

TH Brandenburg
Online Studiengang IT Sicherheit
Fachbereich Informatik und Medien
Netzwerksicherheit
Prof. Dr. Michael Pilgermann

Einsendeaufgabe 1
Wintersemester 2023
Abgabetermin 27. Oktober 2024

Gruppe 11
Mathias Baumbach (Matr-Nr. 20213703)
Mara Schulke (Matr-Nr. 20215853)

Zusammenfassung

Innerhalb dieser Einsendaufgabe werden verschiedene Aspekte der Netzwerksicherheit – vor allem aus der Angreiferperspektive – betrachtet. Im Rahmen der Bearbeitung konnten wir wertvolle Erfahrungen sammeln – gerade die Aufgabe 2.5 (Google-Hacking) hat uns erneut vor Augen geführt wie wichtig eine entsprechende Absicherung von IT-Systemen ist, da wir innerhalb von wenigen Minuten vollen Zugriff auf die Datenbank eines PHP Web-Servers erlangen konnten. Wir haben den Betreiber informiert und anonymisiert. So, dass diese Dokumentation nicht zu einer weiteren Ausnutzung verwendet werden kann.

Inhaltsverzeichnis

1 Vorbereitungen	2
2 Durchführung	3
2.1 Telnet und Wireshark	3
2.2 Schwachstellenscan I	5
2.3 MAC Spoofing	8
2.4 Schwachstellenscan II	11
2.5 Google-Hacking	14

Abbildungsverzeichnis

1 Telnet Datei: Start des Logins	3
2 Telnet Datei: Login Sequenz Dauer	4
3 nmap: Scan-Techniken	5
4 nmap: Port Scan	6
5 nmap: TCP Scan	6
6 nmap: OS Scan	7
7 nmap: UDP Scan	7
8 macspoofing: Ursprünglicher Netzwerk-Verkehr	8
9 macspoofing: Ursprüngliche Source und Destination	9
10 macspoofing: Änderung der MAC-Adresse von eth0	9
11 macspoofing: Veränderter Netzwerk-Verkehr	10
12 wpscan: Fehler	11
13 wpscan: Ergebnis 1/2	12
14 wpscan: Ergebnis 2/2	13
15 Google Hacking: Backup: Schwachstelle	14
16 Google Hacking: Backup: Google Suche	15
17 Google Hacking: Backup: Ergebnis	15
18 Google Hacking: Tokens: Suche	16
19 Google Hacking: Tokens: Ergebnis	17
20 Google Hacking: SQL Injektion: Schwachstelle	17
21 Google Hacking: SQL Injektion: Schwachstelle	18
22 Google Hacking: SQL Injektion: Suche	18
23 Google Hacking: SQL Injektion: Ergebnis	19

1 Vorbereitungen

Die bevorzugte Methode zur Beantwortung der Einsendeaufgabe besteht darin, dass Sie entweder ParrotOS oder Kali Linux als virtuelle Maschine auf Ihrem Rechner installieren. Diese beiden Linux-Distributionen sind speziell auf die IT-Sicherheit ausgelegt und haben daher viele interessante Tools schon vorinstalliert. Tipp zur Durchführung: Laden Sie von ParrotOS eine .ova für virtuelle Maschinen herunter (genauer die .ova der Security Edition; Tipp: das Superuser-Password lautet dann "parrot" und nicht "toor" wie bei anderen Versionen). Falls noch nicht vorhanden, installieren Sie Virtualbox und führen die virtuelle Maschine darin aus. Leider gibt es in der letzten Zeit zunehmend Prozessorarchitekturen (gerade für Mac), bei denen keine Möglichkeit zum Betrieb von virtuellen Maschinen besteht. Sollte das der Fall sein, dann bearbeiten Sie die Aufgaben auf Ihrem eigenen Betriebssystem. Sie müssen dafür die Tools einzeln herunterladen.

2 Durchführung

2.1 Telnet und Wireshark

Aufgabenstellung

Wireshark ist bei ParrotOS und Kali Linux vorinstalliert. Es kann ansonsten von der Wireshark-Seite heruntergeladen werden. In der Wireshark-Datei im Kurs (telnet.pcapng) ist eine Aufzeichnung eines Einlog-Vorgangs mit Telnet zu finden. Suchen Sie darin das Passwort.

Antwort

Nachdem wir die Datei analysiert und den Telnet-Login-Vorgang bei dem Paket 22 lokalisiert haben, konnten wir den Start des Passwords in Text-Form bei dem Paket 23 entdeckt. (Siehe Abb. 1).

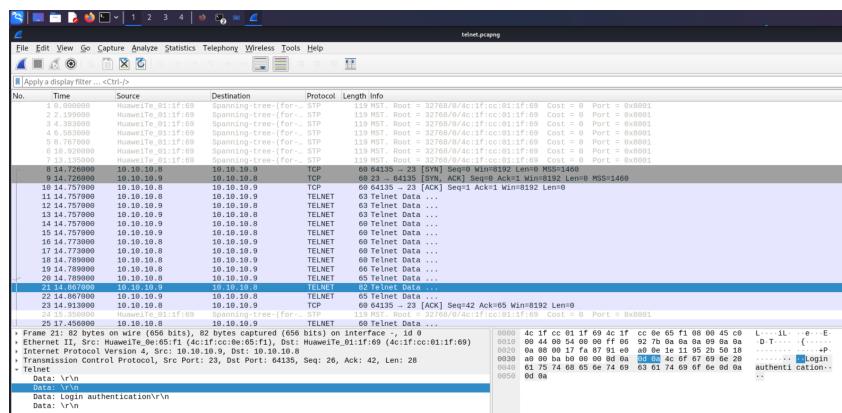


Abbildung 1: Telnet Datei: Start des Logins

Nun enthielt jedes weitere Paket ein Zeichen des Passworts, bis die End-Sequenz in dem Paket 46 gesendet wurde. Zusammengesetzt ergibt dies das Passwort **sshnutzen**. (Siehe Abb. 2).

