Christoph Mahrl | christoph@syslifters.com | +43 660 923 40 70

Pentest Report

Sample Report: External Infrastructure

Pentest for **Security Maximale GmbH**
September 11, 2022

Version 1.0

Inhalt

Vulnerability Overview

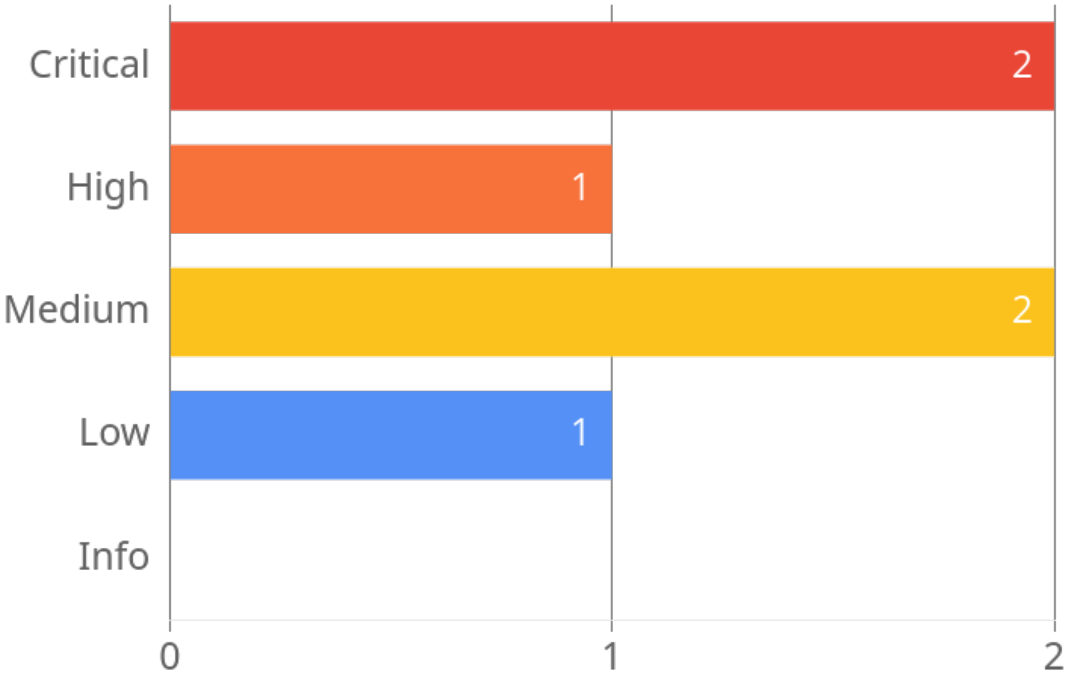


Figure 1 - Distribution of identified vulnerabilities

Finding	Severity	Remediation Status
Confluence Unauthenticated Remote Code Execution	**Critical**	**Offen**
WordPress Default Login	**Critical**	**Behoben**
Accessible .git directory	**High**	**Offen**
Subdomain Takeover	**Medium**	**Offen**
Credentials in password dumps	**Medium**	**Akzeptiert**
Insufficient email authentication	**Low**	**Geändert**

Management Summary

In the course of the security assessment, we were able to take over two infrastructure servers. This allowed us to access sensitive internal project documentation and to view and edit customer data.

This was achieved in two ways: On the one hand, a Confluence server instance was outdated and vulnerable to a remote code execution vulnerability. On the other hand, a WordPress backend could be accessed with the default password of the administrator account. Remote code execution could also be obtained on the WordPress instance by installing a plugin.

In addition, a .git directory on a web server was exposed and thus publicly accessible via the internet. The source code of the application could thus be reconstructed and examined for sensitive content (e.g. credentials, API keys, etc.) and vulnerabilities.

Other, less critical vulnerabilities, such as a subdomain takeover, the occurrence of credentials in public data leaks, and insufficient email authentication should be addressed in a continuous improvement process.

Here is the report. What now?

In this assessment we have identified vulnerabilities with criticality **Critical** and **High**. We recommend that these vulnerabilities be addressed as a matter of priority.

Vulnerabilities with less complex countermeasures and risk **Medium** and below should, according to our recommendation, be fixed prioritised by effort. All other vulnerabilities should either be fixed or addressed as part of a continious improvement process.

Please ensure that you deprovision all users and resources that were provisioned during the pentest as soon as they are no longer required.

Scope and duration

The scope of the pentest included the following publicly reachable systems:

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

The penetration test was conducted using a time-box approach and covered 5 person days.

