

[] : escaped

A PENETRATION TEST OF

Internet Facing Infrastructure Assessment

FOR

Open Network Films

BY

Escaped

CONFIDENTIAL

12 Oct 2026

Escaped
550 W B Street
Floor 4
San Diego
CA 92103

Table of contents

1. Introduction	3
2. Scope	3
3. Open source intelligence (OSINT)	3
4. How to Use This Report	4
5. Executive Summary	5
6. Technical Summary	7
7. Vulnerability Summary	23
8. Visual Summary	25
9. Vulnerability Details	26
9.1 Exposed Apache Tomcat Manager Console with Default Credentials	26
9.2 Linux Kernel AF_ALG "Copy Fail" Local Privilege Escalation	26
9.3 Anonymous SMB Share Enumeration on Externally-Facing Host	27
9.4 Outdated and Insecure SSH Configuration	28
9.5 TLS 1.0 / 1.1 Still Enabled on Externally-Facing Service	29
9.6 Weak Encryption Methods Supported	30
9.7 Missing Security Headers	31
9.8 VNC Service Exposure	32
9.9 JQuery Outdated	33
10. Tools Used During Testing	33

1. Introduction

This report details the results of a manual penetration test of the Open Network Films Internet Facing Infrastructure Assessment performed during December 2025.

This report outlines the results collected during testing and any recommendations on the findings.

Sample Consultant of Escaped was the lead consultant on this test. For more information please email hello@escaped.com.

The test methodology was aligned with OWASP and PTES best practices.

2. Scope

Open Network Films requires penetration testing of their internet-facing infrastructure.

The following IP address ranges were supplied as targets for testing.

203.0.113.1/26
203.0.113.194/26
203.0.113.215/29
203.0.113.194/29
203.0.113.136/31
203.0.113.112/29
203.0.113.32/30
203.0.113.198/30
203.0.113.164/29
203.0.113.132/30
203.0.113.125/32

No user credentials were provided for testing.

3. Open source intelligence (OSINT)

Open Source Intelligence (OSINT) refers to information gathered from publicly accessible sources, such as websites on the Internet. A series of passive tests were conducted using the client's domain, querying platforms like Google, Bing, LinkedIn, and Shodan to identify email addresses and subdomain names.

Hosts found:

opennetworkfilms.example
helpdesk.opennetworkfilms.example
onepasswd.opennetworkfilms.example
partnerships.opennetworkfilms.example
path.opennetworkfilms.example
www.opennetworkfilms.example

4. How to Use This Report

An executive non-technical summary can be found in [Section 5](#) which explains the issues found in layman's terms.

A verbose technical summary detailing how the vulnerabilities were found and how to reproduce them can be found in [Section 6](#).

Concise reference vulnerability information can be found in [Section 7](#)

A risk-based approach is taken to reporting vulnerabilities.

More information can be obtained [here](#) on how the high, medium and low-risk factors are calculated.

We also list informational recommendations where weaknesses in security or policy could lead to a vulnerability.

5. Executive Summary

Escaped conducted a manual, unauthenticated penetration test of Open Network Films' internet-facing infrastructure between December 2025 and January 2026. The objective was to identify externally exploitable weaknesses, validate defensive controls, and assess the overall external attack surface in the context of modern threat activity. Testing was aligned with OWASP and PTES methodologies and focused exclusively on systems reachable from the public internet.

The assessment combined active service enumeration, version and configuration analysis, authenticated and unauthenticated probing of exposed administrative interfaces, and targeted exploitation of confirmed vulnerabilities where doing so was within scope. Approximately 70 internet-reachable hosts were enumerated across the in-scope address ranges. Where exploitation was performed, post-exploitation activity was strictly limited to evidence collection sufficient to demonstrate impact.

During testing, an intrusion detection and prevention mechanism was triggered by assessment activity and required temporary adjustment by the Open Network Films security team to allow testing to proceed. This confirms the presence of active monitoring and defensive controls, but also limited the ability to fully evaluate detection thresholds and response behavior under sustained attack conditions.

Critical and High Severity Findings

The most serious finding identified during the assessment is an externally-exposed Apache Tomcat Manager interface accepting default credentials. The Manager application was reachable directly from the public internet on a single host and authenticated successfully with the credential pair `tomcat / tomcat`. From this position, an attacker can deploy a malicious WAR archive and obtain immediate remote code execution as the Tomcat service account. The combination of internet exposure, default credentials, and a built-in code-deployment endpoint makes this a one-step compromise requiring no exploit development, and accordingly carries the highest priority for remediation.

From the Tomcat foothold, testing confirmed a viable privilege-escalation path on the underlying Linux host using CVE-2026-31431 ("Copy Fail"), a logic flaw in the Linux kernel's `AF_ALG` cryptographic interface that permits an unprivileged local user to corrupt arbitrary readable files within the kernel page cache. The host was running an unpatched Ubuntu 24.04 LTS kernel and was vulnerable to a publicly available, deterministic exploit. Because the affected page cache is shared between containers and the host, the same primitive enabled escape from a containerised workload onto the underlying server. Treated together, the Tomcat exposure and Copy Fail provide a complete, reliable, and largely undetectable path from the public internet to host-level root.

Separately, one Windows host exposes SMB (TCP 445) directly to the public internet and permits unauthenticated null-session enumeration of available shares. While no immediately readable sensitive content was identified during testing, internet-facing SMB also carries a long history of remote code execution vulnerabilities and credential-relay attack surface, and should not be reachable from untrusted networks under any circumstances.

Medium and Low Severity Findings

Beyond the headline issues, the most material remaining risk concerns Secure Shell (SSH) exposure. Multiple hosts run outdated OpenSSH releases, accept password-based authentication, and continue to advertise deprecated key exchange and cipher algorithms. This combination materially increases susceptibility to brute-force credential attacks, credential stuffing, downgrade attacks, and exploitation of any future vulnerability disclosed against the installed SSH versions.

Two related cryptographic findings affect HTTPS-facing services. One legacy endpoint continues to support TLS 1.0 and TLS 1.1, both of which have been formally deprecated by the IETF (RFC 8996) and removed from support by all major browsers; their continued availability permits downgrade attacks and exposes clients to long-known cipher and protocol weaknesses. A separate set of services advertises weak or deprecated symmetric and key exchange primitives that should be retired in favour of modern equivalents.

Additional medium- and low-severity findings include missing HTTP security headers across externally-facing web properties, an exposed VNC remote-desktop service, and an outdated client-side jQuery library. Individually these issues are unlikely to result in immediate compromise; collectively they reduce defensive depth, simplify attacker reconnaissance, and meaningfully increase the impact of any future attack chain.

Positive Observations

Several services demonstrated strong, modern configurations during testing. Multiple HTTPS endpoints offered TLS 1.3 with hardened cipher suites and resisted downgrade attempts. A number of high-value administrative services - including VPN gateways, container management platforms, and enterprise file-transfer infrastructure - were observed to be patched against publicly known exploits and rejected unauthenticated probing.

The triggering of intrusion detection and prevention controls during active testing further indicates the presence of operational monitoring and response capability, which is increasingly uncommon at this scale and represents a meaningful defensive baseline. No evidence of pre-existing compromise was observed during the assessment, and no high-severity unauthenticated remote code execution vulnerability was identified outside of the Tomcat finding noted above.

Strategic Risk Summary

Risk can be substantially reduced by treating the externally-exposed administrative surface as the primary class of remediation. The following actions are recommended in priority order:

- Remove the Apache Tomcat Manager interface from any internet-facing host, rotate all default credentials, and enforce IP allowlisting where the Manager remains required.
- Apply the vendor kernel patches addressing CVE-2026-31431 across all Linux servers; until patching is complete, block `AF_ALG` socket creation via seccomp and treat any affected container as suspect.
- Block TCP 445 (SMB) at the network edge - SMB should not be reachable from the public internet under any circumstance.
- Eliminate obsolete cryptographic and protocol support: disable TLS 1.0 and 1.1, retire weak SSH and TLS cipher suites, and disable password-based SSH authentication on any internet-reachable host.
- Reduce the count of internet-exposed administrative services by enforcing VPN-only or jump-host access to high-value management platforms (container management, file transfer, remote desktop, VPN administrative interfaces).
- Standardise hardened baselines for SSH, TLS, and HTTP-header configuration across all externally-facing services so that current and future deployments do not regress to insecure defaults.

Addressing the first three items removes the demonstrated end-to-end compromise path identified during testing. The remaining items reduce defensive depth and limit the impact of future vulnerabilities, credential compromise, and configuration drift.

6. Technical Summary

6.1 Apache Tomcat Manager - Default Credentials and Remote Code Execution

An externally-reachable Apache Tomcat installation was identified during HTTP service enumeration. Inspection of the management interface revealed that the bundled `/manager/html` application was deployed and reachable from the public internet, with no source-IP restrictions or upstream authentication enforced.

Service

Apache Tomcat 9.x, HTTP/8080, manager-html and manager-script applications enabled and reachable.

Affected Host

HOST	IP ADDRESS	PORT
host33.opennetworkfilms.example	203.0.113.243	8080

Authentication

The Manager interface accepted the credential pair `tomcat / tomcat` on first attempt. These are the default credentials present in unconfigured Tomcat installations and indicate that the `tomcat-users.xml` file was not customised after deployment.

