

# 별칭 컨피그레이션 가져오기 및 내보내기 자동화

## 목차

---

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[배경 정보](#)

[별칭 테이블 내보내기 및 가져오기](#)

[Bash 스크립트를 사용하여 별칭 테이블 내보내기](#)

[설명](#)

[1.- SSH 키 및 경로 컨피그레이션](#)

[2.- 프록시에 연결하고 SSH 터널을 설정합니다.](#)

[3.- 계속 진행하기 전에 5초 동안 일시 중지](#)

[4.- 원격 시스템에서 aliasconfig 파일 내보내기](#)

[5.- 파일을 로컬 디렉토리에 다운로드합니다.](#)

[스크립트 마무리](#)

[SSH 키](#)

[내보낸 파일 확인](#)

[Bash 스크립트를 사용하여 별칭 테이블 가져오기](#)

[설명](#)

[1.- SSH 경로 및 키 컨피그레이션](#)

[2.- 현재 날짜 및 시간 가져오기](#)

[3.- 원격 ESA 서버에 새 aliasconfig 파일 업로드](#)

[4.- 새 aliasconfig 파일을 가져오고 코멘트로 커밋](#)

[5.- 현재 aliasconfig를 인쇄하고 새 로컬 파일에 저장합니다.](#)

[변경 사항 확인](#)

[새 aliasconfig 테이블 항목 확인](#)

[커밋 변경 사항 확인](#)

[스크립트 유연성](#)

[마지막 생각](#)

[참조 링크](#)

---

## 소개

이 문서에서는 Email Security Appliance에서 별칭 가져오기 및 내보내기 컨피그레이션의 작업을 자동화하는 단계에 대해 설명합니다.

## 사전 요구 사항

## 요구 사항

Cisco에서는 다음 항목에 대한 지식을 권장합니다.

- Cisco SEG/ESA(Secure Email Gateway) AsyncOS 16.0.2
- 클라우드 어플라이언스에 대한 명령줄 인터페이스 액세스
- Linux CLI
- 셸 스크립팅

## 사용되는 구성 요소

이 문서의 정보는 다음 소프트웨어를 기반으로 합니다.

- CESA(Cloud Email Security Appliance)
- 바시

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

## 배경 정보

특정 작업을 자동화하는 것이 목적이지만, 일반적으로 수동 개입이 필요한 프로세스도 있습니다. 그러나 현재 별칭 컨피그레이션을 가져오고 내보내는 경우 이러한 작업을 완전히 자동화할 수 있으므로 수동으로 입력할 필요가 없습니다.

## 별칭 테이블 내보내기 및 가져오기

별칭 테이블을 가져오려면 먼저 SSH 및 SCP 액세스를 확인하여 이메일 게이트웨이에 연결할 수 있도록 합니다.

계속하기 전에 별칭 테이블이 어플라이언스 내에 있어야 합니다.

```
(Machine esa1.xyz.iphmx.com) (SERVICE)> clustermode cluster; aliasconfig print
```

```
test: test@example.com, test@example2.com, test@example3.com
test2: test@domain.com, test@domain2.com, test@domain3.com
(Cluster Hosted_Cluster) (SERVICE)>
```

aliasconfig 명령의 export 하위 명령을 사용하여 기존 별칭 테이블을 백업하면 파일(사용자가 지정하는 이름)이 생성되어 리스너의 /configuration 디렉터리에 저장됩니다.

