mindsetters

ABSCHLUSS-REPORT

SECURITY CHECK – Acme AG

PROJEKT: BlackBox
Penetration Test (Web Applikation)
Super FancyWebShop

DATUM: 2. – 5. April 2024

mindsetters GmbH
Urstein Süd 15
5412 Puch bei Hallein
info@mindsetters.com

1. AUSGANGSSITUATION, UMFANG UND ZIELSETZUNG

Wir, die Firma mindsetters GmbH, wurden mit der Durchführung eines Penetrationstests für die Firma **Acme AG** (nachfolgend „Kunde" genannt) beauftragt. Ziel war es, die Webanwendung mit der Domain **webshop.acme.at** auf Sicherheitslücken zu untersuchen. Der abgestimmte Ansatz war eine Blackbox-Methodik ohne Vorkenntnisse über den Quellcode oder der Backend-Infrastruktur. Dadurch wurde sichergestellt, dass der Penetrationstest so realistisch wie möglich durchgeführt wurde, wie es auch externe Angreifer tun würden.

Die folgenden Personen waren involviert:

- Mike Mayers als Hauptansprechpartner auf Kundenseite
- Daniel Craig als Ansprechpartner für Programmcode (externer IT-Beauftragter)
- Sean Connery als Ansprechpartner für IT-Infrastruktur Angelegenheiten auf Kundenseite
- Roger Moore als technischer Ansprechpartner auf Applikationsseite
- Pierce Brosnan als Penetration Tester der mindsetters GmbH

Der Penetrationstest wurde zwischen dem **2. und 5. April 2024** durchgeführt. Für sämtliche Penetrationtest-Aktivitäten gab es auf Kunden- und Applikationsseite keine Einschränkungen hinsichtlich bevorzugter Tageszeiten.

GOALS AND LIMITATIONS

Das Hauptziel des Penetrationstests bestand darin, Schwachstellen zu finden, **welche die Integrität und Funktionsweise auf negative Weise beeinflussen**, sodass z.B. nicht-authorisierte Nutzer privilegierte Aktionen wie das Hinzufügen von zusätzlichen Usern hinzufügen durchführen können.

Es ist wichtig zu beachten, dass ein Penetrationstest nur eine Momentaufnahme ist, da sich die Bedrohungslage ständig weiterentwickelt. Darüber hinaus gibt es insbesondere bei Black-Box-Tests (ohne bekannten Quellcode) **keine Garantie** dafür, dass alle möglichen Schwachstellen gefunden werden. Ein hundertprozentiger Schutz kann zwar nie garantiert werden, die Wahrscheinlichkeit von Sicherheitsproblemen kann jedoch auf ein vertretbares Maß reduziert werden.

2. ZUSAMMENFASSUNG DER ERGEBNISSE

Während des Penetrationstests wurden 12 Schwachstellen identifiziert, wobei **1 davon mit kritischen, 1 mit hohen und 6 mit mittleren Risiko** einzustufen sind.

Sämtliche Ergebnisse und die damit verbundenen Risiken wurden mit dem standardisierten CVSS-Calculator Version 4.0 berechnet: https://www.first.org/cvss/calculator/4.0#

Hinweis: Details zu den einzelnen Schwachstellen sind im nächsten Abschnitt beschrieben (3. Schwachstellen – Details finden). Die in dieser Übersicht beschriebenen Schwachstellen wurden – soweit möglich – **in einfach verständlicher Sprache verfasst und dient üblicherweise nicht-technisch-versierten Personen**, um das einhergehende Risiko zu vermitteln.

