Unità 2 - S5-L3

Data: 22 Ottobre 2025

Alessandro Pietro Salerno

Report di Vulnerability Assessment - Metasploitable2

Oggetto: Report di Valutazione delle Vulnerabilità sulla macchina target Metasploitable2

1. Introduzione

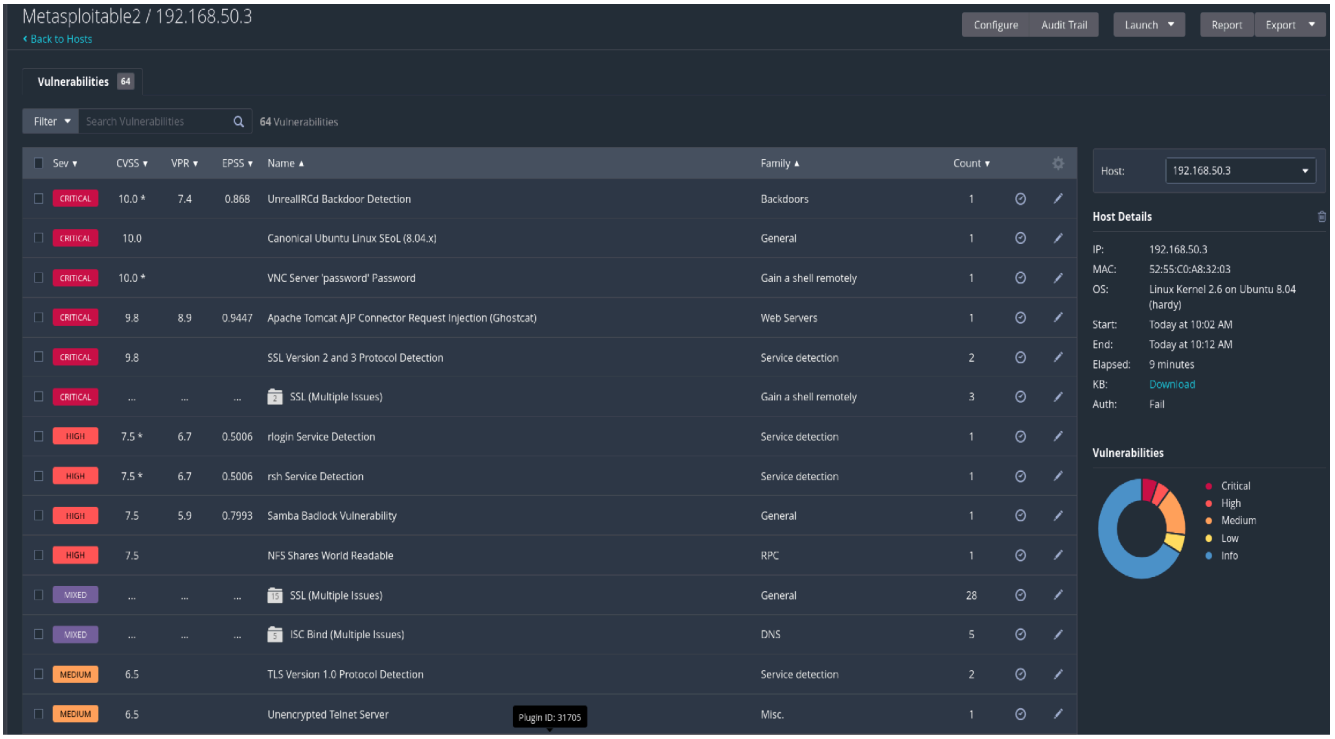
Il presente documento riporta i risultati di un'attività di Vulnerability Assessment (VA) condotta sulla macchina virtuale Metasploitable2. L'obiettivo dell'assessment era identificare le vulnerabilità note presenti sui servizi esposti su porte di rete comuni, in linea con l'esercitazione accademica svolta. L'attività è stata eseguita utilizzando lo strumento di scansione automatica Nessus (versione Essentials).

2. Metodologia

- **Strumento Utilizzato:** Tenable Nessus Essentials.
- **Tipo di Scansione:** È stata configurata una scansione di tipo "Basic Network Scan", personalizzata per focalizzarsi su un elenco specifico di porte.
- **Target:** La macchina virtuale Metasploitable2, identificata con l'indirizzo IP **192.168.50.3.**
- **Porte Scansionate:** La scansione è stata limitata alle seguenti porte TCP: **21, 22, 23, 25, 80, 110, 139, 443, 445, 3389.**
- **Tipo di Analisi:** Scansione non autenticata (black-box), simulando la prospettiva di un attaccante esterno senza credenziali di accesso al sistema.
- **Periodo di Scansione:** Mercoledì, 22 Ottobre 2025 (Inizio: 18:02:56, Fine: 18:12:00).

