

Pentest Report

Demo-Report: Externe Infrastruktur

Pentest für **Security Maximale GmbH**
11. September 2022

Version 1.0

Inhalt

Management Summary	2
Hier ist der Bericht. Was jetzt?	3
Scope und Dauer	4
Schwachstellenübersicht	5
Schwachstellendetails	7
Confluence Unauthentifizierte Remote Code Execution (Critical)	7
WordPress Default Login (Critical)	9
Öffentliches .git-Verzeichnis (High)	11
Subdomain Takeover (Medium)	13
Zugangsdaten in öffentlichen Passwort Dumps (Medium)	15
Unzureichende E-Mail-Authentifizierung (Low)	17
Änderungsverzeichnis	20
Disclaimer	20
Impressum	20

Management Summary

Im Zuge der Sicherheitsprüfung war es uns möglich, zwei Infrastrukturserver zu übernehmen. Dadurch konnten wir auf sensible interne Projektdokumentationen zugreifen sowie Kundendaten einsehen und bearbeiten.

Dies gelang über zwei Wege: Einerseits war eine Confluence Server-Instanz veraltet und anfällig für eine Remote Code Execution Schwachstelle. Andererseits konnte auf ein WordPress-Backend mit dem Standardpasswort des Administratorkontos zugegriffen werden. Durch Installation eines Plugins konnte auf der WordPress Instanz ebenfalls Remote Code Execution erlangt werden.

Zudem war ein .git-Verzeichnis auf einem Webserver offenbart und damit öffentlich über das Internet zugänglich. Der Source Code der Anwendung konnte dadurch rekonstruiert und auf sensible Inhalte (z.B. Zugangsdaten, API-Keys, etc.) sowie Schwachstellen untersucht werden.

Weitere, weniger kritische Schwachstellen, wie etwa ein Subdomain Takeover, das Vorkommen von Zugangsdaten in öffentlichen Passwort-Dumps, sowie unzureichende E-Mail Authentifizierung sollten in einem kontinuierlichen Verbesserungsprozess adressiert werden.

Hier ist der Bericht. Was jetzt?

Schwachstellen mit weniger komplexen Gegenmaßnahmen und Risiko **Medium** und darunter sollten nach unserer Empfehlung nach Aufwand priorisiert behoben werden. Alle anderen Schwachstellen sollten im Rahmen eines kontinuierlichen Verbesserungsprozesses adressiert werden.

Bitte stellt sicher, alle im Zuge des Pentests bereitgestellten Benutzer und Ressourcen zu deprovisionieren, sobald sie nicht mehr benötigt werden.

Scope und Dauer

Der Scope des Pentests umfasste die folgenden öffentlich erreichbaren Systeme:

- `www.example.com`
- `confluence.example.com`
- `git.example.com`
- `mail.example.com`

Der Penetration-Test erfolgte nach einem Time-Box-Ansatz und umfasste 5 Personentage.

Schwachstellenübersicht

Im Rahmen dieses Penetration Tests wurden **2 Critical**, **1 High**, **2 Medium** und **1 Low** Schwachstellen identifiziert:

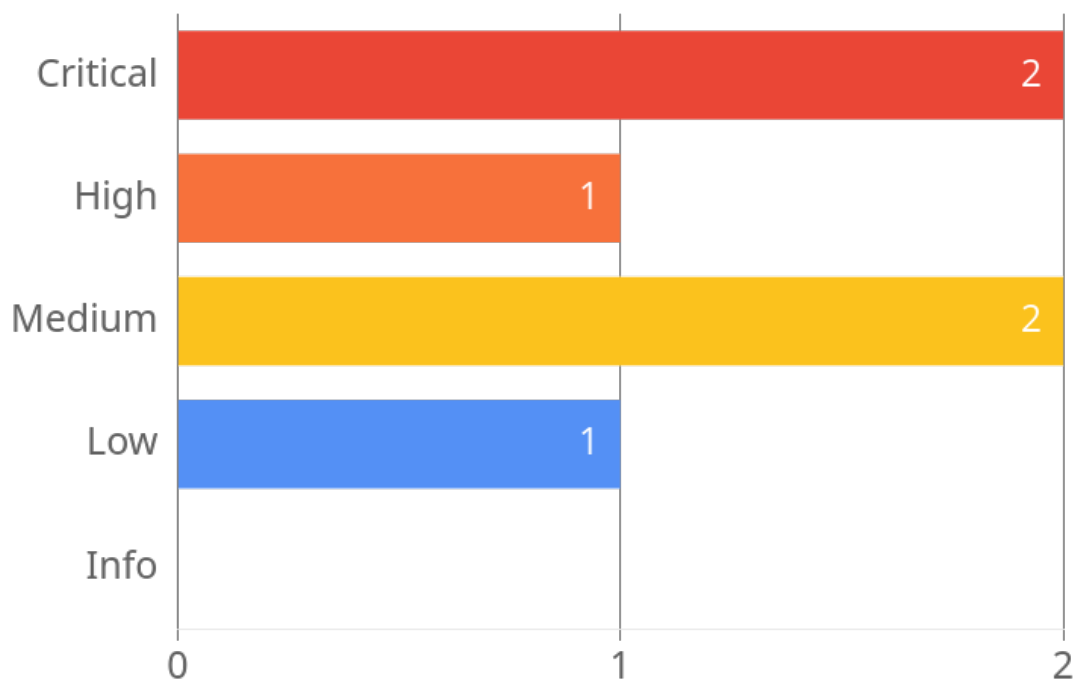


Abbildung 1 - Verteilung der gefundenen Schwachstellen

Finding	Severity	Behebungsstatus
Confluence Unauthentifizierte Remote Code Execution	Critical	Offen
WordPress Default Login	Critical	Behoben
Öffentliches .git-Verzeichnis	High	Offen
Subdomain Takeover	Medium	Offen
Zugangsdaten in öffentlichen Passwort Dumps	Medium	Akzeptiert
Unzureichende E-Mail-Authentifizierung	Low	Geändert

