Werkzeuge für das Aufspüren von Schwachstellen

Gut gesucht

Martin Wundram

Webapplikationen und komplexe Homepages sind mittlerweile kaum noch „händisch" zu testen. Wertvolle Unterstützung für schnellere und bessere Ergebnisse bieten Web-Application-Scanner. Diese Marktübersicht stellt 13 Produkte im Detail vor.

Web-Application-Scanner, auch Web-Security- oder Webschwachstellen-Scanner genannt, dienen dem Testen von Webapplikationen auf Sicherheitslücken und Angriffe wie SQL Injection, Cross-Site Scripting (CSS oder XSS), Cross-Site Request Forgery (XSRF), auf Konfigurationsfehler, veraltete Software, Information Disclosure und vieles mehr (s. Glossar). Zum einen gibt es vollautomatische Scanner, die selbstständig Webapplikationen untersuchen und ohne vorherige Anpassung versuchen, so viele Probleme wie möglich aufzuspüren. Zum anderen existieren teilautomatische, bei denen der Benutzer zu testende Teilbereiche und Profile auswählt und für individuelle Tests anpasst.

Eine besonders enge „Zusammenarbeit" zwischen Scanner und Anwender entsteht, wenn das Programm über einen eingebauten Browser oder als Proxy einzelne Webseitenaufrufe abfängt, diese dann zur Suche nach Sicherheitslücken verändert, erneut an die Webapplikation sendet und die Antworten auswertet (zum Beispiel Replay, Fuzzing oder Brute Force).

Audits können sich auf die reine Webapplikation beschränken oder die darunterliegenden Ebenen einbeziehen (Abb. 1). Einige Web-Application-Scanner bieten dazu Port-Scanner und Vulnerability-Datenbanken. Außerdem unterstützen die Werkzeuge die einzelnen Phasen eines Webscans in unterschiedlichem Umfang. Der vermeintlich wichtigsten Phase, dem Finden von Schwachstellen, gehen die nicht minder wichtigen Phasen der Planung und Vorbereitung voraus (Abb. 2).

Und die besten Funde nutzen nichts, wenn der Experte sie nicht ausreichend dokumentiert und bewertet. Schließlich stehen meist Folgetests an und dann steht man vor der Herausforderung, vergangene Testläufe und -ergebnisse effizient im aktuellen Audit berücksichtigen beziehungsweise auf ihnen aufbauen zu müssen. Sicherheitsscans von Webapplikationen werden hauptsächlich während der Live-Phasen einer Webapplikation durchgeführt, decken aber auch die Fehlerklassen ab, die bereits in Design und Entwicklung der Software entstanden sind (Abb. 3).

Verschieden komplexe Werkzeuge

Es gibt bei den vorgestellten Produkten verschiedene Architekturentwürfe. Am stärksten verbreitet sind Standalone-Scanner, die ein einzelner Anwender bedient. Client-Server-Scanner sind komplexer, bieten aber insbesondere Mehrbenutzerfähigkeit. Das Produkt von Greenbone beispielsweise gibt es als Appliance und Virtual Appliance, die angepasste Versionen unter anderem von OpenVAS (der freien Weiterentwicklung von Nessus) sowie w3af (einem freien Audit-Framework) mit ergänzenden Services zu einem Gesamtpaket bündelt. Seine Bedienung erfolgt wahlweise über ein Web-, Desktop- oder Command-Line-Interface. Besonders schlank, jedoch vergleichsweise funk-

tionsreduziert sind Scanner, die vollständig als Browser-Plug-in arbeiten.

Eine wichtige Komponente vieler Scanner ist ein eingebauter Proxy. Diesen stellt der Auditor optional im Browser ein, damit er alle Requests und Responses an den Scanner leitet. So kann etwa w3af als Stand-alone-Produkt nur mit seinen eingebauten Testmodulen betrieben oder zusätzlich über dessen Proxy von einem Browser mit einzelnen Requests „versorgt“ werden, die er an seinen Fuzzer weiterleitet.

Der Proxy-Ansatz hat gleich mehrere Vorteile: Der Anwender kann den Scanner „durch die Webapplikation führen“. Außerdem laufen komplexe Techniken wie Java, JavaScript, Flash und Silverlight im Browser ab. Der Scanner muss diese Techniken nicht oder nicht vollständig unterstützen, erkennt jedoch die daraus resultierenden Requests und Responses und kann darauf eigene Tests aufsetzen. So sind zwar die Burp Suite und der Zed Attack Proxy Stand-alone-Produkte, machen den Proxy-Ansatz aber zu ihrem zentralen Element. Der Greenbone Security Manager hingegen erledigt seine Tests intern. Auch Websecurify bietet noch keinen Proxy, was sich aber in der nächsten Version ändern soll.

Glossar

Browser Hijacking: Manipulieren von Browsern, indem ein Programm („Browser Hijacker“) Seitenaufrufe und Suchanfragen auf bestimmte Webseiten umleitet.