Short description

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

Affects: confluence.example.com
The installed Confluence Server instance is vulnerable to a remote code execution vulnerability (CVE-2022-26134) that can be exploited without authentication. Due to an Object-Graph Navigation Language (OGNL) injection vulnerability in the web server, arbitrary system commands can be executed remotely. This allows an attacker to completely take over the underlying server.

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

Affects: www.example.com/wp-login.php
The WordPress website had set the default password of the administrator account (`admin`), which allowed access to the backend with highest privileges. An attacker could upload a webshell by installing a WordPress plugin and thereby completely take over the underlying web server.

3. Accessible .git directory (High: 7.5 | Offen)

Affects: www.example.com/.git
The web server revealed the *.git* directory of the web application. The source code of the application can thus be reconstructed and is publicly accessible via the internet. Attackers could gain access to sensitive information such as access data, hard-coded API keys or developer comments.

4. Subdomain Takeover (Medium: 5.8 | Offen)

Affects: git.example.com
The subdomain *git.example.com* was vulnerable to subdomain takover. A CNAME record was stored in the DNS, which referenced an orphaned GitHub page. An attacker could thus gain control over the subdomain and possibly have access to protected information (e.g. cookies) or the possibility to inject malicious scripts.

5. Credentials in password dumps (Medium: 5.3 | Akzeptiert)

Affects: example.com
During the pentest, numerous access data (email addresses, passwords, hashes, etc.) of employees could be found in data password dumps. Password dumps are extensive lists with often millions of captured username-password combinations that are made dataly available on the Internet by thieves after a data theft. Attackers can take advantage of this information and possibly gain access to affected user accounts and services (e.g. VPN, Azure AD, etc.).

6. Insufficient email authentication (Low: 3.7 | Geändert)

Affects: example.com
The domain `example.com` had missing settings regarding email authentication. The three mechanisms of email authentication are Sender Policy Framework (SPF), DomainKeys Identified Mail (DKIM) and Domain-based Authentication, Reporting & Conformance (DMARC). They exist to combat email abuse and protect users from spamming, scams and phishing attacks. If these mechanisms are configured incorrectly or not at all, attackers may be able to send emails on behalf of the domain. Recipients could thus be deceived and induced to take unintended action in the course of a phishing attack, for example. Successful attacks also increase the probability that e-mails from the affected domain will be recognized as spam in the future and thus no longer reach their recipients.

Vulnerability details

1. Confluence Unauthenticated Remote Code Execution

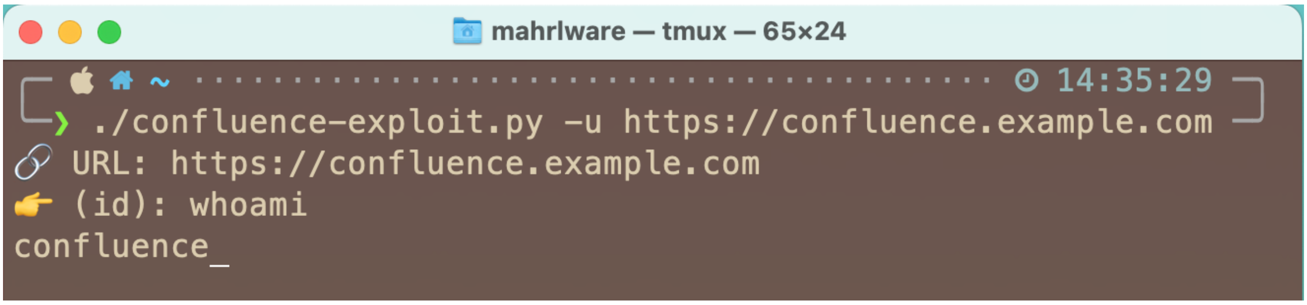
Remediation Status: **Offen**
Criticality: **Critical**
CVSS-Score: **10.0** | CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H
Affects: confluence.example.com

Overview

The installed Confluence Server instance is vulnerable to a remote code execution vulnerability (CVE-2022-26134) that can be exploited without authentication. Due to an Object-Graph Navigation Language (OGNL) injection vulnerability in the web server, arbitrary system commands can be executed remotely. This allows an attacker to completely take over the underlying server.

Description

In the course of the audit, we were able to identify an outdated Confluence Server version under confluence.example.com.



```
╭─  ~ ·································· ⏱ 14:35:29 ─╮
╰─❯ ./confluence-exploit.py -u https://confluence.example.com ─╯
🔗 URL: https://confluence.example.com
👉 (id): whoami
confluence_
```

Figure 2 - Confluence Remote Code Execution

Confluence is a commercial wiki software developed by the company Atlassian and used as an enterprise wiki for the documentation and communication of knowledge as well as knowledge sharing in companies and organisations. On 2 June 2022, Atlassian published a Security Advisory for Confluence Server, which pointed out a critical vulnerability with the CVE number CVE-2022-26134. The vulnerability allowed unauthenticated users to execute arbitrary code on a Confluence Server instance.

