이미지 서명 확인

Cloud Volumes ONTAP

NetApp
July 22, 2024

This PDF was generated from https://docs.netapp.com/ko-kr/test/concept-gcp-verify-signed-image.html on July 22, 2024. Always check docs.netapp.com for the latest.

목차

이미지 서명 확인 1
Google Cloud 서명 이미지를 확인합니다 1
OpenSSL을 사용하여 disk.raw 파일 및 다이제스트 파일 내용을 확인합니다 2

이미지 서명 확인

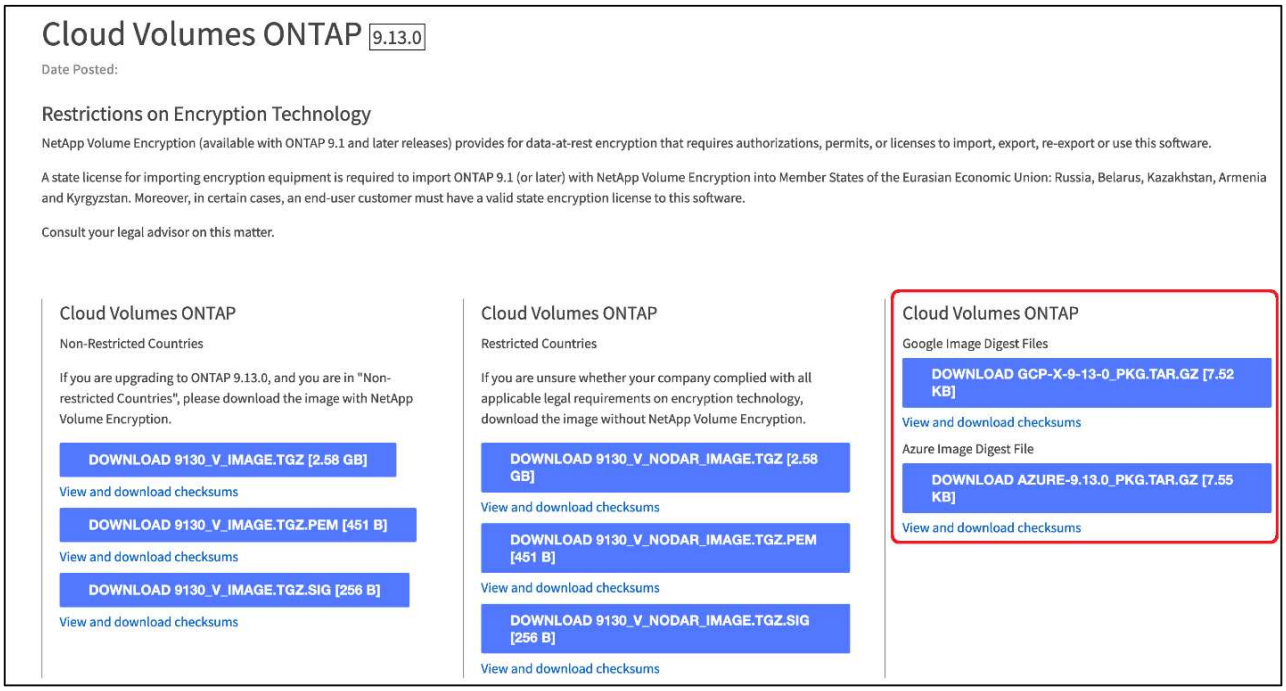
Google Cloud 서명 이미지를 확인합니다

내보낸 Google Cloud 서명 이미지를 확인하려면 NSS에서 이미지 다이제스트 파일을 다운로드하여 disk.raw 파일의 유효성을 검사하고 파일 내용을 다이제해야 합니다.