Brute Force: Angriffsmethode, die darauf basiert, alle Möglichkeiten durchzuprobieren oder zumindest möglichst viele in möglichst kurzer Zeit (etwa bei Passwörtern).

Cross-Site Request Forgery (CSRF oder XSRF): Angriff auf eine Webanwendung, indem man dem Browser eines angemeldeten Benutzers einen manipulierten Request unterschiebt. Dieser Request wird dann „im Namen“ des angemeldeten Benutzers ausgeführt (etwa das verdeckte Hinzufügen eines weiteren Administrator-Accounts mit Passwort).

Cross-Site Scripting (CSS oder XSS): Das Unterschieben von eigenem Code über eine Schnittstelle (etwa ein Webformular), die den vermeintlich vertrauenswürdigen Inhalt ungeprüft übernimmt.

Fuzzer: Programm, das die Zufallsdaten erzeugt.

Fuzzing: Methode, bei der zufällig erzeugte Daten einer Anwendung über die Eingabeschnittstelle zugeführt werden, um ihre Robustheit oder Fehleranfälligkeit zu prüfen.

Information Disclosure: Das (beabsichtigte oder unbeabsichtigte) Preisgeben vertraulicher oder sicherheitskritischer Informationen.

Replay-Angriff: Eine vergangene Aktion wird erneut ausgeführt. Beispielsweise das Vortäuschen einer fremden Identität durch Wiedereinspielen zuvor aufgezeichneter Authentizitätsinformationen.

Spidern/Crawlen: Automatisches Durchforsten des Internet oder einer konkreten Webapplikation nach Webseiten und Sammeln der dort gefundenen Informationen.

SQL Injection: Einschleusen eigener SQL-Befehle über eine Eingabeschnittstelle (z. B. Webformular), die Benutzereingaben ungeprüft übernimmt.

Throttling: Auslassen oder Überspringen von Takten bei Prozessoren, um die Last zu regulieren. Bei Webapplikations-Scannern ist hier hauptsächlich das Regulieren gleichzeitig durchgeführter Anfragen/Tests gemeint, das die verwendete Bandbreite auf dem Testserver steuern soll.

Unterstützung durch automatisierte Scans

Alle Produkte bieten einen vollautomatischen Modus, der nach einigen vorbereitenden Handgriffen zu ersten Ergebnissen führt. Die Burp Suite setzt jedoch im Browser ein händisches Ansurfen der zu testenden Webapplikation voraus. Der Burp Proxy schneidet dazu alle Requests mit und bietet die Möglichkeit, einzelne Websites zu „spidern“ und anschließend automatisiert zu testen. Andere Scanner benötigen diesen einfachen vorbereitenden Schritt nicht.

Der vorliegende Marktüberblick stellt Web-Application-Scanner vor, die als Desktop- oder Appliance-Stand-alone-Produkt eigenständig arbeiten oder als Proxy zwischen Test-Browser und Test-Webapplikation einzubinden sind. Außerdem beherrscht jeder Scanner Standard-Vulnerability-Klassen wie XSS, CSRF oder SQL Injection und bietet einen vollautomatischen Testmodus.

Ein weiteres Kriterium: Die Produkte werden angeboten und betreut von einem Unternehmen oder einer über das Internet erreichbaren Gruppe. Alle Angaben in Artikel und Tabelle beziehen sich auf einen umfangreichen Fragebogen, den neun Open-Source-Projekte und 14 Unternehmen erhielten. Insgesamt füllten 13 Anbieter ihn aus. Insbesondere von amerikanischen Unternehmen fehlen Rückmeldungen, vermutlich da sie sich auf andere Märkte konzentrieren.

Vergleich kaum möglich

Die jeweiligen Scanner differenziert und trennscharf vorzustellen ist nicht einfach. Zwar bieten alle Scanner Tests auf die Standard-Vulnerability-Klassen. Die wahre Stärke eines Scanners liegt aber nicht in der Vielzahl generischer Testverfahren, sondern in der Wahrscheinlichkeit, einen beliebigen, vorher nicht bekannten Fehler tatsächlich zu finden. Vor dieser täglichen Herausforderung stehen auch alle Virenscanner. Hier hilft nur der wertende Vergleich anhand definierter Testumgebungen, um daraus Abschätzungen für die „Gründlichkeit“ in echten Audits herzuleiten.

Einen guten und sehr umfangreichen Vergleichstest hat 2011 Shay Chen durchgeführt (siehe „Alle Links“). Ein weiterer empfehlenswerter Vergleichstest aus dem Jahr 2010 stammt von

iX-TRACT

- Noch immer gehören unsichere Webapplikationen für Cyberkriminelle zu den größten Einfallstoren in nachgelagerte Systeme. Audits mit speziellen Applikationsscannern können helfen, die Schwachstellen aufzudecken.
- Es existieren zahlreiche freie und kommerzielle Werkzeuge mit unterschiedlichem Funktionsumfang und unterschiedlichen Architekturansätzen. Werkzeuge mit eigenem Proxy können besonders punkten.
- Bis man genaue Kenntnis der verschiedenen Scanner-Funktionen hat und sich über die eigenen Anforderungen im Klaren ist, kann es hilfreich sein, erste Schritte mit kostenlosen Werkzeugen durchzuführen.

