NUCLEI VULNERABILITY SCAN

ATTACKER'S NARRATIVE

An attacker targeting the systems identified in the vulnerability scan could orchestrate a sophisticated attack chain beginning with the OpenSSH Terrapin vulnerability (CVE-2023-48795) on scanme.nmap.org. This medium-severity vulnerability allows attackers to bypass integrity checks and downgrade security features. Combined with the multiple SSH weak algorithm configurations detected (weak MAC algorithms, CBC mode ciphers, and weak key exchange algorithms), an attacker could compromise the secure channel, potentially intercepting or modifying communications. The expired SSL certificate and weak cipher suites on zero.webappsecurity.com further compound the risk by enabling man-in-the-middle attacks against web traffic.

The Apache mod_negotiation misconfiguration on scanme.nmap.org provides attackers with additional reconnaissance capabilities through pseudo directory listing. This information disclosure vulnerability could reveal sensitive file structures and naming conventions, allowing attackers to more effectively target subsequent attacks. With knowledge of the server's file structure and the ability to compromise SSH connections, attackers could potentially gain unauthorized access to systems, escalate privileges, or exfiltrate sensitive data.

To remediate these vulnerabilities, immediate action should be taken to update OpenSSH to version 9.6 or later to address the Terrapin vulnerability. SSH configurations should be hardened by disabling weak algorithms, including CBC mode ciphers, weak MAC algorithms, and outdated key exchange methods, while enforcing stronger alternatives like AES-GCM. The expired SSL certificate on zero.webappsecurity.com should be renewed, and TLS configurations should be updated to remove support for weak cipher suites. Finally, Apache's mod_negotiation module should be disabled or properly configured to prevent information disclosure through directory listings.

Nuclei Vulnerability Scan Report

Scan completed: Wed Apr 8 13:49:07 +0000 UTC 2026

Findings Summary

Hostname/IP	Finding	Severity
scanme.nmap.org:22	OpenSSH Terrapin Attack - Detection	medium
scanme.nmap.org:22	SSH Weak Algorithms Supported	medium
scanme.nmap.org	Apache mod_negotiation - Pseudo Directory Listing	low
scanme.nmap.org:22	SSH Diffie-Hellman Modulus <= 1024 Bits	low
scanme.nmap.org:22	SSH Server CBC Mode Ciphers Enabled	low
scanme.nmap.org:22	SSH Weak MAC Algorithms Enabled	low
scanme.nmap.org:22	SSH Weak Key Exchange Algorithms Enabled	low
zero.webappsecurity.com	Weak Cipher Suites Detection	low
zero.webappsecurity.com	Expired SSL Certificate	low

Finding Details

OpenSSH Terrapin Attack - Detection (CVE-2023-48795) on scanme.nmap.org:22

Details: CVE-2023-48795 matched at scanme.nmap.org:22
Protocol: JAVASCRIPT
Full URL: scanme.nmap.org:22
Timestamp: Wed Apr 8 13:43:39 +0000 UTC 2026