## Bash 스크립트를 사용하여 별칭 테이블 내보내기

이 경우 bash 스크립트가 CES 어플라이언스에 연결되고 별칭 파일 내보내기를 진행합니다

bash 스크립트는 다음과 같이 구성됩니다.

```
#!/bin/bash

# Configuration of SSH keys and paths
PROXY_KEY="/full/path/folder/.ssh/id_rsa"
SECOND_KEY="/full/path/folder/.ssh/id_rsa"
LOCAL_PORT="2200"
PROXY_USER="dh-user"
PROXY_HOST="f4-ssh.iphmx.com"
TARGET_HOST="esa1.xyz.iphmx.com"
REMOTE_USER="local_server_user"
REMOTE_FILE="/configuration/filename.csv"
LOCAL_DIR="/full/path/folder/Downloads"
LOCAL_FILE_PATH="${LOCAL_DIR}/aliasconfig-file.csv"

# 1. Connect to the proxy and set up the SSH tunnel
echo "Establishing connection to the proxy..."
ssh -i "$PROXY_KEY" -l "$PROXY_USER" -N -f "$PROXY_HOST" -L "$LOCAL_PORT:${TARGET_HOST}:22"
if [ $? -ne 0 ]; then
    echo "Error: Failed to establish connection to the proxy."
    exit 1
fi
echo "Proxy connection established."

# Pause for 5 seconds before proceeding
sleep 5

# 2. Export the aliasconfig file from the remote system
echo "Exporting aliasconfig file from the remote system..."
ssh -i "$SECOND_KEY" "$REMOTE_USER"@127.0.0.1 -p "$LOCAL_PORT" 'clustermode cluster; aliasconfig export'
if [ $? -ne 0 ]; then
    echo "Error: Failed to export the aliasconfig file."
    exit 1
fi
echo "Aliasconfig file successfully exported."

# Pause for 5 seconds before proceeding
sleep 5

# 3. Download the file to the local directory
echo "Downloading file to the local directory..."
scp -i "$SECOND_KEY" -P "$LOCAL_PORT" -O "$REMOTE_USER"@127.0.0.1:"$REMOTE_FILE" "$LOCAL_DIR" 2>/dev/nu
if [ $? -ne 0 ]; then
    echo "Error: Failed to download the file to the local directory."
    exit 1
fi
echo "File successfully downloaded to: $LOCAL_FILE_PATH"

# Pause for 5 seconds before finalizing
sleep 5

# Finalizing the script
echo "Process completed successfully."
exit 0
```

# 설명

## 1.- SSH 키 및 경로 컨피그레이션

- PROXY\_KEY 및 SECOND\_KEY 인증에 사용되는 SSH 개인 키 파일의 전체 경로입니다. 이 경우 두 키가 동일한 경로로 설정됩니다.
- 예: /full/path/folder/.ssh/id\_rsa
- LOCAL\_PORT: SSH 터널의 로컬 포트(2200)를 지정합니다.
- 프록시 사용자(\_U): 프록시 서버에 연결하는 데 사용되는 사용자 이름.
- 프록시\_호스트: 프록시 서버의 호스트 이름입니다.
- 대상\_호스트: 대상 호스트의 FQDN(정규화된 도메인 이름)이 esa1.xyz.iphmx.com으로 업데이트되었습니다.
- 원격 사용자(\_U): SSH 터널을 통해 원격 어플라이언스에 연결하는 데 사용되는 사용자 이름.
- 원격 파일(\_F): 내보낸 파일이 저장된 원격 시스템의 경로(/configuration/filename.csv).
- 로컬 디렉터리(\_D): 파일을 저장할 로컬 디렉터리는 /full/path/folder/Downloads로 설정됩니다.
- LOCAL\_FILE\_PATH: 다운로드한 파일의 전체 로컬 경로이며 \${LOCAL\_DIR}/aliasconfig-file.csv로 업데이트되었습니다.

## 2.- 프록시에 연결하고 SSH 터널을 설정합니다.

```
echo "Establishing connection to the proxy..."  
ssh -i "$PROXY_KEY" -l "$PROXY_USER" -N -f "$PROXY_HOST" -L "$LOCAL_PORT:${TARGET_HOST}:22"
```

- 목적:
  - 대상 호스트와의 보안 통신을 위해 프록시 서버에 SSH 터널을 설정합니다.
- 업데이트:
  - SSH 개인 키는 이제 /full/path/folder/.ssh/id\_rsa에 있습니다.
  - 대상 호스트 이름이 esa1.xyz.iphmx.com으로 업데이트되었습니다.
- 오류 처리:
  - 연결이 실패하면 오류 메시지가 표시되고 스크립트가 오류 코드와 함께 종료됩니다(종료 1).

## 3.- 계속 진행하기 전에 5초 동안 일시 중지

- 목적:
  - 다음 단계로 진행하기 전에 SSH 터널이 완전히 설정되도록 지연을 적용합니다.