Kommerzielle und freie Web-Application-Scanner

Hersteller/Projekt	Casaba	Greenbone	HP	IBM
Produkt	Watcher and X5S bundle	Greenbone Security Manager	HP WebInspect	Rational AppScan Standard
Homepage	www.casaba.com	www.greenbone.net	www.fortify.com	www.ibm.com
Produktinformationen und Support				
Whitepaper/Produktbeschreibung (s. „Alle Links“)	Wiki	✓	✓	✓
Handbuch (s. „Alle Links“)	✓	✓, plus Learning Center	✓	✓
individueller Support	✓, E-Mail	✓, nach SLA	✓, WWW, Telefon, E-Mail	✓, über IBM Support
Community-Support	✓	✓	✓	✓
Basismerkmale				
Plattform	Windows	Appliance, Virtual Appliance (VBox, ESXi, VMware)	Windows	Windows
Mehrbenutzerfähigkeit	–	✓	✓ [2]	–
Produktarchitektur	Stand-alone Fiddler Plug-in	Appliance und Frontend	WebInspect: Stand-alone, AMP: Server-Client	Stand-alone
Quellcode verfügbar	✓	✓	–	–
vollautomatischer Modus	✓	✓	✓	✓
Features				
getestete Problemklassen	alle relevanten	alle relevanten	alle relevanten	alle relevanten
Test auf Information Disclosure	✓	✓	✓	✓
Vulnerability-Datenbank	–	✓, sehr umfangreich, täglich aktualisiert	✓, sehr umfangreich, täglich aktualisiert	✓, sehr umfangreich, wöchentlich aktualisiert
eingebauter Expertenrat	✓	✓	✓	✓, z. T. mit Videos
Portscanner	–	✓	✓	–
Brute-Force-Modul	–	✓	✓	–
Fuzzer	✓ (x5s)	✓, über integriertes w3af	✓	✓
passiver Testmodus	✓ (Watcher)	✓	✓	✓
vorgefertigte Testprofile	✓	✓	✓	✓
eigene Profile anlegen	✓	✓	✓	✓
GUI	✓	✓	✓	✓
Text-Konsole	–	✓	✓	✓
Diagnose-Plug-in zur Integration in Webanwendung	–	–	✓	✓, Glass-Box-Plug-in
unterstützte Protokolle	HTTP, HTTPS, Compression	alle relevanten	alle relevanten, WS-Security	HTTP, HTTPS, Ö
Upstream-Proxy unterstützt	✓	✓	✓	✓
Authentifikationsmöglichkeiten	alle relevanten	Basic Auth, Form Auth, Session Cookie	viele, auch Client-Side Certificates	viele, auch per Recording
Session-Management	✓, über Browser	✓	✓, auch Login-Macros	✓
Crawler	–	–	✓, inkl. externem Input, Ajax-kompatibel	✓, auch Ajax-kompatibel
Visualisierung der Webanwendung	–	–	✓	✓
vom Parser unterstützte Techniken	k. A.	k. A.	viele, insb. Silverlight	Flash, JavaScript
unterstützte Content Encodings	alle	UTF-8	viele, insb. UTF-8	viele, insb. UTF-8
Request-Manipulation nach Browser-Hijacking	✓, umfangreich, über Proxy	–	✓, umfangreich, z. B. über Proxy	✓, über Proxy
HTTPS-Unterstützung	✓	✓	✓	✓
False-Positives markierbar	✓	✓	✓, und Ignorieren in Folgetests	✓
Pausieren/Wiederaufnehmen von Testläufen	✓, ✓	✓, ✓	✓, ✓	✓, ✓
Unterstützung von Delta-Tests	✓	✓	✓	✓
eingebautes Throttling	✓	–	✓	✓
Scheduler zur Test-Terminierung	–	✓	✓	✓
Integration				
Headless-Betrieb	–	✓	✓	✓
API	✓, Einbinden neuer Tests	✓ (OMP), umfangreich	– für WebInspect, ✓ für HP AMP	✓
Reports	✓, HTML, XML, Team Foundation Server	✓, PDF, XML, HTML, TXT, LaTex, NBE	✓, HTML, PDF, RAW (nativ), RTF (Word), TXT, XLS (Excel)	✓, Word und PDF
anpassbare Reports	–	✓	✓	✓
Compliance-Reports	–	✓, PCI-DSS, IT-Grundschutz	✓, 30 Vorlagen, eigene möglich	✓, über 20 Vorlagen
internes Ticketing-/Tracking-System	–	–	–	✓
Anbindung an Ticketing-/Tracking-System	✓, Integration Team Foundation Server	✓ (via Escalation)	✓	✓
Preise und Verschiedenes				
Preise verschiedener Produkte	kostenfrei	nach Anzahl zu testender IPs	nach Anzahl Nutzer	ca. 18 000 € inkl. Wartung/Support
Bindung an Dongle/Seriennummer/Aktivierung	–, –, –	–, –, –	–, ✓, ✓	–, –, –
deutsche Niederlassung	–	✓	✓	✓
Produkt als SaaS	–	✓, über Parter	✓	✓

