securitum

Security report

SUBJECT

Penetration testing of network infrastructure (WAN)

DATE

2024-02-12 – 2024-02-22

LOCATION

Gdansk, Polska

AUTHOR

Maciej Szymczak

VERSION

1.0

securitum

Executive summary

This report is a summary of the security tests conducted by Securitum. The subject of the tests was the external infrastructure (WAN) of the company [redacted], accessible at the following IP addresses:

- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]

- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]

The network device and server tests were conducted using a blackbox approach, that is, without having additional permissions.

The most significant vulnerabilities found include:

- Denial of Service vulnerabilities in WWW services

During the tests, special emphasis was placed on vulnerabilities that have or may have a negative impact on the confidentiality, integrity, and availability of the processed data.

The security tests were conducted in accordance with Securitum's internal security testing methodologies.

The work involved an approach that included manual testing, which was supported by a range of automated tools, including Burp Suite Professional, Feroxbuster, Nessus Professional, nmap.

The vulnerabilities have been described in detail in the further part of the report.

Risk classification

Vulnerabilities are classified on a five-point scale, that reflects both the probability of exploitation of the vulnerability and the business risk of its exploitation. Below, there is a short description of the meaning of each of the severity levels:

- CRITICAL – exploitation of the vulnerability makes it possible to compromise the server or network device or makes it possible to access (in read and/or write mode) data with a high degree of confidentiality and significance. The exploitation is usually straightforward, i.e. an attacker does not need to gain access to the systems that are difficult to reach and does not need to perform social engineering. Vulnerabilities marked as 'CRITICAL' must be fixed without delay, mainly if they occur in the production environment.

- HIGH – exploitation of the vulnerability makes it possible to access sensitive data (similar to the 'CRITICAL' level), however the prerequisites for the attack (e.g. possession of a user account in an internal system) make it slightly less likely. Alternatively, the vulnerability is easy to exploit, but the effects are somehow limited.

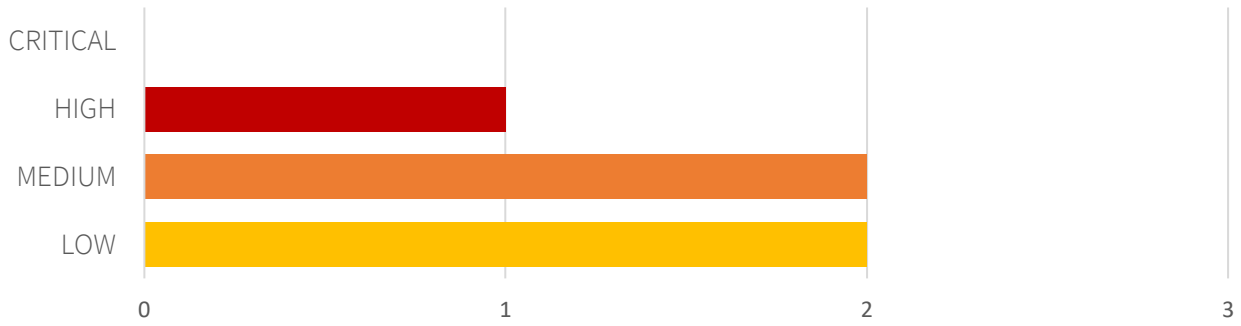
- MEDIUM – exploitation of the vulnerability might depend on external factors (e.g. convincing the user to click on a hyperlink) or other conditions that are difficult to achieve. Furthermore, exploitation of the vulnerability usually allows access only to a limited set of data or to data of a lesser degree of significance.

- LOW – exploitation of the vulnerability results in minor direct impact on the security of the test subject or depends on conditions that are very difficult to achieve in practical manner (e.g. physical access to the server).

- INFO – issues marked as 'INFO' are not security vulnerabilities per se. They aim to point out good practices, the implementation of which will lead to the overall increase of the system security level. Alternatively, the issues point out some solutions in the system (e.g. from an architectural perspective) that might limit the negative effects of other vulnerabilities.

Statistical overview

Below, a statistical summary of vulnerabilities is shown:



Additionally, two INFO issues are reported.

Contents

Change history

Document date	Version	Change description
2024-02-22	1.0	The initial version of the document.