Exploitation Path

From the authenticated Manager session, a JSP webshell was packaged into a Web Application Archive (WAR) and submitted via the Manager's deployment form. The deployment succeeded and the resulting context was immediately reachable. Issuing a request to the deployed endpoint returned an interactive command shell executing as the Tomcat service account, providing arbitrary code execution on the underlying host without any further interaction. End-to-end the chain (login, WAR upload, shell access) completed in well under one minute and required no exploit development.

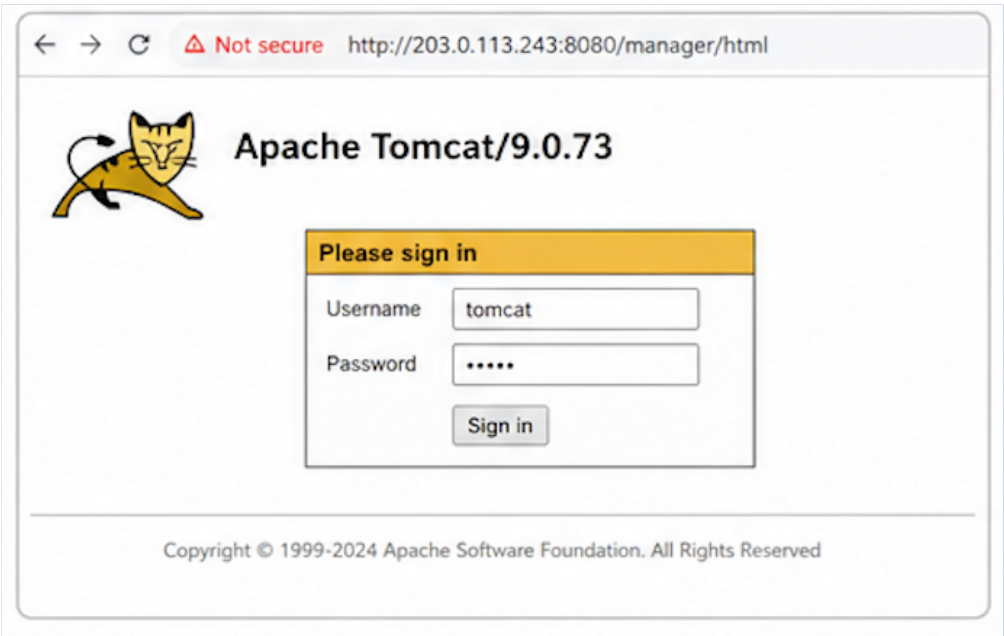


Figure 1 - Tomcat Manager Login Page (Externally Accessible)

Tomcat Web Application Manager

Message: OK

Manager

List Applications HTML Manager Help Manager Help Server Status

Applications

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle >= 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle >= 30 minutes
/examples	None specified	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle >= 30 minutes

Figure 2 - WAR File Upload via Manager Interface

Deploy - Upload WAR file

Select WAR file to upload

WAR file to deploy Choose File shell.war

Deploy

WAR file list

WAR files available for deployment

Name	Size	Last Modified
shell.war	4.62 KB	Dec 15, 2025, 10:42:15 AM

✓ WAR file /shell deployed successfully

Figure 3 - Webshell Command Execution

JSP Web Shell

Command: id Execute

Output:

```
uid=1001(tomcat) gid=1001(tomcat) groups=1001(tomcat)
```

Impact

The combination of internet-facing Manager exposure and default credentials provides a one-step path from anonymous internet attacker to authenticated remote code execution on the host. From this position, an attacker can read filesystem contents accessible to the Tomcat user, pivot to other internal hosts reachable from the server, persist via web-shell or service modification, and (as documented separately under CVE-2026-31431) escalate to root via local kernel exploitation.

Recommendations

- Disable the Manager and Host Manager applications on any Tomcat instance reachable from untrusted networks. If Manager access is genuinely required, restrict it to administrative source IP ranges via a reverse proxy or firewall.
- Remove or rename the default `tomcat`, `admin`, and `manager` account entries from `tomcat-users.xml` and rotate all credentials to long, unique values stored in a credential vault.
- Treat any host that has historically exposed an internet-reachable Manager interface as potentially compromised: review access logs, deployed contexts, file modification timestamps under `webapps/`, and outbound network activity for indicators of prior unauthorised access.

6.2 CVE-2026-31431 "Copy Fail" - Linux Kernel Local Privilege Escalation

Following confirmation of remote code execution via the Apache Tomcat Manager finding above, the underlying host was assessed for local privilege-escalation paths reachable from an unprivileged shell. The host was running an Ubuntu 24.04 LTS kernel that had not received the patches addressing CVE-2026-31431, a logic flaw in the Linux kernel's userspace cryptographic interface disclosed in May 2026.

Affected Host

HOST	IP ADDRESS	PORT
host33.opennetworkfilms.example	203.0.113.243	22

Vulnerability

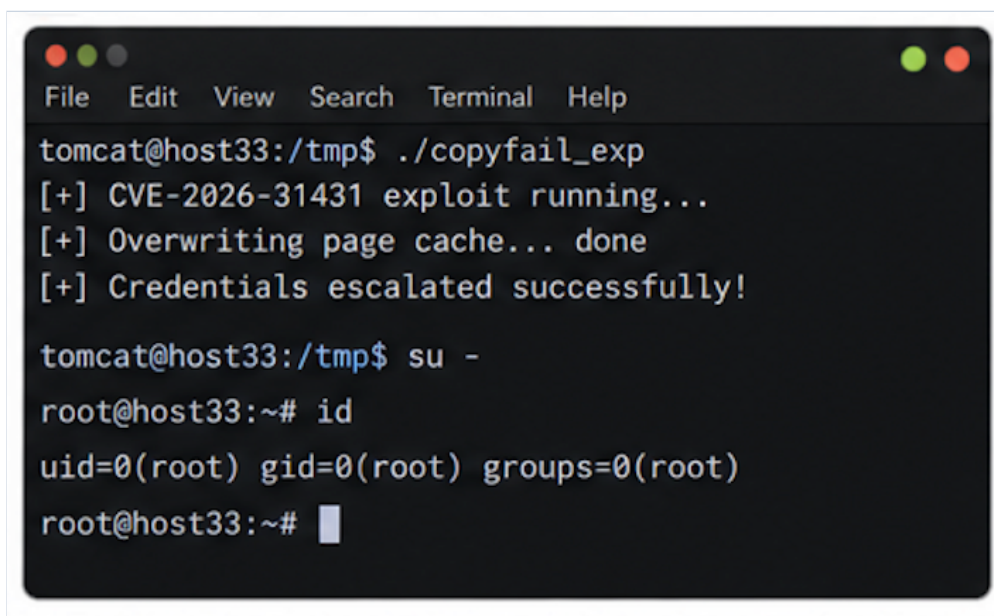
CVE-2026-31431 ("Copy Fail") is a flaw in the Linux kernel's `algif_aead` module, which exposes Authenticated Encryption with Associated Data primitives to userspace via the `AF_ALG` socket family. The vulnerability stems from an in-place memory optimisation introduced in 2017, in which the kernel reuses a single buffer as both source and destination during a cryptographic operation. By manipulating the interaction between `AF_ALG` sockets and the `splice()` system call, a local unprivileged user can trigger a controlled four-byte write into the kernel page cache representation of any file readable by that user. The condition is deterministic and requires no race window or memory corruption primitive beyond the controlled write itself.

Affected Software

The flaw is present in unpatched Linux kernels released between 2017 and the May 2026 disclosure, and was confirmed exploitable on the engagement host (Ubuntu 24.04 LTS). Vendor advisories at the time of testing covered Ubuntu, Red Hat Enterprise Linux, SUSE, Amazon Linux, Debian, Fedora, and Arch.

