Workshops usw.. Erfassung der Sicherheitserfordernisse, die von den einzelnen Gruppen geäußert werden. Dies kann auch während der Erhebung weiterer Anforderungen erfolgen.
3. Dokumentieren der Anforderungen auf überprüfbare und rückverfolgbare Weise.
4. Prüfen der Anforderungen auf Richtigkeit, Vollständigkeit, Verständlichkeit und Eindeutigkeit.

Entwurf: Das System bzw. die Anwendung wird auf der Basis der in den Anforderungen angegebenen Erfordernisse entworfen. Aus den Anforderungen gehen die Sicherheitserfordernisse hervor. Der Entwurf setzt die Erfordernisse in einen funktionierenden Lösungsansatz um.

Folgende entwurfsbezogene Ziele gibt es:

1. Erstellung eines System- oder Anwendungsentwurfs, der den angegebenen Sicherheitsanforderungen genügt

Folgende entwurfsbezogene Aktivitäten gibt es:

1. Analyse der dokumentierten Anforderungen
2. Auswahl der praktikabelsten Herangehensweise für die Entwicklung der Anwendung auf sichere Art
3. Dokumentieren des Entwurfs mittels geeigneter Techniken im Einklang mit dem Softwareentwicklungslebenszyklus. Beim iterativen Ansatz können die Entwurfssitzungen z. B. an einem Whiteboard durchgeführt werden. Bei anderen Prozessen muss der Entwurf u. U. in Form von Modellen ausgedrückt werden.

Implementierung: Dabei handelt es sich um die eigentliche Programmierung.

Folgende implementierungsbezogene Ziele gibt es:

1. Umsetzung von Anforderungen und Entwurf in sicheren Programmcode, der die in den Anforderungen formulierten funktionalen Erfordernisse erfüllt.
2. Implementierung sonstiger benötigter Verfahren oder Technologien (Firewalls, Tokens usw.), um die Sicherheitsvorgaben zu erfüllen

Folgende implementierungsbezogene Aktivitäten gibt es:

1. Erstellung von Programmcode, der den Sicherheitsanforderungen genügt
2. Durchführung von Komponententests zur Überprüfung von Richtigkeit, Wirksamkeit und Sicherheit der Implementierung
3. Durchführung von Komponenten-Reviews zur visuellen Inspektion der Richtigkeit, Wirksamkeit und Sicherheit der Implementierung

Systemtest:

Zu beachten ist, dass bei einigen Softwareentwicklungslebenszyklus-Modellen, wie z. B. den iterativen Ansätzen, innerhalb kurzer Zeit neue Komponenten hinzukommen oder bestehende Komponenten verfeinert werden. Ferner können hier Systemtests viel häufiger als bei anderen, stärker sequenziellen Ansätzen vorkommen.

Folgende systemtestbezogene Ziele gibt es:

1. Durchführung eines Ende-zu-Ende-Tests zur Beobachtung des gesamten Funktionierens und der Performance des kompletten Systems (Hardware, Software, Daten, Menschen und Verfahren) nach der Implementierung verschiedener Systemkomponenten und deren Integration in ein Komplettsystem
2. Testen, dass die Sicherheitsanforderungen aus Systemperspektive richtig implementiert wurden

Folgende systemtestbezogene Aktivitäten gibt es:

1. Durchführung von Sicherheitstests in annähernd der endgültigen Zielumgebung, was einen Übergang von der Entwicklungsumgebung notwendig macht, in der die vorherigen Implementierungs- und Integrationsaktivitäten erfolgten

Abnahmetests:

Das ist die letzte Stufe des Testens. Hier überzeugen sich zukünftige Benutzer des Systems oder deren Vertreter davon, dass das System in der Zielumgebung die benötigten Fähigkeiten liefert.

Folgendes abnahmetestbezogene Ziel gibt es:

1. Durchführung von Sicherheitstests anhand sicherheitsbezogener Abnahmekriterien für das System durch die Nutzer oder mögliche Vertreter, die in ihrem Namen agieren. Im Mittelpunkt der sicherheitsbezogenen Abnahmekriterien stehen häufig die funktionalen Sicherheitskontrollmechanismen und -prozesse.

Folgende abnahmetestbezogene Aktivitäten gibt es:

1. Installieren des Systems in seiner Betriebsumgebung
2. Durchführen von Sicherheitstests anhand von Abnahmekriterien
3. Entscheiden auf Basis der Testergebnisse, ob Abnahme erfolgt

Dabei ist zu beachten, dass System- und Abnahmetests im Wesentlichen „Black-box“- oder Stimulus-Response-Tests ohne Berücksichtigung der internen Struktur oder des Verhaltens von Komponenten innerhalb des Gesamtsystems sind. Vorhergehende Komponenten- und Integrationstests liefern eine ergänzende Bewertungsgrundlage durch ihre besondere Berücksichtigung der internen Komponentenarchitektur und der Interaktion der Komponenten im System.

Wartung:

Nach der Inbetriebnahme eines Systems ist u.U. zusätzliche Entwicklungsarbeit nötig, um Fehler in der freigegebenen Version zu beheben (korrektive Wartung), um sonstigen Änderungen in der Betriebsumgebung Rechnung zu tragen (adaptive Wartung) oder um den Funktionsumfang zu erweitern bzw. zu optimieren (verbessernde Wartung).

Im Mittelpunkt der Sicherheitstest-Perspektive für die Systemwartung steht das Testen von Änderungen zum Beheben von Fehlern (Fehlernachtests) und der Kernfunktionen (Regressionstests). Ziel ist dabei Folgendes:

1. Gewährleisten, dass durch die Wartung keine neuen Schwachstellen im System entstanden sind
2. Prüfen, ob bestehende Sicherheitsvorkehrungen auch nach einer Änderung noch wirksam sind

In diesem Zusammenhang kann die Wartung Folgendes umfassen: Upgrades (z. B. Betriebssystem, Datenbanken), Änderungen am Programmcode, Datenumwandlungen und Migration auf Plattformen.

Im Wesentlichen sollte jede Wartungsaktivität mit derselben Sorgfalt und Aufmerksamkeit wie die ursprüngliche Entwicklungsarbeit durchgeführt werden. Andernfalls besteht die Gefahr der Erzeugung neuer Schwachstellen, die eine ernste Gefährdung der Sicherheit des laufenden Systems darstellen können.

4.2 Die Rolle des Sicherheitstestens in der Anforderungsermittlung

Folgende Überlegungen zu Anforderungen im Allgemeinen müssen nachvollzogen werden:

1. Für viele Unternehmen ist es bereits eine Herausforderung, elementare Benutzeranforderungen zu formulieren, die aussagekräftig, unmissverständlich, vollständig, richtig und testbar sind.
2. Anforderungen ändern sich im Verlauf eines Projektes mit großer Wahrscheinlichkeit. Daher kann die Wartung der Anforderungen eine Herausforderung sein.
3. Es bedarf besonderer Kompetenzen, die Erfordernisse der Benutzer sowie andere Erfordernisse so zu verstehen – z. B. Compliance- und technische Erfordernisse –, sodass man in der Lage ist, sie in Dokumenten niederzulegen oder in Anforderungsmanagement-Werkzeuge einzugeben.
4. Anforderungen können Lücken und Fehler enthalten. Daher ist eine Verifizierung und Validierung erforderlich.
5. Anforderungen *sollten* Vorgaben im Hinblick auf Qualitätsmerkmale wie Sicherheit, Leistung, Brauchbarkeit usw. enthalten. Diese Merkmale werden zugunsten der reinen Funktionalität jedoch häufig übersehen.

Die Herausforderung besteht darin, dass die Sicherheitsperspektive verstanden und in den kompletten Anforderungen für ein Projekt umgesetzt wird. Eine wirksame Technik bei der Evaluierung von Anforderungen ist die Nutzung einer Checkliste als Leitfaden. Diese Checkliste kann eine Vielzahl von Punkten enthalten, um viele Themenbereiche abzudecken. Im Hinblick auf die sicherheitsbezogenen Merkmale ist Folgendes ein guter Ausgangspunkt für die Evaluierung:

Datenschutzerfordernisse

1. Wurden alle Benutzergruppen und ihre Datenschutzerfordernisse ermittelt und dokumentiert?
2. Wurden alle Datentypen, die von dieser Anforderung betroffen sind, ermittelt und die entsprechenden Datenschutzerfordernisse definiert?
3. Wurden die Benutzerzugriffsrechte ermittelt und dokumentiert?

Compliance-Erfordernisse (für Sicherheitsrichtlinien)

1. Wurden alle relevanten Sicherheitsrichtlinien ermittelt und dokumentiert?
2. Wurden Ausnahmen von Sicherheitsrichtlinien ermittelt und dokumentiert?

Gängige Schwachstellen: Diese werden sich mit dem Wandel der Sicherheitsangriffe ebenfalls ändern, sollten aber beim Definieren von Anforderungen als Risiken definiert werden. Später werden sie die Basis für die Sicherheitstests bilden.

1. Wurden alle gängigen und bekannten Sicherheitsschwachstellen für die zu dokumentierende Funktion als bekannte Risiken ermittelt?

Testbarkeit

2. Sind die Anforderungen so formuliert, dass auf der Basis dieses Dokuments Sicherheitstests und andere Tests geschrieben werden können?
3. Werden zu vage Formulierungen wie „die Verarbeitung muss sicher sein“ und „Zugang wird nur autorisiertem Personal gewährt“ ermittelt und präzisiert, damit sie konkret und testbar sind?

Benutzerbarkeit

Es gibt Kompromisse zwischen Sicherheit und Benutzbarkeit. Ein Beispiel: Die Benutzeranmeldung bei einer Website kann so verwirrend und schwierig sein, dass die Kunden aufgeben und sich andere Anbieter suchen.

1. Spiegeln die Anforderungen einen Sicherheitsprozess wider, der in Relation zu der zu spezifizierenden Funktion angemessenen ist?
2. Sind die Sicherheitsverfahren aussagekräftig und verständlich?
3. Werden Maßnahmen spezifiziert, die legitimierten Benutzern, die Probleme beim Zugriff auf Informationen haben, Hilfe zur Verfügung stellen?

Leistung

Es gibt Kompromisse zwischen Sicherheit und Leistung. So kann eine starke Verschlüsselung beispielsweise die Performance bremsen.

1. Spiegeln die Anforderungen eine in Relation zu der zu spezifizierenden Funktion angemessene Wirksamkeit der Sicherheitsvorkehrungen wider?

4.3 Die Rolle des Sicherheitstestens beim Entwurf

Sicherheitsgefährdende Entwurfspraktiken sind zu ermitteln und zu vermeiden. Testbezogene Aktivitäten tragen zur Erkennung von Softwaresystementwürfen bei, welche wahrscheinlich anfällig für Angriffe sind. Sie steuern den Entwurf von Softwaresystemen mit starken, erkennbaren Sicherheitseigenschaften.

Das IEEE Center for Secure Design [IEEE1] empfiehlt folgende zentralen Vorgehensweisen:

1. Vertrauen kann verdient oder entgegengebracht, aber nie vorausgesetzt werden.
2. Es ist ein Authentifizierungsmechanismus zu nutzen, der sich nicht umgehen oder manipulieren lässt.
3. Erst authentifizieren, dann autorisieren.
4. Daten und Steuerbefehle sind streng zu trennen; Steuerbefehle aus nicht vertrauenswürdigen Quellen dürfen nie ausgeführt werden.
5. Es ist ein Vorgehen zu definieren, sodass gewährleistet ist, dass alle Daten explizit validiert werden.
6. Verschlüsselung muss richtig angewendet werden.
7. Es sind sensible Daten und deren Handhabung zu ermitteln.
8. Es ist stets der Benutzer zu berücksichtigen.

9. Es muss verstanden werden, auf welche Weise die Integration externer Komponenten die Angriffsfläche eines Systems verändert.
10. Bei Überlegungen zu zukünftigen Änderungen bei Objekten und Akteuren ist Flexibilität gefordert.

4.4 Die Rolle des Sicherheitstestens bei der Implementierungsarbeit

Wie andere Arten von Tests beginnen Sicherheitstests auf der untersten Implementierungsebene – an separaten Softwarekomponenten, die später das Gesamtsystem bilden. Nach der statischen Evaluierung dieser Komponenten bietet das Testen eine zusätzliche Stufe der Untersuchung des dynamischen Verhaltens in Reaktion auf gültige und ungültige Eingaben.

4.4.1 Sicherheitstests während der Komponententests

4.4.1.1 Überlegungen zum White-Box-/Glass-Box-Testen

Statische Tests, die das gesamte Spektrum an Inspektionen, Walkthroughs, Audits und technischen Review-Aktivitäten einschließen, wurden bereits erwähnt.

So genannte White-Box- und/oder Glass-Box-Tests (strukturell) sind Tests, die auf der Basis des Einblicks in den Softwareentwurf oder die Softwareimplementierung entworfen werden. Black-Box-Tests (funktional und nicht-funktional) basieren hingegen nicht auf dem Zugang zu solchen strukturellen Informationen. Es sind einfache Stimulus-Response-Tests.

Mit White-Box-Tests lassen sich zielgerichtet spezifische Kontrollmechanismen, die im Modul implementiert sind, testen sowie deren Wirksamkeit prüfen. Einblick in die Komponentenstruktur ermöglicht die Messung des Überdeckungsgrades der Tests – als Prozentwert der ausgeführten ausführbaren Anweisungen, Prozentwert der ausgeführten Entscheidungsergebnisse oder Prozentwert der durchlaufenen Logikpfade.

Strukturelle Sicherheitstests können von automatisierten statischen Analysewerkzeugen und Sicherheits-Scannern durchgeführt werden. Fuzz-Tests sind eine Sicherheitstechnik, die dem Auffinden von Sicherheitsschwachstellen durch Eingabe massiver Mengen von Zufallsdaten (Fuzz) in die getestete Komponente bzw. das System dient. White-Box-Fuzz-Tests (an kleinen Softwareblöcken, Funktionen, Klassen) ergeben u.U. in viel kürzerer Zeit brauchbare Ergebnisse als ein Black-Box-Fuzz-Testwerkzeug.

White-Box-Fuzz-Testwerkzeuge können Speicherfehler, Pufferüberläufe usw. entdecken, indem sie den zu testenden Programmcode instrumentieren.

Folgende Sicherheitsschwachstellen lassen sich beim strukturellen Testen ermitteln und beheben:

1. Speicherpufferüberläufe
2. Bösartiger Programmcode, der von einem internen Mitarbeiter oder Auftragnehmer eingeschleust wurde
3. Zugang über eine „Hintertür“ (Zugang über eine nicht dokumentierte Schnittstelle, die nur der Entwickler kennt und die bewusst eingebaut wurde, um die regulären Sicherheitskontrollmechanismen zu umgehen.)

4.4.1.2 Überlegungen zu funktionalen Sicherheitstests

Die Adäquatheit von Sicherheitstests auf jeder Ebene muss durch Nachweis der Abdeckung spezifizierter Sicherheitsanforderungen ermittelt werden. Dies erfolgt ergänzend zur Bewertung der Ergebnisse aus Belastungssituationen, die in den Sicherheitsanforderungen, Sicherheitsrisiko-Bewertungen und ähnlichen Dokumenten nicht explizit aufgeführt sind. Bei der Suche nach Schwachstellen ist Kreativität gefordert, da Tester untersuchen, was Entwickler übersehen haben.

