8

الفصل الثامن

أمن إدارة الشبكات

محتويات الفصل:

8-1 المفاهيم الأساسية لبروتوكول إدارة الشبكات البسيط (SNMP)

8-1-1 البنية المعمارية لإدارة الشبكات

8-1-2 البنية المعمارية لبروتوكول إدارة الشبكات البسيط

8-1-3 المُفوَّضات (Proxies)

8-1-4 الإصدار الثاني لبروتوكول إدارة الشبكات البسيط

8-2 تسهيلات تجمعات SNMPv1

8-2-1 التجمعات وأسماؤها (Communities and Community Names)

8-2-2 خدمة التوثيق (Authentication Service)

8-2-3 سياسة الوصول (Access Policy)

8-2-4 خدمة المُفوَّض (Proxy Service)

8-3 الإصدار الثالث لبروتوكول إدارة الشبكات البسيط

8-3-1 بنية SNMP المعمارية

8-3-2 تمرير الرسائل ونموذج أمن المُستخدِم

8-3-3 التحكم في الوصول المبنيّ على المشهد

8-4 توصيات للمطالعة

8-5 مصادر للمعلومات على الويب

8-6 مصطلحات رئيسة

8-7 أسئلة للمراجعة ومسائل

"إن قيادة كتيبة كبيرة تماثل – من حيث المبدأ – قيادة عدد قليل من الرجال، فهي مجرد مسألة وضع أسس للعلامات والإشارات" – من كتاب "فن الحرب" ، لصن تزو.

لنظم المعالجة الموزَّعة وللشبكات أهمية كبيرة ومتزايدة لدى المنظمات والمؤسسات التجارية والهيئات الحكومية. وحالياً تتجه الأمور داخل المؤسسات نحو استخدام شبكات أكبر وأكثر تعقيداً لدعم مزيدٍ من التطبيقات وعددٍ أكبر من المُستخدِمين؛ ومع النمو المستمر لحجم تلك الشبكات، تتضح حقيقتان مهمتان:

- أصبحت الشبكة ومواردها الملحقة بها وما يرتبط بها من تطبيقات موزَّعة، لا غنى عنها للمؤسسات.
- مع زيادة تعقيد الشبكة زاد احتمال حدوث أعطال بها أو بجزء منها أو تدني أداء الشبكة إلى مستوى غير مقبول.

لذا أصبح من غير الممكن تأسيس شبكات ضخمة وإدارتها بالجهد البشري وحده، وتطلب الأمر استخدام أدوات آلية (مؤتمتة) لإدارة الشبكة. كما زادت الحاجة إلى مثل تلك الأدوات وزادت صعوبة توفيرها للشبكات التي تضم أجهزة وبرمجيات من شركات متعددة. ولتلبية هذه الحاجة، تم تطوير معايير (standards) – للتعامل مع إدارة الشبكة – تشمل الخدمات والبروتوكولات وقواعد معلومات الإدارة.

المعيار الأكثر استخداماً في هذا المجال هو بروتوكول إدارة الشبكات البسيط ((Simple Network Management Protocol (SNMP). ومنذ نشر مواصفاته في عام 1988، تم استخدام بروتوكول SNMP في أعدادٍ متزايدة من الشبكات وفي بيئات متزايدة التعقيد. ومع زيادة انتشاره ظهرت الحاجة إلى إضافة قدرات وظيفية جديدة للبروتوكول لاستيعاب المتطلبات الجديدة. كما اتضح أيضاً وبصورة متزايدة أهمية توفير قدرات أمنية كمكوِّن أساسي من مكوِّنات إدارة الشبكة. وتضمَّن

الإصدار الثاني من SNMP وظائف محسنَّة.[1] أما التحسينات الأمنية فقد جاءت في الإصدار الثالث (SNMPv3).

يبدأ هذا الفصل بوصف الإمكانات الأمنية الأوَّليَّة المتاحة في SNMPv1، ثم يناقش السمات الأمنية الأكثر تطوراً المتوفِّرة في الإصدار الثالث.

8-1 المفاهيم الأساسية لبروتوكول إدارة الشبكات البسيط (SNMP)

يقدم هذا الجزء لمحة عامة عن الإطار الأساسي لبروتوكول SNMP.

8-1-1 البنية المعمارية لإدارة الشبكات

يتألف نظام إدارة الشبكة من مجموعة من أدوات الرصد والمراقبة والتحكم في الشبكة، والمتوفرة بشكلٍ شبه مُجمَّع يتمثل في:

- وجود واجهة تشغيل واحدة مع مجموعة من الأوامر الفعَّالة سهلة الاستخدام لأداء معظم مهام إدارة الشبكة إن لم يكن جميعها.
- وجود أقل عدد من الأجهزة المنفصلة. بمعنى أن تكون معظم المكوِّنات المادية والبرمجيات اللازمة لإدارة الشبكة مُتضمَّنة في جهاز المُستخدِم.

يتألَّف نظام إدارة الشبكة من إضافات متزايدة من الأدوات والبرمجيات التي تعمل بين المكونات الموجودة بالشبكة. وتوجد البرامج المُستخدَمة في إنجاز مهام إدارة الشبكة بأنظمة الحاسبات المضيفة وبمعالجات الاتصالات (على سبيل المثال معالجات المقدمة (front-end processors)، ووحدات تحكم العناقيد الطرفية (terminal cluster controllers). تم تصميم نظام إدارة الشبكة لرؤية الشبكة بأكملها كبنية معمارية موحدة، بما في ذلك العناوين والتسميات المخصصة لكل نقطة والسمات المحدّدة لكل العناصر والوصلات المعروفة للنظام. وتوفر العناصر

[1] يشار إلى الإصدار الثاني بـ SNMPv2 لتمييز الإصدار الجديد عن الإصدار الأصلي الذي يشار إليه الآن عموماً بـ SNMPv1.