Exploitation

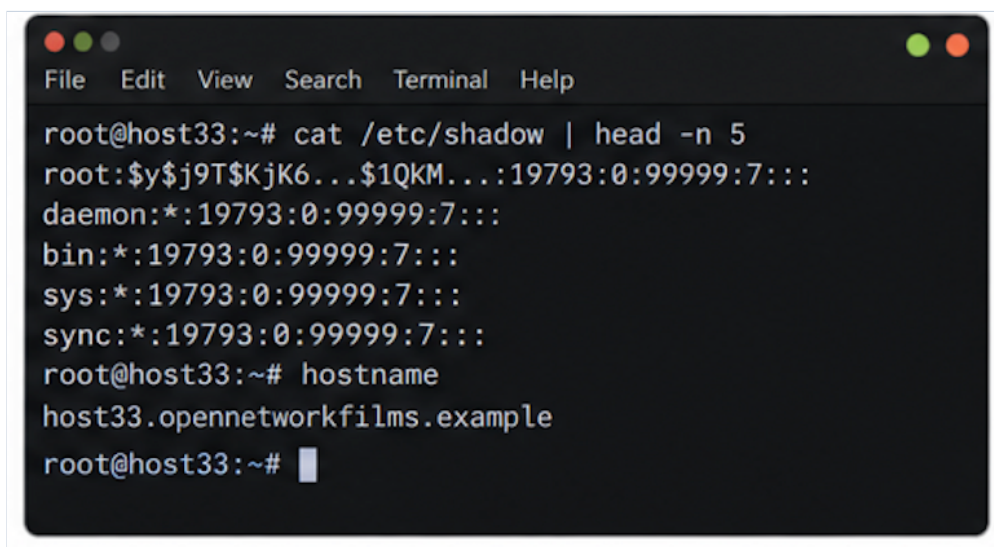
From the unprivileged shell obtained via the Tomcat finding, a publicly available proof-of-concept (a single self-contained script of approximately 732 bytes) was deployed and executed. The PoC opens an `AF_ALG` socket bound to `algif_aead`, prepares an in-memory buffer through `splice()`, and triggers the in-place encryption path against the page-cache representation of `/usr/bin/su`. After approximately one second of operation, executing `/usr/bin/su` from the same shell returned a root prompt without prompting for a password - the corrupted page-cache copy of the binary was used by the kernel for execution while the on-disk file remained byte-identical, leaving no host-based file-integrity evidence of the modification.



```
File Edit View Search Terminal Help
tomcat@host33:/tmp$ ./copyfail_exp
[+] CVE-2026-31431 exploit running...
[+] Overwriting page cache... done
[+] Credentials escalated successfully!

tomcat@host33:/tmp$ su -
root@host33:~# id
uid=0(root) gid=0(root) groups=0(root)
root@host33:~#
```

Figure 5 - Copy Fail Privilege Escalation Exploit



```
File Edit View Search Terminal Help
root@host33:~# cat /etc/shadow | head -n 5
root:$y$j9T$KjK6...$1QKM...:19793:0:99999:7:::
daemon*:19793:0:99999:7:::
bin*:19793:0:99999:7:::
sys*:19793:0:99999:7:::
sync*:19793:0:99999:7:::
root@host33:~# hostname
host33.opennetworkfilms.example
root@host33:~#
```

Figure 6 - Post-Exploit Root Shell Confirmation

Container Escape

To validate the documented container-escape behaviour, the same PoC was executed from inside an unprivileged container running on the host. The exploit succeeded identically and yielded a root shell on the host file system. This is consistent with the public analysis of the flaw: the Linux page cache is shared across containers and the host, and Copy Fail's primitive operates against that shared cache rather than against any container-local resource. Host-level isolation between containers is not preserved against this exploit.

Impact

The Tomcat finding (above) and Copy Fail combine to form a complete external-to-root compromise chain executable by an anonymous internet attacker, requiring no credential phishing, no exploit development, no privileged staging, and producing minimal host-based forensic artefacts. Either finding individually would warrant prompt remediation; together they materially elevate the host to imminent-compromise status.

Recommendations

- Apply the vendor kernel update addressing CVE-2026-31431 immediately on all affected Linux servers. Reboot is required for the patch to take effect.
- Where patching cannot be performed at once, block AF_ALG socket creation via a seccomp profile applied to all containerised workloads and to any service account that does not require it. Where the algif_aead module is not in use, blacklist it from loading entirely.
- Treat any container that previously shared a host with an untrusted workload - including the Tomcat service identified above - as having had host-level access for the duration of that exposure. Review host-level audit logs, persistent service definitions, and outbound network activity for indicators of post-exploitation pivoting.
- Review the broader Linux server estate for kernels older than the May 2026 patch level and prioritise their replacement, particularly any host running an internet-reachable application that could provide initial code execution.

6.3 Anonymous SMB Share Enumeration

One Windows host within the in-scope address range was identified as exposing SMB (Server Message Block, TCP 445) directly to the public internet. Probing of the service confirmed that the host accepted unauthenticated connections and permitted enumeration of available shares without supplying credentials, a configuration commonly referred to as "null session" access.

Service

SMB over TCP/445 on a Windows host. Service responded to NetBIOS-less direct SMB negotiation; SMBv2 was advertised. The host was reachable from arbitrary external source addresses with no perimeter restriction.

Affected Host

HOST	IP ADDRESS	PORT
host01.opennetworkfilms.example	203.0.113.198	445

Finding

Issuing the standard share-enumeration request `smbclient -L //<host> -N` from an external workstation returned a populated list of available shares. No authentication challenge was issued and no credential material was supplied. The contents of individual shares were not enumerated during testing in order to limit the scope of any unintentional data access.

Impact

Null-session share enumeration on its own discloses share names, which frequently themselves leak operationally sensitive information (project codenames, system roles, customer references). Depending on per-share access control, anonymous read or write access to one or more shares may also be possible

and should be presumed until verified otherwise. Independently of share-level access, exposing SMB to the public internet is recognised by every modern hardening guide as an unsupported configuration: SMB has a long and continuing history of pre-authentication remote code execution vulnerabilities (e.g. EternalBlue, SMBGhost), and even where the current build is fully patched, the service remains a high-value target for credential-relay attacks, brute-force authentication, and future zero-day exploitation.

Recommendations

- Block TCP 445 (and TCP 139) at the network edge. SMB should not be reachable from the public internet under any circumstance - internal access between trusted hosts is the only supported deployment model.
- If internal SMB is required, disable null sessions by setting `RestrictAnonymous` and `RestrictAnonymousSAM` appropriately, require SMB signing on both client and server, and enforce SMBv3 with encryption.
- Review the affected host's filesystem and event logs for evidence of prior anonymous access; treat the share name list as known-disclosed and reassess any naming convention that itself communicates sensitive information.

6.4 Secure Shell (SSH) Exposure

Secure Shell services were observed across multiple internet-reachable hosts in the in-scope ranges. Banner inspection, version probing, and active negotiation were used to characterise the configuration of each service. Several common weaknesses were identified across the estate and are summarised below.

Versions

The OpenSSH versions advertised by reachable services span a wide range, including releases that are well past their upstream support window. At least one host advertises a release that has been formally end-of-life for several years and is therefore unlikely to receive any future security update; multiple hosts advertise releases that, while still receiving distribution-vendor backports, are several major versions behind current. Older versions are progressively susceptible to a longer list of publicly documented vulnerabilities and protocol-level weaknesses.

Affected Hosts

IP	PORT	VERSION	PLATFORM	AUTH METHODS	ALGO / CRYPTO NOTES
203.0.113.242	22	OpenSSH 6.6.1	Linux	publickey, keyboard-interactive	Legacy SSH version
203.0.113.243	22	OpenSSH 8.0	Linux	publickey, gssapi-keyex, gssapi-with-mic, password	Baseline weak SSH configuration
203.0.113.243	33001	OpenSSH 8.0	Linux	publickey, gssapi-keyex, gssapi-with-mic, password	SSH on alt port; same weak crypto
203.0.113.193	22	OpenSSH 8.0	Linux	publickey, gssapi-keyex, gssapi-with-mic, password	Baseline weak SSH configuration
203.0.113.195	22	OpenSSH 8.0	Linux	publickey, gssapi-keyex, gssapi-with-mic, password	Baseline weak SSH configuration

IP	PORT	VERSION	PLATFORM	AUTH METHODS	ALGO / CRYPTO NOTES
203.0.113.196	22	OpenSSH 8.0	Linux	publickey, gssapi-keyex, gssapi-with-mic, password	Baseline weak SSH configuration
203.0.113.197	22	OpenSSH 8.0	Linux	publickey, gssapi-keyex, gssapi-with-mic, password	Baseline weak SSH configuration
203.0.113.199	33001	OpenSSH for Windows 9.5	Windows	publickey, password, keyboard-interactive	Password auth; outdated; alt port
203.0.113.206	22	OpenSSH 8.0	Linux	publickey, gssapi-keyex, gssapi-with-mic, password	Baseline weak SSH configuration
203.0.113.223	33001	OpenSSH 8.0	Linux	publickey, gssapi-with-mic, password	SSH on alt port; weak KEX
203.0.113.250	22	OpenSSH 8.0	Linux	publickey, password, keyboard-interactive	Weak KEX; duplicated host keys; SHA-1 algorithms

Authentication

Password-based authentication is enabled on multiple internet-reachable SSH services in addition to public-key authentication. Password authentication on internet-facing SSH is one of the most consistently exploited initial-access vectors observed in real-world incidents - it provides a continuous brute-force and credential-stuffing attack surface, generates persistent log noise that complicates anomaly detection, and creates dependency on individual user password hygiene rather than on cryptographic key material.

Algorithms and Cryptographic Configuration

Several services continue to advertise key-exchange, host-key, cipher, and MAC algorithms that have been deprecated in modern OpenSSH defaults. Notable items observed include legacy diffie-hellman-group exchanges with smaller modulus sizes, CBC-mode ciphers retained for compatibility, and HMAC constructions with deprecated truncation profiles. Individually each represents a reduction in margin against known cryptographic attacks; collectively they indicate that the SSH configuration has not been refreshed against current hardening guidance.