1. Confluence Unauthentifizierte Remote Code Execution (**Critical: 10.0** | **Offen**)

Betrifft: confluence.example.com

Die installierte Confluence Server-Instanz ist anfällig für eine Remote Code Execution Schwachstelle (CVE-2022-26134), die unauthentifziert ausgenutzt werden kann. Aufgrund einer Object-Graph Navigation Language (OGNL) Injection Schwachstelle im Webserver,

können von remote beliebige Systemkommandos ausgeführt werden. Dies erlaubt es einem Angreifer den zugrundeliegenden Server vollständig zu übernehmen.

2. WordPress Default Login (Critical: 9.8 | Behoben)

Betrifft: `www.example.com/wp-login.php`

Die WordPress-Webseite hatte das Standardpasswort des Administratorkontos (`admin`) gesetzt, wodurch mit höchsten Berechtigungen zugegriffen auf das Backend werden konnte. Ein Angreifer könnte durch Installation eines WordPress-Plugins eine Webshell hochladen und dadurch den zugrundeliegenden Webserver vollständig übernehmen.

3. Öffentliches .git-Verzeichnis (High: 7.5 | Offen)

Betrifft: `www.example.com/.git`

Der Webserver offenbarte das `.git`-Verzeichnis der Webanwendung. Der Source Code der Anwendung kann dadurch rekonstruiert werden und ist damit öffentlich über das Internet zugänglich. Angreifer könnten Zugriff auf sensible Informationen wie Zugangsdaten, hart-kodierte API-Keys oder Entwicklerkommentare erhalten.

4. Subdomain Takeover (Medium: 5.8 | Offen)

Betrifft: `git.example.com`

Die Subdomäne `git.example.com` war anfällig für Subdomain Takeover. Im DNS war ein CNAME-Record hinterlegt, welcher auf eine verwaiste GitHub-Page referenzierte. Ein Angreifer könnte damit die Kontrolle über die Subdomain erlangen und hätte gegebenenfalls Zugriff auf geschützte Informationen (z. B. Cookies) oder die Möglichkeit, bösartige Skripte einzuschleusen.

5. Zugangsdaten in öffentlichen Passwort Dumps (Medium: 5.3 | Akzeptiert)

Betrifft: `example.com`

Im Rahmen des Pentests konnten zahlreiche Zugangsdaten (E-Mail Adressen, Passwörter, Hashes, etc.) von Mitarbeitern in öffentlichen Passwort-Dumps gefunden werden. Angreifer können sich diese Informationen zunutze machen und sich so womöglich Zugriff auf betroffene Benutzerkonten und Services (z.B. VPN, Azure AD, etc.) verschaffen.

6. Unzureichende E-Mail-Authentifizierung (Low: 3.7 | Geändert)

Betrifft: `example.com`

Die Domain `example.com` wies fehlende Einstellungen hinsichtlich der E-Mail-Authentifizierung auf. Die drei Mechanismen der E-Mail Authentifizierung sind Sender Policy Framework (SPF), DomainKeys Identified Mail (DKIM) und Domain-based Authentication, Reporting & Conformance (DMARC). Wenn diese Mechanismen fehlen oder falsch konfiguriert sind, können Angreifer in der Lage sein, E-Mails im Namen der Domain zu versenden. Empfänger könnten dadurch z.B. im Zuge eines Phishing-Angriffs getäuscht und zu unbeabsichtigten Aktionen bewegt werden. Erfolgreiche Angriffe erhöhen außerdem die Wahrscheinlichkeit, dass E-Mails der betroffenen Domäne zukünftig als Spam erkannt werden und ihre Empfänger damit nicht mehr erreichen.

Schwachstellendetails

1. Confluence Unauthentifizierte Remote Code Execution

Behebungsstatus: **Offen**

Kritikalität: **Critical**

CVSS-Score: **10.0** | CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H

Betrifft: confluence.example.com

Überblick

Die installierte Confluence Server-Instanz ist anfällig für eine Remote Code Execution Schwachstelle (CVE-2022-26134), die unauthentifiziert ausgenutzt werden kann. Aufgrund einer Object-Graph Navigation Language (OGNL) Injection Schwachstelle im Webserver, können von remote beliebige Systemkommandos ausgeführt werden. Dies erlaubt es einem Angreifer den zugrundeliegenden Server vollständig zu übernehmen.

Beschreibung

Im Zuge der Prüfung konnten wir unter confluence.example.com eine veraltete Confluence Server Version identifizieren.