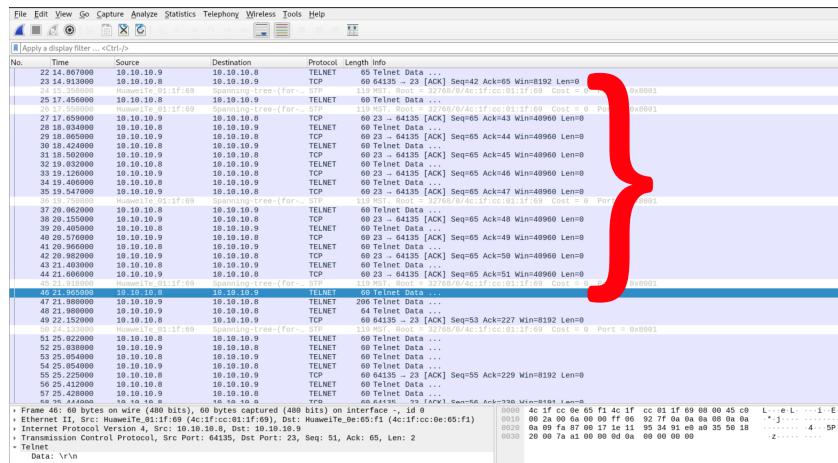


Abbildung 2: Telnet Datei: Login Sequenz Dauer

2.2 Schwachstellenscan I

Aufgabenstellung

Verwenden Sie **nmap** (bei ParrotOS und Kali Linux vorinstalliert), um verschiedene Scans des Testservers nwsmooc.mooin.org durchzuführen. Erklären Sie die Ergebnisse, wobei mindestens drei Tests mit jeweils unterschiedlichen Parametern durchgeführt werden müssen. Versuchen Sie dabei u.a. herauszufinden, welche Dienste auf dem Zielserver installiert sind und welches Betriebssystem verwendet wird.

Antwort

Der Aufgabe entsprechend haben wir drei sinnvolle **nmap** scans durchgeführt. Auf den im Internet verfügbaren ubuntu manpage Seiten sind verschiedene Scan-Techniken mit **nmap** aufgelistet. (Siehe Abb. 3).

Scan-Techniken	
Nmap kennt verschiedene Scan-Techniken, die wie folgt aufgerufen werden können:	
nmap - Scan-Techniken	
Scan Technik	Beschreibung
-sT	Einfacher Connect Scan. Hierbei wird pro zu scannenden Port eine volle TCP-Verbindung auf- und wieder abgebaut. Dieser Scan steht auch zur Verfügung, wenn nmap ohne Root-Recht aufgerufen wird.
-sS	"SYN-Stealth-Scan": Ähnlich -sT, allerdings wird keine komplette TCP-Verbindung aufgebaut, daher unauffälliger. (Standard bei Root-Rechten)
-sU	Scans UDP-Ports statt TCP.
-sN	Ping-Scan: Prüft nur auf Erreichbarkeit über ICMP-Echo-Request, TCP-SYN-Paket auf Port 443, TCP-ACK-Paket auf Port 80 und ICMP-Timestamp-Request. Sinnvoll, um ganze Netzbereiche auf aktive Hosts zu testen.
-sO	Internet-Protokoll (IP) Scan
--scanflags <flags>	Flags für TCP-Scan festlegen
-sI <zombie Host[:Port]>	Idle-Scan (TCP)
-6	Scans auch IPv6-Adressen
Experten-Info:	
Folgende etwas exotischere Techniken (und noch weitere) können in Einzelfällen sinnvoll sein. Es empfiehlt sich aber das vorherige Studium der Manpage oder der Homepage » von nmap, um die Besonderheiten und möglichen Erkenntnisse zu verstehen.	
Besondere Scantechniken	
Scan Technik	Beschreibung
-SF, -SN, -SX	nmap sendet an die zu scannenden Ports bewusst manipulierte bzw. falsche TCP-Pakete. Anhand der Reaktion des Ports (bzw. des Servers) lassen sich ggf. Rückschlüsse ziehen, ob der Port offen oder von einer Firewall geschützt ist. Im Vergleich zu -sT oder -sR ebenfalls unauffälliger.
-SA, -SW	ACK und Window Scan. Für die Einrichtung einer Verbindung zu einem Port. Besonders gut zum Erkennen von Firewalls.

Abbildung 3: nmap: Scan-Techniken

Nach kurzer Recherche haben wir uns für die folgenden vier Scan-Techniken entschieden, da sie in Kombinationen eine gute Aussage-Stärke über das Host-System des Servers und die darauf ausgeführten Programme treffen.

Port Scan

Eine der bekanntesten Techniken mit **nmap** ist der Port-Scan – dieser kann entweder auf die "well-known" Ports beschränkt werden, oder über alle ports von 0-65536 ausgeführt werden. Da wir im Rahmen dieser Aufgabe eine Aussage über die Dienste des Servers treffen sollen, haben wir uns dafür entschieden einen kompletten Port-Scan auszuführen. (Siehe Abb. 4)

```
(agathon@mpc) ~]$ nmap -p- nwsmooc.mooin.org
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-03 15:54 CEST
Nmap scan report for nwsmooc.mooin.org (81.169.230.170)
Host is up (0.46s latency).
Other addresses for nwsmooc.mooin.org (not scanned): 2a01:238:4344:5000:e79a:888:dfb5:6b52
rDNS record for 81.169.230.170: h2524950.stratoserver.net
Not shown: 63940 filtered tcp ports (no-response), 1591 filtered tcp ports (host-unreach)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
443/tcp   open  https
3544/tcp  closed teredo

Nmap done: 1 IP address (1 host up) scanned in 1552.21 seconds
```

Abbildung 4: nmap: Port Scan

Hier lässt sich erkennen, dass der Server einen SSH-Server, einen HTTP- bzw. HTTPS-Server betreibt. Diese Server sind öffentlich und für das Internet zugänglich. Auf dem Port 3544 wird vermutlich eine Torredo-Instanz ausgeführt, nmap kann aber keine genaue Aussage darüber treffen, ob dieser Dienst wirklich aktiv ist (daher wurde dieser Port als CLOSED markiert).