Recommendation

- The patch for *CVE-2022-26134* was released by Attliasian on 3 June, 2022. Further information on the vulnerability can be found in the security advisory from Atlassian, which is linked in the references.
- Systems and software should be kept up to date through regular updates, e.g. as part of a patch management process.

- Critical vulnerabilities and high-risk vulnerabilities should be prioritised and updated as soon as possible (e.g. within 2 weeks).
- Depending on the use case, we also recommend the use of automatic update mechanisms.

2. WordPress Default Login

Remediation Status: **Behoben**
Criticality: **Critical**
CVSS-Score: **9.8** | CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
Affects: www.example.com/wp-login.php

Overview

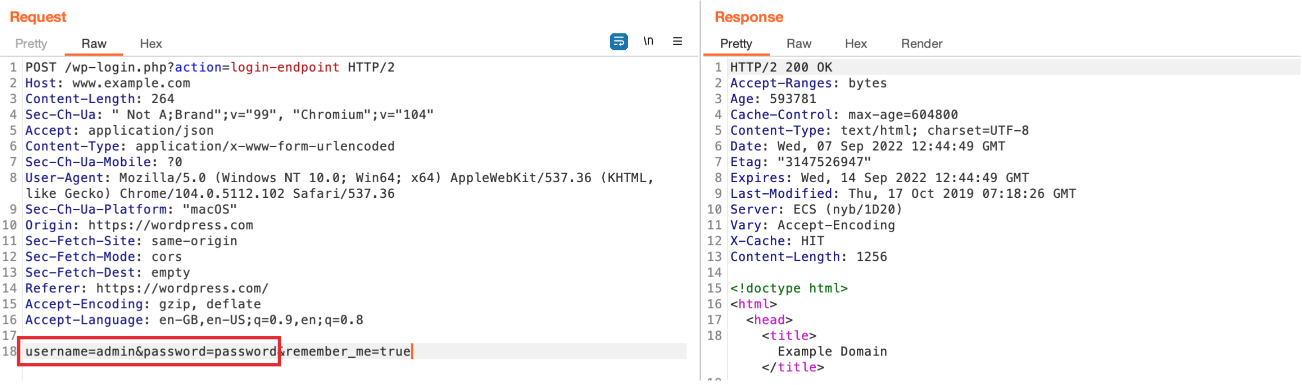
The WordPress website had set the default password of the administrator account (`admin`), which allowed access to the backend with highest privileges. An attacker could upload a webshell by installing a WordPress plugin and thereby completely take over the underlying web server.

Remediation status

The password was changed while the pentest was still running.

Description

The website at *www.example.com* was hosted in the WordPress content management system. The backend login was accessible at www.example.com/wp-login.php. In the course of the audit, the backend could be accessed with the administrator account `admin` with the default password.



```
POST /wp-login.php?action=login-endpoint HTTP/2
Host: www.example.com
Content-Length: 264
Sec-Ch-Ua: " Not A;Brand";v="99", "Chromium";v="104"
Accept: application/json
Content-Type: application/x-www-form-urlencoded
Sec-Ch-Ua-Mobile: ?0
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
Sec-Ch-Ua-Platform: "macOS"
Origin: https://wordpress.com
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://wordpress.com/
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8

username=admin&password=password&remember_me=true
```

```
HTTP/2 200 OK
Accept-Ranges: bytes
Age: 593781
Cache-Control: max-age=604800
Content-Type: text/html; charset=UTF-8
Date: Wed, 07 Sep 2022 12:44:49 GMT
Etag: "3147526947"
Expires: Wed, 14 Sep 2022 12:44:49 GMT
Last-Modified: Thu, 17 Oct 2019 07:18:26 GMT
Server: ECS (nyb/1D20)
Vary: Accept-Encoding
X-Cache: HIT
Content-Length: 1256

<!doctype html>
<html>
  <head>
    <title>
      Example Domain
    </title>
```

Figure 3 - WordPress Default Login

WordPress is a widely used content management system used for creating websites and blogs. Unless otherwise specified during installation, `password` is used by default as the initial password for the administrator account `admin`.

Recommendation

- The password of the administrator account `admin` should be changed immediately.
- Enforce a strong password policy. A strong password is defined by the following characteristics:
  - It should be at least 8 characters long.

