Scan Report

March 2, 2025

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 “full network openvas scan - eaamir”. The scan started at Sun Mar 2 19:29:19 2025 UTC and ended at Sun Mar 2 20:09:46 2025 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
10.0.0.1	0	5	2	0	0
10.0.0.156	0	3	3	0	0
10.0.0.32 EHMAN-LAPTOP	0	1	2	0	0
10.0.0.71 LAPTOP-L9GD05AC	0	0	1	0	0
10.0.0.231	0	0	1	0	0
10.0.0.183	0	0	1	0	0
Total: 6	0	9	10	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 19 results selected by the filtering described above. Before filtering there were 221 results.

2 Results per Host

2.1 10.0.0.1

Host scan start	Sun Mar 2 09:28:32 2025 UTC
Host scan end	Sun Mar 2 19:07:43 2025 UTC

Service (Port)	Threat Level
1883/tcp	Medium
12865/tcp	Medium
80/tcp	Medium
443/tcp	Medium
53/udp	Medium
general/tcp	Low
general/icmp	Low

2.1.1 Medium 1883/tcp

Medium (CVSS: 6.4)

NVT: MQTT Broker Does Not Require Authentication

Summary
The remote MQTT broker does not require authentication.

Quality of Detection (QoD): 80%

Vulnerability Detection Result
Vulnerability was detected according to the Vulnerability Detection Method.

Solution:
Solution type: Mitigation
Enable authentication.

Vulnerability Detection Method
Checks if authentication is required for the remote MQTT broker.
Details: `MQTT Broker Does Not Require Authentication`
OID:1.3.6.1.4.1.25623.1.0.140167
Version used: `2022-07-11T10:16:03Z`

References

```
url: https://www.heise.de/newsticker/meldung/MQTT-Protokoll-IoT-Kommunikation-vo
↪n-Reaktoren-und-Gefaengnissen-oeffentlich-einsehbar-3629650.html
```

[return to 10.0.0.1]

2.1.2 Medium 12865/tcp

Medium (CVSS: 5.0)

NVT: Check for Writesrv Service

Summary
writesrv is running on this port, it is used to send messages to users.

Quality of Detection (QoD): 70%

Vulnerability Detection Result
Vulnerability was detected according to the Vulnerability Detection Method.

Impact

. . . continues on next page . . .

. . . continued from previous page . . .

This service gives potential attackers information about who is connected and who isn't, easing social engineering attacks for example.

Solution:
Solution type: Mitigation
Disable this service if you don't use it.

Vulnerability Detection Method
Details: `Check for Writesrv Service`
OID:1.3.6.1.4.1.25623.1.0.11222
Version used: `2023-08-01T13:29:10Z`

[return to 10.0.0.1]

2.1.3 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.

Quality of Detection (QoD): 80%

Vulnerability Detection Result

```
The following input fields were identified (URL:input name):
http://10.0.0.1/: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

. . . continues on next page . . .

. . . continued from previous page . . .

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 10.0.0.1]

2.1.4 Medium 443/tcp

Medium (CVSS: 4.0)

NVT: SSL/TLS: Diffie-Hellman Key Exchange Insufficient DH Group Strength Vulnerability

Summary
The SSL/TLS service uses Diffie-Hellman groups with insufficient strength (key size < 2048).

Quality of Detection (QoD): 80%

Vulnerability Detection Result
```
Server Temporary Key Size: 1024 bits
```

Impact
An attacker might be able to decrypt the SSL/TLS communication offline.

Solution:
Solution type: Workaround
Deploy (Ephemeral) Elliptic-Curve Diffie-Hellman (ECDHE) or use a 2048-bit or stronger Diffie-Hellman group (see the references).
For Apache Web Servers: Beginning with version 2.4.7, mod_ssl will use DH parameters which include primes with lengths of more than 1024 bits.

. . . continues on next page . . .

. . . continued from previous page . . .

Vulnerability Insight
The Diffie-Hellman group are some big numbers that are used as base for the DH computations. They can be, and often are, fixed. The security of the final secret depends on the size of these parameters. It was found that 512 and 768 bits to be weak, 1024 bits to be breakable by really powerful attackers like governments.

Vulnerability Detection Method
Checks the DHE temporary public key size.
Details: `SSL/TLS: Diffie-Hellman Key Exchange Insufficient DH Group Strength Vulnerabili.`
`↪..`
OID:1.3.6.1.4.1.25623.1.0.106223
Version used: `2024-09-30T08:38:05Z`

References