TCP Scan

Als nächstes haben wir einen tiefgreifenden TCP-Scan ausgeführt, um detailliertere Informationen über die drei (im vorherigen Schritt entdeckte) Dienste herauszufinden. (Siehe Abb. 5)

```
(agathon@mpc) ~]$ nmap -T4 -A nwsmooc.mooin.org
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-03 15:09 CEST
Stats: 0:00:02 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 2.95% done; ETC: 15:10 (0:01:06 remaining)
Nmap scan report for nwsmooc.mooin.org (81.169.230.170)
Host is up (0.87s latency).
Other addresses for nwsmooc.mooin.org (not scanned): 2a01:238:4344:5000:e79a:888:dfb5:6b52
rDNS record for 81.169.230.170: h2524950.stratoserver.net
Not shown: 952 filtered tcp ports (no-response), 45 filtered tcp ports (host-unreach)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.4 (protocol 2.0)
| ssh-hostkey:
|   2048 af:ad:c0:47:7f:31:c5:b8:bd:bb:9d:ad:e2:a3:61:1f (RSA)
|   256 06:53:00:e8:8d:56:cf:de:32:cb:53:d3:33:29:05:cc (ECDSA)
|_  256 2b:03:b2:ef:af:e9:f4:72:de:3b:80:9a:73:af:8d:a2 (ED25519)
80/tcp    open  http     Apache httpd
|_http-title: 500 Internal Server Error
|_http-server-header: Apache
443/tcp   open  ssl/http Apache httpd (PHP 8.1.24)
| ssl-cert: Subject: commonName=nwsmooc.mooin.org
| Subject Alternative Name: DNS:nwsmooc.mooin.org
| Not valid before: 2023-09-25T22:58:29
|_Not valid after: 2023-12-24T22:58:28
|_http-server-header: Apache
| http-robots.txt: 1 disallowed entry
|_/wp-admin/
|_http-title: Netzwerksicherheit MOOC
|_http-generator: WordPress 6.3.1

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 51.69 seconds
```

Abbildung 5: nmap: TCP Scan

In diesem Scan konnten wir weitere Informationen herausfinden, wie z.B. die Versionen / Distributionen des SSH- und HTTP-Servers, die öffentlichen SSH-Schlüssel des Servers und die Konfiguration des Apache- Servers.

Betriebssystem Scan

Um eine Aussage über das ausgeführte Betriebssystem treffen zu können, bietet nmap einen dedizierten Betriebssystem-Scan an – dieser kann durch -O gestartet werden. (Siehe Abb. 6)

```
(agathon@mpc:~]
$ nmap -O nwsmooc.mooin.org
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-03 15:31 CEST
TCP/IP Fingerprinting (for OS scan) requires root privileges.
QUITTING!
[+] (agathon@mpc:~]
[-] $ sudo nmap -O nwsmooc.mooin.org
[sudo] password for agathon: 
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-03 15:31 CEST
Nmap scan report for nwsmooc.mooin.org (81.169.230.178)
Host is up (0.0076s latency).
Other addresses for nwsmooc.mooin.org (not scanned): 2a01:238:4344:5000:e79a:888:dfb5:6b52
rDNS record for 81.169.230.178: h2524950.stratoserver.net
Not shown: 985 filtered tcp ports (no-response), 12 filtered tcp ports (host-prohibited)
PORT      STATE     SERVICE
22/tcp    open      ssh
80/tcp    open      http
443/tcp   open      https
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
OS detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 49.83 seconds
```

Abbildung 6: nmap: OS Scan

Das Ergebnis dieses Scans ist zwar nicht eindeutig, lässt aber darauf deuten, dass der Server eine Linux- Distribution ausführt.

UDP Scan

Zuletzt haben wir noch die bisher wenig betrachteten UDP-Dienste des Servers gescannt. (Siehe Abb. 7)

```
(agathon@mpc:~]
$ nmap -sU nwsmooc.mooin.org
You requested a scan type which requires root privileges.
QUITTING!
[+] (agathon@mpc:~]
[-] $ sudo nmap -sU nwsmooc.mooin.org
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-03 15:33 CEST
Nmap scan report for nwsmooc.mooin.org (81.169.230.170)
Host is up (0.0076s latency).
Other addresses for nwsmooc.mooin.org (not scanned): 2a01:238:4344:5000:e79a:888:dfb5:6b52
rDNS record for 81.169.230.170: h2524950.stratoserver.net
Not shown: 994 filtered udp ports (host-prohibited)
PORT      STATE     SERVICE
135/udp  open|filtered msrpc
137/udp  open|filtered netbios-ns
138/udp  open|filtered netbios-dgm
139/udp  open|filtered netbios-ssn
389/udp  open|filtered ldap
445/udp  open|filtered microsoft-ds
Nmap done: 1 IP address (1 host up) scanned in 1060.16 seconds
```

Abbildung 7: nmap: UDP Scan

Hierbei wurden mehrere Dienste gefunden. Zwei der 4 UDP-Dienste stehen in Verbindung mit Microsoft, was entweder auf eine Windows-basierte Firewall oder einen Irrtum bei dem Betriebssystems-Scan hindeutet.

Jeder Dienst hat den Status `open|filtered`, was auf eine aktive Firewall-Konfiguration hin-deutet.

2.3 MAC Spoofing

Aufgabenstellung

Ebenfalls bereits bei ParrotOS und Kali schon vorinstalliert ist das Tool `macchanger`. Machen Sie sich mit dessen Möglichkeiten vertraut, wobei z.B. ein Video von HackerSploit nützlich sein kann: <https://www.youtube.com/watch?v=bshXz5r-CQA>. Senden Sie anschließend Datenverkehr mit gefälschter MAC-Adresse und zeichnen diesen mit Wireshark auf. Fertigen Sie geeignete Screenshots mit Erklärungen (wie müsste es richtig sein? wo ist die gefälschte MAC-Adresse in Wireshark zu sehen?) dazu an. Mac- und Linux-Nutzer verwenden für die Aufgabe ebenfalls `macchanger` oder `ifconfig`. Windows-Nutzer schauen sich bitte das Youtube-Tutorial an: https://www.youtube.com/watch?v=V3Pcc8b_mOU. Hinweis: Bei der ersten Methode heißt der Eintrag in den erweiterten Einstellungen nicht "Network Address", sondern "Locally Administered Address".

Antwort

Um die MAC-Adresse unter Kali-Linux zu ändern genügte eine einfache Kombination aus dem Programm `macchanger` und `ifconfig`. Zu erst, um eine erfolgreiche Änderung bzw. ein erfolgreiches Spoofing feststellen zu können, haben wir den Netzwerk-Verkehr zum Web-Server der vorherigen Aufgaben aufgezeichnet. (Siehe Abb. 8)