  - It should consist of upper and lower case letters, numbers and special characters.
  - It should not be a frequently used password (e.g. sequence of numbers, sequence of letters, dictionary entry, etc).
- Allow passwords to use all characters, including Unicode and spaces. Ensure that there are no rules that dictate the composition of passwords and/or restrict the type of characters allowed.
- Do not set the maximum password length too low. Users would be discouraged from using passphrases. Also make sure they never abbreviate passwords when storing them.
- Consider additional authentication controls such as two-factor authentication.
- For detailed information and guidance on how to implement a secure authentication mechanism, see OWASP's linked Authentication Cheat Sheet.

3. Accessible .git directory

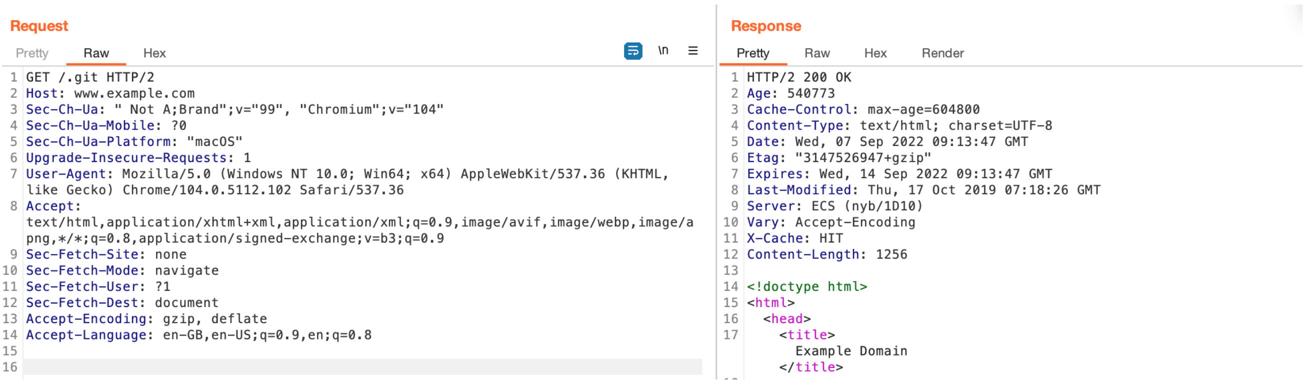
Remediation Status: Offen
Criticality: High
CVSS-Score: 7.5 | CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N
Affects: www.example.com/.git

Overview

The web server revealed the *.git* directory of the web application. The source code of the application can thus be reconstructed and is publicly accessible via the internet. Attackers could gain access to sensitive information such as access data, hard-coded API keys or developer comments.

Description

In the course of the audit, a disclosed .git directory was found for *www.example.com*.



```
GET /.git HTTP/2
Host: www.example.com
Sec-Ch-Ua: " Not A;Brand";v="99", "Chromium";v="104"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "macOS"
Upgrade-Insecure-Requests: 1
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
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/a
png,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8

```

```
HTTP/2 200 OK
Age: 540773
Cache-Control: max-age=604800
Content-Type: text/html; charset=UTF-8
Date: Wed, 07 Sep 2022 09:13:47 GMT
Etag: "3147526947+gzip"
Expires: Wed, 14 Sep 2022 09:13:47 GMT
Last-Modified: Thu, 17 Oct 2019 07:18:26 GMT
Server: ECS (nyb/1D10)
Vary: Accept-Encoding
X-Cache: HIT
Content-Length: 1256

<!doctype html>
<html>
  <head>
    <title>
      Example Domain
    </title>
```

Figure 4 - Accessible .git directory

Source code can be made public through a .git directory accessible via the web server. When using Git for version control, a .git folder is created by default in the root directory of the project, which stores all the project's information, including the commit history of the project files. The .git folder should not be accessible to the public, but this sometimes happens accidentally due to faulty web server configuration.

Accessing the .git directory can make the project's source code publicly available. An attacker could eventually search the source code for credentials, encryption keys, API tokens and developer comments. This also makes it easier to find potential vulnerabilities for possible follow-up attacks.

Recommendation

- To protect the source code of the application, the .git directory should not be publicly accessible.
- Git metadata should therefore be removed from the web server's webroot directory or access to it should be restricted.