النشطة في الشبكة تغذية مرتدَّة منتظمة عن معلومات الحالة إلى مركز التحكم بالشبكة.

يتضمن نموذج إدارة الشبكة المُستخدَم في SNMP العناصر الرئيسة التالية:

- محطة الإدارة.
- وكيل الإدارة.
- قاعدة معلومات الإدارة.
- بروتوكول إدارة الشبكة.

عادةً ما تكون محطة الإدارة جهازاً مستقلاً بذاته أو قد تكون إمكانية ضمن نظام مشترك (shared system). وفي كلتا الحالتين، تُعد محطة الإدارة بمثابة الواجهة للشخص المسؤول عن إدارة الشبكة للتعامل مع نظام إدارة الشبكة. وكحد أدنى ينبغي توفر الآتي لمحطة الإدارة:

- مجموعة من تطبيقات الإدارة لتحليل البيانات، والتعافي من الأخطاء، إلخ.
- واجهة يستطيع من خلالها مدير الشبكة مراقبة الشبكة والتحكم فيها.
- إمكانية ترجمة طلبات مدير الشبكة إلى أوامر فعلية لمراقبة العناصر البعيدة في الشبكة والتحكم فيها.
- قاعدة بيانات للمعلومات المستخلصة من قواعد معلومات الإدارة MIBs لكل الكيانات المُتحكَّم فيها من قِبَل نظام إدارة الشبكة.

ويستهدف التوحيد القياسي لـ SNMP العنصرين الأخيرين فقط.

ويمثل "وكيل الإدارة" العنصر الآخر الفعَّال في نظام إدارة الشبكة. وقد تُجهَّز المنصَّات الأساسية – مثل الأنظمة المضيفة، والجسور (bridges)، والموجّهات (Routers)، والمُجمِّعات (Hubs) – ببروتوكول SNMP حتى يمكن إدارتها من خلال محطة الإدارة. ويقوم وكيل الإدارة بالاستجابة لطلبات المعلومات أو طلبات القيام بعمل معيّن الواردة من أيٍّ من محطات الإدارة، كما يمكنه بشكلٍ غير متزامن إمداد محطة الإدارة بمعلومات مهمة تلقائياً وبدون طلب منها.

ولإدارة الموارد في الشبكة، يتم تمثيل كل مورد ككائن. والكائن أساساً، هو متغير بيانات يمثل سمة واحدة للوكيل المُدار. ويُشار إلى مجموعة الكائنات بـ "قاعدة معلومات الإدارة" ((Management Information Base (MIB). وتعمل MIB كمجموعة من نقاط الوصول (النفاذ) لدى الوكيل تستخدمها محطة الإدارة. وقد تم توحيد هذه الكائنات بطريقة معيارية عبر الأنظمة المختلفة حسب فئتها (مثلاً: تقوم جميع الجسور بدعم نفس كائنات الإدارة). تؤدي محطة الإدارة وظيفة المراقبة عن طريق استرجاع قيم كائنات MIB. كما يمكن أن تتسبب محطة الإدارة في القيام بإجراءٍ محدّد لدى وكيل معين، وتستطيع تغيير إعدادات التهيئة لوكيلٍ ما بتعديل قيم كائنات معيّنة.

يتم ربط محطة الإدارة ووكلائها عن طريق بروتوكول إدارة الشبكة (network management protocol). فالبروتوكول المُستخدَم لإدارة شبكات TCP/IP هو بروتوكول إدارة الشبكة البسيط (SNMP). ويتضمن هذا البروتوكول الإمكانات الرئيسة الآتية:

- جلب (Get): تُمكّن محطة الإدارة من الحصول على قيم الكائنات لدى الوكيل.
- تعيين (Set): تُمكّن محطة الإدارة من تعديل قيم الكائنات لدى الوكيل.
- إخطار (Notify): تُمكّن الوكيل من إخطار محطة الإدارة بالأحداث المهمة.

8-1-2 البنية المعمارية لبروتوكول إدارة الشبكات

صدرت مواصفات SNMP في عام 1988، وسرعان ما أصبحت المعيار القياسي المهيمن على إدارة الشبكات. وتوفر بعض الشركات أنظمة محطات عمل قائمة بذاتها لإدارة الشبكات مبنيَّة على بروتوكول SNMP. كما يوفِّر معظم مصنعي الأجهزة - كالموجّهات (Routers)، والجسور (bridges)، ومحطات العمل، وأجهزة الحاسب - حزماً لوكيل بروتوكول SNMP مما يسمح بإدارة منتجاتهم بواسطة محطة إدارة SNMP.

وكما يشير الاسم، يعدّ بروتوكول SNMP أداة بسيطة لإدارة الشبكة. حيث يُعرِّف قاعدة محدودة وسهلة التنفيذ لمعلومات الإدارة (MIB) تتألَّف من متغيرات قياسية (scalar variables) وجداول ثنائية الأبعاد، كما يُعرِّف أسلوباً بسيطاً لتمكين المدير من الحصول على قيم متغيرات MIB وتعديلها، وتمكين وكيلٍ ما من إرسال إخطارات بشكلٍ تلقائي (unsolicited) (تُعرَف بـ traps). وهذه البساطة هي أساس القوة لبروتوكول SNMP. علاوةً على ذلك، فإن بروتوكول SNMP سهل التنفيذ واستهلاكه متواضع لموارد الشبكة والمعالج. كما أن بنية البروتوكول وقواعد معلومات الإدارة MIB واضحة بما فيه الكفاية وليس من الصعب تحقيق التوافق للعمل البيني (interoperability) لمنتجات من شركات مختلفة لمحطات الإدارة وبرمجيات الوكلاء.

