تكوين وكيل NGINX العكسي للوصول دون الحاجة إلى VPN إلى Cisco Finesse (12.6 ES03)

المحتويات

المقدمة

المتطلبات الأساسية

المتطلبات

المكونات المستخدمة

معلومات أساسية

التغييرات في ES03

ملاحظات الترقية للتكوينات الخالية من الشبكة الخاصة الظاهرية (VPN) القائمة على الطراز ES01

المصادقة

المصادقة بخلاف SSO

مصادقة SSO

مصادقة اتصالات WebSocket

منع هجوم القوة الغاشمة

التسجيل

تثبيت Fail2ban وتكوينه

التحقق من صحة عناوين URL الخاصة بالمورد الثابت

عناوين Caching

التكوين

تكوين مكونات الحل للوصول الأقل إلى VPN

تثبيت OpenRest كوكيل عكسي في DMZ

تثبيت OpenResty

تكوين NGINX

تكوين ذاكرة التخزين المؤقت ل Nginx

تكوين شهادات SSL

استخدام معلمة Diffie-Hellman المخصصة

تأكد من تمكين تدبيس OCSP - التحقق من إبطال الشهادة

تكوين NGINX

تكوين منفذ الوكيل العكسي

تكوين مصادقة TLS المتبادلة بين الوكيل العكسي ومكونات تدفق البيانات

مسح ذاكرة التخزين المؤقت

إرشادات قياسية

تكوين ملف التعيين

استخدام الوكيل العكسي كخادم ملف التعيين

تقوية نواة CentOS 8

تقوية IPtables

تقييد اتصالات العميل

حظر اتصالات العملاء

حظر عناوين IP المميزة

حظر نطاق عناوين IP

حظر جميع عناوين IP في شبكة فرعية

سيلينكس
التحقق من الصحة
Finesse
البيانات المجسمة والمعروضة مباشرة
IDS
الأداء
استكشاف الأخطاء وإصلاحها
وس

المقدمة

يصف هذا المستند كيفية إستخدام وكيل عكسي للوصول إلى سطح المكتب Cisco Finesse دون الاتصال بشبكة VPN تستند إلى إصدارات 12.6 ES03 من Cisco Finesse، و Cisco Unified Intelligence Center (CUIC)، و Cisco Identity Service (IDs).

ملاحظة: لا تدعم Cisco تثبيت NGINX وتكوينه. يمكن مناقشة الاستعلامات المتعلقة بهذا الموضوع على منتديات مجتمع Cisco.

ملاحظة: بالنسبة لعمليات نشر الطراز ES03 التي تتميز بأنها لا تحتوي على شبكات VPN، يمكنك الاطلاع على قائمة المكونات الفردية لتخطيط عمليات الترقية والتحقق من قيود التوافق. Cisco Finesse 12.6 ES03 Readme، CUIC / IDs 12.6 ES03 Readme

المتطلبات الأساسية

المتطلبات

توصي Cisco بأن تكون لديك معرفة بالمواضيع التالية:

- إصدار Cisco Unified Contact Center Enterprise (UCCE)
- Cisco Finesse
- إدارة لينكس
- إدارة الشبكة وإدارة شبكة لينوكس

المكونات المستخدمة

تستند المعلومات الواردة في هذا المستند إلى إصدارات البرامج والمكونات المادية التالية:

- Finesse - 12.6 ES03
- CUIC - 12.6 ES03
- IDs - 12.6 ES03
- UCCE/Hosted Collaboration Solution (HCS) لمركز الاتصال (CC) - الإصدار 11.6 أو إصدار أحدث
- Packaged Contact Center Enterprise (PCCE) - الإصدار 12.5 أو إصدار أحدث

ملاحظة: يلزم أن تكون عمليات نشر PCCE/UCCE 2k في الإصدار 12.6 من CCE ارظن لنشر LD/CUIC المشترك

تم إنشاء المعلومات الواردة في هذا المستند من الأجهزة الموجودة في بيئة معملية خاصة. بدأت جميع الأجهزة المُستخدمة في هذا المستند بتكوين ممسوح (افتراضي). إذا كانت شبكتك قيد التشغيل، فتأكد من فهمك للتأثير المحتمل لأي أمر.

ملاحظة: تم تكوين التكوين المتوفر في هذا المستند وتصليحه واختبار التحميل باستخدام وكيل NGINX العكسي (OpenRest) الذي تم نشره على CentOS 8.0، مقابل نشر نموذج 2000 لمستخدم UCCE. تتوفر معلومات مرجع ملف تعريف الأداء في هذا المستند.

معلومات أساسية

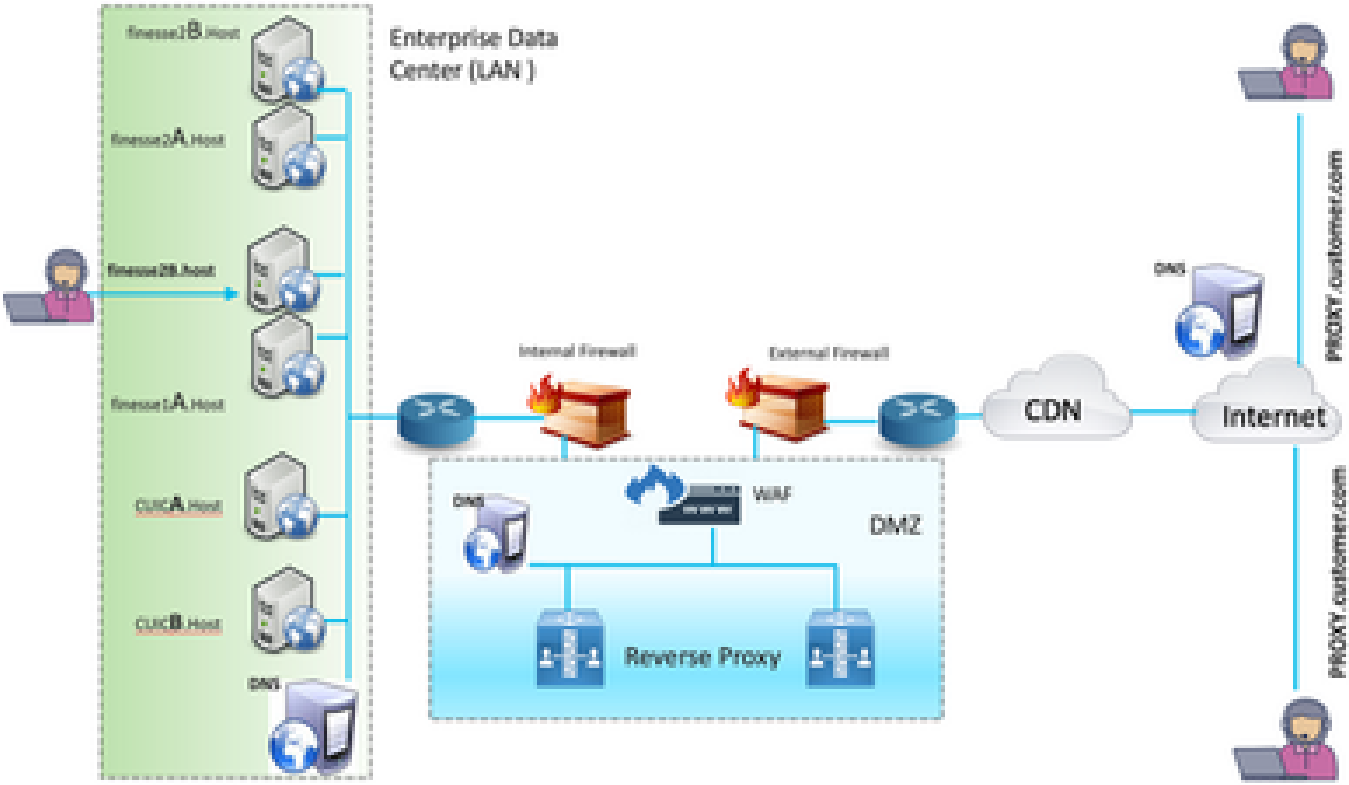
نموذج النشر هذا مدعوم لـ UCCE/PCCE و HCS لحلول UCCE.

يتم دعم نشر وكيل عكسي (متوفر من 12.6 ES01) كخيار للوصول إلى سطح المكتب Cisco Finesse دون الاتصال بشبكة VPN. توفر هذه الميزة المرونة للعملاء في الوصول إلى سطح المكتب Finesse من أي مكان عبر الإنترنت.

لتمكين هذه الميزة، يجب نشر وجوكيل عكسي في المنطقة المجردة من السلاح (DMZ).

يظل الوصول إلى الوسائط دون تغيير في عمليات نشر الوكيل العكسي. للاتصال بالوسائط، يمكن للعملاء إستخدام Cisco Jabber عبر حل الوصول عن بعد وعن بعد (MRA) أو شبكة هاتف محولة عامة (PSTN) أو باستخدام شبكة UCCE المحمول العميل لبرنامج قدرة متنقلة. يوضح هذا المخطط كيف سيدب نشر الشبكة عند الوصول إلى مجموعتي دقة الوكيل العكسي (HA) من خلال زوج واحد عالي التوفر من CUIC وعقدتين من Finesse.

يتم دعم الوصول المتزامن من الوكلاء على الإنترنت والوكلاء الذين يتصلون من شبكة LAN كما هو موضح في هذه الصورة.



ملاحظة: انظر دليل المميزات الخاص بمعايير تحديد وكيل جهة خارجية بدلا من Nginx لدعم هذا النشر.

- دليل مميزة UCCE 12.6 - يوفر نظرة عامة على المميزات وتصميما، بالإضافة إلى تفاصيل التكوين لمميزة عدم وجود شبكة خاصة ظاهرية (VPN).
- دليل أمان UCCE 12.6 - يوفر إرشادات تكوين الأمان لنشر وكيل العكسي.

من المستحسن مراجعة قسم ال يحتوي على شبكات VPN من دليل المميزات ودليل الأمان قبل قراءة هذا المستند.

التغييرات في ES03

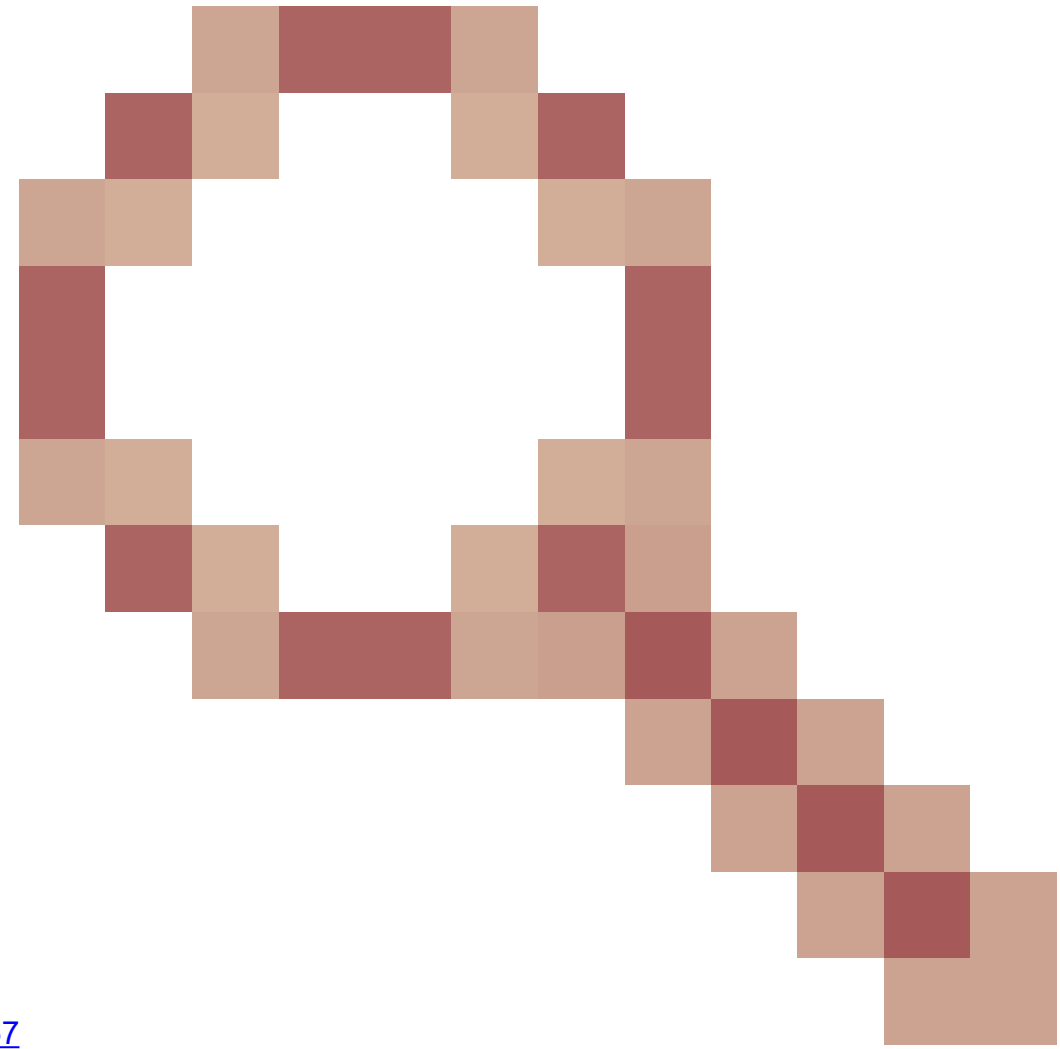
- المميزات الجديدة
  - يتم الآن دعم قدرات المشرف الفاخر عبر الوكيل العكسي.
  - يتم الآن دعم تقارير RealTime و History من CUIC من خلال أدوات Finesse في بيئة وكيل.
  - المصادقة لجميع الطلبات / الاتصالات - تتطلب دعم LOA
    - تتم مصادقة جميع طلبات Finesse / CUIC / IM & Presence (IM&P) في الوكيل قبل السماح لها بإدخال مركز البيانات.
    - كما تم تقييد إتصالات WebSocket و Live Data SocketIO والسماح بها فقط من العملاء الذين قاموا بنجاح بتقديم طلب آمن إلى Finesse.
    - إستشعار هجوم القوة القاسية والتسجيل في الوكيل، والذي يمكن إستخدامه مع Fail2BAN لحظر عناوين IP الضارة.

- تحسينات الأمان لتكوين الوكيل العكسي - تتطلب دعم LOA
  - مصادقة أمان طبقة النقل المتبادل (TLS) بين الوكيل العكسي ومكونات الخادم (Finesse/ID/CUIC/LiveAta).
  - إعدادات SeLinux.
  - قم بتمكين التحقق من الثقة المتبادل لطبقة مآخذ التوصيل الآمنة (SSL) لطلبات الوكيل وخادم المكونات.