4.4.2 Entwurf von Sicherheitstests auf der Komponentenebene

Sehr gute Beispiele für Best Practices beim Programmieren finden sich im Artikel „Top 10 Secure Coding Practices“ [CERT1]. Dort heißt es:

„Die Tests einer jeden Komponente sollten die Prüfung auf mögliche Verstöße gegen diese Praktiken einschließen:

1. Validierung von Eingaben
2. Compiler-Warnungen beachten
3. Architektur und Entwurf gemäß der Sicherheitsrichtlinien
4. Einfach halten (Keep it Simple)
5. Standardmäßiges Blockieren (Default Deny)
6. Einhaltung des Grundsatzes der geringstmöglichen Zugriffsrechte
7. Bereinigung von Daten, die an andere Systeme geschickt werden
8. Gestaffelte Sicherheitsarchitektur (Defense in Depth)
9. Nutzung wirksamer Qualitätssicherungstechniken
10. Anwendung eines sicheren Programmierstandards“

Tests, die durch Abarbeitung solcher Best-Practice-Checklisten durchgeführt werden, müssen die Überprüfung auf mögliche Verstöße gegen diese Praktiken auf Basis einer gut dokumentierten Risikoanalyse, die eine realistische Gefährdungsmodellierung einschließt, umfassen. Mit anderen Worten: Fokussierung auf die wichtigsten Anforderungen im Hinblick auf die Angriffswahrscheinlichkeit und das Schadensausmaß.

4.4.3 Analyse von Sicherheitstests auf Komponentenebene

Eine wichtige Maßzahl für die Angemessenheit ist die Ermittlung des Überdeckungsgrades von Tests. Verschiedene Überdeckungsgrade ergeben sich aus der Art des durchgeführten Tests.

Anforderungsbasierte Tests prüfen das System dahingehend, ob es die ihm vorgegebenen Anforderungen erfüllt. Ohne Berücksichtigung der Implementierung (Black-Box) kann der Überdeckungsgrad wie folgt gemessen werden:

1. Prozentsatz der Gesamtanzahl der getesteten Anforderungen
2. Prozentsatz der spezifizierten getesteten Nutzungs-/Missbrauchsfälle
3. Prozentsatz der getesteten kritischen Funktionen, Szenarien oder Aufgabenpfade

Beim datengetriebenen Testen wird das Verhalten des Systems über eine Vielzahl von Eingabedaten sowie deren Kombination geprüft. Dabei versucht man, so wenig Testwerte wie möglich zu verwenden, indem man den Datenraum in Äquivalenzklassen unterteilt und aus jeder Klasse einen Vertreter wählt (in der Annahme, dass die Elemente dieser Klasse im Hinblick auf ihre Fähigkeit, Fehler zu erkennen, äquivalent sind). Paarweise und n-weise Überdeckungskriterien sind typische Formen von Datenüberdeckungskriterien.

Das modellbasierte Testen ermöglicht die Ermittlung des Überdeckungsgrades im Hinblick auf eine gewählte Modellierungsnotation. Wenn das Modell eine Notation mit Vor- und Nachbedingungen nutzt, kann der Überdeckungsgrad der Ursache-Wirkungsketten und der Überdeckungsgrad aller Disjunkte in der Nachbedingung als Überdeckungskriterium verwendet werden. Bei algebraischen Modellierungsnotationen ist der Überdeckungsgrad der Axiome ein typisches Überdeckungskriterium.

Bei übergangsbasierten Modellen, die explizite Graphen mit Knoten und Kanten nutzen, gehören der Prozentsatz der Knoten (Zustände), der Prozentsatz der Übergänge, der Prozentsatz der Übergangspaare und der Prozentsatz der Zyklen zu den Überdeckungskriterien.

Beim strukturellem Testen wird die tatsächliche Implementierung auf Basis von Transparenz (Einblick) in den Programmcode analysiert. Der Testüberdeckungsgrad wird üblicherweise als Prozentsatz der Pakete, Klassen, Methoden, Entscheidungen oder Zeilen ausführbaren Programmcodes in der Anwendung dokumentiert, die bei den Tests ausgeführt wurden. Letzteres wird als Anweisungsüberdeckung bezeichnet.

Die zyklomatische Komplexität ist eine Maßzahl, die beschreibt, wie viele verschiedene unabhängige Pfade durch ein Element existieren, und sich mit Hilfe eines Kontrollflussgraphs mit Knoten (Entscheidungspunkten) und Kanten (Pfaden) visualisieren lässt. Das stärkste der kontrollflussbasierten Kriterien ist die Pfadüberdeckung. Sie wird über alle Pfade vom Eingang bis zum Ausgang des Kontrollflussgraphen gemessen. Weil ein erschöpfendes Testen von Pfaden aufgrund von Schleifen im Allgemeinen nicht durchführbar ist, lassen sich andere, weniger strenge Kriterien im Hinblick auf die ausgewählten logischen Pfade, die als kritisch gelten (Überdeckung kritischer Pfade), oder den Prozentsatz der geprüften Entscheidungsergebnisse (Zweigüberdeckung) heranziehen.

4.4.4 Sicherheitstests während der Komponentenintegrationstests

Weil Komponenten niederer Ebenen in Subsysteme und letztlich in das komplette Zielsystem integriert werden, sind die Möglichkeiten für Sicherheitsverstöße nicht einfach die Summe der Schwachstellen in den einzelnen, separat betrachteten Komponenten. Aufgrund der Interaktionen zwischen Komponenten und mit größeren Systemen und Organisationseinheiten ergeben sich vielmehr neue Angriffsvektoren.

Andererseits können einige Interaktionen zwischen Komponenten mögliche Abläufe, die zu Sicherheitsverstößen führen, auch mindern oder blockieren. Auch hier gilt: Sicherheitstester müssen kreativ sein, wenn sie nach Fehlern suchen, die von Entwicklern übersehen wurden.

Integrationstests können die Komplexität eines Systementwurfs und die Stabilität seines Verhaltens demonstrieren. Die gewählte Integrationstestvorgehensweise (z. B. Top-Down oder Bottom-Up) kann sich auf den Zeitpunkt des Aufdeckens von Sicherheitsproblemen auswirken oder die Notwendigkeit für zusätzliche sicherheitsspezifische Tests erzeugen.

4.4.5 Entwurf von Sicherheitstests auf der Komponentenintegrationsebene

Wie Komponententests müssen auch Integrationstests auf der Basis einer gut dokumentierten Risikoanalyse, die eine realistische Gefährdungsmodellierung einschließt, entworfen werden. Wenn separate Komponenten zu einem Ganzen integriert werden, ist u.U. die Nutzung eines Testrahmens (in Form von Platzhaltern und Treibern) notwendig, um unvollständige Aufrufpfade auch in einem System während der Integration zu testen. Wenn mehr und mehr implementierte Komponenten zum System hinzukommen, verschwindet dieser Testrahmen schrittweise. Das ermöglicht eine umfassendere Bewertung der Funktionalität sowie die Prüfung neuer Pfade zu Schwachstellen, die pot. ausgenutzt werden könnten.

4.5 Die Rolle des Sicherheitstestens in System- und Abnahmetest-Aktivitäten

Die Rolle des Sicherheitstestens beim Testen des Systems

Systemtests sind die erste durchgängige Ausführung der vollständig integrierten Komponenten. Obwohl sie in der Regel in einer Entwicklungsumgebung erfolgen, sollten sie die entstehenden Eigenschaften des Systems offenlegen, die vor dem Abschluss der Integration nicht zu sehen waren. Sicherheitsanforderungen werden typischerweise im Zusammenhang mit einer oder mehreren funktionalen Anforderungen betrachtet.

Zum Beispiel: „Im Verlauf der Ausführung von x darf das System nicht zulassen, dass y passiert.“ Wenn funktionale Tests durchgeführt werden, sollte der Tester nach Wegen suchen, Sicherheitsbarrieren zu überwinden.

Funktionale Anforderungen, einschließlich der sicherheitsbezogenen, sind in der Regel Muss-Anforderungen. Sonstige Spezifikationen wie Anwendungsfälle, Missbrauchsfälle, Prozessmodelle und Zustandsübergangsmodelle beschreiben darüber hinaus Verfahren, die als Grundlage für die Definition durchgängiger Sicherheitstestszenarien genutzt werden können.

4.5.1 Die Rolle des Sicherheitstestens bei Abnahmetests

Abnahmetests unterscheiden sich von Systemtests dahingehend, dass sie in einer realistischen Betriebsumgebung oder sogar in der tatsächlichen Umgebung, in der das System betrieben wird, durchgeführt werden. Diese Tests ermöglichen eine angemessene Bewertung der Leistung und sonstiger Verhaltensweisen auf der Basis der Interaktionen über externe Schnittstellen. Sie bringen das System zudem letztlich in die Umgebung, in der externe Akteure ständig versuchen würden, Schwächen zu finden.

Bei Abnahmetests sollte idealerweise validiert werden, dass die ursprünglichen Projektziele umgesetzt wurden. Dies wird durch Entwurf und Ausführung von Tests erreicht, die validieren, dass Abnahmekriterien erfüllt werden. In den Abnahmekriterien müssten auch Sicherheitserfordernisse niedergelegt sein.

Der beste Zeitpunkt für das Definieren und Dokumentieren von Abnahmekriterien ist vor der Entwicklung oder dem Kauf des Systems. Deshalb kann zwischen dem Lieferanten und dem Käufer eine erste Übereinkunft getroffen werden, selbst wenn beide demselben Unternehmen angehören. Abnahmekriterien können sich zudem während eines Projekts ändern oder neu entstehen. Daher sollten sie auf ihre Auswirkungen beim Testen der Sicherheit analysiert werden.

Im Kontext von Sicherheitstests können Abnahmekriterien von globalem Charakter sein. Ein Beispiel: In Abnahmekriterien kann es Punkte geben, die festlegen, was im Hinblick auf die Gesamtsystemsicherheit akzeptabel ist. Das würde Kriterien einschließen, die auf alle Systemfunktionen wie Benutzerauthentifizierung, Benutzerrechte, Verschlüsselungsgrade, Audit-Nachweise usw. angewendet werden. In anderen Fällen werden u.U. sicherheitsspezifische Abnahmekriterien benötigt. Manche Funktionen wie z. B. das Anweisen von Zahlungen, die einen bestimmten Betrag übersteigen, können z. B. die Genehmigung von zwei Personen erfordern.

4.6 Die Rolle des Sicherheitstestens bei der Wartung

Mit Regressionstests soll bestätigt werden, dass alle vorher akzeptablen Verhaltensweisen des Systems auch nach vorgenommenen Modifikationen intakt bleiben. Negative Sicherheitstests bestätigen hingegen, dass das System Angriffen zur Überwindung der eingerichteten Sicherheitsvorkehrungen auch weiterhin erfolgreich standhält. Verbesserungen bezüglich Benutzbarkeit oder Effizienz sind besonders anfällig für einen Verlust an Sicherheit.

Im Mittelpunkt von Sicherheitsregressionstests muss die Bestätigung der Erfüllung aller Sicherheitsanforderungen sowie das Testen auf neue Schwachstellen, die u.U. bei den Wartungsaktivitäten erzeugt wurden, stehen.

Regressionstests werden häufig anhand einer Sammlung von Testfällen durchgeführt, die auf dem Test einzelner Funktionen basieren. Bei Sicherheitstests reicht dies jedoch häufig nicht, um Regressionsfehler mit Sicherheitsverlust zu entdecken. Durchgängige Regressionstestszenarien sind robuster und liefern einen höheren Grad an Vertrauen dahingehend, dass vollständige Transaktionen sicher durchgeführt werden können.

Für diese Art von Regressionstests sollte bei jeder Änderung am System eine Reihe von Sicherheitstestszenarien definiert und getestet werden. Beachten Sie, dass sich Systemänderungen auch auf Hardware, Konfigurationsdateien, Betriebssysteme, DBMS, Netzwerke und Software – sowie weitere Systemkomponenten – erstrecken können. Regressionsfehler können eine Folge von Änderungen an jedem dieser Elemente sein. Regressionsfehler können sich auf die Sicherheit auswirken.

Beispielszenarien:

Benutzer können sich bei einer Website anmelden und dort sicher einen Einkauf tätigen, ohne dass ihre personenbezogenen Daten kompromittiert werden.

Benutzer können nur Handlungen ausführen, die in ihren Benutzerrechten definiert sind. (Ein Benutzer, der in der Lohnbuchhaltung arbeitet, darf einen neuen Mitarbeiter hinzufügen, hat aber keinen Zugriff auf dessen Bankdaten.)

5 Testen von Sicherheitsmechanismen – 240 min

Schlüsselbegriffe

Anti-Malware, Authentifizierung, Autorisierung, demilitarisierte Zone, Verschlüsselung, Firewall, Hashfunktion, Gefährdung durch Betriebsangehörige, Angriffserkennungssystem, Malware, Malware-Scan, Netzwerkzone, Pharming, Phishing, Salting, Systemhärtung, Schwachstellenscanner

Lernziele für das Thema „Testen von Sicherheitsmechanismen“

5.1 Systemhärtung

AS-5.1.1 (K2) Das Konzeptsder Systemhärtung und ihrer Rolle bei der Optimierung der Sicherheit verstehen

AS-5.1.2 (K3) Demonstrieren können, wie sich die Wirksamkeit allgemeiner Mechanismen der Systemhärtung testen lässt

5.2 Authentifizierung und Autorisierung

AS-5.2.1 (K2) Den Zusammenhang zwischen Authentifizierung und Autorisierung verstehen sowie bei der Sicherung von Informationssystemen anwenden können

AS-5.2.2 (K3) Demonstrieren können, wie sich die Wirksamkeit allgemeiner Authentifizierungs- und Autorisierungsmechanismen testen lässt

5.3 Verschlüsselung

AS-5.3.1 (K2) Das Konzept der Verschlüsselung sowie ihre Anwendung bei der Sicherung von Informationssystemen verstehen

AS-5.3.2 (K3) Demonstrieren können, wie sich die Wirksamkeit allgemeiner Verschlüsselungsmechanismen testen lässt

5.4 Firewalls und Netzwerkzonen

AS-5.4.1 (K2) Das Konzept der Firewalls und den Einsatz von Netzwerkzonen sowie ihre Anwendung bei der Sicherung von Informationssystemen verstehen

AS-5.4.2 (K3) Demonstrieren können, wie sich die Wirksamkeit bestehender Firewall-Implementierungen und Netzwerkzonen testen lässt

5.5 Angriffserkennung

AS-5.5.1 (K2) Das Konzept der Angriffserkennungswerkzeuge sowie ihre Anwendung bei der Sicherung von Informationssystemen kennen

AS-5.5.2 (K3) Demonstrieren können, wie sich die Wirksamkeit bestehender Angriffserkennungswerkzeuge testen lässt

5.6 Malware-Scan (Schadprogramm-Scan)

AS-5.6.1 (K2) Das Konzept der Malware-Scanner sowie ihrer Anwendung bei der Sicherung von Informationssystemen kennen

AS-5.6.2 (K3) Demonstrieren können, wie sich die Wirksamkeit bestehender Malware-Scanner testen lässt

5.7 Datenmaskierung

AS-5.7.1 (K2) Das Konzept der Datenmaskierung, der Werkzeuge zur Datenmaskierung sowie ihrer Anwendung bei der Sicherung von Informationssystemen kennen

AS-5.7.2 (K3) Demonstrieren können, wie sich die Wirksamkeit von Datenmaskierungsansätzen testen lässt

5.8 Schulung

AS-5.8.1 (K2) Das Konzept der Sicherheitsschulung als Aktivität des Softwareentwicklungslebenszyklus und ihrer Notwendigkeit bei der Sicherung von Informationssystemen kennen

AS-5.8.2 (K3) Demonstrieren können, wie sich die Wirksamkeit von Sicherheitsschulungen testen lässt

5.1 Systemhärtung

Über die Jahre wurde eine Vielzahl von Sicherheitsmechanismen entwickelt, die beim Schutz digitaler und physischer Assets eine zentrale Rolle spielen. Jeder dieser Mechanismen lässt sich auf verschiedene Arten einsetzen – einige über Werkzeuge und Infrastruktur, andere über händische Arbeiten. Keiner dieser Mechanismen reicht in den meisten Fällen alleine aus, um Informationen zu schützen. Jeder Mechanismus hat seine eigenen Vor- und Nachteile.