4. Subdomain Takeover

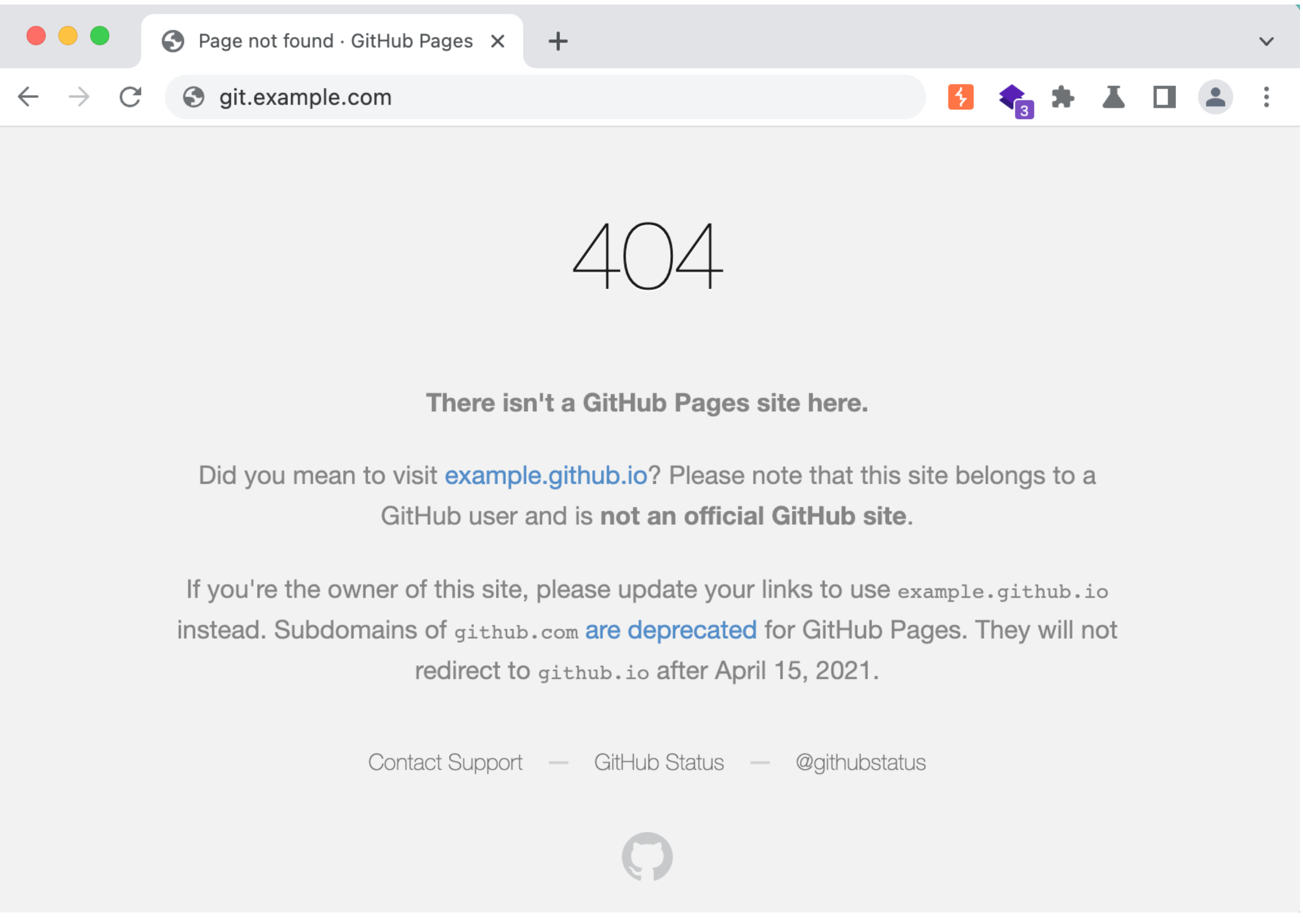
Remediation Status: **Offen**
Criticality: **Medium**
CVSS-Score: **5.8** | CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:N/I:L/A:N
Affects: git.example.com

Overview

The subdomain *git.example.com* was vulnerable to subdomain takover. A CNAME record was stored in the DNS, which referenced an orphaned GitHub page. An attacker could thus gain control over the subdomain and possibly have access to protected information (e.g. cookies) or the possibility to inject malicious scripts.

Description

In the course of the audit, we discovered the possibility of a subdomain takover for the subdomain `git.example.com`. A CNAME record existed in the DNS for the subdomain, which referred to a GitHub page that did not (no longer) exist.


Page not found · GitHub Pages
git.example.com

404

There isn't a GitHub Pages site here.

Did you mean to visit example.github.io? Please note that this site belongs to a GitHub user and is **not an official GitHub site**.

If you're the owner of this site, please update your links to use `example.github.io` instead. Subdomains of `github.com` are deprecated for GitHub Pages. They will not redirect to `github.io` after April 15, 2021.

Contact Support — GitHub Status — @githubstatus

Figure 5 - Subdomain Takover of `git.example.com`

A subdomain takover occurs when an attacker gains control over a subdomain of a target domain. This is the case if the subdomain has a CNAME in the DNS that refers to a service (e.g.

GitHub, AWS/S3, etc.) that has been removed or deleted. An attacker can take over this subdomain by providing their own virtual host and then hosting their own content for it.