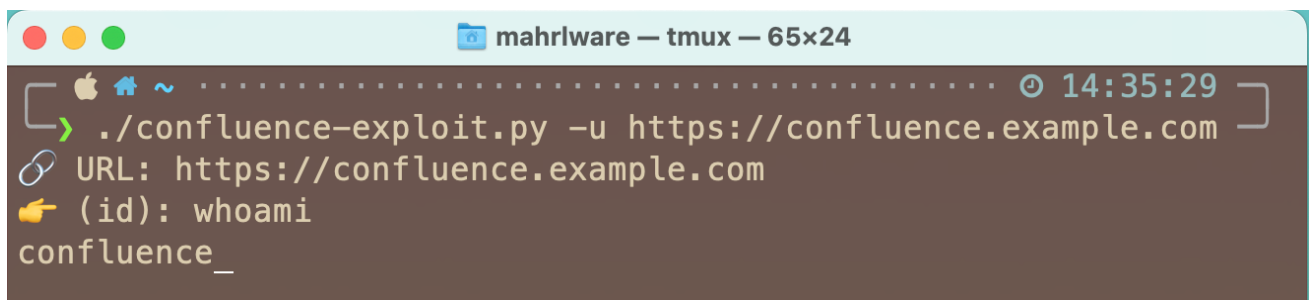
A screenshot of a terminal window titled 'mahrlware — tmux — 65x24'. The terminal shows a command prompt with a green prompt character. The user enters the command './confluence-exploit.py -u https://confluence.example.com'. The output shows 'URL: https://confluence.example.com' and '(id): whoami' followed by a new prompt 'confluence_'. The terminal window has a dark background with light-colored text. The top bar of the window shows standard macOS window controls (red, yellow, green buttons) and the title 'mahrlware — tmux — 65x24'. The bottom right corner of the terminal shows the time '14:35:29'.

Abbildung 2 - Confluence Remote Code Execution

Confluence ist eine kommerzielle Wiki-Software, die vom Unternehmen Atlassian entwickelt und als Enterprise Wiki für die Dokumentation und Kommunikation von Wissen sowie den Wissensaustausch in Unternehmen und Organisationen verwendet wird. Am 02. Juni 2022 veröffentlichte Atlassian eine Security Advisory für Confluence Server, in welchem auf eine kritische Schwachstelle mit der CVE-Nummer CVE-2022-26134 hingewiesen wurde. Die Schwachstelle erlaubte es nicht authentifizierten Benutzern beliebigen Code auf einer Confluence Server-Instanz auszuführen.

Empfehlung

- Der Patch für CVE-2022-26134 wurde am 3. Juni, 2022 von Atlassian veröffentlicht. Weiterführende Informationen zur Schwachstelle sind in der Security Advisory von Atlassian, die in den Referenzen verlinkt ist, zu finden.
- Systeme und Software sollten durch regelmäßige Aktualisierungen z.B. im Zuge eines Patch-Management-Prozesses auf dem neuesten Stand zu halten.
- Kritische Schwachstellen und Schwachstellen mit hohem Risiko sollten priorisiert und ehestmöglich (z.B. innerhalb von 2 Wochen) aktualisiert werden.
- Je nach Anwendungsfall empfehlen wir auch den Einsatz automatischer Update-Mechanismen.

Weiterführende Informationen

- <https://confluence.atlassian.com/doc/confluence-security-advisory-2022-06-02-1130377146.html>

2. WordPress Default Login

Behebungsstatus: **Behoben**

Kritikalität: **Critical**

CVSS-Score: **9.8** | CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

Betrifft: `www.example.com/wp-login.php`

Überblick

Die WordPress-Webseite hatte das Standardpasswort des Administratorkontos (`admin`) gesetzt, wodurch mit höchsten Berechtigungen zugegriffen auf das Backend werden konnte. Ein Angreifer könnte durch Installation eines WordPress-Plugins eine Webshell hochladen und dadurch den zugrundeliegenden Webserver vollständig übernehmen.

Anmerkungen zu Behebungsstatus

Die Zugangsdaten wurden noch während des laufenden Pentests geändert.

Beschreibung

Die Webseite unter `www.example.com` wurde im Content-Management-System WordPress gehostet. Das Backend-Login war unter `www.example.com/wp-login.php` erreichbar. Im Zuge der Prüfung konnte auf das Backend mit dem Administratorkonto `admin` mit dem Standardpasswort zugegriffen werden.

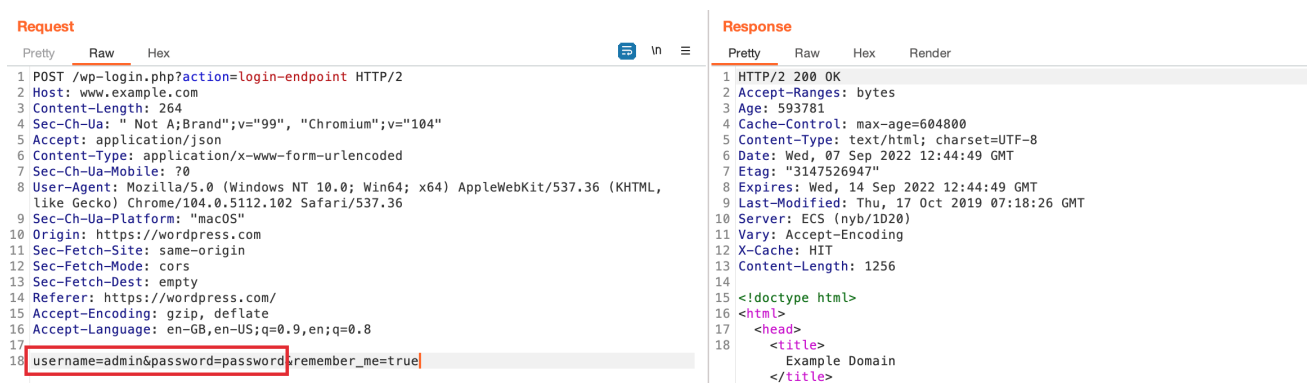


Abbildung 3 - WordPress Default Login

WordPress ist ein weit verbreitetes Content-Management-System, das für die Erstellung von Websites und Blogs verwendet wird. Wenn nicht anders bei der Installation angegeben, wird standardmäßig `password` als Initialpasswort für das Administratorkonto `admin` verwendet.

Empfehlung

- Das Passwort des Administratorkontos `admin` sollte umgehend geändert werden.
- Erzwingt eine strenge Passwortrichtlinie. Ein starkes Passwort wird durch folgende Merkmale definiert:
 - Es sollte mindestens 8 Zeichen lang sein.