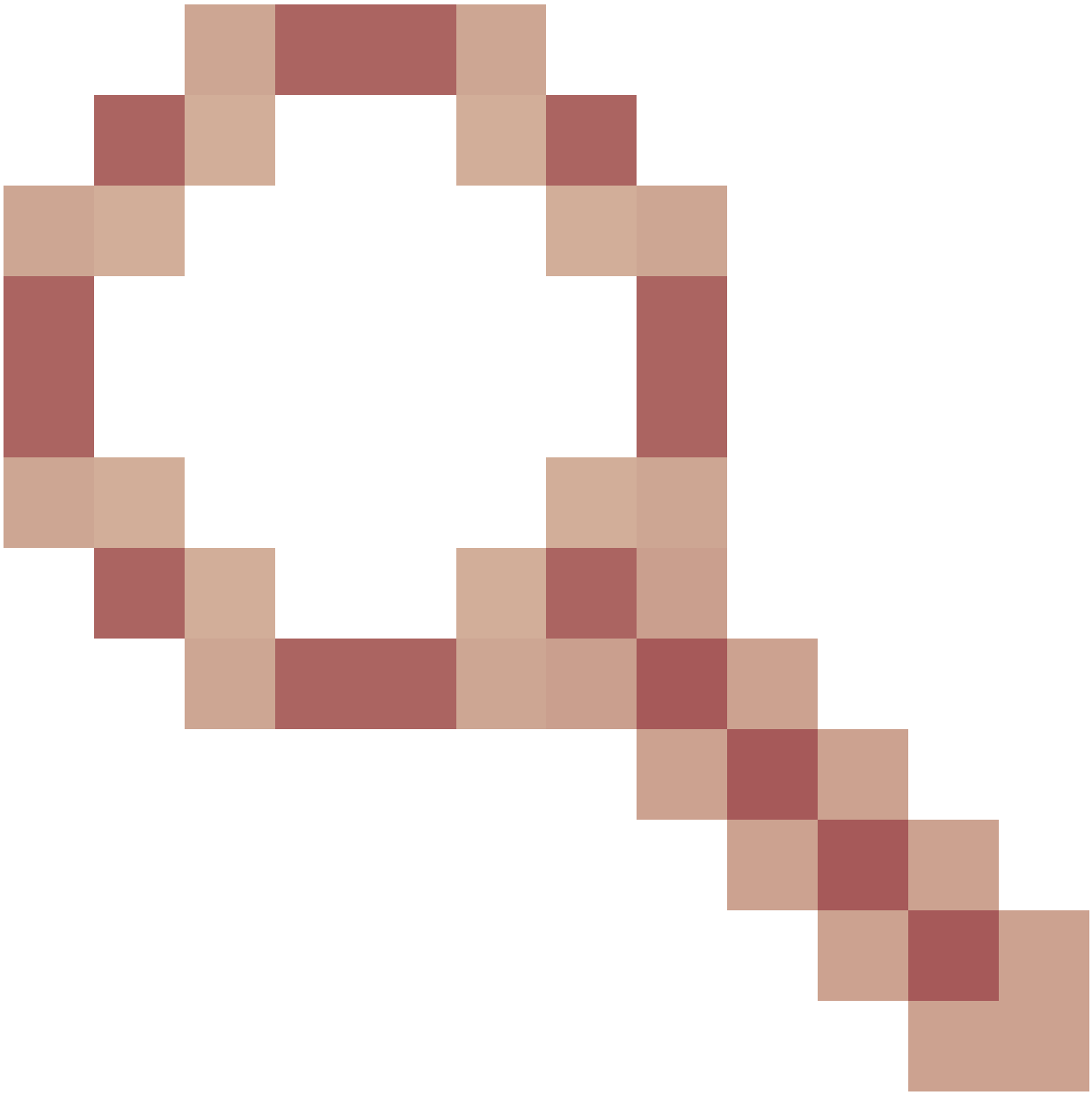
- يتطلب الأمان المحسن لتكوين الوكيل لمنع هجمات قطع الخدمة (DoS) / رفض الخدمة الموزع (DDoS) - دعم LOA
  - حدود معدل طلب Nginx المحسنة لمختلف أجزاء النظام.
  - حدود المعدل ل IpTables.
  - التحقق من طلبات المواراد الثابتة قبل طلب خادم مكون تدفق البيانات.
  - صفحات أخف وزنا والا يمكن مصادقتها ال تصل إلى خادم مكون تدفق البيانات.

- ميزات أخرى متنوعة - تتطلب دعم LUA
  - استجابات مشاركة المواراد عبر المنشأ بالاستشعار التلقائي التي يوفرها الوكيل للمساعدة في التكوين التلقائي وتحسين الأداء

- إصلاح العيوب المتعلقة بميزة أقل من الشبكة الخاصة الظاهرية (VPN)