Vulnerabilities in the infrastructure

[HIGH] SECURITUM-24887-001: WWW server vulnerable to *Denial-of-Service*

SUMMARY

During the audit, it was determined that the installed version of the Apache WWW server is 2.4.57. This version has known vulnerabilities, including:

- CVE-2023-45802,
- CVE-2023-43622,
- CVE-2023-31122,

which are described in more detail on the Apache page:

- https://httpd.apache.org/security/vulnerabilities_24.html

These vulnerabilities, although seemingly harmless, can affect the unavailability of services. The next section shows an example of such an attack, which is not only effective but also difficult to detect – no clear information suggesting that it is an attack is left in the server logs.

PREREQUISITES FOR THE ATTACK

There are none, the service is available from the Internet without any restrictions.

TECHNICAL DETAILS (PROOF OF CONCEPT)

Upon installation, the Apache server is by default vulnerable to DoS (Denial-of-Service) attacks. The policy of the Apache HTTP Server software developers is that the supplied server in default configuration, capable of serving content; however, it is the administrator's responsibility to further configure the service correctly, including protecting against attacks.

In the example below, not only the lack of proper server configuration, which is widely known as the Slowloris vulnerability (CVE-2007-6750), was exploited, but it was also combined with the vulnerability CVE-2023-43622.

The attack concerns the module responsible for handling the HTTP/2 protocol in Apache. Researchers discovered that sending many requests with incorrectly set initial HTTP/2 frame size causes the WWW server to behave similarly to a Slowloris attack – i.e., these connections are not properly closed, which in turn leads to a rapid depletion of the pool of available sockets and thus the unavailability of the WWW service.

Due to the lack of a publicly available exploit for vulnerability CVE-2023-43622, only a fragment of the tool and the course of the attack are presented below. The tested server was a container running the official image of Apache HTTP Server version 2.4.57 (`httpd:2.4.57-alpine`), available at https://127.0.0.1:8443/.

To send modified HTTP/2 communication, one can use the Scapy tool. A fragment of the defined HTTP/2 communication is presented below.

```
H2Frame()/H2SettingsFrame()/H2Setting(id=H2Setting.SETTINGS_INITIAL_WINDOW_SIZE, value=0)
```

Sending many requests while maintaining the connections leads to rapid server unavailability, as shown in the example below:

```
$ python3 CVE-2023-43622_poc.py
.................................................. 50 connections…
..................................................100 connections…
..................................................150 connections…
..................................................200 connections…
..................................................250 connections…
..................................................300 connections…
..................................................350 connections…
..................................................400 connections…
..................................................450 connections…
..................................................500 connections…
..................................................550 connections…
..................................................600 connections…
..................................................650 connections…
..................................................700 connections…
..................................................750 connections…
..................................................800 connections…
..................................................850 connections…
..................................................900 connections…
............^C
Killed. 912 connections were used.
```

After 912 connections, the server stopped accepting new ones, which was confirmed by a parallel call with the `curl` tool (the `-k` parameter was given to accept the *self-signed* certificate):

```
# curl -k https://127.0.0.1:8443/
curl: (28) SSL connection timeout
```

Given the production nature of the tested infrastructure, all tests and the above simulation were conducted in a controlled environment.

LOCATION

Affected resources:

- [redacted] (tcp/80, 443/tcp)
- [redacted] (tcp/80, 443/tcp)
- [redacted] (tcp/80, tcp/443)
- [redacted] (tcp/80, tcp/443)
- [redacted] (tcp/80, tcp/443)

RECOMMENDATION

It is recommended to update the software to the latest stable version. Additionally, protection against other Denial-of-Service attacks, which are detailed in the official documentation, should be implemented:

- https://httpd.apache.org/docs/trunk/misc/security_tips.html#dos

It is also worth considering migrating from the Apache server to, for example, nginx or Caddy, which in their basic configuration are much better adapted to handle excessive network traffic.

[MEDIUM] SECURITUM-24887-002: TLS certificates issues

SUMMARY

During the audit, numerous services were identified that provide services using the TLS protocol. While the concept of traffic encryption is valid, the use of untrusted or invalid certificates negates its purpose.