VERTEILUNG DER ENTDECKTEN SCHWACHSTELLEN NACH RISIKOKLASSEN				
1 KRITISCH	1 HOCH	6 MITTEL	4 NIEDRIG	0 INFO

SCHWACHSTELLEN MIT ERHÖHTEM RISIKO, DIE ZEITNAH BEHOBEN WERDEN SOLLTEN			
ID	Kurzbeschreibung	Risiko	Behebung
ID-LA02-01	SQL-Injection (unauthentifiziert)	Auslesen von Datenbankinformationen wie z.B. Login-Daten (Passwörter) oder persönliche Informationen (Adressdaten); Exfiltrieren von lokalen Serverdaten	Benutzer-Eingaben vor der Verarbeitung validieren bzw. Einsatz von *Parameterisierten Abfragen*
ID-LA02-02	Cross-Origin Resource Sharing (CORS)	Auslesen von sensiblen Informationen wie Sendungs- oder Kundendaten (Schäden, Adressen)	Anpassen der CORS-Header, sodass nur vertrauliche Domains mit der WebApp interagieren dürfen
ID-LA02-03	Entwickler-Projekte ohne Einschränkungen zugänglich	Veröffentlich von sensiblen Informationen wie Adressdaten von Kunden oder Serverkonfigurationen	Zugriff nur für berechtigte Nutzer erlauben und *Directory Listing* deaktivieren
ID-LA02-04	Produktiv-Daten auf Testsystem gespeichert	Veröffentlichung von sensiblen realen Daten wie Kunden- oder Passwort-Informationen	Ausschließliche Nutzung von fiktiven Testdaten auf Testsystemen
ID-LA02-05	Passwörter im Klartext abgespeichert	Zugriff auf (Dritt-) Systeme bei denen die betroffenen Passwörter (wieder-) verwendet werden	Passwörter hashen und Salting anwenden
ID-LA02-06	Test- und Produktivdaten auf denselben DB-Server	Zugriff auf Produktivdaten durch Ausnützung von Schwachstellen; Erhöht die Chance von unbeabsichtigter Datenmanipulation und -Löschung	Physische bzw. virtuelle Trennung von Produktiv- und Test-Datenbanken
ID-LA02-07	Veraltete Software-Komponenten mit Schwachstellen	Offiziell dokumentierte Schwachstellen ermöglichen Angriffe die zum Ausfall von Webservern bzw. Daten-Manipulation führen	Aktualisierung der betroffenen Softwarekomponenten auf die neueste Version

ID-LA02-08	Login-Daten in GET-Request	Erhöht die Wahrscheinlichkeit, dass Dritte auf Login-Daten Zugriff haben	Sensible Daten (Passwörter) via POST-Request übermitteln

SCHWACHSTELLEN MIT GERINGEM RISIKO & INFORMELLE ANMERKUNGEN			
ID	**Kurzbeschreibung**	**Risiko**	Behebung
ID-LA02-09	Unsicherer Hash-Algorithmus für Passwort-Speicherung (SHA-1)	Rainbow-Tables und aktuelle Computer ermöglichen schnelle Berechnung und somit das Wiederherstellen des Klartext-Passwortes	Verwendung von aktuellen Hash-Algorithmen wie z.B. bcrypt
ID-LA02-10	Session Cookie ohne Secure Flag	Vereinfacht das Mitlesen von Datenverkehr des betroffenen Nutzers	Konfiguration des Secure Flags für TLS-Cookies auf Server-Seite
ID-LA02-11	Cookie ohne HttpOnly Flag	Vereinfacht die Durchführung von Attacken wie Benutzer-Session Übernahme	Konfiguration des HttpOnly Flags für Cookies auf Server-Seite
ID-LA02-12	Unverschlüsselte Kommunikation möglich	Unter bestimmten Voraussetzungen (Man-In-The-Middle) kann der vollständige Datenverkehr mitgelesen werden	Konfiguration des Response Headers „Strict-Transport-Security"