3. Riepilogo dei Risultati

La scansione ha identificato un numero significativo di vulnerabilità sulla macchina target 192.168.50.3. La distribuzione delle vulnerabilità rilevate per livello di severità è la seguente:



“ lo screenshot presenta altre informazioni relative la scansione , ed esclude dall’ elenco , ovviamente , tutte le vulnerabilità trovate e non riporta i livelli Low e Info , che sono comunque visibili nel diagramma in basso a destra , differenziate per colore. ”

La presenza di **9 vulnerabilità critiche e 7 alte** indica una superficie d'attacco considerevole e un rischio elevato di compromissione per l'host analizzato.

4. Analisi Dettagliata delle Vulnerabilità Principali

Di seguito viene fornita un'analisi più approfondita di due vulnerabilità più significative identificate durante la scansione sull'host 192.168.50.3.

4.1 Debian OpenSSH/OpenSSL Package Random Number Generator Weakness

- **ID Nessus:** 32314
- **Severità:** Critical
- **Punteggio CVSS v2.0:** 10.0 (AV:N/AC:L/Au:N/C:C/I:C/A:C)
- **Porta/Protocollo:** TCP/22 (SSH)

32314 - Debian OpenSSH/OpenSSL Package Random Number Generator Weakness

Synopsis

The remote SSH host keys are weak.

Description

The remote SSH host key has been generated on a Debian or Ubuntu system which contains a bug in the random number generator of its OpenSSL library.

The problem is due to a Debian packager removing nearly all sources of entropy in the remote version of OpenSSL.

An attacker can easily obtain the private part of the remote key and use this to set up decipher the remote session or set up a man in the middle attack.

See Also

http://www.nessus.org/u?107f9bdc

http://www.nessus.org/u?f14f4224

Solution

Consider all cryptographic material generated on the remote host to be guessable. In particuliar, all SSH, SSL and OpenVPN key material should be re-generated.

Risk Factor

Critical

- **Descrizione:** La chiave host SSH remota è stata generata su un sistema Debian o Ubuntu che contiene un bug nel generatore di numeri casuali della sua libreria OpenSSL. Questo bug, dovuto alla rimozione di quasi tutte le fonti di entropia da parte di un maintainer del pacchetto Debian, rende la porzione privata della chiave facilmente ottenibile. Un attaccante può sfruttare questa debolezza per decifrare le comunicazioni SSH remote o effettuare attacchi Man-in-the-Middle.

- **Impatto Potenziale:** Decifratura del traffico SSH, intercettazione di credenziali, esecuzione di comandi remoti attraverso sessioni SSH compromesse, attacchi Man-in-the-Middle.

- **Raccomandazioni di Remediation:** Tutto il materiale crittografico generato sull'host remoto deve essere considerato prevedibile e quindi insicuro. In particolare, è necessario rigenerare tutte le chiavi SSH, SSL e OpenVPN.

- **Riferimenti:** CVE-2008-0166, BID: 29179, CWE-310

Cliccando nei link sotto la voce “See also” nel report generato da Nessus , è possibile attingere a dell’ ulteriore materiale informativo riguardo la vulnerabilità in merito. Qui ad esempio è riportato il come sia stata scoperta la vulnerabilità : generatore di numeri causali prevedibile su Debian’s openssl package , e da chi : Luciano Bello (probalile alias) ed anche da cosa sia causata : cambio errato di Debian verso delle librerie openssl .

Una spiegazione essenziale è gia stata riportata alla voce “Descrizione” del report , ed ulteriori info in merito la risoluzione della vulnerabilità sono reperibili all’ altro link della sezione “see also” .