## 4.- 원격 시스템에서 aliasconfig 파일 내보내기

```
echo "Exporting aliasconfig file from the remote system..."  
ssh -i "$SECOND_KEY" "$REMOTE_USER"@127.0.0.1 -p "$LOCAL_PORT" 'clustermode cluster; aliasconfig export'
```

- 목적:
  - SSH 터널을 통해 대상 호스트에 연결하고 별칭 컨피그레이션을 aliasconfig-file.csv라는 파일에 내보냅니다.
- 업데이트:
  - 내보낸 파일의 파일 이름이 aliasconfig-file.csv로 업데이트되었습니다.
- 출력 리디렉션:
  - 2>/dev/null은 SSH 명령의 모든 오류 메시지를 억제합니다.
- 오류 처리:
  - 내보내기가 실패하면 오류 메시지가 표시되고 스크립트가 종료됩니다.

## 5.- 파일을 로컬 디렉토리에 다운로드합니다.

```
echo "Downloading file to the local directory..."
scp -i "$SECOND_KEY" -P "$LOCAL_PORT" -O "$REMOTE_USER"@127.0.0.1:"$REMOTE_FILE" "$LOCAL_DIR" 2>/dev/nu
```

- 목적:
  - scp를 사용하여 내보낸 파일을 원격 시스템에서 로컬 디렉토리로 안전하게 복사합니다.
- 업데이트:
  - 로컬 파일이 /full/path/folder/Downloads 디렉터리에 aliasconfig-file.csv로 저장됩니다.
- 오류 처리:
  - 파일 다운로드가 실패하면 오류 메시지가 표시되고 스크립트가 종료됩니다.

## 스크립트 마무리

```
echo "Process completed successfully."
exit 0
```

- 목적:
  - 성공 메시지를 출력하고 모든 작업이 성공적으로 완료되었음을 나타내는 성공 코드(종료 0)와 함께 스크립트를 종료합니다.

## SSH 키

스크립트에서 PROXY\_KEY 및 HOST\_KEY 변수 2개를 확인할 수 있습니다. 이러한 키는 동일하거나 다를 수 있습니다.

PROXY\_KEY는 CESA 서버로 이동하기 위해 반드시 필요한 프록시 클라우드에 연결하는 데 사용됩니다.

HOST\_KEY는 로컬 사용자로 로그인하는 데 사용되는 키이므로 비밀번호가 필요하지 않습니다.

---

참고: CES 프록시에 대한 SSH 액세스를 구성하고 어플라이언스의 로컬 사용자에게 대한 SSH 키를 설정하려면 컨피그레이션 가이드를 참조하십시오.

---

## 내보낸 파일 확인

내보내기 스크립트가 실행되면 해당 스크립트의 내용을 확인하고 어플라이언스 aliasconfig CLI 명령에 있는 원본과 동일한 정보를 포함하는지 확인할 수 있습니다.

```
$ pwd
/full/path/folder/Downloads
$ ls
filename.csv
$ cat filename.csv
# File exported by the CLI at 20250702T125347
test: test@example.com, test@example2.com, test@example3.com
test2: test@domain.com, test@domain2.com, test@domain3.com↵
```

# Bash 스크립트를 사용하여 별칭 테이블 가져오기

현재 별칭 파일을 내보낸 후에는 해당 파일을 수정하고 필요한 항목을 추가한 다음 ESA 별칭 컨피그레이션으로 가져올 수 있습니다.

가져오기 bash 스크립트는 다음과 같이 구성됩니다.

```
#!/bin/bash

# Configuration of SSH keys and paths
SSH_KEY="/full/path/folder/.ssh/id_rsa"
LOCAL_PORT="2200"
REMOTE_USER="local_server_user"
LOCAL_FILE="/full/path/folder/Downloads/new-filename.csv"
OUTPUT_DIR="/full/path/folder/Downloads"

# Get the current local date in the desired format
CURRENT_DATE=$(date +"%Y-%m-%d_%H-%M-%S")

# 1. Upload the new aliasconfig file
echo "Uploading new aliasconfig file to the remote system..."
scp -i "$SSH_KEY" -P "$LOCAL_PORT" -O "$LOCAL_FILE" "$REMOTE_USER"@127.0.0.1:/configuration 2>/dev/null
if [ $? -ne 0 ]; then
    echo "Error: Failed to upload the aliasconfig file."
    exit 1
fi
echo "Aliasconfig file successfully uploaded."

# Pause for 5 seconds before proceeding
sleep 5

# 2. Import the new aliasconfig file and commit with a comment
COMMIT_COMMENT="Importing new entries to aliasconfig - $CURRENT_DATE"
echo "Importing the new aliasconfig file and committing changes..."
ssh -i "$SSH_KEY" "$REMOTE_USER"@127.0.0.1 -p "$LOCAL_PORT" "clustermode cluster; aliasconfig import new"
if [ $? -ne 0 ]; then
    echo "Error: Failed to import the aliasconfig file or commit changes."
    exit 1
fi
echo "Aliasconfig file successfully imported and committed with comment: '$COMMIT_COMMENT'."

# Pause for 5 seconds before proceeding
sleep 5

# 3. Print the current aliasconfig and save it to a new file
OUTPUT_FILE="${OUTPUT_DIR}/current-aliasconfig-${CURRENT_DATE}.txt"
echo "Printing current aliasconfig and saving it to: $OUTPUT_FILE..."
ssh -i "$SSH_KEY" "$REMOTE_USER"@127.0.0.1 -p "$LOCAL_PORT" 'clustermode cluster; aliasconfig print' > $OUTPUT_FILE
if [ $? -ne 0 ]; then
    echo "Error: Failed to print the current aliasconfig."
    exit 1
fi
echo "Current aliasconfig successfully saved to: $OUTPUT_FILE"

# Finalizing the script
echo "Process completed successfully."
exit 0
```