3. SCHWACHSTELLEN – DETAILS

Nachfolgend werden alle gefundenen Schwachstellen nach der generischen CVSS-Risikobewertung sortiert und aus technischer Sicht beschrieben. Jede Schwachstellen-Beschreibung enthält eine allgemeine CVSS-Bewertung (z. B. 9,3 – Kritisch), einen Titel und eine eindeutige ID zur einfacheren Referenzierung. Mittels einer Beschreibung der jeweiligen Schwachstelle wird auch das damit verbundene Risiko vermittelt, wobei auch äußere Rahmenbedingungen berücksichtigt werden (z.B. Firewall/WAF im Einsatz). Die schriftliche Darstellung des Risikos kann von der allgemeinen CVSS-Bewertung abweichen und stellt eine Annäherung an das tatsächliche Risiko dar. Der Abschnitt Behebung enthält Empfehlungen zur Lösung des Problems, die allgemeiner Natur sind wobei die tatsächliche Lösung von den Gegebenheiten des Kunden abhängen kann. Am Ende wird – wo sinnvoll und möglich – der Nachweis der Existenz einer Schwachstelle in einem Proof-Of-Concept vorgelegt.

9.3 Kritisch	SQL-Injection (unauthentifiziert)	ID-LA02-01

CVSS-Link https://www.first.org/cvss/calculator/4.0#CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N

Beschreibung

SQL-Injection-Schwachstellen entstehen, wenn vom Benutzer kontrollierbare Daten auf unsichere Weise in Datenbank-SQL-Abfragen integriert werden. Ein Angreifer kann manipulierte Eingaben ausführen, um aus dem Datenkontext, in dem seine Eingaben verarbeitet werden, auszubrechen und in die Struktur umliegenden Datenbank zuzugreifen (User-Passwörter, etc.).

Über SQL-Injection kann häufig eine Vielzahl schädlicher Angriffe erfolgen, darunter das Lesen oder Ändern kritischer Anwendungsdaten, Eingriffe in die Anwendungslogik, die Ausweitung von Berechtigungen innerhalb der Datenbank und die Übernahme der Kontrolle über den Datenbankserver oder gar des darunterliegenden Betriebssystems.

Über die bestehende Schwachstelle ist es per SQL-Injektion möglich, sämtlich Daten aus der Microsoft SQL-Datenbank zu extrahieren.

Bei den folgenden Endpunkten war eine SQL-Injektion nachweislich möglich (siehe Proof-Of-Concept):

- /t_n_t/Pdok.php [bez Parameter]
- /t_n_t/Pdok.php [typ1 Parameter]
- /t_n_t/Psd.php [ANR JSON Parameter innerhalb des json Parameters]
- /t_n_t/epl.php [sn Parameter]
- /t_n_t/Psd.php [PERSON JSON Parameter innerhalb des json Parameters]

Behebung

Der effektivste Weg, SQL-Injection-Angriffe zu verhindern, besteht darin, parametrisierte Abfragen (auch als „prepared Statements" bezeichnet) für alle Datenbankzugriffe zu verwenden. Dabei handelt es sich um eine Technik, die darauf abzielt, die SQL-Abfrage von den Benutzereingabewerten zu trennen.

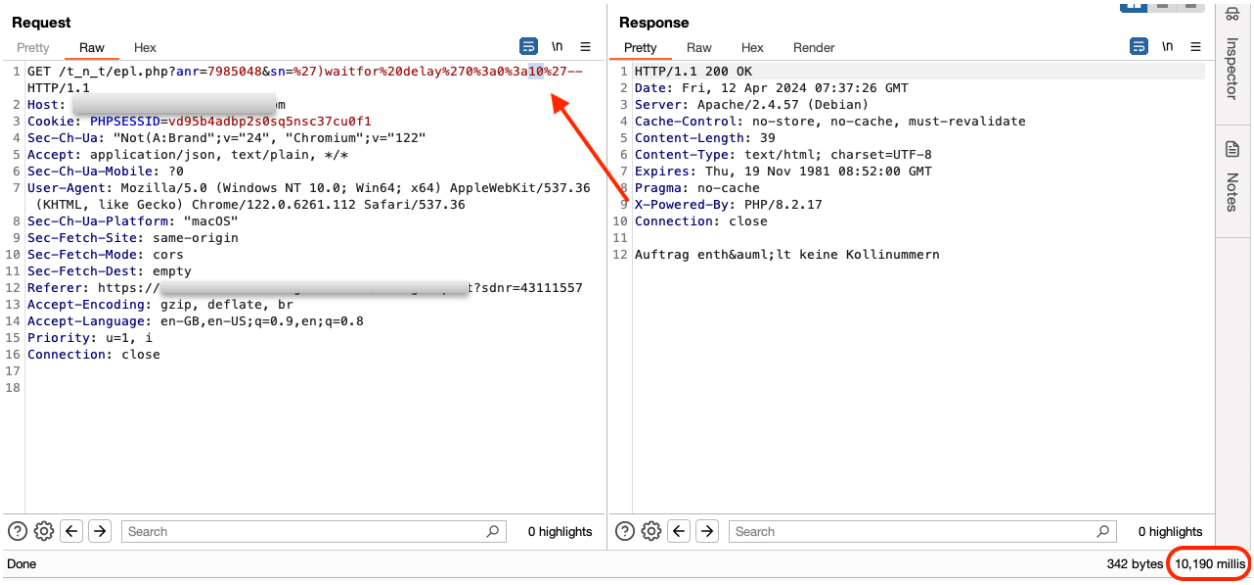
Die Benutzereingaben werden als Parameter übergeben. Sie können keinen ausführbaren Code mehr enthalten, da der Parameter als Literalwert behandelt und auf Typ und Länge überprüft wird.