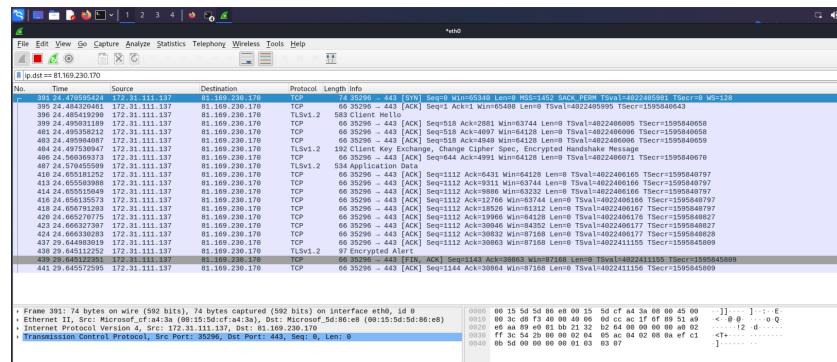


Abbildung 8: macspoofing; Ursprünglicher Netzwerk-Verkehr

Anschließend haben wir mittels `ifconfig` auf dem Host und in der virtuellen Maschine die MAC-Adressen ausgelesen und mit dem Ethernet-Adressen aus dem aufgezeichneten Netzwerk-Verkehr in Verbindung gebracht. (Siehe Abb. 9)

Die Source-Adresse (`eth0` in der VM) lautet: `00:15:5d:cf:a4:3a`

Die Source-Adresse (WSL auf dem Host) lautet: `00:15:5d:5d:86:e8`

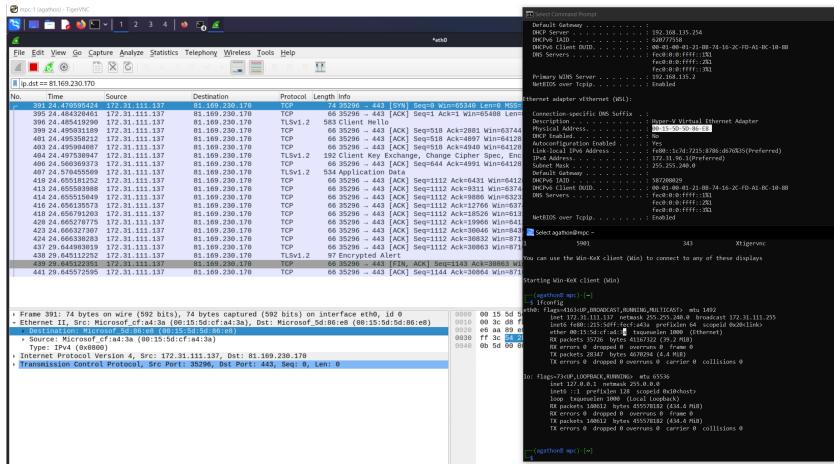


Abbildung 9: macspoofing; Ursprüngliche Source und Destination

In den beiden Screenshots ist zu erkennen, dass die momentane MAC-Adresse der virtuellen Maschine 00:15:5d:cf:a4:3a entspricht.

Nun muss zu erst das `eth0` Interface mittels `ifconfig` deaktiviert werden. Anschließend kann die MAC-Adresse mittels `macchanger` geändert werden. In diesem Beispiel haben wir lediglich die Hersteller-Nummer von `cf:a4:3a` zu `cf:a4:01` geändert und die Vendor-ID belassen, da eine Änderung dieser zu Problemen mit den Treibern führen kann. Anschließend muss das Netzwerk-Interface erneut durch `ifconfig` gestartet werden. (Siehe Abb. 10)

```
(agathon@mpc) -[~]
$ sudo macchanger -m 00:15:5d:cf:a4:01 eth0
Current MAC: 00:15:5d:cf:a4:3a (Microsoft Corporation)
Permanent MAC: 00:15:5d:cf:a4:3a (Microsoft Corporation)
New MAC: 00:15:5d:cf:a4:01 (Microsoft Corporation)

(agathon@mpc) -[~]
$ ifconfig eth0
eth0: flags=4098<POINTOPOINT,BROADCAST,MULTICAST> mtu 1492
        inet 172.31.111.137 netmask 255.255.240.0 broadcast 172.31.111.255
          ether 00:15:5d:cf:a4:01 txqueuelen 1000 (Ethernet)
            RX packets 725 bytes 441475 (431.1 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 749 bytes 95995 (93.7 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(agathon@mpc) -[~]
```

Abbildung 10: macspoofing: Änderung der MAC-Adresse von `eth0`

Bei erneuter Aufzeichnung des Netzwerk-Verkehrs zum Zielsystem lässt sich nun die gerade geänderte MAC-Adresse (`00:15:5d:cf:a4:01`) in dem Ethernet-Header jedes Pakets wiederfinden. (Siehe Abb. 11)

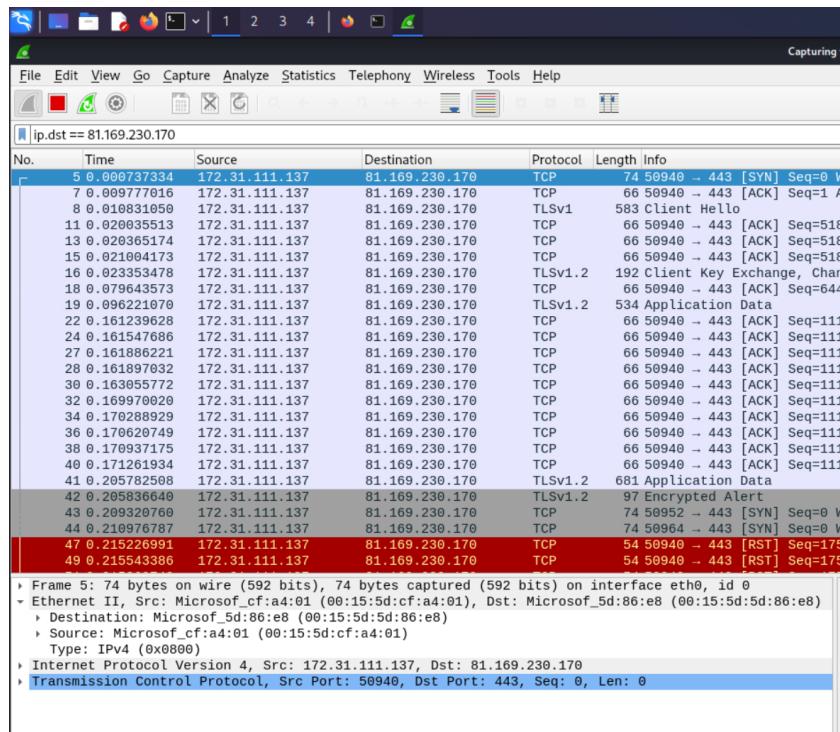


Abbildung 11: macspoofing: Veränderter Netzwerk-Verkehr

2.4 Schwachstellenscan II

Aufgabenstellung

Verwenden Sie das bei ParrotOS und Kali Linux vorinstallierte WPScan (WordPress Vulnerability Scanner) und führen Sie einen Scan von <https://nwsmooc.mooin.org> durch. Erklären Sie die Ergebnisse.

