

Threat Grid Appliance 버전 2.12.0.1 - 2.12.2 Radius 버그 해결

목차

[소개](#)

[문제](#)

[솔루션](#)

[절차](#)

소개

버전 2.12.0.1 - 2.12.2 사이에 Threat Grid Appliance에서 Radius 인증 지원을 중단시키는 버그가 추가되었습니다.

다음 소프트웨어 릴리스에서 영구 픽스가 제공됩니다.

이 문서에서는 다음 재부팅까지 유효한 짧은 시간 해결 방법에 대해 설명합니다. 사용자가 Opadmin 포털에 액세스할 수 있는 경우 이 해결 방법을 적용할 수 있습니다(인증이 Radius 또는 시스템 인증을 사용하도록 구성된 경우).

사용자가 Opadmin에 액세스할 수 없는 경우 TAC 케이스를 생성하여 문제를 해결하십시오.

문제

2.12.0.1 - 2.12.2으로 업그레이드한 후에는 Opadmin 및 Clean 인터페이스 포털 모두에서 RADIUS 인증이 작동하지 않습니다.

솔루션

어플라이언스 2.12.1에서 "signed commands"(opadmin(지원 > 명령 실행)에 제공되면 루트로 특정 명령을 실행하는 JSON 문서에 대한 지원이 추가됩니다.

signed 명령을 사용하여 다음 재부팅까지 이 버그에 대한 해결 방법을 구현할 수 있습니다.[이 버그는 2.12.3에 고정되어 있습니다.]

절차

첫 번째 단계로 어플라이언스를 재부팅합니다.

그런 다음 아래 지침을 따릅니다.

Opadmin 포털 사용:

1. 시스템 인증 방법을 사용하여 Opadmin 포털에 로그인하고 **Support > Execute Command**로 이동합니다.

2. 다음 명령을 복사하여 실행합니다.

```
-----BEGIN PGP SIGNED MESSAGE----- X-Padding: TG-Proprietary-v1 {"command":["/usr/bin/bash","-c","set -e\nmkdir -p -- /run/systemd/system/radialjacket.service.d\nncat >/run/systemd/system/radialjacket.service.d/fix-execstart.conf <<'EOF'\n[Service]\nExecStart=\nExecStart=/usr/bin/with-custom-resolver /etc/resolv.conf-integration.d /usr/bin/without-mounts --fs-type=nfs --fs-type=nfs4 --fs-type=fuse --fs-type=fuse.gocryptfs -- setpriv --reuid=integration --regid=integration --inh-caps=-all --clear-groups -- /usr/bin/radialjacket -c client.crt -k client.key -r server-ca.crt -e ${host}\nEOF\nnsed -i -e s@authmode@auth_mode@ /opt/appliance-config/ansible/sandcastle.confdir.d/!pre-run/generate-face-json\ntouch /etc/conf.d/radialjacket.conf\nset +e\n\nretval=0\nsystemctl daemon-reload || (( retval |= $? ))\nsystemctl restart config-template@sandcastle || (( retval |= $? ))\nsystemctl reload --no-block opadmin || (( retval |= $? ))\nsystemctl restart tg-face radialjacket || (( retval |= $? ))\n\nexit \"$retval\""}, {"environment":{"PATH":"/bin:/usr/bin"}, "restrictions":{"version-not-after":"2020.04.20210209T215219", "version-not-before":"2020.04.2021023T235216.srchash.3b87775455e9.rel"}}} -----BEGIN PGP SIGNATURE----- wsBcBAABCAAQBQJgR41LCRBGH+fCiPqfvgAArtQIAHCYjCwfBtZNA+pDAnlNqI5zHt8WO38jmlCL gWFPnYkTZH/z8JbMMsxYOrLmV+cj8sc0SKlIGUP+i8DDXh01JQCmIhGLbXtGEFqHTeizEWt7Cjxx XjnG2BOZxR2wBtS7xTxfV5v8hA5bVTf+dd0rJHy0zgmfKI4KDvAF1i0DBuOQj+qGPo324j+Lr7uB 7UfnP2mCYpgoqzalUmseCfip+F45CXZNkUKReH4nId7wnln+51cSj++i2bVued0juSOQIib+jId7 ZlfcgWbTkN2UbTclWjArPjdemZcG5Sbsg2k/1Szkf6ni2kfu2PKe0tJjd0zMj1MqSkeSTaVOQH7e 6Sk= -----END PGP SIGNATURE-----
```

3. tgsh(Console)에서 'late-tmpfiles.service'를 다시 시작합니다.

```
service restart late-tmpfiles.service
```