Es wird dringend empfohlen, jede Variable zu parametrisieren, die in Datenbankabfragen integriert wird, auch wenn sie nicht offensichtlich fehlerhaft ist, um zu verhindern, dass es zu Versäumnissen kommt, und um zu verhindern, dass durch Änderungen an anderer Stelle in der Codebasis der Anwendung Schwachstellen entstehen.

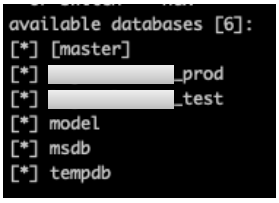
Proof-Of-Concept

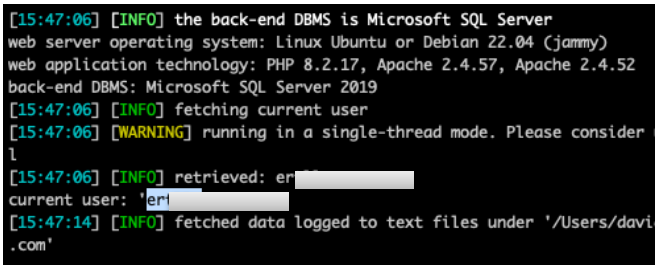
Der folgende Payload im Link löst eine Verzögerung in der Backend-Datenbank um 10 Sekunden aus:

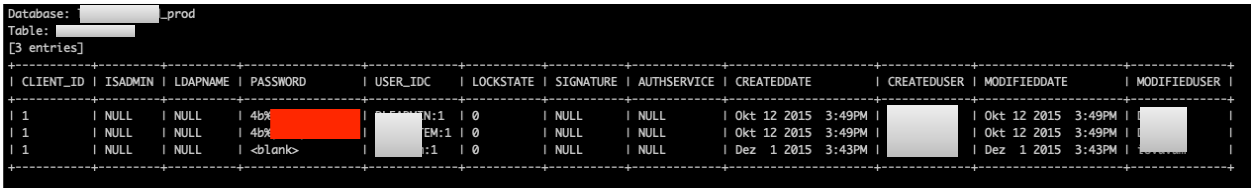
https://webshop.acme.at/epl.php?anr=7985048&sn=%27)waitfor delay%270%3a0%3a10%27--