المواصفات الثلاثة الأساسية هي:

- هيكل معلومات الإدارة للشبكات المبنيَّة على مجموعة بروتوكولات TCP/IP (RFC 1155): يصف كيفية تعريف الكائنات المُدارة المضمَّنة في قاعدة معلومات الإدارة MIB.
- قاعدة معلومات الإدارة لإدارة الشبكات البينية التي تعتمد على بروتوكولات TCP/IP: MIB-II (RFC 1213)؛ يصف الكائنات المُدارة المُضمَّنة في MIB.
- بروتوكول إدارة الشبكات البسيط (RFC 1157): يُعرِّف البروتوكول المُستخدَم لإدارة هذه الكائنات.

لقد صُمِّم SNMP ليكون بروتوكولاً ضمن طبقة التطبيقات في مجموعة بروتوكولات TCP/IP، حيث يعمل من خلال بروتوكول وحدة بيانات المُستخدِم (UDP)، وقد تم تعريفه في طلب التعليقات رقم RFC 768. وبالنسبة لمحطة إدارة قائمةً بذاتها، تتحكم عملية الإدارة (manager process) في الوصول إلى قاعدة معلومات الإدارة المركزية (Central MIB) بمحطة الإدارة كما توفر واجهة لمدير الشبكة. وتنجز عملية الإدارة تلك مهام إدارة الشبكة باستخدام SNMP المبنيّ على

بروتوكول بيانات المَستخدِم (UDP) وبروتوكول الإنترنت (IP) والبروتوكولات ذات الصلة المُعتمِدة على الشبكة (مثل Ethernet، FDDI، X.25).

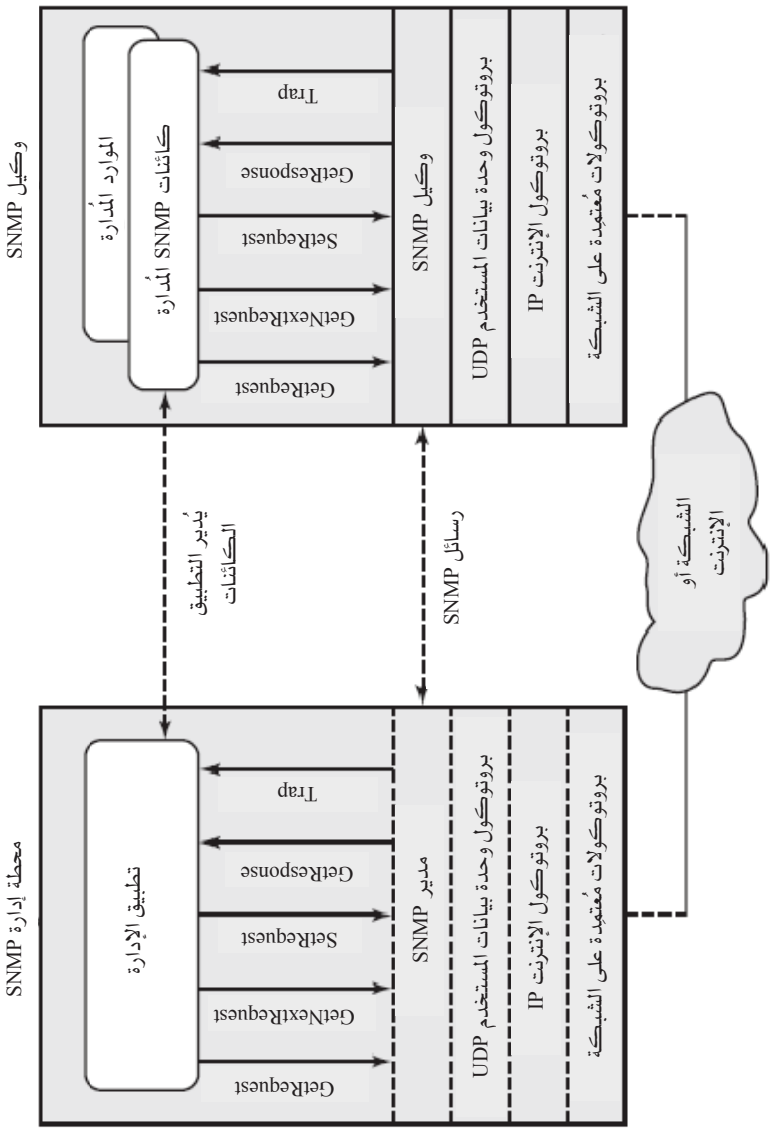
يتعين أيضاً على كل وكيل تنفيذ بروتوكولات SNMP وUDP وIP. وبالإضافة إلى ذلك، هناك عملية وكيل (agent process) تقوم بتفسير رسائل SNMP وبالتحكم في قاعدة معلومات الإدارة لديه. يحتاج جهاز وكيل يدعم تطبيقات أخرى مثل FTP إلى كلٍّ من بروتوكولي TCP وUDP.

يوضّح الشكل 8-1 سياق بروتوكول SNMP. تصدر محطة الإدارة ثلاثة أنواع من رسائل SNMP نيابةً عن تطبيقات الإدارة؛ وهي: GetRequest، وGetNextRequest، وSetRequest. والرسالتان الأوليان هما شكلان متباينان من Get. يُرَد على تلك الرسائل الثلاث من قِبَل الوكيل في شكل رسالة GetResponse، والتي يتم تمريرها إلى تطبيق الإدارة. وبالإضافة إلى ذلك، يمكن أن يُصدِر الوكيل رسالة إخطار تلقائي (trap) استجابةً لحدثٍ ما يؤثر على MIB والموارد التحتية المُدارة.