- Es sollte aus Groß- und Kleinbuchstaben, Ziffern und Sonderzeichen bestehen.
- Es sollte kein häufig verwendetes Passwort sein (z.B. Zahlenfolge, Buchstabenfolge, Wörterbucheintrag, etc).
- Erlaubt für Passwörter die Verwendung aller Zeichen, einschließlich Unicode und Leerzeichen. Stellt dabei sicher, dass es keine Regeln gibt, welche die Zusammensetzung von Passwörtern vorgeben und/oder die Art der zulässigen Zeichen einschränken.
- Setzt die maximale Passwortlänge nicht zu niedrig an. Benutzer würden dadurch abgehalten werden, Passphrasen zu verwenden. Achten Sie auch darauf, dass sie Passwörter beim Speichern niemals abkürzen.
- Erwägt zusätzliche Authentifizierungskontrollen wie z. B. Zwei-Faktor-Authentifizierung.
- Detaillierte Informationen und Hilfestellungen wie ihr einen sicheren Authentifizierungsmechanismus umsetzen könnt, findet ihr im verlinkten Authentication Cheat Sheet von OWASP.

Weiterführende Informationen

- https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html

3. Öffentliches .git-Verzeichnis

Behebungsstatus: **Offen**

Kritikalität: **High**

CVSS-Score: **7.5** | CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N

Betrifft: `www.example.com/.git`

Überblick

Der Webserver offenbarte das `.git`-Verzeichnis der Webanwendung. Der Source Code der Anwendung kann dadurch rekonstruiert werden und ist damit öffentlich über das Internet zugänglich. Angreifer könnten Zugriff auf sensible Informationen wie Zugangsdaten, hart-kodierte API-Keys oder Entwicklerkommentare erhalten.

Beschreibung

Im Zuge der Prüfung konnte für `www.example.com` ein offenes `.git`-Verzeichnis festgestellt werden.

Request				Response			
Pretty	Raw	Hex		Pretty	Raw	Hex	Render
1 GET /.git HTTP/2				1 HTTP/2 200 OK			
2 Host: www.example.com				2 Age: 540773			
3 Sec-Ch-Ua: " Not A;Brand";v="99", "Chromium";v="104"				3 Cache-Control: max-age=604800			
4 Sec-Ch-Ua-Mobile: ?0				4 Content-Type: text/html; charset=UTF-8			
5 Sec-Ch-Ua-Platform: "macOS"				5 Date: Wed, 07 Sep 2022 09:13:47 GMT			
6 Upgrade-Insecure-Requests: 1				6 Etag: "3147526947+gzip"			
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.5112.102 Safari/537.36				7 Expires: Wed, 14 Sep 2022 09:13:47 GMT			
8 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9				8 Last-Modified: Thu, 17 Oct 2019 07:18:26 GMT			
9 Sec-Fetch-Site: none				9 Server: ECS (nyb/1D10)			
10 Sec-Fetch-Mode: navigate				10 Vary: Accept-Encoding			
11 Sec-Fetch-User: ?1				11 X-Cache: HIT			
12 Sec-Fetch-Dest: document				12 Content-Length: 1256			
13 Accept-Encoding: gzip, deflate				13			
14 Accept-Language: en-GB,en-US;q=0.9,en;q=0.8				14 <!doctype html>			
15				15 <html>			
16				16 <head>			
17				17 <title>			
18				18 Example Domain			
19				19 </title>			
20				20			

Abbildung 4 - Öffentlich erreichbares .git-Verzeichnis

Source Code kann durch ein über den Webserver zugängliches `.git`-Verzeichnis an die Öffentlichkeit gelangen. Bei Verwendung von Git zur Versionskontrolle wird standardmäßig ein `.git`-Ordner im Root-Verzeichnis des Projekts erstellt, der alle Informationen des Projekts, einschließlich der Commit-Historie der Projektdateien, speichert. Der `.git`-Ordner sollte nicht für die Öffentlichkeit zugänglich sein, was jedoch manchmal aufgrund fehlerhafter Webserver-Konfiguration versehentlich passiert.

Durch den Zugriff auf das `.git`-Verzeichnis kann der Source Code des Projekts öffentlich zugänglich werden. Ein Angreifer könnte im Source Code schließlich nach Zugangsdaten, Encryption-Keys, API-Tokens und Entwicklerkommentaren suchen. Auch das Finden potenzieller Schwachstellen für etwaige Folgeangriffe wird dadurch erleichtert.

Empfehlung

- Um den Source Code der Anwendung zu schützen, sollte das `.git`-Verzeichnis nicht öffentlich erreichbar sein.

- Git-Metadaten sollten daher vom Webroot-Verzeichnis des Webservers entfernt bzw. der Zugriff darauf eingeschränkt werden.

4. Subdomain Takeover

Behebungsstatus: **Offen**

Kritikalität: **Medium**

CVSS-Score: **5.8** | CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:N/I:L/A:N

Betrifft: git.example.com

Überblick

Die Subdomäne *git.example.com* war anfällig für Subdomain Takeover. Im DNS war ein CNAME-Record hinterlegt, welcher auf eine verwaiste GitHub-Page referenzierte. Ein Angreifer könnte damit die Kontrolle über die Subdomain erlangen und hätte gegebenenfalls Zugriff auf geschützte Informationen (z. B. Cookies) oder die Möglichkeit, bösartige Skripte einzuschleusen.

Beschreibung

Im Zuge der Prüfung stellten wir die Möglichkeit eines Subdomain Takeovers für die Subdomäne *git.example.com* fest. Für die Subdomäne existierte ein CNAME-Record im DNS, welcher auf eine nicht (mehr) existente GitHub-Page verwies.

