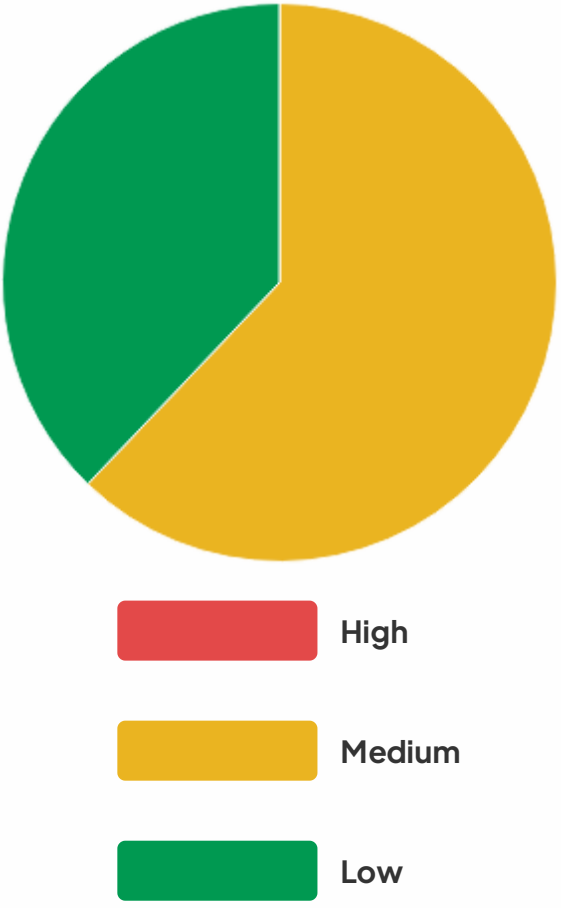
Cod Unic Înregistrare:
RO42445365-J05/632/2020
RO88 RZBR 0000 0600 2445
2606 Raiffeisen Bank
RO06 TREZ 0765 069X XX02 3710
Trezoreria
Municipiului Oradea

DeForce Tehnologic S.R.L.
0755 522 942
dani@deforce.eu
P-ța Em. Gojdu, nr. 41
(Cetatea Oradea, Corp E, Etaj 2)

DeForce Core Security Scan Report

2026-06-12 21:01



Top 3 Vulnerabilities By Severity

1. SSL/TLS: Deprecated SSLv2 and SSLv3 Protocol Detection
   CVSS: 5.90
2. SSL/TLS: Report Weak Cipher Suites
   CVSS: 5.90
3. Weak Key Exchange (KEX) Algorithm(s) Supported (SSH)
   CVSS: 5.30

Top 3 Most Common Vulnerabilities

1. TCP Timestamps Information Disclosure
   Found 6 times
   Max CVSS: 2.60
2. ICMP Timestamp Reply Information Disclosure
   Found 6 times
   Max CVSS: 2.10
3. SSL/TLS: Certificate Expired
   Found 4 times
   Max CVSS: 5.00

Content

192.168.1.1

Medium

SSL/TLS: Deprecated SSLv2 and SSLv3 Protocol Detection

Medium

SSL/TLS: Report Weak Cipher Suites

Medium

Weak Key Exchange (KEX) Algorithm(s) Supported (SSH)

Medium

Weak Host Key Algorithm(s) (SSH)

Medium

SSL/TLS: Server Certificate / Certificate in Chain with RSA keys less than 2048 bits

Medium

SSL/TLS: Renegotiation DoS Vulnerability (CVE-2011-1473, CVE-2011-5094)

Medium

DNS Cache Snooping Vulnerability (UDP) - Active Check

Medium

Weak Encryption Algorithm(s) Supported (SSH)

Medium

SSL/TLS: Deprecated TLSv1.0 and TLSv1.1 Protocol Detection

Low

Weak MAC Algorithm(s) Supported (SSH)

Low

SSL/TLS: TLS/SPDY Protocol Information Disclosure Vulnerability (CRIME)

Low

TCP Timestamps Information Disclosure

Low

ICMP Timestamp Reply Information Disclosure

192.168.1.178

Medium

Weak Host Key Algorithm(s) (SSH)

Medium

Weak Key Exchange (KEX) Algorithm(s) Supported (SSH)

Medium

SSL/TLS: Renegotiation DoS Vulnerability (CVE-2011-1473, CVE-2011-5094)

Medium

SSL/TLS: Certificate Expired

Medium

SSL/TLS: Certificate Expired

Medium

SSL/TLS: Known Untrusted / Dangerous Certificate Authority (CA) Detection

Medium

SSL/TLS: Known Untrusted / Dangerous Certificate Authority (CA) Detection

Low

TCP Timestamps Information Disclosure

Low

ICMP Timestamp Reply Information Disclosure

192.168.1.188

Medium

Weak Host Key Algorithm(s) (SSH)

Medium

Weak Key Exchange (KEX) Algorithm(s) Supported (SSH)

Medium

SSL/TLS: Known Untrusted / Dangerous Certificate Authority (CA) Detection

Medium

SSL/TLS: Renegotiation DoS Vulnerability (CVE-2011-1473, CVE-2011-5094)