For example, if `subdomain.example.com` pointed to a GitHub page and the user decided to delete their GitHub page, an attacker can now create a GitHub page and thereby claim `subdomain.example.com` as their own.

If an attacker succeeds in a subdomain takover, he may be able to read cookies from the main domain, perform cross-site scripting or bypass a content security policy (CSP) that may have been set. This would give him access to protected information (including logins) or the possibility to send malicious content to users.

Recommendation

- Remove the CNAME record for the subdomain `git.example.com` from the DNS.
- Furthermore, it is recommended to create an inventory of all domains and the associated hosting providers. The inventory should be maintained and updated when changes occur.
- To do this, define a standard process for provisioning and deprovisioning hosts to ensure that virtual hosts and DNS records are created or deleted in the correct order. When deprovisioning, we recommend starting with the removal of DNS records.

5. Credentials in password dumps

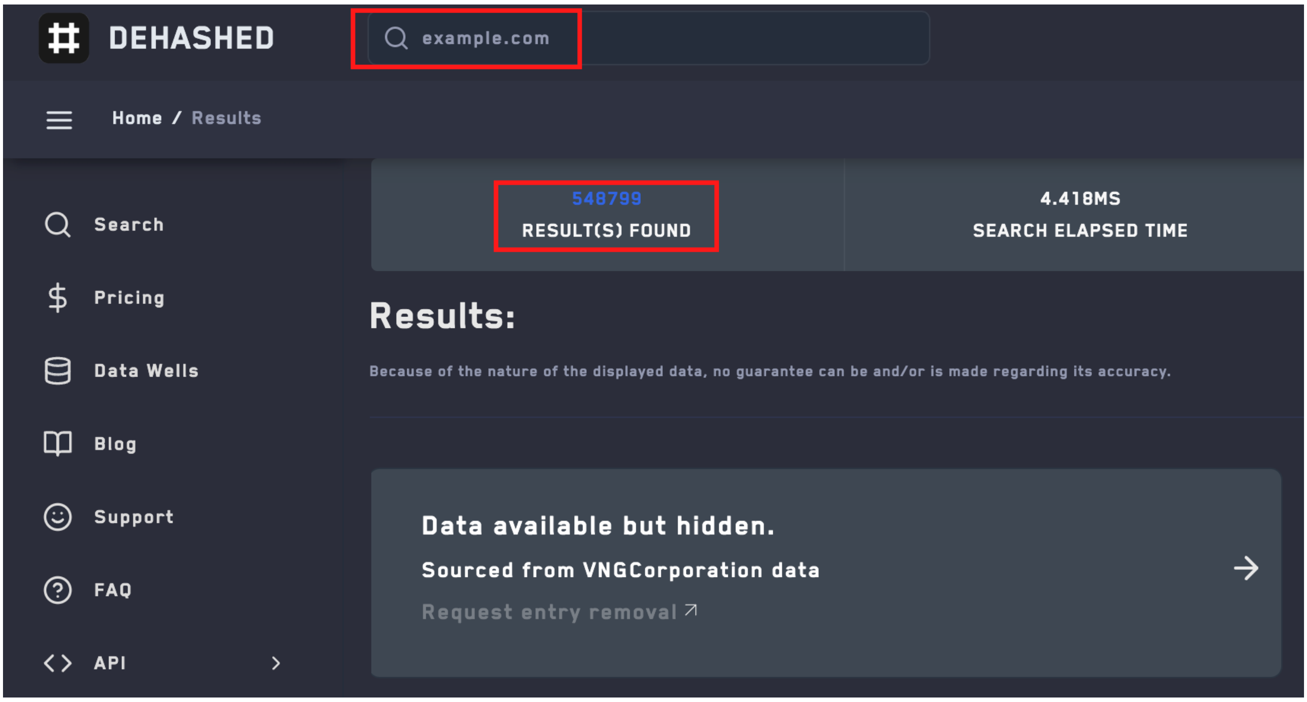
Remediation Status: **Akzeptiert**
Criticality: **Medium**
CVSS-Score: **5.3** | CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N
Affects: example.com

Overview

During the pentest, numerous access data (email addresses, passwords, hashes, etc.) of employees could be found in data password dumps. Password dumps are extensive lists with often millions of captured username-password combinations that are made dataly available on the Internet by thieves after a data theft. Attackers can take advantage of this information and possibly gain access to affected user accounts and services (e.g. VPN, Azure AD, etc.).

Description

Using the online service Dehashed, we searched for access data for the domain **example.com**. The search yielded 548.799 different entries (including duplicates). Numerous email addresses, plain text passwords, password hashes as well as personal data in various dumps could be identified.



548799
RESULT(S) FOUND

4.418MS
SEARCH ELAPSED TIME

Results:

Because of the nature of the displayed data, no guarantee can be and/or is made regarding its accuracy.