Template Information

Key	Value
Name	OpenSSH Terrapin Attack - Detection
Authors	pussycat0x
Tags	cve, cve2023, packetstorm, seclists, js, ssh, network, passive, openbsd, vkev, vuln
Severity	medium
Description	The SSH transport protocol with certain OpenSSH extensions, found in OpenSSH before 9.6 and other products, allows remote attackers to bypass integrity checks such that some packets are omitted (from the extension negotiation message), and a client and server may consequently end up with a connection for which some security features have been downgraded or disabled, aka a Terrapin attack. This occurs because the SSH Binary Packet Protocol (BPP), implemented by these extensions, mishandles the handshake phase and mishandles use of sequence numbers. For example, there is an effective attack against SSH's use of ChaCha20-Poly1305 (and CBC with Encrypt-then-MAC). The bypass occurs in chacha20-poly1305@openssh.com and (if CBC is used) the -etm@openssh.com MAC algorithms. This also affects Maverick Synergy Java SSH API before 3.1.0-SNAPSHOT, Dropbear through 2022.83, Ssh before 5.1.1 in Erlang/OTP, PuTTY before 0.80, AsyncSSH before 2.14.2, golang.org/x/crypto before 0.17.0, libssh before 0.10.6, libssh2 through 1.11.0, Thorn Tech SFTP Gateway before 3.4.6, Tera Term before 5.1, Paramiko before 3.4.0, jsch before 0.2.15, SFTPGo before 2.5.6, Netgate pfSense Plus through 23.09.1, Netgate pfSense CE through 2.7.2, HPN-SSH through 18.2.0, ProFTPD before 1.3.8b (and before 1.3.9rc2), ORYX CycloneSSH before 2.3.4, NetSarang XShell 7 before Build 0144, CrushFTP before 10.6.0, ConnectBot SSH library before 2.2.22, Apache MINA sshd through 2.11.0, sshj through 0.37.0, TinySSH through 20230101, trilead-ssh2 6401, LANCOM LCOS and LANconfig, FileZilla before 3.66.4, Nova before 11.8, PKIX-SSH before 14.4, SecureCRT before 9.4.3, Transmit5 before 5.10.4, Win32-OpenSSH before 9.5.0.0p1-Beta, WinSCP before 6.2.2, Bitvise SSH Server before 9.32, Bitvise SSH Client before 9.33, KiTTY through 0.76.1.13, the net-ssh gem 7.2.0 for Ruby, the mscdex ssh2 module before 1.15.0 for Node.js, the thrussh library before 0.35.1 for Rust, and the Russh crate before 0.40.2 for Rust.
Remediation	One can address this vulnerability by temporarily disabling the affected chacha20-poly1305@openssh.com encryption and -etm@openssh.com MAC algorithms in the configuration of the SSH server (or client), and instead utilize unaffected algorithms like AES-GCM.
CVSS-Metrics	[CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:H/A:N](https://www.first.org/cvss/calculator/3.1#CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:H/A:N)
CWE-ID	[CWE-354](https://cwe.mitre.org/data/definitions/354.html)
CVE-ID	[CVE-2023-48795](https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2023-48795)
CVSS-Score	5.90
vendor	openbsd
product	openssh

Request

```
const m = require("nuclei/ssh");
const c = m.SSHClient();
const response = c.ConnectSSHInfoMode(Host, Port);

function SupportsChaCha20() {
    const CiphersClientServer = response.ServerKex.CiphersClientServer;
    const csexists = CiphersClientServer.includes("chacha20-poly1305@openssh.com");
    csexists;
    const CiphersServerClient = response.ServerKex.CiphersServerClient;
    const scexist = CiphersServerClient.includes("chacha20-poly1305@openssh.com");
    scexist;
    return csexists || scexist;
}

function SupportsCbcEtm() {
    const EncryCCS = response.ServerKex.CiphersClientServer;
    const EncryCCSsuf = EncryCCS.some(value => value.endsWith("-cbc"));
    EncryCCSsuf;

    const Macacs = response.ServerKex.MACsClientServer;
    const MacacsSuf = Macacs.some(value => value.endsWith("-etm@openssh.com"));
    MacacsSuf;

    const EncrySC = response.ServerKex.CiphersServerClient;
    const EncrySCSuf = EncrySC.some(value => value.endsWith("-cbc"));
```

```
    EncrySCSuf;

    const Macasc = response.ServerKex.MACsServerClient;
    const MacascSuf = Macasc.some(value => value.endsWith("-etm@openssh.com"));
    MacascSuf;
    return EncryCCSsuf && MacacsSuf || EncrySCSuf && MacascSuf;

}

function SupportsStrictKex() {
    const SuStrictKex = response.ServerKex.KexAlgos;
    const hasSuffix = SuStrictKex.some(value => value.endsWith("kex-strict-s-v00@openssh.com"));
    return hasSuffix;

}

function IsVulnerable() {
    const vuln = ((SupportsChaCha20() || SupportsCbcEtm()) && !SupportsStrictKex())
    if (vuln === true) {
        return ("Vulnerable to Terrapin");
    }
}
Export(IsVulnerable())
```