```
url: https://weakdh.org/
url: https://weakdh.org/sysadmin.html
```

[return to 10.0.0.1]

2.1.5 Medium 53/udp

Medium (CVSS: 5.0)

NVT: DNS Cache Snooping Vulnerability (UDP) - Active Check

Summary
The DNS server is prone to a cache snooping vulnerability.

Quality of Detection (QoD): 70%

Vulnerability Detection Result

```
Received (an) answer(s) for a non-recursive query for "example.com".
Result:
23.215.0.138
```

Impact
Attackers might gain information about cached DNS records which might lead to further attacks.
Note: This finding might be an acceptable risk if you:
- trust all clients which can reach the server
- do not allow recursive queries from outside your trusted client network.

Solution:
Solution type: Mitigation

. . . continues on next page . . .

. . . continued from previous page . . .

There are multiple possible mitigation steps depending on location and functionality needed by the DNS server:
- Disable recursion
- Don't allow public access to DNS Servers doing recursion
- Leave recursion enabled if the DNS Server stays on a corporate network that cannot be reached by untrusted clients

Vulnerability Insight

DNS cache snooping is when someone queries a DNS server in order to find out (snoop) if the DNS server has a specific DNS record cached, and thereby deduce if the DNS server's owner (or its users) have recently visited a specific site.
This may reveal information about the DNS server's owner, such as what vendor, bank, service provider, etc. they use. Especially if this is confirmed (snooped) multiple times over a period.
This method could even be used to gather statistical information - for example at what time does the DNS server's owner typically access his net bank etc. The cached DNS record's remaining TTL value can provide very accurate data for this.
DNS cache snooping is possible even if the DNS server is not configured to resolve recursively for 3rd parties, as long as it provides records from the cache also to 3rd parties (a.k.a. 'lame requests').

Vulnerability Detection Method

Sends a crafted DNS query and checks the response.
Details: `DNS Cache Snooping Vulnerability (UDP) - Active Check`
OID:1.3.6.1.4.1.25623.1.0.146591
Version used: `2025-01-21T05:37:33Z`

References

```
url: https://www.cs.unc.edu/~fabian/course_papers/cache_snooping.pdf
url: https://docs.microsoft.com/en-us/troubleshoot/windows-server/networking/dns
↪-server-cache-snooping-attacks
url: https://kb.isc.org/docs/aa-00509
url: https://kb.isc.org/docs/aa-00482
```

[return to 10.0.0.1]

2.1.6 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.

. . . continues on next page . . .

. . . continued from previous page . . .

Quality of Detection (QoD): 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: 934487104
Packet 2: 934488239
```

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
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 10.0.0.1]

2.1.7 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 (QoD): 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: `2025-01-21T05:37:33Z`

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 10.0.0.1]

2.2 10.0.0.156

Host scan start	Sun Mar 2 09:28:32 2025 UTC
Host scan end	Sun Mar 2 09:35:57 2025 UTC

Service (Port)	Threat Level
8762/tcp	Medium
general/tcp	Low
8762/tcp	Low
general/icmp	Low

2.2.1 Medium 8762/tcp

Medium (CVSS: 5.3)

NVT: Weak Host Key Algorithm(s) (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 host key algorithm(s).

Quality of Detection (QoD): 80%

Vulnerability Detection Result

```
The remote SSH server supports the following weak host key algorithm(s):
host key algorithm | Description
-----------------------------------------------------------------------------
↪---------
ssh-dss            | Digital Signature Algorithm (DSA) / Digital Signature Stand
↪ard (DSS)
```

Solution:

Solution type: Mitigation

Disable the reported weak host key algorithm(s).