[SECURITY] [DSA 1571-1] New openssl packages fix predictable random number generator

- *To*: debian-security-announce@lists.debian.org
- *Subject*: [SECURITY] [DSA 1571-1] New openssl packages fix predictable random number generator
- *From*: Florian Weimer <fw@deneb.enyo.de>
- *Date*: Tue, 13 May 2008 14:06:39 +0200
- *Message-id*: <[🔎] 87od7az9v4.fsf@mid.deneb.enyo.de>
- *Reply-to*: debian-security@lists.debian.org

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

- ------------------------------------------------------------------------
Debian Security Advisory DSA-1571-1                 security@debian.org
http://www.debian.org/security/                        Florian Weimer
May 13, 2008                          http://www.debian.org/security/faq
- ------------------------------------------------------------------------

Package        : openssl
Vulnerability  : predictable random number generator
Problem type   : remote
Debian-specific: yes
CVE Id(s)      : CVE-2008-0166

Luciano Bello discovered that the random number generator in Debian's
openssl package is predictable.  This is caused by an incorrect
Debian-specific change to the openssl package (CVE-2008-0166).  As a
result, cryptographic key material may be guessable.

This is a Debian-specific vulnerability which does not affect other
operating systems which are not based on Debian.  However, other systems
can be indirectly affected if weak keys are imported into them.