Hinweise:

- Es ist keine Registrierung beim Anbieter erforderlich. Durch eine Registrierung würde man ein Token erhalten, um Schwachstellentests durchführen zu können. So ist die Aufgabe auf eine Informationssammlung beschränkt.
 - Bei einem Test mit ParrotOS gab es zunächst eine Fehlermeldung, dass ein Update der Datenbank nicht möglich sei. Ein allgemeines Update (`sudo apt-get update && apt-get upgrade`) konnte dieses Problem beseitigen.

Sollten Sie Mac oder Linux verwenden, dann installieren Sie WPScan direkt von Github. Sollten Sie keine Möglichkeit zur Durchführung dieses Aufgabenteils finden, sprechen Sie die Betreuenden auf eine Ersatzaufgabe an.

Antwort

wpscan ist ein Open-Source-Tool, das für die Sicherheitsüberprüfung von WordPress-Websites entwickelt wurde, um potenzielle Schwachstellen und Sicherheitsprobleme in diesen zu erkennen. Es ermöglicht die Analyse von WordPress Webseiten nach eigenen Angaben seit über 10 Jahren und hat in dieser Zeit einen Katalog von mehr als 43000 Core, Plugin oder Theme Schwachstellen aufgebaut (Quelle <https://wpscan.com>). Ein Scan einer Webseite ist direkt von der wpscan.com Webseite möglich, oder vom eigenen Client aus mit dem CLI Programm **wpscan**.

Man bietet neben der kostenfreien Version für "Researcher" mit maximalen 25 API Calls pro Tag, auch eine nicht kostenlose Enterprise Version an.

Die Nutzung ist dabei gänzlich einfach. Der einfache Quick Scan wird durch den Aufruf `wpscan -url <webseite>` aufgerufen. Dabei erhielten wir jedoch statt eines Ergebnisses eine Fehler-Meldung über einen 403-Status-Code. (Siehe Abb. 12)

Abbildung 12: wpscan: Fehler

Der hier referenzierte HTTP-Statuscode bedeutet, dass der Server den Zugriff auf die angefragte Seite verweigert. Dies kann auf fehlende Zugriffsberechtigungen oder eine Web Application Firewall (WAF) hinweisen. Eine WAF soll Websites und Webanwendungen vor böswilligen Anfragen und Zugriffen schützen, indem sie den Datenverkehr zwischen dem Client und dem

Webserver analysiert, um Angriffe wie SQL-Injection, Cross-Site Scripting (XSS) und Distributed Denial of Service (DDoS) zu erkennen und zu blockieren. Sie verwendet dazu Regeln und Signaturen und ist ein wichtiger Bestandteil um die Sicherheit einer Webseite zu gewährleisten und Angriffe auf Anwendungsebene abzuwehren.

Der Parameter `-random-user-agent` aus dem gegebenen Hinweis versucht diesen Sicherheitsmechanismus zu umgehen, indem automatisch zufällige Benutzer-Agenten für jede HTTP-Anfrage, die es während des Scans an die WordPress-Website sendet, generiert um diese Anfragen weniger auffällig und weniger vorhersehbar zu gestalten um nicht automatisiert zu wirken. Da wir kein Bot-Netz oder ähnliches verwenden, ist die IP-Adresse der Anfragen jedoch die gleiche. Dies schien jedoch nicht von der Firewall verhindert zu werden, denn mit jenem Parameter lässt sich ein Scan durchführen. (Siehe Abb. 13)

```
(agathon@mpc)-[~]
$ wpScan --random-user-agent --url https://nwsmooc.mooin.org
[+] URL: https://nwsmooc.mooin.org/ [81.169.230.170]
[+] Started: Tue Oct 3 17:02:11 2023

Interesting Finding(s):
[+] Headers
| Interesting Entries:
| - Server: Apache
| - X-Powered-By: PHP/8.1.24
| Found By: Headers (Passive Detection)
| Confidence: 100%
[+] robots.txt found: https://nwsmooc.mooin.org/robots.txt
| Interesting Entries:
| - /wp-admin/
| - /wp-admin/admin-ajax.php
| Found By: Robots Txt (Aggressive Detection)
| Confidence: 100%
[+] WordPress readme found: https://nwsmooc.mooin.org/readme.html
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
[+] The external WP-Cron seems to be enabled: https://nwsmooc.mooin.org/wp-cron.php
| Found By: Direct Access (Aggressive Detection)
| Confidence: 60%
| References:
| - https://www.iplocation.net/defend-wordpress-from-ddos
| - https://github.com/wpscanteam/wpScan/issues/1299
[+] WordPress version 6.3.1 identified (Latest, released on 2023-08-29).
| Found By: Rss Generator (Passive Detection)
| - https://nwsmooc.mooin.org/feed/, <generator>https://wordpress.org/?v=6.3.1</generator>
| - https://nwsmooc.mooin.org/comments/feed/, <generator>https://wordpress.org/?v=6.3.1</generator>
```

Abbildung 13: wpScan: Ergebnis 1/2

In einer detaillierten Ansicht erhalten wir Informationen ausgegeben, die durch den Scan gewonnen wurden konnte. (Siehe Abb. 14)

- Ein Apache Webserver mit der PHP Version 8.1.24 liefert die Webseite aus (diese ist aktuell – Release Datum 28.09.2023)
- WordPress ist installiert in der Version 6.3.1 (ebenfalls zum Zeitpunkt des Scans aktuell mit Release Datum 29.08.2023). Ein RSS Feed ist vorhanden unter angegebener URL.