Medium

SSL/TLS: Certificate Expired

Medium

SSL/TLS: Certificate Expired

Medium

SSL/TLS: Known Untrusted / Dangerous Certificate Authority (CA) Detection

Low

TCP Timestamps Information Disclosure

Low

ICMP Timestamp Reply Information Disclosure

192.168.1.108

Low

TCP Timestamps Information Disclosure

Low

ICMP Timestamp Reply Information Disclosure

192.168.1.110

Low

TCP Timestamps Information Disclosure

Low

ICMP Timestamp Reply Information Disclosure

192.168.1.232

Low

TCP Timestamps Information Disclosure

Low

ICMP Timestamp Reply Information Disclosure

Scan Summary

7 Hosts Scanned

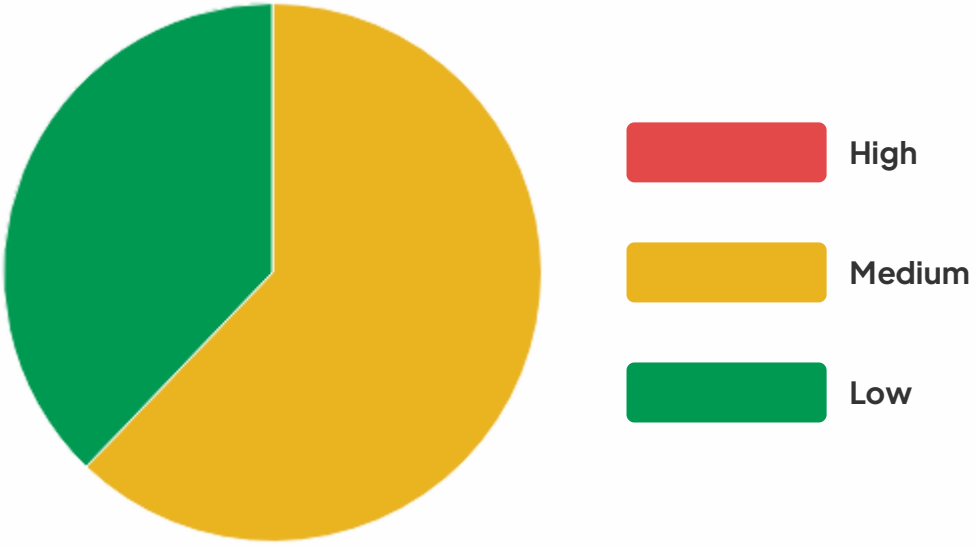
37 Total Findings

15 Unique CVEs

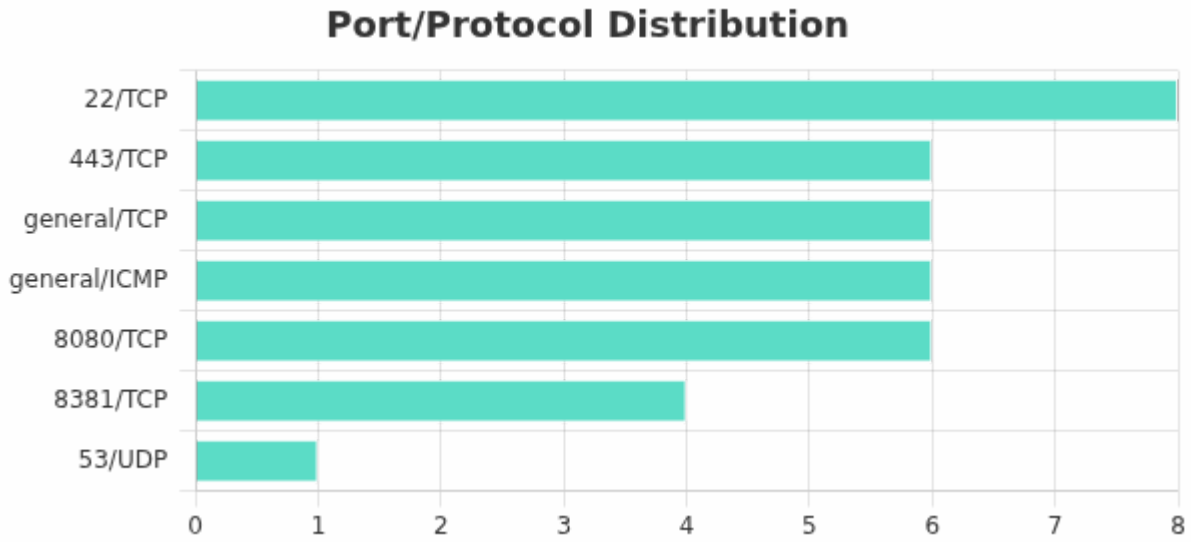
5.9 Max CVSS

0 Min CVSS

Vulnerabilities by Severity



Service/ Port distribution



Operating Systems Identified

Identified OS: 2

Applications Identified

Identified Applications: 11

192.168.1.1

Medium **443/tcp**

CVSS 5.9 - SSL/TLS: Deprecated SSLv2 and SSLv3 Protocol Detection - QoD: 98 - OID: N/A - Ver: 2026-06-03 07:18:10

Status: new

Times Seen: 1

First Seen: 2026-06-12 21:01

Last Seen: 2026-06-12 21:01

Summary: It was possible to detect the usage of the deprecated SSLv2 and/or SSLv3 protocol on this system.

Result: In addition to TLSv1.0+ the service is also providing the deprecated SSLv3 protocol and supports one or more ciphers. Those supported ciphers can be found in the 'SSL/TLS: Report Supported Cipher Suites' (OID: 1.3.6.1.4.1.25623.1.0.802067) VT.

Impact: An attacker might be able to use the known cryptographic flaws to eavesdrop the connection between clients and the service to get access to sensitive data transferred within the secured connection. Furthermore newly uncovered vulnerabilities in this protocols won't receive security updates anymore.

Affected: All services providing an encrypted communication using the, SSLv2 and/or SSLv3 protocols.