نظراً لأن SNMP يعتمد على UDP – وهو بروتوكول لاتوصيلي – فإن SNMP نفسه بروتوكول لاتوصيلي. فلا تتم المحافظة على الوصلات الجارية بين محطة الإدارة ووكلائها، وإنما يُعدُّ كل تبادل مُعامَلة منفصلة بين محطة الإدارة والوكيل.

8-1-3 المُفوَّضات (Proxies)

في الإصدار الأول من بروتوكول SNMP يجب أن تدعم كل محطات الإدارة والوكلاء (agents) بروتوكولات UDP وIP. وهذا يُقصِر الإدارة المباشرة على مثل تلك الأجهزة ويستبعد الأجهزة الأخرى كبعض الجسور وأجهزة المودم التي لا تدعم أيًّا من بروتوكولات مجموعة TCP/IP. وعلاوةً على ذلك، قد توجد نظم عديدة صغيرة (مثل أجهزة الحاسبات الشخصية، ومحطات العمل، ووحدات التحكم القابلة للبرمجة) التي تُنفِّذ فعلاً مجموعة TCP/IP من أجل دعم تطبيقاتها، لكنها لا ترغب في تحمل العبء الإضافي لبروتوكول SNMP أو لعمليات الوكيل أو لصيانة قاعدة معلومات الإدارة (MIB).



الشكل 8-1: دور/ وظيفة SNMP.

ولاستيعاب الأجهزة التي لا تُنفِّذ بروتوكول SNMP تم تطوير مفهوم المُفوَّض (Proxy). وفي هذا النظام يعدّ وكيل SNMP بمثابة مُفوَّض (Proxy) لواحدٍ أو أكثر من الأجهزة الأخرى؛ بمعنى آخر يتصرف وكيل SNMP نيابةً عن تلك الأجهزة المُفوِّضة (proxied devices).

يبيّن الشكل 8-2 النوع المُستخدَم في الغالب لبنية البروتوكول. حيث تُرسِل محطة الإدارة استفسارات عن جهاز معين إلى الوكيل المُفوَّض، فيقوم الوكيل المُفوَّض بتحوّيل كل استفسار إلى بروتوكول الإدارة الذي يستخدمه الجهاز المعني. وعندما يتلقى الوكيل رداً على استفسارٍ ما، يقوم بتمريره إلى محطة الإدارة. وبالمثل، إذا تم إرسال إبلاغ عن حدثٍ من أي نوع إلى المُفوَّض، يقوم المُفوَّض بدوره بإرساله إلى محطة الإدارة في شكل رسالة إخطار تلقائي.

يُتيح الإصدار الثاني من بروتوكول SNMP ليس فقط استخدام مجموعة بروتوكولات TCP/IP بل يسمح بغيرها كذلك. ويستهدف SNMPv2 تحديداً العمل على سلة بروتوكولات الأنظمة المفتوحة (OSI). ومن ثَمَّ، يمكن استخدام SNMPv2 لإدارة تشكيلة أكبر من التكوينات الشبكية. وفيما يتعلق بالمُفوَّضات، فإن الأجهزة التي لا تدعم تنفيذ SNMPv2 يمكن إدارتها فقط بواسطة المُفوَّض. ويشمل ذلك حتى الأجهزة التي تدعم SNMPv1؛ أي أنه في حالة جهاز يُنفِّذ برنامج الوكيل لبروتوكول SNMPv1، يمكن الوصول إليه من مدير SNMPv2 فقط عن طريق جهاز مُفوَّض يُنفِّذ وكيل SNMPv2 ومدير SNMPv1.

في بروتوكول SNMPv2 تُعرَف الحالات المُبيَّنة في الفقرة السابقة بعلاقات المُفوَّض الخارجي (foreign proxy relationships). وبالإضافة إلى ذلك، يدعم SNMPv2 علاقات المُفوَّضين المحليين التي يدعم فيها الجهاز المُفوِّض بروتوكول SNMPv2. وفي هذه الحالة، يتصل مدير SNMPv2 بوكيل SNMPv2 والذي يقوم بدور المدير للوصول إلى الجهاز المُفوِّض الذي يقوم بدور وكيل SNMPv2. والهدف من دعم هذا النوع من التعامل غير المباشر هو تمكين المُستخدِمين من تهيئة أنظمة شبكات هرمية لامركزية الإدارة، كما سنناقش لاحقاً.