Clients using the services are forced to accept so-called “exceptions” in browsers (or other clients), which means they are not able to verify the validity of the certificate and therefore blindly accept what is displayed on the screen. The inability to verify the correctness of the certificate may lead to potential security breaches. By performing a successful Man-in-the-Middle attack, an attacker could deliberately exploit such a company policy (i.e., working on untrusted certificates) to persuade employees to accept the exception again, thereby gaining access to confidential data.

Read more:

- https://pl.wikipedia.org/wiki/Atak_man_in_the_middle

PREREQUISITES FOR THE ATTACK

A successful Man-in-the-Middle attack is necessary.

TECHNICAL DETAILS (PROOF OF CONCEPT)

To verify the details regarding TLS handling and the correctness of the used certificates, the tools `testssl.sh` and `openssl` were used. The findings from the tool's operation are presented below:

[redacted] (tcp/443) – expired certificate:

```
|-Subject : O=Digital Signature Trust Co./CN=DST Root CA X3
|-Not After : [redacted] 2021 GMT
```

[redacted] (tcp/443) – self-signed certificate:

```
|-Subject : O=Acme Co/CN=Kubernetes Ingress Controller Fake Certificate
|-Issuer : O=Acme Co/CN=Kubernetes Ingress Controller Fake Certificate
```

[redacted] (tcp/443) – self-signed certificate:

```
|-Subject : O=Acme Co/CN=Kubernetes Ingress Controller Fake Certificate
|-Issuer : O=Acme Co/CN=Kubernetes Ingress Controller Fake Certificate
```

[redacted] (tcp/21) – expired certificate:

```
|-Subject : O=Digital Signature Trust Co./CN=DST Root CA X3
|-Not After : [redacted] 2021 GMT
```

[redacted] (tcp/443) – expired certificate:

```
|-Subject : O=Digital Signature Trust Co./CN=DST Root CA X3
|-Not After : [redacted] 2021 GMT
```

[redacted] (tcp/443) – expired CA certificate:

```
|-Subject : O=Digital Signature Trust Co./CN=DST Root CA X3
|-Not After : [redacted] 2021 GMT
```

[redacted] (tcp/443) – expired CA certificate:

```
|-Subject : O=Digital Signature Trust Co./CN=DST Root CA X3
|-Not After : [redacted]2021 GMT
```

[redacted] (tcp/443) – incorrect certificate chain order:

```
|-Subject : CN=docker.[redacted].eu
|-Issuer : C=US/O=Let's Encrypt/CN=R3
```

[redacted] (tcp/443) – incomplete certificate chain:

```
|-Subject : CN=servicedesk.[redacted].eu
|-Issuer : C=US/O=Let's Encrypt/CN=R3
```

[redacted] (tcp/443) – expired certificates:

```
|-Subject : O=Digital Signature Trust Co./CN=DST Root CA X3
|-Not After : [redacted] 2021 GMT
```

and

```
|-Subject : CN=*.kube.[redacted].pl
|-Not After : [redacted] 2024 GMT
```

It is also worth mentioning the incorrect configuration – i.e., the server supports the outdated and unsafe TLS v1.0 protocol.

[redacted] (tcp/8895) – self-signed certificate:

```
|-Subject : C=PL/ST=[redacted]/O=[redacted]/CN=[redacted]
|-Issuer : C=PL/ST=[redacted]/O=[redacted]/CN=[redacted]
```

[redacted] (tcp/443) – expired certificate:

```
|-Subject : O=Digital Signature Trust Co./CN=DST Root CA X3
|-Not After : [redacted] 2021 GMT
```

LOCATION

[redacted] (tcp/443)
[redacted] (tcp/443)
[redacted] (tcp/443, tcp/8895)
[redacted] (tcp/443)
[redacted] (tcp/443)
[redacted] (tcp/443)
[redacted] (tcp/443)
[redacted] (tcp/443)
[redacted] (tcp/21, tcp/443)
[redacted] (tcp/443)

RECOMMENDATION

It is recommended to use only trusted TLS certificates. Additionally, services (in this case – WWW and FTP servers) should be configured to support the TLS v1.3 protocol, with possible support also for TLS v1.2. Attention should also be paid to additional parameters, depending on the WWW server.