Insight: The SSLv2 and SSLv3 protocols contain known cryptographic flaws, like:, - CVE-2014-3566: Padding Oracle On Downgraded Legacy Encryption (POODLE), - CVE-2016-0800: Decrypting RSA with Obsolete and Weakened eNcryption (DROWN)

Method: Checks the used SSL protocols of the services provided by this system.

Solution (Mitigation): It is recommended to disable the deprecated SSLv2 and/or SSLv3 protocols in favor of the TLSv1.2+ protocols. Please see the references for more resources supporting you with this task.

Medium **443/tcp**

CVSS 5.9 - SSL/TLS: Report Weak Cipher Suites - QoD: 98 - OID: N/A - Ver: 2026-06-03 07:18:10

Status: new

Times Seen: 1

First Seen: 2026-06-12 21:01

Last Seen: 2026-06-12 21:01

Summary: This routine reports all weak SSL/TLS cipher suites accepted by a service.

Result: 'Weak' cipher suites accepted by this service via the SSLv3 protocol: TLS_ECDHE_RSA_WITH_RC4_128_SHA TLS_RSA_WITH_RC4_128_SHA TLS_RSA_WITH_SEED_CBC_SHA 'Weak' cipher suites accepted by this service via the TLSv1.0 protocol: TLS_ECDHE_RSA_WITH_RC4_128_SHA TLS_RSA_WITH_RC4_128_SHA TLS_RSA_WITH_SEED_CBC_SHA 'Weak' cipher suites accepted by this service via the TLSv1.1 protocol: TLS_ECDHE_RSA_WITH_RC4_128_SHA TLS_RSA_WITH_RC4_128_SHA TLS_RSA_WITH_SEED_CBC_SHA 'Weak' cipher suites accepted by this service via the TLSv1.2 protocol: TLS_ECDHE_RSA_WITH_RC4_128_SHA TLS_RSA_WITH_RC4_128_SHA TLS_RSA_WITH_SEED_CBC_SHA

Impact: This could allow remote attackers to obtain sensitive information or have other, unspecified impacts.

Affected: All services providing an encrypted communication using weak, SSL/TLS cipher suites.

Insight: These rules are applied for the evaluation of the cryptographic, strength:, - RC4 is considered to be weak (CVE-2013-2566, CVE-2015-2808), - Ciphers using 64 bit or less are considered to be vulnerable to brute force methods and, therefore considered as weak (CVE-2015-4000), - 1024 bit RSA authentication is considered to be insecure and therefore as weak, - Any cipher considered to be secure for only the next 10 years is considered as medium, - Any other cipher is considered as strong

Method: Checks previous collected cipher suites. NOTE: No severity for SMTP services with 'Opportunistic TLS' and weak cipher suites on port 25/tcp is reported. If too strong cipher suites are configured for this service the alternative would be to fall back to an even more insecure cleartext communication.

Solution (Mitigation): The configuration of this services should be changed so that it does not accept the listed weak cipher suites anymore. Please see the

references for more resources supporting you with this task.

Medium 22/tcp

CVSS 5.3 - Weak Key Exchange (KEX) Algorithm(s) Supported (SSH) - QoD: 80 - OID: N/A - Ver: 2024-06-14 05:05:48

Status: new

Times Seen: 1

First Seen: 2026-06-12 21:01

Last Seen: 2026-06-12 21:01

Summary: The remote SSH server is configured to allow / support weak key exchange (KEX) algorithm(s).

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 SHA-1

Impact: An attacker can quickly break individual connections.

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.

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

Solution (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.

Medium 22/tcp

CVSS 5.3 - Weak Host Key Algorithm(s) (SSH) - QoD: 80 - OID: N/A - Ver: 2024-06-14 05:05:48

Status: new

Times Seen: 1

First Seen: 2026-06-12 21:01

Last Seen: 2026-06-12 21:01

Summary: The remote SSH server is configured to allow / support weak host key algorithm(s).

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 Standard (DSS)

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)

Solution (Mitigation): Disable the reported weak host key algorithm(s).

Medium **443/tcp**

CVSS 5.3 - SSL/TLS: Server Certificate / Certificate in Chain with RSA keys less than 2048 bits - QoD: 80 - OID: N/A - Ver: 2021-12-10 12:48:00

Status: new

Times Seen: 1

First Seen: 2026-06-12 21:01

Last Seen: 2026-06-12 21:01

Summary: The remote SSL/TLS server certificate and/or any of the certificates in the certificate chain is using a RSA key with less than 2048 bits.

Result: The remote SSL/TLS server is using the following certificate(s) with a RSA key with less than 2048 bits (public-key-size:public-key-algorithm:serial:issuer): 1024:RSA:00DF3E423BE063DE80:CN=tplinkwifi.net,C=CN (Server certificate)

Impact: Using certificates with weak RSA key size can lead to unauthorized exposure of sensitive information.

Insight: SSL/TLS certificates using RSA keys with less than 2048 bits are, considered unsafe.

Method: Checks the RSA keys size of the server certificate and all certificates in chain for a size < 2048 bit.

Solution (Mitigation): Replace the certificate with a stronger key and reissue the certificates it signed.

Medium 443/tcp

CVSS 5 - SSL/TLS: Renegotiation DoS Vulnerability (CVE-2011-1473, CVE-2011-5094) - QoD: 70 - OID: N/A - Ver: 2026-03-13 05:54:58

Status: new

Times Seen: 1

First Seen: 2026-06-12 21:01

Last Seen: 2026-06-12 21:01

Summary: The remote SSL/TLS service is prone to a denial of service (DoS) vulnerability.