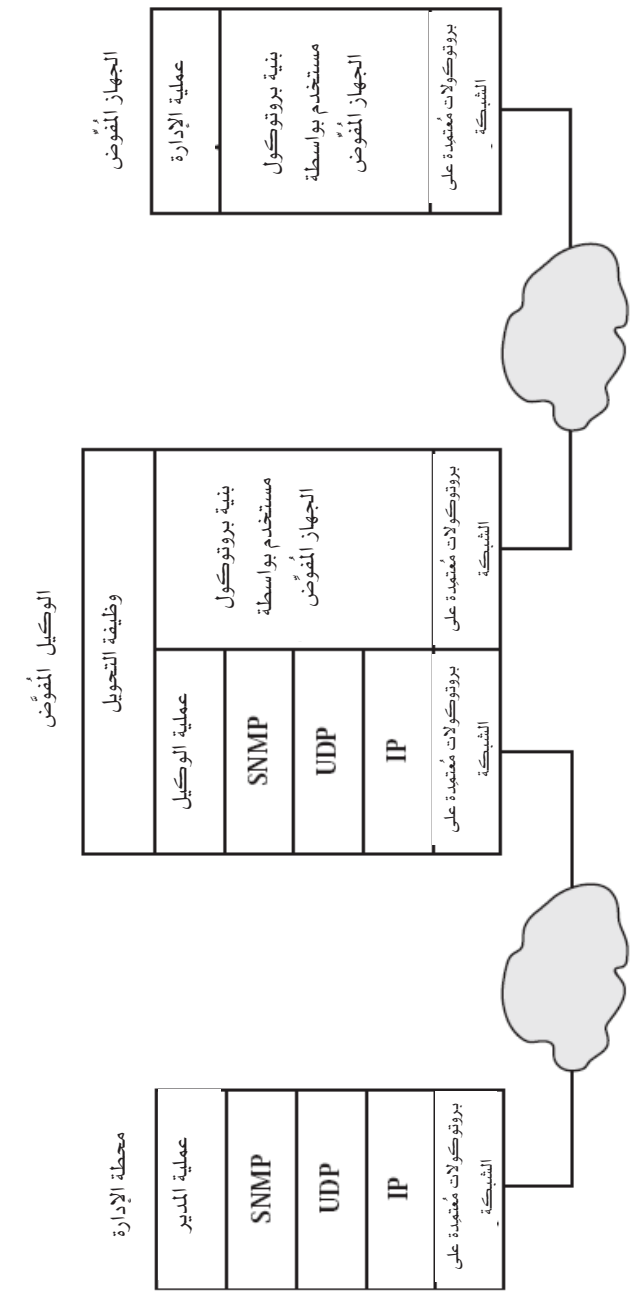
8-1-4 الإصدار الثاني لبروتوكول إدارة الشبكات البسيط (SNMPv2)

تكمن قوة بروتوكول SNMP في بساطته، حيث يوفر مجموعة أساسية من أدوات إدارة الشبكة في حزمة سهلة التنفيذ وسهلة التهيئة. ومع زيادة اعتماد المُستخدِمين أكثر وأكثر على SNMP لإدارة الشبكات والتي بدورها تزداد توسعاً وتزداد حمولتها، بدأت تتضح أوجه القصور في هذا البروتوكول، ويمكن تقسيمها إلى ثلاث فئات:

- الافتقار إلى الدعم المطلوب لإدارة الشبكات الموزَّعة.
- أوجه القصور الوظيفية.
- القصور الأمني.

تمت معالجة الفئتين الأوليين من أوجه القصور في SNMPv2، والذي صدر في عام 1993، ثم صدرت نسخته المنقّحة (RFCs 1901, 1904 ... 1908, 2578, 2579) في عام 1996. وسرعان ما اكتسب SNMPv2 الدعم، وأُعلِنت المنتجات من عدد من الشركات خلال أشهر من إصداره. أما أوجه القصور الأمني فقد تمت معالجتها في الإصدار الثالث للبروتوكول (SNMPv3).

سنلخص بإيجاز فيما تبقى من هذا الجزء السمات الجديدة المتوفرة في SNMPv2. وبعد ذلك سنقوم بفحص سمات الأمن لكلٍّ من SNMPv1 وSNMPv3 بالتفصيل في الجزأين 8-2 و8-3 على التوالي.



الشكل 8-2: تهيئة المُفوَّض.