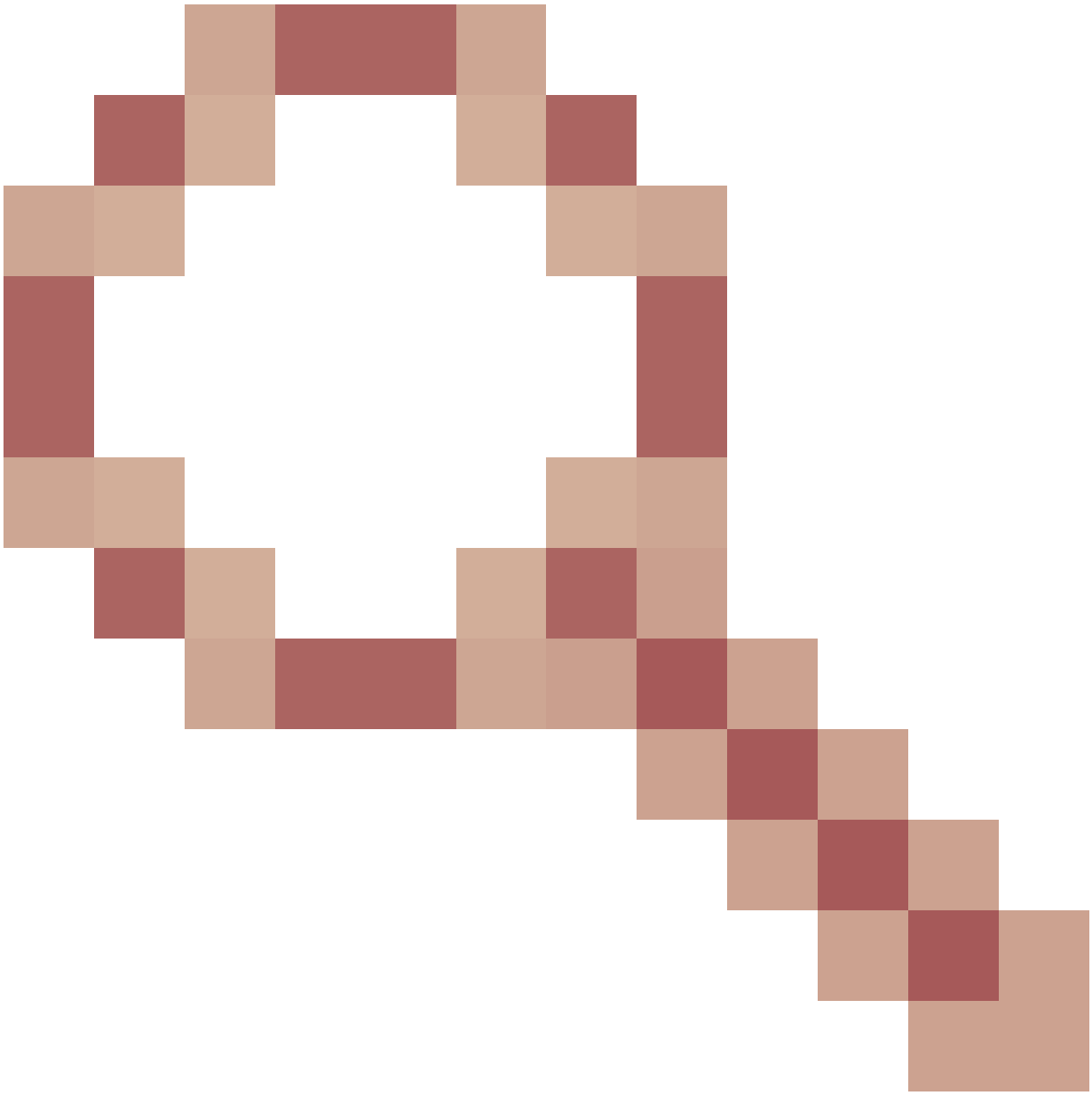


  - CSCwa26057
CSCwa26057

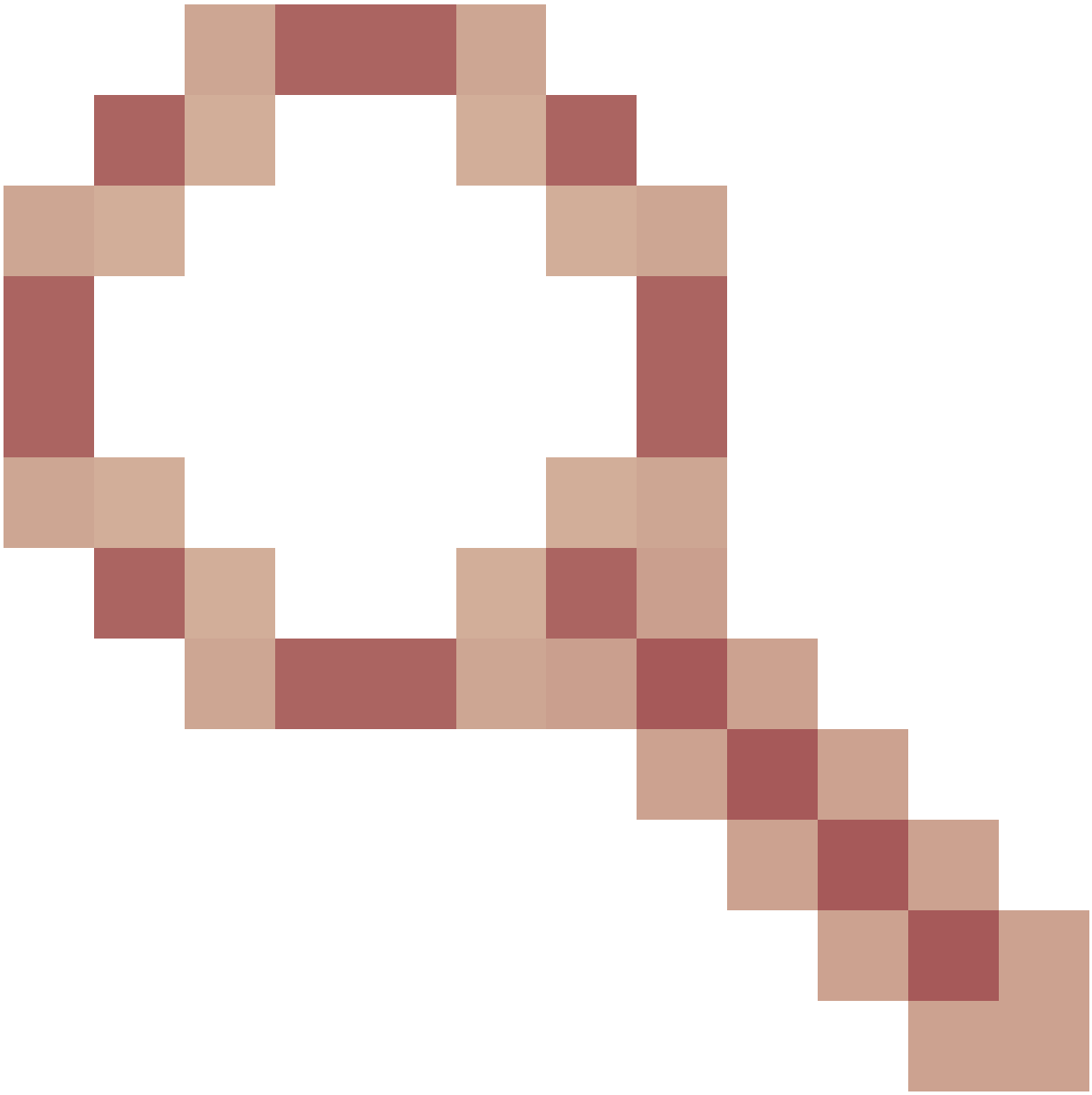


" /> - شهادات متعددة يتم تقديمها إلى الوكيل أثناء الدخول الدقيق إلى سطح المكتب

- CSCwa24471

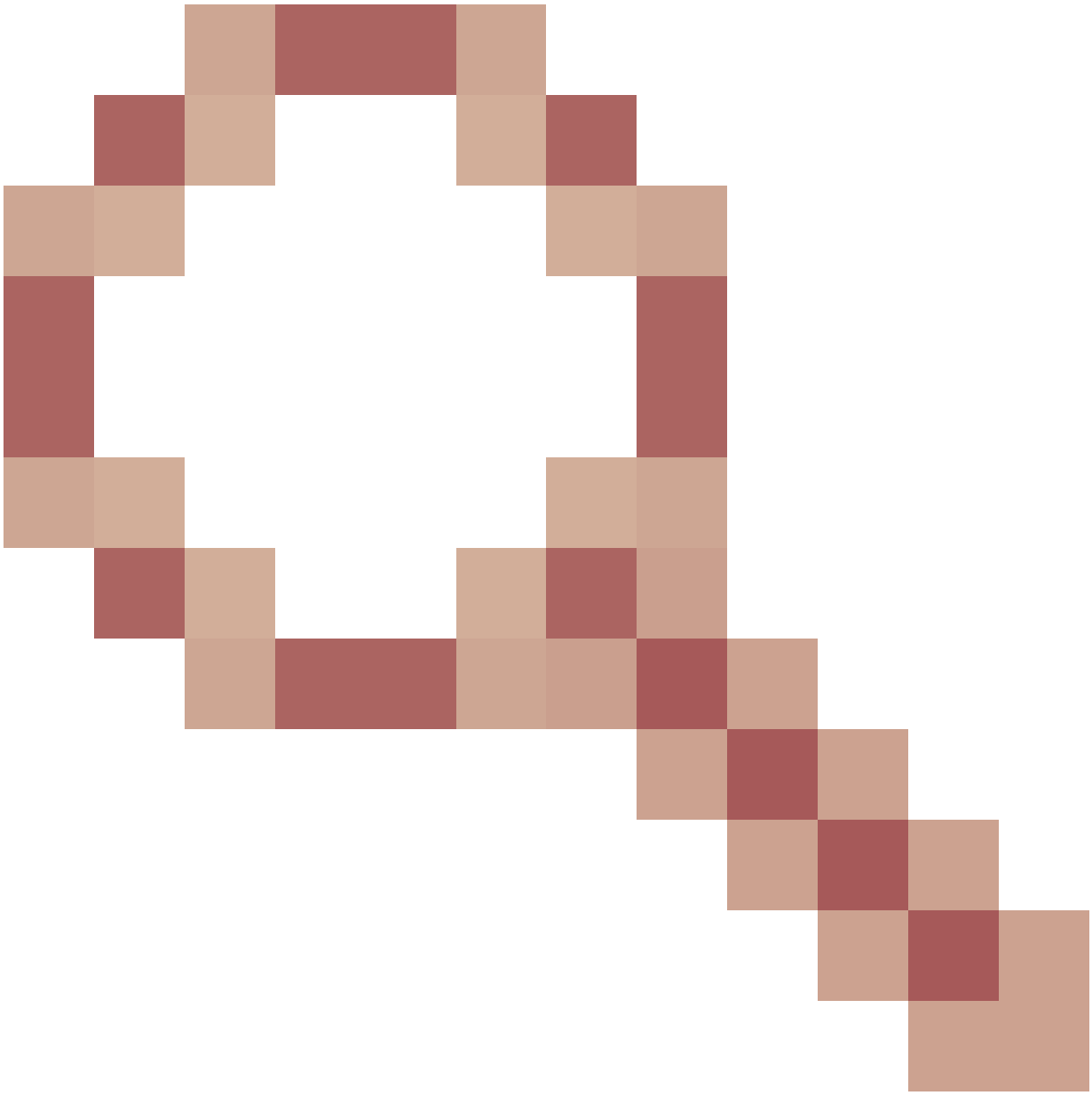


- لا تظهر صفحة تسجيل الدخول إلى FindSE اسم FQDN لوكيل SSO
  - CSCwa24519



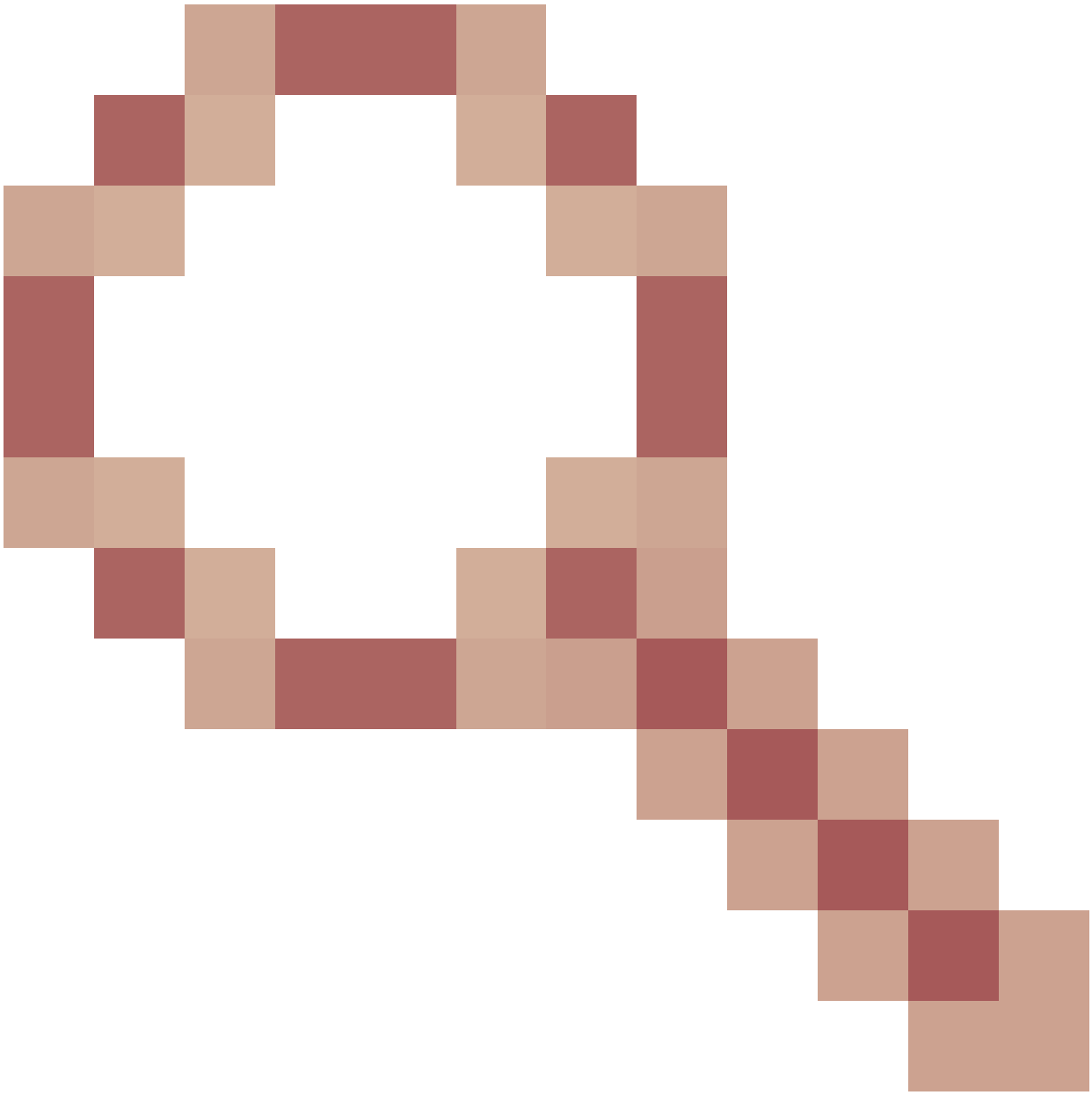
فشل إعادة تشغيل خدمة وكيل الويب إذا لم يكن اسم المضيف العكسي للوكيل قابلا للحل من المكون

- CSCwa23252



: يتم كسر ثقة النبض الوكيل عندما يكون العمق أكثر من واحد لسلسلة الشهادات المصدقة

- CSCwa46459



تعرض نقاط الضعف في يوم Log4j صفر في خدمة الويب

ملاحظات الترقية للتكوينات الخالية من الشبكة الخاصة الظاهرية (VPN) القائمة على الطراز ES01

- يتطلب تكوين ES03 تثبيت NGINX مع دعم LUA.
- متطلبات الشهادة
  - يتطلب Cisco Finesse و CUIC و IdS إضافة شهادة مضيف Nginx / OpenRest إلى مخزن Tomcat الموثوق وإعادة التشغيل التي تم تنفيذها، قبل أن يتمكن تكوين Nginx ES02 من الاتصال بخادم الخادم الخادم الخادم بنجاح.
  - يلزم تكوين شهادات خادم تدفق البيانات من الخادم Cisco Finesse و CUIC و IDs إلى الخادم Nginx لاستخدام التكوين المستند إلى ES03.

ملاحظة: يوصى بإزالة تكوين NGINX القائم على ES01 قبل تثبيت تكوينات ES03 NGINX.

ملاحظة: تتطلب برامج تكوين ES03 أيضا تثبيت ES03 COP المتوافق في Cisco Finesse و CUIC و IDs.

المصادقة

تقدم Finesse 12.6 ES03 المصادقة في الوكيل. يتم دعم المصادقة لتسجيل الدخول الأحادي (SSO) وعمليات النشر بخلاف SSO.

قبل الوكيل يتم قبولها في الوكيل يتم فرض المصادقة لجميع الطلبات والبروتوكولات التي يتم فرضها بواسطة إعادة توجيهها إلى خوادم مكونات الخادم، حيث يتم إجراء المصادقة التي يتم فرضها بواسطة مكونات الخادم، حيث يتم إجراء المصادقة التي يتم فرضها بواسطة إعادة توجيهها إلى خوادم مكونات الخادم، حيث يتم إجراء المصادقة التي يتم بيانات اعتماد تسجيل الدخول إلى خوادم المكونات محليا أيضا. تستخدم جميع المصادقة بيانات اعتماد تسجيل الدخول إلى Finesse الشائعة لمصادقة الطلبات.

تتم مصادقة الاتصالات المتواصلة، مثل مقابس الويب التي تعتمد على بروتوكولات مثل بروتوكول التواجد (XMPP) للمصادقة والاتصال الإرسال، التطبيق مثل المراسلة الممتدة وبروتوكول التواجد (XMPP) للمصادقة والاتصال الإرسال، في الوكيل عن طريق التحقق من صحة عنوان IP الذي تم إجراء مصادقة تطبيق ناجحة منه قبل إنشاء اتصال مأخذ التوصيل.

المصادقة بخلاف SSO

لا تتطلب المصادقة بخلاف SSO أي تكوينات إضافية وستعمل مع البرامج النصية المطلوبة. بمجرد إجراء إستبدال البرامج النصية للتكوين Nginx. تعتمد المصادقة على اسم المستخدم وكلمة المرور المستخدمين لتسجيل الدخول إلى Finesse. سيتم التحقق من صحة الوصول إلى جميع نقاط النهاية باستخدام خدمات مصادقة Finesse.

يتم تخزين قائمة المستخدمين الصحيحين مؤقتا في الوكيل محليا (يتم تحديثها كل 15 دقيقة)، والتي يتم إستخدامها للتحقق من صحة المستخدم في التخزين المؤقت كل 15 دقيقة)، والتي يتم إستخدامها للتحقق من صحة المستخدم في الطلب. يتم التحقق من صحة بيانات اعتماد المستخدم عن طريق إعادة توجيه الطلب إلى URI الذي تم تكوينه من Finesse وبعد ذلك يتم تخزين تجزئة بيانات الاعتماد مؤقتا محليا (مخزنة مؤقتا لمدة 15 دقيقة) لمصادقة الطلبات الجديدة محليا. في حالة حدوث أي تغيير على اسم المستخدم أو كلمة المرور، سيتم تنفيذها فقط بعد 15 دقيقة.

مصادقة SSO

تتطلب مصادقة SSO أن يقوم المسؤول بتكوين مفتاح تشفير رمز IDs المميز في خادم Nginx لملف التكوين. يمكن الحصول على مفتاح تشفير رمز IdS المميز من خادم IdS باستخدام أمر CLI show ids secret. يجب تكوين المفتاح كجزء من أحد الحلول #Must-change التي يجب أن يؤديها المسؤول في البرامج النصية قبل أن تعمل مصادقة SSO.

ارجع إلى دليل مستخدم SSO لتكوينات SAML للمعرفات التي سيتم إجراؤها لدقة الوكيل للعمل مع المعرفات.

وبمجرد تكوين مصادقة SSO، يمكن إستخدام زوج من العلامات المميزة الصالحة للوصول إلى أي من نقاط النهاية في النظام. يتحقق تكوين الوكيل من بيانات الاعتماد من خلال اعتراض طلبات إسترداد الرمز المميز التي تم إجراؤها على المعرفات أو من خلال فك تشفير الرموز المميزة الصالحة ثم تخزينها مؤقتا محليا لمزيد من عمليات التحقق من الصحة.

مصادقة إتصاالت WebSocket

ال يمكن مصادقة إتصاالت WebSocket باستخدام رأس التخويل القياسي، نظرا ألن الروسو WebSocket األصلية في المستعرض. ذ تنفيذ عمليات قبل من قدمة ريغ مصصخملا في الوارد المصادقة معلومات ال تمنع حيث التطبيق، مستوى مصادقة بروتوكوالت WebSocket، وبالتالي يمكن للكيانات الضارة تقديم هجمات فر ض الخدمة (DoS) أو فرض الخدمة (DDoS) من خالل إنشاء إتصاالت ال تعد وال تحصى إلرباك النظام.

من أجل تقليل هذا االحتمال، تحتوي تكوينات الوكيل العكسي ل NGINX التي تم توفيرها على عمليات تحقق محددة للسماح بقبول إتصاالت مأخذ ويب فقط من عناوين IP هذه التي أن تمت مصادقتها بنجاح إلجراء طلب REST قبل إنشاء اتصال مأخذ ويب. وهذا يعني أن العمالء الذين يحاولون إنشاء إتصاالت مأخذ ويب، قبل إصدار طلب REST، سيحصلون اآلن على خطأ فشل التفويض وليس سيناريو إستخدام مدعوما.

منع هجوم القوة الغاشمة

تمنع البرامج النصية للمصادقة Finesse 12.6 ES02 بشكل فعال الهجمات العنيفة التي يمكن إستخدامها لتخمين كلمة مرور المستخدم. وهو يقوم بذلك من خالل حظر عناوين IP المستخدمة للوصول إلى الخدمة، بعد عدد معين من المحاوالت الفاشلة في وقت قصير. سيتم رفض هذه الطلبات بواسطة خطأ العميل 418. يمكن الوصول إلى تفاصيل عناوين IP المحظورة من الملفات <nginx-install-directory>/logs/blocking.log و<nginx-install-directory>/logs/error.log.

يمكن تكوين عدد الطلبات الفاشلة والفاصل الزمني ومدة الحظر. التكوينات موجودة في ملف <nginx-install-directory>/conf/conf.d/maps.conf.

```
## These two constants indicate five auth failures from a client can be allowed in thirty seconds.
## if  the threshold is crossed,client ip will be blocked.
map $host $auth_failure_threshold_for_lock {
    ## Must-change Replace below two parameters as per requirement
    default 5 ;
}

map $host $auth_failure_counting_window_secs {
    ## Must-change Replace below two parameters as per requirement
    default 30;
}

## This indicates duration of blocking a client to avoid brute force attack
map $host $ip_blocking_duration {
    ## Must-change Replace below parameter as per requirement
    default 1800;
}
```

التسجيل

للعثور على عناوين IP المحظورة، قم بتشغيل األوامر التالية من الدليل <nginx-install-

directory>/log.

```
grep "will be blocked for" blocking.log
grep "IP is already blocked." error.log
```

```
2021/10/29 17:30:59 [emerg] 1181750#1181750: *19 [lua] block_unauthorized_users.lua:153:
_redirectAndSendError(): 10.68.218.190 will be blocked for 30 minutes for exceeding retry limit.,
client: 10.68.218.190, server: saproxy.cisco.com, request:
"GET /finesse/api/SystemInfo?nocache=1636456574482 HTTP/2.0", host: "saproxy.cisco.com:8445",
referrer: "https://saproxy.cisco.com:8445/desktop/container/?locale=en_US&"

2021/10/29 19:21:00 [error] 943068#943068: *43 [lua] block_unauthorized_users.lua:53: 10.70.235.30 ::
IP is already blocked..., client: 10.70.235.30, server: saproxy.cisco.com, request:
"GET /finesse/api/SystemInfo?nocache=1635591686497 HTTP/2.0", host: "saproxy.cisco.com:8445",
referrer: "https://saproxy.cisco.com:8445/desktop/container/?locale=en_US"
```

يوصى بأن يتكامل العملاء مع Fail2BAN أو ما شابه ذلك بالإضافة الحظر إلى قواعد IPtable/جدار الحماية.