A good source of example configurations is the tool prepared by Mozilla:

- https://ssl-config.mozilla.org/

[MEDIUM] SECURITUM-24887-003: FTP Server vulnerable to *Denial-of-Service* attack

SUMMARY

During the test, the FTP server was identified, the software vsftpd version 3.0.3, which has a known vulnerability resulting, among other things, from incorrect configuration. Like the vulnerability described in SECURITUM-24887-001, this service can also be attacked in a similar manner, causing the unavailability of the service.

The software author has also corrected several issues related to TLS handling in the latest release, which can be read about in detail on the official vsftpd page:

- https://security.appspot.com/vsftpd.html

PREREQUISITES FOR THE ATTACK

None, the service is available from the Internet.

TECHNICAL DETAILS (PROOF OF CONCEPT)

The vulnerability was identified based on the software version, which can be read by connecting directly to the service, for example, using the `nc` command:

```
$ nc [redacted] 21
220 (vsFTPd 3.0.3)
```

LOCATION

[redacted] (tcp/21)

RECOMMENDATION

It is recommended to update to the latest stable version of vsftpd. Correct configuration is also necessary (including the connection limit per IP, etc.), which is described in the documentation:

- https://security.appspot.com/vsftpd/vsftpd_conf.html

Options to pay attention to include `connect_timeout`, `data_connection_timeout` and `accept_timeout`.

[LOW] SECURITUM-24887-004: Unsupported Jetty software (part of Sonatype Nexus Repository)

SUMMARY

During the audit, outdated Jetty software in version 9.4.51.v20230217 was identified. Based on the specific HTTP response, it was determined that the Jetty server is part of the Sonatype Nexus Repository software. The outdated component simultaneously indicates that the main software is also outdated – at least since November 2023.

The Jetty server version 9.4.x lost support in June 2022:

- https://github.com/jetty/jetty.project/issues/7958

That means since then, this component has not received updates or security patches. Lack of regular updates leads to exposure to known attacks. It is worth mentioning that many manufacturers (especially of closed-source software) often do not explicitly write about fixing vulnerabilities in a given release, which is why it is particularly important to maintain the latest stable version possible.

PREREQUISITES FOR THE ATTACK

There are none, the service is available from the Internet.

TECHNICAL DETAILS (PROOF OF CONCEPT)

The software version was determined based on the following HTTP response:

```
$ curl -ik https://[redacted]
HTTP/1.1 400 Bad Request
Date: Thu, 22 Feb 2024 11:03:44 GMT
Server: Jetty(9.4.51.v20230217)

[…]

<title>Error 400 Not a Docker request</title>
</head>
<body><h2>HTTP ERROR 400 Not a Docker request</h2>
<table>
<tr><th>URI:</th><td>/</td></tr>
<tr><th>STATUS:</th><td>400</td></tr>
<tr><th>MESSAGE:</th><td>Not a Docker request</td></tr>
<tr><th>SERVLET:</th><td>-</td></tr>
</table>
<hr/><a href="https://eclipse.org/jetty">Powered by Jetty:// 9.4.51.v20230217</a><hr/>
[…]
```

At the same time, the specific `HTTP ERROR 400 Not a Docker request` response allowed to determine that this is likely a Sonatype Nexus Repository system in a version lower than 3.62.0 (released in November 2023), as confirmed by the release note accompanying this version:

3.62.0	November 7, 2023	• New Cleanup Preview Experience for Pro Customers Using PostgreSQL
		• New Combined Helm Chart for AWS, Azure, or On-Premises High Availability Deployments
		• Azure HA Performance Data
		• Support Zip Improvements
		  ◦ Generate zip for all nodes
		  ◦ Download link persists
		• Expanded Audit Logging
		  ◦ Include records for "Clear Cache" and "Change (server) order" LDAP events.
		  ◦ Added logging for when you create, update, or delete a routing rule.
		• Upgraded Jetty from version 9.4.51.v20230217 to version 9.4.53.v20231009

LOCATION

[redacted] (tcp/443)

RECOMMENDATION

It is recommended to update the software to the latest stable and supported version.