Data available but hidden.
Sourced from VNGCorporation data
Request entry removal

Figure 6 - Credentials in public password dumps

The identified credentials are attached to the report.

These credentials could be used by attackers to carry out password spraying and credential stuffing attacks. If employees use the same passwords for internal company services and multi-factor authentication is not enforced, these attacks could be successful.

Recommendation

- The passwords of affected user accounts should be reset if necessary.
- To reduce the risk of ending up in a password dump, business email addresses should not be used for private purposes. Employees should be adequately informed in this regard.
- We recommend continuous monitoring of the domain, e.g. with the help of services such as Kaduu.

6. Insufficient email authentication

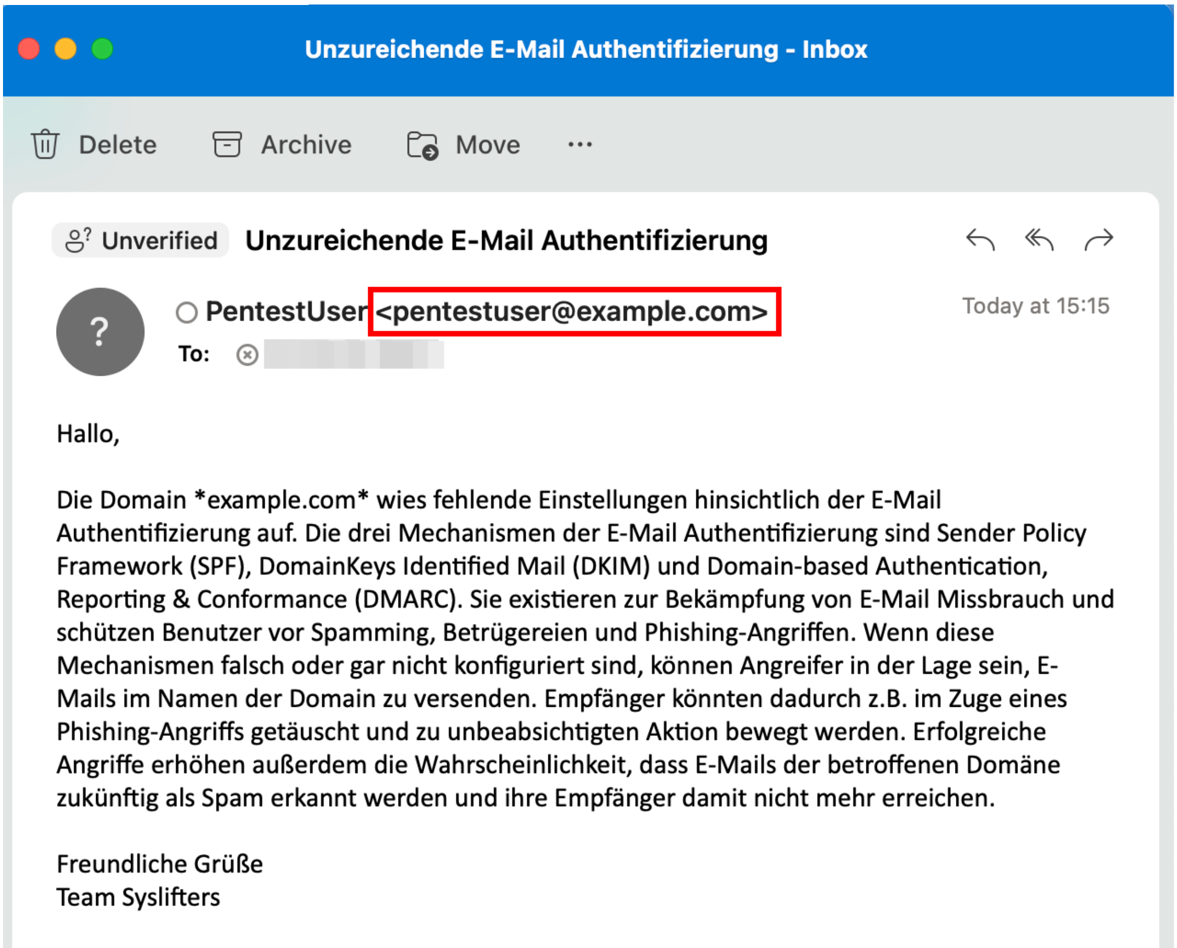
Remediation Status: **Geändert**
Criticality: **Low**
CVSS-Score: **3.7** | CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:N/A:L
Affects: example.com

Overview

The domain `example.com` had missing settings regarding email authentication. The three mechanisms of email authentication are Sender Policy Framework (SPF), DomainKeys Identified Mail (DKIM) and Domain-based Authentication, Reporting & Conformance (DMARC). They exist to combat email abuse and protect users from spamming, scams and phishing attacks. If these mechanisms are configured incorrectly or not at all, attackers may be able to send emails on behalf of the domain. Recipients could thus be deceived and induced to take unintended action in the course of a phishing attack, for example. Successful attacks also increase the probability that e-mails from the affected domain will be recognized as spam in the future and thus no longer reach their recipients.