تثبيت Fail2ban وتكوينه

يمسح Fail2ban ملفات السجل ويحظر IPs أن يظهر العلامات الضارة - الكثير جدا من مرات الفشل كلمة المرور، يبحث عن مخارج، وما إلى ذلك. وبشكل عام، يتم إستخدام Fail2Ban بعد ذلك لتحديث قواعد جدار الحماية لرفض عناوين IP لفترة محددة من الوقت، مع أنه يمكن تكوين أي إجراء آخر تعسفي (على سبيل المثال، إرسال بريد إلكتروني). لمزيد من المعلومات، تفضل بزيارة موقع https://www.fail2ban.org/.

يمكن تكوين Fail2ban لمراقبة الحظر.log لتحديد عناوين IP التي تم حظرها بواسطة Nginx عند اكتشاف الهجمات العنيفة، وحظرها لمدة قابلة للتكوين. خطوات تثبيت Fail2ban وتكوينه على CentOS BackProxy هي كما يلي:

1. تثبيت Fail2ban باستخدام yum.

```
yum update && yum install epel-release
yum install fail2ban
```

2. إنشاء سجن محلي.

تتيح تكوينات السجن للمسؤول تكوين خصائص مختلفة مثل المنافذ التي سيتم منعها من الوصول إليها بواسطة أي عنوان IP محظور، والمدة التي يظل فيها عنوان IP محظورا، وتكوين عامل التصفية المستخدم لتعريف عنوان IP المحظور من ملف السجل المراقب، وما إلى ذلك. الخطوات الخاصة بإضافة تكوين مخصص لحظر عناوين IP التي يتم حظرها من الوصول إلى خوادم الويب هي كما يلي:

2.1. انتقل إلى دليل تثبيت Fail2ban (في هذا المثال /etc/fail2ban)

```
cd /etc/fail2ban
```

2.2. قم بعمل نسخة من prison.conf في السجن.local لإبقاء التغييرات المحلية معزولة.

```
cp jail.conf jail.local
```

2.3. أضف تكوينات السجن هذه إلى نهاية سجن الملف.محلي، واستبدل المنافذ الموجودة في القالب بأخرى فعلية. تحديث تكوينات وقت الحظر حسب الحاجة.

```
# Jail configurations for HTTP connections.
[finesse-http-auth]
enabled = true
# The ports to be blocked. Add any additional ports.
port = http,https,<finesse-ports>,<cuic-ports>,<any-other-ports-to-be-blocked>
# Path to nginx blocking logs.
logpath = /usr/local/openresty/nginx/logs/blocking.log
# The filter configuration.
filter = finesseban
# Block the IP from accessing the port, once the IP is blocked by lua.
maxretry= 1
# Duration for retry set to 3 mins. Doesn't count as the maxretry is 1
findtime= 180
# Lock time is set to 3 mins. Change as per requirements.
bantime = 180
```

3. تكوين عامل تصفية.

عامل تصفية يقول لـ Fail2ban ما يجب البحث عنه في السجلات لتحديد المضيف الذي يجب حظره. الخطوات الخاصة بإنشاء عامل تصفية هي كما يلي:

3.1. إنشاء عامل تصفية.d/finesseban.conf.

```
touch filter.d/finesseban.conf
```

3.2. أضف هذه الأسطر إلى عامل تصفية الملف.d/finesseban.conf.

```
[Definition]
# The regex match that would cause blocking of the host.
```

```
failregex = <HOST> will be blocked for
```

4. بدء Fail2ban.

قم بتشغيل هذا الأمر لبدء Fail2ban.

```
fail2ban-client start
```

فتح ملفات سجل Fail2ban والتحقق من عدم وجود أخطاء. بشكل افتراضي، تنتقل السجلات الخاصة ب failed2ban إلى الملف /var/log/fail2ban.log.

التحقق من صحة عناوين URL الخاصة بالمورد الثابت

ويتم بنشاط تتبع جميع نقاط النهاية الصالحة التي يمكن الوصول إليها بطريقة غير مصدق عليها في النصوص التنفيذية لنظام ES03.

يتم رفض الطلبات لهذه المسارات غير المصدق عليها بشكل نشط، في حالة طلب URI غير صالح، بدون إرسال هذه الطلبات إلى خادم الخادم.

عناوين Caching

عندما يكون طلب الخيارات الأولى ناجحا، يتم تخزين رؤوس الاستجابة لبيانات اعتماد السماح بالتحكم في الوصول، وبيانات التحكم في الوصول في الأصل، أو أساليب التحكم في الوصول، ورؤوس التعريض للتحكم في الوصول، وبيانات اعتماد السماح بالتحكم في الوصول مؤقتا لمدة خمس دقائق في الوكيل. يتم تخزين هذه الرؤوس مؤقتا لكل خادم دفق خاص.

التكوين

يصف هذا المستند تكوين NGINX كوكيل عكسي المراد إستخدامه لتمكين الوصول غير المحدود VPN. يتم توفير مكون حل UCCE والوكيل وإصدارات نظام التشغيل المستخدمة للتحقق من الإرشادات المقدمة. يجب تكييف التعليمات ذات الصلة مع نظام التشغيل/الوكيل الذي تختاره.

- إصدار Nginx المستخدم - OpenRest 1.19.9.1
- نظام التشغيل المستخدم للتكوين - CentOS 8.0

ملاحظة: يمكن تنزيل تكوين NGINX الموضح من صفحة تنزيل برنامج Finesse، الإصدار 12.6(1)ES3.

تكوين مكونات الحل للوصول الأقل إلى VPN

بعد تكوين الوكيل، قم بتكوين مكونات الحل (Finesse/ CUIC / IDs) للوصول الأقل إلى VPN

باستخدام اسم المضيف المخطط و IP للوكيل/الخدمات المستخدمة للوصول إلى الحل باستخدام هذه الأوامر.

```
utils system reverse-proxy allowed-hosts add
utils system reverse-proxy config-uri <uri> add
```

يمكن العثور على تفاصيل هذه الأوامر في دليل ميزة UCCE 12.6 ويجب الإشارة إليها قبل إستخدام هذا المستند.

تثبيت OpenRest كوكيل عكسي في DMZ

يوضح هذا القسم خطوات تثبيت الوكيل المستندة إلى OpenRest. عادة ما يتم تكوين الوكيل العكسي كجهاز مخصص في المنطقة المنزوعة السلاح للشبكة (DMZ) كما هو موضح في مخطط النشر الذي تمت الإشارة إليه سابقا.

1. قم بتثبيت نظام التشغيل الذي تختاره باستخدام مواصفات الأجهزة المطلوبة. قد تختلف تعديلات معلمات Kernel و IPv4 بناء على نظام التشغيل المحدد ويتم نصح المستخدمين بإعادة التحقق من هذه الجوانب إذا كان إصدار نظام التشغيل المختار مختلفا.
2. قم بتكوين واجهتي شبكة. سيلزم وجود واجهة واحدة للوصول العام من عملاء الإنترنت وآخر للاتصال بالخوادم في الشبكة الداخلية.
3. قم بتثبيت OpenResty.

يمكن إستخدام أي منكهات من NGINX لهذا الغرض، طالما أنها تستند إلى NGINX 1. 19+ وتدعم LUA:

- إنجينكس بلس
- مصدر NGINX المفتوح (يلزم تحويل المصدر المفتوح Nginx برمجيا مع وحدات LUA المستندة إلى OpenResty لاستخدامها)
- أوبن ريستي
- مصادر GetPageSpeed الإضافية

ملاحظة: تم إختبار التكوين المتوفر باستخدام OpenRest 1.19 ومن المتوقع أن يعمل مع التوزيعات الأخرى التي تحتوي على تحديثات طفيفة فقط، إن وجدت.

تثبيت OpenResty

1. تثبيت OpenResty. راجع حزم OpenRest Linux. كجزء من تثبيت OpenRest، سيتم تثبيت Nginx في هذا الموقع وإضافة مسار OpenRest إلى متغير PATH عن طريق الإضافة في الملف ~/.bashrc.

   ```
   export PATH=/usr/local/openresty/bin:$PATH
   ```

2. بدء/إيقاف NGINX.

- لبدء Nginx، أدخل **openresty**.
- لإيقاف Nginx، أدخل **openresty -s stop**.

تكوين NGINX

يتم شرح التكوين لتثبيت Nginx المستند إلى OpenRest. الدلالات الافتراضية ل OpenResty هي:

- <nginx-install-directory> = /usr/local/openresty/nginx
- <OpenResty-install-directory> = /usr/local/openresty

1. قم بتنزيل الملف واستخراجه من صفحة تنزيل برنامج Finesse الإصدار 12.6(1)ES03 (12.6-ES03-reverse-proxy-config.zip) التي تحتوي على تكوين الوكيل العكسي ل Nginx.
2. انسخ nginx.conf و nginx/conf.d/ و nginx/html/ من دليل تكوين الوكيل العكسي المستخرج إلى <nginx-install-directory>/conf و <nginx-install-directory>/conf/conf.d/ و<nginx-install-directory>/html/على التوالي.
3. انسخ دليل nginx/lua من دليل تكوين الوكيل العكسي المستخرج داخل <nginx-install-directory>.
4. انسخ محتويات Lualib إلى <OpenResty-install-directory>/lualib/resty.
5. قم بتكوين تدوير سجل NGINX عن طريق نسخ ملف nginx/logrotate/saproxy إلى <nginx-install-directory>/logrotate/folder. قم بتعديل محتويات الملف للإشارة إلى دلائل السجل الصحيحة إذا لم يتم إستخدام افتراضيات Nginx.
6. يجب تشغيل NGINX بحساب خدمة غير مفضل مخصص، والذي يجب تأمينه وأن يحتوي على طبقة غير صحيحة (أو كما وهو منطبق على نظام التشغيل المختار).
7. ابحث عن السلسلة يجب التغيير" في الملفات الموجودة ضمن المجلدات المستخرجة والمسماة html وconf.d واستبدل القيم المشار إليها بإدخالات مناسبة.
8. ضمنت أن كل الاستبدال إجباري تم، والتي يتم وصفها مع التعليقات يجب التغيير في ملفات التكوين.
9. تأكد من أن أدلة التخزين المؤقت التي تم تكوينها ل CUIC و Finesse تم إنشاؤها تحت <nginx-install-directory>/cache مع هذه الأدلة المؤقتة.
   - <nginx-install-directory>/cache/client_temp
   - <nginx-install-directory>/cache/proxy_temp

ملاحظة: التشكيل المقدم هو لعينة من عمليات النشر لعام 2000 ويجب توسيعه على نحو مناسب من أجل عملية نشر أكبر.

تكوين ذاكرة التخزين المؤقت ل Nginx

بشكل افتراضي، يتم تخزين مسارات ذاكرة التخزين المؤقت للوكيل في نظام الملفات. ونوصي بتغييرها إلى محركات أقراص داخل الذاكرة عن طريق إنشاء موقع ذاكرة تخزين مؤقت في TMPFS كما هو موضح هنا.

1. قم بإنشاء أدلة لمسارات التخزين المؤقت للوكيل المختلفة ضمن /home.

   على سبيل المثال، يجب إنشاء هذه الأدلة ل Finesse الأساسية. يجب اتباع الخطوات نفسها لمقامات Finesse الثانوية و CUIC.

```
mkdir -p /home/primaryFinesse/rest
mkdir -p /home/primaryFinesse/desktop
mkdir -p /home/primaryFinesse/shindig
mkdir -p /home/primaryFinesse/openfire
mkdir -p /home/primaryCUIC/cuic
mkdir -p /home/primaryCUIC/cuicdoc
mkdir -p /home/client_temp
mkdir -p /home/proxy_temp
```

```
echo "tmpfs /home/primaryFinesse/rest tmpfs size=1510M,rw,auto,noexec,nodev,nosuid,gid=root,uid=root,mode=1700 0 0" >>
/etc/fstab echo "tmpfs /home/primaryFinesse/desktop tmpfs size=20M,rw,auto,noexec,nodev,nosuid,gid=root,uid=root,mode=1700 0 0"
>>
/etc/fstab echo "tmpfs /home/primaryFinesse/shindig tmpfs size=500M,rw,auto,noexec,nodev,nosuid,gid=root,uid=root,mode=1700 0 0"
>>
/etc/fstab echo "tmpfs /home/primaryFinesse/openfire tmpfs size=10M,rw,auto,noexec,nodev,nosuid,gid=root,uid=root,mode=1700 0 0"
>>
/etc/fstab echo "tmpfs /home/primaryCUIC/cuic tmpfs size=100M,rw,auto,noexec,nodev,nosuid,gid=root,uid=root,mode=1700 0 0" >>
/etc/fstab echo "tmpfs /home/primaryCUIC/cuicdoc tmpfs size=100M,rw,auto,noexec,nodev,nosuid,gid=root,uid=root,mode=1700 0 0"
>>
/etc/fstab echo "tmpfs /home/client_temp tmpfs size=2048M,rw,auto,noexec,nodev,nosuid,gid=root,uid=root,mode=1700 0 0" >>
/etc/fstab echo "tmpfs /home/proxy_temp tmpfs size=2048M,rw,auto,noexec,nodev,nosuid,gid=root,uid=root,mode=1700 0 0" >>
/etc/fstab
```

ملاحظة: قم بزيادة ذاكرة التخزين المؤقت للعميل و proxy_temp بمقدار 1 غيغابايت لكل مجموعة Finesse جديدة مضافة إلى التكوين.

2. قم بتركيب نقاط التحميل الجديدة باستخدام الأمر **mount -av**.
3. قم بالتحقق من أن نظام الملفات قد قام بتثبيت نقاط التحميل الجديدة باستخدام الأمر **df -h**.
4. قم بتغيير موقع proxy_cache_path في ملفات تكوين Finesse و CUIC cache.

على سبيل المثال، لتغيير المسارات دليل Finesse الأساسي، انتقل إلى <nginx-install-directory>conf/conf.d/finesse/cache وقم بتغيير موقع ذاكرة التخزين المؤقت الحالي /usr/local/openresty/nginx/cache/finesse25/ إلى موقع Filesystem (نظام الملفات) الذي تم إنشاؤه حديثا /home/primaryFinesse.

```
##Must-change /usr/local/openresty/nginx/cache/finesse25 location would change depending on folder extraction proxy_cache_path
/home/primaryFinesse/desktop levels=1:2 use_temp_path=on keys_zone=desktop_cache_fin25:10m max_size=15m inactive=3y
use_temp_path=off; proxy_cache_path /home/primaryFinesse/shindig levels=1:2 use_temp_path=on
keys_zone=shindig_cache_fin25:10m max_size=500m inactive=3y use_temp_path=off; proxy_cache_path
/home/primaryFinesse/openfire levels=1:2 use_temp_path=on keys_zone=openfire_cache_fin25:10m max_size=10m inactive=3y
use_temp_path=off; proxy_cache_path /home/primaryFinesse/rest levels=1:2 use_temp_path=on keys_zone=rest_cache_fin25:10m
max_size=1500m inactive=40m use_temp_path=off;
```

5. اتبع الخطوات نفسها للمقامات الثانوية والخاصة بمكونات Finesse.