It is strongly recommended that all cryptographic key material which has
been generated by OpenSSL versions starting with 0.9.8c-1 on Debian
systems is recreated from scratch.  Furthermore, all DSA keys ever used
on affected Debian systems for signing or authentication purposes should
be considered compromised; the Digital Signature Algorithm relies on a
secret random value used during signature generation.
```

4.2 Apache Tomcat AJP Connector Request Injection (Ghostcat)

- **ID Nessus:** 134862

- **Severità:** High
- **Punteggio CVSS v3.0:** 9.8 (AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H)
- **Porta/Protocollo:** TCP/8009 (AJP)

134862 - Apache Tomcat AJP Connector Request Injection (Ghostcat)

Synopsis

There is a vulnerable AJP connector listening on the remote host.

Description

A file read/inclusion vulnerability was found in AJP connector. A remote, unauthenticated attacker could exploit this vulnerability to read web application files from a vulnerable server. In instances where the vulnerable server allows file uploads, an attacker could upload malicious JavaServer Pages (JSP) code within a variety of file types and gain remote code execution (RCE).

See Also

http://www.nessus.org/u?8ebe6246
http://www.nessus.org/u?4e287adb
http://www.nessus.org/u?cbc3d54e
https://access.redhat.com/security/cve/CVE-2020-1745
https://access.redhat.com/solutions/4851251
http://www.nessus.org/u?dd218234
http://www.nessus.org/u?dd772531

- **Descrizione:** È stata riscontrata una vulnerabilità di tipo file read/inclusion nel connettore AJP di Apache Tomcat. Un attaccante remoto non autenticato può sfruttare questa vulnerabilità per leggere file dell'applicazione web (es. file di configurazione, codice sorgente). Nei casi in cui il server consenta l'upload di file, un attaccante potrebbe caricare codice malevolo (es. JSP) e ottenere l'esecuzione di comandi remoti (RCE).

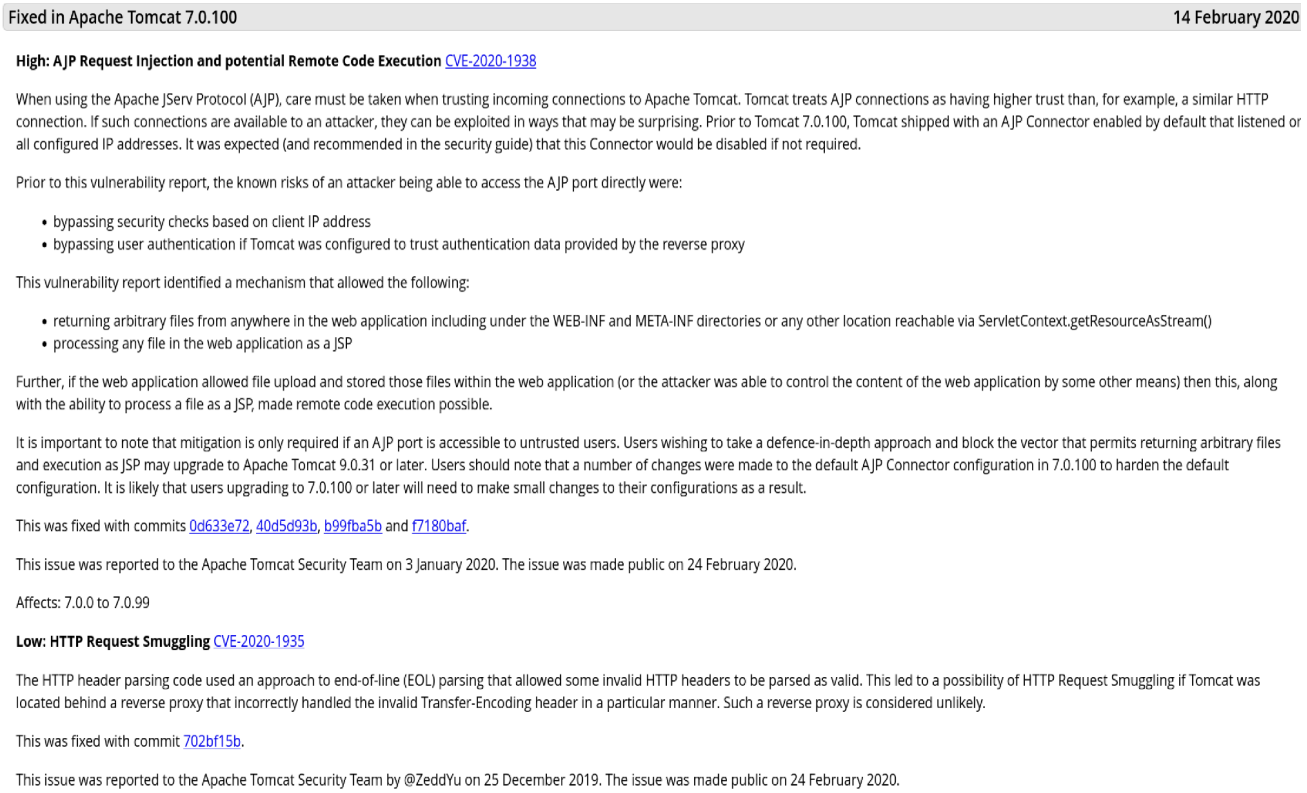
- **Impatto Potenziale:** Lettura di file sensibili dell'applicazione web, potenziale esecuzione di codice remoto sul server, compromissione dell'applicazione e del server sottostante.

- **Raccomandazioni di Remediation:** Aggiornare il server Tomcat alle versioni 7.0.100, 8.5.51, 9.0.31 o successive. In alternativa, aggiornare la

configurazione del connettore AJP per richiedere l'autenticazione. Se il connettore AJP non è necessario, disabilitarlo.

- **Riferimenti:** CVE-2020-1745, CVE-2020-1938, CISA KEV (Known Exploited Vulnerability)

I link della sezione “sse also” del report rimandno a delle spiegazioni dettagliate , sul sito Apache Tomcat , in merito la vulnerabilità in esame . (i primi link rimandano a spiegazione molto simili) . Altri link invece rimandano al sito redhat .



Fixed in Apache Tomcat 7.0.100 — 14 February 2020

High: AJP Request Injection and potential Remote Code Execution CVE-2020-1938

When using the Apache JServ Protocol (AJP), care must be taken when trusting incoming connections to Apache Tomcat. Tomcat treats AJP connections as having higher trust than, for example, a similar HTTP connection. If such connections are available to an attacker, they can be exploited in ways that may be surprising. Prior to Tomcat 7.0.100, Tomcat shipped with an AJP Connector enabled by default that listened on all configured IP addresses. It was expected (and recommended in the security guide) that this Connector would be disabled if not required.

Prior to this vulnerability report, the known risks of an attacker being able to access the AJP port directly were:

- bypassing security checks based on client IP address
- bypassing user authentication if Tomcat was configured to trust authentication data provided by the reverse proxy

