Schwachstellenbewertung

Zielsetzung

- Identifikation und Bewertung von Schwachstellen in Systemen, Netzwerken und Anwendungen, die von Angreifern ausgenutzt werden könnten.

Vorgehensweise

1. Erstellen einer möglichst vollständigen Asset-Liste, um alle relevanten Komponenten zu identifizieren.
2. Bewertung kritischer Anwendungen und Systeme basierend auf ihrer Bedeutung und ihren Risiken.
3. Analyse und Dokumentation von Schwachstellen sowie Priorisierung nach Kritikalität.

Erfolgsfaktoren

- Regelmäßigkeit der Bewertung.
- Priorisierung nach Risiko und Kritikalität der Systeme.
- Dokumentation zur Nachvollziehbarkeit und für nachfolgende Maßnahmen.

Schwachstellenscans

Zielsetzung

- Aufdecken von potenziellen Sicherheitsproblemen in Systemen, Netzwerken und Anwendungen, sowohl automatisiert als auch manuell.

Vorgehensweise

1. Einrichtung von regelmäßigen, automatisierten Scans auf Servern, Netzwerkgeräten und Anwendungen.
2. Manuelle Überprüfung spezifischer Schwachstellen, die automatisierte Tools möglicherweise übersehen oder Falsch bewertet haben (False Positives).
3. Analyse und Priorisierung gefundener Schwachstellen. (Schwachstellenbewertung)
4. Berichtserstellung und Empfehlungen zur Behebung.

Erfolgsfaktoren

- Einsatz bewährter Tools (z. B. Nessus, OpenVAS, Burp Suite).
- Kombinierter Ansatz aus automatisierten und manuellen Scans.
- Berichterstattung zur Erfassung aller entdeckten Schwachstellen.

Open-Source-Analysen

Zielsetzung

- Sicherstellung der Sicherheit und Lizenzkonformität von Open-Source-Komponenten.

Vorgehensweise

1. Einsatz von Tools zur Identifikation und Analyse von Open-Source-Komponenten (z. B. Snyk, Black Duck, trivy, Dependency Check).
2. Überprüfung auf bekannte Schwachstellen und Lizenzprobleme.
3. Priorisierung und Behebung kritischer Sicherheitslücken. (Schwachstellenbewertung)
4. Regelmäßige Aktualisierung der Open-Source-Software, um Sicherheitslücken zu schließen.

Erfolgsfaktoren

- Regelmäßige Überprüfung auf neue Schwachstellen.
- Konformität mit den Lizenzanforderungen.
- Dokumentation aller eingesetzten Open-Source-Komponenten.

Netzwerksicherheitsbewertungen

Zielsetzung

- Bewertung der Sicherheitskonfiguration und Schutzmechanismen der Netzinfrastruktur.

Vorgehensweise

1. Analyse der Netzwerkkonfiguration, Segmentierung und Zugangskontrollen.
2. Bewertung von Firewalls, VPNs und Intrusion-Detection-Systemen.
3. Durchführung von Netzwerksicherheits-Scans zur Identifizierung potenzieller Schwachstellen.
4. Erstellen eines Berichts zur Optimierung der Netzwerksicherheit.

Erfolgsfaktoren

- Aktualität der Sicherheitskonfigurationen.
- Einsatz von Intrusion-Detection- und Präventionssystemen.
- Überprüfung der Netzwerkprotokolle auf unzulässigen Zugriff.

Lückenanalysen

Zielsetzung

- Identifikation von Abweichungen zwischen dem aktuellen Sicherheitsstatus und anerkannten Standards oder Best Practices.

Vorgehensweise

1. Erstellung eines Vergleichs zwischen aktuellen Maßnahmen und Standards (z. B. ISO 27001, OWASP).
2. Identifizierung von Lücken und Risikobereichen.
3. Erstellen eines Maßnahmenplans zur Beseitigung identifizierter Schwachstellen.
4. Regelmäßige Überprüfung und Anpassung an neue Sicherheitsstandards.

Erfolgsfaktoren

- Orientierung an branchenspezifischen Standards.
- Dokumentation der Abweichungen und geplanten Maßnahmen.
- Stetige Anpassung der Sicherheitsmaßnahmen.

Überprüfung der physischen Sicherheit

Zielsetzung

- Sicherstellung der physischen Sicherheit von Anlagen, Einrichtungen und sensiblen Informationen.

Vorgehensweise

1. Bewertung von Zugangskontrollen (z. B. biometrische Systeme, Zutrittskarten).
2. Überprüfung der Überwachungssysteme und Sicherheitsprotokolle.
3. Durchführung von Sicherheitsschulungen für Personal in kritischen Bereichen.
4. Dokumentation aller Maßnahmen und Überprüfung der Wirksamkeit.

Erfolgsfaktoren

- Regelmäßige Inspektionen der Sicherheitsausrüstung.
- Strenge Zugangskontrollen zu sensiblen Bereichen.
- Schulung des Personals zu Sicherheitsrichtlinien.