Result: The following indicates that the remote SSL/TLS service is affected: Protocol Version | Successful re-done SSL/TLS handshakes (Renegotiation) over an existing / already established SSL/TLS connection ----- TLSv1.0 | 10 TLSv1.1 | 10 TLSv1.2 | 10

Impact: The flaw might make it easier for remote attackers to cause a DoS (CPU consumption) by performing many renegotiations within a single connection.

Affected: Every SSL/TLS service which does not properly restrict, client-initiated renegotiation.

Insight: The flaw exists because the remote SSL/TLS service does not, properly restrict client-initiated renegotiation within the SSL and TLS protocols., Note: The referenced CVEs are affecting OpenSSL and Mozilla Network Security Services (NSS) but, both are in a DISPUTED state with the following rationale:, > It can also be argued that it is the responsibility of server deployments, not a security, library, to prevent or limit renegotiation when it is inappropriate within a specific environment., Both CVEs are still kept in this VT as a reference to the origin of this flaw.

Method: Checks if the remote service allows to re-do the same SSL/TLS handshake (Renegotiation) over an existing / already established SSL/TLS connection.

Solution (VendorFix): Users should contact their vendors for specific patch information. A general solution is to remove/disable renegotiation capabilities altogether from/in the affected SSL/TLS service.

Medium 53/udp

CVSS 5 - DNS Cache Snooping Vulnerability (UDP) - Active Check - QoD: 70 - OID: N/A - Ver: 2025-11-25 05:40:35

Status: new

Times Seen: 1

First Seen: 2026-06-12 21:01

Last Seen: 2026-06-12 21:01

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

Result: Received (an) answer(s) for a non-recursive query for "example.com". Result: 104.20.23.154

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.

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').

Method: Sends a crafted DNS query and checks the response. Notes: - No scan result is expected if localhost (127.0.0.1) was scanned (self scanning) - If the scanned network is e.g. a private LAN / private WAN which contains systems not accessible to the public (access restricted) and it is accepted that the service is disclosing information to this network please set the 'Network type' configuration of the following VT to e.g. 'Private LAN' or 'Private WAN': Global variable settings (OID: 1.3.6.1.4.1.25623.1.0.12288)

Solution (Mitigation): 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 - If the risk is accepted either create an override for this result or configure the 'Private LAN' setting mentioned earlier

Medium **22/tcp**

CVSS 4.3 - Weak Encryption Algorithm(s) Supported (SSH) - QoD: 80 - OID: N/A - Ver: 2024-06-14 05:05:48

Status: new

Times Seen: 1

First Seen: 2026-06-12 21:01

Last Seen: 2026-06-12 21:01

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

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

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.

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

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

Medium **443/tcp**

CVSS 4.3 - SSL/TLS: Deprecated TLSv1.0 and TLSv1.1 Protocol Detection - QoD: 98 - OID: N/A - Ver: 2026-06-03 07:18:10

Status: new

Times Seen: 1

First Seen: 2026-06-12 21:01

Last Seen: 2026-06-12 21:01

Summary: It was possible to detect the usage of the deprecated TLSv1.0 and/or TLSv1.1 protocol on this system.

Result: In addition to TLSv1.2+ the service is also providing the deprecated TLSv1.0 and TLSv1.1 protocols and supports one or more ciphers. Those supported ciphers can be found in the 'SSL/TLS: Report Supported Cipher Suites' (OID: 1.3.6.1.4.1.25623.1.0.802067) VT.

Impact: An attacker might be able to use the known cryptographic flaws to eavesdrop the connection between clients and the service to get access to sensitive data transferred within the secured connection. Furthermore newly uncovered vulnerabilities in this protocols won't receive security updates anymore.

Affected: - All services providing an encrypted communication using the, TLSv1.0 and/or TLSv1.1 protocols, - CVE-2023-41928: Kiloview P1 4G and P2 4G Video Encoder, - CVE-2024-41270: Gorush v1.18.4, - CVE-2025-3200: Multiple products from Wiesemann & Theis

Insight: The TLSv1.0 and TLSv1.1 protocols contain known cryptographic, flaws like:, - CVE-2011-3389: Browser Exploit Against SSL/TLS (BEAST), - CVE-2015-0204: Factoring Attack on RSA-EXPORT Keys Padding Oracle On Downgraded Legacy, Encryption (FREAK)

Method: Checks the used TLS protocols of the services provided by this system.

Solution (Mitigation): It is recommended to disable the deprecated TLSv1.0 and/or TLSv1.1 protocols in favor of the TLSv1.2+ protocols. Please see the references for more resources supporting you with this task.

Low **22/tcp**

CVSS 2.6 - Weak MAC Algorithm(s) Supported (SSH) - QoD: 80 - OID: N/A - Ver: 2024-06-14 05:05:48

Status: new

Times Seen: 1

First Seen: 2026-06-12 21:01

Last Seen: 2026-06-12 21:01

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

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 hmac-sha1-96

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

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

Low **443/tcp**

CVSS 2.6 - SSL/TLS: TLS/SPDY Protocol Information Disclosure Vulnerability (CRIME) - QoD: 98 - OID: N/A - Ver: 2025-01-17 15:39:18

Status: new

Times Seen: 1

First Seen: 2026-06-12 21:01

Last Seen: 2026-06-12 21:01

Summary: The TLS/SPDY protocols are prone to an information disclosure vulnerability.

Result: The remote service might be vulnerable to the "CRIME" attack because it provides the following TLS compression methods: Protocol:Compression Method TLSv1.0:DEFLATE TLSv1.1:DEFLATE TLSv1.2:DEFLATE SSLv3:DEFLATE

Impact: A man-in-the-middle attacker can exploit this issue to gain access to sensitive information that may aid in further attacks.