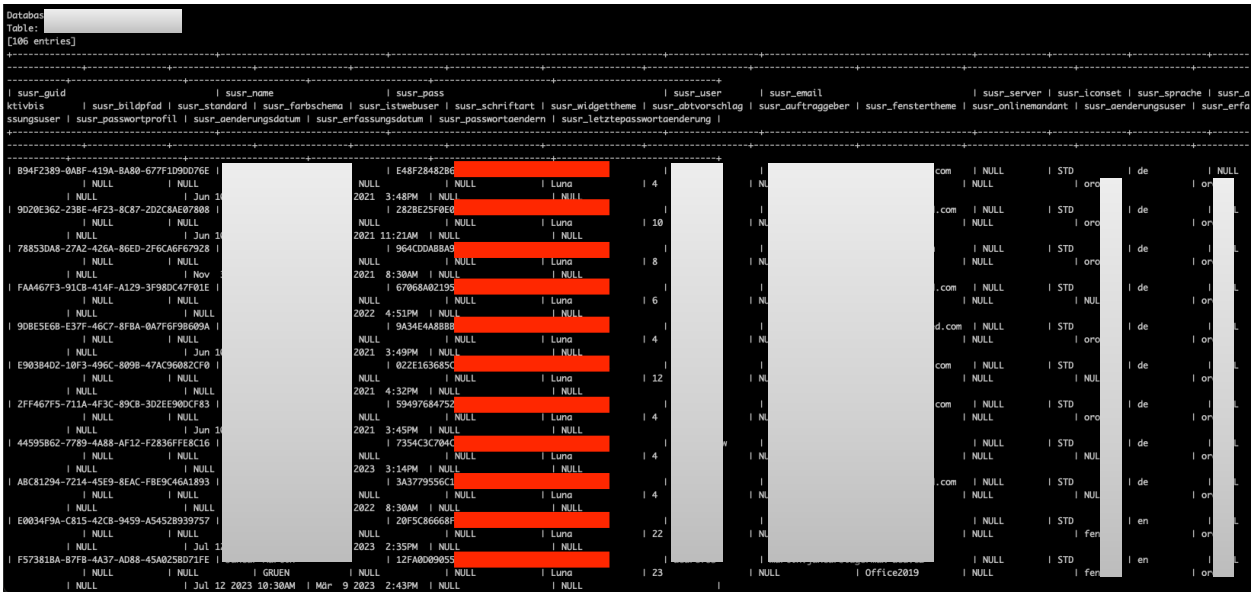


Bei den folgenden Abbildungen handelt es sich nur um einen kleinen Auszug der Daten, die aus der Microsoft SQL-Datenbank extrahiert werden konnten:








Database
Table:
[106 entries]
usr_guid	usr_name	usr_pass	usr_user	usr_email	usr_server	usr_iconset	usr_sprache	usr_aktivbis	usr_bildpfad	usr_standard	usr_farbschema	usr_istwebuser	usr_schriftart	usr_widgettheme	usr_abtvorschlag	usr_auftraggeber	usr_fenstertheme	usr_onlinemandant	usr_aenderungsuser	usr_erfassungsuser	usr_passwortprofil	usr_aenderungsdatum	usr_erfassungsdatum	usr_passwortaendern	usr_letztepasswortaenderung
894F2389-0ABF-419A-8A80-677F1D9DD76E	NULL	NULL	NULL	NULL	Luna	4	NULL	STD	de	NULL	NULL	Jun 10 2021 3:48PM	NULL	NULL
9D20E362-23BE-4F23-8C87-2D2C8AE07808	NULL	NULL	NULL	NULL	Luna	10	NULL	STD	de	NULL	NULL	Jun 10 2021 11:21AM	NULL	NULL
78853DA8-27A2-426A-86ED-2F6CA6F67928	NULL	NULL	NULL	NULL	Luna	8	NULL	STD	de	NULL	NULL	Nov 2021 8:30AM	NULL	NULL
FAA467F3-91CB-414F-A129-3F98DC47F01E	NULL	NULL	NULL	NULL	Luna	6	NULL	STD	de	NULL	NULL	NULL	2022 4:51PM	NULL	NULL
9DBE5E6B-E37F-46C7-8FBA-0A7F6F9B609A	NULL	NULL	NULL	NULL	Luna	4	NULL	STD	de	NULL	NULL	Jun 10 2021 3:49PM	NULL	NULL
E903B4D2-10F3-496C-809B-47AC96082CF0	NULL	NULL	NULL	NULL	Luna	12	NULL	STD	de	NULL	NULL	NULL	2021 4:32PM	NULL	NULL
2FF467F5-711A-4F3C-89CB-3D2EE90DCF83	NULL	NULL	NULL	NULL	Luna	4	NULL	STD	de	NULL	NULL	Jun 10 2021 3:45PM	NULL	NULL
44595B62-7789-4A88-AF12-F236FFE8C16	NULL	NULL	NULL	NULL	Luna	4	NULL	STD	de	NULL	NULL	NULL	2023 3:14PM	NULL	NULL
ABC81294-7214-45E9-8EAC-FBE9C46A1893	NULL	NULL	NULL	NULL	Luna	4	NULL	STD	de	NULL	NULL	NULL	2022 8:30AM	NULL	NULL
E0034F9A-C815-42CB-9459-A5452B939757	NULL	NULL	NULL	NULL	Luna	22	NULL	STD	en	NULL	NULL	Jul 12 2023 2:35PM	NULL	NULL
F573818A-B7FB-4A37-AD88-45A025BD71FE	NULL	NULL	GRUEN	NULL	Luna	23	NULL	Office2019	NULL	STD	en	NULL	Jul 12 2023 10:30AM	Mär 9 2023 2:43PM	NULL	NULL