# 설명

## 1.- SSH 경로 및 키 컨피그레이션

- SSH\_키: 원격 서버에 대해 안전하게 인증하는 데 사용되는 SSH 개인 키 파일의 경로입니다.
- LOCAL\_PORT: SSH 터널에 대해 지정된 로컬 포트입니다.
- 원격 사용자(\_U): 원격 서버에 대한 인증을 위한 사용자 계정입니다.
- 로컬 파일(\_F): 가져올 aliasconfig CSV 파일의 로컬 경로입니다.
- 출력\_디렉터리: 가져오기 프로세스 후에 현재 구성의 복사본이 저장되는 로컬 폴더입니다.

## 2.- 현재 날짜 및 시간 가져오기

```
CURRENT_DATE=$(date +%Y-%m-%d_%H-%M-%S)
```

- 목적:
  - 파일 이름 및 주석에 사용할 수 있도록 현재 날짜와 시간을 특정 형식으로 저장합니다.
- 업데이트:
  - 타임스탬프로 로그와 백업을 손쉽게 추적하고 구성할 수 있습니다.

## 3.- 원격 ESA 서버에 새 aliasconfig 파일 업로드

aliasconfig 파일의 새 콘텐츠:

```
# File exported by the CLI at 20250709T112719
test: new-data@example.com, new-date@example2.com, new-date@example3.com
test2: new-date@domain.com, new-data@domain2.com, new-data@domain3.com
```

파일 업로드 지침을 계속 진행합니다.

```
echo "Uploading new aliasconfig file to the remote system..."
scp -i "$SSH_KEY" -P "$LOCAL_PORT" -O "$LOCAL_FILE" "$REMOTE_USER"@127.0.0.1:/configuration 2>/dev/null
if [ $? -ne 0 ]; then
    echo "Error: Failed to upload the aliasconfig file."
    exit 1
fi
echo "Aliasconfig file successfully uploaded."
```

- 목적:

- SCP over SSH를 사용하여 로컬 시스템에서 원격 서버의 /configuration 디렉토리로 aliasconfig CSV 파일을 전송합니다.
- 업데이트:
  - 업로드가 실패하면 스크립트에 오류가 표시되고 불완전한 가져오기를 방지하기 위해 중지됩니다.

#### 4.- 새 aliasconfig 파일을 가져오고 코멘트로 커밋

```
COMMIT_COMMENT="Importing new entries to aliasconfig - $CURRENT_DATE"
echo "Importing the new aliasconfig file and committing changes..."
ssh -i "$SSH_KEY" "$REMOTE_USER"@127.0.0.1 -p "$LOCAL_PORT" "clustermode cluster; aliasconfig import new"
if [ $? -ne 0 ]; then
    echo "Error: Failed to import the aliasconfig file or commit changes."
    exit 1
fi
echo "Aliasconfig file successfully imported and committed with comment: '$COMMIT_COMMENT'."
```

- 목적:
  - SSH를 통해 연결하고 업로드된 CSV 파일을 별칭 컨피그레이션으로 가져온 다음 추적을 위해 타임스탬프가 포함된 코멘트로 변경 사항을 커밋합니다.
- 업데이트:
  - 가져오기 또는 커밋이 실패하면 오류 메시지가 표시되고 컨피그레이션을 일관되게 유지하기 위해 스크립트가 종료됩니다.

