CYBER2 – Executive Summary

Webserver Enumeration

Kandidaten:	Martin Perotto	martin.perotto@stud.hslu.ch	20-298-592
	Fabian Stalder	fabian.stalder@stud.hslu.ch	20-296-729
	Maurice Suter	maurice.suter.01@stud.hslu.ch	20-296-752
Dozierende:	Sebastian Obermeier		
Eingereicht am:	03.10.2022		

Hochschule Luzern - Informatik
Studiengang Information & Cyber Security
Suurstoffi 1
6343 Rotkreuz

Inhaltsverzeichnis

1. Einleitung

Im ersten Teil dieses Executive Summary wird eine theoretische Einführung über die verschiedenen Arten des Vulnerability Scanning gegeben. Anschliessend werden diverse Tools vorgestellt, um Schwachstellen und Verzeichnisse auf Webservern zu finden.
Im letzten Teil gibt es ein Cheat-Sheet, welches als Referenz für die nächsten CTFs im Modul CYBER2 verwendet werden kann.

2. Webserver auf Schwachstellen scannen

Das Ziel eines Web Application Vulnerability Scanner ist es, Schwachstellen anhand von automatisierten Programmen an einem Webserver zu identifizieren. Die automatisierten Programme, welche zu einer Software zusammengefasst sind, werden auch als Scanner bezeichnet.

Die Scanner suchen beispielsweise aktiv nach folgenden Webserver Schwachstellen:

- SQL Injection
- Cross-site scripting (XSS)
- Unsichere Konfigurationen

(OWASP)

Schwachstellen-Scanner können in zwei grosse Bereiche unterteilt werden:

- Network-based Scanners werden unter anderem dazu eingesetzt, von extern nach Schwachstellen zu suchen. Wie der Name bereits erwähnt, werden Schwachstellen ohne weitere Software über das Netzwerk gesucht. Ein Network-based Scanner sucht Schwachstellen wie offene Ports, Softwareversionen welche durch einen Exploit missbraucht werden oder Buffer-Overflows. (EC-Council, 2009)

- Host-based Scanners funktionieren in den meisten Fällen nicht ohne eine zusätzliche Software, auch Agent genannt, welche auf dem zu scannenden Host installiert werden muss. Host-based Scanners suchen nach fehlerhaften Berechtigungen und zu schwachen Passwörter. (EC-Council, 2009)

Ein weiterer Teil sind Web Application Scanner, welche den Fokus auf Webapplikationen und deren Verwundbarkeiten spezialisiert sind. Web Application Scanner können unter anderem Schwachstellen im Bereich von Sessions, Access Control Mechanismen wie auch bei persistentem Cross-Site Scripting finden. (Andrew)

Je nach Software (beispielsweise Nessus) ist ein Scanner, welcher ein Network-based Scan durchführt, in der Lage aktive Hosts im Netzwerk zu finden (Discovery), auf diesen Hosts Schwachstellen mit Hilfe von Versionsabgleichen zu suchen und sogar den dazu passenden Exploit durchzuführen.

Bei Web Applikation Scanner wie BurpSuite gibt es unterschiedliche Möglichkeiten anhand von Modulen, Schwachstellen zu finden. Dazu gehören die folgenden Techniken:

- Das Modul «Scanner / Spider» sammelt den Inhalt der Website und crawlt diese zu einem Tree zusammen. Im nächsten Schritt werden die zusammengefassten Ergebnisse und der bis zu diesem Zeitpunk ausgetauschte Verkehr auf Schwachstellen untersucht.
- Mit dem Modul «Intruder» können automatisierte Attacken ausgeführt werden, indem gewisse Payloads in den bereits mitgeschnittenen Request ersetzt werden.
- Mit dem Modul «Repeater» können manuell HTTP und Websocket Request manipuliert und so die Response der Applikation analysiert werden.

(Hochschule Wismar, 2020)

Es gibt noch weitere Module in der Burp Suite. Dieser kurze Überblick soll aufzeigen wie vielfältig die Herangehensweisen beim Testen von Applikationen, in unserem Fall eine Webapplikation, sein können.

3. Tools

Es gibt viele verschiedene Tools, die helfen einen Webserver auf Schwachstellen zu scannen. Die Unterscheidung der verschiedenen Kategorien wurde bereits im vorherigen Kapitel vorgenommen.

Im Folgenden werden verschiedene Tools kurz vorgestellt. Es wird dabei vor allem auf die jeweiligen Einsatzorte, Stärken und Grenzen der jeweiligen Tools geschaut.