✓ ja/vorhanden/trifft zu; – nein/nicht vorhanden/trifft nicht zu; k. A. keine Angabe; [1] für Android und Java geplant; [2] über Web-Interfaces in HP AMP; [3] erster Seitenabruf per externem Browser notwendig; [4] aber (noch) kein eingebauter Crawler; [5] in Kombination mit HIAB; [6] zum Beispiel google.com, GHDB und xssed.com

	Mavituna	N-Stalker	NTO	Outpost	Owasp
	Netsparker	N-Stalker Web Application Security Scanner 2012	NTOSpider, NTO Enterprise	WAS Web Application Scanner	OWASP Zed Attack Proxy
	www.mavitunasecurity.com	www.nstalker.com	www.ntobjectives.com	www.outpost24.com	www.owasp.org
	✓	✓	✓	✓	✓
	✓	✓	✓	✓	✓
	✓, WWW, Telefon, E-Mail	✓, WWW, Telefon, E-Mail	✓, Telefon, E-Mail	✓, 24 × 7	✓, E-Mail, User Groups
	✓	✓	–	–	✓
	Windows	Windows	Windows	Linux (Appliance), VMware Virtual Appliance (Linux)	Windows, Linux, Mac, Ö (Java)
	✓	–	✓	✓	–
	Stand-alone	Stand-alone	Stand-alone und Client-Server	Appliance und Frontend	Stand-alone
	–	–	–	–	✓
	✓	✓	✓	✓	✓
	alle relevanten	alle relevanten	alle relevanten	alle relevanten	alle relevanten
	✓	✓	✓	✓	✓ (DirBuster)
	✓	✓, sehr umfangreich, täglich aktualisiert	–	✓, sehr umfangreich, täglich aktualisiert	–
	✓	✓	✓	✓	✓
	–	geplant	✓	✓	✓
	✓	✓	✓	✓	✓ (DirBuster)
	–	✓	✓	–	✓
	✓	✓	✓	–	✓
	✓	✓	✓	✓	–
	✓	✓	✓	✓	–
	✓	✓	✓	✓	✓
	✓, GUI wird mitgestartet	✓	✓	–	✓
	–	–	geplant für 2012	–	–
	alle relevanten	alle relevanten	alle relevanten	alle relevanten	HTTP 1.1, gzip compression
	✓	✓	✓	✓ [5]	✓
	viele, auch Kerberos, NTLM und Zertifikate	viele, auch NTLM und Zertifikate	viele, auch Kerberos, NTLM und Zertifikate	viele, auch NTLM und Zertifikate	viele, auch NTLM und Zertifikate, Smartcards
	✓, auch Login-Macros und CAPTCHA-Support	✓	✓, auch Login-Macros	✓	✓, auch Berücksichtigung von CSRF-Formularschutz
	✓, auch Ajax-kompatibel	✓, auch Ajax-kompatibel, inkl. eigener JavaScript-Engine	✓, auch Ajax-kompatibel	✓, auch Ajax-kompatibel	✓, (noch) nicht Ajax-kompatibel
	✓	✓	✓	✓	–
	viele, Flash über int. Proxy	viele, außer Java	viele	JavaScript, XML, CSS	k. A.
	viele, insb. UTF-8	viele, insb. UTF-8	viele, insb. UTF-8	k. A.	UTF-8
	✓, umfangreich über Proxy	✓, umfangreich über Proxy	✓, eingebauter Browser	–	✓, umfangreich
	✓	✓	✓	✓	✓
	–	✓	✓	✓, inkl. Verifikation	✓
	✓, ✓	✓, ✓	✓ (ab 2012)	–	nur pausieren
	✓	geplant	✓ (ab 2012)	✓	–
	–	✓	✓	✓	✓
	✓	✓, teilweise	✓	✓	–
	✓	✓	✓	✓	✓
	✓, für Reporting und Authentifizierung	–	✓, Basisfunktionen wie Start/Stop	✓, umfangreich	✓, umfangreich (REST API)
	✓, HTML, PDF, XML, CSV	✓, PDF, RTF	✓, HTML, PDF, XML	✓, XLS, XML	✓, HTML, XML
	✓	✓	–	✓	–, Inhalte können angepasst werden
	✓, mehrere Vorlagen	✓, OWASP Top 10	✓, mehrere Vorlagen	✓, PCI-DSS	–
	–	–	✓	✓	–
	–	–	✓	✓	–
	k. A.	ca. 500–6000 US-$, nach Anzahl zu testender Webseiten	ab 25 000 US-$ jährlich	nach Anzahl zu testender IPs/ Webapplikationen	kostenfrei
	–, ✓, –	–, ✓, ✓	–, –, –	–, ✓, ✓	–, –, –
	–	–	–	✓	✓
	–	✓	✓	✓	–