ملاحظة: تأكد من إضافة مجموع جميع وحدات توسعة محركات الأقراص TMPFS التي تم إنشاؤها في جميع الخطوات السابقة إلى تحديد الحجم النهائي للذاكرة لأغراض النشر، حيث إن محركات الأقراص هذه هي كتل ذاكرة تم تكوينها لتبدو كأقراص للتطبيق وتستهلك مساحة كبيرة من الذاكرة.

تكوين شهادات SSL

إستخدام شهادات موقعة ذاتيا - عمليات نشر الاختبار

يجب إستخدام الشهادات الموقعة ذاتيا فقط حتى يكون الوكيل العكسي جاهزا ليتم دمجها في الإنتاج. في نشر الإنتاج، أستخدم الشهادة الموقعة من قبل المرجع المصدق (CA) فقط.

1. إنشاء شهادات Nginx لمحتوى مجلد SSL. قبل إنشاء الشهادات، تحتاج إلى إنشاء مجلد يسمى ssl تحت /usr/local/openresty/nginx. تحتاج إلى إنشاء شهادتين بمساعدة هذه الأوامر (واحدة ل <reverseproxy_primary_fqdn>وأخرى ل <reverseproxy_secondary_fqdn>).
   a. **sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /usr/local/openresty/nginx/ssl/nginx.key -out /usr/local/openresty/nginx/ssl/nginx.crt** (قم بتمرير اسم المضيف ك : <reverseproxy_primary_fqdn>)
   b. **sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /usr/local/openresty/nginx/ssl/nginxnode2.key -out /usr/local/openresty/nginx/ssl/nginxnode2.crt** (قم بتمرير اسم المضيف ك :<reverseproxy_secondary_fqdn>)
   c. تأكد من أن مسار الشهادة هو /usr/local/openresty/nginx/ssl/nginx.crt و/usr/local/openresty/nginx/ssl/nginxnode2.crt، انظر لتكوينهما بالفعل في ملفات تكوين Finesse Nginx.
2. قم بتغيير إذن المفاتيح الخاص 400 (r—).
3. قم بتكوين جدار الحماية/IPTABLES على الوكيل العكسي لتمكين الاتصال من جدار الحماية ليتوافق مع المنافذ التي تم تكوين خادم Nginx عليها للاستماع.
4. قم بإضافة عنوان IP واسم المضيف الخاص ب Finesse، والمعرفات، و CUIC أسفل إدخال /etc/host على خادم الوكيل العكسي.
5. ارجع إلى دليل ميزات الحل للتكوينات التي سيتم إجراؤها على خوادم المكونات لتكوين مضيف Nginx وكيل عكسي.

ملاحظة: التشكيل المقدم هو عينة من عمليات النشر لعام 2000 ويجب توسيعه على نحو مناسب من أجل عملية نشر أكبر.

إستخدام الشهادة الموقعة من CA - عمليات نشر الإنتاج

يمكن تثبيت شهادة موقعة من CA على الوكيل العكسي باستخدام الخطوات التالية:

1. إنشاء طلب توقيع الشهادة (CSR).

   لإنشاء CSR والمفاتيح الخاص،**openssl req -new -newkey rsa:4096 -keyout nginx.key -out nginx.csr**أدخل بعد تسجيل الدخول إلى الوكيل. اتبع التعليمات الفورية، وقدم التفاصيل. وهذا يؤدي إلى

إنشاء CSR (nginx.csr في المثال) ومفتاح RSA الخاص (nginx.key في المثال) لوحدات بت القوة 4096.

على سبيل المثال:

[root@reverseproxyhost.companyname.com ssl]# openssl req -new -newkey rsa:4096 -keyout nginx.key -out nginx.csr Generating a RSA private key .....+++++ .....+++++ writing new private key to 'nginx.key' Enter PEM pass phrase:passphrase Verifying - Enter PEM pass phrase:passphrase ----- You are about to be asked to enter information that will be incorporated into your certificate request. What you are about to enter is what is called a Distinguished Name or a DN. There are quite a few fields but you can leave some blank For some fields there will be a default value, If you enter '.', the field will be left blank. ----- Country Name (2 letter code) [XX]:US State or Province Name (full name) []:CA Locality Name (eg, city) [Default City]:Orange County Organization Name (eg, company) [Default Company Ltd]:CompanyName Organizational Unit Name (eg, section) []:BusinessUnit Common Name (eg, your name or your server's hostname) []:reverseproxyhostname.companydomain.com Email Address []:john.doe@comapnydomain.com Please enter the following 'extra' attributes to be sent with your certificate request A challenge password []:challengePWD An optional company name []:CompanyName

اكتب عبارة مرور PEM، حيث سيتم إستخدامها لفك تشفير المفتاح الخاص أثناء النشر.

2. الحصول على الشهادة الموقعة من المرجع المصدق.

إرسال CSR إلى المرجع المصدق والحصول على الشهادة الموقعة.

ملاحظة: إذا كانت الشهادة المتلقاة من المرجع المصدق ليست سلسلة شهادات تحتوي على جميع الشهادات ذات الصلة، فعليك تأليف جميع الشهادات ذات الصلة في ملف سلسلة شهادات واحد.

3. نشر الشهادة والمفاتح.

قم بفك تشفير المفاتيح الذي تم إنشاؤه مسبقا كجزء من الخطوة الأولى باستخدام **openssl rsa -in nginx.key -out nginx_decrypted.key** الأمر. ضع شهادة CA الموقعة والمفاتيح الذي تم فك تشفيره داخل المجلد /usr/local/openresty/nginx/ssl في جهاز الوكيل العكسي. تحديث/إضافة تكوينات SSL المتعلقة بالشهادة في تكوينات Nginx في ملف التكوين /usr/local/openresty/nginx/conf/conf.d/ssl/ssl.conf.

ssl_certificate /usr/local/openresty/nginx/ssl/ca_signed_cert.crt; ssl_certificate_key /usr/local/openresty/nginx/ssl/nginx_decrypted.key;

4. قم بتكوين الأذونات للشهادات.

chmod 400 /usr/local/openresty/nginx/ssl/ca_signed_cert.crt Enter and **chmod 400 /usr/local/openresty/nginx/ssl/nginx_decrypted.key**، بحيث يكون للشهادة إذن للقراءة فقط وتقتصر على المالك.

5. إعادة تحميل NGINX.

إستخدام معلمة Diffie-Hellman المخصصة

إنشاء معلمة Diffie-Hellman مخصصة باستخدام الأوامر التالية:

```
openssl dhparam -out /usr/local/openresty/nginx/ssl/dhparam.pem 2048 chmod 400 /usr/local/openresty/nginx/ssl/dhparam.pem
```

قم بتغيير تكوين الخادم لاستخدام المعلمات الجديدة في الملف /usr/local/openresty/nginx/conf/conf.d/ssl/ssl.conf:

```
ssl_dhparam /usr/local/openresty/nginx/ssl/dhparam.pem;
```

تأكد من تمكين تدبيس OCSP - التحقق من إبطال الشهادة

ملاحظة: لتمكين ذلك، يجب أن يستخدم الخادم شهادة موقعة من CA ويجب أن يكون لدى الخادم حق الوصول إلى CA الذي وقع على الشهادة.

إضافة/تحديث هذه التكوين في file/usr/local/openresty/nginx/conf/conf.d/ssl/ssl.conf:

```
ssl_stapling on; ssl_stapling_verify on;
```

تكوين NGINX

يجب تعديل ملف تكوين Nginx الافتراضي (/usr/local/openresty/nginx/conf/nginx.conf) الموجود في الدليل الأمامي وتوفير الدليل. يجب إستخدام هذه المحتوى لتعديل ملف التكوين هذه الإدخالات لفرض الإعدادات المثبتة بواسطة برنامج Nginx.

```
# Increasing number of worker processes will not increase the processing the request. The number of work
# in system CPU. Nginx provides "auto" option to automate this, which will spawn one worker for each CPU
worker_processes  auto;

# Process id file location
pid /usr/local/openresty/nginx/logs/nginx.pid;

# Binds each worker process to a separate CPU
worker_cpu_affinity auto;

#Defines the scheduling priority for worker processes. This should be calculated by "nice" command. In
worker_priority 0;


error_log  /usr/local/openresty/nginx/logs/error.log info;

#user root root;

# current limit on the maximum number of open files by worker processes, keeping 10 times of worker_con

worker_rlimit_nofile 102400;

events {
```

```
    multi_accept on;


    # Sets the maximum number of simultaneous connections that can be opened by a worker process.
    # This should not be more the current limit on the maximum number of open files i.e. hard limit of
    # The appropriate setting depends on the size of the server and the nature of the traffic, and can
    worker_connections  10240;
    #debug_connection 10.78.95.21
}


http {

    include       mime.types;


    default_type  text/plain;


    ## Must-change Change with DNS resolver ip in deployment
    resolver 192.168.1.3;


    ## Must-change change lua package path to load lua libraries
    lua_package_path  "/usr/local/openresty/lualib/resty/?.lua;/usr/local/openresty/nginx/lua/?.lua;;"


    ## Must-change change proxy_temp folder as per cache directory configurations
    proxy_temp_path /usr/local/openresty/nginx/cache/proxy_temp 1 2 ;
    ## Must-change change client_temp folder as per cache directory configurations
    client_body_temp_path /usr/local/openresty/nginx/cache/client_temp 1 2 ;


    lua_shared_dict userlist 50m;
    lua_shared_dict credentialsstore 100m;
    lua_shared_dict userscount 100k;
    lua_shared_dict clientstorage 100m;
    lua_shared_dict blockingresources 100m;
    lua_shared_dict tokencache_saproxy 10M;
    lua_shared_dict tokencache_saproxy125 10M;
    lua_shared_dict ipstore 10m;
    lua_shared_dict desktopurllist 10m;
    lua_shared_dict desktopurlcount 100k;
    lua_shared_dict thirdpartygadgeturllist 10m;
    lua_shared_dict thirdpartygadgeturlcount 100k;
    lua_shared_dict corsheadersstore 100k;


    init_worker_by_lua_block {
        local UsersListManager = require('users_list_manager')
        local UnauthenticatedDesktopResourcesManager = require("unauthenticated_desktopresources_manage
        local UnauthenticatedResourcesManager = require("unauthenticated_thirdpartyresources_manager")
        --  Must-change Replace saproxy.cisco.com with reverseproxy fqdn

        if  ngx.worker.id() == 0 then
            UsersListManager.getUserList("saproxy.cisco.com", "https://saproxy.cisco.com:8445/finesse/a
            UnauthenticatedDesktopResourcesManager.getDesktopResources("saproxy.cisco.com", "https://sa
            UnauthenticatedResourcesManager.getThirdPartyGadgetResources("saproxy.cisco.com", "https://
        end
    }

    include conf.d/*.conf;
```

```
    sendfile        on;

    tcp_nopush     on;

    server_names_hash_bucket_size 512;
```

تكوين منفذ الوكيل العكسي

بشكل افتراضي، يستمع تكوين Nginx على المنفذ 8445 لطلبات Finesse. في وقت ما، يمكن تمكين منفذ واحد فقط من وكيل عكسي لدعم طلبات Finesse، على سبيل المثال 8445. إذا كان المنفذ 443 بحاجة إلى دعم، فعليك بتحرير ملف <nginx-install-directory>conf/conf.d/finesse.conf لتمكين الاستماع على 443 وتعطيل الاستماع على 8445.

تكوين مصادقة TLS المتبادلة بين الوكيل العكسي ومكونات تدفق البيانات

يمكن تمكين مصادقة شهادة SSL للعميل للاتصالات من مضيفي الوكيل العكسي على مكونات CCBU الأمامية CUIC/Finesse/IDs/LiveAta عبر خيار CLI ل CVOS الجديد والذي يستخدم النظام الوكيل العكسي للعميل-المصادقة enable/disable/status.

بشكل افتراضي، يتم تعطيل هذا الأمر ويجب تمكينه بشكل صريح بواسطة المسؤول لأدبت ،الخيار هذا تمكين بمجرد .لكل مستقل بشكل للتحميل مدخل كل على CLI تنفيذ خلال من Cisco خدمات مصادقة بدء مدخل المدخل الخارجي فقط من نظام يعمل على التي الخدمات من ويب وكيل عكسي موثوق بهم ومضيفين الذين أنشئ من خلال الاتصالات التي TLS مصافحة في العميل تتم إضافة مضيف الوكيل العكسي جزء من إضافة النظام، CLI ل <proxy-host>ل CLI.

أدناه هو كتلة التكوين النسخ الشيء في ملفات تكوين الوكيل على وجه التحديد ssl.conf و ssl2.conf

#Must-change /usr/local/openresty/nginx/ssl/nginx.crt change this location accordingly proxy_ssl_certificate /usr/local/openresty/nginx/ssl/nginx.crt; #Must-change /usr/local/openresty/nginx/ssl/nginx.key change this location accordingly proxy_ssl_certificate_key /usr/local/openresty/nginx/ssl/nginx.key;

يمكن أن تكون شهادة SSL المستخدمة لحركة المرور الصادرة (الوكيل إلى المدخل) مماثلة لشهادة SSL التي تم تكوينها لحركة المرور الواردة (موصل SSL لتلك مداخل المكون). إذا تم استخدام شهادة SSL الموقعة ذاتيا على أنها proxy_ssl_certificate بدلا من تحميلها إلى مكونات تدفق البيانات (Finesse/Id/CUIC/LiveAta)، يجب أن يتم مصادقتها بنجاح.