7.1 Hoch	Cross-Origin Resource Sharing (CORS)	ID-LA02-02

CVSS-Link https://www.first.org/cvss/calculator/4.0#CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:P/VC:H/VI:N/VA:N/SC:N/SI:N/SA:N

Beschreibung

Die Anwendung implementiert für HTTP-Anfragen eine HTML5 CORS-Richtlinie (Cross-Origin Resource Sharing), die den Zugriff von jeder Domäne aus ermöglicht. Die Ziel-Webapplikation hat den Zugriff von der angeforderten Request-Source *https://www.mindsetters.com* zugelassen. Da der Header „Vary: Origin" in der Antwort nicht vorhanden war, wird er möglicherweise von Reverse-Proxys und Zwischenservern zwischengespeichert. Dadurch kann ein Angreifer möglicherweise Cache-Poisoning-Angriffe durchführen.

Eine HTML5 CORS-Richtlinie (Cross-Origin Resource Sharing) steuert, ob und wie Inhalte, die auf anderen Domänen ausgeführt werden. Weiters wird mit dieser Richtlinie entschieden, ob eine bidirektionale Interaktion mit einer bestimmten Domäne durchgeführt werden kann, die die Richtlinie definiert. Die Richtlinie ist variabel und kann Zugriffskontrollen pro Anfrage basierend auf der URL und anderen Merkmalen von HTTP-Requests anwenden.

Durch das Vertrauen auf beliebige Domänen wird die *Same-Origin-Richtlinie* effektiv deaktiviert und eine wechselseitige Interaktion durch Websites Dritter ermöglicht. Sofern die Antwort nicht nur aus ungeschützten öffentlichen Inhalten besteht, stellt diese Richtlinie wahrscheinlich ein Sicherheitsrisiko dar.

Wenn die Site den Header *Access-Control-Allow-Credentials: true* angibt, können Drittanbieter-Seiten potentiell privilegierte Aktionen ausführen und vertrauliche Informationen abrufen. Auch wenn dies nicht der Fall ist, können Angreifer u.U. IP-basierten Zugriffskontrollen umgehen, indem sie Proxys über die Browser der Benutzer verwenden.

Die folgenden Endpunkte wurden mit dieser Schwachstelle identifiziert:

- /Pdok.php
- /Perf.php
- /Psd.php
- /Psdq.php
- /Tagesuebersicht.php
- /bd.php
- /cll.php
- /cllsel.php
- /dglist.php
- /langjs.php
- /logon2.php
- /menu.php
- /mwb.php
- /mwbdetail.php
- /mwbemp.php
- /mwbkun.php

- /statovrw.php
- /stats.php

Behebung

Richtige Konfiguration von Cross-Origin-Anfragen

Wenn eine Webressource vertrauliche Informationen enthält, sollte die Quell-Domäne ordnungsgemäß im *Access-Control-Allow-Origin-Header* angegeben werden.

