Scan Report

July 1, 2024

Summary

This document reports on the results of an automatic security scan. All dates are displayed using the timezone "Coordinated Universal Time", which is abbreviated "UTC". The task was "Potato task". The scan started at Mon Jul 1 16:45:14 2024 UTC and ended at Mon Jul 1 17:01:03 2024 UTC. The report first summarises the results found. Then, for each host, the report describes every issue found. Please consider the advice given in each description, in order to rectify the issue.

Contents

1 Result Overview

Host	High	Medium	Low	Log	False Positive
10.0.2.11 potato	0	3	3	0	0
Total: 1	0	3	3	0	0

Vendor security updates are not trusted.
Overrides are off. Even when a result has an override, this report uses the actual threat of the result.
Information on overrides is included in the report.
Notes are included in the report.
This report might not show details of all issues that were found.
Issues with the threat level “Log” are not shown.
Issues with the threat level “Debug” are not shown.
Issues with the threat level “False Positive” are not shown.
Only results with a minimum QoD of 70 are shown.

This report contains all 6 results selected by the filtering described above. Before filtering there were 57 results.

2 Results per Host

2.1 10.0.2.11

Host scan start	Mon Jul 1 16:46:18 2024 UTC
Host scan end	Mon Jul 1 17:00:56 2024 UTC

Service (Port)	Threat Level
2112/tcp	Medium
80/tcp	Medium
general/icmp	Low
general/tcp	Low
22/tcp	Low

2.1.1 Medium 2112/tcp

Medium (CVSS: 6.4)

NVT: Anonymous FTP Login Reporting

Summary
Reports if the remote FTP Server allows anonymous logins.

. . . continues on next page . . .

. . . continued from previous page . . .

Quality of Detection: 80

Vulnerability Detection Result

```
It was possible to login to the remote FTP service with the following anonymous
↪account(s):
anonymous:anonymous@example.com
ftp:anonymous@example.com
Here are the contents of the remote FTP directory listing:
Account "anonymous":
-rw-r--r--   1 ftp      ftp           901 Aug  2  2020 index.php.bak
-rw-r--r--   1 ftp      ftp            54 Aug  2  2020 welcome.msg
Account "ftp":
-rw-r--r--   1 ftp      ftp           901 Aug  2  2020 index.php.bak
-rw-r--r--   1 ftp      ftp            54 Aug  2  2020 welcome.msg
```

Impact

Based on the files accessible via this anonymous FTP login and the permissions of this account an attacker might be able to:
- gain access to sensitive files
- upload or delete files.

Solution:
Solution type: Mitigation
If you do not want to share files, you should disable anonymous logins.

Vulnerability Insight

A host that provides an FTP service may additionally provide Anonymous FTP access as well. Under this arrangement, users do not strictly need an account on the host. Instead the user typically enters 'anonymous' or 'ftp' when prompted for username. Although users are commonly asked to send their email address as their password, little to no verification is actually performed on the supplied data.
Remark: NIST don't see 'configuration issues' as software flaws so the referenced CVE has a severity of 0.0. The severity of this VT has been raised by Greenbone to still report a configuration issue on the target.

Vulnerability Detection Method

Details: `Anonymous FTP Login Reporting`
OID:1.3.6.1.4.1.25623.1.0.900600
Version used: `2021-10-20T09:03:29Z`

References

`cve: CVE-1999-0497`

Medium (CVSS: 4.8)

NVT: FTP Unencrypted Cleartext Login

Summary
The remote host is running a FTP service that allows cleartext logins over unencrypted connections.

Quality of Detection: 70

Vulnerability Detection Result

```
The remote FTP service accepts logins without a previous sent 'AUTH TLS' command
↪. Response(s):
Non-anonymous sessions: 331 Password required for openvasvt
Anonymous sessions:     331 Anonymous login ok, send your complete email address
↪ as your password
```

Impact
An attacker can uncover login names and passwords by sniffing traffic to the FTP service.