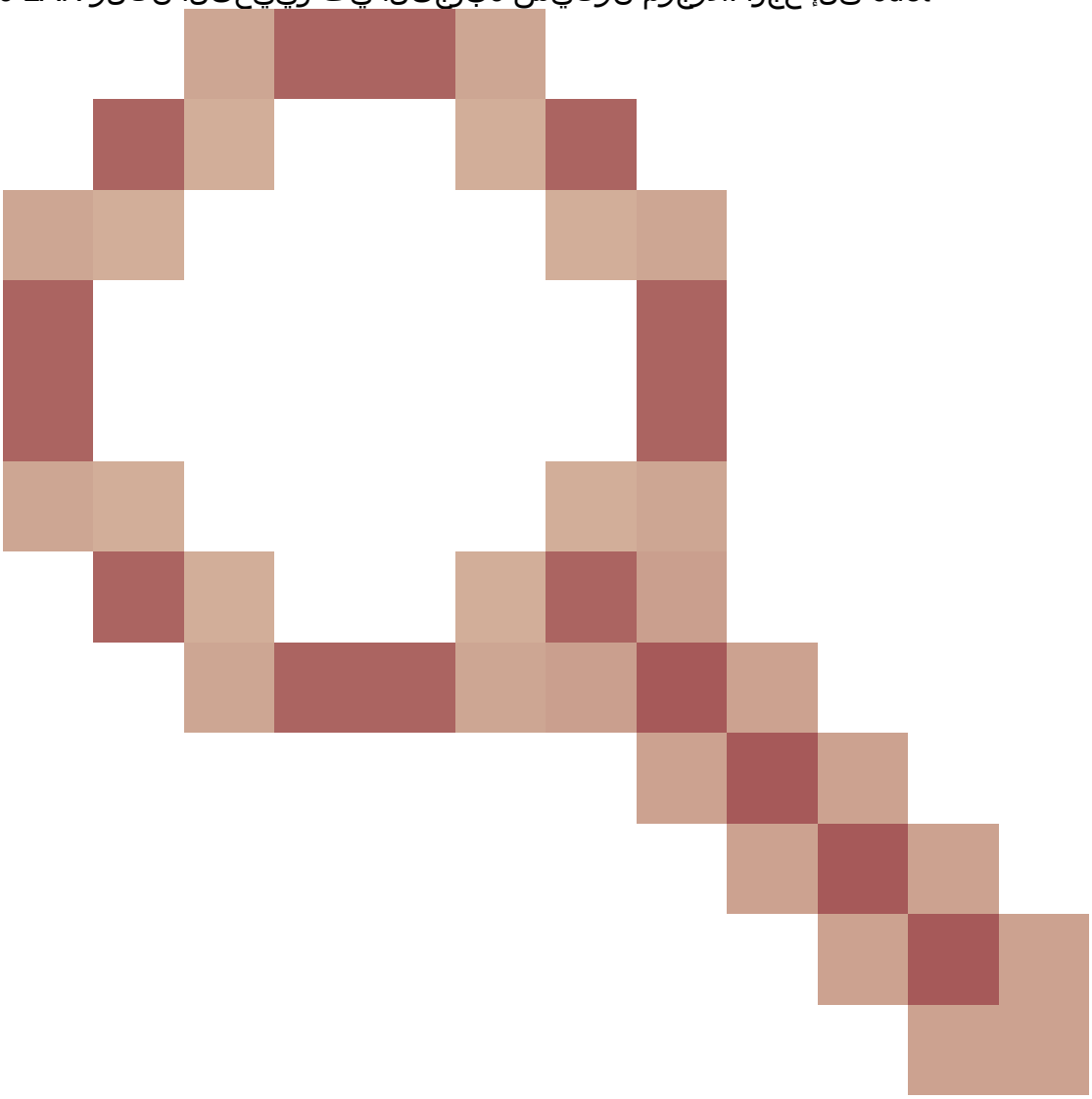
لا يتم تمكين التحقق من صحة شهادة المدخل عند التشغيل العكسي إختياريا ويتم تعطيله بشكل افتراضي. إذا كنت ترغب في تحقيق مصادقة TLS المتبادلة الكاملة بين الوكيل العكسي ومضيفي المدخل، يلزم عدم التعليق على التكوين أدناه من ملفات ssl.conf و ssl2.conf.

#Enforce upstream server certificate validation at proxy -> #this is not mandated as per CIS buit definitely adds to security. #It requires the administrator to upload all upstream server certificates to the proxy certificate store #Must-Change Uncomment below lines IF need to enforce upstream server certificate validation at proxy #proxy_ssl_verify on; #proxy_ssl_trusted_certificate /usr/local/openresty/nginx/ssl/finesse25.crt;

proxy_ssl_trusted_certificate: This file should contain the all upstream certificate enteries concatenated together

تخيارات لتكوين مصادقة TLS المتبادلة:

- بمجرد تمكين هذه الميزة على شهادة عميل مكونات CCBU سيتم طلبها من عملاء LAN أيضا أثناء مصافحة TLS. في حالة تثبيت أي شهادات عميل/شخصي على مستعرضات أجهزة العميل قد تختار عرض منبثق ليطلب المستخدم إختيار الشهادة المناسبة لمصادقة العميل. على الرغم من أنه ال يهم أي شهادة يختار المستخدم النهائي وأن يضغط إلغاء على الطلبات المنبثقة سينجح ذلك لأن مصادقة العميل غير مفروضة لعملاء شبكة LAN ولكن التغيير في التجربة سيكون موجودا. ارجع إلى cdet



CSCwa26057
للحصول على مزيد من التفاصيل.
- فشلت خدمة وكيل ويب لمكونات البث في الظهور في حالة إضافة مضيف وكيل إلى القائمة المسموح بها والتي ال يمكن حلها بواسطة خدمة وكيل ويب. تأكد من أن مضيفي الوكيل العكسي الذين تمت إضافتهم إلى القائمة المسموح بها قابلة للحل من مكون تدفق البيانات من خالل البحث عن DNS.

مسح ذاكرة التخزين المؤقت

يمكن مسح ذاكرة التخزين المؤقت للوكيل العكسي باستخدام

/clearCache.sh

الأمر.

إرشادات قياسية

يصف هذا القسم بإيجاز الإرشادات القياسية التي يجب اتباعها عند إعداد Nginx كخادم وكيل.

وهذه المبادئ التوجيهية مستمدة من [مركز أمن الإنترنت](). لمزيد من التفاصيل حول كل دليل توجيهي، ارجع إلى نفس الإجراء.

1. يوصى دائما باستخدام أحدث إصدار ثابت من OpenSSL و OpenRest.
2. ينصح بتثبيت Nginx في حاملات صرف منفصل.
3. يجب أن يكون معرف عملية Nginx مملوكا من قبل المستخدم الجذري (أو حسب ما ينطبق على نظام التشغيل المختار) ويجب أن يكون لديه الإذن 644 (rw—) أو أكثر صرامة.
4. يجب أن يقوم Nginx بحظر طلبات الأجهزة المضيفة غير المعروفة. تأكد من أن كل كتلة خادم تحتوي على توجيه server_name محدد بشكل صريح. للتحقق، ابحث في كافة كتل الخادم في دليل nginx.conf وnginx/conf.d وتحقق من أن كافة كتل الخادم تحتوي على server_name.
5. يجب أن ينصت Nginx فقط على المنافذ المعتمدة. ابحث في كافة كتل الخادم في دليل nginx.conf وnginx/conf.d وابحث عن الاستماع إلى التوجيهات للتحقق من أن المنافذ المعتمدة فقط مفتوحة للاستماع.
6. وانظر لأن Cisco Finesse لا تدعم HTTP، فهل يوصى بحظر منفذ HTTP للخادم الوكيل كذلك.
7. يجب أن يكون بروتوكول Nginx SSL وTLS 1.2. يجب إزالة دعم بروتوكولات SSL القديمة. كما يجب تعطيل شفرات SSL الضعيفة.
8. ينصح بإرسال سجلات الوصول والأخطاء Nginx إلى خادم syslog البعيد.
9. ينصح بتثبيت وحدة mod_security النمطية التي تعمل كجدار حماية لتطبيق ويب. راجع دليل ModSecurity للحصول على مزيد من المعلومات. لاحظ أنه لم يتم التحقق من صحة حمل Nginx داخل الوحدة النمطية mod_security الموجودة.

تكوين ملف التعيين

يتطلب نشر الوكيل العكسي الخاص بسطح مكتب Finesse ملف تخطيط لتكوين قائمة مجموعات أسماء/منافذ المضيف المرئية خارجيا وتخطيطها لأسماء الخوادم والمنافذ الفعلية التي يتم إستخدامها من قبل خوادم Finesse والمعرفات و CUIC. هذا ملف التعيين الذي تم تكوينه على الخوادم الداخلية وهو تكوين المفتاح الذي يسمح بإعادة توجيه العملاء المتصلين عبر الإنترنت إلى الأجهزة المضيفة والمنافذ المطلوبة التي يتم إستخدامها على الإنترنت.

يجب نشر ملف التعيين على خادم ويب يمكن الوصول إليه من قبل الخوادم المكونة، ويجب تكوين URI الخاص به لكي يعمل النشر. يوصى بتكوين ملف التعيين باستخدام خادم ويب مخصص متوفر داخل الشبكة. وفي حالة عدم توفر خادم من هذا القبيل، يمكن إستخدام الوكيل العكسي بدلا من ذلك، وهو ما سيتطلب إمكانية الوصول إلى الوكيل من داخل الشبكة كما يمثل مخاطرة تعريض المعلومات للعملاء الخارجيين الذين يمكنهم الوصول غير المصرح به إلى المنطقة المجردة من السلاح. ويورد الجزء التالي بالتفصيل كيف يمكن تحقيق ذلك.

ارجع إلى دليل المميزات للحصول على الخطوات الصحيحة لتكوين URI مل ف التعيين عل ى جميع الخوادم المكونة ولمزيد من التفاصيل حول كيفية إنشاء بيانات مل ف التعيين.

إستخدام الوكيل العكسي كخادم مل ف التعيين

تكون هذه الخطوات مطلوبة ف قط إذا كان الوكيل العكسي مستخدم أيضا كمضيف مل ف تعيين الوكيل.

1. قم بتكوين اسم مضيف الوكيل العكسي في وحدة تحكم المجال المستخدمة من قبل مضيفي Finesse/CUIC والمعرفات التي يمكن حل عنوان IP الخاص بها.
2. قم بتحميل الشهادات الموقعة ل Nginx التي تم إنشاؤها على الك العقد تحت ثقة Cmplatform أو تشغيل الخادم.
3. قم بتحديث القيم التي تغييرها في <NGINX_HOME>/html/proxymap.txt.
4. إعادة تحميل تكوينات NGINX باستخدام **nginx -s reload** الأمر.
5. تحقق من إمكانية الوصول إلى مل ف التكوين من مضيف شبكة آخر باستخدام الأمر**curl**.

تقوية نواة CentOS 8

إذا كان نظام التشغيل المختار هو CentOS 8، يوصى بإجراء التعزيز/الضبط kernel باستخدام تكوينات sysctl هذه للتثبيتات التي تستخدم خادم مخصص لاستضافة الوكيل.

```
## Configurations for kernel hardening - CentOS8. The file path is /etc/sysctl.conf
## Note that the commented configurations denote that CentOS 8's default value matches
## the recommended/tested value, and are not security related configurations.

# Avoid a smurf attack
net.ipv4.icmp_echo_ignore_broadcasts = 1

# Turn on protection for bad icmp error messages
net.ipv4.icmp_ignore_bogus_error_responses = 1

# Turn on syncookies for SYN flood attack protection
net.ipv4.tcp_syncookies = 1

# Turn on and log spoofed, source routed, and redirect packets
net.ipv4.conf.all.log_martians = 1
net.ipv4.conf.default.log_martians = 1

# Turn off routing
net.ipv4.ip_forward = 0
net.ipv4.conf.all.forwarding = 0
net.ipv6.conf.all.forwarding = 0

net.ipv4.conf.all.mc_forwarding = 0
net.ipv6.conf.all.mc_forwarding = 0

# Block routed packets
net.ipv4.conf.all.accept_source_route = 0
net.ipv4.conf.default.accept_source_route = 0
net.ipv6.conf.all.accept_source_route = 0
net.ipv6.conf.default.accept_source_route = 0

# Block ICMP redirects
net.ipv4.conf.all.accept_redirects = 0
```

```
net.ipv4.conf.default.accept_redirects = 0
net.ipv6.conf.all.accept_redirects = 0
net.ipv6.conf.default.accept_redirects = 0
net.ipv4.conf.all.secure_redirects = 0
net.ipv4.conf.default.secure_redirects = 0
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.default.send_redirects = 0

# Filter routing packets with inward-outward path mismatch(reverse path filtering)
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.default.rp_filter = 1

# Router solicitations & advertisements related.
net.ipv6.conf.default.router_solicitations = 0
net.ipv6.conf.default.accept_ra_rtr_pref = 0
net.ipv6.conf.default.accept_ra_pinfo = 0
net.ipv6.conf.default.accept_ra_defrtr = 0
net.ipv6.conf.default.autoconf = 0
net.ipv6.conf.default.dad_transmits = 0
net.ipv6.conf.default.max_addresses = 1
net.ipv6.conf.all.accept_ra = 0
net.ipv6.conf.default.accept_ra = 0

# Backlog - increased from default 1000 to 5000.
net.core.netdev_max_backlog = 5000

# Setting syn/syn-ack retries to zero, so that they don't stay in the queue.
net.ipv4.tcp_syn_retries = 0
net.ipv4.tcp_synack_retries = 0

# Max tcp listen backlog. Setting it to 511 to match nginx config
net.core.somaxconn = 511

# Reduce the duration of connections held in TIME_WAIT(seconds)
net.ipv4.tcp_fin_timeout = 6

# Maximum resources allotted
# fs.file-max = 2019273
# kernel.pid_max = 4194304
# net.ipv4.ip_local_port_range = 32768 60999

# TCP window size tuning
# net.ipv4.tcp_window_scaling = 1
# net.core.rmem_default = 212992
# net.core.rmem_max = 212992
# net.ipv4.tcp_rmem = 4096 87380 6291456
# net.ipv4.udp_rmem_min = 4096
# net.core.wmem_default = 212992
# net.core.wmem_max = 212992
# net.ipv4.tcp_wmem = 4096 16384 4194304
# net.ipv4.udp_wmem_min = 4096
# vm.lowmem_reserve_ratio = 256 256 32 0 0
# net.ipv4.tcp_mem = 236373 315167 472746

# Randomize virtual address space
kernel.randomize_va_space = 2

# Congestion control
# net.core.default_qdisc = fq_codel
# net.ipv4.tcp_congestion_control = cubic

# Disable SysReq
```

```
kernel.sysrq = 0

# Controls the maximum size of a message, in bytes
kernel.msgmnb = 65536

# Controls the default maximum size of a message queue
kernel.msgmax = 65536

# Controls the eagerness of the kernel to swap.
vm.swappiness = 1
```

يوصى بإعادة التشغيل بعد إجراء التغييرات الموصى بها.

تقوية IPtables

IPtables هو تطبيق يسمح لمسؤول النظام بتكوين جداول IPv4 و IPv6 والسلاسل والقواعد التي يوفرها جدار حماية نواة Linux.

يتم تكوين قواعد IPtables هذه لتأمين تطبيق الوكيل من الهجمات العنيفة بالقوة عن طريق تقييد الوصول في جدار حماية نواة ليونكس.

تشير التعليقات الموجودة في التكوين إلى الخدمة التي يتم حصرها باستخدام القواعد.

