SERVICIUL
TEHNOLOGIA INFORMAȚIEI
ȘI SECURITATE CIBERNETICĂ

Vulnerabilitate exploatată activ în Zimbra Collaboration Suite

CVE-2026-73570 · Executare de cod la distanță, neautentificată

Comunicat de securitate | STISC

21.08.2026

PUBLIC · TLP:CLEAR

Comunicat de securitate

SUMAR

Serviciul Tehnologia Informației și Securitate Cibernetică (STISC) informează despre o vulnerabilitate critică în **Zimbra Collaboration Suite**, identificată prin **CVE-2026-73570**, care este **exploatată activ**.

Vulnerabilitatea permite unui atacator **neautentificat** să execute comenzi arbitrare pe serverul de poștă electronică prin simpla transmitere a unui mesaj e-mail. Nu sunt necesare credențiale și nu este necesară nicio interacțiune din partea utilizatorului sau a administratorului. Un singur mesaj este suficient.

Informații despre vulnerabilitate

Identificator	CVE-2026-73570
Produs afectat	Zimbra Collaboration Suite (ZCS)
Tip	OS command injection (CWE-78)
Scor CVSS v3.1	8.9 – ridicat (CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:C/C:H/I:H/A:L)
Vector de atac	SMTP
Autentificare	Nu este necesară
Interacțiunea utilizatorului	Nu este necesară
Impact	RCE cu privilegiile utilizatorului `zimbra`
Versiuni afectate	Toate versiunile anterioare versiunii 10.1.20
Versiune corectată	10.1.20 (publicată la 20.07.2026)
Publicare	13.08.2026

Atenție – nu există corecție pentru versiunile mai vechi

Corecția a fost publicată **exclusiv pentru ramura 10.1**. Pentru ramurile **10.0.x**, **9.0.0** și **8.8.15** nu există și nu va exista o versiune corectată, acestea aflându-se în afara perioadei de suport. Organizațiile care utilizează aceste ramuri **nu pot remedia vulnerabilitatea prin actualizare în cadrul ramurii curente** și trebuie să planifice migrarea către versiunea 10.1.20 sau ulterioară.

Descrierea tehnică

Vulnerabilitatea se află în componenta `zimbra-snmp`, expusă prin watcher-ul de loguri `swatchdog` (`zmlogswatch`), atunci când notificările SNMP sunt activate. Mecanismul de exploatare parcurge patru etape:

Etapa	Descriere
1	Atacatorul transmite un mesaj SMTP în care **envelope sender** conține payload-ul.
2	Postfix înregistrează adresa **verbatim** în `/var/log/zimbra.log`.
3	`swatchdog` (proces `perl`) parsează linia și interpolează textul, **nefiltrat**, în argumentul `zmServiceName` al apelului `snmptrap`, executat prin shell.
4	Shell-ul evaluează substituția de comandă `$( )` din argument, rezultând **RCE**.

Lanțul de procese caracteristic, util pentru detecție, este următorul:

```
perl  →  sh -c ... snmptrap ... $( <comanda atacatorului> )
```

Condiții de aplicabilitate

Sistemul este vulnerabil dacă sunt îndeplinite **simultan** următoarele condiții:

Condiție	Verificare
Versiune anterioară 10.1.20	`su - zimbra -c "zmcontrol -v"`
Pachetul `zimbra-snmp` instalat	`zmcontrol status` → linia `snmp`
Notificări SNMP activate	`su - zimbra -c "zmlocalconfig snmp_notify"` → `= 1`
`swatchdog` activ	`zmswatchctl status`

Aceste condiții sunt îndeplinite în configurația implicită, prin urmare majoritatea instalărilor neactualizate trebuie considerate vulnerabile până la proba contrarie.