Sicherheitstester müssen die Nuancen jeder Abwehrmaßnahme verstehen, damit entsprechende Tests für die Verifizierung und Validierung ihrer Wirksamkeit entwickelt werden können. Sicherheitstester mit Advanced-Level-Zertifikat müssen die Implikationen der einzelnen in diesem Kapitel beschriebenen Mechanismen verstehen, um eine Testarchitektur zu entwerfen, die einen Rahmen für das kontinuierliche Testen der Sicherheit liefert.

5.1.1 Verstehen des Konzepts der Systemhärtung

Moderne Systeme werden immer komplexer. Proportional dazu wächst ihre Angriffsfläche. Schwachstellen sind Folge von Entwurfsfehlern (durch Schwachstellen im Entwurf), Fehlern im Quellcode (durch Schwachstellen in der Konstruktion) oder fehlende Stringenz bei der Konfiguration der Systeme (durch Schwachstellen der Konfiguration).

Systemhärtung ist der schrittweise Prozess der Verkleinerung der Angriffsfläche durch Anwendung einer Sicherheitsrichtlinie und verschiedener Ebenen des Schutzes. Hauptziel ist es dabei, das System abzusichern und die Risiken für die Gefährdung der Sicherheit zu reduzieren.

Je nach Kontext kann das Härten auf verschiedenen Ebenen ansetzen:

1. Härten einer Software- oder Hardwarekomponente
2. Härten eines Produkts bzw. einer Anwendung
3. Härten eines Systems
4. Härten eines Systems von Systemen

Folgende unternehmensbezogene und technische Sicherheitsvorkehrungen sind umzusetzen:

1. Löschen nicht benötigter Software (kann Fehler enthalten)
2. Löschen nicht benötigter Bibliotheken und Entwicklerwerkzeuge (können Fehler enthalten)
3. Löschen nicht benötigter Konten/Anmeldedaten (Angriffsvektoren)
4. Löschen nicht benötigter Anwendungen (können Fehler enthalten) und Netzwerkdienste (Angriffsvektoren)
5. Entfernen nicht benötigter Peripheriegeräte und Hardwareschnittstellen (z. B. USB-Ports, Kartenleser)
6. Sofortiges Patchen von Systemen und Installieren von Software-Updates (z. B. automatische Aktualisierung)
7. Aktualisieren von Konfigurationen
8. Befolgung von Programmierregeln („Security by Construction“, Schwachstellen vermeiden im Programmcode vermeiden)
9. Entsprechendes Konfigurieren des Remote-Anmeldeservers (z. B. rsyslog), damit der Angreifer nur die Protokolldateien auf dem kompromittierten Rechner, nicht aber auf dem Anmeldeserver löschen kann

Folgende Sicherheitsmechanismen sind einzusetzen:

10. Starke Authentifizierung und effizientes Management der Autorisierung (nur die Rechte gewähren, die für die Aktionen der jeweiligen Rolle nötig sind)
11. Verschlüsselung (Kommunikation und Nutzung lokaler Speicher)
12. Firewalls (personen-, system- oder webanwendungsbezogen) und definierte Sicherheitszonen (z. B. Ausführung in Sandbox)
13. Angriffserkennungssysteme
14. Anti-Malware/Anti-Spyware
15. Maskierung von Daten und Anwendungen (z. B. Schutz gegen Reverse-Engineering)

Systemhärtung ist entscheidend für den Schutz sensibler Assets eines Unternehmens. Dennoch müssen die Sicherheitsregeln auf der richtigen Ebene angewendet und im Hinblick auf die Benutzbarkeit des Systems abgewogen werden. Im Extremfall besteht bei dieser Form des Kompromisses die Gefahr, dass die Schutzmaßnahmen deaktiviert werden, weil sie die Produktivität des Unternehmens hemmen.

5.1.2 Testen der Wirksamkeit der Mechanismen der Systemhärtung

Das Testen der Wirksamkeit der Systemhärtungsmechanismen lässt sich auf verschiedene Arten vornehmen. Die Tests hängen vom Charakter des zu härtenden Systems oder der Anwendung, der Schutzwürdigkeit der zu schützenden Assets sowie der ermittelten Gefährdungen ab. Mit dem Härten des Systems wird erreicht, dass der Zugang zum System auf die richtigen Rollen beschränkt wird, nur die benötigten Dienste geöffnet und Software-Updates für Anwendungen überwacht werden. Zum Testen der Wirksamkeit der Systemhärtung müssen daher Tests entworfen werden, mit denen sich ermitteln lässt, ob die Härtung funktioniert und ob sie an den richtigen Orten und auf die richtige Weise wirkt. Ferner ist es wichtig nach Maßnahmen der Systemhärtung zu suchen, die zu restriktiv und angesichts der gegebenen Sicherheitsrisiken übertrieben sind.

Einige Systemhärtungstests können review- oder audit-basiert sein. Es gibt aber auch Tests, die darauf basieren, ob bestimmte Benutzergruppen bestimmte Aktionen durchführen oder auf bestimmte Daten zugreifen können.

Folgende Tests sind denkbar:

1. Prüfung der Konfiguration von Datenbank- und Anwendungsservern im Hinblick auf Änderung der voreingestellten Passwörter
2. Prüfung der Systemkonfiguration auf nicht benötigte Dienste und Netzwerkports
3. Prüfung der Versionen von Komponenten, Bibliotheken und Anwendungen, um zu ermitteln, ob sie veraltet und damit anfällig sind

Sicherheitsorientierte Analyseprogramme können bei der Suche nach Schwachstellen besonders hilfreich sein. Zur Vereinfachung der Aufgaben der Schwachstellenerkennung kann ein Schwachstellenscanner verwendet werden – vor allem bei komplexen Systemen (z. B. eine über mehrere Standorte verteilte Umgebung). Mit statischen Analysewerkzeugen lassen sich Verstöße gegen Programmierregeln ermitteln, die Schwachstellen im Programmcode erzeugen können.

5.2 Authentifizierung und Autorisierung

5.2.1 Der Zusammenhang zwischen Authentifizierung und Autorisierung

Sensible Assets eines Unternehmens (z. B. Kontonummern von Kunden, Entwurf eines neuen Produkts) müssen geschützt werden. Nur autorisierte Personen dürfen Zugriff auf sie haben.

Authentifizierung basiert auf der Verifizierung einer Benutzer-ID sowie der Beantwortung folgender Fragen:

1. Wer ist der Benutzer?
2. Ist der Benutzer der, der er zu sein vorgibt?

Je nach Notwendigkeit des Schutzes vor Angriffen zum Diebstahl von Identitäten oder Passwörtern können verschiedene Implementierungen von Authentifizierungsmechanismen genutzt werden. Das schließt die Erkennung schwacher Passwörter, das Arbeiten mit Einmal-Passwörtern (OTP), die Nutzung von biometrischen Merkmalen (z. B. Fingerabdruck), Softwarezertifikaten, Zertifikaten in Hardware-Security-Tokens und ähnliche Mittel der Authentifizierung ein.

Je nach Architektur eines Systems, Anwendungskontext und den Erfordernissen eines Unternehmens (einfache Verwaltung von Anmeldungen/Passwörtern) können Authentifizierungsmechanismen die lokale Authentifizierung, Serverauthentifizierung, Netzwerkauthentifizierung, SSO (Single Sign-On) und ähnliche Mittel einschließen.

Autorisierung wird für Folgendes genutzt:

1. Um zu prüfen, ob der authentifizierte Benutzer die Rechte für die Ausführung einer Aktion hat (z. B. Anmeldung bei einem Server ohne Berechtigung Daten zu ändern bzw. Autorisierung zur Nutzung eines FTP-Servers, aber nur mit Zugriff auf dedizierte Ressourcen)
2. Um zu ermitteln, welche Zugangsebene und Berechtigungen den einzelnen Systemressourcen zugestanden werden sollen

Es gibt einen engen Zusammenhang zwischen Authentifizierung und Autorisierung: Es gilt der Grundsatz, dass ein nicht authentifizierter Benutzer auf einem System keine Rechte oder eingeschränkte Rechte hat (nicht autorisiert ist, sensible Daten zu ändern). Bei der Website eines Händlers beispielsweise kann ein nicht autorisierter Benutzer die aufgelisteten Produkte einsehen. Vor dem Kauf eines Artikels muss er jedoch ein Benutzerkonto anlegen. Der authentifizierte Benutzer kann Artikel kaufen, jedoch keine administrativen Funktionen ausführen.

5.2.2 Testen der Wirksamkeit von Authentifizierungs- und Autorisierungsmechanismen

Ziel von Angreifern ist es, Passwörter zu stehlen oder die Authentifizierungssysteme zu umgehen, um nicht autorisierte Aktionen durchzuführen. In der Regel nutzen sie dazu verschiedene Schwachstellen: Programmierfehler (fehlende Filterung von Eingaben), alte anfällige Versionen von Bibliotheken, Systemkonfigurationsfehler (Beibehaltung voreingestellter Passwörter, Standardrechte) und schwache Passwörter (das am häufigsten verwendete Passwort ist z. B. „123456“).

Ein weiteres Problem: Es gibt zwar zu befolgende Passwortregeln, der Benutzer lässt jedoch die nötige Sorgfalt bei der Geheimhaltung des Passworts vermissen. Damit werden die Passwortregeln ausgehebelt. Zudem müssen die Passwortregeln aktuelle empfohlene Praktiken für die Passwortdefinition widerspiegeln. Empfehlungen dieser Art finden

sich z. B. in den „Password Construction Guidelines“ des SANS Institute [SANS2] oder in den Regelungen des Passwortgebrauchs des IT-Grundschutz [BSIITG].

Folgende Tests von Authentifizierungs- und Autorisierungsmechanismen gibt es u.a.:

1. Brute-Force- und Wörterbuchangriffe zur Ermittlung von Benutzerpasswörtern. Im ersten Schritte ließe sich „123456“, „111111“, das Geburtsdatum, den Namen des Haustiers usw. ausprobieren.
2. Ausnutzung fehlender Filterung von Eingaben, z. B. zum Einschleusen von SQL-Befehlen, um ohne bekannte Kombination aus Anmeldename/Passwort authentifiziert zu werden.
3. Eingabe nicht autorisierter URIs (../../ in einem FTP-Konto) oder URLs (Site-Adressen/Admin), um Zugriff auf sensible Daten zu erlangen.

Ein weiteres Beispiel ist die Ausnutzung einer Schwachstelle im Zielsystem (weil es vielleicht nicht aktualisiert wurde), um die Kontrolle über das System zu übernehmen und die eigenen Zugriffsrechte zu erweitern (Privilege Escalation).

5.3 Verschlüsselung

5.3.1 Verstehen des Konzepts der Verschlüsselung

Um die Offenlegung sensibler Daten zu verhindern, auch wenn am Speicherort oder beim Austausch zwischen Client und Server der Zugriff auf die Daten möglich ist, kann ein Verschlüsselungsmechanismus genutzt werden. Hashing und Salting sind Methoden, die bei der Verschlüsselung genutzt werden.

Verschlüsselung ist der Prozess der Umwandlung von Klartextdaten in verschlüsselte Daten mittels eines kryptographischen Algorithmus und geheimer Schlüssel, so dass nur autorisierte Personen auf die Klartextdaten zugreifen können, indem sie sie entschlüsseln. Die geheimen Schlüssel sind nur autorisierten Benutzern bekannt und werden unter ihnen ausgetauscht. Ziel ist es, die Daten stark genug zu verschlüsseln, dass ein Angreifer, dem es gelungen ist, verschlüsselte Daten zu stehlen, diese nicht entschlüsseln kann. Die Verwendung von Verschlüsselungsalgorithmen hilft, die Vertraulichkeit, Integrität und Verfügbarkeit sensibler Assets zu gewährleisten und deren Manipulation zu verhindern.

Zum Schutz von Informationen können Verschlüsselungsprotokolle genutzt werden:

1. Gespeichert in einem System, z. B. verschlüsselte Passwörter in einer Datenbank, logisch verschlüsselte Laufwerke, komplett verschlüsselte Festplatten
2. Während des Datentransfers, z. B. verschlüsselte E-Mails, Datenübertragungsprotokolle (SSL, TLS)

Die grundlegendsten und bekanntesten Verschlüsselungsprotokolle sind:

1. Symmetrische Verschlüsselung: Nutzung eines gemeinsamen geheimen Schlüssels
2. Asymmetrische Verschlüsselung: Nutzung eines privaten und eines öffentlichen Schlüssels

5.3.2 Testen der Wirksamkeit gängiger Verschlüsselungsmechanismen

Einige Verschlüsselungsmechanismen sind bekanntermaßen schwach – primär bedingt durch die Kürze der geheimen Schlüssel oder statische Schlüssel. Andere Mechanismen sind anfällig, weil sie entweder nicht unter Einhaltung der Best Practices implementiert wurden oder Fehler im Programmcode enthalten (z. B. Pufferüberlauf).

Tests von Verschlüsselungsmechanismen müssen Folgendes einschließen:

1. Tests auf Schwachstellen im Entwurf:
2. Prüfung auf Verwendung richtiger Verfahren bzw. Modi bei der symmetrischen Verschlüsselung
3. Prüfung, ob Verschlüsselungsschlüssel zu kurz sind (seit 2015 gelten RSA-Schlüssel mit weniger als 2048 Bit z. B. als unsicher)
4. Validierung der Gültigkeit von Zertifikaten sowie der Fähigkeit, einen Alarm auszulösen, wenn ein selbst-signiertes Zertifikat verwendet wird (zur Verhinderung von Man-In-The-Middle-Angriffen kann SSL genutzt werden)
5. Replay-Angriffe (z. B. Angriffe auf WEP-Protokolle [Wired Equivalent Privacy])
6. Angriffe auf Verschlüsselungsprotokolle zur Ermittlung ihrer Stärke [Bittau]
7. Tests auf konstruktionsbedingte Schwachstellen:
8. Code-Reviews (z. B. zur Verifizierung, dass die Standardfunktion „random()“ nicht für die Erzeugung von Zufallszahlen (Seed) genutzt wird, weil der Zufallsalgorithmus relativ einfach zu knacken ist)
9. Fuzz-Testen zur Ausnutzung unerwarteten Systemverhaltens
10. Timing-Angriffe (Analyse der benötigten Zeit für die Ausführung von Verschlüsselungsalgorithmen)
11. Analyse der Leistungsaufnahme für Hardware (genutzt zur Identifikation von Seitenkanalangriffen für verschlüsselte Hardwaregeräte)
12. Tests auf Schwachstellen in der Konfiguration:
13. Prüfung der Konfiguration von Verschlüsselungsprotokollen (z. B. serverseitige TLS-Konfiguration (Transport Layer Sicherheit), clientseitige autorisierte Protokolle, basierend auf TLS-Konfigurationsleitfaden für Administratoren)
14. Reihenfolge und Auswahl der TLS-Verschlüsselungsalgorithmen auf der Serverseite, um zu prüfen, ob es Mittel gibt, die Verwendung eines schwächeren Verschlüsselungsalgorithmus neu zu verhandeln bzw. zu erzwingen.
15. Tests zur Prüfung der Alterung von Verschlüsselungsmechanismen, die schwach und überwindungsanfällig geworden sind

5.4 Firewalls und Netzwerkzonen

Verstehen des Konzepts von Firewalls

Laut [Chapman 2000] ist „eine Firewall eine Komponente oder eine Reihe von Komponenten, die den Zugang zwischen einem geschützten Netzwerk und dem Internet, oder zwischen anderen Netzen beschränken“. Eine Firewall implementiert eine Sicherheitsrichtlinie, die auf der Definition autorisierter und verbotener Datenübertragungsvorgänge basiert, und setzt diese durch. Eine Firewall kann host-basiert (Software, die auf einem Host läuft und Eingaben/Ausgaben von Anwendungen überwacht) oder netzwerkbasiert (Software, die den Verkehr zwischen Netzwerken überwacht) sein.