- `robots.txt` (Anweisungen/Berechtigungen für Web-Crawler von Suchmaschinen) und Wordpress-Readme sind vorhanden – Ausgabe mit Pfad
- WP-Cron ist aktiviert – dies lässt eine automatisierte Aktualisierung der Software-Komponenten der Wordpress-Installation annehmen
- twentysixteen ist das installierte Wordpress-Theme (ebenfalls zum Zeitpunkt des Scans aktuell mit Release Datum 29.03.2023). Eine eigene Readme ist unter der angegebenen URL zu erreichen. Eine kurze Description und eine Style URI sind auslesbar.
- Ein Plugin-Scan mit passiver Scan-Methode findet keine installierten Plugins.
- Es werden keine scanbaren Config-Backups gefunden

```

[...]
| WordPress theme in use: twentysixteen
| Author URI: https://wwwoc.kali.org/wp-content/themes/twentysixteen/
| Latest Version: 2.9 (up to date)
| Author: Automattic, Inc.
| Readme: https://wwwoc.kali.org/wp-content/themes/twentysixteen/readme.txt
| Style: https://wwwoc.kali.org/wp-content/themes/twentysixteen/style.css?ver=20230328
| Style Name: Twenty Sixteen
| Style URI: https://wwwoc.kali.org/themes/twentysixteen/
| Description: Twenty Sixteen is a modernized take on an ever-popular WordPress layout – the horizontal masthead ...
| Author: Automattic, Inc.
| Author URI: https://wordpress.org/
| Found By: Cms Style In Homepage (Passive Detection)
| Configuration File: https://wwwoc.kali.org/wp-content/themes/twentysixteen/style.css?ver=20230328
| Version: 2.9 (Match)
| Found By: Style (Passive Detection)
| - https://wwwoc.kali.org/wp-content/themes/twentysixteen/style.css?ver=20230328, Match: "Version: 2.9"
| Generating All Plugins (via Passive Methods)
| No Plugins Found.
| Generating Config Backups (via Passive and Aggressive Methods)
| Checking Config Backups - Time: 00:00:04
| (137 / 137) 100.00% Time: 00:00:04
[...]
||| No WPScan API Token given, as a result vulnerability data has not been output.
||| You can get a free API token with 25 daily requests by registering at https://wpscan.com/register
| Finished: Tue Oct 24 17:00:20 2023
| Requests Done: 178
| Cache Hits: 0
| Data Sent: 56.103 kB
| Total Headers: 1000
| Memory used: 277.256 MB
| Elapsed Time: 00:00:04
[...]

```

Abbildung 14: wpscan: Ergebnis 2/2

Da wir, wie in der Aufgabenstellung beschrieben, keine Registrierung beim `wpscan` Anbieter durchführen und daher keinen API-Key besitzen, endet unser Scan an dieser Stelle.

2.5 Google-Hacking

Aufgabenstellung

Mit "Google Hacking" ist gemeint, dass man die Google Suche zum Auffinden von Softwareinstalltionen mit Schwachstellen nutzen kann. Eine Sammlung von Beispielen ist bei <https://www.exploit-db.com/google-hacking-database/> zu finden. Erklären Sie anhand von drei selbstgewählten Beispielen, was man damit herausfinden kann. Achtung: Firefox und Google Chrome warnen teilweise beim Aufruf der Seite und bezeichnen diese als riskant. Man kann die Seite aber aus einem Browser innerhalb von Kali Linux oder ParrotOS aufrufen, dann kommt keine Warnung.

Antwort

Innerhalb der Exploit-Datenbank haben wir uns drei, unserer Meinung nach, besonders interessante Beispiele ausgesucht.

Das erste unserer Beispiele beruht auf dem Prinzip Google-Filter zu verwenden um von Google indexierte Dateien mit bestimmten Namen oder Dateiendungen zu finden. So können zum Beispiel falsch konfigurierte Web-Server aus Versehen vertrauliche Daten öffentlich zugänglich machen. Ein interessantes Beispiel ist hierbei nach rohen SQL-Backups zu suchen. Diese könnten, bei schlechten Sicherheitsmaßnahmen des Servers, rohe Nutzerdaten enthalten und somit entweder einen Angriff auf die betroffenen Accounts ermöglichen oder weiter verwendet werden um Zugriff auf andere, möglicherweise interessantere, Systeme zu erlangen. (Siehe Abb. 15)

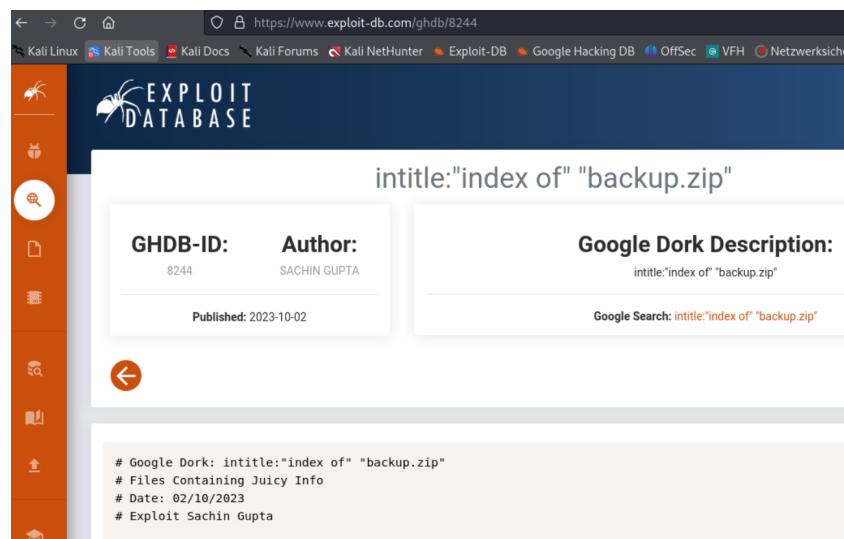


Abbildung 15: Google Hacking: Backup: Schwachstelle

Bei einer Google-Suche nach Dateiverzeichnissen, die eine Datei namens backup.zip enthalten sind wir sofort auf einen Web-Server gestoßen, bei dem sowohl eine backup.zip und eine backup.sql Datei versehentlich veröffentlicht wurden. (Siehe Abb. 16 & 17)