Acțiunile atacatorului după exploatare

În campaniile care exploatează acest tip de vulnerabilitate, executarea inițială de cod este urmată, de regulă, de acțiuni destinate menținerii accesului și obținerii de credențiale:

Acțiune	Consecință
Copierea `localconfig.xml` în webroot	Expunerea integrală a parolelor de serviciu
Plasarea de webshell-uri `.jsp` în webroot	Acces persistent, independent de patch-ul vulnerabilității inițiale
`zmprov generateDomainPreAuthKey` pe domenii	Posibilitatea de a emite tokene de autentificare valide pentru **orice cutie poștală** din domeniu, fără cunoașterea parolei
Enumerare de conturi (`zmprov -l gaa`) și creare de conturi cu `zimbraIsAdminAccount TRUE`	Persistență care supraviețuiește ștergerii webshell-urilor
Decodare de payload prin `base64 -d | bash` sau `zmpython -mbase64 -d | bash`	Shell interactiv (reverse shell) către infrastructura atacatorului
Scrierea de fișiere canary în webroot	Confirmarea execuției codului

Actualizarea nu este suficientă

Dacă serverul a fost deja compromis, actualizarea **nu elimină** fișierele plasate de atacator, conturile create și nu invalidează credențialele sau cheile deja exfiltrate. Verificarea semnelor de compromitere este **obligatorie** și trebuie efectuată înainte de a considera incidentul închis.

Verificarea semnelor de compromitere

Verificarea vizează fișierele apărute recent și pe cele care nu aparțin instalării oficiale, în directoarele servite public:

```
/opt/zimbra/jetty_base/webapps/
/opt/zimbra/jetty/webapps/
/opt/zimbra/mailboxd/webapps/
/opt/zimbra/data/tmp/
/tmp/
```

A. Fișiere create recent de utilizatorul `zimbra`

```
find /opt/zimbra/jetty_base/webapps /opt/zimbra/jetty/webapps      /opt/zimbra/mailboxd/webapps
/opt/zimbra/data/tmp /tmp      -xdev -user zimbra -mtime -30      -printf '%TY-%Tm-%Td %TH:%TM:%TS %s %p
' 2>/dev/null | sort
```

Orice fișier apărut în aceste directoare în afara unei operațiuni planificate de administrare sau actualizare trebuie tratat ca suspect și analizat.

B. Fișiere de tip script sau executabil

```
find /opt/zimbra/jetty_base/webapps /opt/zimbra/jetty/webapps      /opt/zimbra/mailboxd/webapps /tmp -xdev
-user zimbra -mtime -30      \( -name '*.jsp' -o -name '*.sh' -o -name '*.py' -o -name '*.pl' -o -name
'*.jar' \)      -printf '%TY-%Tm-%Td %TH:%TM:%TS %s %p
' 2>/dev/null | sort
```

Rezultatul se compară cu o instalare curată de aceeași versiune. Instalarea conține JSP-uri legitime. Verificați și fișierele **fără extensie** – sunt frecvent folosite pentru a masca scripturi sau binare. Un fișier `.jsp` a cărui dată de modificare diferă de restul instalării este indiciul principal.

C. Dovada executării

```
find /opt/zimbra -xdev -name '*_jsp*.class'      -printf '%TY-%Tm-%Td %TH:%TM:%TS %p
' 2>/dev/null | sort
```

Jetty compilează fiecare JSP în bytecode înainte de execuție. Un fișier `.class` a cărui denumire nu corespunde niciunui `.jsp` legitim demonstrează că un fișier a fost **executat**, iar această dovadă **persistă chiar dacă fișierul sursă a fost șters**.

D. Urme ale vectorului de livrare în jurnalul de poștă

```
zgrep -hE 'Service status change|zmServiceStatusTrap'       /var/log/zimbra.log* /var/log/maillog*
2>/dev/null
```

Prezența acestui șablon în combinație cu o adresă de expeditor neobișnuită indică o tentativă de exploatare. Aceste jurnale acoperă, de regulă, o perioadă mai lungă decât telemetria soluțiilor de securitate și sunt esențiale pentru stabilirea momentului compromiterii.

E. Accesări ale fișierelor identificate la pașii A și B

```
zgrep -h '<denumirea fișierului identificat>'       /opt/zimbra/log/nginx.access.log*
/opt/zimbra/log/access_log* 2>/dev/null
```

Un cod de răspuns **200** arată că fișierul a fost **accesat cu succes din exterior**. Absența înregistrărilor este concludentă numai dacă jurnalul acoperă perioada analizată – verificați mai întâi intervalul de timp acoperit de jurnal.