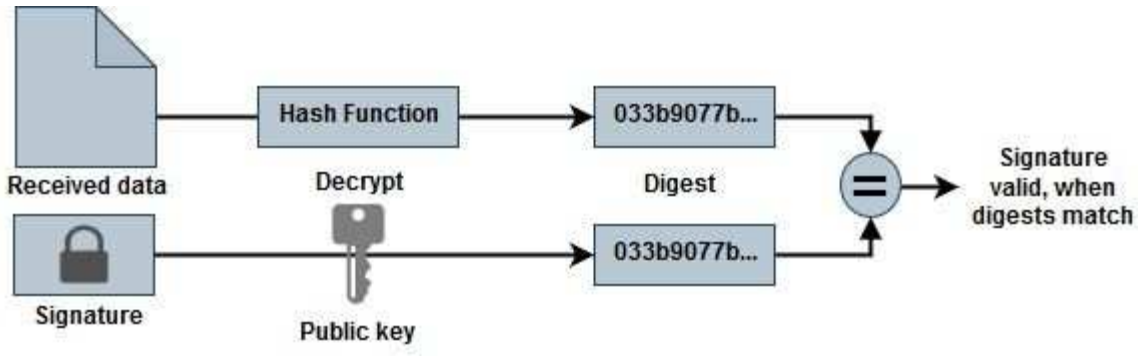
서명된 이미지 검증 워크플로 요약

다음은 Google Cloud 서명 이미지 검증 워크플로 프로세스에 대한 개요입니다.

- 에서 "NSS"에서 다음 파일이 포함된 Google Cloud 아카이브를 다운로드합니다.
  - 서명된 다이제스트(.SIG)
  - 공개 키(.pem)가 포함된 인증서
  - 인증서 체인(.pem)



- 변환된 disk.raw 파일을 다운로드합니다
- 인증서 체인을 사용하여 인증서의 유효성을 검사합니다
- 인증서에 공개 키가 포함되어 있는 서명된 다이제스트를 확인합니다
  - 공개 키를 사용하여 서명된 다이제스트를 해독하여 이미지 파일의 다이제스트를 추출합니다
  - 다운로드한 disk.raw 파일의 다이제스트를 만듭니다
  - 검증을 위해 두 개의 다이제스트 파일을 비교합니다



OpenSSL을 사용하여 disk.raw 파일 및 다이제스트 파일 내용을 확인합니다

Google Cloud에서 다운로드한 disk.raw 파일을 을 통해 사용 가능한 다이제스트 파일 내용과 비교하여 확인할 수 있습니다 "NSS" OpenSSL 사용.

> 이미지의 유효성을 검사하는 OpenSSL 명령은 Linux, Mac OS 및 Windows 시스템과 호환됩니다.