Hauptaufgabe einer Firewall ist es, den Verkehr zwischen Netzwerkzonen mit einem unterschiedlichen Grad an Vertrauenswürdigkeit, zu kontrollieren, indem sie die Daten filtert, die sich im Netzwerk bewegen. Auf diese Weise wird bösartiger Verkehr aus einer nicht vertrauenswürdigen Zone erkannt und blockiert.

Eine Netzwerkzone ist ein identifiziertes Subnetzwerk mit einem definierten Grad von Vertrauenswürdigkeit:

1. Internet/öffentliche Zone, die nicht als vertrauenswürdig gelten
2. Verschiedene Sicherheitszonen, die auch als entmilitarisierte Zonen oder DMZs bezeichnet werden; mit verschiedenen Graden an Vertrauenswürdigkeit
3. Private/interne Netzwerke, die als am vertrauenswürdigsten gelten

Die Netzwerkzonen sind Bestandteile der Konfiguration der Firewall: Sie dienen dem Definieren autorisierter Datenströme zwischen den verschiedenen Netzwerken. Sämtlicher verbotener Verkehr wird blockiert.

Eine Firewall filtert den Datenverkehr in der Regel anhand folgender Aspekte:

1. Quell- und Zieladressen sowie Quell- und Zielprotokolle (Ethernet- oder IP-Adressen, TCP/UDP-Ports usw.)
2. Protokolloptionen (Fragmentierung, TTL usw.)
3. Größe der Daten

WAFs (Web Application Firewalls) filtern den Datenverkehr auch anhand folgender Aspekte:

1. Benutzer, die die Endpunkte der Verbindungen darstellen
2. Datenfilterung (z. B. unter Verwendung von Musterbeschreibungen)

5.4.1 Testen der Wirksamkeit von Firewalls

Aufgrund der Anzahl der Protokolle, ihrer verschiedenen Optionen und der Komplexität der zu schützenden Netzwerke ist es schwierig, eine Firewall effizient zu konfigurieren. Tests der Wirksamkeit von Firewalls müssen Folgendes einschließen:

1. Port-Scans, um die richtige Implementierung der Sicherheitsrichtlinie zu prüfen
2. Verwendung fehlerhafter Netzwerkpakete und Netzwerk-Fuzz-Testen zur Ausnutzung eines unerwarteten Systemverhaltens (z. B. Dienstblockaden)
3. Fragmentierungsangriffe zur Umgehung von Filterfunktionen mit dem Ziel, einen Angriff hinter der Firewall fortzusetzen

Ein weiteres Beispiel für Tests mit Zielrichtung WAF ist die Kodierung und Komprimierung von Daten oder ihre Maskierung, um bösartige Daten, die den Angriff tragen, zu verbergen.

5.5 Angriffserkennung

5.5.1 Verstehen des Konzepts von Werkzeugen zur Angriffserkennung

Die Zahl der Angriffe auf Computersysteme steigt von Jahr zu Jahr. Angriffstechniken entwickeln sich schnell weiter, und kein System ist zu 100 % sicher.

Ein Angriffserkennungssystem (IDS: Intrusion Detection System) ist ein System (eigenständige(s) Gerät/Anwendung), das Aktivitäten auf verschiedenen Schichten (von der Netzwerk- bis zur Anwendungsschicht, 7 Schichten des OSI-Modells) überwacht, um Verstöße gegen die Sicherheitsrichtlinie zu erkennen. Bei Abweichungen vom Normverhalten werden Alarme ausgelöst, die sich zur Einleitung weiterer Maßnahmen nutzen lassen (wie z. B. Verkehrsblockierung, virtuelles Patching).

Bezüglich der IDS-Standardisierung beschreibt das Intrusion Detection Exchange Format der Internet Engineering Task Force ein Entwurfsmodell für ein IDS, das auf zwei Sicherheitsmodellen basiert:

1. Negatives Sicherheitsmodell (signaturbasierte Erkennung oder Black-List-Erkennung): Es gilt die Regel, dass „alles erlaubt ist, was nicht ausdrücklich verboten ist“. Die Angriffserkennung nutzt als Basis eine Liste mit bekannten Angriffen oder Mustern.
2. Positives Sicherheitsmodell (verhaltensbasierte Erkennung oder White-List-Erkennung): Es gilt die Regel, dass „alles verboten ist, was nicht ausdrücklich erlaubt ist“. Die Angriffserkennung basiert auf der Spezifikation des Verhaltens des zu schützenden Systems, z. B. die Merkmale einer Eingabe in einer als regulärer Ausdruck beschriebenen Form. Ein Angriff wird erkannt, wenn das Verhalten vom normalen oder erwarteten Verhalten des Systems abweicht. Als vertrauenswürdig verifizierter Verkehr kann genutzt werden, um die Spezifikation zu erzeugen.

Ein IDS unterscheidet sich dahingehend von einer Firewall, als dass eine Firewall den Verkehr nach außen überwacht, um Angriffe zu stoppen, während das IDS verdächtige Vorgänge analysiert und bei Bestätigung einen Alarm auslöst.

5.5.2 Testen der Wirksamkeit von Werkzeugen der Angriffserkennung

Die signaturbasierte Erkennung lässt sich leicht umgehen, weil nur bekannte Angriffsmuster erkannt werden. Tests können folgende Umgehungstechniken einschließen:

1. Zeichenkodierung oder Modifikation von Daten (z. B. Hinzufügen eines Leerzeichens, Zeilenumbruchs usw.)
2. IP-Fragmentierung, TCP-Segmentierung
3. Verschlüsselung, Maskierung
4. URL-Kodierung

Die verhaltensbasierte Erkennung erzeugt viele falsch-positive und falsch-negative Ergebnisse. Ein falsch-negatives Ergebnis ist jeder Alarm, der hätte ausgelöst werden müssen, aber nicht ausgelöst wurde. Falsch-negative Ergebnisse können eintreten, wenn ein neues Angriffsmuster entwickelt wurde, das ein signaturbasiertes IDS nicht kennt, oder eine Regel so geschrieben wurde, dass bestimmte Angriffe nicht als solche erkannt werden. Außerdem ist die Genauigkeit eines Erkennungsverfahrens zu berücksichtigen.

5.6 Malware-Scans (Schadprogramm-Scans)

5.6.1 Verstehen des Konzepts der Malware-Scanner

Bösartiger Programmcode kann Server und die Computer der Endbenutzer so kompromittieren, dass seine Urheber die erwarteten Zugriffsrechte und die gewünschten sensiblen Daten erhalten. Der bösartige Programmcode wird mit Hilfe verschiedener Mittel wie E-Mails mit bösartigen Anhängen, gefälschten URLs, clientseitiger Code-Ausführung usw. im Zielsystem platziert.

Ein Malware-Scanner (Schadprogramm-Scanner) ist ein Programm, das bösartigen Programmcode aus verschiedenen Quellen erkennt, analysiert und entfernt sowie verschiedene Erkennungsziele hat: Malware, Phishing und Pharming.

Dazu bedient er sich in erster Linie einer signaturbasierten Strategie. Das Prinzip besteht darin, in einer Datenbank nach bekannten Datenmustern zu suchen, die verdächtigen Programmcode signalisieren. Neue Malware oder Malware, deren Signatur in der Datenbank nicht erfasst ist, werden jedoch nicht erkannt und können das Opfer infizieren. Häufig ist in diesen Scannern eine Heuristik eingebettet, die auch leichte Variationen von bekannten, bösartigen Mustern erkennt und dazu beiträgt, dieses Problem zu bekämpfen.

5.6.2 Testen der Wirksamkeit von Malware-Scannern

Entwickler von Malware und Hintertüren nutzen verschiedene Techniken, um ihren Programmcode vor Reverse-Engineering und Erkennung durch Malware-Scanner zu schützen. Dazu gehören:

1. Ausnutzung von Systembibliotheksfunktionen (z. B. FindWindow, was genutzt werden kann, um einen Malware-Scanner zu schließen)
2. Zeichenfolgenmaskierung zur Verhinderung der Erkennung des Verhaltens von bösartigem Programmcode (z. B. durch Verschlüsselung). Ein Beispiel dafür wäre die Einbettung von Java Script in ein PDF-Dokument. Ein weiteres Beispiel ist die Nutzung von Komprimierungswerkzeugen wie UPX (Ultimate Packer for eXecutables).

3. Funktionen für das dynamische Laden von Bibliotheken (z. B. zur Limitierung der Analyse von bösartigem Programmcode)
4. Automatische Aktualisierung von Anwendungen (z. B. Skype-Trojaner)

Malware (Schadprogramme) können auch andere Hardwareressourcen wie die GPU (Grafikprozessor) nutzen, um bösartigen Programmcode zu entpacken und im Speicher abzulegen, damit er vom Prozessor ausgeführt wird. In diesem Fall lässt sich das Malware vor seiner Ausführung nicht analysieren.

Aus der Perspektive der funktionalen Tests kann ein Werkzeug wie Eicar [EICAR] (Malware-Testdatei) verwendet werden, um die Wirksamkeit von Malware-Scannern zu testen, ohne dazu echten bösartigen Programmcode entwickeln zu müssen.

Eine wichtige Überlegung bei der Implementierung einer neuen oder dem Upgraden einer bestehenden Anti-Malware-Anwendung ist das Testen der Implementierung auf einer repräsentativen Plattform vor ihrer Bereitstellung im gesamten Unternehmen. Es kommt vor, dass Anti-Malware-Software fälschlicherweise legitime Betriebssystemdateien als Malware identifiziert, unter Quarantäne stellt und damit die gesamte Rechenkapazität des Unternehmens lahmlegt.

5.7 Datenmaskierung

5.7.1 Verstehen des Konzepts der Datenmaskierung

Datenmaskierung ist ein Mechanismus, der bewirkt, dass Daten und Quellcode für Menschen nicht lesbar sind.

Diese Technik dient vorrangig dem Schutz sensibler Daten gegen:

1. Kopieren, Umgehung von Lizenzschutzmechanismen
2. Reverse-Engineering, um Programmcode untersuchen zu können und auf Schwachstellen abzuklopfen

Datenmaskierung kann auch dazu verwendet werden, den Mitarbeitern eines Unternehmens (Support-Mitarbeiter, Funktionstester usw.) die Arbeit mit nicht sensiblen Daten zu ermöglichen, während die eigentlich sensiblen Informationen nicht lesbar sind. Die Datenmaskierung wird mitunter auch als „Datenanonymisierung“ bezeichnet, in diesen Fällen werden dadurch personenbezogene Daten anonym gehalten.

Maskierung kann auch zum Schutz von Quellcode gegen einfaches Kopieren und Einfügen (z. B. zum Schutz eines neuen innovativen Algorithmus) und spätere Wiederverwendung nach Reverse-Engineering genutzt werden.

Mitunter müssen Entwickler ihren Programmcode optimieren, um ihn effizienter zu machen. Das kann in maskiertem Quellcode resultieren (z. B. weil Teile des Codes in Assembler-Sprache kodiert werden). Einige Angriffe auf Webanwendungsebene bestehen aus dem Einschleusen (Injection) von Skripts. Um damit Erfolg zu haben, müssen Angreifer die Struktur der Website und der HTML-Seiten kennen. Maskierung kann dazu beitragen, sensible und kritische HTML-Seiten zu schützen (z. B. Verbindungs- und Verwaltungsseiten).

Es können verschiedene Maskierungstechniken verwendet werden: base64-Kodierung, XORing, Zufalls-Umbenennungsfunktionen, Überschreiben von Methoden, Löschen von Tabulator-, Return- und Leerzeichen, Shuffling usw. Auch die Verschlüsselung ist eine Maskierungstechnik. Allerdings mit Einschränkungen, weil verschlüsselte Daten nur für die Besitzer gültiger Schlüssel lesbar bleiben.

Hinweis: Die Datenmaskierung wird von Angreifern häufig verwendet, um ihren böswilligen Programmcode und ihre Angriffe zu verschleiern.

5.7.2 Testen der Wirksamkeit von Datenmaskierungsverfahren

Eine strikte Konfigurationskontrolle zwischen maskierten Daten und den für die Maskierung genutzten Schlüsseln ist nötig, um sicherzustellen, dass die richtigen Versionen der Schlüssel verwendet werden. Andernfalls können die Daten nicht zur Verwendung demaskiert werden.

Weil bei einigen Tests auch private Daten involviert sein können, ist für Testzwecke u.U. eine Datenmaskierung nötig, um Produktionsdaten, die in einer Systemtestumgebung verwendet werden, zu anonymisieren. Sensible Daten wie Benutzerdaten, die von einem Gesundheitsdatensystem genutzt werden, dürfen für Tester nicht offen einsehbar sein. Folgender Test ist denkbar:

1. Brute-Force- und Wörterbuchangriffe als Versuch, aus maskierten Daten Klardaten zu machen

Folgende Tests zur Verifizierung der Maskierung können u.a. durchgeführt werden:

1. Reverse-Engineering von Java-Byte-Code (z. B. Neuerzeugung von Java-Quellcode mittels Java Decompiler) oder .Net-Programmen (z. B. Erzeugung von .Net-Quellcode mit .NET Reflector)
2. Brute-Force-Angriffe, weil bestimmte Maskierungsmechanismen anfällig sind (z. B. durch Verwendung von unXOR [Chopitea])

Theoretisch kann binärer Programmcode sich nicht selbst gegen Demaskierung schützen, weil stets Debugging genutzt werden kann. Es gibt zwar Werkzeuge für den Schutz von binärem Programmcode gegen Dekompilierung, dennoch bleiben Risiken und Beschränkungen, was den Schutz proprietärer Informationen angeht, die durch binärem Programmcode abgebildet werden.

5.8 Schulungen

5.8.1 Die Bedeutung von Sicherheitsschulungen

Das schwächste Glied in der Sicherheitskette ist häufig der Mensch. Daher bedarf es der konsequenten und kontinuierlichen Schulung, um die Bedeutung folgender etablierter Sicherheitsrichtlinien in Erinnerung zu rufen und zu betonen, warum Richtlinien benötigt werden. Diese Schulung muss im Softwareentwicklungslebenszyklus-Prozess erfolgen und entsprechend aktualisiert werden, wenn neue Richtlinien hinzukommen oder neue Gefährdungen entstehen. Thema der Schulungen muss immer auch die Erkennung von Angriffen mittels sozialer Manipulation und Gefährdung durch Betriebsangehörige sein.

5.8.2 Testen der Wirksamkeit von Sicherheitsschulungen

Thema einer Sicherheitsschulung könnte sein, wie wichtig sicherere Passwörter und deren Geheimhaltung sind.