Response

```
Vulnerable to Terrapin
```

References

- https://github.com/RUB-NDS/Terrapin-Scanner
- https://terrapin-attack.com/
- http://packetstormsecurity.com/files/176280/Terrapin-SSH-Connection-Weakening.html
- http://seclists.org/fulldisclosure/2024/Mar/21
- http://www.openwall.com/lists/oss-security/2023/12/18/3

SSH Weak Algorithms Supported (ssh-weak-algo-supported) on scanme.nmap.org:22

Details: ssh-weak-algo-supported matched at scanme.nmap.org:22
Protocol: JAVASCRIPT
Full URL: scanme.nmap.org:22
Timestamp: Wed Apr 8 13:43:58 +0000 UTC 2026

Template Information

Key	Value
Name	SSH Weak Algorithms Supported
Authors	pussycat0x
Tags	js, enum, ssh, misconfig, network, vuln
Severity	medium
Description	SSH weak algorithms are outdated cryptographic methods that pose security risks. Identifying and disabling these vulnerable algorithms is crucial for enhancing the overall security of SSH connections.
shodan-query	product:"OpenSSH"

Request

```
let m = require("nuclei/ssh");
let c = m.SSHClient();
let response = c.ConnectSSHInfoMode(Host, Port);
Export(response);
```

Response

```
{
  "Banner": "",
  "ServerID": {
    "Raw": "SSH-2.0-OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.13",
    "ProtoVersion": "2.0",
    "SoftwareVersion": "OpenSSH_6.6.1p1",
    "Comment": "Ubuntu-2ubuntu2.13"
  },
  "ClientID": null,
  "ServerKex": {"cookie":"ImZrOcymjGaRAlUXTATVqw==","kex_algorithms":["curve25519-sha256@libssh.org","ecdh-sha2-nistp256","ecdh-sha2-nistp384","ecdh-sha2-nistp521","diffie-hellman
... (truncated)
```

References

- https://www.tenable.com/plugins/nessus/90317

Apache mod_negotiation - Pseudo Directory Listing (apache-mod-negotiation-listing:exposed_files) on scanme.nmap.org

Details: apache-mod-negotiation-listing:exposed_files matched at scanme.nmap.org
Protocol: HTTP
Full URL: http://scanme.nmap.org/index
Timestamp: Wed Apr 8 13:43:08 +0000 UTC 2026

Template Information

Key	Value
Name	Apache mod_negotiation - Pseudo Directory Listing
Authors	0x_akoko
Tags	apache, misconfig, exposure, mod-negotiation
Severity	low
Description	Detected Apache server with mod_negotiation and MultiViews enabled, exposing a pseudo directory listing when invalid Accept headers are sent to extensionless filenames.
CVSS-Metrics	[CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N](https://www.first.org/cvss/calculator/3.1#CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N)
CWE-ID	[CWE-538](https://cwe.mitre.org/data/definitions/538.html)
CVSS-Score	5.30

Request

```
GET /index HTTP/1.1
Host: scanme.nmap.org
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/14.0.3 Safari/605.1.15
Accept: fake/fake
Accept-Encoding: gzip
```

Response

```
HTTP/1.1 406 Not Acceptable
Connection: close
Content-Length: 429
Alternates: {"index.html" 1 {type text/html}}
Content-Type: text/html; charset=iso-8859-1
Date: Wed, 08 Apr 2026 13:43:09 GMT
Server: Apache/2.4.7 (Ubuntu)
Tcn: list
Vary: negotiate

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>406 Not Acceptable</title>
</head><body>
<h1>Not Acceptable</h1>
<p>An appropriate representation of the requested resource /index could not be found on this server.</p>
Available variants:
<ul>
<li><a href="index.html">index.html</a> , type text/html</li>
</ul>
<hr>
<address>Apache/2.4.7 (Ubuntu) Server at scanme.nmap.org Port 80</address>
</body></html>
```

References

- https://www.acunetix.com/vulnerabilities/web/apache-mod_negotiation-filename-bruteforcing/
- https://cwe.mitre.org/data/definitions/538.html

SSH Diffie-Hellman Modulus <= 1024 Bits (ssh-diffie-hellman-logjam) on scanme.nmap.org:22