단계

1. OpenSSL을 사용하여 인증서를 확인합니다.

스크립트를 표시하려면 클릭합니다

```
# Step 1 - Optional, but recommended: Verify the certificate using
OpenSSL

# Step 1.1 - Copy the Certificate and certificate chain to a
directory
$ openssl version
LibreSSL 3.3.6
$ ls -l
total 48
-rw-r--r--@ 1 example-user  engr  8537 Jan 19 15:42 Certificate-
Chain-GCP-CVO-20230119-0XXXXX.pem
-rw-r--r--@ 1 example-user  engr  2365 Jan 19 15:42 Certificate-GCP-
CVO-20230119-0XXXXX.pem

# Step 1.2 - Get the OSCP URL
$ oscp_url=$(openssl x509 -noout -ocsp_uri -in <Certificate-
Chain.pem>)
$ oscp_url=$(openssl x509 -noout -ocsp_uri -in Certificate-Chain-
GCP-CVO-20230119-0XXXXX.pem)
$ echo $oscp_url
http://ocsp.entrust.net

# Step 1.3 - Generate an OCSP request for the certificate
$ openssl ocsp -issuer <Certificate-Chain.pem> -CAfile <Certificate-
Chain.pem> -cert <Certificate.pem> -reqout <request.der>
$ openssl ocsp -issuer Certificate-Chain-GCP-CVO-20230119-0XXXXX.pem
-CAfile Certificate-Chain-GCP-CVO-20230119-0XXXXX.pem -cert
Certificate-GCP-CVO-20230119-0XXXXX.pem -reqout req.der

# Step 1.4 - Optional: Check the new file "req.der" has been
generated
$ ls -l
total 56
-rw-r--r--@ 1 example-user  engr  8537 Jan 19 15:42 Certificate-
Chain-GCP-CVO-20230119-0XXXXX.pem
-rw-r--r--@ 1 example-user  engr  2365 Jan 19 15:42 Certificate-GCP-
CVO-20230119-0XXXXX.pem
-rw-r--r--  1 example-user  engr   120 Jan 19 16:50 req.der

# Step 1.5 - Connect to the OCSP Manager using openssl to send the
OCSP request
$ openssl ocsp -issuer <Certificate-Chain.pem> -CAfile <Certificate-
Chain.pem> -cert <Certificate.pem> -url ${ocsp_url} -resp_text
-respout <response.der>
```

```
$ openssl ocsp -issuer Certificate-Chain-GCP-CVO-20230119-0XXXXX.pem
-CAfile Certificate-Chain-GCP-CVO-20230119-0XXXXX.pem -cert
Certificate-GCP-CVO-20230119-0XXXXX.pem  -url ${ocsp_url} -resp_text
-respout resp.der
OCSP Response Data:
    OCSP Response Status: successful (0x0)
    Response Type: Basic OCSP Response
    Version: 1 (0x0)
    Responder Id: C = US, O = "Entrust, Inc.", CN = Entrust Extended
Validation Code Signing CA - EVCS2
    Produced At: Jan 19 15:14:00 2023 GMT
    Responses:
    Certificate ID:
      Hash Algorithm: sha1
      Issuer Name Hash: 69FA640329AB84E27220FE0927647B8194B91F2A
      Issuer Key Hash: CE894F8251AA15A28462CA312361D261FBF8FE78
      Serial Number: 5994B3D01D26D594BD1D0FA7098C6FF5
    Cert Status: good
    This Update: Jan 19 15:00:00 2023 GMT
    Next Update: Jan 26 14:59:59 2023 GMT

    Signature Algorithm: sha512WithRSAEncryption
         0b:b6:61:e4:03:5f:98:6f:10:1c:9a:f7:5f:6f:c7:e3:f4:72:
         f2:30:f4:86:88:9a:b9:ba:1e:d6:f6:47:af:dc:ea:e4:cd:31:
         af:e3:7a:20:35:9e:60:db:28:9c:7f:2e:17:7b:a5:11:40:4f:
         1e:72:f7:f8:ef:e3:23:43:1b:bb:28:1a:6f:c6:9c:c5:0c:14:
         d3:5d:bd:9b:6b:28:fb:94:5e:8a:ef:40:20:72:a4:41:df:55:
         cf:f3:db:1b:39:e0:30:63:c9:c7:1f:38:7e:7f:ec:f4:25:7b:
         1e:95:4c:70:6c:83:17:c3:db:b2:47:e1:38:53:ee:0a:55:c0:
         15:6a:82:20:b2:ea:59:eb:9c:ea:7e:97:aa:50:d7:bc:28:60:
         8c:d4:21:92:1c:13:19:b4:e0:66:cb:59:ed:2e:f8:dc:7b:49:
         e3:40:f2:b6:dc:d7:2d:2e:dd:21:82:07:bb:3a:55:99:f7:59:
         5d:4a:4d:ca:e7:8f:1c:d3:9a:3f:17:7b:7a:c4:57:b2:57:a8:
         b4:c0:a5:02:bd:59:9c:50:32:ff:16:b1:65:3a:9c:8c:70:3b:
         9e:be:bc:4f:f9:86:97:b1:62:3c:b2:a9:46:08:be:6b:1b:3c:
         24:14:59:28:c6:ae:e8:d5:64:b2:f8:cc:28:24:5c:b2:c8:d8:
         5a:af:9d:55:48:96:f6:3e:c6:bf:a6:0c:a4:c0:ab:d6:57:03:
         2b:72:43:b0:6a:9f:52:ef:43:bb:14:6a:ce:66:cc:6c:4e:66:
         17:20:a3:64:e0:c6:d1:82:0a:d7:41:8a:cc:17:fd:21:b5:c6:
         d2:3a:af:55:2e:2a:b8:c7:21:41:69:e1:44:ab:a1:dd:df:6d:
         15:99:90:cc:a0:74:1e:e5:2e:07:3f:50:e6:72:a6:b9:ae:fc:
         44:15:eb:81:3d:1a:f8:17:b6:0b:ff:05:76:9d:30:06:40:72:
         cf:d5:c4:6f:8b:c9:14:76:09:6b:3d:6a:70:2c:5a:c4:51:92:
         e5:cd:84:b6:f9:d9:d5:bc:8d:72:b7:7c:13:9c:41:89:a8:97:
         6f:4a:11:5f:8f:b6:c9:b5:df:00:7e:97:20:e7:29:2e:2b:12:
         77:dc:e2:63:48:87:42:49:1d:fc:d0:94:a8:8d:18:f9:07:85:
```

```
          e4:d0:3e:9a:4a:d7:d5:d0:02:51:c3:51:1c:73:12:96:2d:75:
          22:83:a6:70:5a:4a:2b:f2:98:d9:ae:1b:57:53:3d:3b:58:82:
          38:fc:fa:cb:57:43:3f:3e:7e:e0:6d:5b:d6:fc:67:7e:07:7e:
          fb:a3:76:43:26:8f:d1:42:d6:a6:33:4e:9e:e0:a0:51:b4:c4:
          bc:e3:10:0d:bf:23:6c:4b
WARNING: no nonce in response
Response Verify OK
Certificate-GCP-CVO-20230119-0XXXXX.pem: good
    This Update: Jan 19 15:00:00 2023 GMT
    Next Update: Jan 26 14:59:59 2023 GMT

# Step 1.5 - Optional: Check the response file "response.der" has
been generated. Verify its contents.
$ ls -l
total 64
-rw-r--r--@ 1 example-user  engr  8537 Jan 19 15:42 Certificate-
Chain-GCP-CVO-20230119-0XXXXX.pem
-rw-r--r--@ 1 example-user  engr  2365 Jan 19 15:42 Certificate-GCP-
CVO-20230119-0XXXXX.pem
-rw-r--r--  1 example-user  engr   120 Jan 19 16:50 req.der
-rw-r--r--  1 example-user  engr   806 Jan 19 16:51 resp.der

# Step 1.6 - Verify the chain of trust and expiration dates against
the local host
$ openssl version -d
OPENSSLDIR: "/private/etc/ssl"
$ OPENSSLDIR=$(openssl version -d | cut -d '"' -f2)
$ echo $OPENSSLDIR
/private/etc/ssl

$ openssl verify -untrusted <Certificate-Chain.pem> -CApath <OpenSSL
dir> <Certificate.pem>
$ openssl verify -untrusted Certificate-Chain-GCP-CVO-20230119-
0XXXXX.pem -CApath ${OPENSSLDIR} Certificate-GCP-CVO-20230119-
0XXXXX.pem
Certificate-GCP-CVO-20230119-0XXXXX.pem: OK
```