This vulnerability report identified a mechanism that allowed the following:

- returning arbitrary files from anywhere in the web application including under the WEB-INF and META-INF directories or any other location reachable via ServletContext.getResourceAsStream()
- processing any file in the web application as a JSP

Further, if the web application allowed file upload and stored those files within the web application (or the attacker was able to control the content of the web application by some other means) then this, along with the ability to process a file as a JSP, made remote code execution possible.

It is important to note that mitigation is only required if an AJP port is accessible to untrusted users. Users wishing to take a defence-in-depth approach and block the vector that permits returning arbitrary files and execution as JSP may upgrade to Apache Tomcat 9.0.31 or later. Users should note that a number of changes were made to the default AJP Connector configuration in 7.0.100 to harden the default configuration. It is likely that users upgrading to 7.0.100 or later will need to make small changes to their configurations as a result.

This was fixed with commits 0d633e72, 40d5d93b, b99fba5b and f7180baf.

This issue was reported to the Apache Tomcat Security Team on 3 January 2020. The issue was made public on 24 February 2020.

Affects: 7.0.0 to 7.0.99

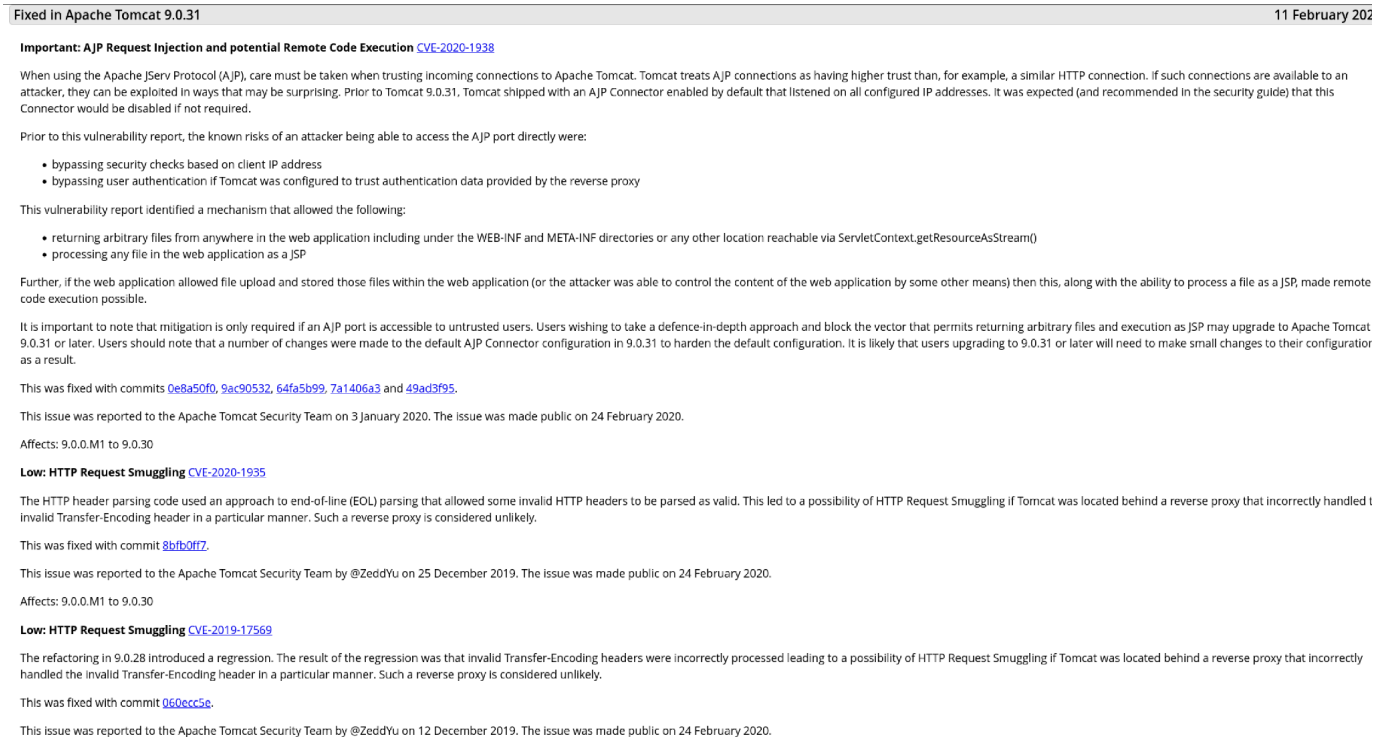
Low: HTTP Request Smuggling CVE-2020-1935

The HTTP header parsing code used an approach to end-of-line (EOL) parsing that allowed some invalid HTTP headers to be parsed as valid. This led to a possibility of HTTP Request Smuggling if Tomcat was located behind a reverse proxy that incorrectly handled the invalid Transfer-Encoding header in a particular manner. Such a reverse proxy is considered unlikely.

This was fixed with commit 702bf15b.

This issue was reported to the Apache Tomcat Security Team by @ZeddYu on 25 December 2019. The issue was made public on 24 February 2020.

“questi sono 2 screeshot relativamente del primo e del terzo link . Le spiegazioni quasi identiche. “



Fixed in Apache Tomcat 9.0.31 11 February 202

Important: AJP Request Injection and potential Remote Code Execution CVE-2020-1938

When using the Apache JServ Protocol (AJP), care must be taken when trusting incoming connections to Apache Tomcat. Tomcat treats AJP connections as having higher trust than, for example, a similar HTTP connection. If such connections are available to an attacker, they can be exploited in ways that may be surprising. Prior to Tomcat 9.0.31, Tomcat shipped with an AJP Connector enabled by default that listened on all configured IP addresses. It was expected (and recommended in the security guide) that this Connector would be disabled if not required.

Prior to this vulnerability report, the known risks of an attacker being able to access the AJP port directly were:

- bypassing security checks based on client IP address
- bypassing user authentication if Tomcat was configured to trust authentication data provided by the reverse proxy

This vulnerability report identified a mechanism that allowed the following:

- returning arbitrary files from anywhere in the web application including under the WEB-INF and META-INF directories or any other location reachable via ServletContext.getResourceAsStream()
- processing any file in the web application as a JSP

Further, if the web application allowed file upload and stored those files within the web application (or the attacker was able to control the content of the web application by some other means) then this, along with the ability to process a file as a JSP, made remote code execution possible.