Vulnerability Detection Method

Checks the supported host key algorithms of the remote SSH server.

Currently weak host key algorithms are defined as the following:

- ssh-dss: Digital Signature Algorithm (DSA) / Digital Signature Standard (DSS)

Details: `Weak Host Key Algorithm(s) (SSH)`

. . . continues on next page . . .

. . . continued from previous page . . .

OID:1.3.6.1.4.1.25623.1.0.117687
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/rfc8332
url: https://www.rfc-editor.org/rfc/rfc8709
url: https://www.rfc-editor.org/rfc/rfc4253#section-6.6
```

Medium (CVSS: 5.3)

NVT: Weak Key Exchange (KEX) 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 key exchange (KEX) algorithm(s).

Quality of Detection (QoD): 80%

Vulnerability Detection Result

```
The remote SSH server supports the following weak KEX algorithm(s):
KEX algorithm                 | Reason
-----------------------------------------------------------------------------
↪---
diffie-hellman-group1-sha1 | Using Oakley Group 2 (a 1024-bit MODP group) and SH
↪A-1
```

Impact
An attacker can quickly break individual connections.

Solution:
Solution type: Mitigation
Disable the reported weak KEX algorithm(s)
- 1024-bit MODP group / prime KEX algorithms:
Alternatively use elliptic-curve Diffie-Hellmann in general, e.g. Curve 25519.

. . . continues on next page . . .

. . . continued from previous page . . .

Vulnerability Insight

- 1024-bit MODP group / prime KEX algorithms:

Millions of HTTPS, SSH, and VPN servers all use the same prime numbers for Diffie-Hellman key exchange. Practitioners believed this was safe as long as new key exchange messages were generated for every connection. However, the first step in the number field sieve-the most efficient algorithm for breaking a Diffie-Hellman connection-is dependent only on this prime.
A nation-state can break a 1024-bit prime.

Vulnerability Detection Method

Checks the supported KEX algorithms of the remote SSH server.
Currently weak KEX algorithms are defined as the following:
- non-elliptic-curve Diffie-Hellmann (DH) KEX algorithms with 1024-bit MODP group / prime
- ephemerally generated key exchange groups uses SHA-1
- using RSA 1024-bit modulus key

Details: `Weak Key Exchange (KEX) Algorithm(s) Supported (SSH)`
OID:1.3.6.1.4.1.25623.1.0.150713
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://weakdh.org/sysadmin.html
url: https://www.rfc-editor.org/rfc/rfc9142
url: https://www.rfc-editor.org/rfc/rfc9142#name-summary-guidance-for-implem
url: https://www.rfc-editor.org/rfc/rfc6194
url: https://www.rfc-editor.org/rfc/rfc4253#section-6.5
```

Medium (CVSS: 4.3)

NVT: Weak Encryption 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 encryption algorithm(s).

. . . continues on next page . . .

... continued from previous page ...

Quality of Detection (QoD): 80%

Vulnerability Detection Result

```
The remote SSH server supports the following weak client-to-server encryption al
↪gorithm(s):
3des-cbc
aes128-cbc
aes256-cbc
twofish-cbc
twofish128-cbc
twofish256-cbc
The remote SSH server supports the following weak server-to-client encryption al
↪gorithm(s):
3des-cbc
aes128-cbc
aes256-cbc
twofish-cbc
twofish128-cbc
twofish256-cbc
```

Solution:
Solution type: Mitigation
Disable the reported weak encryption algorithm(s).

Vulnerability Insight

- The 'arcfour' cipher is the Arcfour stream cipher with 128-bit keys. The Arcfour cipher is believed to be compatible with the RC4 cipher [SCHNEIER]. Arcfour (and RC4) has problems with weak keys, and should not be used anymore.
- The 'none' algorithm specifies that no encryption is to be done. Note that this method provides no confidentiality protection, and it is NOT RECOMMENDED to use it.
- A vulnerability exists in SSH messages that employ CBC mode that may allow an attacker to recover plaintext from a block of ciphertext.