Affected: Services enabling TLS compression or supporting the SPDY protocol below SPDY/4 via HTTPS.

Solution (Mitigation): Disable TLS compression in the configuration of this services. If SPDY below 4 is used upgrade the webserver to a version which supports the successor protocol SPDY/4 or HTTP/2. Please see the references for more resources supporting you with this task.

Low **general/tcp**

CVSS 2.6 - TCP Timestamps Information Disclosure - QoD: 80 - OID: N/A - Ver: 2023-12-15 16:10:08

Status: new

Times Seen: 1

First Seen: 2026-06-12 21:01

Last Seen: 2026-06-12 21:01

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

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: 3019907439 Packet 2: 3019908514

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

Affected: TCP implementations that implement RFC1323/RFC7323.

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

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.

Solution (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.

Low general/icmp

CVSS 2.1 - ICMP Timestamp Reply Information Disclosure - QoD: 80 - OID: N/A - Ver: 2025-01-21 05:37:33

Status: new

Times Seen: 1

First Seen: 2026-06-12 21:01

Last Seen: 2026-06-12 21:01

Summary: The remote host responded to an ICMP timestamp request.

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.

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.

Method: Sends an ICMP Timestamp (Type 13) request and checks if a Timestamp Reply (Type 14) is received.

Solution (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)

192.168.1.178

Medium 22/tcp

CVSS 5.3 - Weak Host Key Algorithm(s) (SSH) - QoD: 80 - OID: N/A - Ver: 2024-06-14 05:05:48

Status: new

Times Seen: 1

First Seen: 2026-06-12 21:01

Last Seen: 2026-06-12 21:01

Summary: The remote SSH server is configured to allow / support weak host key algorithm(s).

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 Standard (DSS)

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)

Solution (Mitigation): Disable the reported weak host key algorithm(s).

Medium 22/tcp

CVSS 5.3 - Weak Key Exchange (KEX) Algorithm(s) Supported (SSH) - QoD: 80 - OID: N/A - Ver: 2024-06-14 05:05:48

Status: new

Times Seen: 1

First Seen: 2026-06-12 21:01

Last Seen: 2026-06-12 21:01

Summary: The remote SSH server is configured to allow / support weak key exchange (KEX) algorithm(s).

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 SHA-1

Impact: An attacker can quickly break individual connections.

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.

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

Solution (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.

Medium **8080/tcp**

CVSS 5 - SSL/TLS: Renegotiation DoS Vulnerability (CVE-2011-1473, CVE-2011-5094) - QoD: 70 - OID: N/A - Ver: 2026-03-13 05:54:58

Status: new

Times Seen: 1

First Seen: 2026-06-12 21:01

Last Seen: 2026-06-12 21:01

Summary: The remote SSL/TLS service is prone to a denial of service (DoS) vulnerability.

Result: The following indicates that the remote SSL/TLS service is affected: Protocol Version | Successful re-done SSL/TLS handshakes (Renegotiation) over an existing / already established SSL/TLS connection ----- TLSv1.2 | 10

Impact: The flaw might make it easier for remote attackers to cause a DoS (CPU consumption) by performing many renegotiations within a single connection.

Affected: Every SSL/TLS service which does not properly restrict, client-initiated renegotiation.

Insight: The flaw exists because the remote SSL/TLS service does not, properly restrict client-initiated renegotiation within the SSL and TLS protocols., Note: The referenced CVEs are affecting OpenSSL and Mozilla Network Security Services (NSS) but, both are in a DISPUTED state with the following rationale:, > It can also be argued that it is the responsibility of server deployments, not a security, library, to prevent or limit renegotiation when it is inappropriate within a specific environment., Both CVEs are still kept in this VT as a reference to the origin of this flaw.

Method: Checks if the remote service allows to re-do the same SSL/TLS handshake (Renegotiation) over an existing / already established SSL/TLS connection.

Solution (VendorFix): Users should contact their vendors for specific patch information. A general solution is to remove/disable renegotiation capabilities altogether from/in the affected SSL/TLS service.

Medium **8080/tcp**

CVSS 5 - SSL/TLS: Certificate Expired - QoD: 99 - OID: N/A - Ver: 2026-03-13 05:54:58

Status: new

Times Seen: 1

First Seen: 2026-06-12 21:01

Last Seen: 2026-06-12 21:01

Summary: The remote server's SSL/TLS certificate has already expired.

Result: The certificate of the remote service expired on 2022-09-14 06:04:16. Certificate details: fingerprint (SHA-1) | 40345DCBF21749C914720878EC5DAC7EB7BB69A3 fingerprint (SHA-256) | D6C3724D38069510AD549FFA77C41262B51F690D4B0BA37C10D152E274DA4716 issued by | CN=localhost public key algorithm | RSA public key size (bits) | 2048 serial | 455D0F2992A4312E74D367496E689FACCD0A6E1E signature algorithm | sha256WithRSAEncryption subject | CN=localhost subject alternative names (SAN) | None valid from | 2022-08-15 06:04:16 UTC valid until | 2022-09-14 06:04:16 UTC

Insight: This script checks expiry dates of certificates associated with, SSL/TLS-enabled services on the target and reports whether any have already expired.

Solution (Mitigation): Replace the SSL/TLS certificate by a new one.

Medium **8381/tcp**

CVSS 5 - SSL/TLS: Certificate Expired - QoD: 99 - OID: N/A - Ver: 2026-03-13 05:54:58

Status: new

Times Seen: 1

First Seen: 2026-06-12 21:01

Last Seen: 2026-06-12 21:01

Summary: The remote server's SSL/TLS certificate has already expired.