Folgende Tests sind denkbar:

1. Social Engineering, um Benutzer bei einem fingierten Telefongespräch mit einem Support-Mitarbeiter dazu zu bringen, ihr Passwort zu verraten
2. Suche auf Schreibtischen nach Haftnotizen mit notierten Passwörtern (vor allem unter Tastaturen)

3. Nutzung von Passwort-Audit-Werkzeugen zur Ermittlung schwacher Passwörter. Dabei besteht allerdings die Gefahr, dass Passwörter für den Tester sichtbar sind.

Ein weiteres Beispiel: Ein Entwickler legt ein Feld zur Dateneingabe fälschlicherweise so an, dass dort SQL-Befehle eingegeben werden können. Dadurch kann ein Sicherheitstester einen SQL-Befehl einschleusen und den Inhalt einer Kundendatenbank einsehen. Das ist Zeichen dafür, dass der Entwickler Schulungsbedarf zu sicheren Programmierpraktiken hat. Gut wäre es auch, die Programmierpraktiken anderer Entwickler zu prüfen, um zu sehen, ob diese Praktik weit verbreitet ist und eine allgemeine Initiative zur Prozessoptimierung angebracht ist.

Ein drittes Beispiel: Ein Tester versucht, unbefugt Zugang zu einem Büro zu erlangen und Unterlagen einzusehen, die dort offen herumliegen.

6 Menschliche Faktoren beim IT-Sicherheitstest – 105 min

Schlüsselbegriffe

Angreifer, Bot-Netz, Computer-Forensik, Hacker, Erkundung, Skriptkiddy

Lernziele für das Thema „Menschliche Faktoren beim IT-Sicherheitstest“

6.1 Verstehen der Angreifer-Motivation

AS-6.1.1 (K2) Erläutern können, wie menschliches Verhalten zu Sicherheitsrisiken führen kann und inwieweit es die Wirksamkeit von Sicherheitstests beeinträchtigt

AS-6.1.2 (K3) Für ein gegebenes Szenario Wege identifizieren zu können, über die ein Angreifer wichtige Informationen über ein Ziel erlangen könnte sowie Maßnahmen zum Schutz der Umgebung ergreifen können

AS-6.1.3 (K2) Die allgemeinen Motive und Quellen für die Durchführung von Angriffen auf Computersysteme erläutern können

AS-6.1.4 (K4) Ein Angriffsszenario (Angriff durchgeführt und entdeckt) analysieren sowie mögliche Quellen und Motive für den Angriff ermitteln können

6.2 Social Engineering

AS-6.2.1 (K2) Erläutern können, wie Sicherheitsvorkehrungen durch Social Engineering umgangen werden können

6.3 Sicherheitsbewusstsein

AS-6.3.1 (K2) Die Bedeutung des Sicherheitsbewusstseins für eine Organisation verstehen

AS-6.3.2 (K3) Ausgehend von bestimmten Testergebnisse entsprechende Maßnahmen zur Erhöhung des Sicherheitsbewusstseins einleiten können

6.1 Verstehen der Angreifer

Im Hinblick auf die Informationssicherheit stellt der Mensch sowohl die größte Gefahr als auch den schwächsten Punkt in der Gefahrenabwehr dar.

Sicherheitsrelevante Angriffe werden von Menschen mit unterschiedlichen Motiven und Fähigkeiten durchgeführt. Darüber hinaus sind es (meist) Menschen, die Sicherheitsangriffe überhaupt ermöglichen. Nur die Sicherheitstechnik und deren Implementierung zu kennen reicht nicht, um sich wirksam gegen Angriffe zu verteidigen. Man muss auch die Mentalität, die Motive und die Methoden der böswilligen Angreifer kennen und wissen, wo die menschgemachten Schwächen in der Abwehr liegen.

6.1.1 Der Einfluss des menschlichen Verhaltens auf Sicherheitsrisiken

Die wichtigste Phase bei jedem Angriff ist die Phase der Informationsbeschaffung (Erkundung), in der der Angreifer versucht, Informationen zum Ziel zu sammeln. Alle Informationen über ein Unternehmen, verwendete Systeme usw., die (mitunter unwissentlich) öffentlich einsehbar und im Internet verfügbar sind, werden gefunden und können/werden bei einem Angriff Verwendung finden. Nicht „ob“, sondern „wann“ ist die Frage. Neben den vom Unternehmen offiziell veröffentlichten Informationen, legen auch Mitarbeiter über ihre sozialen Netzwerke unternehmensbezogene Informationen offen. Umfang und Inhalt dieser Informationen ändern sich ständig. Häufig liefern sie Angreifern wichtige Erkenntnisse.

Angreifer nutzen beim Angriff auf ein System keine Sicherheitsrichtlinie und keine vordefinierten Verfahren. Auf der Basis der Informationen, die sie gesammelt haben, entscheiden sie über ihre Strategie. Für jeden Angriff bringen sie ihr Wissen auf den neuesten Stand, indem sie selektive Suchen durchführen bereits bekannte IP-Adressen ‚besuchen‘.

Wenn die Sicherheitsrichtlinie für ein Unternehmen formuliert wird, geschieht dies in der Regel auf der Basis der gegebenen Umstände und verfügbaren Fakten. Mitunter schließt das nicht alle öffentlich zugänglichen Informationen ein. Falls doch, ändern sich diese wahrscheinlich jedoch mit der Zeit. Sicherheitstest, die bei ihrer Entwicklung wirksam waren, decken vielleicht nicht mehr genug ab, wenn sich veröffentlichte Informationen ändern.

6.1.2 Die Mentalität von Angreifern verstehen

Während der Erkundungsphase versucht der Angreifer, sich mit passiven und/oder aktiven Mitteln alle möglichen Informationen über das Ziel zu besorgen. Die meiste IT-Ausrüstung mit Schnittstellen zu öffentlichen Netzwerken hinterlässt in diesen Netzwerken sog. Fußabdrücke (footprints). Diese können gefunden werden und werden auch gefunden. Die ersten Quellen für die Suche nach Informationen zum Ziel sind Google (einschließlich Google Earth und Street View) oder andere Suchmaschinen, Shodan [Web-5], Facebook, LinkedIn und andere soziale Netzwerke. IP-Adressen, Webseiten, Telefonnummern, Namen und E-Mail-Adressstrukturen, Betriebssystem und Anwendungen liefern einem Angreifer hilfreiche Informationen.

Google-Suchmaschinen können u.a. dazu verwendet werden, um spezifische Informationen über ein Ziel zu finden. In der Google Hacking Database [Web-4] finden sich hunderte entsprechender Abfragen. Shodan [Web-5] ist ein weiteres Werkzeug für die Suche nach spezifischen Informationen, z. B. danach, welche Unternehmen in einer bestimmten Gegend einen Apache-Server mit einer anfälligen Version nutzen.

Die meisten dieser Informationen lassen sich passiv ermitteln, ohne dazu Verbindung zum Zielsystem aufnehmen zu müssen. Weitere Werkzeuge:

1. Whois [Web-13]
2. Ripe-Datenbank (europäische IP-Netzwerke) [Web-12]
3. DNS-Suchläufe [Web-25]

Bei aktiven Erkundungstechniken nutzt der Angreifer Werkzeuge zur Entdeckung von Hosts, offenen Ports, Betriebssystemen und Anwendungen durch direkten Zugriff auf das System. Hier kommen folgende Methoden und Werkzeuge zum Einsatz:

1. Pinging – Fping [Web-15], Hping [Web-19]
2. TCP/UDP-Scan – Nmap [Web-20], Zenmap [Web-21]
3. Betriebssystemerkennung – Nmap [Web-20], Xprobe2 [Web-22]
4. Service Fingerprinting (Nmap bietet Funktionen zur zusätzlichen Ermittlung des Typs und der Version des Dienstes, der auf dem entdeckten offenen Port läuft. Dies erfolgt durch Vergleich des „Fingerabdrucks“ des entdeckten Dienstes mit den Fingerabdrücken in der Nmap-eigenen Datenbank.)

Weil das Hacken in ein System in den meisten, wenn nicht allen Ländern gesetzlich verboten ist, muss der Hacker nach dem Hack sämtliche hinterlassene Spuren verwischen. Das kann auch dazu dienen, seinen unerkannten Aufenthalt im System zu verlängern, die Nutzung des Systems später fortzusetzen und das gehackte System oder Systemnetzwerk für den Angriff auf andere Systeme zu nutzen (Botnetze). Dafür kann der Angreifer Werkzeuge wie NetCat [Web-14] oder Websites wie IP TRacer [Web-7] sowie tunneln oder die Änderung von Protokolldateien nutzen.

Weitere Methoden und Werkzeuge für das Verwischen der Spuren sind Tarnkappen-Werkzeuge, Rootkits und Datei-Streaming. Alle oder die meisten der hier erwähnten Werkzeuge sind über das Internet erhältlich. Wenn Sie die neueste Version von Kali Linux [Web-17] herunterladen und auf der OWASP-Site [OWASP1] suchen, können Sie auf viele dieser Werkzeuge zugreifen.

6.1.3 Allgemeine Motive und Quellen für Angriffe auf Computersysteme

Viele Angriffe und Einbrüche in Informationssysteme erfolgen von innerhalb des Unternehmens. Systembenutzer mit bösen Absichten (interne Hacker oder Betriebsangehörige/Mitarbeiter) versuchen als autorisierte Netzwerkbenutzer, das System zu kompromittieren. Meist ist Rache das Motiv, in jüngster Zeit geht der Trend aber eher in Richtung Wirtschaftsspionage oder Diebstahl.

Externe Hacker sind nur für einen geringen Teil der Angriffe verantwortlich. Eines der ersten Motive für das Hacken von Informationssystemen war – und ist es heute noch – schlicht und ergreifend Neugier. Informationen von großen Unternehmen oder Organisationen zu besitzen – im Wissen darum, dass andere diese nicht haben – ist ein starker Antrieb (Prestige). Weitere Motive sind Ruhmsucht, Herausforderung, Langeweile und Rache, wobei letztere als die gefährlichste Form gilt (größte Motivation).

Häufig werden Angreifer nach ihren Motiven und Fähigkeiten kategorisiert. Das untere Ende des Spektrums bilden die so genannten „Skriptkiddies“, die einfach nur von anderen entwickelte Angriffsmittel nutzen. Das obere Ende machen professionelle Organisationen und Einzelpersonen aus (staatliche Hacker, Hacktivisten). Als Hacktivismus wird der Angriff auf Systeme aus vorrangig politischen, aber auch wirtschaftlichen oder anderen demografischen Gründen bezeichnet.

Die Motive reichen vom Herumprobieren aus Spaß bis hin zum kompletten Lahmlegen eines Systems oder Unternehmens aus vielerlei Gründen (politische, ideologische, wirtschaftliche Gründe, Form der Kriegsführung, Terrorismus).

Die Hacking-Fähigkeiten reichen von Einzelpersonen mit gewissen System- und Netzwerkkenntnissen, die mit einem einfachen PC arbeiten, bis hin zu speziell ausgebildeten Profis mit Zugang zu Laboren, Proxy-Netzwerken und sonstigem technischen Equipment. Eine Vorstellung von den potenziellen Angreifern zu haben, hilft einem Unternehmen, den notwendigen Schutz zu implementieren. Außerdem dient dieses Wissen als Leitfaden für die Sicherheitsteststrategie.

6.1.4 Angriffsszenarien und -Motive

Ein Sicherheitsstörfall ist als sicherheitsrelevantes Systemereignis definiert, bei dem die Sicherheitsrichtlinie des Systems verletzt wird. [RFC2828]

Herauszufinden, was passiert ist und wer für den sicherheitsbezogenen Störfall verantwortlich ist, ist das Ziel der Computer-Forensik [Web-8, BSIITG], die vorrangig nach digitalen Spuren von Angriffen sucht.

Der Spurensicherungsprozess umfasst drei Phasen:

1. Erfassung und Authentifizierung
2. Analyse
3. Berichterstellung

6.1.4.1 Erfassung und Authentifizierung

Der Fehler- und Abweichungsmanagementprozess des Unternehmens muss das System in den ursprünglichen Zustand (vor dem Angriff) zurückversetzen, nachdem die Beweise gesichert wurden. Es beginnt, wenn der Systemadministrator vom IDS oder anderen Überwachungswerkzeugen gewarnt wird. Weitere typische Symptome von Sicherheitsstörfällen sind:

1. Verdächtige Protokolleinträge
2. Unerklärliche Benutzerkonten
3. Modifizierte Dateien/Ordner
4. Ausführung ungewöhnlicher Dienste
5. Ungewöhnliches Systemverhalten
6. Erfolglose Anmeldeversuche

Nach der Warnung ist folgender Prozess abzuarbeiten:

1. Fertigen Sie einen Schnappschuss oder eine Kopie des zu untersuchenden Systems an, um alle nötigen Beweise zu sichern.
2. Nach der Authentifizierung der Beweise (eine echte und vollständige Kopie) erzeugen Sie eine Kopie und legen diese an einem sicheren Ort ab.
3. Analysieren Sie die Beweise.
4. Nach Abschluss des forensischen Prozesses beseitigen Sie die Ursache des Störfalls (Ausmerzung).

5. Das System wird in seinen Normalzustand zurückversetzt (Wiederherstellung).

Während dieser Schritte werden sämtliche Schwachstellen mit Patches oder durch Installation neuer Software beseitigt. Bei der Protokollierung der Ergebnisse in einem Bericht, müssen der genutzte Prozess und die dabei genutzten Werkzeuge beschrieben werden.

6.1.4.2 Analyse

Nach Hacking-Versuchen lässt sich durch Prüfung der Protokolldateien des Systems und aktiver Netzwerkverbindungen u.U. der Ursprung der Angriffe ermitteln. Es müssen Kopien von allen Protokolldateien angelegt und die Prozessstatus-Informationen erfasst werden. Während eines laufenden Angriffs kann es sinnvoll sein Systeminformationen zu erfassen, die in Zusammenhang mit dem/den Angreifer(n) stehen, bevor man sie aussperrt.

Jeder Angriff über das Internet lässt sich auf die Ursprungs-IP-Adresse zurückverfolgen, unabhängig davon ob E-Mail oder Internetverbindungen genutzt wurden. Das ist nur eine Frage der Zeit, des Geldes und des Aufwands sowie der Abwägung der damit einhergehenden Kosten. Die meisten Angreifer nutzen Proxys oder Proxy-Ketten, das Tor-Netzwerk [Web-9] oder andere kostenlose Anonymisierungsdienste, um ihre reale IP-Adresse zu verschleiern. Je mehr Proxys Angreifer nutzen, desto länger dauert die Ermittlung der Ursprungsadresse. Diese Nachforschungen können auch durch Gesetze behindert werden, die an den jeweiligen Standorten der Proxy-Server gelten.

Die Entdeckung von Eindringlingen und die Verfolgung einer IP-Adresse zurück zu ihrem Ursprung kann mit Werkzeugen wie Netstat (Windows) [Web-10], Tracert [Web-11] und der IP Tracer-Website [Web-7] erfolgen. Netstat zeigt die Verbindungen zu einem Rechner, Ports und laufende Dienste an. Mit diesem Werkzeug kann nach merkwürdigen oder unbekannten IP-Adressen oder Portnummern gesucht werden. Hinweis: Auch im Microsoft Windows-Betriebssystem gibt es ein tracert-Dienstprogramm (in Linux und OS/X heißt es „traceroute“), die oben genannten, webbasierten Dienste sind jedoch unabhängig von diesen Betriebssystem-Dienstprogrammen.