2. 다운로드한 disk.raw 파일, 서명 및 인증서를 디렉터리에 넣습니다.
3. OpenSSL을 사용하여 인증서에서 공개 키를 추출합니다.
4. 추출된 공개 키를 사용하여 서명을 해독하고 다운로드한 disk.raw 파일의 내용을 확인합니다.

스크립트를 표시하려면 클릭합니다

```
# Step 1 - Place the downloaded disk.raw, the signature and the
certificates in a directory
$ ls -l
-rw-r--r--@ 1 example-user  staff  Jan 19 15:42 Certificate-Chain-
GCP-CVO-20230119-0XXXXX.pem
-rw-r--r--@ 1 example-user  staff  Jan 19 15:42 Certificate-GCP-CVO-
20230119-0XXXXX.pem
-rw-r--r--@ 1 example-user  staff  Jan 19 15:42 GCP_CVO_20230119-
XXXXXX_digest.sig
-rw-r--r--@ 1 example-user  staff  Jan 19 16:39 disk.raw

# Step 2 - Extract the public key from the certificate
$ openssl x509 -pubkey -noout -in (certificate.pem) >
(public_key.pem)
$ openssl x509 -pubkey -noout -in Certificate-GCP-CVO-20230119-
0XXXXX.pem > CVO-GCP-pubkey.pem

$ ls -l
-rw-r--r--@ 1 example-user  staff  Jan 19 15:42 Certificate-Chain-
GCP-CVO-20230119-0XXXXX.pem
-rw-r--r--@ 1 example-user  staff  Jan 19 15:42 Certificate-GCP-CVO-
20230119-0XXXXX.pem
-rw-r--r--@ 1 example-user  staff  Jan 19 17:02 CVO-GCP-pubkey.pem
-rw-r--r--@ 1 example-user  staff  Jan 19 15:42 GCP_CVO_20230119-
XXXXXX_digest.sig
-rw-r--r--@ 1 example-user  staff  Jan 19 16:39 disk.raw

# Step 3 - Decrypt the signature using the extracted public key and
verify the contents of the downloaded disk.raw
$ openssl dgst -verify (public_key) -keyform PEM -sha256 -signature
(signed digest) -binary (downloaded or obtained disk.raw)
$ openssl dgst -verify CVO-GCP-pubkey.pem -keyform PEM -sha256
-signature GCP_CVO_20230119-XXXXXX_digest.sig -binary disk.raw
Verified OK

# A failed response would look like this
$ openssl dgst -verify CVO-GCP-pubkey.pem -keyform PEM -sha256
-signature GCP_CVO_20230119-XXXXXX_digest.sig -binary
../sample_file.txt
Verification Failure
```