Solution:
Solution type: Mitigation
Enable FTPS or enforce the connection via the 'AUTH TLS' command. Please see the manual of the FTP service for more information.

Vulnerability Detection Method
Tries to login to a non FTPS enabled FTP service without sending a 'AUTH TLS' command first and checks if the service is accepting the login without enforcing the use of the 'AUTH TLS' command.
Details: `FTP Unencrypted Cleartext Login`
OID:1.3.6.1.4.1.25623.1.0.108528
Version used: `2023-12-20T05:05:58Z`

[return to 10.0.2.11]

2.1.2 Medium 80/tcp

Medium (CVSS: 4.8)

NVT: Cleartext Transmission of Sensitive Information via HTTP

Summary
The host / application transmits sensitive information (username, passwords) in cleartext via HTTP.

. . . continues on next page . . .

. . . continued from previous page . . .

Quality of Detection: 80

Vulnerability Detection Result

```
The following input fields were identified (URL:input name):
http://potato/admin/:password
http://potato/admin/?D=A:password
```

Impact

An attacker could use this situation to compromise or eavesdrop on the HTTP communication between the client and the server using a man-in-the-middle attack to get access to sensitive data like usernames or passwords.

Solution:

Solution type: Workaround

Enforce the transmission of sensitive data via an encrypted SSL/TLS connection. Additionally make sure the host / application is redirecting all users to the secured SSL/TLS connection before allowing to input sensitive data into the mentioned functions.

Affected Software/OS

Hosts / applications which doesn't enforce the transmission of sensitive data via an encrypted SSL/TLS connection.

Vulnerability Detection Method

Evaluate previous collected information and check if the host / application is not enforcing the transmission of sensitive data via an encrypted SSL/TLS connection.

The script is currently checking the following:

- HTTP Basic Authentication (Basic Auth)
- HTTP Forms (e.g. Login) with input field of type 'password'

Details: `Cleartext Transmission of Sensitive Information via HTTP`

OID:1.3.6.1.4.1.25623.1.0.108440

Version used: `2023-09-07T05:05:21Z`

References

```
url: https://www.owasp.org/index.php/Top_10_2013-A2-Broken_Authentication_and_Se
↪ssion_Management
url: https://www.owasp.org/index.php/Top_10_2013-A6-Sensitive_Data_Exposure
url: https://cwe.mitre.org/data/definitions/319.html
```

[return to 10.0.2.11]

2.1.3 Low general/icmp

Low (CVSS: 2.1)

NVT: ICMP Timestamp Reply Information Disclosure

Summary
The remote host responded to an ICMP timestamp request.

Quality of Detection: 80

Vulnerability Detection Result

```
The following response / ICMP packet has been received:
- ICMP Type: 14
- ICMP Code: 0
```

Impact
This information could theoretically be used to exploit weak time-based random number generators in other services.

Solution:
Solution type: Mitigation
Various mitigations are possible:
- Disable the support for ICMP timestamp on the remote host completely
- Protect the remote host by a firewall, and block ICMP packets passing through the firewall in either direction (either completely or only for untrusted networks)

Vulnerability Insight
The Timestamp Reply is an ICMP message which replies to a Timestamp message. It consists of the originating timestamp sent by the sender of the Timestamp as well as a receive timestamp and a transmit timestamp.

Vulnerability Detection Method
Sends an ICMP Timestamp (Type 13) request and checks if a Timestamp Reply (Type 14) is received.
Details: `ICMP Timestamp Reply Information Disclosure`
OID:1.3.6.1.4.1.25623.1.0.103190
Version used: `2023-05-11T09:09:33Z`

References

```
cve: CVE-1999-0524
url: https://datatracker.ietf.org/doc/html/rfc792
url: https://datatracker.ietf.org/doc/html/rfc2780
cert-bund: CB-K15/1514
cert-bund: CB-K14/0632
dfn-cert: DFN-CERT-2014-0658
```

[return to 10.0.2.11]

2.1.4 Low general/tcp

Low (CVSS: 2.6)

NVT: TCP Timestamps Information Disclosure

Summary

The remote host implements TCP timestamps and therefore allows to compute the uptime.