Recommendations

- Upgrade all internet-reachable SSH services to a current OpenSSH release supported by the operating system vendor; replace any host running an end-of-life version as a priority.
- Disable password authentication on internet-facing SSH (`PasswordAuthentication no` , `ChallengeResponseAuthentication no` , `UsePAM no` as appropriate). Require public-key authentication and, where the operating model supports it, hardware-backed keys or certificate-based authentication.
- Restrict the advertised `KexAlgorithms`, `HostKeyAlgorithms`, `Ciphers`, and `MACs` to a curated modern allowlist (e.g. the Mozilla "modern" SSH profile or equivalent vendor guidance) and remove any algorithm not on that list.
- Where the operational model permits, place SSH access behind a VPN or jump host so that the service is not directly reachable from the public internet at all.

6.5 TLS Configuration and Weak Cryptographic Primitives

Externally-facing HTTPS services were enumerated and their TLS configurations characterised using `ssllscan` and `testssl.sh`. Two related but distinct concerns were identified: continued support for deprecated TLS protocol versions on a single endpoint, and continued advertisement of weak cryptographic primitives across a broader set of services.

Deprecated TLS Versions

One internet-facing HTTPS endpoint was found to support TLS 1.0 and TLS 1.1 in addition to TLS 1.2. Both TLS 1.0 and TLS 1.1 have been formally deprecated by the IETF (RFC 8996, March 2021) and have been removed from default support in all major browsers and operating systems. Their continued availability permits protocol downgrade attacks against any client that still negotiates them, and exposes that traffic to a long catalogue of well-known cryptographic and protocol weaknesses (BEAST, POODLE, weakened MAC constructions, lack of authenticated encryption modes). Even where current well-behaved clients negotiate TLS 1.2 or 1.3 by preference, a server that retains 1.0 and 1.1 in its supported list represents an attacker-influenceable downgrade target and should be remediated.

Weak Cipher Suites and Primitives

A separate set of services advertise cipher suites and cryptographic primitives that, while not strictly broken, fall below the minimum profile recommended by current vendor and standards-body guidance. Items observed across the affected hosts include CBC-mode cipher suites retained for legacy compatibility, RSA key exchange suites that do not provide forward secrecy, MAC constructions based on truncated SHA-1, and Diffie-Hellman parameters of insufficient size. Individually each item represents a small reduction in margin against known attacks; collectively they indicate that TLS configuration has not been refreshed against modern guidance.

Affected Hosts

IP	PORT	TLS VERSIONS ENABLED	LEGACY / WEAK ELEMENTS	NOTES
203.0.113.242	443	TLS 1.0, 1.1, 1.2	TLS 1.0/1.1 enabled; CBC/SHA1 suites	Retains deprecated protocol versions - downgrade target. Disable 1.0/1.1 and require TLS 1.2 minimum.
203.0.113.254	4443	TLS 1.2	CBC-mode suites; RSA key exchange (no PFS); SHA-1 MAC; DH params < 2048	Modern protocol only but cipher list below current vendor guidance. Tighten to forward-secret AEAD suites.

Notes

It is worth noting that several other HTTPS endpoints in the same address ranges were observed to be configured to a current standard - TLS 1.3 only or 1.2/1.3 with hardened cipher suites and forward secrecy - demonstrating that the capability and intent to deploy a hardened configuration exists within the environment. The findings above represent specific services that have not yet been brought to that standard, rather than a systemic gap.

Recommendations

- On the affected endpoint, disable TLS 1.0 and TLS 1.1 in the service's TLS configuration. Require TLS 1.2 as a minimum, and prefer TLS 1.3 wherever the application stack supports it.
- Across all externally-facing HTTPS services, restrict the advertised cipher suites to a curated modern allowlist (e.g. the Mozilla "intermediate" or "modern" TLS profile, or equivalent vendor guidance).

Remove any suite that is not forward-secret, any CBC-mode suite, and any suite based on RSA key exchange.

- Replace Diffie-Hellman parameters smaller than 2048 bits with 2048-bit or larger values, and prefer ECDHE over finite-field DH where supported by the platform.
- Re-test the affected services with `ssllscan` or equivalent after remediation and confirm that only TLS 1.2 / 1.3 and the curated cipher list are advertised.

6.6 HTTP Security Headers

Externally-facing web services were assessed for the presence and configuration of standard browser-enforced security headers. The expectation is that any modern internet-facing web property advertises a complete and correctly configured set of headers that constrain mixed-content loading, prevent framing, declare a content-security policy, and enforce HTTPS via HSTS. The observed configuration is incomplete across multiple in-scope services.

Headers Assessed

For each in-scope HTTP service, response headers were captured for the root resource and for representative authenticated paths. Headers assessed include `Strict-Transport-Security`, `Content-Security-Policy`, `X-Frame-Options`, `X-Content-Type-Options`, `Referrer-Policy`, `Permissions-Policy`, and the cookie security flags `Secure`, `HttpOnly`, and `SameSite`.

Findings

Across the assessed services, header coverage falls into three groups:

Comprehensive

A subset of services advertise the full set of recommended headers with values appropriate to their function - HSTS with a non-trivial `max-age` and `includeSubDomains`, a restrictive Content-Security-Policy, frame-ancestors restrictions, and correct cookie flags. These services represent the intended configuration baseline.

Partial

A larger group of services advertise some headers but omit others, advertise values that do not match the protective intent of the header (for example HSTS with `max-age=0` or with no `includeSubDomains`; an X-Frame-Options value of `SAMEORIGIN` on a service that should be `DENY`), or apply headers inconsistently across paths within the same service.

Absent

A smaller group of services advertise none of the recommended headers and rely entirely on browser-default behaviour. These services are the highest priority for remediation since they offer no defence against clickjacking, mixed-content downgrade, MIME-type sniffing attacks, or referrer leakage.

Affected Hosts

IP	PORT	SERVICE	SECURITY HEADER STATUS	NOTES
203.0.113.254	4443	HTTPS (Custom)	Comprehensive	HSTS, CSP, XFO, XCTO, XSS Protection present
203.0.113.243	8443	Tomcat Manager	Partial	HSTS only; missing CSP and X-Frame-Options

IP	PORT	SERVICE	SECURITY HEADER STATUS	NOTES
203.0.113.250	443	CrushFTP Web Admin	Partial	HSTS only; missing CSP and Permissions-Policy
203.0.113.194	443	GlobalProtect	None	Administrative interface exposed
203.0.113.231	443	GlobalProtect	None	No clickjacking or XSS protections
203.0.113.210	443	GlobalProtect	None	Same configuration across gateways
203.0.113.164	443	GlobalProtect	None	No header hardening
203.0.113.134	443	GlobalProtect	None	No header hardening
203.0.113.242	443	IBM Aspera Faspex	None	File transfer service; version disclosed
203.0.113.206	9001	Portainer Agent	None	Internet-facing container management
203.0.113.223	443	Resilio Sync Web	None	Sync management interface exposed
203.0.113.198	443	HTTPS	None	Generic web service; relies on browser defaults

Notes

Missing headers do not on their own constitute an exploitable vulnerability and are correctly classified as Low severity. Their value is in defence in depth: a correctly configured set of headers materially reduces the impact of cross-site scripting, clickjacking, MIME-type-confusion, and protocol-downgrade attacks should any related vulnerability be introduced in the future. The cost of remediation is low (a configuration change at the web server, reverse proxy, or application framework level), the operational risk is minimal, and the resulting hardening is broad.

Recommendations

- Deploy a baseline security-header set on every externally-facing HTTP service, applied at the reverse proxy or ingress layer so that individual application teams cannot inadvertently regress the configuration. The Mozilla Observatory and OWASP Secure Headers Project both publish concrete recommended values that map directly to the headers above.
- Set `Strict-Transport-Security: max-age=31536000; includeSubDomains` at minimum on all HTTPS endpoints, with `preload` added once the value has been stable in production.
- Define a Content-Security-Policy per service rather than copying a permissive baseline; start in `report-only` mode for any application with substantial dynamic content and tighten progressively.
- Audit cookie issuance across all authenticated endpoints and ensure `Secure`, `HttpOnly`, and a deliberate `SameSite` value are set on every session cookie.

6.7 VNC Remote Desktop Exposure

An internet-facing host within the in-scope ranges was identified as exposing a Virtual Network Computing (VNC) service. VNC is a graphical remote-desktop protocol that has historically been deployed without authentication or with weak default authentication, and is widely flagged as a high-value target by both opportunistic scanners and credential-brute-force tooling.

Service

VNC service responding on a standard VNC TCP port. Banner negotiation identified an RFB protocol version consistent with a current upstream release. The service was reachable from arbitrary external source addresses with no perimeter restriction.

Affected Host

HOST	IP ADDRESS	PORT
host19.opennetworkfilms.example	203.0.113.226	5900

Authentication

The service required authentication during the RFB handshake; brute-force or credential-stuffing attempts were not performed during testing. The presence of authentication is a positive observation, but does not by itself justify exposing VNC to the public internet - the service has a long history of authentication-bypass and remote code execution vulnerabilities, several common deployments use a shared static password rather than per-user credentials, and the underlying RFB protocol does not natively support multi-factor authentication or modern session controls.

Notes

Even where the current VNC deployment is fully patched and configured with a strong password, exposing it to the public internet has the same structural problems as exposing SSH or RDP without restriction: the service provides interactive desktop access to the host on success, attracts continuous credential brute-forcing that obscures legitimate access logging, and creates a dependency on the secrecy of a single secret rather than on cryptographic key material or attested device identity.

