Dieser Beitrag ist mit freundlicher Unterstützung von Marius Rothenbücher, IT Security Consultant bei der SySS GmbH, entstanden, dem Marktführer für Penetrationstests.

Beispielhafte Vorgehensweise bei einem Penetrationstest

Aufgrund der Komplexität und Vielfalt von IT-Infrastrukturen und ihren Komponenten bieten sich vielfältige Gestaltungsmöglichkeiten von Penetrationstests an. Viele Dienstleister bieten aus diesem Grund unterschiedliche Testmodule an, die flexibel nach Wunsch zusammengestellt werden können. Trotz aller Variabilität ist der grundlegende Ablauf eines Penetrationstests meist ähnlich und, es gibt bestimmte Standardtestphasen, die in der Regel Bestandteile eines jeden Penetrationstests sind. Im Folgenden wird somit beispielhaft das methodische Vorgehen bei der Analyse einer Webserviceschnittstelle beschrieben.

ENUMERATION

Zu Beginn verschafft sich der Penetrationstester einen Überblick über die Funktionalität der Anwendung aus Sicht eines gewöhnlichen Nutzers, z. B. mithilfe der Bedienungsanleitung oder einer Dokumentation. Anschließend werden weitere Funktionen enumeriert, die nicht aus der Dokumentation oder Bedienung ersichtlich sind – diese haben oft ein höheres Angriffspotenzial, weil sie eventuell bei der Entwicklung als Debugging-Möglichkeit dienten oder administrative Funktionalitäten bereitstellen.

Zu den Enumerationsmethoden zählen z. B. eine Whois-Abfrage, über die ermittelt werden kann, wer der Eigentümer einer Domain ist bzw. wer eine bestimmte Webseite oder Anwendung hostet, oder die Ermittlung von erreichbaren TCP- und UDP-Diensten, die auf bestimmten Ports „lauschen“, um Daten mit einem Client auszutauschen. Die Erreichbarkeit von TCP- und UDP-Diensten kann mit einem Portscan geprüft werden.

Weiter geht es mit der Software, die die unterschiedlichen Dienste bereitstellt. Die Software erwartet Daten in einem bestimmten Format, was wiederum in einem Protokoll standardisiert ist (z. B. HTTP, FTP oder SMTP). Beim Austausch dieser Daten werden häufig der Name und die Version der eingesetzten Software übertragen. An dieser Stelle kann direkt nach bekannten Exploits gesucht und ggf. erste Schwachstellen abgeleitet werden. Unbekannte Schwachstellen sind allerdings öfter in spezifischen Eigenentwicklungen zu finden, die im Gegensatz zu häufig eingesetzter Standardsoftware noch gar nicht oder nicht so intensiv geprüft wurde.

Über das von Webdiensten genutzte HTTP-Protokoll und die vorgesehene URI (Unique Resource Identifier) kann – meist automatisiert – nach API-Endpunkten, Dateien und Skripten gesucht und auf diese Weise weitere Funktionalitäten identifiziert werden.

SCHWACHSTELLENERKENNUNG

Sobald der Penetrationstester die Funktionalität des Systems maximiert und verstanden hat, wird er nach Schwachstellen suchen, um so bestimmte Programmabläufe zu kontrollieren. Teile des Systems gelten dann als kompromittiert.

Die Beeinflussung eines Systems erfolgt über Eingaben, die mit Payloads so optimiert sind, dass sie zu Fehlinterpretationen seitens des Systems führen. Anhand von unterschiedlichen Antworten bezüglich Inhalt, Größe oder Antwortdauer können Rückschlüsse auf die interne Verarbeitung gezogen werden. So erhofft sich der Penetrationstester, sogenannte Injection-Schwachstellen zu finden.

Bei einer **Injection-Attacke** wird eine Abfrage- oder Auszeichnungssprache angegriffen. Bekannte Abfragesprachen sind beispielsweise LDAP und SQL. Bekannte Auszeichnungssprachen sind beispielsweise CSS, HTML, Templates und XML. Abfragesprachen beschreiben, welche Daten aus einer bestimmten Quelle (z. B. Datenbank oder Dateisystem) geladen werden sollen. Um nur die im aktuellen Kontext relevanten Daten zu erhalten, gibt es Filter- und Sortiermechanismen. Werden Benutzereingaben nicht korrekt von der eigentlichen Abfrage getrennt oder maskiert, kann ein Angreifer die Abfrage manipulieren und weitere Daten aus der Quelle auslesen (z. B. SQL Injection).