Nikto
Nikto ist ein Open Source Schwachstellen-Scanner, der umfassende Tests gegen Webserver durchführt. Dies passiert auf mehreren Ebenen. Es wird nach potenziell gefährlichen Programmen, veralteten Softwareversionen, Konfigurationsproblemen und vielem mehr gesucht. Nikto wurde nicht als unauffälliges Tool entworfen. Es testet einen Webserver schnellstmöglich und wird daher in Logfiles oder von einem IDS/IPS mit wenig Aufwand erkannt. Es wurden in der Zwischenzeit jedoch Methoden entwickelt, den Scan unauffälliger durchzuführen. Nikto ist auf Kali Linux bereits vorinstalliert.
(Sullo & Lodge, o. J.)

Nessus
Nessus ist ebenfalls ein Schwachstellen-Scanner. Unterschiedlich sind hier vor allem die Lizenzbedingungen und die Vorgehensweise. Nessus ist proprietär und damit nicht Open Source. Es gibt eine kostenlose, jedoch eingeschränkte Version.

Im Gegensatz zu Nikto wird Nessus nicht ausschliesslich für Webserver eingesetzt. Dieses Tool prüft jegliche Hosts in einem Netzwerk auf bekannte Schwachstellen. Nessus besitzt über 175'000 Plugins und unterscheidet zwischen über 75'000 CVEs. Plugins sind kleine Programme, die Schwachstellen aufsuchen. Nessus ist nur eingeschränkt via Terminal benutzbar.
(Tenable, o. J.)

Burp Suite
Dieses Tool wurde bereits im vergangenen Kapitel vorgestellt. Burp Suite ist vor allem auf Webapplikationen ausgelegt. Es wird versucht aus dem Netzwerkverkehr (Request, Response) möglichst viele Informationen zu gewinnen. Burp Suite ist auf Kali Linux bereits vorinstalliert. Jedoch ist der komplette Funktionsumfang nur mit kaufbaren Lizenzen zu nutzen.

Nmap
Auch Nmap kann dank der Nmap Scripting Engine (NSE) für die Schwachstellen-Suche genutzt werden. Dank der NSE können User Skripte erstellen und teilen. (Nmap.org, o. J.) Es gibt diverse Skripts, die das Finden von Webserver-Schwachstellen ermöglichen. Nmap ist auf Kali Linux bereits vorinstalliert und frei verfügbar.

4. Versteckte Verzeichnisse auf dem Webserver finden

Um Verzeichnisse auf einem Webserver zu finden, können Tools wie Dirbuster und Gobuster eingesetzt werden. Diese beiden Tools erlauben es mit wenig Aufwand, Verzeichnisse oder Dokumente, die auf einem Webserver liegen, zu finden. Die beiden Tools sollten mit Wörterlisten eingesetzt werden. Der Einsatz von Wörterlisten ist aus dem Grund zu empfehlen, da es sich bei den meisten Webserververzeichnissen um bekannte Namensgebungen handelt und so viel Zeit gespart werden kann bei der Suche. Falls ein Webserver nicht oft genutzte Ordner-Namen verwendet, kann mit dem Tool Dirbuster auch ein Brute Force Mode verwendet werden.

Das Tool Gobuster unterstützt keinen Brute Force Mode. Doch das Tool erlaubt es auch mit Hilfe von Wörterlisten Subdomänen zu finden, die existieren. Eine weitere Funktion von Gobuster, die einem das Testen von Schwachstellen erleichtern kann, ist der Fuzzing Mode. Dieser erlaubt es einem mit vordefinierten Daten eine Website auf mögliche Schwachstellen zu testen und zu sehen, wie diese mit den Verschiedenen Inhalten umgehen kann.

Grundsätzlich können sowohl Gobuster als auch Dirbuster verwendet werden, um versteckte Verzeichnisse zu finden. Dirbuster könnte auch mit einer grafischen Oberfläche bedient werden.
Geht es um die rekursive Suche bei versteckten Verzeichnissen auf dem Webserver, können mit Dirbuster tief verschachtelte Unterseiten entdeckt werden. Gobuster hat in dieser Kategorie das Nachsehen, da es pro Befehl jeweils nur eine Unterseite «tiefer» Verzeichnisse sammeln kann.
Ein weiterer Vorteil von Gobuster ist die Performance, die Suche läuft im Vergleich zu Dirbuster schneller ab. Aus diesem Grund sollte bei umfangreichen Websiten Gobuster verwendet werden. (Hörnemann, 2020)