Recommendations

- Remove direct internet exposure of the VNC service. Place the service behind a VPN or jump host and require authenticated access to the perimeter before any RFB connection is permitted.
- If interactive remote desktop is genuinely required from arbitrary external locations, replace VNC with a modern equivalent that supports federated identity, multi-factor authentication, and session recording (commercial remote-access platforms or a properly configured RDP gateway are common choices).
- Confirm that the VNC password in use is unique, long, and rotated; review any historical configuration that uses a shared default password and replace it.
- Where the host runs other administrative services (SSH, web admin interfaces), apply the same restriction - VNC is rarely the only administrative service exposed on a host that exposes it at all.

6.8 Outdated Client-Side jQuery Library

One or more externally-facing web applications were observed loading an outdated version of the jQuery JavaScript library. The version identified is several major releases behind current and is no longer receiving upstream security fixes for known issues.

Finding

Inspection of script references in the served HTML and verification of the loaded library version (via the global `jQuery.fn.jquery` property) confirmed the outdated release. The same version is documented as affected by several historical cross-site scripting and prototype-pollution issues that have been fixed in later releases. Whether any specific historical issue is exploitable in the deployed application depends on how jQuery is used by the surrounding code; the relevant control is to upgrade the library rather than to attempt to verify each historical issue in isolation.

Affected Hosts

HOST	IP ADDRESS	PORT
host64.opennetworkfilms.example	203.0.113.250	443
host65.opennetworkfilms.example	203.0.113.254	4443

Notes

An outdated client-side library is correctly classified as a low-severity finding in isolation - exploitation generally requires an additional surrounding condition (a reflected input, an attacker-controlled property name, a specific use of an affected API). The reason it is reported as a defensive-hygiene matter is that the cost of remediation is trivial (a version pin update and a regression check), while leaving it in place permanently retains the historical attack surface and signals to attackers and automated scanners that frontend dependency management is not actively maintained.

Recommendations

- Upgrade the loaded jQuery release to the current 3.x series (or remove jQuery altogether if the surrounding application no longer relies on it; a meaningful proportion of modern frontend code does not).
- Add the frontend dependency set to whatever software-composition-analysis tooling is in use for backend dependencies, so that future advisories against client-side libraries are surfaced via the same workflow as backend ones.
- Audit the served HTML for other locally-hosted third-party libraries that may be on similarly outdated versions; jQuery is rarely the only such dependency on a long-lived web property.

6.9 UDP Service Exposure

UDP scanning is intentionally limited in any external assessment - UDP responses are unreliable, port scanners cannot easily distinguish "no service" from "filtered" without sustained probing, and scanning the full UDP port range against a sizeable address block is operationally noisy. The findings below reflect targeted UDP probing of well-known service ports across the in-scope ranges, rather than an exhaustive enumeration.

Services Identified

UDP probing identified responses on a small number of well-known service ports across the in-scope estate. Services observed include DNS (UDP/53) on hosts that appear to be operating as authoritative or caching resolvers, NTP (UDP/123) on hosts providing time synchronisation, IKE (UDP/500) and NAT-Traversal (UDP/4500) on hosts operating as IPsec VPN endpoints, and SNMP (UDP/161) on a single host. No unsolicited UDP services were observed responding on non-standard ports.

Findings

The DNS, NTP, and IPsec services responded with banners and behaviour consistent with intentional, hardened deployments - DNS resolvers refused recursive queries from external sources, NTP responded only to standard time queries and not to monlist or version queries, and IKE responded with a vendor-neutral negotiation profile that did not expose unnecessary aggressive-mode behaviour or weak transform sets. None of these services represent material risk in their current configuration; the observation is recorded primarily to confirm that they were enumerated and assessed.

The SNMP service represents a marginal exposure. SNMP responded with a default community string of `public`, granting read-only access to a limited subset of MIB tables. The information disclosed is non-sensitive and consists of standard system-identifier OIDs (uptime, hostname, OS string), but the presence of an unrestricted default community on an internet-reachable host suggests that broader SNMP configuration review is warranted.

Recommendations

- Disable SNMP on internet-facing interfaces, or - if internal SNMP polling against an externally-reachable interface is genuinely required - restrict source addresses at the network or service layer and replace the default community string with a non-default value. Migrate to SNMPv3 with authentication and encryption where the management platform supports it.
- Confirm that NTP responses cannot be amplified - verify that `monlist` is disabled and that the service does not respond with a disproportionately large response to small queries (the historical NTP-amplification DDoS pattern).
- Confirm that DNS resolvers do not recursively answer queries for arbitrary external domains from external clients (open resolver configuration), and that response-rate-limiting is enabled on authoritative deployments.

6.10 Palo Alto GlobalProtect VPN Portal Exposure

One internet-facing host within the in-scope ranges was identified as operating a Palo Alto Networks GlobalProtect portal and gateway. GlobalProtect is the user-facing VPN component of the Palo Alto firewall product family and is, by deployment model, expected to be reachable from arbitrary external addresses so that remote users can establish VPN sessions. Its inclusion here is not a finding of misconfiguration but a structural observation about attack surface.

Product

Palo Alto Networks GlobalProtect portal and gateway, served over HTTPS on TCP/443. The deployed version, as identified by portal response fingerprinting, is consistent with a current major release receiving active vendor security maintenance.

Protocol and Authentication

The portal presented a standard authentication interface and did not appear to permit anonymous enumeration of users or groups. Pre-authentication paths (the portal landing page, the configuration retrieval endpoint, the SAML/SSO redirect handler) responded with versions and behaviour consistent with the deployed major release. No active exploitation of any historical pre-authentication vulnerability was attempted during testing.

Notes

GlobalProtect has been the subject of several high-impact pre-authentication vulnerabilities in recent years (CVE-2024-3400 being the most prominent), each of which has been weaponised at scale within hours of public disclosure. The product is correctly identified by every external attacker reconnaissance toolkit as a high-priority target. The current deployment appears patched against publicly known issues, but its internet exposure means that any future disclosure of a similar pre-authentication issue will be exploitable against this host immediately on disclosure, before any defender response is possible.

This is not a defect of the GlobalProtect product or of the current deployment - it is a structural property of operating any internet-reachable VPN portal. The risk is reduced, not eliminated, by hardening;

mitigations focus on minimising the window of exposure to a future zero-day rather than on changing the present configuration.

Recommendations

- Subscribe to Palo Alto Networks' security advisory feed and ensure that GlobalProtect-affecting advisories trigger an out-of-band patch workflow rather than the standard maintenance window - the historical pattern is that exploitation begins within hours of disclosure.
- Where the user population permits, restrict GlobalProtect portal reachability by source geography or by an upstream allowlist (for example, by placing the portal behind a CDN or DDoS-protection service that performs source-address filtering before traffic reaches the firewall).
- Enforce multi-factor authentication for all VPN logins; do not permit single-factor password-based VPN authentication regardless of password strength policy.
- Confirm that VPN session telemetry (login attempts, source addresses, failed authentications) is forwarded to the central log platform with retention sufficient to support post-incident investigation.

6.11 Resilio Sync Service Exposure

One internet-facing host within the in-scope ranges was identified as exposing the Resilio Sync peer-to-peer file synchronisation service on its default management port (TCP/3839). Resilio Sync (formerly BitTorrent Sync) is a commercial product that uses BitTorrent-style peer discovery and transfer for file replication between trusted endpoints, and is sometimes deployed in production for distributing large media or build artefacts between sites. Its inclusion here is recorded as a high-value service exposure rather than an exploited finding.

Service

Resilio Sync, identified by the response of its web management interface on TCP/3839 over HTTPS. The product version, as identified by interface fingerprinting, is consistent with a current vendor release. The service was reachable from arbitrary external source addresses with no perimeter restriction.

Authentication

The web management interface required authentication and did not return identifying information from unauthenticated paths. Folder-level peer discovery (the data-plane component of the protocol) was not actively probed during testing in order to avoid generating sync traffic against operational data sets.

Notes

Exposing the Resilio Sync management interface to the public internet has two structural concerns independent of the present configuration. First, the management interface authenticates with a username and password and provides full administrative control over which folders are synchronised, with which peers, and to where - an attacker with valid credentials can read or write the entire content of any synced folder. Second, the underlying protocol assumes a trust model based on shared "secrets" embedded in folder identifiers; if any such secret is leaked or guessed, an attacker can join the peer set and exfiltrate or inject content without further authentication. Neither concern represents an immediate vulnerability in the deployed software, but both represent material risk that does not exist if the management interface is not internet-reachable.

Recommendations

- Remove direct internet exposure of the Resilio Sync management interface. Restrict TCP/3839 to administrative source addresses or place the management interface behind a VPN.

- Audit the set of folders currently being synchronised by the host and the secrets associated with each. Treat any secret that has been transmitted via a less-secure channel (email, chat, ticket attachment) as compromised and rotate it.
- Where Resilio Sync is being used to distribute production artefacts between sites, consider whether a centrally-controlled distribution mechanism (object storage with signed URLs, a content delivery network, or a pull-based artefact registry) better matches the organisation's audit and access-control requirements.