Vulnerability Detection Method

Checks the supported encryption algorithms (client-to-server and server-to-client) of the remote SSH server.
Currently weak encryption algorithms are defined as the following:
- Arcfour (RC4) cipher based algorithms
- 'none' algorithm
- CBC mode cipher based algorithms

Details: `Weak Encryption Algorithm(s) Supported (SSH)`
OID:1.3.6.1.4.1.25623.1.0.105611
Version used: `2024-06-14T05:05:48Z`

Product Detection Result

Product: `cpe:/a:ietf:secure_shell_protocol`

... continues on next page ...

. . . continued from previous page . . .

Method: `SSH Protocol Algorithms Supported`
OID: 1.3.6.1.4.1.25623.1.0.105565)

References

```
url: https://www.rfc-editor.org/rfc/rfc8758
url: https://www.kb.cert.org/vuls/id/958563
url: https://www.rfc-editor.org/rfc/rfc4253#section-6.3
```

[return to 10.0.0.156]

2.2.2 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 (QoD): 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: 322567083
Packet 2: 322568179
```

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.

. . . continues on next page . . .

. . . continued from previous page . . .

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
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 10.0.0.156]

2.2.3 Low 8762/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 (QoD): 80%

Vulnerability Detection Result

```
The remote SSH server supports the following weak client-to-server MAC algorithm
↪(s):
hmac-md5
hmac-sha1-96
The remote SSH server supports the following weak server-to-client MAC algorithm
↪(s):
hmac-md5
```

. . . continues on next page . . .

. . . continued from previous page . . .

```
hmac-sha1-96
```

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
- 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 10.0.0.156]

2.2.4 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 (QoD): 80%

Vulnerability Detection Result

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

. . . continues on next page . . .

. . . continued from previous page . . .

```
- 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: `2025-01-21T05:37:33Z`

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 10.0.0.156]

2.3 10.0.0.32

Host scan start	Sun Mar 2 09:28:32 2025 UTC
Host scan end	Sun Mar 2 10:01:07 2025 UTC

Service (Port)	Threat Level
135/tcp	Medium
general/icmp	Low
general/tcp	Low

2.3.1 Medium 135/tcp

Medium (CVSS: 5.0)

NVT: DCE/RPC and MSRPC Services Enumeration Reporting

Summary

Distributed Computing Environment / Remote Procedure Calls (DCE/RPC) or MSRPC services running on the remote host can be enumerated by connecting on port 135 and doing the appropriate queries.

Quality of Detection (QoD): 80%

Vulnerability Detection Result

```
Here is the list of DCE/RPC or MSRPC services running on this host via the TCP p
↪rotocol:
Port: 49664/tcp
     UUID: 12345778-1234-abcd-ef00-0123456789ac, version 1
     Endpoint: ncacn_ip_tcp:10.0.0.32[49664]
     Named pipe : lsass
     Win32 service or process : lsass.exe
     Description : SAM access
     UUID: 51a227ae-825b-41f2-b4a9-1ac9557a1018, version 1
     Endpoint: ncacn_ip_tcp:10.0.0.32[49664]
     Annotation: Ngc Pop Key Service
     UUID: 8fb74744-b2ff-4c00-be0d-9ef9a191fe1b, version 1
     Endpoint: ncacn_ip_tcp:10.0.0.32[49664]
     Annotation: Ngc Pop Key Service
     UUID: b25a52bf-e5dd-4f4a-aea6-8ca7272a0e86, version 2
     Endpoint: ncacn_ip_tcp:10.0.0.32[49664]
     Annotation: KeyIso
Port: 49665/tcp
     UUID: d95afe70-a6d5-4259-822e-2c84da1ddb0d, version 1
     Endpoint: ncacn_ip_tcp:10.0.0.32[49665]
Port: 49666/tcp
     UUID: 3a9ef155-691d-4449-8d05-09ad57031823, version 1
     Endpoint: ncacn_ip_tcp:10.0.0.32[49666]
     UUID: 86d35949-83c9-4044-b424-db363231fd0c, version 1
     Endpoint: ncacn_ip_tcp:10.0.0.32[49666]
Port: 49667/tcp
     UUID: f6beaff7-1e19-4fbb-9f8f-b89e2018337c, version 1
     Endpoint: ncacn_ip_tcp:10.0.0.32[49667]
     Annotation: Windows Event Log
Port: 49670/tcp
     UUID: 0b6edbfa-4a24-4fc6-8a23-942b1eca65d1, version 1
     Endpoint: ncacn_ip_tcp:10.0.0.32[49670]
     UUID: 12345678-1234-abcd-ef00-0123456789ab, version 1
```

. . . continues on next page . . .