5. Cheat-Sheet

5.1 Webserver Schwachstellen-Scan

Nikto		
Syntax	**Beispiel**	**Beschreibung**
`-h`	`nikto -h 192.168.156.57`	IP oder Hostname des Webservers
`-p`	`nikto -h 192.168.156.57 -p 80`	Port, der gescannt werden soll
`-o`	`nikto -h 192.168.156.57 -p 80 -o result.txt`	Die Rückgabe wird in die Datei «result.txt» gespeichert
`-f`	`nikto -h 192.168.156.57 -p 80 -o result -F json`	Format der zu erstellenden Datei

Tabelle 1: Übersicht Nikto

Nmap		
Syntax	**Beispiel**	**Beschreibung**
`div.`	`nmap --top-ports 20 192.168.156.0/24 --open`	Suche nach den 20 meistverbreiteten, offenen Ports in einem Subnetz
`-sV`	`nmap -sV --script=http-enum 192.168.156.57`	Versucht bei offenen Ports eine Service-/Versionsinformation zu erhalten
`--script`	`nmap -sV --script=http-enum 192.168.156.57`	Führt ein Skript aus, hier: http-enum

Tabelle 2: Übersicht Nmap

5.2 Webserver Verzeichnisse finden

Dirbuster		
Syntax	**Beispiel**	**Beschreibung**
`-H`	`dirbuster -H`	Ohne GUI (headless)
`-u`	`dirbuster -H -u https://192.168.156.57`	Ziel URL setzen
`-l`	`dirbuster -H https://192.168.156.57 -l /usr/share/dirbuster/wordlist/directory-list-2.3-medium.txt`	Pfad zur Wordlist
`-t`	`dirbuster -H https://192.168.156.57 -t 20`	Anzahl aktiver Session-Threads
`-s`	`dirbuster -H 192.168.156.57 -l /usr/share/dirbuster/wordlist/directory-list-2.3-medium.txt -s /`	Startpunkt, ab dem Verzeichnisse gesucht werden

Tabelle 3: Übersicht Dirbuster

Gobuster		
Syntax	**Beispiel**	**Beschreibung**
`--url --word-list`	`gobuster dir --url https://192.168.156.57 --wordlist /usr/share/gobuster/wordlist.txt`	Pfad zur Wordlist, um Directorys und Files zu finden
`-domain --word-list`	`gobuster dns –domain google.com --wordlist /usr/share/gobuster/wordlist.txt`	Suchen nach Subdomains
`--word-list`	`gobuster s3 –-wordlist /usr/share/gobuster/wordlist.txt`	Suchen nach Amazon S3 Buckets
`--url --word-list`	`gobuster vhost --url https://192.168.156.57 --wordlist /usr/share/gobuster/wordlist.txt`	Suchen nach Virtual Hosts auf dem Webserver

Tabelle 4: Übersicht Gobuster | (Linux Command Library)

6. Verweise

6.1 Tabellenverzeichnis

7. Literaturverzeichnis

Andrew, D. *The Ultimate Guide to Vulnerability Scanning*. https://www.intruder.io/guides/the-ultimate-guide-to-vulnerability-scanning

EC-Council. (2009). *Ethical Hacking and Countermeasures: Attack Phases* (1. Aufl.). https://books.google.ch/books?id=DeAEAAAAQBAJ&pg=SA6-PA3&dq=vulnerability+scanners+host+based+network-based&hl=de&sa=X&ved=2ahUKEwix8j_tLL6AhUB_rsIHZc7DGUQ6AF6BAgDEAI#v=onepage&q=vulnerability%20scanners%20host%20based%20network-based&f=false

Hochschule Wismar. (2020, 23. Februar). *BurpSuite – IT-Forensik Wiki*. https://it-forensik.fiw.hs-wismar.de/index.php/BurpSuite

Hörnemann, J. (2020). *Pentest-Tools #2 – gobuster*. https://aware7.com/de/blog/pentest-tools-2-gobuster/

Linux Command Library (Hrsg.). *gobuster: Brute-forces hidden paths on web servers and more.* https://linuxcommandlibrary.com/man/gobuster

Nmap.org. (o. J.). *Nmap Scripting Engine (NSE).* Nmap.org. https://nmap.org/book/man-nse.html

OWASP (Hrsg.). *Vulnerability Scanning Tools | OWASP Foundation*. https://owasp.org/www-community/Vulnerability_Scanning_Tools

Sullo, C. & Lodge, D. (o. J.). *Nikto2.* CIRT.net. https://cirt.net/Nikto2

Tenable. (o. J.). *Nessus Vulnerability Assessment*. https://www.tenable.com/products/nessus