Im Header einer E-Mail, die Viren enthält, ist u.U. die IP-Adresse des Internet Service Provider (ISP) aufgeführt, der die E-Mail verschickt hat. Bei den meisten webbasierten E-Mail-Clients (Gmail, Yahoo, Outlook.com) ist das die IP-Adresse des Email-Providers. Um die tatsächliche IP-Adresse herauszufinden, muss man sich den „X-Originating-IP“-Wert anschauen. Die Whois-Datenbanken [Web-13] führen zu den Details, mit denen sich der ISP kontaktieren lässt, um die Untersuchung fortzuführen. Es ist zu bedenken, dass E-Mails von privaten Servern und offenen Relay-Mail-Servern stammen können. In diesem Fall kann es sehr schwer sein, den tatsächlichen Ursprung einer E-Mail zu ermitteln.

Die Untersuchung von Angriffen, bei denen Botnetze genutzt wurden, ist ebenfalls schwierig. Der Angreifer braucht keine Online-Verbindung mit dem Bot-Server oder den Bot-Clients. Das Tracing ist daher sehr schwierig bis nahezu unmöglich. In diesem Fall kann die Untersuchung der Bot-Clients zum Bot-Server führen. Man braucht jedoch Zugriff auf den Server, um den tatsächlichen Ursprung des Angriffs ermitteln zu können. Die Betreiber dieser Server wissen u.U. gar nicht, dass ihre Rechner Teil eines Botnetzes sind.

6.1.4.3 Berichterstattung

Die Art und Weise wie über Sicherheitsschwachstellen zu berichten ist, wird in Kapitel 7 beschrieben.

6.2 Social Engineering

Wir können alle denkbaren technischen Abwehrmaßnahmen realisieren, um digitale Assets vor Angriffen von außen zu schützen. Letzten Endes müssen Mitarbeiter (Benutzer und Administratoren) jedoch Zugriff auf diese Assets haben, um ihre Arbeit zu erledigen. Möglicherweise müssen sie sich authentifizieren, um über ihre PCs, Notebooks, Smartphones, Tablets oder andere Endgeräte Zugang zu erhalten. Jede physische Schutzmaßnahme für Assets im Büro oder das Büro selbst ist wirkungslos, wenn sich der Schutz des Heimarbeitsplatzes des IT-Managers leicht überwinden lässt.

Es ist der Mensch und sein Verhalten, der die größte Gefährdung für die Sicherheit darstellt. Wenn Menschen nachlässig mit sensiblen Informationen umgehen, hinterlässt dies zu viele Spuren, die zu sicheren Orten führen, und

diese Informationen dringen zu einfach an die Öffentlichkeit (über Gespräche oder auf elektronischem Weg).

Social Engineering ist die Kunst, den Menschen unter Nutzung seines allgemeinen Verhaltens als Angriffsvektor zu instrumentalisieren. Als soziales Wesen neigt der Mensch dazu, Fremden zu vertrauen und zu helfen. Das erzeugt eine Schwachstelle, an der Angriffe ansetzen können. Durch Manipulieren, Beeinflussen und Überreden hilfsbereiter Menschen versucht der Angreifer, an Zugangsdaten oder andere sensible Informationen zu gelangen.

Das kann durch direkte Interaktion mit dem Menschen oder unter Verwendung von Computer-/Netzwerkausrüstung erfolgen.

Die direkte, menschliche Interaktion schließt folgende Methoden ein:

1. Tailgating bzw. Piggybacking (ein Angreifer ohne Zugangsberechtigung hängt sich an autorisierte Mitarbeiter an, um Zutritt zu einem zugangsbeschränkten Bereich zu erhalten)
2. Lauschen (die vertraulichen Gespräche anderer ohne deren Wissen mithören)
3. Schulter-Surfen (jemandem ohne sein Wissen über die Schulter schauen, während er am Computer oder an Unterlagen arbeitet)
4. Telefongespräche (sich am Telefon als leitender Angestellter oder Support-Mitarbeiter ausgeben, um von arglosen Mitarbeitern Passwörter zu erfahren)

Computergestütztes Social Engineering kann wie folgt erfolgen:

1. Senden von E-Mails, die mit Schadsoftware infiziert sind.
2. Verwenden von Chat- oder Instant-Messaging-Programmen. Über Chat- und Instant-Messaging-Programme lassen sich anonyme Gespräche mit anderen irgendwo auf der Welt führen, ohne dass die wahre Identität des Gesprächspartners bekannt ist. Darüber hinaus lassen sich Daten über Instant Messenger leicht ausspähen.
3. Verwendung von Popup-Meldungen. So kann beispielsweise auf dem Computerbildschirm ein Fenster mit einer Meldung für den Benutzer auftauchen, in der es heißt, die Netzwerkverbindung sei unterbrochen.

Der Benutzer wird aufgefordert, seinen Benutzernamen und sein Passwort erneut einzugeben. Ein vorher, vom Eindringling installiertes Programm kann diese Daten dann an einen anderen Ort übermitteln.

1. Verschicken von Spam-E-Mails. Spam-E-Mails enthalten betrügerische Angebote und Links. Bei Klicken auf diese Links kann Schadsoftware installiert werden, die ein ganzes Netzwerk seines Schutzes beraubt.
2. Menschen dazu bringen, infizierte (manipulierte) Websites zu besuchen. Diese Phishing-Versuche können an viele verschickt werden oder stark individualisiert sein (Spear-Phishing).

Gegen Social Engineering gibt es kein alleiniges Gegenmittel. Es lassen sich Vorkehrungen zur Schadensminimierung treffen (z. B. möglichst geringe Zugriffsrechte, die das Ausführen der zugewiesenen Tätigkeit gerade noch zulassen, Aufteilung von Pflichten, rotierende Weitergabe von Pflichten). Die wichtigste Vorkehrung besteht jedoch in Aufklärung und Schaffung von Sicherheitsbewusstsein auf allen Ebenen des Unternehmens.

6.3 Sicherheitsbewusstsein

6.3.1 Die Bedeutung des Sicherheitsbewusstseins

Wie bereits in anderen Kapiteln dieses Lehrplans erwähnt, wandelt sich das Gefährdungsmodell ständig. Netzwerke wachsen, neue Anwendungen kommen auf den Markt, neue Schnittstellen gehen in Betrieb und neue Schwachstellen werden erzeugt und entdeckt.

Neben diesen technischen Aspekten gibt es noch den Faktor Mensch. Risiken, die einst ermittelt, aber nicht zu echten Problemen wurden, geraten leicht in Vergessenheit und entsprechende Sicherheitsvorkehrungen werden so vernachlässigt. Dadurch ergibt sich eine größere Angriffsfläche für Hacking-Angriffe und Social Engineering. Damit Sicherheitsadministratoren und alle anderen Mitarbeiter wachsam und über Änderungen am Gefährdungsmodell informiert bleiben, bedarf es regelmäßiger Schulungen zur Erhöhung des Sicherheitsbewusstseins. Diese Schulungen können auf verschiedene Benutzergruppen ausgerichtet sein: Entwickler, operatives Geschäft, Management, normale Mitarbeiter.

6.3.2 Schärfung des Sicherheitsbewusstseins

Es ist wichtig, ein „sicherheitsbewusstes“ Denken zu pflegen. Neben allgemeinen Informationen über Sicherheitsvorkehrungen im Unternehmen, sollten in den Schulungen echte Fälle behandelt werden – die bei Sicherheitstests entdeckt wurden oder als tatsächliche Störfälle auftraten. Aufbauend auf diesen Fällen müsste es einfacher sein, die im Unternehmen zu implementierenden Sicherheitsvorkehrungen oder Änderungen zu besprechen.

Ein Entwurf für diesen Abschnitt der Schulung muss Antworten auf folgende Fragen enthalten:

1. Wie sind die Angreifer (bzw. wir) vorgegangen?
2. Welche Folgen hatte das für das Unternehmen?
3. Was kostete es, den Störfall zu untersuchen und aufzuarbeiten?
4. Was kostete es, das Problem zu beheben?
5. Wie hätte sich der Störfall vermeiden lassen?
6. Welche Änderungen werden eingeleitet?

7 Auswertung von Sicherheitstests und Abschlussberichte – 70 min

Schlüsselbegriffe

Abnahmekriterium, Angriffsvektor, Dashboard, Endekriterium

Lernziele für das Thema „Auswertung von Sicherheitstests und Abschlussberichte“

7.1 Auswertung von Sicherheitstests

AS-7.1.1 (K2) Die Gründe verstehen, warum Sicherheitserwartungen und Abnahmekriterien mit der Veränderung des Umfangs und der Ziele eines Projekts angepasst werden müssen

7.2 Abschlussberichterstattung für Sicherheitstests

AS-7.2.1 (K2) Die Wichtigkeit verstehen, Ergebnisse von Sicherheitstests vertraulich und sicher zu halten

AS-7.2.2 (K2) Verstehen, warum die richtigen Steuer- und Datenerfassungsmechanismen geschaffen werden müssen, damit die Ausgangsdaten für die Sicherheitstest-Statusberichte zeitnah und präzise bereitgestellt werden können (z. B. ein Sicherheitstest-Dashboard)

AS-7.2.3 (K4) Einen gegebenen Sicherheitstest-Zwischenberichts analysieren können, um die Genauigkeit, Verständlichkeit und Zweckmäßigkeit für die Stakeholder erfassen zu können

7.1 Auswertung von Sicherheitstests

Die Messung von Sicherheitstestergebnissen und die Bewertung des Status im Hinblick auf Sicherheitserwartungen, Endekriterien und/oder Abnahmekriterien sind erforderlich, um zu bestimmen, ob die Tests abgeschlossen sind.

Zu Beginn eines Projekts ist es schwierig, alle Sicherheitsrisiken zu kennen. Darüber hinaus ändern sich die Erwartungen von Stakeholdern und Benutzern im Hinblick auf den nötigen Grad an Sicherheit mitunter. So könnte beispielsweise das Wissen um eine neue Gefährdung die Stakeholder veranlassen, mehr Sicherheit als ursprünglich gedacht zu fordern. Das ist einer der Gründe, warum Sicherheitsrisikobewertungen im Verlauf eines Projekts überarbeitet und die Ergebnisse in die Planung und Umsetzung des Sicherheitskonzept einfließen müssen.

7.2 Abschlussberichterstattung für Sicherheitstests

7.2.1 Vertraulichkeit von Sicherheitstestergebnissen

Natürlich weiß der typische Tester nach Abschluss der Tests mehr über das Testobjekt als die meisten Entwickler oder Programmierer. Mit gründlichen Tests lassen sich die entscheidenden Schwächen und Stärken des Systems finden. Dasselbe gilt für Sicherheitstests.

Durch Testen der Sicherheitsimplementierung lassen sich versteckte Schlupflöcher und Sicherheitsschwachstellen aufspüren. In diesem Zusammenhang muss an die möglichen negativen Folgen gedacht werden, die es hat, wenn über die direkten Stakeholder hinaus andere Personen Kenntnis von diesen Schwachstellen erhalten. Generell ist es ratsam, Informationen nur den Personen verfügbar zu machen, die diese kennen müssen. Das gilt vor allem für Ergebnisse von Sicherheitstests; diese Art von Informationen nur mit Bedacht weiterzugeben, gilt als gute Praxis.

7.2.2 Schaffung der richtigen Steuer- und Datenerfassungsmechanismen für Sicherheitstest-Statusberichte

Den Folgen einer Sicherheitsschwachstelle wird üblicherweise eine höhere Brisanz als „normalen“ Fehlern zugeschrieben. Daraus folgt, dass die Merkmale des Fehlers und die mit ihm einhergehenden Risiken präziser dokumentiert werden müssen. In den meisten Projekten werden Sicherheitsfehler mit höherer Schwere als funktionale Fehler eingestuft.

Aus letzterem folgt, dass das Management stärkeren Fokus auf Sicherheitsfehler, ihre Risiken und Behebungsmöglichkeiten legt. In Berichten über Sicherheitsfehler müssen das mögliche Schadensausmaß eines entdeckten Problems sowie die Genauigkeit der Testergebnisse auf klar definierte und zeitnahe Weise sorgfältig bewertet werden. Es ist gute Praxis, mit dem Management zu besprechen, wie und wann es Zugriff auf die Sicherheitsfehlerberichte haben will.

7.2.3 Analysieren von Sicherheitstest-Zwischenberichten

Sicherheitstestberichte können während des gesamten Testprozesses oder erst nach Abschluss der Tests angelegt werden (bei Abschluss der Systemsicherheitstests oder der Abnahmetests). Eine frühzeitige Erstellung von Berichten über Sicherheitstests ist ratsam, weil dann mehr Zeit für die Beseitigung von Sicherheitsschwachstellen bleibt. Wenn der Sicherheitstestprozess an den in diesem Lehrplan beschriebenen Prozess angelehnt ist, kann das Testteam während der gesamten Testaktivitäten Schwachstellen entdecken und seine Beobachtungen dokumentieren.

Der Sicherheitstestbericht sollte folgende Abschnitte enthalten:

1) Berichts-ID
2) Zusammenfassung
    a) Zusammenfassung für die Geschäftsführung (Management Summary)
    b) Wichtigste Erkenntnisse (Key Findings)
3) Abweichungen
    a) Genutzter Testprozess
    b) Abweichungen vom geplanten Testprozess
    c) Genutzte Methoden und Werkzeuge (Konfigurationen, Richtlinien)
4) Umfassende Bewertung
    a) Auswertung des Überdeckungsgrades der Tests anhand der im Testkonzept angegebenen Kriterien
    b) Erläuterung für Elemente oder Features, die nicht getestet wurden
5) Zusammenfassung der Ergebnisse
    a) Zusammenfassung der Ergebnisse der Sicherheitstests
    b) Liste mit allen behobenen Sicherheitsschwachstellen und der Art ihrer Behebung
    c) Liste mit allen nicht behobenen Schwachstellen
6) Auswertung
    a) Auswertung der gewonnenen Testergebnisse und ihres Status anhand der Endekriterien
    b) Ermittelte Risiken (Klassifizierungen) und Schadensausmaß nicht behobener Sicherheitsschwachstellen
7) Zusammenfassung der Aktivitäten
8) Genehmigungen

Die Wirksamkeit der Berichterstellung für Sicherheitstests hängt von folgenden Faktoren ab:

1. Zeitpunkt des Berichts
2. Inhalt des Berichts
3. Empfänger des Berichts
4. Abstimmung des Inhalts auf den Informationsbedarf der Empfänger

Für die unterschiedlichen Erfordernisse der verschiedenen Stakeholder werden u.U. mehrere Berichte benötigt. So wird sich der Inhalt eines Berichts für die Geschäftsführung vom Inhalt des Berichts für den Systemarchitekten unterscheiden.