> **CE INDICĂ O COMPROMITERE**
>
> 1. Fișiere apărute în rădăcina web în afara unei actualizări planificate.
> 2. Fișiere `.class` compilate fără un `.jsp` legitim corespunzător.
> 3. Copii ale fișierelor de configurare ale aplicației în directoare servite public.
> 4. Conturi de administrator create recent.
> 5. Chei de preautentificare modificate fără o operațiune planificată.

Verificarea proceselor și a conexiunilor

G. Procese care rulează sub utilizatorul de serviciu

```
ps -u zimbra -f
```

Se investighează execuția, în contextul utilizatorului `zimbra`, a interpretoarelor și utilitarelor: `sh`, `bash`, `curl`, `wget`, `python3`, `perl`, `nc`, `socat`, `ssh`, `base64`, `chmod`.

> **CORELARE OBLIGATORIE**
>
> Prezența izolată a uneia dintre aceste comenzi **nu demonstrează** o compromitere. Concluzia se formulează numai prin corelarea a patru elemente: **procesul-părinte**, **linia de comandă completă**, **momentul execuției** și **conexiunile de rețea asociate**.

H. Conexiuni de rețea

```
ss -plant
lsof -nP -u zimbra -i
```

Prioritare: conexiuni outbound către adrese fără legătură cunoscută cu infrastructura organizației; porturi neobișnuite; procese de tip shell care inițiază conexiuni; conexiuni apărute imediat după un eveniment `Service status change`.

> **Secvența completă – cel mai puternic indicator**
>
> Eveniment `Service status change` cu argument anormal → proces executat ca `zimbra` → shell / downloader / interpretor → scriere de fișier în webroot sau `/tmp` → conexiune externă. O astfel de secvență se tratează ca **incident de securitate**, nu ca vulnerabilitate neremediată.

Regula de detecție

Regulă EDR/SIEM. Combinația de mai jos identifică exploatarea și nu apare în funcționarea normală a aplicației:

```
parent = perl
  AND  CommandLine contains 'snmptrap'
  AND  argumentul zmServiceName conține caractere în afara [a-z0-9_-]
       ( $(  `  ;  |  &  >  < )
```

> **PRECIZARE**
>
> Valoarea legitimă a argumentului `zmServiceName` este un token simplu de serviciu (`mailbox`, `mta`). Orice caracter în afara setului `[a-z0-9_-]` în acest argument indică injectare.
>
> O regulă bazată exclusiv pe `$(` este **insuficientă**: exploatarea funcționează identic prin backtick, prin `;` și prin `|`. Alertarea pe simpla prezență a `snmptrap` generează, invers, un volum ridicat de alarme false – apelurile fără metacaractere fac parte din funcționarea normală.

Detecția prin telemetrie de scriere a fișierelor

Pe lângă regula de proces, se monitorizează evenimentele de creare de fișiere în directoarele servite public:

```
eveniment = creare/scriere de fișier
  AND  calea începe cu /opt/zimbra/*/webapps/ sau /tmp/
  AND  procesul care scrie NU este un proces de actualizare cunoscut
```

Limitare importantă a acestei metode

Scrierile efectuate prin **redirectare de shell** (`> fișier`) pot să **nu genereze deloc** evenimente de tip „fișier nou" în telemetria produselor de securitate. O regulă construită exclusiv pe evenimente de scriere **poate rata plasarea unui webshell**. Această metodă se folosește **în completarea** verificărilor pe disc (A–C), niciodată în locul lor.