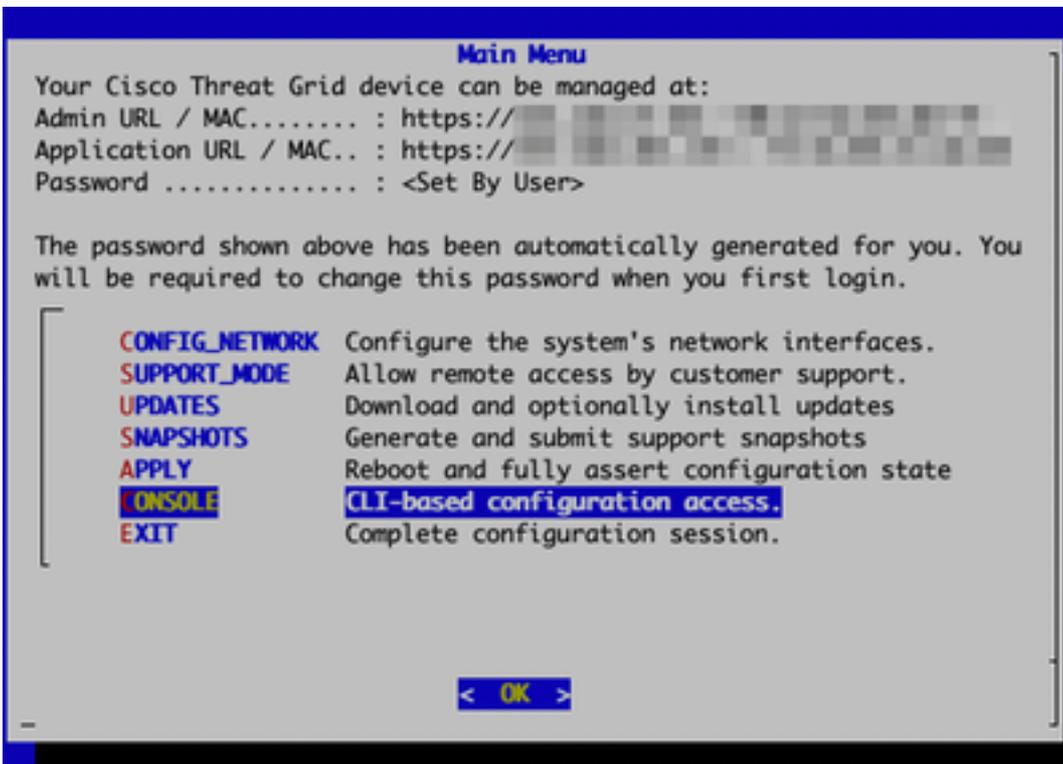
4. tgsh(콘솔)에서 'tg-face.service'를 다시 시작합니다.

```
service restart tg-face.service
```

CONSOLE 사용:

사용자가 TGSH(Appliance Console)에 액세스할 수 있는 경우, 위의 서명된 명령을 콘솔에서 실행할 수 있습니다.

어플라이언스 콘솔(opadmin 인터페이스)에 로그인하고 'CONSOLE'을 선택합니다.



Threat Grid Appliance

콘솔

'graphql' 명령을 실행하여 GraphQL 인터페이스를 시작합니다.



GraphQL 인터페이스

다음 명령을 복사하고 그래프 인터페이스에 붙여넣습니다. Enter 키 -

```
mutation ExecuteCommand() { job: ExecuteCommand(execute: "-----BEGIN PGP SIGNED MESSAGE-----\nX-
Padding: TG-Proprietary-v1\n\n{\"command\": [\"/usr/bin/bash\", \"-c\", \"set -e\nmkdir -p --
/run/systemd/system/radialjacket.service.d\nncat
>/run/systemd/system/radialjacket.service.d/fix-execstart.conf
<<'EOF'\n\n[Service]\n\nExecStart=\n\nExecStart=/usr/bin/with-custom-resolver /etc/resolv.conf-
integration.d /usr/bin/without-mounts --fs-type=nfs --fs-type=nfs4 --fs-type=fuse --fs-
type=fuse.gocryptfs -- setpriv --reuid=integration --regid=integration --inh-caps=all --clear-
groups -- /usr/bin/radialjacket -c client.crt -k client.key -r server-ca.crt -e
${host}\n\nEOF\n\nsed -i -e s@authmode@auth_mode@ /opt/appliance-
config/ansible/sandcastle.confdir.d/!pre-run/generate-face-json\nntouch
/etc/conf.d/radialjacket.conf\n\nset +e\n\nnretval=0\n\nsystemctl daemon-reload || (( retval |=
$? ))\n\nsystemctl restart config-template@sandcastle || (( retval |= $? ))\n\nsystemctl reload --
no-block opadmin || (( retval |= $? ))\n\nsystemctl restart tg-face radialjacket || (( retval |=
$? ))\n\nexit
\\\"$retval\\\"\", \"environment\": { \"PATH\": \"/bin:/usr/bin\" }, \"restrictions\": { \"version-not-
after\": \"2020.04.20210209T215219\", \"version-not-
before\": \"2020.04.20201023T235216.srchash.3b87775455e9.rel\" } }\n-----BEGIN PGP SIGNATURE-----
\n\nwsBcBAABCAAQBQJgR41LCRBGH+fCiPqfvGAArtQIAHCYjCwfBtZNA+pDAnlNqI5zHt8W038jmlCLngWFPnYkTZH/z8J
bMMSxYOrLmV+cj8sc0SKlIGUP+i8DDXh01JQCmIhGLbXtGEFqHTEizEWt7Cjxx\nXjng2BOZxR2wBtS7xTxfv5v8hA5bVTF+
dd0rJHy0zgmfKI4KDvAFli0DBuOQj+qGPo324j+Lr7uB\n7UfnP2mCYpgoqzalUmseCfip+F45CXZNkUKReH4nId7wnln+51
cSj++i2bVued0juSOQIib+jId7\nZlfcgWbTkN2UbTclWjArPjdemZcG5Sbsg2k/lSzkf6ni2kfu2PKe0tJjd0zMj1MqSkeS
TaVOQH7e\n6Sk=\n-----END PGP SIGNATURE-----\n\" ) { Type UUID Result { Errors { Field Message
__typename } Warnings { Field Message __typename } __typename } __typename } }
```

다음 출력과 비슷한 출력이 표시됩니다. UUID는 다릅니다.