저작권 정보

Copyright © 2024 NetApp, Inc. All Rights Reserved. 미국에서 인쇄된 본 문서의 어떠한 부분도 저작권 소유자의 사전 서면 승인 없이는 어떠한 형식이나 수단(복사, 녹음, 녹화 또는 전자 검색 시스템에 저장하는 것을 비롯한 그래픽, 전자적 또는 기계적 방법)으로도 복제될 수 없습니다.

NetApp이 저작권을 가진 자료에 있는 소프트웨어에는 아래의 라이센스와 고지사항이 적용됩니다.

본 소프트웨어는 NetApp에 의해 '있는 그대로' 제공되며 상품성 및 특정 목적에의 적합성에 대한 명시적 또는 묵시적 보증을 포함하여(이에 제한되지 않음) 어떠한 보증도 하지 않습니다. NetApp은 대체품 또는 대체 서비스의 조달, 사용 불능, 데이터 손실, 이익 손실, 영업 중단을 포함하여(이에 국한되지 않음), 이 소프트웨어의 사용으로 인해 발생하는 모든 직접 및 간접 손해, 우발적 손해, 특별 손해, 징벌적 손해, 결과적 손해의 발생에 대하여 그 발생 이유, 책임론, 계약 여부, 엄격한 책임, 불법 행위(과실 또는 그렇지 않은 경우)와 관계없이 어떠한 책임도 지지 않으며, 이와 같은 손실의 발생 가능성이 통지되었다 하더라도 마찬가지입니다.

NetApp은 본 문서에 설명된 제품을 언제든지 예고 없이 변경할 권리를 보유합니다. NetApp은 NetApp의 명시적인 서면 동의를 받은 경우를 제외하고 본 문서에 설명된 제품을 사용하여 발생하는 어떠한 문제에도 책임을 지지 않습니다. 본 제품의 사용 또는 구매의 경우 NetApp에서는 어떠한 특허권, 상표권 또는 기타 지적 재산권이 적용되는 라이센스도 제공하지 않습니다.

본 설명서에 설명된 제품은 하나 이상의 미국 특허, 해외 특허 또는 출원 중인 특허로 보호됩니다.

제한적 권리 표시: 정부에 의한 사용, 복제 또는 공개에는 DFARS 252.227-7013(2014년 2월) 및 FAR 52.227-19(2007년 12월)의 기술 데이터-비상업적 품목에 대한 권리(Rights in Technical Data -Noncommercial Items) 조항의 하위 조항 (b)(3)에 설명된 제한사항이 적용됩니다.

여기에 포함된 데이터는 상업용 제품 및/또는 상업용 서비스(FAR 2.101에 정의)에 해당하며 NetApp, Inc.의 독점 자산입니다. 본 계약에 따라 제공되는 모든 NetApp 기술 데이터 및 컴퓨터 소프트웨어는 본질적으로 상업용이며 개인 비용만으로 개발되었습니다. 미국 정부는 데이터가 제공된 미국 계약과 관련하여 해당 계약을 지원하는 데에만 데이터에 대한 전 세계적으로 비독점적이고 양도할 수 없으며 재사용이 불가능하며 취소 불가능한 라이센스를 제한적으로 가집니다. 여기에 제공된 경우를 제외하고 NetApp, Inc.의 사전 서면 승인 없이는 이 데이터를 사용, 공개, 재생산, 수정, 수행 또는 표시할 수 없습니다. 미국 국방부에 대한 정부 라이센스는 DFARS 조항 252.227-7015(b)(2014년 2월)에 명시된 권한으로 제한됩니다.

상표 정보

NETAPP, NETAPP 로고 및 http://www.netapp.com/TM에 나열된 마크는 NetApp, Inc.의 상표입니다. 기타 회사 및 제품 이름은 해당 소유자의 상표일 수 있습니다.