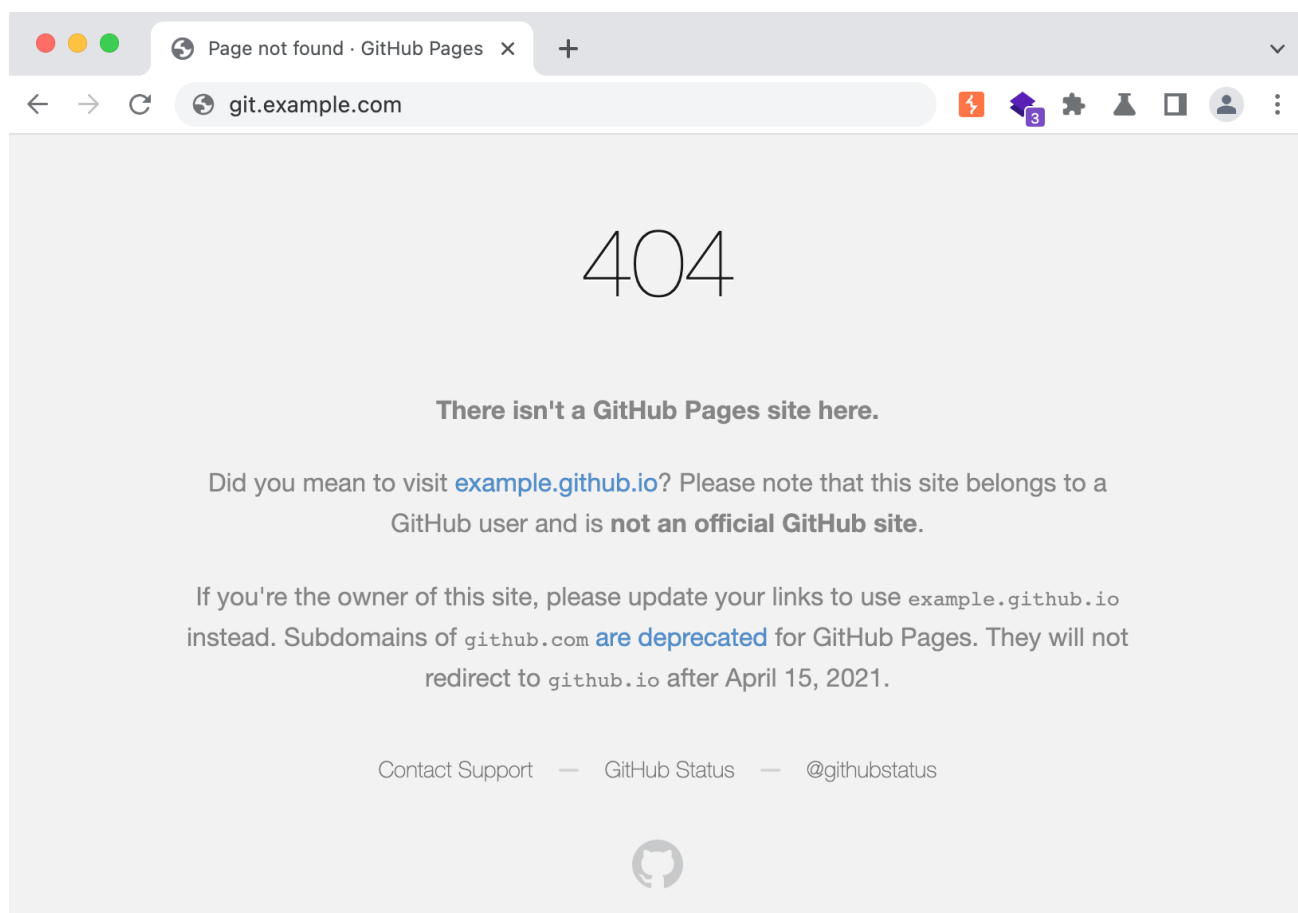


Abbildung 5 - Subdomain Takeover von *git.example.com*

Ein Subdomain Takover liegt vor, wenn ein Angreifer die Kontrolle über eine Subdomain einer Zieldomain erlangt. Das ist dann der Fall, wenn die Subdomain einen CNAME im DNS hinterlegt hat, die auf einen Dienst verweist (z.B. GitHub, AWS/S3, etc.) der entfernt oder gelöscht wurde. Ein Angreifer kann diese Subdomäne übernehmen, indem er einen eigenen virtuellen Host bereitstellt und dann seine eigenen Inhalte für sie hostet.

Wenn beispielsweise `subdomain.example.com` auf eine GitHub-Seite verwies und der Benutzer beschloss, seine GitHub-Seite zu löschen, kann ein Angreifer nun eine GitHub-Seite erstellen und dadurch `subdomain.example.com` für sich beanspruchen.

Gelingt einem Angreifer ein Subdomain Takover, kann er unter Umständen Cookies der Hauptdomäne lesen, Cross-Site-Scripting durchführen oder eine ggfs. gesetzte Content-Security-Policy (CSP) umgehen. Damit hätte er Zugriff auf geschützte Informationen (einschließlich Anmeldungen) oder die Möglichkeit böartige Inhalte an Benutzer zu senden.

Empfehlung

- Entfernt den CNAME-Record für die Subdomäne `git.example.com` aus dem DNS
- Weiters ist die Erstellung eines Inventars aller Domänen und der zugehörigen Hosting-Anbieter empfehlenswert. Das Inventar sollte gepflegt sein und bei Änderungen auf einen aktuellen Stand gebracht werden.
- Definiert dazu einen Standardprozess für das Provisioning und Deprovisioning von Hosts, um sicherzustellen, dass virtuelle Hosts und DNS-Einträge in der richtigen Reihenfolge erstellt bzw. gelöscht werden. Bei der Deprovisionierung empfehlen wir mit der Entfernung der DNS-Einträge zu beginnen.