Fortsetzung: Kommerzielle und freie Web-Application-Scanner

Hersteller/Projekt	Portswigger	w3af	Watobo	Websecurify
Produkt	Burp Suite Professional	w3af	Web Application Toolbox	Websecurify
Homepage	portswigger.net/burp/	www.w3af.org	watobo.sourceforge.net	www.websecurify.com
Produktinformationen und Support				
Whitepaper/Produktbeschreibung (s. „Alle Links“)	–	–	–	✓
Handbuch (s. „Alle Links“)	✓	✓	✓	✓
individueller Support	✓, E-Mail	–, aber Mailing-Liste	✓, E-Mail	✓
Community-Support	–	✓	–	✓
Basismerkmale				
Plattform	Windows, Linux, Mac, Ö (Java)	Linux, Windows, BSD	Windows, Linux, Mac, Ö (Ruby)	Windows, Linux, Mac OS X, Apple iOS (iPhone), Google Chrome, Mozilla Firefox [1]
Mehrbenutzerfähigkeit	–	–	–	✓
Produktarchitektur	Stand-alone	Stand-alone	Stand-alone	Mobile, Desktop, Client-Server und Browser-Plug-in
Quellcode verfügbar	–	✓	✓	✓
vollautomatischer Modus	✓ [3]	✓	✓ [4]	✓
Features				
getestete Problemklassen	alle relevanten	alle relevanten	alle relevanten, Fokus auf manuellen Tests	alle relevanten
Test auf Information Disclosure	✓	✓	✓ (z. B. mit DirBuster)	✓
Vulnerability-Datenbank	–	–, aber einzelne Module können spez. Vulns. enthalten	–, Anbindung z. B. an Nikto-DB möglich	–
eingebauter Expertenrat	✓	✓, aber nicht umfassend	✓, aber nicht umfassend	✓, auch anpassbar
Portscanner	–	–	–	–
Brute-Force-Modul	✓	✓	✓, per Fuzzer	✓, noch nicht manuell
Fuzzer	✓	✓	✓	✓, noch nicht manuell
passiver Testmodus	✓	✓	✓	✓
vorgefertigte Testprofile	–	✓	✓	–
eigene Profile anlegen	–	✓	✓	–
GUI	✓	✓	✓	✓
Text-Konsole	✓	✓	in Entwicklung	✓
Diagnose-Plug-in zur Integration in Webanwendung	–	✓	–	–
unterstützte Protokolle	alle relevanten	HTTP 1.1, Gzip Compression, E-Tag	HTTP 1.1	alle relevanten
Upstream-Proxy unterstützt	✓	✓	✓	✓
Authentifikationsmöglichkeiten	viele, auch NTLM und Zertifikate	Basisverfahren	Basisverfahren und Client-Zertifikate	viele, auch NTLM, Zertifikate, Smartcards
Session-Management	✓, auch Makros, Logout-Detection und Berücksichtigung von CSRF-Formularschutz	✓, über verschiedene Plug-ins	✓, auch Makros, Logout-Detection und Berücksichtigung von CSRF-Formularschutz	✓
Crawler	✓, auch Ajax-kompatibel	✓, jedoch nicht Ajax/JavaScript-kompatibel, aber befragt auch externe Quellen [6]	–, noch nicht, aber Anbindung externer Crawler möglich	✓, auch Ajax-kompatibel (experimentell)
Visualisierung der Webanwendung	✓	✓	–	✓, als Erweiterung
vom Parser unterstützte Techniken	viele	XML, CSS	k. A.	viele, auch PDF-Parsing
unterstützte Content Encodings	viele, insb. UTF-8	viele, insb. UTF-8	viele, insb. UTF-8	viele, insb. UTF-8
Request-Manipulation nach Browser-Hijacking	✓, umfangreich	✓, umfangreich	✓, umfangreich	✓, umfangreich
HTTPS-Unterstützung	✓	✓	✓	✓
False-Positives markierbar	✓	✓	✓	–
Pausieren/Wiederaufnehmen von Testläufen	✓	nur pausieren	nur pausieren	✓
Unterstützung von Delta-Tests	✓	–	geplant	–
eingebautes Throttling	✓	–	✓	✓, aber nicht anpassbar
Scheduler zur Test-Terminierung	✓	–	–	–
Integration				
Headless-Betrieb	✓	✓	geplant	✓
API	✓, umfangreich	–	–	✓, umfangreich
Reports	✓, HTML, XML	✓, HTML, XML, TXT	–	✓, HTML, CSV, RTF, XML, JSON
anpassbare Reports	✓	✓	–	–
Compliance-Reports	–	–	–	–
internes Ticketing-/Tracking-System	–	–	–	–
Anbindung an Ticketing-/Tracking-System	–	–	–	–
Preise und Verschiedenes				
Preise verschiedener Produkte	225 € pro Jahr (Basisversion kostenfrei)	kostenfrei	kostenfrei	mobile 17 US-$, Desktop 240 US-$, Server individuell
Bindung an Dongle/Seriennummer/Aktivierung	–, ✓, ✓	–, –, –	–, –, –	–, –, –
deutsche Niederlassung	–	–	–	–
Produkt als SaaS	–	–	–	in Vorbereitung