6.12 Portainer Container Management Exposure

One internet-facing host within the in-scope ranges was identified as exposing a Portainer instance and an associated Portainer Agent. Portainer is a web-based management interface for Docker, Docker Swarm, and Kubernetes that provides administrative control over containers, images, networks, and volumes. Its inclusion here is a high-value service exposure observation rather than an exploited finding.

Product

Portainer Community Edition, served over HTTPS, with an associated Portainer Agent reachable on its standard agent port. The deployed Portainer release identified during fingerprinting is consistent with a current major version receiving active vendor maintenance; the agent version is consistent with the same release.

Authentication

The Portainer web interface required authentication and did not return identifying information from unauthenticated paths. The Portainer Agent endpoint required a pre-shared key and did not respond to unauthenticated requests with anything beyond a transport-layer error. No exploitation of either component was attempted during testing.

Notes

Portainer is, by design, a privileged administrative interface - an authenticated user can create, modify, and destroy containers; mount arbitrary paths from the host filesystem into containers; execute arbitrary commands inside running containers; and (depending on configuration) modify the Docker daemon or Kubernetes cluster state. Compromise of a Portainer credential is therefore equivalent to compromise of every container managed by the instance and, in many deployments, of the underlying host.

The agent component represents a separate exposure. The Portainer Agent is the channel by which the central Portainer instance issues commands to managed nodes; if the agent's pre-shared key is leaked, an attacker can issue arbitrary container-management commands to the affected host directly, bypassing the Portainer UI entirely. Both components have historically been the subject of authentication-bypass and authorisation-bypass advisories; current builds are patched, but future advisories of similar severity are reasonably foreseeable.

Independently of either of these concerns, exposing a container-management interface to the public internet substantially elevates the value of any single credential compromise, brute-force opportunity, or session-hijacking primitive. The standard deployment model places this category of tool entirely behind a VPN or jump host.

Recommendations

- Remove direct internet exposure of both the Portainer web interface and the Portainer Agent. Place administrative access behind a VPN or jump host and restrict the agent endpoint to the central Portainer instance's source address only.

- Enforce multi-factor authentication for all Portainer accounts and disable the Portainer "internal" authentication backend in favour of an external identity provider where the deployment supports it.
- Rotate the Portainer Agent pre-shared key on a defined cadence and on any suspected compromise; treat the key as a high-value secret comparable to a Docker daemon socket.
- Subscribe to Portainer security advisories and patch out-of-band on any advisory affecting the agent or web interface, given the privileged scope of any successful exploitation.

6.13 CrushFTP File Transfer Server Exposure

One internet-facing host within the in-scope ranges was identified as operating a CrushFTP managed file transfer server. CrushFTP is a commercial enterprise file-transfer product that supports SFTP, FTPS, HTTPS, and a web-based administrative interface. Like the other entries in this section, its inclusion is recorded as a high-value service exposure rather than an actively exploited finding.

Product

CrushFTP server, identified by HTTP response fingerprinting on the management interface and by SFTP banner inspection. The deployed version is consistent with a current vendor release that has received the patches addressing recent advisories. Multiple file-transfer protocols (SFTP, FTPS, HTTPS) were observed responding on their respective standard ports.

Authentication

All file-transfer protocols required authentication and did not respond to anonymous logins. The administrative web interface required authentication and did not disclose product internals from unauthenticated paths. No active exploitation of any historical vulnerability was attempted during testing.

Notes

The managed file transfer product category - CrushFTP, MOVEit, GoAnywhere, Cleo Harmony, and equivalents - has been the focus of sustained adversary attention in recent years. Several products in this category have been the subject of pre-authentication remote code execution vulnerabilities exploited at scale within hours of disclosure (and in some cases before disclosure), with confirmed mass-exploitation campaigns targeting the financial, healthcare, and media sectors. CrushFTP itself was the subject of CVE-2024-4040, a pre-authentication path-traversal vulnerability that was added to the CISA Known Exploited Vulnerabilities catalogue within days of disclosure.

The current CrushFTP deployment appears patched and is not actively exploitable through any publicly known issue at the time of testing. The structural concern is identical to that recorded for GlobalProtect above: any future pre-authentication vulnerability in the product will be exploitable against this host immediately on disclosure, before any defender response is possible. The product's role - handling and storing customer file-transfer payloads - also means that compromise carries an unusually high data-loss impact relative to its operational profile.

Recommendations

- Subscribe to the CrushFTP vendor advisory feed and ensure that CrushFTP-affecting advisories trigger an out-of-band patch workflow rather than the standard maintenance window.
- Where the user population permits, restrict the administrative web interface to administrative source addresses only; the file-transfer protocols themselves can remain reachable for the user population while the management plane is restricted.

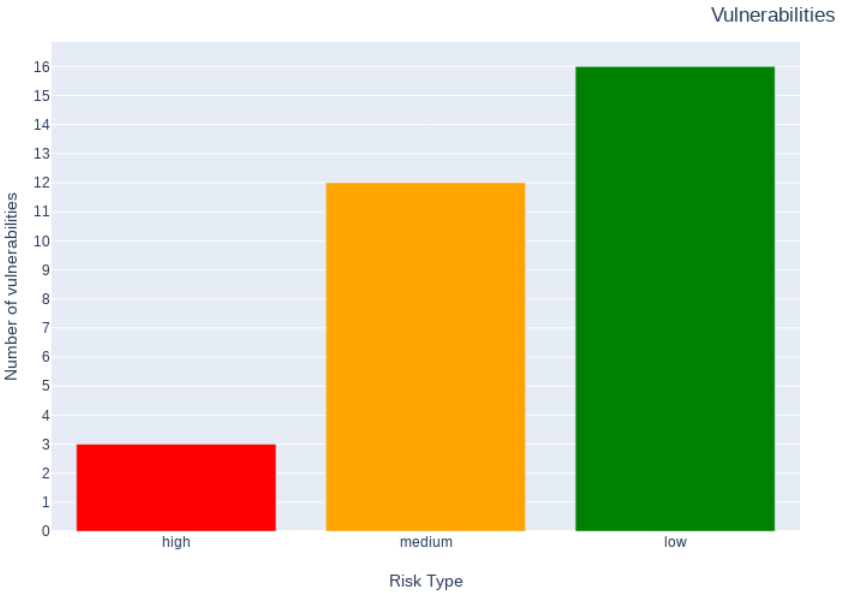
- Enforce multi-factor authentication for all administrative and end-user accounts. Where the file-transfer relationship is machine-to-machine, prefer SSH key authentication over password authentication on the SFTP protocol.
- Confirm that file-transfer telemetry (login attempts, file uploads and downloads, administrative actions) is forwarded to the central log platform with sufficient retention to support post-incident investigation, and that abnormal-volume or abnormal-source-address activity triggers an alert.
- Consider whether the file-transfer relationships served by this host could be migrated to a cloud-hosted equivalent (managed SFTP, signed-URL object storage with short-lived credentials) that shifts the patching and edge-protection responsibility to a service provider.

7. Vulnerability Summary

RISK	VULNERABILITY	HOSTNAME	IP ADDRESS	PORT
9.1	Exposed Apache Tomcat Manager Console with Default Credentials	host33.opennetworkfilms.example	203.0.113.243	8080
7.8	Linux Kernel AF_ALG "Copy Fail" Local Privilege Escalation	host32.opennetworkfilms.example	203.0.113.242	22
7.5	Anonymous SMB Share Enumeration on Externally-Facing Host	host01.opennetworkfilms.example	203.0.113.198	445
6.5	Outdated and Insecure SSH Configuration	host32.opennetworkfilms.example	203.0.113.242	22
6.5	Outdated and Insecure SSH Configuration	host33.opennetworkfilms.example	203.0.113.243	22
6.5	Outdated and Insecure SSH Configuration	host33.opennetworkfilms.example	203.0.113.243	33001
6.5	Outdated and Insecure SSH Configuration	host53.opennetworkfilms.example	203.0.113.193	22
6.5	Outdated and Insecure SSH Configuration	host55.opennetworkfilms.example	203.0.113.195	22
6.5	Outdated and Insecure SSH Configuration	host56.opennetworkfilms.example	203.0.113.196	22
6.5	Outdated and Insecure SSH Configuration	host57.opennetworkfilms.example	203.0.113.197	22
6.5	Outdated and Insecure SSH Configuration	host58.opennetworkfilms.example	203.0.113.199	33001
6.5	Outdated and Insecure SSH Configuration	host59.opennetworkfilms.example	203.0.113.206	22
6.5	Outdated and Insecure SSH Configuration	host61.opennetworkfilms.example	203.0.113.223	33001
6.5	Outdated and Insecure SSH Configuration	host64.opennetworkfilms.example	203.0.113.250	22