Result: The certificate of the remote service expired on 2022-09-14 06:04:16. Certificate details: fingerprint (SHA-1) |

40345DCBF21749C914720878EC5DAC7EB7BB69A3 fingerprint (SHA-256) | D6C3724D38069510AD549FFA77C41262B51F690D4B0BA37C10D152E274DA4716 issued by | CN=localhost public key algorithm | RSA public key size (bits) | 2048 serial | 455D0F2992A4312E74D367496E689FACCD0A6E1E signature algorithm | sha256WithRSAEncryption subject | CN=localhost subject alternative names (SAN) | None valid from | 2022-08-15 06:04:16 UTC valid until | 2022-09-14 06:04:16 UTC

Insight: This script checks expiry dates of certificates associated with, SSL/TLS-enabled services on the target and reports whether any have already expired.

Solution (Mitigation): Replace the SSL/TLS certificate by a new one.

Medium **8080/tcp**

CVSS 5 - SSL/TLS: Known Untrusted / Dangerous Certificate Authority (CA) Detection - QoD: 99 - OID: N/A - Ver: 2024-06-14 05:05:48

Status: new

Times Seen: 1

First Seen: 2026-06-12 21:01

Last Seen: 2026-06-12 21:01

Summary: The service is using an SSL/TLS certificate from a known untrusted and/or dangerous certificate authority (CA).

Result: The certificate of the remote service is signed by the following untrusted and/or dangerous CA: Issuer: CN=localhost Certificate details: fingerprint (SHA-1) | 40345DCBF21749C914720878EC5DAC7EB7BB69A3 fingerprint (SHA-256) | D6C3724D38069510AD549FFA77C41262B51F690D4B0BA37C10D152E274DA4716 issued by | CN=localhost public key algorithm | RSA public key size (bits) | 2048 serial | 455D0F2992A4312E74D367496E689FACCD0A6E1E signature algorithm | sha256WithRSAEncryption subject | CN=localhost subject alternative names (SAN) | None valid from | 2022-08-15 06:04:16 UTC valid until | 2022-09-14 06:04:16 UTC

Impact: An attacker could use this for man-in-the-middle (MITM) attacks, accessing sensible data and other attacks.

Method: The script reads the certificate used by the target host and checks if it was signed by a known untrusted and/or dangerous CA.

Solution (Mitigation): Replace the SSL/TLS certificate with one signed by a trusted CA.

Medium **8381/tcp**

CVSS 5 - SSL/TLS: Known Untrusted / Dangerous Certificate Authority (CA) Detection - QoD: 99 - OID: N/A - Ver: 2024-06-14 05:05:48

Status: new

Times Seen: 1

First Seen: 2026-06-12 21:01

Last Seen: 2026-06-12 21:01

Summary: The service is using an SSL/TLS certificate from a known untrusted and/or dangerous certificate authority (CA).

Result: The certificate of the remote service is signed by the following untrusted and/or dangerous CA: Issuer: CN=localhost Certificate details: fingerprint (SHA-1) | 40345DCBF21749C914720878EC5DAC7EB7BB69A3 fingerprint (SHA-256) | D6C3724D38069510AD549FFA77C41262B51F690D4B0BA37C10D152E274DA4716 issued by | CN=localhost public key algorithm | RSA public key size (bits) | 2048 serial | 455D0F2992A4312E74D367496E689FACCD0A6E1E signature algorithm | sha256WithRSAEncryption subject | CN=localhost subject alternative names (SAN) | None valid from | 2022-08-15 06:04:16 UTC valid until | 2022-09-14 06:04:16 UTC

Impact: An attacker could use this for man-in-the-middle (MITM) attacks, accessing sensible data and other attacks.

Method: The script reads the certificate used by the target host and checks if it was signed by a known untrusted and/or dangerous CA.

Solution (Mitigation): Replace the SSL/TLS certificate with one signed by a trusted CA.

Low general/tcp

CVSS 2.6 - TCP Timestamps Information Disclosure - QoD: 80 - OID: N/A - Ver: 2023-12-15 16:10:08

Status: new

Times Seen: 1

First Seen: 2026-06-12 21:01

Last Seen: 2026-06-12 21:01

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

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: 1938729361 Packet 2: 1938729471

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

Affected: TCP implementations that implement RFC1323/RFC7323.

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

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.

Solution (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.

Low general/icmp

CVSS 2.1 - ICMP Timestamp Reply Information Disclosure - QoD: 80 - OID: N/A - Ver: 2025-01-21 05:37:33

Status: new

Times Seen: 1

First Seen: 2026-06-12 21:01

Last Seen: 2026-06-12 21:01

Summary: The remote host responded to an ICMP timestamp request.

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.

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.

Method: Sends an ICMP Timestamp (Type 13) request and checks if a Timestamp Reply (Type 14) is received.

Solution (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)

192.168.1.188

Medium 22/tcp

CVSS 5.3 - Weak Host Key Algorithm(s) (SSH) - QoD: 80 - OID: N/A - Ver: 2024-06-14 05:05:48

Status: new

Times Seen: 1

First Seen: 2026-06-12 21:01

Last Seen: 2026-06-12 21:01

Summary: The remote SSH server is configured to allow / support weak host key algorithm(s).

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 Standard (DSS)

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)

Solution (Mitigation): Disable the reported weak host key algorithm(s).

Medium 22/tcp

CVSS 5.3 - Weak Key Exchange (KEX) Algorithm(s) Supported (SSH) - QoD: 80 - OID: N/A - Ver: 2024-06-14 05:05:48

Status: new

Times Seen: 1

First Seen: 2026-06-12 21:01

Last Seen: 2026-06-12 21:01

Summary: The remote SSH server is configured to allow / support weak key exchange (KEX) algorithm(s).