Fragebögen und Software-Scans

Zielsetzung

- Sammlung detaillierter Informationen über Sicherheitspraktiken und Konfigurationen eingesetzter Software. Software-Scans um Sicherheitslücken, Schwachstellen oder Fehlkonfigurationen in einer Anwendung aufzudecken

Vorgehensweise

1. Erstellung und Verteilung von Fragebögen zu Sicherheitsrichtlinien und -konfigurationen.
2. Automatisierte Scans der Software auf Schwachstellen.
3. Dokumentation und Auswertung der Fragebögen und Scan-Ergebnisse.
4. Weitergabe der Ergebnisse an das verantwortliche Team zur Behebung.

Erfolgsfaktoren

- Umfassende Gestaltung der Fragebögen.
- Nutzung automatisierter Tools zur Schwachstellenerkennung. (BurpSuite etc.)
- Nachvollziehbare Dokumentation der Ergebnisse.

Quellcodeprüfungen

Zielsetzung

- Identifikation von Sicherheitslücken im Quellcode.

Vorgehensweise

1. Analyse des Codes mithilfe von statischen und dynamischen Codeanalysetools (z. B. SonarQube).
2. Manuelle Codeüberprüfung durch erfahrene Entwickler, sofern notwendig.
3. Dokumentation der entdeckten Sicherheitslücken und Kodierungsfehler.
4. Vorschläge zur Codeverbesserung und Fehlerbehebung.

Erfolgsfaktoren

- Einsatz von bewährten Tools zur Quellcodeanalyse.
- Verfügbarkeit erfahrener Entwickler für die manuelle Überprüfung.
- Regelmäßige Codeüberprüfungen, besonders vor Releases.

Szenariobasierte Tests

Zielsetzung

- Simulation realistischer Bedrohungsszenarien zur Bewertung der Reaktionsfähigkeit.

Vorgehensweise

1. Entwicklung spezifischer Bedrohungsszenarien (z. B. Phishing-Angriffe).
2. Durchführung der Tests zur Bewertung der Reaktionsfähigkeit des Systems und Personals.
3. Dokumentation der Ergebnisse und Analyse der Schwachstellen.
4. Schulungen und Optimierungen basierend auf den Testergebnissen.

Erfolgsfaktoren

- Realitätsnahe Bedrohungsszenarien.
- Schulung des Personals zur Reaktion auf Sicherheitsbedrohungen.
- Kontinuierliche Verbesserung der Reaktionsmaßnahmen.

Kompatibilitätstests

Zielsetzung

- Sicherstellung der Interoperabilität und Sicherheit von Systemen und Anwendungen in verschiedenen Umgebungen.

Vorgehensweise

1. Testen der Anwendung in verschiedenen IT-Umgebungen.
2. Sicherstellen, dass Sicherheitsprotokolle in allen Umgebungen greifen.
3. Überprüfung auf Abhängigkeiten oder Inkompatibilitäten.
4. Dokumentation und Anpassung der Systeme bei Kompatibilitätsproblemen.

Erfolgsfaktoren

- Breite Testabdeckung in unterschiedlichen Umgebungen.
- Analyse aller Systeme auf Interoperabilität.
- Dokumentation und Behebung aller identifizierten Inkompatibilitäten.

Leistungstests

Zielsetzung

- Sicherstellung, dass Systeme und Anwendungen unter Last stabil und sicher bleiben.

Vorgehensweise

1. Testen der Systeme unter verschiedenen Belastungen.
2. Identifizierung von Engpässen und Schwachstellen bei hoher Auslastung.
3. Dokumentation der Testergebnisse und Optimierung der Systemleistung.
4. Regelmäßige Durchführung, besonders bei System-Updates.

Erfolgsfaktoren

- Einsatz spezialisierter Tools zur Leistungsmessung.
- Fokussierung auf kritische Komponenten unter Last.
- Dokumentation und Analyse von Engpässen.

Die imbus AG hat für diese Testart einen eigenen spezialisierten Bereich.

End-to-End-Tests

Zielsetzung

- Sicherstellung des sicheren und fehlerfreien Ablaufs aller Systemkomponenten von Anfang bis Ende.

Vorgehensweise

1. Erstellen eines umfassenden Testplans für den gesamten Prozessablauf.
2. Durchführung der Tests zur Überprüfung der Integrität und Sicherheit.
3. Dokumentation aller Testergebnisse und Identifizierung von Optimierungsmöglichkeiten.
4. Anpassung der Prozesse basierend auf den Testergebnissen.

Erfolgsfaktoren

- Detailgenaue Planung und Testabdeckung.
- Überprüfung aller Schnittstellen und Datenflüsse.
- Dokumentation und Optimierung nach jedem Test.

Die imbus AG hat für diese Testart einen eigenen spezialisierten Bereich.

Penetrationstests

Zielsetzung

- Identifizierung von Schwachstellen durch simulierte, kontrollierte Angriffe.

Vorgehensweise

1. Definition des Testziels und Umfangs (z. B. Netzwerk, Webanwendungen).
2. Durchführung von Penetrationstests durch erfahrene ethische Hacker.
3. Dokumentation und Analyse der entdeckten Schwachstellen.
4. Priorisierte Behebung der identifizierten Schwachstellen und Sicherheitslücken.

Erfolgsfaktoren

- Durchführung durch erfahrene Penetrationstester.
- Regelmäßige Penetrationstests auf kritischen Systemen.
- Dokumentation zur Analyse und Nachverfolgbarkeit.