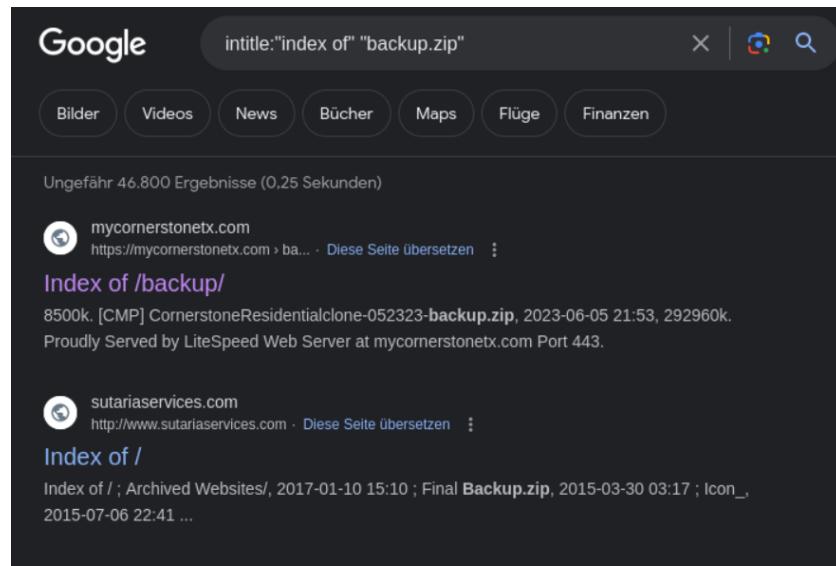


Abbildung 16: Google Hacking: Backup: Google Suche

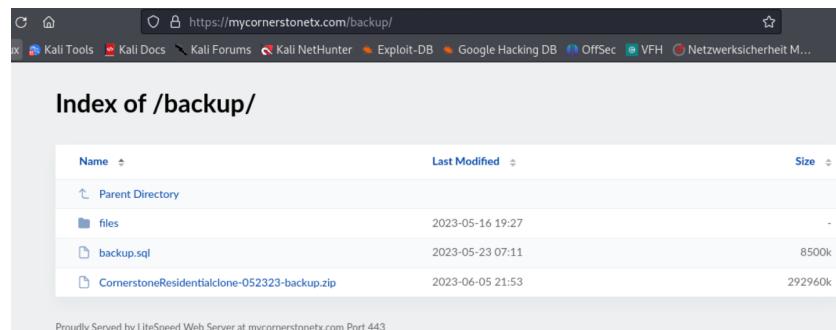


Abbildung 17: Google Hacking: Backup: Ergebnis

Das zweite unserer Beispiele ist die Suche nach Webseiten die Tokens oder ähnlich vertrauliche Informationen im Klartext enthalten. Dies ist ein besonders hohes Risiko bei Seiten, die Quellcode enthalten (Wie z.B. Pastebin oder GitHub). Das Beispiel in unserer Abbildung sucht nach Tokens auf öffentlich-zugänglichen Pastebin-Seiten.

Bei einer Suche nach pastebin.com Seiten, die entweder einen Benutzernamen, ein Passwort oder einen Schlüssel enthalten haben wir ebenfalls sofort Ergebnisse erzielen können. (Siehe Abb. 18 & 19)

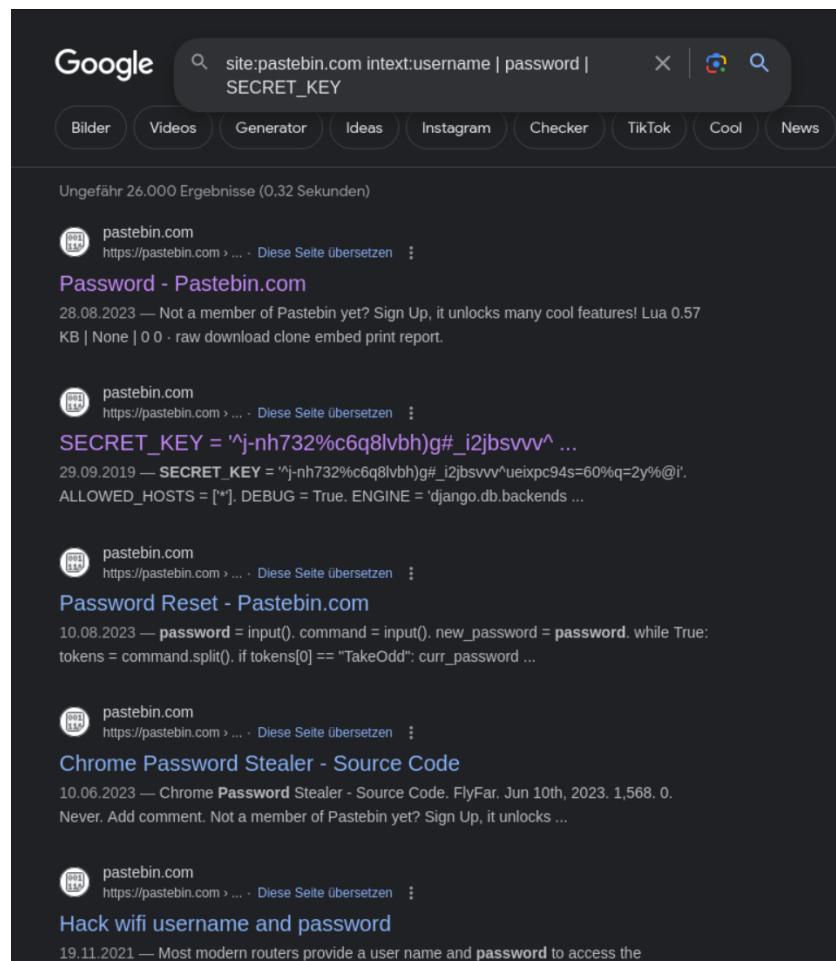


Abbildung 18: Google Hacking: Tokens: Suche

```

text 2.15 KB | None | 🔍 0
1. SECRET_KEY = '^Aj-rh732kc6q8lvbh)g#_12}bsvvv^ueixpc94s=60%q=2y%0'
2. ALLOWED_HOSTS = ['*']
3. DEBUG = True
4.
5. ENGINE = 'django.db.backends.postgresql_psycopg2'
6. NAME = 'pekhom'
7. USER = 'circle'
8. PASSWORD = 'circle123456789'
9. HOST = 'localhost'
10. PORT = ''
11.
12. # backend credentials
13. BACKEND_AUTH_URL = 'http://192.168.1.162:5000/api-service-auth/'
14. BACKEND_AUTH_USERNAME = 'pekhom_service'
15. BACKEND_AUTH_PASSWORD = 'pekhom_service123456789'
16. BACKEND_ACCOUNT_URL = 'http://192.168.1.162:5000/api/v1/system-services/user-info/'
17. BACKEND_USER_VALID_URL = 'http://192.168.1.162:5000/api/v1/system-services/user-pin-validation/'
18. BACKEND_SERVICE_NAME = 'PEKHOM Service'
19. BACKEND_SERVICE_KEY =
`c7ab0e1add3b6ba28af358e603d81da9fe5b1b3f73fa240b7d2dc1269bbf3f4.c903009324d8d568bacf33509c8438c80f524d1e711c8e2361a
20.