Description

In the course of the audit, we discovered missing or incorrect settings regarding e-mail authentication in the domain *example.com*. This circumstance could be exploited to send phishing emails on behalf of these domains.



Unzureichende E-Mail Authentifizierung

PentestUser <pentestuser@example.com>
To: [redacted]
Today at 15:15

Hallo,

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). Sie existieren zur Bekämpfung von E-Mail Missbrauch und schützen Benutzer vor Spamming, Betrügereien und Phishing-Angriffen. Wenn diese Mechanismen falsch oder gar nicht 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.

Freundliche Grüße
Team Syslifters

Figure 7 - Insufficient email authentication allows domain spoofing

As an email sender, it is important to follow best practices regarding email security and rely on best practices to protect the domain's reputation. The Sender Policy Framework (SPF), DomainKeys Identified Mail (DKIM), and Domain-based Authentication, Reporting & Conformance (DMARC) mechanisms are three important authentication methods to properly authenticate email and combat abuse through spamming or phishing. These three mechanisms in combination ensure that an email is not spoofed and that the sender is truly authorized to send email from a particular domain.

SPF is an authentication method that defines IP addresses in a TXT record in DNS. Only these IP addresses are authorized to send emails on behalf of a specific domain. With SPF, the receiving mail server checks whether an SPF record for the sender's domain has been published in the DNS. If an SPF record exists and the IP address of the sender is included in this record, the e-mail will be delivered without any problems. However, if the IP address is not included in the SPF record, the email is either discarded or delivered to the spam folder.

DKIM is another authentication method that relies on digital signatures to effectively detect email spoofing. With DKIM, a recipient can verify that an email actually originates from the owner of a particular domain. Before an email is sent, a digital signature is attached to the message, which a recipient can verify using the sender's data key. The data key is published in the DNS and can therefore be retrieved by anyone at any time. A valid signature guarantees that the e-mail has not been modified since the signature was applied.

The third mechanism regarding e-mail security is DMARC. DMARC helps combat spoofing and phishing attacks by preventing the unauthorized use of a domain in the From header of e-mails. Using appropriate policies, DMARC ensures that email is properly authenticated based on DKIM and SPF and blocks fraudulent activity. DMARC policies are published in DNS and control how unauthenticated email is handled. A DMARC policy can be configured for one of three actions. Unauthenticated email is either delivered to the recipient normally, moved to the spam folder, or discarded. DMARC also provides a reporting capability that allows domain owners to track where email is being sent from with their domain as the sender address.

Recommendation

- Configure SPF, DKIM, and DMARC for all domains from which email is sent. Authenticated emails make it easier to maintain and monitor the reputation of the sender.
- Each IP address from which email is sent should always have a PTR record in DNS. A PTR record allows an IP address to resolve to a hostname.
- Keep SPF records as simple as possible and do not define more hosts than necessary in SPF records. Also make sure that includes never exceed the limit of 10 lookups.
- Define small address blocks such as /24 or /30 if address blocks with CIDR notation are specified in SPF entries.
- For DKIM, ensure that keys are at least 1,024 bits long. Signatures created with keys less than 1,024 bits are often ignored.
- Rotate DKIM keys regularly (e.g., once a year).
- If an email service is offered, make sure that a separate DKIM key is used for each customer.
- Also sign any bounce messages with DKIM.
- We recommend to use DMARC. This will give you information about fraudulent emails using your domain. These can be identified and blocked using DMARC, which also improves your domain's reputation.
- When using DMARC, make sure your messages have an "identifier alignment". This ensures that at least one of the domains authenticated by SPF or DKIM matches the domain specified in the From header address.

List of Changes

Version	Date	Description	Author
0.1	2022-09-09	Draft	Christoph Mahrl
0.9	2022-09-11	Review	Aron Molnar
1.0	2022-09-11	Final Version	Christoph Mahrl

Disclaimer

We cannot guarantee that all existing vulnerabilities and security risks have actually been discovered. This is due to limited time resources and limited knowledge of the pentester about the IT infrastructure, software, source code, users, etc. Extensive collaboration between the client and penetration testers increases the efficiency of the penetration test. This includes, for example, the disclosure of details of internal systems or the provisioning of test users.

This penetration test represents a snapshot at the time of testing. No future security risks can be derived from it.

Imprint

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