ملاحظة: إذا كان المسؤولون يستخدمون منفذا مختلفا أو يوسعون الوصول إلى خوادم متعددة باستخدام نفس المنافذ، فيجب القيام بالحجم المناسب لهذه المنافذ وفقا لهذه الأرقام.

```
## Configuration for iptables service
## The file path is /etc/sysconfig/iptables
## Make a note for must-change values to be replaced.
## Restart of the iptable service is required after applying following rules

*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]

# Ensure loopback traffic is configured
-A INPUT -i lo -j ACCEPT
-A OUTPUT -o lo -j ACCEPT
-A INPUT -s 127.0.0.0/8 -j DROP

# Ensure ping openeded only for the particular source and blocked for rest
# Must-Change: Replace the x.x.x.x with valid ip address
-A INPUT -p ICMP --icmp-type 8 -s x.x.x.x -j ACCEPT

# Ensure outbound and established connections are configured
-A INPUT -p tcp -m state --state RELATED,ESTABLISHED -j ACCEPT
-A OUTPUT -p tcp -m state --state NEW,RELATED,ESTABLISHED -j ACCEPT

# Block ssh for external interface
# Must-Change: Replace the ens224 with valid ethernet interface
-A INPUT -p tcp -i ens224 --dport 22 -j DROP
```

```
# Open inbound ssh(tcp port 22) connections
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT


# Configuration for finesse 8445 port
-A INPUT -p tcp -m tcp --dport 8445 --tcp-flags SYN SYN -m connlimit --connlimit-above 10 --connlimit-ma
-A INPUT -p tcp -m tcp --dport 8445 --tcp-flags SYN SYN -m connlimit --connlimit-above 10 --connlimit-ma
 -A INPUT -p tcp -m tcp --dport 8445 --tcp-flags SYN SYN -m hashlimit --hashlimit-upto 6/sec --hashlimi
-A INPUT -p tcp -m tcp --dport 8445 --tcp-flags SYN SYN -m limit --limit 1/min --limit-burst 1 -j LOG -
-A INPUT -p tcp -m tcp --dport 8445 --tcp-flags SYN SYN -j DROP

# Configuration for IdS 8553 port
-A INPUT -p tcp -m tcp --dport 8553 --tcp-flags SYN SYN -m connlimit --connlimit-above 6 --connlimit-ma
-A INPUT -p tcp -m tcp --dport 8553 --tcp-flags SYN SYN -m connlimit --connlimit-above 6 --connlimit-ma
-A INPUT -p tcp -m tcp --dport 8553 --tcp-flags SYN SYN -m hashlimit --hashlimit-upto 2/sec --hashlimit
-A INPUT -p tcp -m tcp --dport 8553 --tcp-flags SYN SYN -m limit --limit 1/min --limit-burst 1 -j LOG -
-A INPUT -p tcp -m tcp --dport 8553 --tcp-flags SYN SYN -j DROP

# Configuration for IdP 443 port
-A INPUT -p tcp -m tcp --dport 443 --tcp-flags SYN SYN -m connlimit --connlimit-above 8 --connlimit-mas
-A INPUT -p tcp -m tcp --dport 443 --tcp-flags SYN SYN -m connlimit --connlimit-above 8 --connlimit-mas
-A INPUT -p tcp -m tcp --dport 443 --tcp-flags SYN SYN -m hashlimit --hashlimit-upto 4/sec --hashlimit-
-A INPUT -p tcp -m tcp --dport 443 --tcp-flags SYN SYN -m limit --limit 1/min --limit-burst 1 -j LOG --
-A INPUT -p tcp -m tcp --dport 443 --tcp-flags SYN SYN -j DROP


# Must-Change: A2A file transfer has not been considered for below IMNP configuration.
# For A2A for support, these configuration must be recalculated to cater different file transfer scenar

# Configuration for IMNP 5280 port
-A INPUT -p tcp -m tcp --dport 5280 --tcp-flags SYN SYN -m connlimit --connlimit-above 30 --connlimit-m
-A INPUT -p tcp -m tcp --dport 5280 --tcp-flags SYN SYN -m connlimit --connlimit-above 30 --connlimit-m
-A INPUT -p tcp -m tcp --dport 5280 --tcp-flags SYN SYN -m hashlimit --hashlimit-upto 20/sec --hashlimi
-A INPUT -p tcp -m tcp --dport 5280 --tcp-flags SYN SYN -m limit --limit 1/min --limit-burst 1 -j LOG -
-A INPUT -p tcp -m tcp --dport 5280 --tcp-flags SYN SYN -j DROP

# Configuration for IMNP 15280 port
-A INPUT -p tcp -m tcp --dport 15280 --tcp-flags SYN SYN -m connlimit --connlimit-above 30 --connlimit-
-A INPUT -p tcp -m tcp --dport 15280 --tcp-flags SYN SYN -m connlimit --connlimit-above 30 --connlimit-
-A INPUT -p tcp -m tcp --dport 15280 --tcp-flags SYN SYN -m hashlimit --hashlimit-upto 20/sec --hashlim
-A INPUT -p tcp -m tcp --dport 15280 --tcp-flags SYN SYN -m limit --limit 1/min --limit-burst 1 -j LOG
-A INPUT -p tcp -m tcp --dport 15280 --tcp-flags SYN SYN -j DROP


# Configuration for IMNP 25280 port
-A INPUT -p tcp -m tcp --dport 25280 --tcp-flags SYN SYN -m connlimit --connlimit-above 30 --connlimit-
-A INPUT -p tcp -m tcp --dport 25280 --tcp-flags SYN SYN -m connlimit --connlimit-above 30 --connlimit-
-A INPUT -p tcp -m tcp --dport 25280 --tcp-flags SYN SYN -m hashlimit --hashlimit-upto 20/sec --hashlim
-A INPUT -p tcp -m tcp --dport 25280 --tcp-flags SYN SYN -m limit --limit 1/min --limit-burst 1 -j LOG
-A INPUT -p tcp -m tcp --dport 25280 --tcp-flags SYN SYN -j DROP


# Configuration for CUIC 8444 port
-A INPUT -p tcp -m tcp --dport 8444 --tcp-flags SYN SYN -m connlimit --connlimit-above 6 --connlimit-ma
-A INPUT -p tcp -m tcp --dport 8444 --tcp-flags SYN SYN -m connlimit --connlimit-above 6 --connlimit-ma
-A INPUT -p tcp -m tcp --dport 8444 --tcp-flags SYN SYN -m hashlimit --hashlimit-upto 2/sec --hashlimit
-A INPUT -p tcp -m tcp --dport 8444 --tcp-flags SYN SYN -m limit --limit 1/min --limit-burst 1 -j LOG -
-A INPUT -p tcp -m tcp --dport 8444 --tcp-flags SYN SYN -j DROP

# Configuration for CUIC 8447 port
-A INPUT -p tcp -m tcp --dport 8447 --tcp-flags SYN SYN -m connlimit --connlimit-above 6 --connlimit-ma
-A INPUT -p tcp -m tcp --dport 8447 --tcp-flags SYN SYN -m connlimit --connlimit-above 6 --connlimit-ma
```

```
-A INPUT -p tcp -m tcp --dport 8447 --tcp-flags SYN SYN -m hashlimit --hashlimit-upto 2/sec --hashlimit
-A INPUT -p tcp -m tcp --dport 8447 --tcp-flags SYN SYN -m limit --limit 1/min --limit-burst 1 -j LOG -
-A INPUT -p tcp -m tcp --dport 8447 --tcp-flags SYN SYN -j DROP

# Configuration for LiveData 12005  port
-A INPUT -p tcp -m tcp --dport 12005 --tcp-flags SYN SYN -m connlimit --connlimit-above 10 --connlimit-
-A INPUT -p tcp -m tcp --dport 12005 --tcp-flags SYN SYN -m connlimit --connlimit-above 10 --connlimit-
-A INPUT -p tcp -m tcp --dport 12005 --tcp-flags SYN SYN -m hashlimit --hashlimit-upto 6/sec --hashlimi
-A INPUT -p tcp -m tcp --dport 12005 --tcp-flags SYN SYN -m limit --limit 1/min --limit-burst 1 -j LOG
-A INPUT -p tcp -m tcp --dport 12005 --tcp-flags SYN SYN -j DROP

# Configuration for LiveData 12008  port
-A INPUT -p tcp -m tcp --dport 12008 --tcp-flags SYN SYN -m connlimit --connlimit-above 10 --connlimit-
-A INPUT -p tcp -m tcp --dport 12008 --tcp-flags SYN SYN -m connlimit --connlimit-above 10 --connlimit-
-A INPUT -p tcp -m tcp --dport 12008 --tcp-flags SYN SYN -m hashlimit --hashlimit-upto 6/sec --hashlimi
-A INPUT -p tcp -m tcp --dport 12008 --tcp-flags SYN SYN -m limit --limit 1/min --limit-burst 1 -j LOG
-A INPUT -p tcp -m tcp --dport 12008 --tcp-flags SYN SYN -j DROP

# Block all other ports
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited

COMMIT
```

يمكن تطبيق هذه القواعد مباشرة من خلال تحرير /etc/sysconfig/iptables يدويا أو بدلا من ذلك، حفظ التكوين في ملف iptables.conf وتنفيذ cat iptables.conf >>/etc/sysconfig/iptables لتطبيق القواعد.

يجب إعادة تشغيل خدمة IPtables بعد تطبيق القواعد. أدخل **systemctl restart iptables** لإعادة تشغيل خدمة IPtables.

تقييد إتصالات العميل

بالإضافة إلى تكوين IPtables السابق، يوصى باستخدام هذه المعرفة للتثبيتات التي تعرف نطاق العناوين للعملاء الذين يستخدمون الوكيل لتأمين قواعد الوصول للوكيل. يمكن أن يوفر هذا الأمر دفعات كبيرة من الأرباح عندما يتعلق الأمر بتأمين الوكيل من شبكات خبيثة غالبا ما يتم إنشاؤها في نطاق عناوين IP الخاصة بالبلدان التي تحتوي على قواعد أكثر تساهلا فيما يتعلق بالأمان على الإنترنت. لذلك، يوصى بشدة بتقييد نطاقات عناوين IP لنطاقات IP المستندة إلى IP (البلد/الولاية أو ISP) إذا كنت متأكدا من أنماط الوصول.

حظر إتصالات العملاء

ومن المفيد أيضا معرفة كيفية حظر نطاق معين من العناوين عند تحديد هوية ليتم تنفيذه من عنوان IP أو نطاق من عناوين IP. في مثل هذه الحالات، يمكن حظر الطلبات من عناوين IP هذه باستخدام قواعد جدول البيانات.

حظر عناوين IP المميزة

لحظر عناوين IP مميزة متعددة، أضف خطا إلى ملف تكوين IPTables لكل عنوان IP.

عل سبيل المثال، لحظر العناوين 192.0.2.3و192.0.2.4، أدخل:

<#root>

```
iptables -A INPUT -s
```

192.0.2.3

```
 -j DROP iptables -A INPUT -s
```

192.0.2.4

```
 - j DROP.
```

حظر نطاق عناوين IP

حظر عناوين IP المتعددة في نطاق وإضافة سطر واحد إلى ملف تكوين IPTables باستخدام نطاق IP.

عل سبيل المثال، لحظر العناوين من 192.0.2.3 إلى 192.0.2.35، أدخل:

```
iptables -A INPUT -m iprange --src-range 192.0.2.3-192.0.2.35 -j DROP.
```

حظر جميع عناوين IP في شبكة فرعية

قم بحظر جميع عناوين IP في شبكة فرعية بأكملها عن طريق إضافة سطر واحد إلى ملف تكوين IPTables باستخدام تدوين التوجيه بين المجالات دون فئات لنطاق عنوان IP. عل سبيل المثال، لحظر كل عناوين الفئة C، أدخل:

```
iptables -A INPUT -s 192.0.0.0/16 -j DROP.
```

سيلينكس

SELinux هو إطار عمل لأمان المنصات مدمج كتعزيز في نظام تشغيل لينوكس. يتم توفير OpenRest حيث يتم توفير سياسات SELinux وإضافتها لتشغيل الإجراء الخاص بتثبيت الوكيل العكسي التالي.

1. أوقف العملية باستخدام الأمر **openresty -s stop**.
2. قم بتكوين خادم OpenRest وتشغيله باستخدام الأمر **systemctl** حتى تبدأ عملية OpenResty تلقائيا أثناء التمهيد. أدخل هذه الأوامر كمستخدم جذر.
   a. انتقل إلى /usr/lib/system/system.
   b. افتح ملف يسمى openResty.service.

c. قم بتحديث محتوى الملف حسب موقع ملف PIDF.

```
[Unit]
Description=The OpenResty Application Platform
After=syslog.target network-online.target remote-fs.target nss-lookup.target
Wants=network-online.target

[Service]
Type=forking
PIDFile=/usr/local/openresty/nginx/logs/nginx.pid
ExecStartPre=/usr/local/openresty/nginx/sbin/nginx -t
ExecStart=/usr/local/openresty/nginx/sbin/nginx
ExecReload=/bin/kill -s HUP $MAINPID
ExecStop=/bin/kill -s QUIT $MAINPID
PrivateTmp=true

[Install]
WantedBy=multi-user.target
```

d. كمستخدم جذري، أدخل **sudo systemctl enable openresty**.

e. **systemctl start openresty / systemctl stop openresty** قم بتشغيل / إيقاف خدمة OpenRest باستخدام الأمر وتأكد من بدء / إيقاف العملية كمستخدم جذري.