Tehnici MITRE ATT&CK relevante

Tehnică	Denumire	Relevanță
T1190	Exploit Public-Facing Application	Exploatarea serviciului expus în Internet
T1059	Command and Scripting Interpreter	Execuție de comenzi la nivel de sistem de operare
T1059.004	Unix Shell	Execuție shell pe serverul-gazdă
T1105	Ingress Tool Transfer	Descărcarea de payload-uri secundare
T1505.003	Web Shell	Instalarea unui webshell în webroot
T1071	Application Layer Protocol	Comunicații outbound în post-exploatare

Prioritizarea analizei

Nivel	Palier	Criteriu
1	Expunere	ZCS < 10.1.20 cu `zimbra-snmp` instalat și `snmp_notify = 1`
2	Proces	Proces rulat ca `zimbra` : shell, downloader sau interpretor
3	Fișiere	Scrieri în webroot sau `/tmp`
4	Rețea	Proces neobișnuit urmat de conexiune outbound
5	Corelare	Secvența completă – se tratează ca incident confirmat

> **VALIDAREA EXPUNERII**
>
> Expunerea se stabilește prin verificarea versiunii și a configurației (`zmcontrol -v` , `zmlocalconfig snmp_notify`).
>
> **Nu validați vulnerabilitatea prin transmiterea unui payload real către un sistem de producție.**

Măsuri recomandate

Nr.	Măsura
1	**Actualizați la versiunea 10.1.20 sau ulterioară.** Pentru ramurile 10.0.x, 9.0.0 și 8.8.15 planificați migrarea – nu există corecție.
2	Efectuați verificările A–F înainte de a considera incidentul închis.
3	Dacă actualizarea imediată nu este posibilă, evaluați dezactivarea temporară a componentei de monitorizare SNMP.
4	La orice indiciu de compromitere, **rotiți toate secretele** din `localconfig.xml` și **toate cheile de preautentificare** pe domenii.
5	Verificați conturile de administrator create recent.
6	**Nu ștergeți** fișierele suspecte înainte de a le conserva ca probă.
7	Comutați soluția EDR din modul *detect-only* în modul de blocare.
8	Limitați expunerea la internet a interfețelor de administrare.

Validarea după remediere

```
su - zimbra -c "zmcontrol -v"          # trebuie: 10.1.20 sau superior
su - zimbra -c "zmcontrol status"
su - zimbra -c "zmlocalconfig snmp_notify"
```

Actualizarea nu închide un incident deja produs

Verificările A–H se repetă **după** actualizare. Un webshell plasat înainte de patch rămâne funcțional, iar credențialele și cheile exfiltrate rămân valabile până la rotire.

Acțiuni la identificarea unor indicatori de compromitere

Nr.	Acțiune
1	Izolați sistemul prin mecanismele aprobate de răspuns la incidente.
2	Conservați jurnalele și telemetria înainte de orice intervenție.
3	Colectați hash-urile SHA-256 ale fișierelor suspecte.
4	Reconstituiți arborele de procese și conexiunile outbound asociate.
5	Verificați mecanismele de persistență: conturi, sarcini programate, unități de serviciu.
6	Rotiți credențialele și cheile de preautentificare.
7	Aplicați ZCS 10.1.20 sau ulterior.
8	Monitorizați reapariția acelorași indicatori.

MĂSURĂ TEMPORARĂ

Dacă notificările SNMP nu sunt necesare operațional, dezactivarea lor elimină condiția de exploatare, până la finalizarea actualizării:

```
su - zimbra -c "zmlocalconfig -e snmp_notify=0"

# verificare:
su - zimbra -c "zmlocalconfig snmp_notify"
# rezultat așteptat: snmp_notify = 0
```

Producătorul nu a publicat o soluție alternativă oficială; unica remediere indicată este actualizarea. Dezactivarea notificărilor SNMP este o **mitigare temporară**, nu substituie instalarea versiunii 10.1.20 sau ulterioare și nu are efect asupra unui sistem deja compromis.

Raportare

Organizațiile care identifică semne de compromitere sunt rugate să contacteze centrul de răspuns la incidente de securitate cibernetică competent pentru sectorul din care fac parte, iar în lipsa unui centru sectorial desemnat, echipa națională de răspuns la incidente.

Conservați jurnalele și fișierele suspecte în forma originală. Acestea sunt esențiale pentru stabilirea momentului și a amplorii compromiterii, iar ștergerea lor face analiza imposibilă.

Sursa vulnerabilității: anunțul de securitate al producătorului și înregistrarea CVE-2026-73570. Exploatarea activă a fost semnalată public începând cu 17.08.2026.