8 Sicherheitstestwerkzeuge – 55 min

Schlüsselbegriffe

keine

Lernziele für das Thema „Sicherheitstestwerkzeuge“

8.1 Arten und Funktionen von Sicherheitstestwerkzeugen

AS-8.1.1 (K2) Die Funktion statischer und dynamischer Analysewerkzeuge bei Sicherheitstests erläutern können

8.2 Werkzeugauswahl

AS-8.2.1 (K4) Sicherheitstesterfordernisse, denen mit einem oder mehreren Werkzeugen Rechnung getragen wird, analysieren und dokumentieren können

AS-8.2.2 (K2) Die Probleme mit Open-Source-Werkzeugen verstehen

AS-8.2.3 (K2) Beurteilen können, ob ein Anbieter in der Lage ist, Werkzeuge häufig zu aktualisieren oder im Hinblick auf Sicherheitsgefährdungen auf dem neuesten Stand zu halten

8.1 Arten und Funktionen von Sicherheitstestwerkzeugen

Die von der Hacking-Community entwickelten Exploits haben die Entwicklung von Sicherheitstestwerkzeugen zur Abwehr dieser Gefährdungen vorangetrieben. Schon mit den ersten Hacking-Aktivitäten (wie dem Knacken von Passwörtern) wurden einfache Werkzeuge erfunden, entwickelt und von ihren Nutzern verbessert. Werkzeuge, die sich als wirksam erwiesen, wurden innerhalb der Hacker-Community weitergeben und weiterentwickelt. Zunächst wurden diese Werkzeuge für konkrete Aufgaben und Umgebungen entwickelt. Die Nutzerfreundlichkeit war kein Thema, weil fast alle Benutzer einen technischen Hintergrund hatten. Einige Hacker-Werkzeuge wurden letztlich sogar zur Basis für legale Sicherheitstestwerkzeuge, die von Administratoren und Testern im Bereich Informationssicherheit genutzt werden.

Ein Beispiel: „John the Ripper“ war ein frühes Open-Source-Werkzeug zum Knacken von Passwörtern, das Hackern ursprünglich zum Erraten von Passwörtern für Unix-Netzwerke und -Anwendungen diente. Heute wird es in weiterentwickelter Form für legitime Zwecke wie der Erkennung schwacher Unix-Passwörter genutzt. [Web-26]

Als große Anbieter von Test- und Softwareentwicklungswerkzeugen sowie Spezialwerkzeugen mit der Entwicklung von Sicherheitstestwerkzeugen begannen, wuchsen der Funktionsumfang und die Benutzerfreundlichkeit vieler dieser Werkzeuge. Dieser große Funktionsumfang führte jedoch auch zu komplexeren Werkzeugkonfigurationen und Implementierungsproblemen.

Parallel zur Entstehung der frühen Sicherheitswerkzeuge entstanden die ersten Open-Source-Werkzeugversionen von Frameworks wie Nessus, Metasploit und anderen. Sie boten bessere und mehr Funktionen sowie in einigen Fällen auch eine intuitiv bedienbare grafische Benutzeroberfläche.

Heute sind enorm viele Sicherheitstestwerkzeuge verfügbar. Für nahezu jede Umgebung oder Aufgabe findet man ein dediziertes Testwerkzeug, entweder als Open-Source- oder als Lizenzprodukt. Die Herausforderung ist bei all diesen Werkzeugen, dass die meisten von ihnen intelligente Systeme sind, die mit nicht standardisierten Tests arbeiten. Alle Entwickler dieser Systeme sind sich mehr oder weniger darüber einig, wie Schutzmechanismen zu testen oder nach Schwachstellen zu suchen ist. Diese Werkzeuge können jedoch mit unterschiedlichen Testdaten, unterschiedlichen Testimplementierungen und unterschiedlichen Interpretationen der Ergebnisse arbeiten.

Mit Sicherheitstestwerkzeugen lässt sich die Bewertung von Sicherheitsmechanismen automatisieren. Sicherheitstestwerkzeuge können zudem genutzt werden, um unbekannte Arten von Schwachstellen zu entdecken. Mit dem Wissen darum, dass sich Schutzmechanismen oder Schwachstellen desselben Typs unterschiedlich implementieren lassen, ist die Auswahl und Nutzung von Sicherheitstestwerkzeugen eine Herausforderung für den Sicherheitstester, weil sich die Werkzeuge darin unterscheiden, wie sie Schwachstellen finden und Schutzmechanismen prüfen.

Auf den Websites von Web Application Security Consortium [Web-18] und OWASP [OWASP1] finden sich Listen mit kategorisierten Werkzeugen. Kali Linux [Web-17], ein Framework für Penetrationstests, bietet weitere Möglichkeiten der Klassifizierung von Sicherheitstestwerkzeugen.

Es gibt sowohl kommerzielle wie auch Open-Source- Sicherheitswerkzeuge. Zu dem Zeitpunkt, zu dem wir diesen Lehrplan entwickelten (2016), fanden wir nur eine limitierte Anzahl von Ressourcen, die einen mehr oder weniger umfassenden Überblick über zuverlässige und vertrauenswürdige Open-Source-Sicherheitswerkzeuge boten. Eine dieser Listen mit Sicherheitswerkzeugen finden Sie unter https://sectools.org [Web-24]. Es wird erwartet, dass der erfahrene Sicherheitstester seine eigene Liste mit verfügbaren Werkzeugen pflegt und aktuell hält, weil sich das Angebot ständig ändert.

Sowohl statische als auch dynamische Analysewerkzeuge sind bei Sicherheitstests hilfreich. Der Vorteil von statischen Tests liegt darin, dass sie sehr früh im Entwicklungszyklus durchgeführt werden können. Statische Analysewerkzeuge sind für die meisten Programmiersprachen verfügbar und beinhalten in der Regel eine Berichtsfunktion für Sicherheitsaspekte.

Der Unterschied zwischen dynamischen und statischen Testwerkzeugen im Sicherheitsbereich ist im Vergleich zu anderen Testarten mitunter nicht klar erkennbar. Die Definition des statischen Testens bezieht sich auf die Durchführung von Testaktivitäten, wenn sich das zu testende System oder Objekt nicht im Betriebsmodus befindet. Es ist nicht ungewöhnlich, dass dynamische Sicherheitstestwerkzeuge statt der zu testenden Anwendung das System sondieren. So gesehen werden diese dynamischen Testwerkzeuge als eine Art statische Testwerkzeuge eingesetzt. Ein Beispiel: Ein dynamisches Sicherheitstestwerkzeug kann einen statischen Scan einer Datenbank durchführen. Wird allerdings das gesamte System als Testobjekt betrachtet, dann sind die Werkzeuge in der Tat dynamische Testwerkzeuge.

8.2 Werkzeugauswahl

8.2.1 Analysieren und Dokumentieren von Sicherheitstesterfordernissen

Folgende Dokumente können u.a. eine Testbasis für den IT-Sicherheitstest bilden:

1. die Sicherheitsrichtlinie des Unternehmens
2. die Testrichtlinie des Unternehmens
3. Ergebnisse von Gefährdungs- und Risikoanalysen für das aktuelle System/Projekt
4. Anforderungen und andere Systemspezifikationen
5. Systemarchitektur und -entwurf
6. Sicherheits(test)strategie
7. das zu testende System bzw. die Anwendung
8. bekannte Sicherheitsgefährdungen, Exploits und Schwachstellen
9. Benutzerprofile

Alle oben genannten Punkte und viele weitere können Informationen zu Gefährdungen und ausnutzbaren Schwachstellen liefern. Anforderungen und Entwurfsdokumente müssen Angaben dazu enthalten, wie die Daten oder Informationen geschützt werden. Daraus leitet sich Folgendes ab:

1. zu testende Schnittstellen (einschließlich der GUI)
2. zu überprüfende Protokolle und Standards
3. Kodierrichtlinien, die sichere Programmierpraktiken vorgeben
4. zu prüfende Systemkomponenten-Konfigurationen (gehärtet)

Es muss bestimmt werden, ob Sicherheitstests eine Entwicklungsaktivität oder eine Wartungs-/Betriebsaktivität sind. All diese Informationen ergeben die Anforderungen für die zu verwendenden Sicherheitstestwerkzeuge.

8.2.2 Probleme mit Open-Source-Werkzeugen

In [ISTQB_ATM_SYL] werden die Probleme, die es mit Open-Source-Werkzeugen geben kann, ausführlich besprochen.

Wie eingangs erwähnt kommen viele Sicherheitstestwerkzeuge aus dem Open-Source-Bereich. Es gibt sie unter einer Vielzahl von Lizenzmodellen, die jeweils die kostenlose Nutzung und die Modifikation des Quellcodes erlauben. Nicht für

alle Unternehmen oder Projekte werden Open-Source-Werkzeuge für den Entwicklungsprozess in Erwägung gezogen.

Werkzeuge unter diesen Lizenzen bieten viele Vorteile, aber auch Nachteile. In vielen Fällen sind Open-Source-Werkzeuge kostenlos erhältlich. Im Unternehmen muss es jedoch die technischen Kapazitäten für ihre Betreuung und Konfiguration geben. Fehlen diese, können sie möglicherweise beim Entwickler der Software eingekauft werden, was allerdings wiederum Kosten verursacht. Administrations- und Benutzerhandbücher, sofern vorhanden, sind meist für ein bestimmtes (technisches) Publikum verfasst und decken u.U. nicht den gesamten Funktionsumfang des Werkzeugs ab. In jüngster Zeit bieten Media-Kanäle wie YouTube eine zusätzliche Informationsquelle für die Verwendung dieser Werkzeuge.

Folgende Aspekte sind bei der ROI-Kalkulation für ein Open-Source-Werkzeug zu berücksichtigen:

1. der limitierte Funktionsumfang des Werkzeugs (meist werden keine weiteren oder anderen Funktionen angeboten)
2. der Zeitaufwand für das Erlernen der Administration, Konfiguration und Verwendung des Werkzeugs
3. die im Verlauf des Lebenszyklus in Benutzerforen und -gruppen zu investierende Zeit
4. die für Software-Updates und Software-Upgrades benötigte Zeit (und die interne Upgrade-Richtlinie)
5. die weitere Entwicklung des Werkzeugs (einige verschwinden oder werden zu kommerziellen Produkten)
6. die Reaktionsschnelligkeit der Support-Community für das Werkzeug

Bei den meisten Unternehmen oder Projekten ist die Anzahl der Lizenzen, die für Sicherheitstestwerkzeuge benötigt wird, auf eine oder einige wenige beschränkt. Nur große Unternehmen werden mehr Lizenzen in Erwägung ziehen. Wie viele Lizenzen gebraucht werden, hängt in erster Linie vom Funktionsumfang ab, den das Werkzeug bietet (wie z. B. Webanwendungen, Webdienste, Code-Analyse, usw.), sowie der erwarteten Häufigkeit und Nutzungszeit dieser Dienste sowie der Anzahl der Sicherheitstester, die mit dem Werkzeug arbeiten.

8.2.3 Beurteilung der Fähigkeiten eines Werkzeuganbieters

Wird ein Werkzeug von einem Anbieter eingekauft, sollte dieser Anbieter eine Reihe von Services zur Verfügung stellen, so dass der Sicherheitstestservice beginnen und der benötigte Umfang an internem Support ausgeweitet werden kann.

Folgende Faktoren können zur Beurteilung der Fähigkeiten des Anbieters genutzt werden:

1. Art der angebotenen Lizenzen (feste Lizenz/Client-Lizenz/Floating-Lizenz/Token-Lizenz)
2. Skalierbarkeitsoptionen für Lizenzen (pro Funktionsbereich, Anzahl der Lizenzen)
3. Helpdesk-/Support-Angebot (Support-Zeiten)
4. Forum/Benutzer-Community
5. Häufigkeit Software-Updates
6. Administrations- und Benutzerhandbücher
7. Support- und Wartungsverträge

9 Standards und Branchentrends – 40 min

Schlüsselbegriffe

konsensbasierter Standard

Lernziele für das Thema „Standards und Branchentrends“

9.1 Sicherheitsteststandards

9.1.1 (K2) Die Vorteile der Verwendung von Sicherheitsteststandards kennen und wissen, wo sie zu finden sind

9.1.2 (K2) Den Unterschied in der Anwendbarkeit von Standards in regulatorischen und vertraglichen Situationen verstehen

9.2 Anwenden von Sicherheitsstandards

9.2.1 (K2) Den Unterschied zwischen obligatorischen (normativen) und optionalen (informativen) Klauseln in Standards kennen

9.3 Branchentrends

9.3.1 (K2) Informationsquellen für Branchentrends in der Informationssicherheit kennen

9.1 Verstehen von Sicherheitsteststandards

Standards bzw. Normen verschiedener Arten bilden einen fachlichen Konsens oder gesetzliche Verpflichtungen ab. Ein konsensbasierter Standard repräsentiert die fundierte Meinung eines sachkundigen Expertengremiums und wird für die freiwillige Verwendung (in Teilen oder komplett) in vertraglichen Vereinbarungen zwischen Leistungsanbietern und Kunden zur Verfügung gestellt. Darüber hinaus gibt es Pseudo-Standards, die von formloseren oder sich selbst definierenden Gruppen stammen und anbieterspezifisch sein können.

In regulierten Branchen (u.a. Medizin, Finanzwesen, Verkehr und Energiewesen) können staatliche Stellen die Einhaltung selbst erlassener Vorschriften oder ihre Interpretationen von sonstigen freiwilligen Standards fordern.

9.1.1 Die Vorteile der Verwendung von Sicherheitsteststandards

Standards und Normen im Allgemeinen bieten Orientierung und Einheitlichkeit bei der Durchführung einer Aufgabe. In der Regel werden Standards von einschlägigen Experten entwickelt, die sich auf einen Konsens bezüglich wirksamer Praktiken verständigt haben. Die Nutzung von Sicherheitsteststandards und –Normen hat folgende Vorteile:

1. Sie geben einen Rahmen für die Sicherheitstests vor, so dass nicht mit einem weißen Blatt Papier begonnen werden muss.
2. Sie definieren wirksame Schutzmaßnahmen und Tests auf die gängigsten Sicherheitsangriffe.
3. Die Standards lassen sich auf die Erfordernisse des Projekts oder Unternehmens zuschneiden.
4. Mit der Einhaltung anerkannter Sicherheitsteststandards lässt sich die geforderte Sorgfalt beim IT-Sicherheitstest belegen.

9.1.2 Anwendbarkeit von Standards in regulatorischen und vertraglichen Situationen

Bei regulierten Aktivitäten müssen sich alle Parteien ihrer Verpflichtung bewusst sein, geltende Standards und Normen einzuhalten. Verstöße könnten die Genehmigung des zu entwickelnden Produkts verzögern oder ganz verhindern und im Extremfall in finanziellen oder strafrechtlichen Sanktionen münden.

Insbesondere die Gesetzgebung zum Datenschutz und zur IT-Sicherheit sind stark durch nationale und europäische Initiativen geprägt, die die Etablierung eines einheitlichen Rechtsrahmens in Europa zum Ziel haben. Zu nennen sind in diesem Zusammenhang das BSI-Gesetz [BSIG] und die Netz- und Informationssicherheitsrichtlinie der Europäischen Union (EU-NIS) [ÉUNIS].

Das BSI Gesetz (Gesetz zur Erhöhung der Sicherheit informationstechnischer Systeme oder umgangssprachlich IT-Sicherheitsgesetz), ist ein am 25.07.2015 in Kraft getretenes Gesetz der deutschen Bundesregierung, welches die Betreiber kritischer Infrastrukturen zur Einhaltung bestimmter Mindestanforderungen im Bereich der IT-Sicherheit verpflichtet. Zu den Verpflichtungen zählen unter anderem die Benennung einer Kontaktstelle zur Kommunikation mit dem BSI, die Meldung erheblicher Störungen an das BSI, sowie die Bereitschaft, in Sicherheitsfragen Beratung und Unterstützung durch das BSI bzw. kompetente Dritte anzunehmen. Zusätzlich sind alle Betreiber dazu verpflichtet, Maßnahmen zum Schutz der IT-Sicherheit nach Stand der Technik umzusetzen und mindestens alle zwei Jahre einen Nachweis über die Umsetzung der gesetzlich geforderten Maßnahmen zu führen. Maßnahmen und Verpflichtungen werden derzeit durch die Etablierung sektorenspezifischer Standards weiter ausgeführt.