5. Zugangsdaten in öffentlichen Passwort Dumps

Behebungsstatus: **Akzeptiert**

Kritikalität: **Medium**

CVSS-Score: **5.3** | CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N

Betrifft: example.com

Überblick

Im Rahmen des Pentests konnten zahlreiche Zugangsdaten (E-Mail Adressen, Passwörter, Hashes, etc.) von Mitarbeitern in öffentlichen Passwort-Dumps gefunden werden. Angreifer können sich diese Informationen zunutze machen und sich so womöglich Zugriff auf betroffene Benutzerkonten und Services (z.B. VPN, Azure AD, etc.) verschaffen.

Beschreibung

Mittels Dehashed wurde online nach Zugangsdaten für die Domäne **example.com** gesucht. Die Suche ergab **548799** verschiedene Einträge (inklusive Duplikate). Es konnten zahlreiche Email Adressen, Klartextpasswörter, Passwort-Hashes sowie personenbezogene Daten in verschiedenen Dumps identifiziert werden.

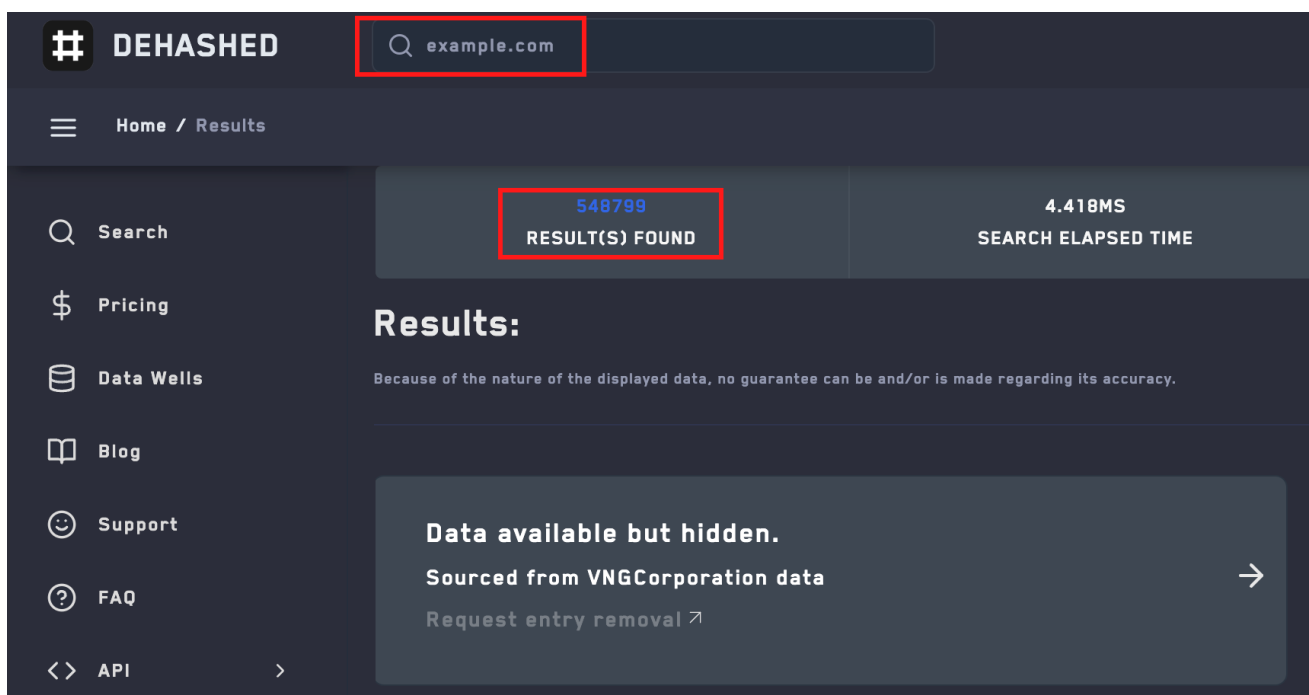


Abbildung 6 - Zugangsdaten in öffentlichen Passwort-Dumps

Die betroffenen Benutzerkonten und Klartextpasswörter sind dem Bericht beigelegt.

Passwort-Dumps sind umfangreiche Listen mit oft Millionen von erbeuteten Nutzernamen-Passwort-Kombinationen, die nach einem Datendiebstahl von Dieben öffentlich im Internet bereitgestellt werden. Diese Informationen könnten in gezielten Angriffen verwendet werden. Im Worst-Case findet ein Angreifer valide Benutzerzugänge, um beispielsweise Zugriff auf interne Services z.B. über ein VPN zu erhalten.

Empfehlung

- Die Passwörter betroffener Benutzerkonten sollten ggfs. zurückgesetzt werden.
- Um das Risiko in einem Passwort-Dump zu landen zu reduzieren, sollten geschäftliche Mailadressen nicht für private Zwecke verwendet werden. Mitarbeiter sollten diesbezüglich hinreichend aufgeklärt werden.
- Wir empfehlen ein laufendes Monitoring der Domäne z.B. mithilfe von Services wie [Kaduu](#).