Details: ssh-diffie-hellman-logjam matched at scanme.nmap.org:22
Protocol: JAVASCRIPT
Full URL: scanme.nmap.org:22
Timestamp: Wed Apr 8 13:43:49 +0000 UTC 2026

Template Information

Key	Value
Name	SSH Diffie-Hellman Modulus <= 1024 Bits
Authors	pussycat0x
Tags	js, enum, ssh, misconfig, network, discovery
Severity	low
Description	SSH weak algorithms are outdated cryptographic methods that pose security risks. Identifying and disabling these vulnerable algorithms is crucial for enhancing the overall security of SSH connections.
shodan-query	product:"OpenSSH"

Request

```
let m = require("nuclei/ssh");
let c = m.SSHClient();
let response = c.ConnectSSHInfoMode(Host, Port);
Export(response);
```

Response

```
{
  "Banner": "",
  "ServerID": {
    "Raw": "SSH-2.0-OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.13",
    "ProtoVersion": "2.0",
    "SoftwareVersion": "OpenSSH_6.6.1p1",
    "Comment": "Ubuntu-2ubuntu2.13"
  },
  "ClientID": null,
  "ServerKex": {"cookie":"ImZrOcymjGaRAlUXTATVqw==","kex_algorithms":["curve25519-sha256@libssh.org","ecdh-sha2-nistp256","ecdh-sha2-nistp384","ecdh-sha2-nistp521","diffie-hellman
... (truncated)
```

References

- https://access.redhat.com/solutions/4278651

SSH Server CBC Mode Ciphers Enabled (ssh-cbc-mode-ciphers) on scanme.nmap.org:22

Details: ssh-cbc-mode-ciphers matched at scanme.nmap.org:22
Protocol: JAVASCRIPT
Full URL: scanme.nmap.org:22
Timestamp: Wed Apr 8 13:43:58 +0000 UTC 2026

Template Information

Key	Value
Name	SSH Server CBC Mode Ciphers Enabled
Authors	pussycat0x
Tags	js, enum, ssh, misconfig, network, vuln
Severity	low
Description	"SSH Server CBC Mode Ciphers Enabled" signifies that the SSH server supports Cipher Block Chaining (CBC) mode ciphers, which are known for potential vulnerabilities. This configuration poses a security risk, and it's recommended to disable CBC ciphers in favor of more secure alternatives for enhanced protection during data transmission.
shodan-query	product:"OpenSSH"

Request

```
let m = require("nuclei/ssh");
let c = m.SSHClient();
let response = c.ConnectSSHInfoMode(Host, Port);
Export(response);
```

Response

```
{
  "Banner": "",
  "ServerID": {
    "Raw": "SSH-2.0-OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.13",
    "ProtoVersion": "2.0",
    "SoftwareVersion": "OpenSSH_6.6.1p1",
    "Comment": "Ubuntu-2ubuntu2.13"
  },
  "ClientID": null,
  "ServerKex": {"cookie":"ImZrOcymjGaRAlUXTATVqw==","kex_algorithms":["curve25519-sha256@libssh.org","ecdh-sha2-nistp256","ecdh-sha2-nistp384","ecdh-sha2-nistp521","diffie-hellman
... (truncated)
```

References

- https://www.tenable.com/plugins/nessus/70658

SSH Weak MAC Algorithms Enabled (ssh-weak-mac-algo) on scanme.nmap.org:22

Details: ssh-weak-mac-algo matched at scanme.nmap.org:22
Protocol: JAVASCRIPT
Full URL: scanme.nmap.org:22
Timestamp: Wed Apr 8 13:43:58 +0000 UTC 2026

Template Information

Key	Value
Name	SSH Weak MAC Algorithms Enabled
Authors	pussycat0x
Tags	js, enum, ssh, misconfig, network, vuln
Severity	low
Description	The system's SSH configuration poses a security risk by allowing weak Message Authentication Code (MAC) algorithms, potentially exposing it to vulnerabilities and unauthorized access. It is crucial to update and strengthen the MAC algorithms for enhanced security.
shodan-query	product:"OpenSSH"