[LOW] SECURITUM-24887-005: Outdated software and excessive HTTP headers

SUMMARY

The audit revealed that some HTTP services disclose their software version in response headers. The mere fact of presenting the software version is not a direct threat to the infrastructure as long as the software uses the latest version. In the tested cases, some services are not in the latest versions and have known vulnerabilities.

PREREQUISITES FOR THE ATTACK

Depending on the vulnerability, it is worth noting that all indicated services are available from the Internet.

TECHNICAL DETAILS (PROOF OF CONCEPT)

The following services reveal the type and version of software used.

[redacted] (tcp/80, tcp/443) – outdated version of Apache server:

```
Apache/2.4.56 (Debian)
```

[redacted] (tcp/80, tcp/443) – outdated version of Apache server:

```
Apache/2.4.57 (Debian)
```

[redacted] (tcp/80) – outdated version of Apache server:

```
Apache/2.4.57 (Debian)
```

[redacted] (tcp/443) – discussed in the SECURITUM-24887-004 point:

```
Jetty(9.4.51.v20230217)
```

[redacted] (tcp/443) – unsupported version of nginx server:

```
nginx/1.22.1
```

[redacted] (tcp/80):

```
Microsoft-IIS/10.0
```

[redacted] (tcp/443):

```
Microsoft-HTTPAPI/2.0
```

[redacted] (tcp/80, tcp/443):

```
Microsoft-HTTPAPI/2.0
```

[redacted] (tcp/80):

```
Apache/2.4.57 (Debian)
```

[redacted] (tcp/80, tcp/443) unsupported version of nginx server:

```
nginx/1.18.0
```

[redacted] (tcp/80/www) - outdated version of Apache server:

```
Apache/2.4.56 (Debian)
```

[redacted] (tcp/80, tcp/443) – unsupported version of nginx server:

```
nginx/1.22.1
```

LOCATION

IP addresses and port numbers have been listed in the Technical Details section.

RECOMMENDATION

It is recommended to perform regular system updates, along with installed services. Updates should be carried out in accordance with a developed update policy (schedule).

Informational issues

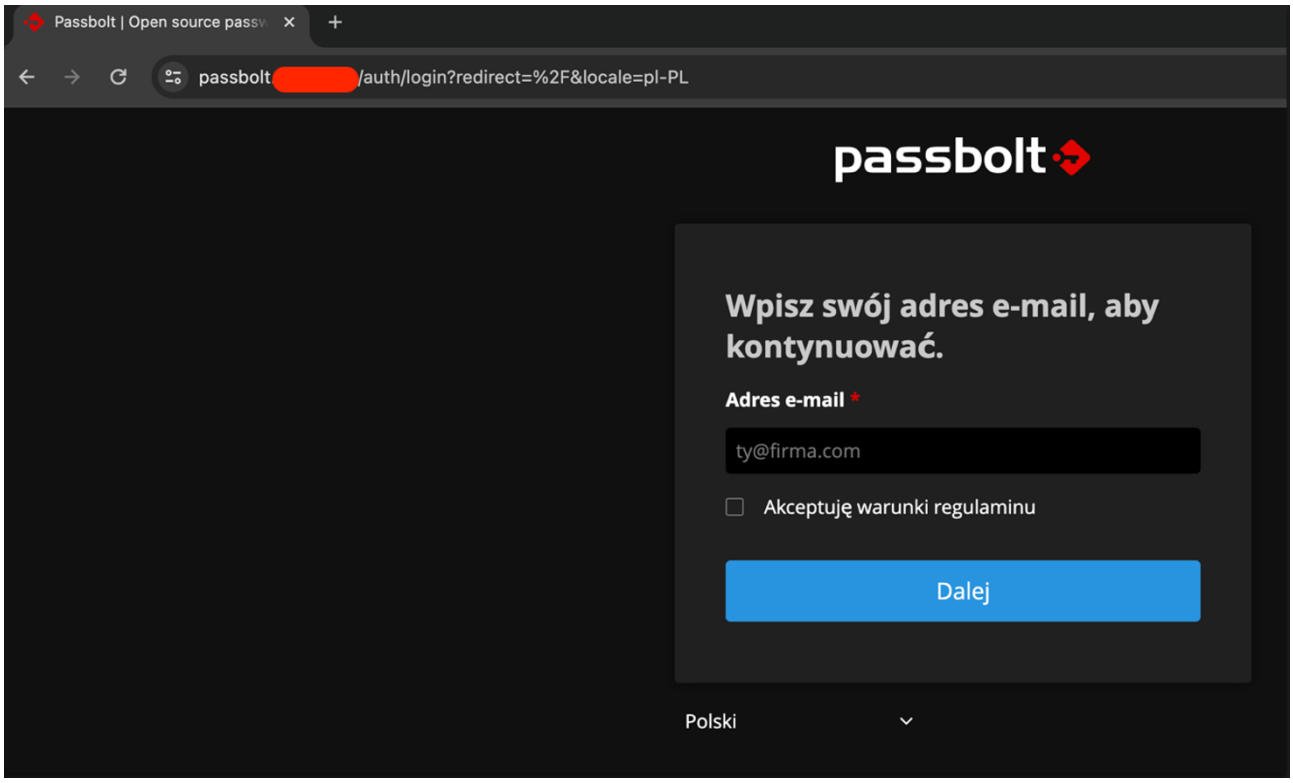
[INFO] SECURITUM-24887-006: Credential sharing service accessible from the Internet