It is important to note that mitigation is only required if an AJP port is accessible to untrusted users. Users wishing to take a defence-in-depth approach and block the vector that permits returning arbitrary files and execution as JSP may upgrade to Apache Tomcat 9.0.31 or later. Users should note that a number of changes were made to the default AJP Connector configuration in 9.0.31 to harden the default configuration. It is likely that users upgrading to 9.0.31 or later will need to make small changes to their configuration as a result.

This was fixed with commits 0e8a50f0, 9ac90532, 64fa5b99, 7a1406a3 and 49ad3f95.

This issue was reported to the Apache Tomcat Security Team on 3 January 2020. The issue was made public on 24 February 2020.

Affects: 9.0.0.M1 to 9.0.30

Low: HTTP Request Smuggling CVE-2020-1935

The HTTP header parsing code used an approach to end-of-line (EOL) parsing that allowed some invalid HTTP headers to be parsed as valid. This led to a possibility of HTTP Request Smuggling if Tomcat was located behind a reverse proxy that incorrectly handled t invalid Transfer-Encoding header in a particular manner. Such a reverse proxy is considered unlikely.

This was fixed with commit 8bfb0ff7.

This issue was reported to the Apache Tomcat Security Team by @ZeddYu on 25 December 2019. The issue was made public on 24 February 2020.

Affects: 9.0.0.M1 to 9.0.30

Low: HTTP Request Smuggling CVE-2019-17569

The refactoring in 9.0.28 introduced a regression. The result of the regression was that invalid Transfer-Encoding headers were incorrectly processed leading to a possibility of HTTP Request Smuggling if Tomcat was located behind a reverse proxy that incorrectly handled the invalid Transfer-Encoding header in a particular manner. Such a reverse proxy is considered unlikely.

This was fixed with commit 060ecc5e.

This issue was reported to the Apache Tomcat Security Team by @ZeddYu on 12 December 2019. The issue was made public on 24 February 2020.

L’ AJP Connector è come una "porta di servizio" (chiamata porta AJP, di solito la 8009) che Tomcat usa per parlare in modo efficiente con un altro web server (come Apache HTTP Server) che gli sta davanti per gestire il traffico. Di solito, si fida molto delle connessioni che arrivano su questa porta. Il problema (la vulnerabilità) era che causa di un errore di configurazione , un attaccante esterno, **senza bisogno di password**, poteva ingannare Tomcat tramite questa porta AJP.

Così dunque . come gia spiegato , l’ attaccante poteva **leggere file sensibili** ed **eseguire codice malevolo caricandolo tramite un file**

Sul sito è spiegato che il problema è stato risolto , nelle versioni più recenti di Tomcat (come 9.0.31+, 7.0.100+ qui riportate negli screeshot) , cambiando la configurazione predefinita della porta AJP per renderla più sicura, ad esempio richiedendo una "password" (secret) per usarla o disabilitandola se non serve.

5. Conclusioni

L'attività di Vulnerability Assessment ha rivelato un elevato numero di vulnerabilità sulla macchina Metasploitable2, incluse diverse problematiche critiche e alte facilmente sfruttabili, come evidenziato dall'analisi delle prime due vulnerabilità. Questo conferma la natura intrinsecamente insicura della macchina target, progettata a scopo didattico per l'addestramento alla sicurezza. Sebbene Metasploitable2 sia un ambiente di test, l'analisi delle sue vulnerabilità fornisce

un'importante opportunità pratica per comprendere le debolezze comuni nei sistemi reali e le relative tecniche di mitigazione.