Lassen Sie nur vertrauenswürdige Websites zu

Es mag offensichtlich erscheinen, aber die im *Access-Control-Allow-Origin-Header* angegebenen Quell-Domänen sollten nur vertrauenswürdige Sites sein. Insbesondere die dynamische Widerspiegelung von Quellen aus *Cross-Origin-Anfragen* ohne Validierung ist leicht ausnutzbar und sollte vermieden werden.

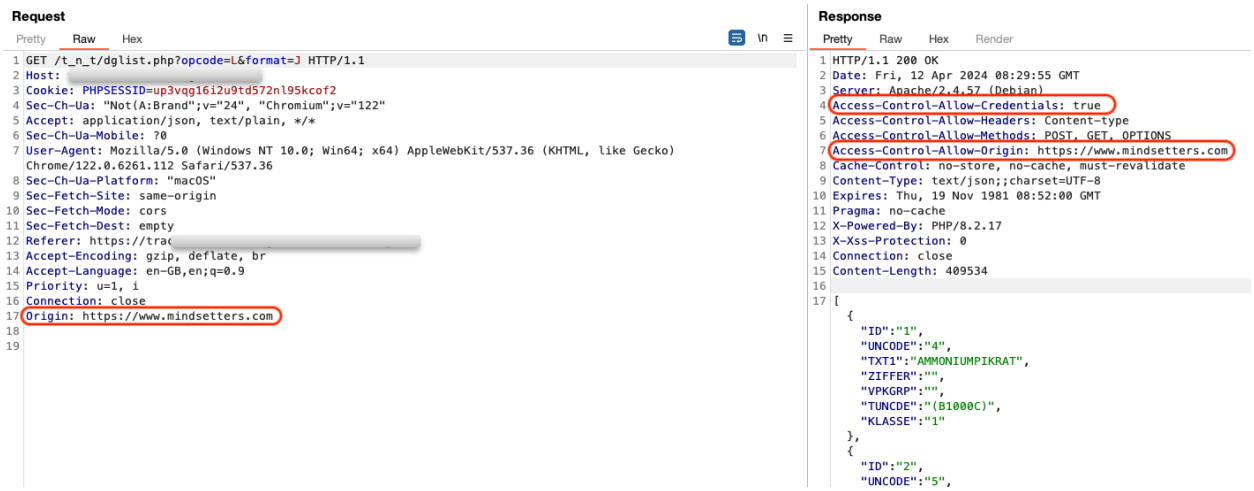
Vermeiden Sie es, den Wert *null* für die Whitelist zu verwenden

Vermeiden Sie die Verwendung des Headers *Access-Control-Allow-Origin: null*. Quellen-übergreifende Ressourcenaufrufe aus internen Dokumenten und Sandbox-Anfragen können als Quell *null* angeben. CORS-Header sollten im Hinblick auf vertrauenswürdige Quell-Domänen für private und öffentliche Server ordnungsgemäß definiert werden.