Weiterführende Informationen

- <https://haveibeenpwned.com>
- <https://www.dehashed.com>

6. Unzureichende E-Mail-Authentifizierung

Behebungsstatus: **Geändert**

Kritikalität: **Low**

CVSS-Score: **3.7** | CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:N/A:L

Betrifft: example.com

Überblick

Die Domain `example.com` wies fehlende Einstellungen hinsichtlich der E-Mail-Authentifizierung auf. Die drei Mechanismen der E-Mail Authentifizierung sind Sender Policy Framework (SPF), DomainKeys Identified Mail (DKIM) und Domain-based Authentication, Reporting & Conformance (DMARC). Wenn diese Mechanismen fehlen oder falsch konfiguriert sind, können Angreifer in der Lage sein, E-Mails im Namen der Domain zu versenden. Empfänger könnten dadurch z.B. im Zuge eines Phishing-Angriffs getäuscht und zu unbeabsichtigten Aktion bewegt werden. Erfolgreiche Angriffe erhöhen außerdem die Wahrscheinlichkeit, dass E-Mails der betroffenen Domäne zukünftig als Spam erkannt werden und ihre Empfänger damit nicht mehr erreichen.

Beschreibung

Im Zuge der Prüfung stellten wir fehlende bzw. fehlerhafte Einstellungen hinsichtlich E-Mail Authentifizierung in der Domäne `example.com` fest. Die betroffene Domäne wies weder SPF, noch DKIM und DMARC Records auf. Dieser Umstand konnte ausgenutzt werden, um Phishing-Mails im Namen dieser Domäne zu versenden.

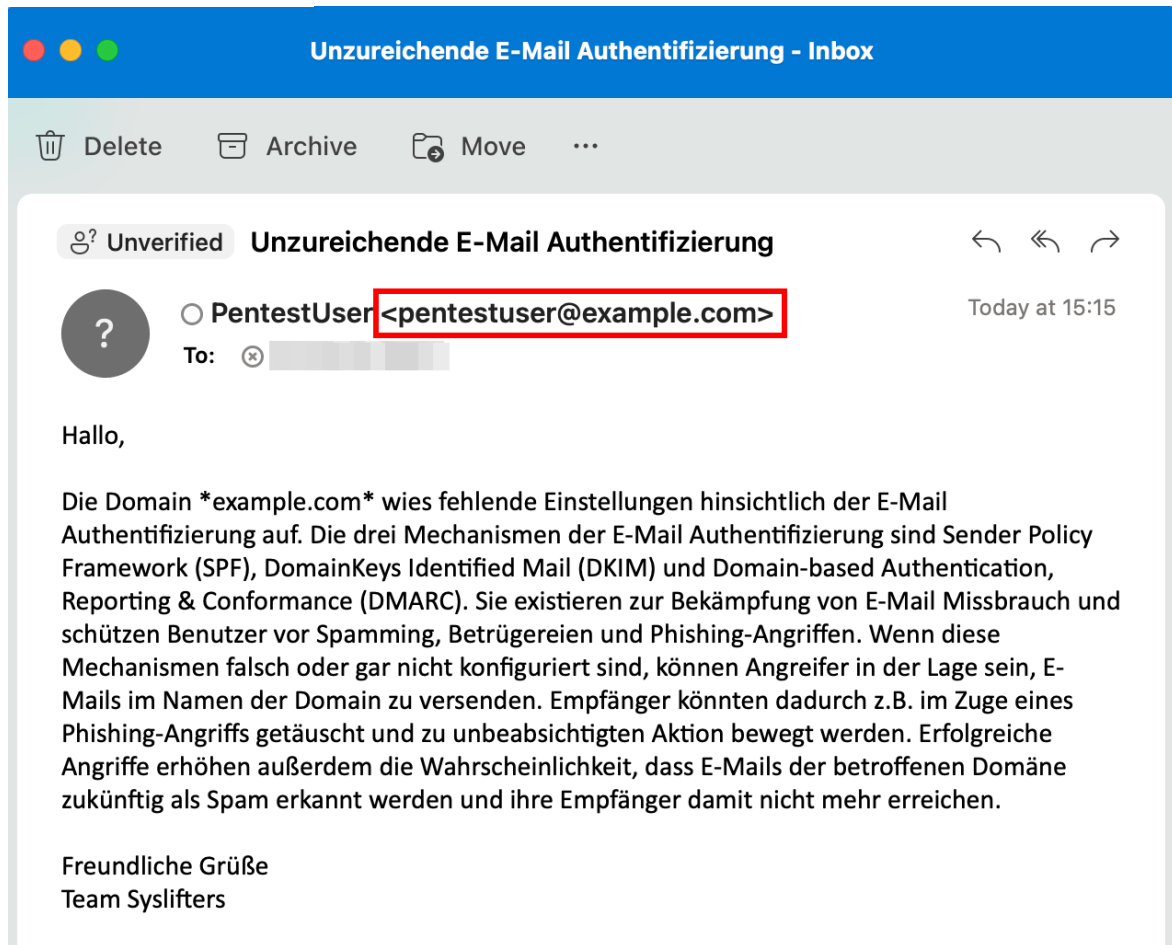


Abbildung 7 - Fehlende E-Mail Einstellungen erlauben Domain-Spoofing

Als E-Mail-Absender ist es wichtig, Best Practices hinsichtlich E-Mail Sicherheit zu befolgen und auf bewährte Verfahren zu setzen, um die Reputation der Domäne zu schützen. Die Mechanismen Sender Policy Framework (SPF), DomainKeys Identified Mail (DKIM) und Domain-based Authentication, Reporting & Conformance (DMARC) sind drei wichtige Authentifizierungsmethoden, um E-Mails ordnungsgemäß zu authentifizieren und Missbrauch durch Spamming oder Phishing zu bekämpfen. Diese drei Mechanismen in Kombination stellen sicher, dass eine E-Mail nicht gefälscht und der Absender wirklich befugt ist, E-Mails von einer bestimmten Domäne zu versenden.