❖ إدارة الشبكات الموزَّعة

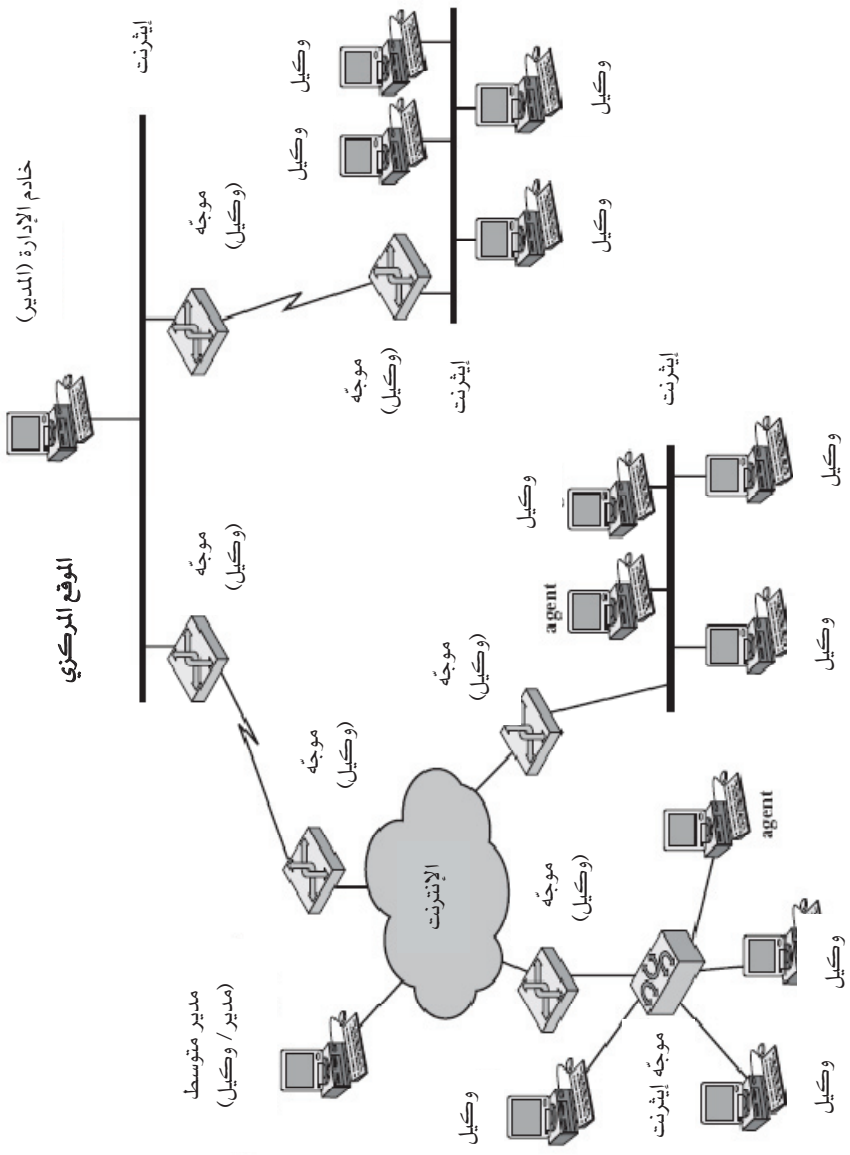
في النظم التقليدية للإدارة المركزية للشبكات، يقوم نظام مضيف واحد بدور محطة إدارة الشبكة؛ وقد تكون هناك محطة إدارة احتياطية أو اثنتان. أما ما تبقى من أجهزة الشبكة فيحتوي على برامج الوكلاء (agent software) وقاعدة معلومات الإدارة (MIB)، وذلك للسماح بالمراقبة والتحكم من محطة الإدارة. ونظراً لنمو حجم الشبكات وزيادة حركة المرور بها، أصبح مثل هذا النظام المركزي غير قابل للتطبيق عملياً. يرجع ذلك إلى العبء الكبير الذي تتحمله محطة الإدارة، إضافةً إلى حركة المرور العالية من التقارير الواردة من كل وكيل على حِدَة والتي يجب أن تشق طريقها إلى المقر الرئيس عبر الشبكة بأكملها. في مثل هذه الظروف، تعمل الطريقة اللامركزية الموزَّعة بشكلٍ أفضل (انظر الشكل 8-3). ففي نظام لامركزي لإدارة الشبكات، يمكن وجود عدد من محطات الإدارة بالمستوى الأعلى، والتي قد يُشار إليها بخادمات الإدارة (management servers). ويمكن لكل خادم منها أن يدير مباشرة بعضاً من مجموعة الوكلاء. وقد يُفوِّض خادم الإدارة مسؤولية إدارة عدد من الوكلاء إلى مدير وسيط، والذي يقوم بدور المدير في المراقبة والتحكم في الوكلاء المُندرجين تحت مسؤوليته، كما يقوم أيضاً بأداء دور الوكيل في تقديم المعلومات وتَلَقِّي التعليمات من خادم إدارة في مستوى أعلى. هذا النوع من البنية المعمارية يوزع عبء المعالجة ويقلل حركة المرور الكلية على الشبكة.

يدعم SNMPv2 استراتيجية شديدة المركزية أو استراتيجية موزَّعة لإدارة الشبكات. في الحالة الأخيرة، تؤدي بعض النظم دوراً مزدوجاً حيث تضطلع بدور المدير والوكيل. فعندما تقوم بدور الوكيل، تتلقى تلك النظم الأوامر من نظام إدارة أعلى. تتعلق بعض هذه الأوامر بقاعدة معلومات الإدارة المحلية لدى الوكيل، في حين تتطلب أوامر أخرى من الوكيل القيام بدور المُفوَّض للأجهزة البعيدة. وفي هذه الحالة، يتولى الوكيل المُفوَّض أولاً دور المدير للوصول لمعلومات عند وكيل بعيد ثم يتولى دور الوكيل لتمرير تلك المعلومات إلى مدير أعلى.

❖ تحسينات وظيفية

يشير الجدول 8-1 إلى التحسينات الوظيفية التي أضيفت في SNMPv2 مقارنةً بالإصدار الأول (SNMPv1). ويُعرَّف كلٌ من البروتوكولين بمجموعة الأوامر التي يُرسلها في صورة وحدات بيانات بروتوكول (PDU). وفي حالة SNMPv1، هناك خمسة أوامر. يُصدر المدير الأمر Get إلى وكيل لاسترجاع قيم كائنات معيّنة في قاعدة معلومات الإدارة (MIB). أما الأمر GetNext فيستغل البنية الشجرية (الهرمية) للكائنات في MIB. عندما تتم تسمية كائن في الأمر GetNext، يبحث الوكيل عن الكائن التالي في الشجرة ويسترجع القيمة الخاصة به. ويُعدّ الأمر GetNext مفيداً لأنه يسمح لمديرٍ ما بتصفّح بنية شجرية عند وكيلٍ ما في حالة عدم معرفة مجموعة الكائنات المدعومة من قِبَل ذلك الوكيل بالضبط. ويتيح الأمر Set لمديرٍ ما تحديث القيم لدى وكيلٍ ما؛ كما يُستخدَم أيضاً في إضافة الصفوف في الجداول وحذفها. أما الأمر GetResponse فيُستخدَم من قِبَل وكيل للاستجابة لتعليمات المدير. وأخيراً، يُمكِّن الأمر Trap وكيلاً ما من إرسال معلومات إلى مدير دون انتظار طلب من الإدارة. فمثلاً؛ يمكن تهيئة وكيل ليرسل إخطاراً تلقائياً عند فشل ارتباط معين أو عند تجاوز حركة المرور حداً معيناً.