. . . continued from previous page . . .

```
     Endpoint: ncacn_ip_tcp:10.0.0.32[49670]
     Named pipe : spoolss
     Win32 service or process : spoolsv.exe
     Description : Spooler service
     UUID: 4a452661-8290-4b36-8fbe-7f4093a94978, version 1
     Endpoint: ncacn_ip_tcp:10.0.0.32[49670]
     UUID: 76f03f96-cdfd-44fc-a22c-64950a001209, version 1
     Endpoint: ncacn_ip_tcp:10.0.0.32[49670]
     UUID: ae33069b-a2a8-46ee-a235-ddfd339be281, version 1
     Endpoint: ncacn_ip_tcp:10.0.0.32[49670]
Port: 49677/tcp
     UUID: 367abb81-9844-35f1-ad32-98f038001003, version 2
     Endpoint: ncacn_ip_tcp:10.0.0.32[49677]
Note: DCE/RPC or MSRPC services running on this host locally were identified. Re
↪porting this list is not enabled by default due to the possible large size of
↪this list. See the script preferences to enable this reporting.
```

Impact
An attacker may use this fact to gain more knowledge about the remote host.

Solution:
Solution type: Mitigation
Filter incoming traffic to this ports.

Vulnerability Detection Method
Details: DCE/RPC and MSRPC Services Enumeration Reporting
OID:1.3.6.1.4.1.25623.1.0.10736
Version used: 2022-06-03T10:17:07Z

[return to 10.0.0.32]

2.3.2 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 (QoD): 80%

Vulnerability Detection Result
The following response / ICMP packet has been received:

. . . continues on next page . . .

. . . continued from previous page . . .

```
- 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: `2025-01-21T05:37:33Z`

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 10.0.0.32]

2.3.3 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.

. . . continues on next page . . .

. . . continued from previous page . . .

Quality of Detection (QoD): 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: 10566036
Packet 2: 10567822
```

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
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 10.0.0.32]

2.4 10.0.0.71

Host scan start	Sun Mar 2 19:30:00 2025 UTC
Host scan end	Sun Mar 2 20:09:43 2025 UTC

Service (Port)	Threat Level
general/tcp	Low

2.4.1 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 (QoD): 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: 3810989113
Packet 2: 3810990401
```

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.

. . . continues on next page . . .

. . . continued from previous page . . .

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
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 10.0.0.71]

2.5 10.0.0.231

Host scan start Sun Mar 2 09:28:32 2025 UTC
Host scan end Sun Mar 2 19:22:01 2025 UTC

Service (Port)	Threat Level
general/tcp	Low

2.5.1 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 (QoD): 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: 1153502371
Packet 2: 59781217
```

Impact

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

. . . continues on next page . . .

. . . continued from previous page . . .

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
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 10.0.0.231]

2.6 10.0.0.183

Host scan start	Sun Mar 2 09:28:38 2025 UTC
Host scan end	Sun Mar 2 10:39:06 2025 UTC

Service (Port)	Threat Level
general/tcp	Low

2.6.1 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 (QoD): 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: 2428862679
Packet 2: 3036620607
```

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
url: https://datatracker.ietf.org/doc/html/rfc7323
url: https://web.archive.org/web/20151213072445/http://www.microsoft.com/en-us/d
```

. . . continues on next page . . .

. . . continued from previous page . . .

```
↪ownload/details.aspx?id=9152
url: https://www.fortiguard.com/psirt/FG-IR-16-090
```

[return to 10.0.0.183]

This file was automatically generated.