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 SHA-1

Impact: An attacker can quickly break individual connections.

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.

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

Solution (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.

Medium **8080/tcp**

CVSS 5 - SSL/TLS: Known Untrusted / Dangerous Certificate Authority (CA) Detection - QoD: 99 - OID: N/A - Ver: 2024-06-14 05:05:48

Status: new

Times Seen: 1

First Seen: 2026-06-12 21:01

Last Seen: 2026-06-12 21:01

Summary: The service is using an SSL/TLS certificate from a known untrusted and/or dangerous certificate authority (CA).

Result: The certificate of the remote service is signed by the following untrusted and/or dangerous CA: Issuer: CN=localhost Certificate details: fingerprint (SHA-1) | 5DD177B670A0B017AB853A78FA6260DC53F0AFB4 fingerprint (SHA-256) | 9040D1DBA3804CA974175C12E264A31130AB8433E4FD6DCBAA58623DDD848EB3 issued by | CN=localhost public key algorithm | RSA public key size (bits) | 2048 serial | 6B3FA81025EE3FF49515860EAE3B1E755DA46CE7 signature algorithm | sha256WithRSAEncryption subject | CN=localhost subject alternative names (SAN) | None valid from | 2022-08-15 06:04:15 UTC valid until | 2022-09-14 06:04:15 UTC

Impact: An attacker could use this for man-in-the-middle (MITM) attacks, accessing sensible data and other attacks.

Method: The script reads the certificate used by the target host and checks if it was signed by a known untrusted and/or dangerous CA.

Solution (Mitigation): Replace the SSL/TLS certificate with one signed by a trusted CA.

Medium **8080/tcp**

CVSS 5 - SSL/TLS: Renegotiation DoS Vulnerability (CVE-2011-1473, CVE-2011-5094) - QoD: 70 - OID: N/A - Ver: 2026-03-13 05:54:58

Status: new

Times Seen: 1

First Seen: 2026-06-12 21:01

Last Seen: 2026-06-12 21:01

Summary: The remote SSL/TLS service is prone to a denial of service (DoS) vulnerability.

Result: The following indicates that the remote SSL/TLS service is affected: Protocol Version | Successful re-done SSL/TLS handshakes (Renegotiation) over an existing / already established SSL/TLS connection ----- TLSv1.2 | 10

Impact: The flaw might make it easier for remote attackers to cause a DoS (CPU consumption) by performing many renegotiations within a single connection.

Affected: Every SSL/TLS service which does not properly restrict, client-initiated renegotiation.

Insight: The flaw exists because the remote SSL/TLS service does not, properly restrict client-initiated renegotiation within the SSL and TLS protocols., Note: The referenced CVEs are affecting OpenSSL and Mozilla Network Security Services (NSS) but, both are in a DISPUTED state with the following rationale:, > It can also be argued that it is the responsibility of server deployments, not a security, library, to prevent or limit renegotiation when it is inappropriate within a specific environment., Both CVEs are still kept in this VT as a reference to the origin of this flaw.

Method: Checks if the remote service allows to re-do the same SSL/TLS handshake (Renegotiation) over an existing / already established SSL/TLS connection.

Solution (VendorFix): Users should contact their vendors for specific patch information. A general solution is to remove/disable renegotiation capabilities altogether from/in the affected SSL/TLS service.

Medium **8080/tcp**

CVSS 5 - SSL/TLS: Certificate Expired - QoD: 99 - OID: N/A - Ver: 2026-03-13 05:54:58

Status: new

Times Seen: 1

First Seen: 2026-06-12 21:01

Last Seen: 2026-06-12 21:01

Summary: The remote server's SSL/TLS certificate has already expired.

Result: The certificate of the remote service expired on 2022-09-14 06:04:15. Certificate details: fingerprint (SHA-1) | 5DD177B670A0B017AB853A78FA6260DC53F0AFB4 fingerprint (SHA-256) | 9040D1DBA3804CA974175C12E264A31130AB8433E4FD6DCBAA58623DDD848EB3 issued by | CN=localhost public key algorithm | RSA public key size (bits) | 2048 serial | 6B3FA81025EE3FF49515860EAE3B1E755DA46CE7 signature algorithm | sha256WithRSAEncryption subject | CN=localhost subject alternative names (SAN) | None valid from | 2022-08-15 06:04:15 UTC valid until | 2022-09-14 06:04:15 UTC

Insight: This script checks expiry dates of certificates associated with, SSL/TLS-enabled services on the target and reports whether any have already expired.

Solution (Mitigation): Replace the SSL/TLS certificate by a new one.

Medium **8381/tcp**

CVSS 5 - SSL/TLS: Certificate Expired - QoD: 99 - OID: N/A - Ver: 2026-03-13 05:54:58

Status: new

Times Seen: 1

First Seen: 2026-06-12 21:01

Last Seen: 2026-06-12 21:01

Summary: The remote server's SSL/TLS certificate has already expired.

Result: The certificate of the remote service expired on 2022-09-14 06:04:15. Certificate details: fingerprint (SHA-1) | 5DD177B670A0B017AB853A78FA6260DC53F0AFB4 fingerprint (SHA-256) | 9040D1DBA3804CA974175C12E264A31130AB8433E4FD6DCBAA58623DDD848EB3 issued by | CN=localhost public key algorithm | RSA public key size (bits) | 2048 serial | 6B3FA81025EE3FF49515860EAE3B1E755DA46CE7 signature algorithm | sha256WithRSAEncryption subject | CN=localhost subject alternative names (SAN) | None valid from | 2022-08-15 06:04:15 UTC valid until | 2022-09-14 06:04:15 UTC

Insight: This script checks expiry dates of certificates associated with, SSL/TLS-enabled services on the target and reports whether any have already expired.