✓ ja/vorhanden/trifft zu; – nein/nicht vorhanden/trifft nicht zu; k. A. keine Angabe; [1] für Android und Java geplant; [2] über Web-Interfaces in HP AMP; [3] erster Seitenabruf per externem Browser notwendig; [4] aber (noch) kein eingebauter Crawler; [5] in Kombination mit HIAB; [6] zum Beispiel google.com, GHDB und xssed.com

Larry Suto (siehe „Alle Links“). Darüber hinaus liefern die „Web Application Security Scanner Evaluation Criteria“ (WASSEC) des Web Application Security Consortium eine sinnvolle und strukturierte Referenz für den Vergleich und die Evaluation von Web-Application-Scannern.

Die in Web-Application-Scanner eingebauten Vulnerability-Datenbanken betrachtet die Fachwelt durchaus kritisch. Diese Datenbanken enthalten Tests für konkrete und bekannte Sicherheitslücken spezifischer Webapplikationen (zum Beispiel „SQL-Injection in Datei XY der Forensoftware ABC in Version 47.11 für Parameter 'example'“). Beispielsweise bringen HP WebInspect und der Web Application Security Scanner 2012 solche Datenbanken mit. Der Vorteil ist ein effizienter und treffgenauer Test auf alle aktuell bekannten Lücken in verbreiteten Anwendungen. Es droht jedoch ein falsches Sicherheitsgefühl, wenn der Scanner keine Treffer findet – wo er doch auf so viele Lücken testet. Außerdem können sich Testläufe unnötig verzögern, wenn stumpf die ganze Vulnerability-Datenbank durchgetestet wird.

Integrierte Portscanner können bei der Suche nach nicht offensichtlichen Webapplikationen auf ungewöhnlichen Ports hilfreich sein. Sie gehören aber nicht zum Kern eines Webscanners und externe Programme wie *nmap* übernehmen diese Funktion problemlos. Die Mehrheit der hier vorgestellten Produkte enthält keinen eingebauten Portscanner.

Ajax und andere Schwierigkeiten

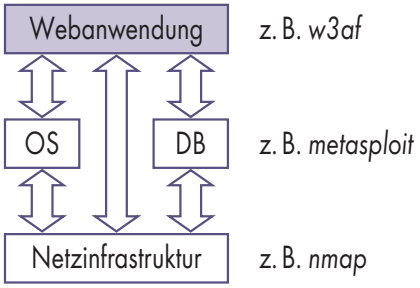
Die Preise der vorgestellten Produkte reichen von 0 Euro für Open-Source-Lösungen bis circa 18 000 EUR jährlich für kommerzielle Produkte inklusive Wartung und Support. Das teuerste Programm muss aber nicht immer zwingend das beste sein. Kostenlose und preiswerte Web-Application-Scanner können sogar sehr gut sein und je nach Einsatzzweck alle wesentlichen Anforderungen erfüllen (siehe Vergleich von Acunetix Web Vulnerability Scanner, Burp Suite Pro und w3af in [1]). Die preislichen Schwergewichte wie Rational AppScan und NTOSpider bieten zumindest das umfassendste Leistungsspektrum und im Falle von IBM neben deutschsprachiger Software auch deutschsprachigen Support und eine Niederlassung in Deutschland.

Das Web 2.0 hat große Auswirkungen auf die Web-Application-Scanner. Zum einen müssen diese eine zu untersuchende Webapplikation mit einem Crawler oder Spider möglichst vollständig erfassen, also jede serverseitige Ressource finden. Zum anderen müssen sie die erfassten Ressourcen inhaltlich testen. Die Mehrheit der Produkte enthält Crawler, die mit Ajax zurechtkommen. HP WebInspect und Watobo Web Application Toolbox berücksichtigen auch den Input extern angebundener Crawler. w3af befragt sogar externe Suchmaschinen wie Google und GHDB, unterstützt dafür aber kein Ajax. Watobo hat noch keinen Crawler eingebaut. Damit der Crawler die gesamte Webapplikation erkennen und durchsuchen kann, sollte das Produkt möglichst viele Authentifizierungsmöglichkeiten unterstützen und neben selbstständigem Login/Re-Login auch auf CSRF-Formularschutz und geänderte Session-IDs reagieren können.