Vermeiden Sie Platzhalter in internen Netzwerken

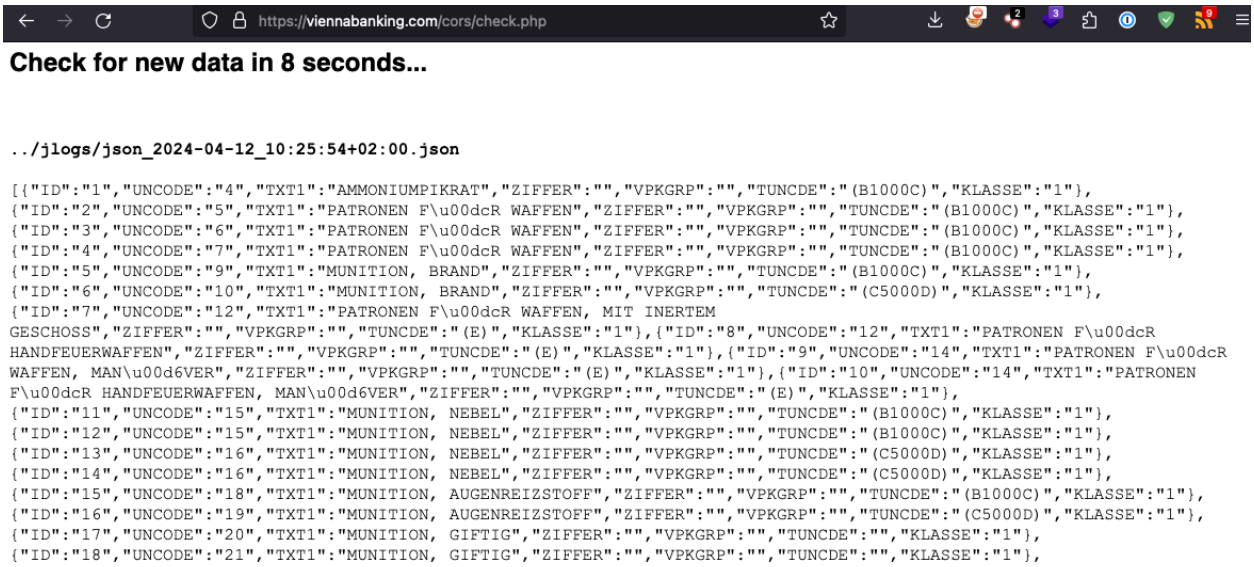
Vermeiden Sie die Verwendung von Platzhaltern in internen Netzwerken. Das alleinige Vertrauen in die Netzwerkkonfiguration zum Schutz interner Ressourcen reicht nicht aus, wenn interne Browser auf nicht vertrauenswürdige externe Domänen zugreifen können.

Proof-Of-Concept



Zum Ausnutzen der Schwachstelle reicht der Besuch einer von Angreifern präperierte Webseite aus:

Nach Aufruf der Angreifer-Webseite erhält dieser die folgenden Rückgabe-Informationen:



4. HERANGEHENSWEISE UND TEST-METHODIK

Alle Angriffstechniken und verwendeten Tools orientierten sich zum Großteil an den OWASP Top 10 in ihrer aktuellen Version.

Die folgenden Aktivitäten wurden nach bestem Wissen und Gewissen während des Penetrationstests durchgeführt – unabhängig davon, ob Schwachstellen identifiziert wurden oder nicht:

- Verwundbare Datei-Uploads
- SQL-Injection-Angriffe
- XSS-Angriffe (Cross-Site-Scripting)
- Cross-Site Request Forgery (CSRF)
- File-Inclusion-Angriffe
- Portscans des Webservers (Nmap)
- Überprüfung auf unverschlüsselte Kommunikation (z. B. HTTP)
- HTTP-Smuggling
- Open Redirection Angriffe
- Cookie-Manipulation
- Directory Listing Unterseiten
- Web-Cache-Poisoning-Angriffe
- Prüfung auf möglichen unauthorisierten Informations- bzw. Datenzugriff
- Auffinden potenzieller Zugriffs- und Autorisierungsprobleme
- Auffinden bekannter Schwachstellen in Software von Drittanbietern und anderen Komponenten von Drittanbietern (Bibliotheken, Vorlagen, Frameworks, etc.)

Methodik

Die Tests wurden hauptsächlich manuell durchgeführt. Zur Effizienzsteigerung werden unterstützend automatische Werkzeuge eingesetzt. Unterstützende Tools – insbesondere solche, die eine hohe Anzahl an Anfragen verursachen – werden so konfiguriert, dass das Kommunikationsvolumen auf ein für die Systeme erträgliches Maß reduziert wird (sofern keine Schwachstelle identifiziert wird, die Gegenteiliges behauptet).

Sofern nicht ausdrücklich anders angegeben, erfolgt die Überprüfung als Black-Box-Test mit erfahrungsbasierter Priorisierung nach Zeit und Umfang. Die Dauer und der Umfang der Prüfung wurden auf Basis von Erfahrungswerten anhand der Angaben des Kunden und einer Risiko-Kosten-Nutzen-Abwägung unsererseits festgelegt.