Während das BSI-Gesetz bereits als nationales Gesetz Gültigkeit hat und sich an die Betreiber kritischer Infrastrukturen richtet, ist die Netz- und Informationssicherheitsrichtlinie eine europäische Richtlinie mit nationaler Umsetzung bis Ende

2018. Wie bereits durch das BSI-Gesetz geregelt, fordert die EU-NIS die Meldung von Sicherheitsvorfällen und der Einhaltung von Sicherheitsanforderungen. Während das BSI-Gesetz sich allein auf kritische Infrastrukturen bezieht, spricht die EU-NIS von wesentlichen Diensten, zu denen neben den kritischen Infrastrukturen auch große Cloud-Anbieter, Suchmaschinenbetreiber oder Betreiber von Online-Marktplätzen gehören. Auch hier wird das BSI als Kontrollinstanz prüfen, ob die Betreiber wesentlicher Dienste die neuen Auflagen einhalten.

Im Bereich des Datenschutzes sind das Bundesdatenschutzgesetz (BDSG) [BDSG] und die Verordnung zum Schutz natürlicher Personen bei der Verarbeitung personenbezogener Daten und zum freien Datenverkehr (EU-DSGVO) [EUDSGVO] relevant. Die EU-DSGVO ersetzt nationales Recht, d.h. sie hat auch ohne nationale Gesetzgebung direkte Geltung und weist im Vergleich zum alten BDSG insbesondere einen erhöhten Sanktionsrahmen in Höhe von maximal vier Prozent des weltweiten Jahresumsatzes einer Organisation oder bis zu 20 Millionen Euro auf. Mit der Novellierung des BDSG zum 27.04.2017 wurden relevante Anforderungen der EUDSGVO in das BDSG übernommen.

In vertraglichen Situationen bieten Standards eine akzeptable und praktische Grundlage für die Aushandlung einer Vereinbarung über Projekt- und Produktanforderungen; sie bieten einen guten Ausgangspunkt für die Verhandlungen. Konsensbasierte Standards ermöglichen das Kommunizieren von Best Practices sowie deren Übernahme oder Anpassung an die jeweilige Situation.

Sofern Standards/Normen nicht einseitig von einer Behörde oder in einem nicht verhandelbaren Vertrag vorgegeben werden, können sie als Grundstruktur für eine ausgehandelte Vereinbarung genutzt oder als Verhaltenskodex an die eigene Arbeit angelegt werden. Wird ein Vertrag auf der Basis der Zusage oder Einwilligung, bestimmte Standards einzuhalten, vergeben, hat die Partei die Pflicht, diese Standards streng zu befolgen und jegliche Abweichungen zu dokumentieren.

9.1.3 Auswahl von Sicherheitsstandards

Mit Sicherheit lassen sich nicht alle Sicherheitsstandards auf alle Situationen anwenden. Es liegt in der Verantwortung des jeweiligen Unternehmens, den/die geeignetste(n) Standard(s) für seine Systeme, Anwendungen, sensiblen digitalen Assets, Risikostufen und Compliance-Anforderungen zu ermitteln. Klar sein muss auch, dass viele Standards auf die spezifischen Anforderungen eines Unternehmens zugeschnitten werden können.

Eine Liste mit gängigen Sicherheitsstandards und –Normen finden Sie in Kapitel 10.

9.2 Anwenden von Sicherheitsstandards

Standards und Normen sind von präzisen sprachlichen Formulierungen geprägt: So gibt das Wort *muss/müssen* obligatorische Vorgaben an, die zu befolgen sind, um standardkonform zu sein. Die Wörter *sollte(n)* und *kann/können* verweisen auf optionale Aufgaben, deren Ausführung für die Konformität mit dem Standard nicht zwingend erforderlich ist.

Ein typischer Fehlgebrauch ist es, diese klare Unterscheidung zu missachten, indem man entweder einen optionalen Punkt verpflichtend vorschreibt oder ein obligatorisches Element als optional behandelt.

Unternehmens- oder projektspezifische Situationen können Abweichungen vom strengen Sinn eines geltenden Standards erfordern. Begründungen für Auslassungen, Modifikationen oder Erweiterungen des Inhalts des Standards müssen dokumentiert und von allen Parteien vereinbart werden.

9.3 Branchentrends

9.3.1 Informationsquellen für Branchentrends in der Informationssicherheit

Sowohl allgemeine und branchenspezifische Nachrichtendienste (Publikationen, Websites, E-Mail-Newsletter) als auch Events (Kongresse, Handelsmessen, Treffen von Berufsverbänden) bieten Informationen und Diskussionen über neue oder wachsende Bedrohungen. Wer einem guten/aktiven Fachverband oder einer Praxisgemeinschaft angehört, erhält sehr wahrscheinlich zeitnah Neuigkeiten zur Thematik. Bei der Geschwindigkeit, mit der sich neue Exploits entwickeln, können elektronische Warnsysteme die schnellste Form der Reaktion bieten.

Das regelmäßige Auftauchen der häufigsten oder schadensträchtigsten Exploits in Publikationen lässt häufig Rückschlüsse auf allgemeinere Trends zu. Besondere Aufmerksamkeit sollte man jedoch Problemen widmen, die spezifischer für die Branche, den Anwendungsbereich oder die Produkte sind, mit denen man arbeitet. Über diese Probleme wird mit größerer Wahrscheinlichkeit in Fachpublikationen und branchenspezifischen Nachrichtendiensten oder auf Technik-Kongressen und Fachveranstaltungen berichtet.

9.3.2 Prüfen von Sicherheitstestpraktiken auf Optimierungspotenzial

Bei Einführung neuer Technologien oder neuer Anwendungen für bestehende Technologie gibt es häufig ein Fenster, in dem die Chancen für Missbrauch und Ausnutzung größer sind, weil sich Risiken und Beschränkungen erst später abzeichnen.

Denken Sie beispielsweise an mobile Geräte mit standortbasierten Diensten. Für ein Mehr an Komfort oder andere Anreize scheinen Menschen bereitwillig zu akzeptieren, dass ihre Bewegungen und Aktivitäten minutengenau verfolgt werden.

Kriminelle, wirtschaftliche und politische Akteure sowie Hacktivisten operieren mit unterschiedlichsten Motiven und ständig wachsenden Ressourcen. Erpressungs- und Schutzprogramme haben sich aus der physischen in die digitale Welt verlagert.

Große Ad-hoc-Netzwerke mit ideologisch-motivierten Individuen lassen sich sehr kurzfristig gegen die Ziele ihres Zorns richten. Wirtschaftsspionage ist häufig gut finanziert und motiviert. Staaten, die auf wirtschaftliche und militärische Vorteile aus sind, lassen sich das besonders viel kosten und glauben, sie seien immun gegen Sanktionen oder Gegenreaktionen.

Weil sich die Gefährdungen ständig ändern und komplexer werden, müssen Sicherheitstester stets bereit sein, sich der nächsten Gefährdung zu stellen. Bewusstsein für die Branche, Beobachtung von Sicherheitstrends und Erwerb der geeignetsten Werkzeuge bilden für ein Unternehmen die besten Schutzmaßnahmen.

10Quellenangaben

10.1 ISTQB-Dokumente

[ISTQB_FL_SYL] ISTQB Foundation Syllabus, 2011 [ISTQB_ATM_SYL]

ISTQB Advanced Test Manager Syllabus, 2012

[ISTQB_ATTA_SYL] ISTQB Advanced Technical Test Analyst Syllabus, 2012

10.2 Gesetze

[BDSG] – Bundesdatenschutzgesetz, online: http://www.gesetze-im-internet.de/bdsg_1990/BDSG.pdf

[BSIG] – Gesetz über das Bundesamt für Sicherheit in der Informationstechnik (BSI-Gesetz – BSIG); https://www.bsi.bund.de/DE/DasBSI/Gesetz/gesetz_node.html

[EUDSGVO] – Verordnung (EU) 2016/679 des Europäischen Parlaments und des Rates vom 27. April 2016 zum Schutz natürlicher Personen bei der Verarbeitung personenbezogener Daten, zum freien Datenverkehr und zur Aufhebung der Richtlinie 95/46/EG (Datenschutz-Grundverordnung) (Text von Bedeutung für den EWR);

http://eur-lex.europa.eu/legal-content/DE/ALL/?uri=OJ:L:2016:119:TOC

[EUNIS] – Richtlinie (EU) 2016/1148 des Europäischen Parlaments und des Rates vom 6. Juli 2016 über Maßnahmen zur Gewährleistung eines hohen gemeinsamen Sicherheitsniveaus von Netz- und Informationssystemen in der Union; http://eur-lex.europa.eu/legal-content/DE/TXT/?uri=OJ:L:2016:194:TOC

10.3 Standards/Normen

[ISO/IEC/IEEE 29119-3] ISO/IEC/IEEE-Standard: Software-und Systemengineering – Software-Test – Teil 3: Testdokumentation

[IEEE 12207] ISO/IEC/IEEE-Standard: Software-und Systemengineering – Software-Lebenszyklusprozesse

[COBIT] COBIT – http://www.isaca.org

[ISO27001] ISO Standard ISO27001: Informationstechnik – IT-Sicherheitsverfahren – Informationssicherheits-Managementsysteme – Anforderungen

[ISO27005] ISO Standard ISO27005: Informationstechnik - IT-Sicherheitsverfahren - Informationssicherheits-Risikomanagement

[ISO31000] ISO Standard ISO31000: Risikomanagement - Allgemeine Anleitung zu den Grundsätzen und zur Implementierung eines Risikomanagements

[ISO31010] ISO Standard ISO31010: Risikomanagement - Verfahren zur Risikobeurteilung (IEC/ISO 31010:2009); Deutsche Fassung EN 31010:2010

[BSIITG] Bundesamt für Sicherheit in der Informationstechnik, Hrsg., IT-Grundschutz-Kataloge. Köln: Bundesanzeiger, 2016.

[BSI200-1] Bundesamt für Sicherheit in der Informationstechnik, BSI-Standard 200-1: Managementsysteme für Informationssicherheit (ISMS), Version 1.0, Bonn, 2017.

[BSI200-2] Bundesamt für Sicherheit in der Informationstechnik, Hrsg., BSI-Standard 200-2: IT-Grundschutz-Vorgehensweise, Version 1.0, Bonn, 2017.

[BSI200-3] Bundesamt für Sicherheit in der Informationstechnik, Hrsg., BSI-Standard 200-3: Risikomanagement, Version 1.0, Bonn, 2017.

[PCI] – Payment Card Industry Standard – https://www.pcisecuritystandards.org/

10.4 Bücher

[Chapman, 2000] Chapman, Cooper, Zwicky, Building Internet Firewalls, O'Reilly & Associates, 2000.

[Jackson, 2010] Jackson, Christopher; Network Security Auditing, 2010.

[Lund, 2011] Mass Soldal Lund, Bjørnar Solhaug, Ketil Stølen: Model-Driven Risk Analysis, The CORAS Approach, Springer Verlag Berlin Heidelberg 2011, ISBN: 978-3-642-12322-1

10.5 Artikel

[ComputerWeekly] http://www.computerweekly.com/news/2240113549/Cattles-lost-backup-tapes-highlight-risk-of-unencrypted-data-storage

[Northcutt, 2014] Northcutt, Stephen; Security Controls, SANS Institute.

[Washington Post, 2007] http://www.washingtonpost.com/wp-dyn/content/article/2007/05/04/AR2007050402152.html

10.6 Leitfäden

[Bittau] Cryptographic protection of TCP Streams (tcpcrypt) https://tools.ietf.org/html/draft-bittau-tcp-crypt-04

[BSIPT] Bundesamt für Sicherheit in der Informationstechnik: Ein Praxis-Leitfaden für IS-Penetrationstests, Version 1.2, 2016

[CERT1] Top 10 Secure Coding Practices
https://www.securecoding.cert.org/confluence/display/seccode/Top+10+Secure+Coding+Practices

[CERT2] http://www.cert.org/secure-coding/publications/index.cfm

[CERT3] http://www.cert.org/secure-coding/tools/index.cfm

[ETSI203251] ETSI DEG 203 251: Methods for Testing and Specification (MTS) Risk-based security

[IEEE1] Avoiding the Top 10 Security Flaws

http://cybersecurity.ieee.org/center-for-secure-design/avoiding-the-top-10-security-flaws.html

[NIST 800-30] NIST Special Publication 800-30, Rev 1, Guide for Conducting Risk Assessments (2012)

[NISTIR 7298] Glossary of Key Information - Security Terms, Revision 2 (2013)

[OWASP1] OWASP Secure Coding Practices Quick Reference Guide
https://www.owasp.org/index.php/OWASP_Secure_Coding_Practices_-_Quick_Reference_Guide

[OWASP2] OWASP Risk Rating Methodology https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology

[OWASP3] OWASP Sample Authorization Form https://www.owasp.org/index.php?title=Authorization_form

[SANS1] 25 Most Dangerous Software Errors – http://www.sans.org

[SANS2] Password Construction Guidelines – https://www.sans.org/security-resources/policies/general/pdf/password-construction-guidelines

10.7 Berichte

[WhiteHat Security, 2014] https://www.whitehatsec.com

[ETSI101583] ETSI DTS 101 583: Methods for Testing and Specification (MTS) - Security Testing Terminology

[ETSI101182] ETSI DTR 101 182: Methods for Testing and Specification (MTS) - Security Testing Case Study Experience

10.8 Internet

[CERT4] Vulnerability Notes Database – http://www.kb.cert.org/vuls/

[Chopitea] tomchop.me/2012/12/yo-dawg-i-heard-you-like-xoring/

[EICAR] www.eicar.org

[HPI1] Hasso Plattner Institut: Datenbank für IT-Angriffsanalysen – https://hpi-vdb.de/vulndb/

[RFC2828] Internet Security Glossary – http://www.rfc-archive.org/getrfc.php?rfc=2828

[Web-2] National Vulnerability Database – https://web.nvd.nist.gov/view/ncp/repository

[Web-3] Website Security Statistics Report – https://www.whitehatsec.com/resource/stats.html

[Web-4] The Google Hacking Database – http://hackersforcharity.org/ghdb

[Web-5] Shodan – shodanhq.com

[Web-6] NetCat – http://sectools.org/tool/netcat/

[Web-7] IP Tracer – http://www.ip-adress.com/ip_tracer

[Web-8] Computer Forensics, Cybercrime and Steganography Resources – http://www.forensics.nl

[Web-9] Tor Project – https://www.torproject.org/

[Web-10] Netstat – https://technet.microsoft.com/en-us/library/Bb490947.aspx

[Web-11] Tracert – http://www.tracert.com

[Web-12] RIPE Scan – https://www.ripe.net

[Web-13] Whois – https://www.whois.net/

[Web-14] NetCat – http://netcat.sourceforge.net/

[Web-15] Fping – http://fping.org

[Web-16] Hidetools – http://hidetools.com/

[Web-17] Kali Linux – https://www.kali.org/

[Web-18] Web Application Security Consortium – http://www.webappsec.org/

[Web-19] Hping - http://www.hping.org/

[Web-20] Nmap – https://nmap.org/

[Web-21] Zenmap – https://nmap.org/zenmap/

[Web-22] Xprobe2 – http://null-byte.wonderhowto.com/how-to/hack-like-pro-conduct-os-fingerprinting-with-xprobe2-0148439/

[Web-24] Top 125 Network Security Tools – https://sectools.org

[Web-25] DNS Lookup – https://who.is/dns/

[Web-26] John the Ripper – http://www.openwall.com/john/