Request

```
let m = require("nuclei/ssh");
let c = m.SSHClient();
let response = c.ConnectSSHInfoMode(Host, Port);
Export(response);
```

Response

```
{
  "Banner": "",
  "ServerID": {
    "Raw": "SSH-2.0-OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.13",
    "ProtoVersion": "2.0",
    "SoftwareVersion": "OpenSSH_6.6.1p1",
    "Comment": "Ubuntu-2ubuntu2.13"
  },
  "ClientID": null,
  "ServerKex": {"cookie":"ImZrOcymjGaRAlUXTATVqw==","kex_algorithms":["curve25519-sha256@libssh.org","ecdh-sha2-nistp256","ecdh-sha2-nistp384","ecdh-sha2-nistp521","diffie-hellman
... (truncated)
```

References

- https://www.tenable.com/plugins/nessus/71049

SSH Weak Key Exchange Algorithms Enabled (ssh-weakkey-exchange-algo) on scanme.nmap.org:22

Details: ssh-weakkey-exchange-algo matched at scanme.nmap.org:22
Protocol: JAVASCRIPT
Full URL: scanme.nmap.org:22
Timestamp: Wed Apr 8 13:43:58 +0000 UTC 2026

Template Information

Key	Value
Name	SSH Weak Key Exchange Algorithms Enabled
Authors	pussycat0x
Tags	js, enum, ssh, misconfig, network, vuln
Severity	low
Description	SSH Weak Key Exchange Algorithms Enabled indicates that the SSH server or client is configured to allow the use of less secure key exchange methods, posing a potential security risk during the establishment of secure connections. It's crucial to update configurations to prioritize stronger key exchange algorithms.
shodan-query	product:"OpenSSH"

Request

```
let m = require("nuclei/ssh");
let c = m.SSHClient();
let response = c.ConnectSSHInfoMode(Host, Port);
Export(response);
```

Response

```
{
  "Banner": "",
  "ServerID": {
```

```
    "Raw": "SSH-2.0-OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.13",
    "ProtoVersion": "2.0",
    "SoftwareVersion": "OpenSSH_6.6.1p1",
    "Comment": "Ubuntu-2ubuntu2.13"
  },
  "ClientID": null,
  "ServerKex": {"cookie":"ImZrOcymjGaRAlUXTATVqw==","kex_algorithms":["curve25519-sha256@libssh.org","ecdh-sha2-nistp256","ecdh-sha2-nistp384","ecdh-sha2-nistp521","diffie-hellman
... (truncated)
```


References

- https://www.tenable.com/plugins/nessus/153953

Weak Cipher Suites Detection (weak-cipher-suites:tls-1.0) on zero.webappsecurity.com

Details: weak-cipher-suites:tls-1.0 matched at zero.webappsecurity.com
Protocol: SSL
Full URL: zero.webappsecurity.com:443
Timestamp: Wed Apr 8 13:44:28 +0000 UTC 2026

Template Information

Key	Value
Name	Weak Cipher Suites Detection
Authors	pussycat0x
Tags	ssl, tls, misconfig, vuln
Severity	low
Description	A weak cipher is defined as an encryption/decryption algorithm that uses a key of insufficient length. Using an insufficient length for a key in an encryption/decryption algorithm opens up the possibility (or probability) that the encryption scheme could be broken.

References

- https://www.acunetix.com/vulnerabilities/web/tls-ssl-weak-cipher-suites/
- http://ciphersuite.info

Expired SSL Certificate (expired-ssl) on zero.webappsecurity.com

Details: expired-ssl matched at zero.webappsecurity.com
Protocol: SSL
Full URL: zero.webappsecurity.com:443
Timestamp: Wed Apr 8 13:49:07 +0000 UTC 2026

Template Information

Key	Value
Name	Expired SSL Certificate
Authors	pdteam
Tags	ssl, tls, vuln
Severity	low
Description	After an SSL certificate expires, you will no longer be able to communicate over a secure, encrypted HTTPS connection.
Remediation	Purchase or generate a new SSL/TLS certificate to replace the existing one.

References

- https://www.acunetix.com/vulnerabilities/web/tls-ssl-certificate-about-to-expire/