3.7	TLS 1.0 / 1.1 Still Enabled on Externally-Facing Service	host32.opennetworkfilms.example	203.0.113.242	443
3.1	Weak Encryption Methods Supported	host65.opennetworkfilms.example	203.0.113.254	4443
3.1	Missing Security Headers	host32.opennetworkfilms.example	203.0.113.242	443
3.1	Missing Security Headers	host33.opennetworkfilms.example	203.0.113.243	8443
3.1	Missing Security Headers	host54.opennetworkfilms.example	203.0.113.194	443
3.1	Missing Security Headers	host59.opennetworkfilms.example	203.0.113.206	9001
3.1	Missing Security Headers	host61.opennetworkfilms.example	203.0.113.223	443
3.1	Missing Security Headers	host63.opennetworkfilms.example	203.0.113.231	443
3.1	Missing Security Headers	host64.opennetworkfilms.example	203.0.113.250	443
3.1	Missing Security Headers	host65.opennetworkfilms.example	203.0.113.254	4443
3.1	Missing Security Headers	host66.opennetworkfilms.example	203.0.113.210	443
3.1	Missing Security Headers	host67.opennetworkfilms.example	203.0.113.198	443
3.1	Missing Security Headers	host68.opennetworkfilms.example	203.0.113.164	443
3.1	Missing Security Headers	host69.opennetworkfilms.example	203.0.113.134	443
3.1	VNC Service Exposure	host19.opennetworkfilms.example	203.0.113.226	5900
2.6	JQuery Outdated	host64.opennetworkfilms.example	203.0.113.250	443
2.6	JQuery Outdated	host65.opennetworkfilms.example	203.0.113.254	4443

8. Visual Summary



9. Vulnerability Details

9.1 Exposed Apache Tomcat Manager Console with Default Credentials

CVSS:9.1

DESCRIPTION

Host `host33.opennetworkfilms.example` exposes the Apache Tomcat Manager application to the public internet. The Manager interface accepts the default credential pair `tomcat / tomcat`, granting authenticated access to the WAR file deployment endpoint. Deploying a malicious WAR results in immediate remote code execution as the Tomcat service account.

HOW TO REPRODUCE

1. Browse to `http://host33.opennetworkfilms.example:8080/manager/html`.
2. Authenticate as `tomcat / tomcat`.
3. Upload a JSP webshell packaged as a WAR via the Deploy form.
4. Request the deployed context; an interactive shell is returned as the Tomcat service user.

HOST AFFECTED

IP ADDRESS	HOSTNAME	PORT
203.0.113.243	host33.opennetworkfilms.example	8080

RECOMMENDATIONS

Disable the Manager application from any internet-facing interface. If Manager access is genuinely required, restrict it to administrative source IP ranges via a reverse proxy or firewall, rotate all default credentials, and enforce strong, unique passwords. Disable the host-manager application entirely if unused.

REFERENCES

Apache Tomcat Manager How-To - <https://tomcat.apache.org/tomcat-9.0-doc/manager-howto.html>

CWE-798 - Use of Hard-coded Credentials - <https://cwe.mitre.org/data/definitions/798.html>

OWASP A07:2021 - Identification and Authentication Failures - https://owasp.org/Top10/A07_2021-Identification_and_Authentication_Failures/

9.2 Linux Kernel AF_ALG "Copy Fail" Local Privilege Escalation

CVSS:7.8

DESCRIPTION

A logic flaw in the Linux kernel's `algif_aead` module (`AF_ALG` userspace crypto API) allows a local unprivileged user to perform a controlled 4-byte write into the kernel's page cache of any readable file. The flaw stems from an in-place memory optimization introduced in 2017 where the kernel reuses source memory as the destination during cryptographic operations. By abusing the interaction between `AF_ALG` sockets and `splice()`, an attacker can corrupt privileged binaries such as `/usr/bin/su` in the

page cache without modifying them on disk, escalating to UID 0. Because the page cache is shared across containers and the host, this also enables container-to-host escape.

Confirmed on host `host32.opennetworkfilms.example`, which runs Ubuntu 24.04 LTS with an unpatched kernel.

HOW TO REPRODUCE

1. Obtain a low-privilege shell on `host32.opennetworkfilms.example` via the existing weak SSH configuration finding.
2. Drop the public ~732-byte Copy Fail proof-of-concept and execute it as the unprivileged user.
3. The PoC opens an `AF_ALG` socket bound to `algif_aead`, `splice()` s data into it, and triggers the in-place encryption path against the page cache of `/usr/bin/su`.
4. After ~1 second, run `/usr/bin/su`; the corrupted `setuid` path returns a root shell.
5. From inside a container on the same host, the same PoC succeeds because the page cache is shared with the host, demonstrating container escape.

HOST AFFECTED

IP ADDRESS	HOSTNAME	PORT
203.0.113.242	host32.opennetworkfilms.example	22

RECOMMENDATIONS

Apply vendor kernel patches for CVE-2026-31431 immediately. Distribution advisories are available from Canonical, Red Hat, SUSE, Amazon Linux, Debian, Fedora, and Arch. If patching cannot be performed at once: block `AF_ALG` socket creation via `seccomp` or by removing the `algif_aead` module, and isolate Linux servers from untrusted local users. Treat any container that previously shared a host with an untrusted workload as suspect.

REFERENCES

NVD - CVE-2026-31431 - <https://nvd.nist.gov/vuln/detail/CVE-2026-31431>

Microsoft Security Blog - Copy Fail - <https://www.microsoft.com/en-us/security/blog/2026/05/01/cve-2026-31431-copy-fail-vulnerability-enables-linux-root-privilege-escalation/>

Ubuntu Security Notice - <https://ubuntu.com/security/CVE-2026-31431>

9.3 Anonymous SMB Share Enumeration on Externally-Facing Host

CVSS:7.5

DESCRIPTION

Host `host01.opennetworkfilms.example` exposes SMB (TCP 445) to the public internet and permits anonymous (null-session) share enumeration. An unauthenticated attacker can list available shares and, depending on share permissions, may be able to read or write file contents. SMB on the public internet additionally exposes the host to a long history of remote code execution vulnerabilities (e.g. EternalBlue) and credential-relay attacks.

HOW TO REPRODUCE

1. From an external workstation, run: `smbclient -L //host01.opennetworkfilms.example -N`

2. The command returns a populated share list without prompting for credentials, confirming that null sessions are accepted.

HOST AFFECTED

IP ADDRESS	HOSTNAME	PORT
203.0.113.198	host01.opennetworkfilms.example	445

RECOMMENDATIONS

Block TCP 445 (and 139) at the network edge - SMB should never be reachable from the public internet. If internal SMB is required, disable null sessions via `RestrictAnonymous` / `RestrictAnonymousSAM`, require SMB signing, and enforce SMBv3 with encryption.

REFERENCES

Microsoft - SMB Security Best Practices - <https://learn.microsoft.com/en-us/windows-server/storage/file-server/smb-security>

CISA Alert - SMB Security Best Practices - <https://www.cisa.gov/news-events/alerts/2017/01/16/smb-security-best-practices>

CWE-200 - Exposure of Sensitive Information - <https://cwe.mitre.org/data/definitions/200.html>

9.4 Outdated and Insecure SSH Configuration

CVSS:6.5

DESCRIPTION

Several internet-facing hosts were identified running SSH services with outdated or weak configurations that expose them to known, publicly documented vulnerabilities affecting OpenSSH and SSH protocol implementations. Enumeration revealed support for legacy authentication mechanisms and password-based authentication, which are associated with multiple historical and still-relevant CVEs impacting confidentiality and access control.

Although no active exploitation was performed, the exposed configurations materially increase the likelihood of a successful attack using well-known techniques documented in public advisories.

HOW TO REPRODUCE

```
nmap -p <port> --script ssh-auth-methods <host>
```

HOST AFFECTED

IP ADDRESS	HOSTNAME	PORT
203.0.113.242	host32.opennetworkfilms.example	22
203.0.113.243	host33.opennetworkfilms.example	22
203.0.113.243	host33.opennetworkfilms.example	33001
203.0.113.193	host53.opennetworkfilms.example	22
203.0.113.195	host55.opennetworkfilms.example	22
203.0.113.196	host56.opennetworkfilms.example	22

203.0.113.197	host57.opennetworkfilms.example	22
203.0.113.199	host58.opennetworkfilms.example	33001
203.0.113.206	host59.opennetworkfilms.example	22
203.0.113.223	host61.opennetworkfilms.example	33001
203.0.113.250	host64.opennetworkfilms.example	22

RECOMMENDATIONS

- **Upgrade OpenSSH** to the latest version to address known vulnerabilities and enhance security. Ensure the update supports stronger cryptographic algorithms like ed25519 and ecdsa.
- **Restrict SSH Access** by limiting the SSH port (22) to trusted IP addresses using VPN or firewall rules. Consider changing the SSH port to a non-standard one to reduce exposure to automated attacks.
- **Enable Strong Authentication** by disabling password-based authentication in favor of key-based methods. Implement multi-factor authentication (MFA) for additional security.
- **Harden SSH Configuration** by disabling outdated protocols (e.g., SSH v1) and weak algorithms (e.g., RSA < 2048 bits, DSA). Set PermitRootLogin no and MaxAuthTries 3 in /etc/ssh/sshd_config to prevent unauthorized root access and reduce brute-force attempts.