Nützliche Tipps zur Fehlerbehebung

Alle Produkte lassen sich auch in einem passiven Testmodus betreiben, der keine aktiven Angriffe durchführt. Während der Crawler läuft, testet der Scanner auf unmittelbar erkennbare Probleme wie Information Disclosure (etwa identifizierbare E-Mail-Adressen oder Kreditkartennummern), unsichere Session-Parameter oder fehlende Verschlüsselung.

Einige Produkte setzen einen Schwerpunkt auf Nutzerkomfort und eingebaute Expertenfunktionen, die zu den jeweiligen Lücken eine Erklärung und Klassifizierung liefern (kritisch oder harmlos) sowie zusätzliche Tipps zur Behebung. Hier können alle vorgestellten Produkte punkten, Rational AppScan liefert sogar zum Teil Videos mit.



Ein erfolgreicher Angriff auf die Webanwendung kann Cyberkriminellen den Zugriff auf die nachgelagerten Systeme ermöglichen. Mit den Schwachstellen-Scannern kann man auf verschiedenen Ebenen nach Sicherheitslücken suchen (Abb. 1).

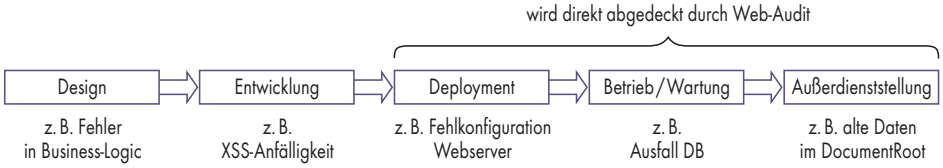
Alle Produkte prüfen die verbreiteten Angriffstechniken. Da ein Detailvergleich den Rahmen dieser Marktübersicht sprengen würde, sei auf die unter „Alle Links“ zu findenden Whitepapers und Produktinformationen verwiesen. Wichtiger Bestandteil von Web-Application-Scannern ist ein leistungsfähiger Fuzzer zur (teil)automatisierten Manipulation von Requests. Ein fehlendes Brute-Force-Modul kann meist zumindest in Grundzügen mit dem Fuzzer nachgebaut werden. Brute-Force-Techniken sind neben Passworttests beispielsweise auch für die Suche nach SQL-Injection-Schwachstellen nützlich. Hierzu legt der Anwender manuell eine Liste von Teststrings und zu testende Parameter fest. Um auch SSL-/TLS-gesicherte Verbindungen im Proxy-Betrieb verarbeiten zu können, generieren die jeweiligen Produkte ein eigenes Root-CA-Zertifikat und hören die Übertragung als Man-in-the-Middle ab.

Weitere Scan-Werkzeuge

Produkt	Hersteller/Projekt	Webseite
Freie Software		
Arachni	Arachni	arachni-scanner.com
Paros	Chinotec Technologies Company	www.parosproxy.org
Skipfish	Google	code.google.com/p/skipfish
Kommerzielle Produkte		
Acunetix Web Vulnerability Scanner	Acunetix	www.acunetix.com
Chorizo	Mayflower	https://chorizo-scanner.com
Hailstorm	Cenzic	www.cenzic.com
Nessus	Tenable Network Security	tenable.com
NeXpose	Rapid7	www.rapid7.com
ParosPro	Milescan Technologies	www.milescan.com
Retina Websecurity Scanner	eEye Digital Security	www.eeye.com
WebApp360	ncircle	www.ncircle.com
WebKing	Parasoft	www.parasoft.com

Für diese Scanner lagen keine durch die Hersteller ausgefüllten Fragebögen vor.

Ein Audit gliedert sich in mehrere Teile, für die die Verantwortlichen Eingriffstiefe, Rahmenbedingungen sowie Werkzeuge festlegen (Abb. 2).



Ob bei Inbetriebnahme, Wartung oder Abschalten einer Anwendung – ein Webapplikations-Scanner kann in jeder Phase gute Dienste leisten und Schwachstellen aufdecken (Abb. 3).

Ein interessantes Feature bietet zum Beispiel Rational AppScan mit seinem „Glass Box" genannten Modul zur Integration in die zu testende Webapplikation. Scanner, die diese Funktion bieten, können über einen reinen Black-Box-Test hinaus durch Zugriff auf die getestete Software feststellen, ob alle Skripte erfasst und getestet wurden und an welcher Stelle im Code ein Problem auftritt.

Verflixte Fehlalarme

Auch in der allgemeinen Programmbedienung gibt es Unterschiede. Nicht alle Produkte ermöglichen es, Treffer als False-Positive zu markieren und von der weiteren Bearbeitung und dem Reporting auszunehmen. Ebenso wenig ermöglichen alle Lösungen das Pausieren oder Speichern und spätere Fortsetzen von Testläufen. Dies kann in einigen Situationen jedoch sehr nützlich sein. Etwa, wenn es zu einer temporären Unterbrechung der Internetverbindung kommt oder die Notwendigkeit besteht, einen mehrstündigen Test zu stoppen.