Solution (Mitigation): Replace the SSL/TLS certificate by a new one.

Medium 8381/tcp

CVSS 5 - SSL/TLS: Known Untrusted / Dangerous Certificate Authority (CA) Detection - QoD: 99 - OID: N/A - Ver: 2024-06-14 05:05:48

Status: new

Times Seen: 1

First Seen: 2026-06-12 21:01

Last Seen: 2026-06-12 21:01

Summary: The service is using an SSL/TLS certificate from a known untrusted and/or dangerous certificate authority (CA).

Result: The certificate of the remote service is signed by the following untrusted and/or dangerous CA: Issuer: CN=localhost Certificate details: fingerprint (SHA-1) | 5DD177B670A0B017AB853A78FA6260DC53F0AFB4 fingerprint (SHA-256) | 9040D1DBA3804CA974175C12E264A31130AB8433E4FD6DCBAA58623DDD848EB3 issued by | CN=localhost public key algorithm | RSA public key size (bits) | 2048 serial | 6B3FA81025EE3FF49515860EAE3B1E755DA46CE7 signature algorithm | sha256WithRSAEncryption subject | CN=localhost subject alternative names (SAN) | None valid from | 2022-08-15 06:04:15 UTC valid until | 2022-09-14 06:04:15 UTC

Impact: An attacker could use this for man-in-the-middle (MITM) attacks, accessing sensible data and other attacks.

Method: The script reads the certificate used by the target host and checks if it was signed by a known untrusted and/or dangerous CA.

Solution (Mitigation): Replace the SSL/TLS certificate with one signed by a trusted CA.

Low general/tcp

CVSS 2.6 - TCP Timestamps Information Disclosure - QoD: 80 - OID: N/A - Ver: 2023-12-15 16:10:08

Status: new

Times Seen: 1

First Seen: 2026-06-12 21:01

Last Seen: 2026-06-12 21:01

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

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: 1938728932 Packet 2: 1938729044

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

Affected: TCP implementations that implement RFC1323/RFC7323.

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

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.

Solution (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.

Low general/icmp

CVSS 2.1 - ICMP Timestamp Reply Information Disclosure - QoD: 80 - OID: N/A - Ver: 2025-01-21 05:37:33

Status: new

Times Seen: 1

First Seen: 2026-06-12 21:01

Last Seen: 2026-06-12 21:01

Summary: The remote host responded to an ICMP timestamp request.

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.

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.

Method: Sends an ICMP Timestamp (Type 13) request and checks if a Timestamp Reply (Type 14) is received.

Solution (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)

192.168.1.108

Low general/tcp

CVSS 2.6 - TCP Timestamps Information Disclosure - QoD: 80 - OID: N/A - Ver: 2023-12-15 16:10:08

Status: new

Times Seen: 1

First Seen: 2026-06-12 21:01

Last Seen: 2026-06-12 21:01

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

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: 6914418 Packet 2: 6914530

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

Affected: TCP implementations that implement RFC1323/RFC7323.

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

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.

Solution (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.

Low general/icmp

CVSS 2.1 - ICMP Timestamp Reply Information Disclosure - QoD: 80 - OID: N/A - Ver: 2025-01-21 05:37:33

Status: new

Times Seen: 1

First Seen: 2026-06-12 21:01

Last Seen: 2026-06-12 21:01

Summary: The remote host responded to an ICMP timestamp request.

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.

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.

Method: Sends an ICMP Timestamp (Type 13) request and checks if a Timestamp Reply (Type 14) is received.

Solution (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)

192.168.1.110

Low general/tcp

CVSS 2.6 - TCP Timestamps Information Disclosure - QoD: 80 - OID: N/A - Ver: 2023-12-15 16:10:08

Status: new

Times Seen: 1

First Seen: 2026-06-12 21:01

Last Seen: 2026-06-12 21:01

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

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: 1938794501 Packet 2: 1938794611

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

Affected: TCP implementations that implement RFC1323/RFC7323.

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

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.

Solution (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.

Low **general/icmp**

CVSS 2.1 - ICMP Timestamp Reply Information Disclosure - QoD: 80 - OID: N/A - Ver: 2025-01-21 05:37:33

Status: new

Times Seen: 1

First Seen: 2026-06-12 21:01

Last Seen: 2026-06-12 21:01

Summary: The remote host responded to an ICMP timestamp request.

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.

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.

Method: Sends an ICMP Timestamp (Type 13) request and checks if a Timestamp Reply (Type 14) is received.

Solution (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)

192.168.1.232

Low general/tcp

CVSS 2.6 - TCP Timestamps Information Disclosure - QoD: 80 - OID: N/A - Ver: 2023-12-15 16:10:08

Status: new

Times Seen: 1

First Seen: 2026-06-12 21:01

Last Seen: 2026-06-12 21:01

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

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: 3459682034 Packet 2: 3459683166

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

Affected: TCP implementations that implement RFC1323/RFC7323.

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

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.

Solution (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.

Low general/icmp

CVSS 2.1 - ICMP Timestamp Reply Information Disclosure - QoD: 80 - OID: N/A - Ver: 2025-01-21 05:37:33

Status: new

Times Seen: 1

First Seen: 2026-06-12 21:01

Last Seen: 2026-06-12 21:01

Summary: The remote host responded to an ICMP timestamp request.

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.

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.

Method: Sends an ICMP Timestamp (Type 13) request and checks if a Timestamp Reply (Type 14) is received.

Solution (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)