REFERENCES

OpenSSH User Enumeration Vulnerability (CVE-2018-15473) - <https://nvd.nist.gov/vuln/detail/CVE-2018-15473>

OpenSSH Roaming Information Leak (CVE-2016-0777 / CVE-2016-0778) - <https://nvd.nist.gov/vuln/detail/CVE-2016-0777>

OpenSSH Client Information Leak (CVE-2020-14145) - <https://nvd.nist.gov/vuln/detail/CVE-2020-14145>

9.5 TLS 1.0 / 1.1 Still Enabled on Externally-Facing Service

CVSS:3.7

DESCRIPTION

Host host32.opennetworkfilms.example continues to support TLS 1.0 and TLS 1.1 on its HTTPS service. Both protocols have been formally deprecated by the IETF (RFC 8996) and removed from support by all major browsers and operating systems. Their continued availability permits downgrade attacks and exposes clients to known cipher and protocol weaknesses (BEAST, POODLE, weakened MAC constructions).

HOW TO REPRODUCE

1. Run: `sslscan host32.opennetworkfilms.example:443`
2. The output's 'Supported Server Cipher Suites' section lists protocol versions TLSv1.0 and TLSv1.1 alongside TLSv1.2.

HOST AFFECTED

IP ADDRESS	HOSTNAME	PORT
203.0.113.242	host32.opennetworkfilms.example	443

RECOMMENDATIONS

Disable TLS 1.0 and TLS 1.1 in the service's TLS configuration. Require TLS 1.2 (or preferably 1.3) only. Remove cipher suites tied to the deprecated protocols. Re-test with `ssllscan` to confirm only TLSv1.2 / TLSv1.3 are advertised.

REFERENCES

RFC 8996 - Deprecating TLS 1.0 and TLS 1.1 - <https://datatracker.ietf.org/doc/html/rfc8996>

Mozilla TLS Configuration Guide - https://wiki.mozilla.org/Security/Server_Side_TLS

NIST SP 800-52 Rev. 2 - <https://csrc.nist.gov/publications/detail/sp/800-52/rev-2/final>

9.6 Weak Encryption Methods Supported

CVSS:3.1

DESCRIPTION

The affected service supports **deprecated and weak SSL/TLS cryptographic algorithms** that do not meet current security best practices. These algorithms provide **insufficient protection against modern cryptographic attacks** and, in some cases, are vulnerable to known weaknesses that can undermine the confidentiality and integrity of encrypted communications.

Support for weak cipher suites (such as legacy CBC-based ciphers, SHA-1-based MACs, or obsolete key exchange mechanisms) increases the risk that an attacker could **downgrade a secure connection** or exploit protocol weaknesses to decrypt or manipulate traffic. While exploitation often requires a man-in-the-middle position, the continued availability of these cipher suites represents a **systemic weakness in the TLS configuration**.

Even when stronger cipher suites are also supported, the presence of deprecated options expands the attack surface, increases exposure to downgrade attacks, and is inconsistent with current guidance from industry standards such as NIST and OWASP.

HOW TO REPRODUCE

```
nmap -p <port> --script ssl-enum-ciphers <host>
```

HOST AFFECTED

IP ADDRESS	HOSTNAME	PORT
203.0.113.254	host65.opennetworkfilms.example	4443

RECOMMENDATIONS

Disable the weak ciphers highlighted in the report.

REFERENCES

Data Encryption Standard - https://en.wikipedia.org/wiki/Data_Encryption_Standard#Brute_force_attack

NIST SP 800-52 Rev. 2 - <https://csrc.nist.gov/publications/detail/sp/800-131a/rev-2/final>

OWASP Transport Layer Protection Cheat Sheet - https://cheatsheetseries.owasp.org/cheatsheets/Transport_Layer_Protection_Cheat_Sheet.html

9.7 Missing Security Headers

CVSS:3.1

DESCRIPTION

Several industry-standard HTTP security headers are absent, reducing the application's protection against common client-side attacks. Without these headers, browsers are not instructed to enforce stricter handling of content, framing, MIME types, or transport rules. This increases exposure to risks such as clickjacking, content injection, MIME-sniffing, and weakened HTTPS enforcement.

Missing headers do not usually result in direct compromise, but they meaningfully lower the browser's defensive posture. Attackers can more easily embed the site in malicious iframes, trick the browser into interpreting content as a different MIME type, leak referrer information, or attempt client-side injection attacks that a proper CSP would limit.

HOW TO REPRODUCE

curl -I URL

Review the response headers.

HOST AFFECTED

IP ADDRESS	HOSTNAME	PORT
203.0.113.242	host32.opennetworkfilms.example	443
203.0.113.243	host33.opennetworkfilms.example	8443
203.0.113.194	host54.opennetworkfilms.example	443
203.0.113.206	host59.opennetworkfilms.example	9001
203.0.113.223	host61.opennetworkfilms.example	443
203.0.113.231	host63.opennetworkfilms.example	443
203.0.113.250	host64.opennetworkfilms.example	443
203.0.113.254	host65.opennetworkfilms.example	4443
203.0.113.210	host66.opennetworkfilms.example	443
203.0.113.198	host67.opennetworkfilms.example	443
203.0.113.164	host68.opennetworkfilms.example	443
203.0.113.134	host69.opennetworkfilms.example	443

RECOMMENDATIONS

Implement the full set of recommended security headers, including:

- Strict-Transport-Security to enforce HTTPS.
- Content-Security-Policy to restrict where scripts, images, and resources load from.
- X-Frame-Options to prevent clickjacking.
- X-Content-Type-Options: nosniff to stop MIME-sniffing.
- Referrer-Policy: strict-origin-when-cross-origin to limit information leakage.
- Permissions-Policy to restrict browser APIs (camera, mic, geolocation).

REFERENCES

OWASP – HTTP Security Headers Cheat Sheet - <https://cheatsheetseries.owasp.org/cheatsheets/HTTP-Headers-Cheat-Sheet.html>

OWASP – Secure Headers Project - <https://owasp.org/www-project-secure-headers/>

9.8 VNC Service Exposure

CVSS:3.1

DESCRIPTION

A Virtual Network Computing (VNC) service was identified as exposed on an internet-accessible host. VNC provides remote graphical desktop access and is not designed to be safely exposed to untrusted networks without strong compensating controls. When accessible externally, VNC services significantly increase the attack surface and may allow unauthorized access through weak authentication, brute-force attacks, credential reuse, or exploitation of client/server implementation flaws.

VNC protocols typically provide limited native security controls, and many deployments lack modern encryption or multi-factor authentication. Even when authentication is enabled, exposed VNC services are commonly targeted by automated scanning and credential-stuffing activity.

HOW TO REPRODUCE

```
nmap -p 5900 -sV <host>
```

HOST AFFECTED

IP ADDRESS	HOSTNAME	PORT
203.0.113.226	host19.opennetworkfilms.example	5900

RECOMMENDATIONS

Do not expose VNC services directly to the internet

- Restrict VNC access to internal networks only
- Place VNC behind a VPN or zero-trust access solution
- Enforce strong authentication and unique credentials
- Disable VNC entirely where not operationally required
- Consider replacing VNC with more secure remote administration methods designed for external access

REFERENCES

NIST SP 800-41 Rev.1 – Guidelines on Firewalls and Firewall Policy - <https://csrc.nist.gov/publications/detail/sp/800-41/rev-1/final>

9.9 JQuery Outdated

CVSS:2.6

DESCRIPTION

The application serves an **outdated version of the jQuery JavaScript library**. Older jQuery releases are affected by multiple publicly documented security issues that can introduce **client-side attack vectors**, particularly when used in conjunction with untrusted input or vulnerable third-party plugins.

While jQuery vulnerabilities rarely result in direct server-side compromise, outdated versions have been associated with **cross-site scripting (XSS) conditions, unsafe HTML parsing, and prototype manipulation issues** under specific usage patterns. The continued use of an outdated library increases exposure to these risks and may weaken the application's overall client-side security posture.

HOW TO REPRODUCE

- Browse to the affected application endpoint.
- Open the browser developer tools.
- Inspect loaded JavaScript resources and identify the jQuery version in use (e.g., via jquery.js filename or jQuery.fn.jquery in the console).
- Compare the identified version against the latest stable jQuery release.
- Confirm that the version in use is affected by known issues documented in public advisories.

HOST AFFECTED

IP ADDRESS	HOSTNAME	PORT
203.0.113.250	host64.opennetworkfilms.example	443
203.0.113.254	host65.opennetworkfilms.example	4443

RECOMMENDATIONS

Upgrade jQuery to the latest supported stable version and verify compatibility with any dependent scripts or plugins. Where feasible, remove unused libraries and ensure third-party JavaScript dependencies are regularly reviewed and maintained.

REFERENCES

jQuery Security Advisories - <https://jquery.com/upgrade-guide/>

10. Tools Used During Testing

theharvester - <https://github.com/laramies/theHarvester>

Burp Suite Pro - <https://portswigger.net/burp>

GoBuster - <https://github.com/OJ/gobuster>

Gowitness - <https://github.com/sensepost/gowitness>

dirb - <http://dirb.sourceforge.net>

nmap - <https://nmap.org>

ssllscan - <https://github.com/rbsec/ssllscan>

wireshark - <https://www.wireshark.org>

cyberchef - <https://gchq.github.io/CyberChef>