SPF ist eine Authentifizierungsmethode, die IP-Adressen in einem TXT-Record im DNS definiert. Ausschließlich diese IP-Adressen sind berechtigt, E-Mails im Namen einer bestimmten Domäne zu versenden. Bei SPF prüft der empfangende Mailserver, ob ein SPF-Record für die Domäne des Absenders im DNS veröffentlicht wurde. Sofern ein SPF-Record existiert und die IP-Adresse des Absenders in diesem Eintrag enthalten ist, wird die E-Mail problemlos zugestellt. Wenn die IP-Adresse allerdings nicht im SPF-Record enthalten ist, wird die E-Mail entweder verworfen oder im Spam-Ordner zugestellt.

DKIM ist eine weitere Authentifizierungsmethode, die auf digitale Signaturen beruht, um damit E-Mail-Spoofing effektiv zu erkennen. Mit DKIM kann ein Empfänger überprüfen, ob eine E-Mail tatsächlich vom Eigentümer einer bestimmten Domäne stammt. Vor Versand einer E-Mail wird dazu eine digitale Signatur an die Nachricht angehängt, die ein Empfänger mit dem öffentlichen Schlüssel des Absenders überprüfen kann. Der öffentliche Schlüssel ist im

DNS publiziert und kann damit von jedem jederzeit abgerufen werden. Eine gültige Signatur garantiert, dass die E-Mail seit dem Anbringen der Signatur nicht verändert wurde.

Der dritte Mechanismus hinsichtlich E-Mail Sicherheit ist DMARC. DMARC hilft bei Bekämpfung von Spoofing und Phishing-Angriffen, indem es die unbefugte Verwendung einer Domäne im From-Header von E-Mails verhindert. Mittels entsprechender Richtlinien stellt DMARC sicher, dass E-Mails auf Grundlage von DKIM und SPF ordnungsgemäß authentifiziert und betrügerische Aktivitäten blockiert. DMARC-Richtlinien werden im DNS veröffentlicht und steuern, wie mit nicht authentifizierten E-Mails umgegangen werden soll. Eine DMARC-Richtlinie kann für eine von drei Aktionen konfiguriert sein. Nicht authentifizierte E-Mails werden beim Empfänger entweder ganz normal zugestellt, in den Spam-Ordner verschoben oder verworfen. DMARC bietet zudem eine Berichtsfunktion, mit denen Domäneneigentümer nachvollziehen können, von wo aus E-Mails mit ihrer Domäne als Absenderadresse gesendet werden.

Empfehlung

- Konfiguriert für alle Domänen, von denen E-Mails versendet werden, SPF, DKIM und DMARC. Authentifizierte E-Mails vereinfachen es, die Reputation des Absenders zu wahren und zu überwachen.
- Jede IP-Adresse, von der aus E-Mails versendet werden, sollte grundsätzlich über einen PTR-Record im DNS verfügen. Ein PTR-Record ermöglicht die Auflösung einer IP-Adresse zu einem Hostnamen.
- Haltet SPF-Einträge so einfach wie möglich und definiert nicht mehr Hosts als nötig in SPF-Einträgen. Achtet auch darauf, dass bei Includes niemals die Grenze von 10 Lookups überschritten werden.
- Definiert kleine Adressblöcke wie /24 oder /30, sofern in SPF-Einträgen Adressblöcke mit der CIDR-Notation angegeben werden.
- Achtet bei DKIM darauf, dass Schlüssel mindestens 3.072 Bit lang sind. Signaturen, die mit Schlüsseln von weniger als 1.024 Bit erstellt wurden, werden häufig ignoriert.
- Rotiert regelmäßig DKIM-Schlüssel (z.B. einmal im Jahr).
- Sofern ein E-Mail Dienst angeboten wird, vergewissert euch, dass für jeden Kunden ein eigener DKIM-Schlüssel verwendet wird.
- Signiert auch etwaige Bounce-Nachrichten mit DKIM.
- Wir empfehlen DMARC zu nutzen. Damit erhält ihr Auskunft über betrügerische E-Mails, die eure Domain verwenden. Diese können mittels DMARC identifiziert und blockiert werden, was zudem die Reputation eurer Domäne verbessert.
- Bei Verwendung von DMARC, vergewissert euch, dass eure Nachrichten eine "Identifier Alignment" aufweisen. Dadurch ist sichergestellt, dass mindestens eine der durch SPF oder DKIM authentifizierten Domänen mit der in der From-Header-Adresse angegebenen Domäne übereinstimmt.

Änderungsverzeichnis

Version	Datum	Beschreibung	Autor
0.1	2022-09-09	Initiale Erstellung	Christoph Mahrl
0.9	2022-09-11	Review	Aron Molnar
1.0	2022-09-11	Finale Version	Christoph Mahrl

Disclaimer

Wir können nicht garantieren, dass alle vorhandenen Schwachstellen und Sicherheitsrisiken tatsächlich entdeckt wurden. Das ist den beschränkten Zeitressourcen und dem limitierten Wissen der Pentester über die IT-Infrastruktur, Software, Source-Code, Benutzer etc. geschuldet. Eine umfassende Zusammenarbeit zwischen Auftraggeber und Penetration Testern erhöht die Effizienz des Penetration Tests. Das umfasst zum Beispiel die Offenlegung von Details interner Systeme oder die Provisionierung von Test-Benutzern.

Dieser Penetration Test stellt eine Momentaufnahme zum Zeitpunkt der Prüfung dar. Es lassen sich keine zukünftigen Sicherheitsrisiken davon ableiten.

Impressum

Syslifters GmbH | Eitzersthal 75 | 2013 Göllersdorf
FN 578505 v | Bezirksgericht Hollabrunn
syslifters.com | **Dedicated to Pentests**