SUMMARY

The Passbolt product is a system for sharing credentials. It is critical for the organization as the sensitive matter of data is processed. Such software should not be exposed directly to the Internet unless necessary.

TECHNICAL DETAILS (PROOF OF CONCEPT)

Below is a screenshot of the login panel that secures access to the password manager:



LOCATION

[redacted] (tcp/443, tcp/80)

RECOMMENDATION

It is recommended to hide the login panel behind additional security – e.g., mutual TLS (mTLS) or possibly a VPN.

[INFO] SECURITUM-24887-007: Default welcome pages of WWW servers

SUMMARY

Improperly configured HTTP services display default welcome pages, allowing a potential attacker to better understand the environment of their target.

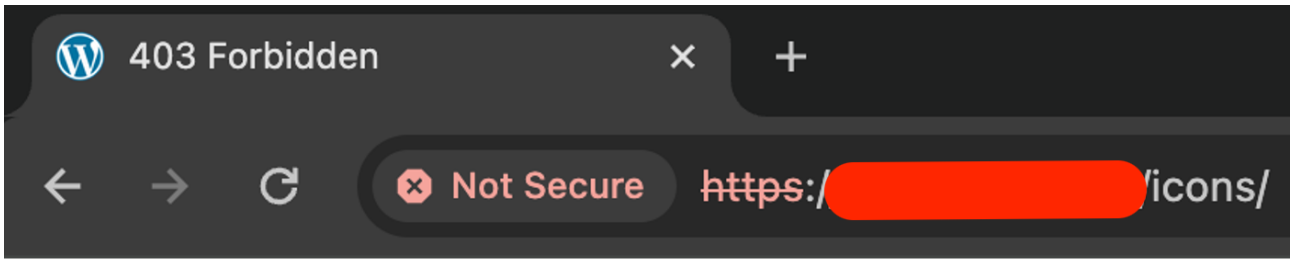
PREREQUISITES FOR THE ATTACK

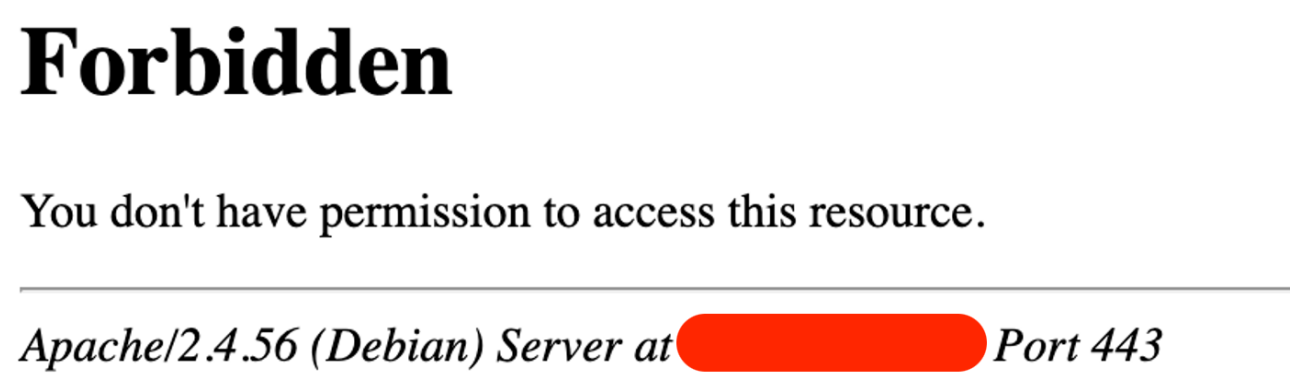
There are none, services are available from the Internet.