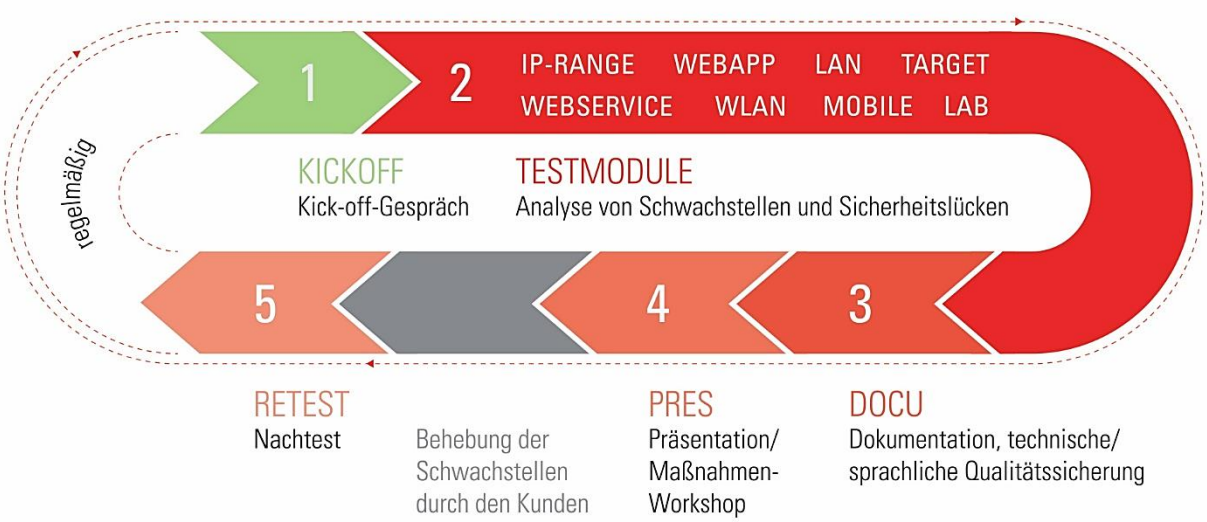
Auszeichnungssprachen beschreiben, wie die Daten aus einer bestimmten Quelle (z. B. Datenbank, Benutzereingaben) strukturiert/dargestellt werden sollen. Um die Daten an bestimmten Stellen in der Struktur einzubinden, gibt es Direktausgaben oder Platzhalter. Häufig handelt es sich bei diesen Daten auch um Benutzereingaben. Werden Benutzereingaben nicht korrekt maskiert, kann ein Angreifer die Auszeichnungssprache und die Struktur sowie die Darstellung manipulieren (z. B. Cross-Site Scripting).

Falls die Abfrage- oder Auszeichnungssprache von einem serverseitigen Parser interpretiert wird und dieser zusätzliche Funktionalitäten wie das Auflisten von Verzeichnissen, Lesen von Dateien oder die direkte Ausführung von Konsolenbefehlen ermöglicht, können diese vom Angreifer ebenfalls ausgenutzt werden. XML-Parser sind beispielsweise bekannt für sogenannte XXE-Schwachstellen (XML External Entity), bei denen unter Umständen auf sensible Systemdateien zugegriffen und der Inhalt von Verzeichnissen aufgelistet werden kann.

SCHWACHSTELLENAUSNUTZUNG

Falls nun aktiv ausnutzbare Schwachstellen gefunden werden, können diese für die Beschaffung weiterer Informationen über potenzielle Schwachstellen verwendet werden. Wenn bereits ausreichend Informationen und Rechte eingesehen wurden und kein weiterer Informationsgewinn durch die Ausnutzung von Schwachstellen besteht, wird der Penetrationstester auf die Ausnutzung der bereits gefundenen Schwachstellen verzichten.

ABLAUFSCHEMA EINES PENETRATIONSTESTS



IP-RANGE	Analyse Ihrer z. B. über das Internet erreichbaren Adressbereiche oder ausgewählter interner Systeme
WEBAPP	Prüfung Ihrer Webapplikation aus verschiedenen Angreiferperspektiven
WEBSERVICE	Detaillierte Untersuchung Ihrer angebotenen Webservices (z. B. SOAP, REST)
LAN	Analyse Ihrer Systeme aus dem lokalen Netz z. B. Reinigungspersonal-/Praktikantenszenario, Client-, VoIP-, VLAN-, Active Directory-, SAP-Analyse, Prüfung von Produktionsanlagen und kritischer Infrastrukturen
TARGET	Prüfung der Anfälligkeit Ihrer Clients für zielgerichtete Angriffe aus dem Internet
WLAN	Prüfung Ihrer WLAN-Infrastruktur
MOBILE	Detaillierter Sicherheitstest von mobilen Apps und Geräten (z. B. iOS, Android) sowie Mobile-Device-Management-Lösungen
LAB	Individuelle Labortests: Software-, Hardware- und Geräteprüfungen, Internet-of-Things-Produkttests