Vorgefertigte Testprofile oder die Option, eigene Profile anzulegen, ermöglichen es, schnell und effizient eine größere Menge gleichartiger Tests durchzuführen oder spezifische Tests zu späteren Zeitpunkten auf Knopfdruck zu wiederholen. Eingebaute Scheduler übernehmen die zeitliche Terminierung von Tests und auch die automatisierte Testwiederholung. Damit eng verbunden ist die Unterstützung von Delta-Tests, dem Vergleich zweier Testläufe. Dies hilft besonders dann, wenn nach einer Fehlerbehebung durch die Webentwickler ein erneuter Test ansteht.

Zur Steuerung der in Anspruch genommenen Bandbreite und der erzeugten Serverlast bieten viele Produkte eine Throttling-Konfiguration (s. Glossar), entweder vor einem Testlauf oder sogar dynamisch währenddessen. Dies ist insbesondere dann sinnvoll, wenn man Live-Systeme testen möchte und die Last niedrig bleiben soll. Schließlich hilft eine passende Visualisierung der getesteten Webapplikation dem schnellen Überblicken ihrer Bestandteile und der ausgeführten Tests. Und eine Mehrbenutzerfähigkeit kann in der Arbeit in Teams erforderlich sein, etwa wenn mehrere Tester koordiniert einen gemeinsamen Audit durchführen.

Besonders benutzerfreundlich ist fraglos die Variante Software as a Service. Einige Hersteller bündeln ihre Produkte optional mit der Expertise eigener Penetrationstester oder der von Partnerunternehmen und übernehmen damit den kompletten Audit von Webapplikationen.

Team-Player bevorzugt

Zuverlässig alle Fehler finden ist nur die eine Seite der Medaille, die Ergebnisse für weitere Zwecke zu dokumentieren ist ebenso unerlässlich. Lediglich Watobo bietet keine Möglichkeit, Reports mit den Funden zu erstellen. Die übrigen Lösungen enthalten Report-Funktionen, teilweise in verschiedenen Formaten. Da nicht alle Produkte eine Anpassung der Reports anbieten, ist zumindest der Export in ein leicht zu parsendes Format wie XML oder CSV wichtig.

Nicht zwingend, aber komfortabel sind vorgefertigte Compliance-Reports etwa nach PCI-DSS, OWASP Top 10 oder IT-Grundschutz. Sie ermöglichen es, schnell und ohne großen Aufwand aussagekräftige Dokumentationen für bestimmte Einsatzzwecke zu erstellen. Zur weiteren Vorgehensweise nach dem Finden von Schwachstellen verwenden Webentwickler oft ein Ticketing-System. Rational AppScan und NTOSpider haben gleich eines integriert, bieten aber wie einige andere Produkte ebenfalls die Möglichkeit, ein externes System mit Funden zu versorgen.

Gilt es, einen Scanner in eine Testumgebung einzubetten oder per Skript zu automatisieren, sollte er einen Headless-Betrieb per Konsole oder noch besser eine leistungsfähige API zur Verfügung stellen. Alle hier vorgestellten Produkte können einen Upstream-Proxy ansprechen, was etwa wichtig sein kann, wenn zwecks Dokumentation alle Übertragungen zentral protokolliert oder upstream ein zusätzlicher Scanner zum Einsatz kommen soll.

Fazit

Die hier vorgestellten Produkte bieten alle die wesentlichen Basisfunktionen, die für Audits von Webapplikationen notwendig sind. So wichtig wie die Kenntnis von Umfang, Qualität und Preis sind jedoch auch die Kenntnis der eigenen Anforderungen und der individuellen Vorlieben und Arbeitsweise eines Auditors oder Penetrationstesters. Erstkäufern und Erstnutzern kommt vielleicht die Möglichkeit gelegen, mit kostenfreien Scannern erste Erfahrungen zu machen und später zu entscheiden, ob und wenn ja, welchen individuellen Mehrwert die zum Teil umfangreicheren Features kommerzieller Lösungen bieten. Für fortgeschrittene Benutzer kann auch der Einsatz mehrerer Werkzeuge infrage kommen. (ur)

MARTIN WUNDRAM

ist Geschäftsführer der TronicGuard GmbH und beschäftigt sich dort mit der Sicherheit von Webapplikationen. Er ist von der IHK zu Köln öffentlich bestellter und vereidigter Sachverständiger für Systeme und Anwendungen der Informationsverarbeitung.

Literatur

[1] Martin Wundram; Schwachstellensuche; Die schwächste Stelle; Drei Webapplikations-Scanner im Vergleich; *iX* 9/2011, S. 72

Alle Links: www.ix.de/ix1205092