1. تثبيت Selinux
   - بشكل افتراضي، سيتم تثبيت بعض حزم SELinux فقط في CentOS.
   - يلزم تثبيت حزمة النهج المعيارية-Devel وتبعياتها لإنشاء سياسة SelInux.
   - دخلت هذا أمر in order to ركبت policyResourceLs-devel

```
yum install policycoreutils-devel
```

   - تأكد من أنه بعد تثبيت الحزمة، فإن **sepolicy** الأمر يعمل.

```
usage: sepolicy [-h] [-P POLICY]

                {booleans,communicate,generate,gui,interface,manpage,network,transition}
                ...

SELinux Policy Inspection Tool
```

2. أنشئ مستخدم جديد لنظام التشغيل Linux وخريطة بينية باستخدام مستخدم لنظام التشغيل Linux

   a. أدخل **semanage login -l** لعرض التخطيط بين مستخدمي Linux ومستخدمي SELinux.

```
[root@loadproxy-cisco-com ~]# semanage login -l

Login Name           SELinux User         MLS/MCS Range        Service
```

```
__default__             unconfined_u            s0-s0:c0.c1023          *                   *
root                    unconfined_u            s0-s0:c0.c1023          *
```

b. كجذر، قم بإنشاء مستخدم Linux جديد (مستخدم nginx) الذي تم تعيينه إلى مستخدم SELinux user_u.

```
useradd -Z user_u nginxuser
[root@loadproxy-cisco-com ~]# passwd nginxuser
Changing password for user nginxuser.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

c. لعرض التخطيط بين nginxuser وuser_u، أدخل هذا الأمر كجذر:

```
[root@loadproxy-cisco-com ~]# semanage login -l

Login Name              SELinux User            MLS/MCS Range           Service

__default__             unconfined_u            s0-s0:c0.c1023          *
nginxuser               user_u                  s0                      *
root                    unconfined_u            s0-s0:c0.c1023          *
```

d. تسجيل الدخول إلى SelInux _default__login بشكل افتراضي تم تعيينه إلى مستخدم SelInux غير المحصور_u. وهو مطلوب أن يجعل user_u أن يكون حصري افتراضيا مع هذا أمر:

```
semanage login -m -s user_u -r s0 __default__
```

للتحقق من عمل الأمر بشكل صحيح، أدخل **semanage login -l**. وينبغي لها أن تنتج هذا الناتج:

```
Login Name              SELinux User            MLS/MCS Range           Service

__default__             user_u                  s0                      *
nginxuser               user_u                  s0                      *
root                    unconfined_u            s0-s0:c0.c1023          *
```

e. تعديل nginx.conf وإجراء تغيير المملكية ل nginxuser.

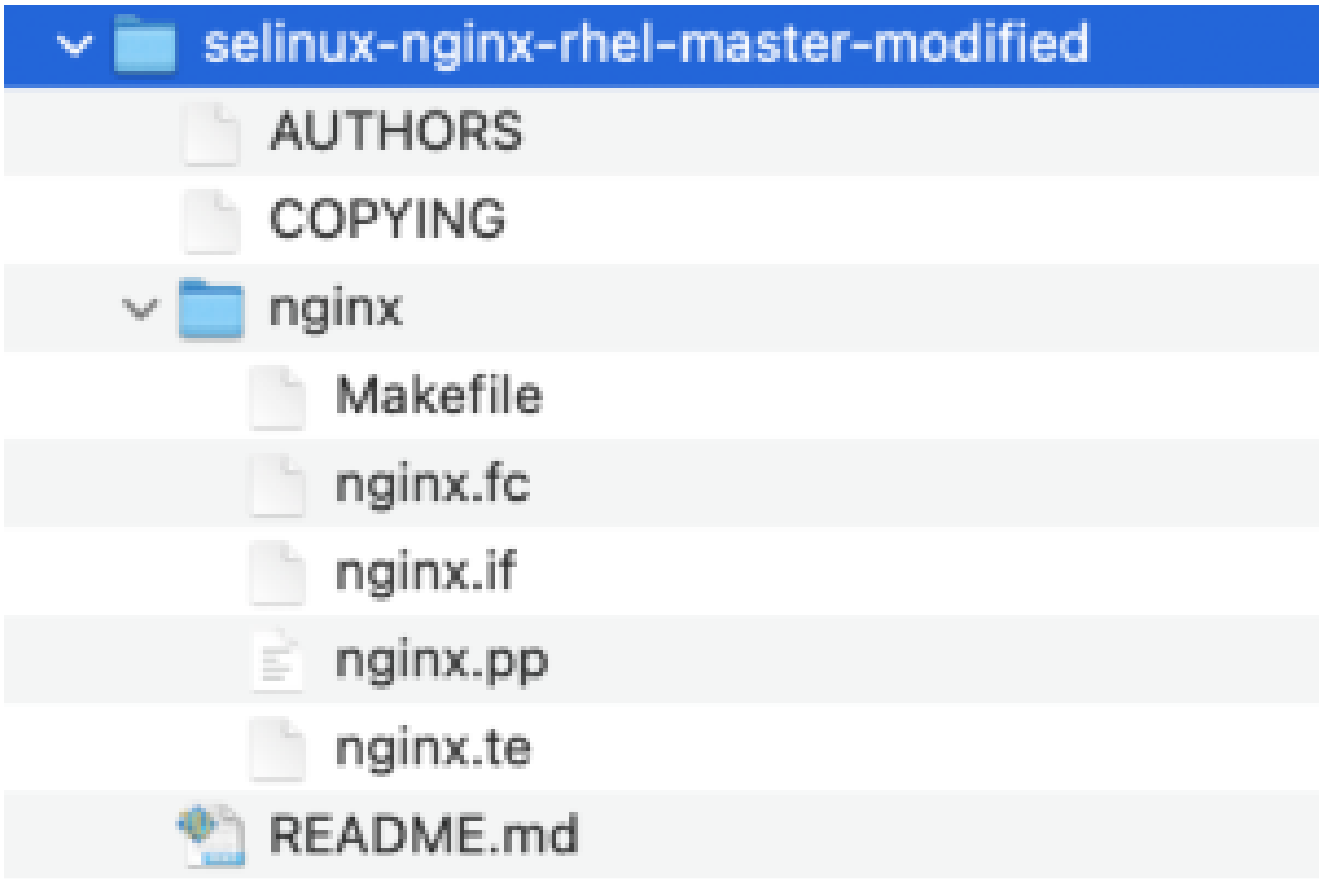
i. أدخل **chown -R nginxuser:nginxuser** * في دليل <OpenResty-install-directory.

ii. قم بتعديل ملف nginx.conf لتضمين nginxuser كمستخدم لتشغيل العمليات المنفذة.

```
.........
user nginxuser nginxuser;
..........
```

اكتب سياسة SeInux ل Nginx

1. بدلا من إنشاء نهج مخصص افتراضي جديد ل Nginx باستخدام الأمر، **sepolicy generate --init /usr/bin/nginx** من المفترض البدء بنهج موجود.
2. تم تعديل الملفات nginx.fc (ملف سياقات الملفات) وnginx.te (ملف فرض النوع) التي يتم تنزيلها من عنوان URL المتوفر لتالئم إستخدام الوكيل العكسي.
3. يمكن إستخدام هذا الإصدار المعدل كمرجع نظرا لأنه قد تم إصالحه لحالة إستخدام معينة.
4. قم بتنزيل الملف selinux-nginx-rhel-master-modified.tar من [صفحة تنزيل برامج الملفات](#).



5. استخرج ملف .tar وانتقل إلى دليل nginx بداخله.
6. افتح ملف .fc وتحقق من مسارات الملف المطلوبة من ملف nginx المثبت وذاكرة التخزين المؤقت وملف pid.
7. تحويل التكوين برمجيا باستخدام **make** الأمر.

8. سيتم إنشاء ملف nginx.pp.
9. قم بتحميل النوع باستخدام **semodule** الأمر.

```
semodule -i nginx.pp
```

10. انتقل إلى /root وقم بإنشاء ملف فارغ يسمى touch /.autorelabel.
11. أعد تمهيد النظام.
12. أدخل هذا الأمر للتحقق من تحميل النوع بنجاح.

```
semodule --list-modules=full
```

```
[root@loadproxy-cisco-com ~]# semodule --list-modules=full
400 nginx                  pp
200 container              pp
200 flatpak                pp
100 abrt                   pp
100 accountsd              pp
100 acct                   pp
100 afs                    pp
100 aiccu                  pp
100 aide                   pp
100 ajaxterm               pp
100 alsa                   pp
```

13. يجب تشغيل Nginx دون أي انتهاك. (ستتوفر الانتهاكات في /var/log/messages و/var/log/audit/audit.log).
14. دخلت هذا أمر in order to فحصت الحالة من Nginx.

```
ps -aefZ | grep nginx
```

```
[root@loadproxy-cisco-com ~]# ps -aefZ |grep nginx
system_u:system_r:nginx_t:s0    root        1686       1  0 16:14 ?        00:00:00 nginx: master process /usr/bin/nginx
system_u:system_r:nginx_t:s0    nginxus+    1687    1686  0 16:14 ?        00:00:00 nginx: worker process
system_u:system_r:nginx_t:s0    nginxus+    1688    1686  0 16:14 ?        00:00:00 nginx: worker process
system_u:system_r:nginx_t:s0    nginxus+    1689    1686  0 16:14 ?        00:00:00 nginx: worker process
system_u:system_r:nginx_t:s0    nginxus+    1690    1686  0 16:14 ?        00:00:00 nginx: worker process
system_u:system_r:nginx_t:s0    nginxus+    1691    1686  0 16:14 ?        00:00:00 nginx: worker process
system_u:system_r:nginx_t:s0    nginxus+    1692    1686  0 16:14 ?        00:00:00 nginx: worker process
system_u:system_r:nginx_t:s0    nginxus+    1693    1686  0 16:14 ?        00:00:00 nginx: worker process
system_u:system_r:nginx_t:s0    nginxus+    1694    1686  0 16:14 ?        00:00:00 nginx: worker process
system_u:system_r:nginx_t:s0    nginxus+    1695    1686  0 16:14 ?        00:00:00 nginx: cache manager process
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 root 2543 2252  0 16:17 pts/0 00:00:00 grep --color=auto nginx
```

15. يجب الآن الوصول إلى سطح المكتب الوكيل/المشرف الخاص بشركة Finesse.

التحقق من الصحة

استخدم هذا القسم لتأكيد عمل التكوين بشكل صحيح.

Finesse

1. اطلب https://<reverseproxy:port>/finesse/api/SystemInfo. من DMZ وتحقق مما إذا كان من الممكن الوصول إليه.
2. حدد قيم <host>في كل من <primaryNode> و<secondaryNode> أسماء مضيفات وكيل عكسي صالحة. يجب ألا تكون أسماء بيوت فارغة.

البيانات المجسمة والمعروضة مباشرة

1. إذا تم ملاحظة أسماء مضيفات Finesse في الاستجابة بدلا من أسماء مضيفات الوكيل العكسية، فتحقق من صحة تكوينات تعيين الوكيل والأجهزة المضيفة المسموح بها التي تمت إضافتها بشكل صحيح في خوادم Finesse كما هو موضح في القسم "تعبئة بيانات ترجمة الشبكة" من "الوصول الذي لا يحتوي على شبكات VPN إلى Finesse Desktop" في [دليل ميزة Finesse 12.6 UCCE.](#)
2. إذا تم تحميل الأدوات الذكية LiveData بشكل صحيح في Finesse Desktop، فإن تكوينات وكيل CUIC و LiveData ملائمة.
3. للتحقق من صحة تكوين CUIC و LiveData، قم بإجراء طلبات HTTP إلى عناوين URL هذه من DMZ وانظر ما إذا كان يمكن الوصول إليه.
   - https://<reverseproxy:cuic_port>/cuic/rest/about
   - https://<reverseproxy:ldweb_port>/livedata/security
   - https://<reverseproxy:ldsocketio_port>/security

IDS

للتحقق من صحة تكوين المعرفات، قم بتنفيذ الخطوات التالية:

1. سجل الدخول إلى واجهة IdSAdmin في https://<ids_lan_host:ids_port>:8553/idsadmin من الشبكة المحلية (LAN) حيث لا يتم عرض واجهة المسؤول عبر الوكيل العكسي.
2. أختر إعدادات > ثقة المعرفات.
3. تحقق من إدراج عقدة ناشر نظام المجموعة الوكيل في صفحة بيانات تعريف تنزيل SP، ثم انقر فوق التالي.
4. تحقق من عرض وكيل IDP بشكل صحيح إذا تم تكوينه على صفحة تحميل بيانات تعريف IDP وانقر فوق التالي.
5. قم ببدء إختبار SSO عبر جميع عقد نظام المجموعة للوكيل من صفحة إختبار SSO وتحقق من نجاح جميع الاختبارات. وهذا يتطلب اتصال جهاز العميل لعكس عقد الوكيل.

الأداء

يتوفر تحليل البيانات الخاص بالتقاط أفضل أداء مكافئ، والذي تم إجراؤه باستخدام أداة nmon، من [صفحة تنزيل برنامج Finesse، الإصدار 12.6(1) ES03](#) (load_result.zip). تمثل البيانات حالة الوكيل لعمليات سطح المكتب والمشرف، على نموذج نشر 2000 UCCE باستخدام تسجيلات الدخول إلى نظام SSO وتقارير CUIC LD كما تم تكوينها في التخطيط الافتراضي ل 2000 مستخدم لمدة ثماني ساعات. يمكن إستخدامه لاستخلاص متطلبات الحوسبة والقرص والشبكة للتثبيت باستخدام Nginx على الأجهزة المماثلة.

استكشاف الأخطاء وإصلاحها

وس

1. عمليات إعادة توجيه سطح المكتب التي لا تتم من خلال الوكيل
   1. تحقق من تكوين أسماء المضيف في الحالات الصحيحة وفقا لأسماء المضيفة الفعلية الخاصة بالأجهزة الافتراضية في تكوينات مختلفة مثل proxyMap.txt و server_filter file وما إلى ذلك.
   2. تأكد من إضافة المعرفات مع اسم المضيف الصحيح في مخزون CCE حيث يتم دفع نفس المعلومات إلى المكونات عند تسجيلها ل SSO من مسؤول ويب CCE.
2. إذن، لا تحدث عمليات الدخول
   1. تأكد من تأسيس ثقة IDs-IDP للمضيف الوكيل.

سكينيلس

1. إذا لم يتم بدء تشغيل Nginx بشكل افتراضي أو أن سطح مكتب وكيل Finesse غير قابل للوصول، فقم بتعيين SELinux إلى الوضع متساهل باستخدام هذا الأمر:

```
setenforce 0
```

2. أعد تشغيل Nginx باستخدام **systemctl restart nginx** الأمر.
3. وستتاح الانتهاكات في /var/log/messages و/var/log/audit/audit.log.
4. ويلزم إعادة إنشاء ملف te. مع السماح بالقواعد لمعالجة تلك الانتهاكات بواسطة أي من هذه الأوامر:

```
cat /var/log/audit/audit.log | audit2allow -m nginx1 > nginx1.te. # this will create nginx1.te fil
                or
ausearch -c 'nginx' --raw | audit2allow -M  my-nginx  # this will create my-nginx.te file
```

5. قم بتحديث الملف nginx.te الأصلي الموجود في دليل Selinux-nginx-rhel-master-modified/nginx باستخدام قواعد السماح التي تم إنشاؤها حديثا.
6. التحويل البرمجي للنفس باستخدام **make** الأمر.
7. سيتم إعادة إنشاء ملف nginx.pp.
8. قم بتحميل النهج بواسطة أمر الفواصل الزمنية.

```
semodule -i nginx.pp
```

9. أجعل SelInux تفرض الوضع باستخدام هذا الأمر:

```
setenforce
```

10. أعد تمهيد النظام.
11. كرر هذا الإجراء حتى يتم إصلاح المخالفات المطلوبة.

حول هذه الترجمة
ترجمت Cisco هذا المستند باستخدام مجموعة من التقنيات الآلية والبشرية لتقديم محتوى دعم للمستخدمين في جميع أنحاء العالم بلغتهم الخاصة. يُرجى ملاحظة أن أفضل ترجمة آلية لن تكون دقيقة كما هو الحال مع الترجمة الاحترافية التي يقدمها مترجم محترف. تخلي Cisco Systems مسئوليتها عن دقة هذه الترجمات وتُوصي بالرجوع دائمًا إلى المستند الإنجليزي الأصلي (الرابط متوفر).