#### 5.- 현재 aliasconfig를 인쇄하고 새 로컬 파일에 저장합니다.

```
OUTPUT_FILE="${OUTPUT_DIR}/current-aliasconfig-${CURRENT_DATE}.txt"
echo "Printing current aliasconfig and saving it to: $OUTPUT_FILE..."
ssh -i "$SSH_KEY" "$REMOTE_USER"@127.0.0.1 -p "$LOCAL_PORT" 'clustermode cluster; aliasconfig print' > $OUTPUT_FILE
if [ $? -ne 0 ]; then
    echo "Error: Failed to print the current aliasconfig."
    exit 1
fi
echo "Current aliasconfig successfully saved to: $OUTPUT_FILE"
```

- 목적:
  - SSH를 통해 연결하고 현재 별칭 컨피그레이션을 인쇄한 다음 백업 및 감사를 위해 로컬 타임스탬핑 파일에 출력을 저장합니다.
- 업데이트:
  - 작업이 실패하면 스크립트가 오류를 표시하고 중지하여 불완전한 결과를 방지합니다.

## 변경 사항 확인

스크립트가 실행되면 별칭 컨피그레이션 테이블에서 변경 사항을 확인하고 시스템 로그에서 커밋

을 확인할 수 있습니다.

## 새 aliasconfig 테이블 항목 확인

새 변경 사항이 테이블에 적용되었습니다.

```
(Machine esa1.xyz.iphmx.com) (SERVICE)> clustermode cluster; aliasconfig print
```

```
test: new-data@example.com, new-data@example2.com, new-data@example3.com  
test2: new-data@domain.com, new-data@domain2.com, new-data@domain3.com
```

## 커밋 변경 사항 확인

이 명령을 사용하면 ESA에 커밋된 변경 사항을 추적하고 검토할 수 있습니다. 여기에는 변경 사용자 및 변경 날짜가 포함됩니다.

```
(Machine esa1.xyz-66.iphmx.com) (SERVICE)> grep "commit" system_logs
```

```
Wed Jul 9 11:29:42 2025 Info: PID 95790: User local_server_user commit changes: Importing new entries
```

## 스크립트 유연성

이 스크립트는 현재 Bash로 작성되었지만 Python, PowerShell 또는 Perl과 같은 다른 스크립팅 또는 프로그래밍 언어로 쉽게 조정 또는 재작성될 수 있으므로 다양한 관리자 및 환경의 기본 설정 또는 요구 사항에 더 잘 부합합니다. 이러한 유연성을 통해 핵심 논리와 워크플로를 유지하는 동시에 운영 요구 사항과 가장 밀접한 언어나 툴을 활용할 수 있습니다.

## 마지막 생각

이 가져오기/내보내기 스크립트는 어플라이언스에서 직접 별칭 컨피그레이션을 관리하기 위한 실용적이고 효율적인 솔루션을 제공합니다. 구성 파일의 업로드, 가져오기 및 백업을 자동화하여 관리자가 수동 작업 없이 변경 사항을 안전하고 안정적으로 도입할 수 있습니다. 이 스크립트는 프로세스를 간소화할 뿐 아니라 타임스탬프가 지정된 백업 및 커밋 주석을 통해 추적 가능성을 보장합니다.

또한 이러한 스크립트를 사용하면 특히 여러 가지 변경 또는 대량 업데이트가 필요한 경우 환경의 일관성과 규정 준수를 유지하는 데 도움이 됩니다. 현재 컨피그레이션의 정기적인 백업은 추가 보안 계층을 제공하여 필요한 경우 신속한 복구 또는 롤백을 지원합니다.

전반적으로 이 접근 방식을 통해 팀은 컨피그레이션 업데이트를 더욱 확신, 제어 및 효율적으로 관리할 수 있습니다. 향후 조정을 위해 스크립트를 손쉽게 조정하여 다른 유형의 컨피그레이션 파일을 처리하거나 필요에 따라 추가 유지 관리 작업을 자동화할 수 있습니다.

## 참조 링크

- [CES\(Cloud Email Security\) 솔루션의 CLI\(Command Line Interface\) 액세스](#)
- [CLI 지침: PuTTY \[Windows/PC 사용자\]](#)
- [비밀번호 없이 ESA에 로그인하도록 SSH 공개 키 인증을 구성하는 방법](#)

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.