Quality of Detection: 80

Vulnerability Detection Result

```
It was detected that the host implements RFC1323/RFC7323.
The following timestamps were retrieved with a delay of 1 seconds in-between:
Packet 1: 680131116
Packet 2: 680132176
```

Impact

A side effect of this feature is that the uptime of the remote host can sometimes be computed.

Solution:

Solution type: Mitigation

To disable TCP timestamps on linux add the line 'net.ipv4.tcp_timestamps = 0' to /etc/sysctl.conf. Execute 'sysctl -p' to apply the settings at runtime.

To disable TCP timestamps on Windows execute 'netsh int tcp set global timestamps=disabled'

Starting with Windows Server 2008 and Vista, the timestamp can not be completely disabled.

The default behavior of the TCP/IP stack on this Systems is to not use the Timestamp options when initiating TCP connections, but use them if the TCP peer that is initiating communication includes them in their synchronize (SYN) segment.

See the references for more information.

Affected Software/OS

TCP implementations that implement RFC1323/RFC7323.

Vulnerability Insight

The remote host implements TCP timestamps, as defined by RFC1323/RFC7323.

Vulnerability Detection Method

Special IP packets are forged and sent with a little delay in between to the target IP. The responses are searched for a timestamps. If found, the timestamps are reported.

Details: `TCP Timestamps Information Disclosure`

OID:1.3.6.1.4.1.25623.1.0.80091

Version used: `2023-12-15T16:10:08Z`

References

`url: https://datatracker.ietf.org/doc/html/rfc1323`

. . . continues on next page . . .

. . . continued from previous page . . .

```
url: https://datatracker.ietf.org/doc/html/rfc7323
url: https://web.archive.org/web/20151213072445/http://www.microsoft.com/en-us/d
↪ownload/details.aspx?id=9152
url: https://www.fortiguard.com/psirt/FG-IR-16-090
```

[return to 10.0.2.11]

2.1.5 Low 22/tcp

Low (CVSS: 2.6)

NVT: Weak MAC Algorithm(s) Supported (SSH)

Product detection result

```
cpe:/a:ietf:secure_shell_protocol
Detected by SSH Protocol Algorithms Supported (OID: 1.3.6.1.4.1.25623.1.0.105565
↪)
```

Summary

The remote SSH server is configured to allow / support weak MAC algorithm(s).

Quality of Detection: 80

Vulnerability Detection Result

```
The remote SSH server supports the following weak client-to-server MAC algorithm
↪(s):
umac-64-etm@openssh.com
umac-64@openssh.com
The remote SSH server supports the following weak server-to-client MAC algorithm
↪(s):
umac-64-etm@openssh.com
umac-64@openssh.com
```

Solution:

Solution type: Mitigation

Disable the reported weak MAC algorithm(s).

Vulnerability Detection Method

Checks the supported MAC algorithms (client-to-server and server-to-client) of the remote SSH server.

Currently weak MAC algorithms are defined as the following:

- MD5 based algorithms
- 96-bit based algorithms

. . . continues on next page . . .

. . . continued from previous page . . .

- 64-bit based algorithms
- 'none' algorithm

Details: `Weak MAC Algorithm(s) Supported (SSH)`
OID:1.3.6.1.4.1.25623.1.0.105610
Version used: `2024-06-14T05:05:48Z`

Product Detection Result
Product: `cpe:/a:ietf:secure_shell_protocol`
Method: `SSH Protocol Algorithms Supported`
OID: 1.3.6.1.4.1.25623.1.0.105565)

References
url: https://www.rfc-editor.org/rfc/rfc6668
url: https://www.rfc-editor.org/rfc/rfc4253#section-6.4

[return to 10.0.2.11]

This file was automatically generated.