```

Abbildung 19: Google Hacking: Tokens: Ergebnis

Die letzte, und wohl kritischste Schwachstelle, die wir im Rahmen dieser Aufgabe gefunden haben, ist eine SQL-Injektion durch einen falsch programmierten PHP-Web-Server. Die Exploit-Datenbank listet hierzu drei typische Suchbefehle, wobei wir uns für den ersten entschieden haben, da er die größte Erfolgswahrscheinlichkeit hatte. (Siehe Abb. 20 & 21)

Date	Dork	Category	Author
2023-05-08	inurl:php?sql=select* ext:php	Files Containing Juicy Info	dmr
2015-12-08	intext:"Please select file to upload:" ext:php	Footholds	anonymous
2010-11-15	inurl:select_file2.php'	Advisories and Vulnerabilities	anonymous

Abbildung 20: Google Hacking: SQL Injektion: Schwachstelle

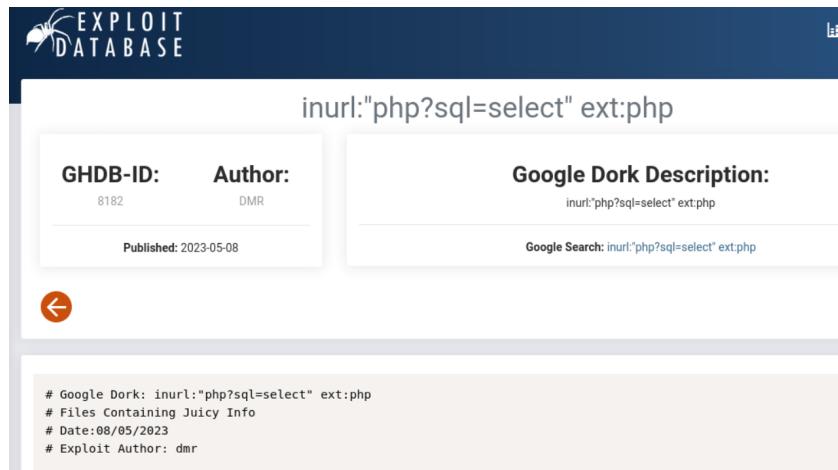


Abbildung 21: Google Hacking: SQL Injektion: Schwachstelle

Bei einer Google-Suche, kam neben einem PHP-Tutorial, sofort eine öffentliche PHP-Datei, die rohe SQL-Befehle als Query-Parameter annimmt. (Siehe Abb. 22)

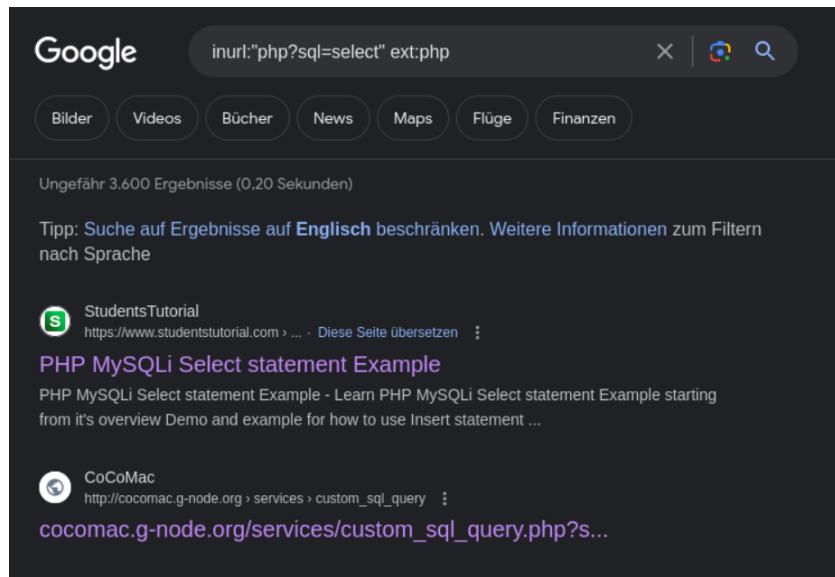
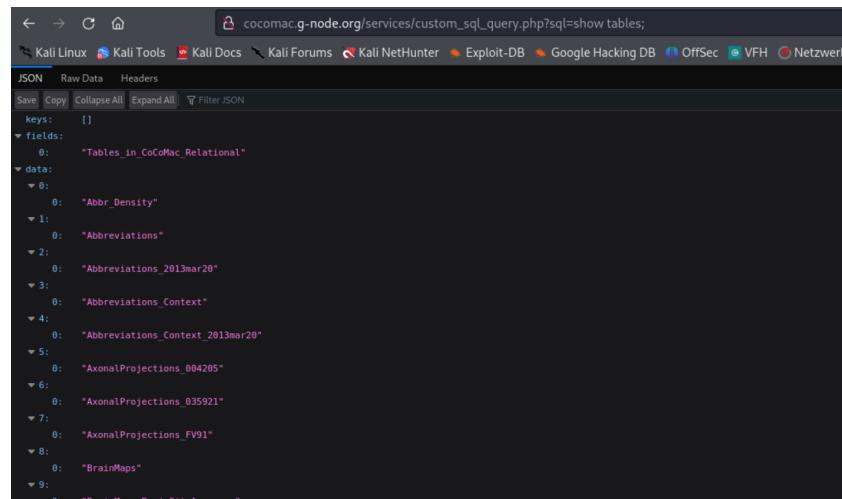


Abbildung 22: Google Hacking: SQL Injektion: Suche

Beim Aufruf dieser Seite wurden ursprünglich lediglich nicht vertrauliche Datensätze geladen, allerdings konnten wir zeigen, dass das SQL-Statement austauschbar ist, in dem wir (wie in Abb. 23 gezeigt) einen `show tables;` Befehl ausgeführt haben um die verschiedenen Tabellen in der angebundenen Datenbank aufzulisten.



```
keys: []
fields:
  0: "Tables_in_CoCoMac_Relational"
data:
  0:
    0: "Abbr_Density"
  1:
    0: "Abbreviations"
  2:
    0: "Abbreviations_2013mar20"
  3:
    0: "Abbreviations_Context"
  4:
    0: "Abbreviations_Context_2013mar20"
  5:
    0: "AxonalProjections_004205"
  6:
    0: "AxonalProjections_035921"
  7:
    0: "AxonalProjections_FV91"
  8:
    0: "BrainMaps"
  9:
    0: "CoCoMac_Relational"
```

Abbildung 23: Google Hacking: SQL Injektion: Ergebnis

Diese Schwachstelle könnte nun genutzt werden, um Beispielsweise neue Admin-Nutzer anzulegen, bestehende zu löschen, die Daten innerhalb der Datenbank zu manipulieren oder abzufragen etc. Über die Abfrage nach weiteren Tabellen hinaus, haben wir keine Befehle ausgeführt um strafrechtliche Probleme zu vermeiden.