يتضمن SNMPv2 كافة الأوامر الموجودة في SNMPv1 إضافةً إلى أمرين جديدين؛ أهمهما الأمر Inform، والذي يتم إرساله من محطة إدارة إلى أخرى، ويتضمن معلومات تتعلق بظروف أو أحداث في الجهاز المُرسِل، كما في الأمر Trap. يتميز الأمر Inform بإمكانية استخدامه لبناء تكوين يتعاون فيه عدد من المديرين في مسؤولية إدارة شبكة كبيرة.



الشكل 8-3: مثال لنظام موزَّع لإدارة الشبكة.

الجدول 8-1: مقارنة وحدات بيانات البروتوكول (PDUs) في SNMPv1، وSNMPv2.

الوصف	الاتجاه	SNMPV2 PDU	SNMPV1 PDU
طلب قيمة لكل كائن مسجّل في القائمة	من مدير إلى وكيل	GetRequest	GetRequest
طلب القيمة التالية لكل كائن مسجّل في القائمة	من مدير إلى وكيل	GetNextRequest	GetNextRequest
طلب قيم متعددة	من مدير إلى وكيل	GetBulkRequest	--
تعيين قيمة لكل كائن مسجّل في القائمة	من مدير إلى وكيل	SetRequest	SetRequest
إرسال معلومات غير مطلوبة	من مدير إلى مدير	InformRequest	--
رد على طلب مدير	من وكيل إلى مدير أو من مدير إلى مدير (SNMPv2)	Response	GetResponse
إرسال معلومات غير مطلوبة	من وكيل إلى مدير	SNMPv2-Trap	Trap

الأمر الجديد الآخر هو GetBulk، ويتيح لمديرٍ ما استرداد كتلة كبيرة من البيانات في وقتٍ واحد، وبالتحديد تم تصميم الأمر GetBulk لنقل جداول كاملة باستخدام أمر واحد.

فرق أخير: يُعدُّ الأمر Get أمراً أوّليًّا (atomic) في حالة SNMPv1 لكنه ليس كذلك في حالة SNMPv2. فمثلاً إذا تطلب أمر Get في بروتوكول SNMPv1 استرجاع قيم عدد من الكائنات وكان أحد تلك الكائنات (على الأقل) غير موجود عند المُفوَّض، يتم رفض الأمر بأكمله. أما بالنسبة لـ SNMPv2 فيسمح الأمر Get

باسترجاع نتائج جزئية، ولذا يسمح أمر Get غير الأوَّليّ (nonatomic) للمدير باستخدام قدرات الشبكة بشكلٍ أكفأ.

2-8 تسهيلات تجمعات SNMPv1

يوفِّر SNMPv1 – كما تم تعريفه في RFC 1157 – تسهيلات أمن بدائية مبنيَّة على مفهوم التجمع. تتيح تلك التسهيلات مستوىً معيناً من الأمن ولكنه أمنٌ مُعرَّضٌ لشتى أنواع الهجمات [JIAN02، CERT02].

1-2-8 التجمعات وأسماؤها (Communities and Community Names)

تنطوي إدارة الشبكات مثل غيرها من التطبيقات الموزَّعة الأخرى على التفاعل بين عدد من كيانات التطبيق المُعتمِدة على بروتوكول تطبيقٍ ما. وفي حالة إدارة الشبكة ببروتوكول SNMP، تكون كيانات التطبيق هي تطبيقات المدير وتطبيقات الوكيل التي تَستخدِم بروتوكول SNMP. تتسم إدارة الشبكة ببروتوكول SNMP بعدة خصائص غير معهودة في التطبيقات الموزَّعة الأخرى.

ينطوي التطبيق على علاقة "واحد إلى متعدد" (one-to-many) بين مدير ومجموعة من الوكلاء: فالمدير قادر على الحصول على قِيَم كائنات في الوكلاء وتعديلها، وقادر أيضاً على تَلَقِّي الإخطارات التلقائية من الوكلاء. وهكذا من وجهة النظر التشغيلية أو من وجهة نظر التحكم، فإن المدير "يدير" عدداً من الوكلاء. وقد يكون هناك عددٌ من المديرين يدير كلٌّ منهم كافة الوكلاء في تشكيلة النظام أو مجموعة جزئية منهم. ويمكن أن تتداخل هذه المجموعات الجزئية.

نحتاج أيضاً للنظر إلى إدارة شبكة SNMP بوصفها علاقة "واحد إلى متعدد" بين وكيل ومجموعة من المديرين. حيث يتحكم كل وكيل في عناصر قاعدة معلومات الإدارة المحلية الخاصة به، ويجب أن يكون قادراً على التحكم في استخدام تلك القاعدة من قِبَل عددٍ من المديرين. هناك ثلاثة جوانب لهذا التحكم:

- خدمة التوثيق: قد يرغب الوكيل في قصر الوصول إلى قاعدة معلومات الإدارة على المديرين المُرخّص لهم.
- سياسة الوصول: قد يرغب الوكيل في إعطاء امتيازات وصول متباينة لمديرين مختلفين.
- خدمة المُفوَّض: قد يقوم الوكيل بدور المُفوَّض لوكلاء آخرين. وقد يشمل ذلك تنفيذ خدمة التوثيق: و/أو سياسة الوصول بالنسبة للوكلاء الآخرين على النظام المُفوَّض.

تتصل جميع هذه الجوانب بالشؤون الأمنية. ففي بيئة تقسَّم فيها المسؤولية عن مكوِّنات الشبكة بين عدد من الكيانات الإدارية، يحتاج الوكلاء إلى حماية أنفسهم وحماية قواعد معلومات الإدارة الخاصة بهم من الوصول غير المرغوب فيه أو غير المصرّح به. يقوم SNMP – كما تم تعريفه في طلب تعليقات RFC 1157 – بتوفير قدرة بدائية ومحدودة لمثل هذا الأمن؛ ألا وهو مفهوم "التجمُّع" (Community).