TECHNICAL DETAILS (PROOF OF CONCEPT)

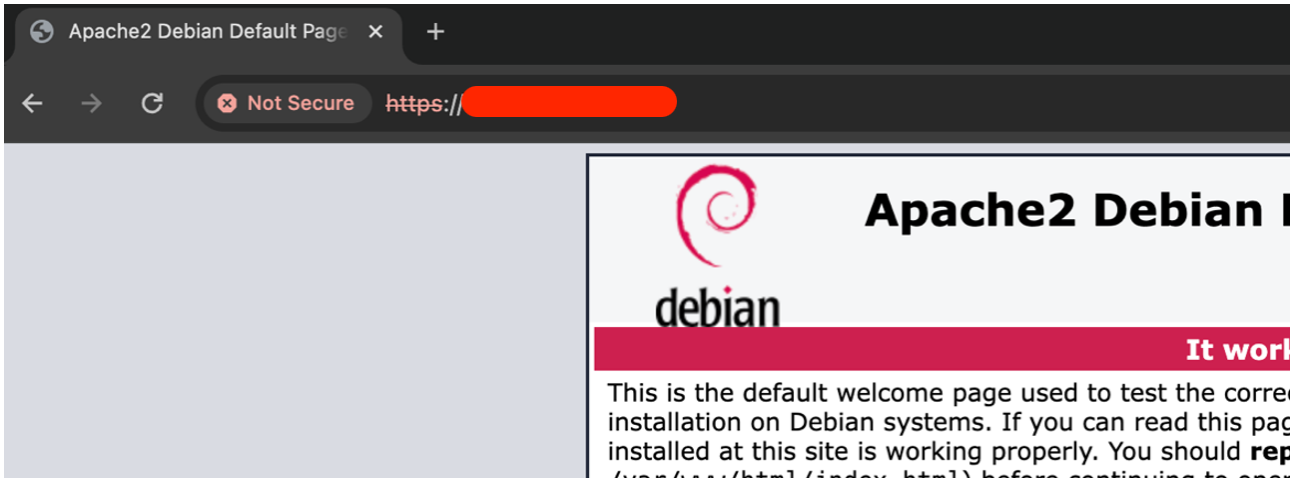
The screenshots below show the identified welcome pages of WWW servers.

[redacted] (tcp/443) - the default *vhost* of the Apache server was identified using the standard `/icons` directory – an `HTTP 403 Forbidden` error indicates that this directory is blocked but simultaneously confirms the existence of such a *vhost*:

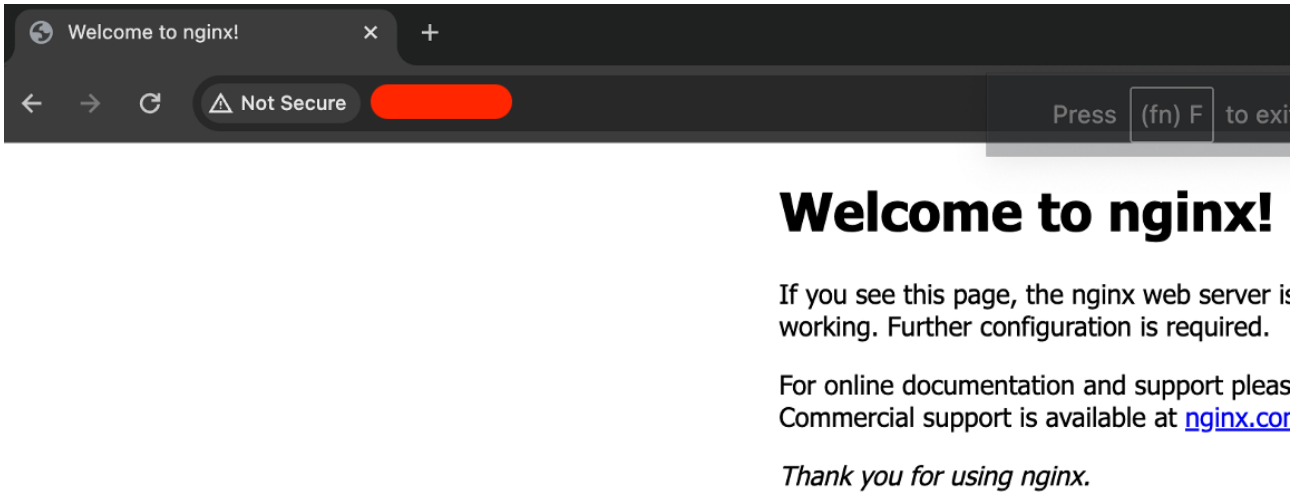




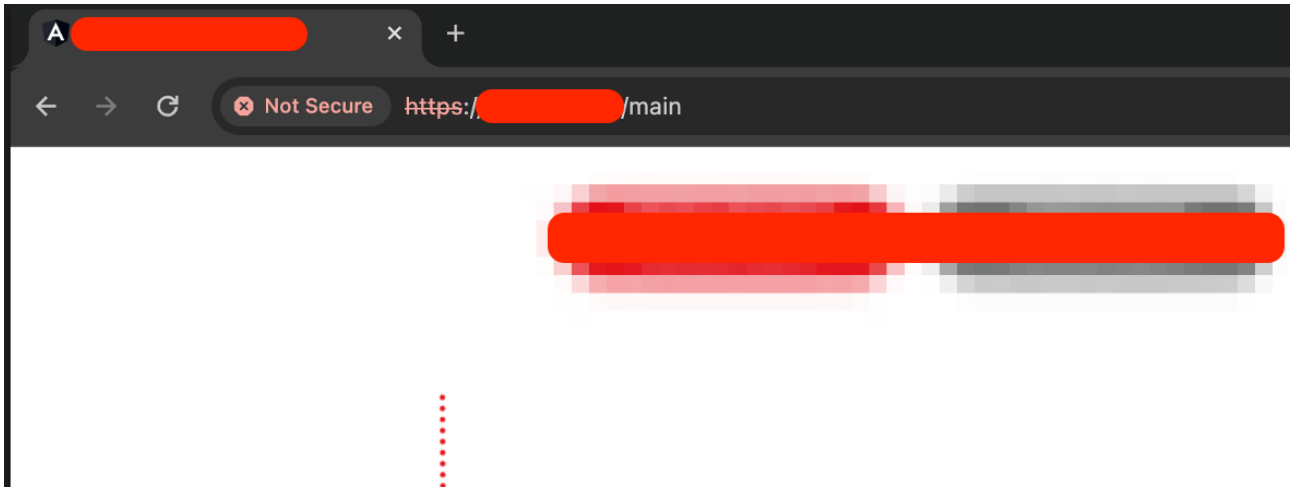
[redacted] (tcp/443) - when connecting directly to the IP address (without using a domain name), the server displays the standard Apache page:



[redacted] (tcp/80) - when connecting directly to the IP address (without using a domain name), the server displays the standard nginx page:



[redacted] (tcp/443) - when connecting directly to the IP address (without using a domain name), the server displays a WWW application:



LOCATION

- [redacted] (tcp/443)
- [redacted] (tcp/443)
- [redacted] (tcp/80)
- [redacted] (tcp/443)

RECOMMENDATION

It is recommended to configure HTTP servers in such a way that only domains (*vhosts*) under which a given service should be available are served. Direct queries (e.g., by IP or another domain) should be rejected (e.g., using the HTTP 444 error code – in nginx, this will cause an immediate connection closure).