تجمُّع SNMP هو علاقة تُعَرِّف التوثيق والتحكم في الوصول وخصائص المُفوَّض بين وكيل ومجموعة من المديرين في SNMP. يعدّ مفهوم التجمُّع مفهوماً محلياً يتم تعريفه لدى الوكيل. حيث يؤسِّس الوكيل عدة تجمعات لكل تجمعٍ منها نفس المزيج من خدمة التوثيق، وسياسة التحكم في الوصول، وخصائص المُفوَّض. ويتم إعطاء كل تجمُّع اسماً فريداً (لدى هذا الوكيل). ويتم تعريف المديرين من أعضاء هذا التجمُّع بهذا الاسم والذي يجب أن يستخدمونه في جميع عمليات أوامر Get وSet. ويمكن للوكيل إنشاء عددٍ من التجمعات التي قد تتداخل نتيجة عضوية مدير بأكثر من تجمُّع.

نظراً لأن تعريف التجمعات يتم محلياً عند الوكيل، فيمكن أن يستخدم وكلاء مختلفون نفس الاسم. وليست هوية هذه الأسماء ذات صلة ولا تشير إلى أي تشابه بين التجمعات المُعرَّفة. ولذا يجب على المدير أن يتتبع أسماء التجمعات المقترنة بكل الوكلاء المرغوب في الوصول إليها.

8-2-2 خدمة التوثيق

الغرض من خدمة التوثيق في SNMPv1 هو التأكيد للمستلم أن رسالة SNMPv1 التي استلمها هي من المصدر الذي تدّعي أنها منه. يوفّر SNMPv1 إمكانية متواضعة للتوثيق. تتضمن كل رسالة Put أو Get من مدير إلى وكيل اسم تجمّع. ويُعدُّ هذا الاسم بمثابة كلمة سر، وتُعدُّ الرسالة موثّقة إذا كان المرسل يعرف كلمة السر.

مع هذا الشكل المحدود من التوثيق، يُحجِم عددٌ من مديري الشبكات عن السماح لأي شيء بخلاف مراقبة الشبكة؛ أي عمليات Get وTrap. ومن الواضح أن التحكم في الشبكة، عن طريق Set، يعدّ موضوعاً أكثر حساسية. ويمكن أن يُستخدم اسم التجمُّع لتشغيل إجراءات التوثيق، حيث يُوظف الاسم ببساطة كجهاز أوَّليّ لفحص كلمة السر. ويمكن أن تشمل إجراءات التوثيق استخدام التشفير وإزالة التشفير لدوال توثيق أكثر أمناً، وهذا يتجاوز نطاق RFC 1157.

8-2-3 سياسة الوصول (Access Policy)

يقصر الوكيل الوصول إلى قاعدة معلومات الإدارة الخاصة به على مجموعة مختارة من المديرين عن طريق تعريف تجمُّع يضم تلك المجموعة من المديرين. باستخدام أكثر من تجمُّع، يمكن أن يوفّر الوكيل فئات مختلفة من حقوق الوصول إلى MIB لمديرين مختلفين. وهناك جانبان لهذا التحكم في الوصول:

- مشهد قاعدة معلومات الإدارة في SNMP: مجموعة جزئية من الكائنات الموجودة في MIB. ويمكن تعريف مشهد مختلف لكل تجمُّع. ومجموعة الكائنات بمشهد معين لا تنتمي بالضرورة إلى شجرة فرعية واحدة من قاعدة معلومات الإدارة.
- نمط الوصول في SNMP: هناك نمطان فقط؛ الأول: نمط "قراءة - فقط" (Read-Only)، والثاني: نمط "قراءة - كتابة" (Read-Write)؛ يتم تحديد نمط وصول لكل تجمُّع.

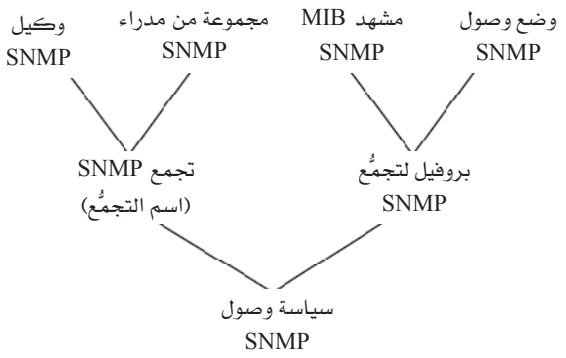
يُشار إلى تركيبة مشهد قاعدة معلومات الإدارة ونمط الوصول بالشكل العام (بروفيل) لتجمُّع SNMP. وهكذا، يتكون الشكل العام للتجمع من مجموعة جزئية من كائنات MIB مُعرَّفة عند الوكيل، بالإضافة إلى نمط وصول لتلك الكائنات. يُطبَّق نمط الوصول في SNMP بشكلٍ موحد على كافة الكائنات الموجودة في مشهد MIB. ولذا فإنه إذا تم اختيار نمط الوصول "قراءة - فقط"، فإنه ينطبق على كافة الكائنات في المشهد ويقتصر وصول المديرين على عمليات "قراءة - فقط".

يقترن الشكل العام للتجمُّع مع كل تجمُّع يتم تعريفه من قِبَل الوكيل ؛ ويُشار إلى التركيبة المكوَّنة من تجمُّع SNMP والشكل العام له بـ "سياسة وصول SNMP". ويوضّح الشكل 8-4 مختلف المفاهيم سالفة الذكر.

8-2-4 خدمة المُفوَّض

مفهوم التجمُّع مفيدٌ أيضاً في دعم خدمة المُفوَّض. وهنا نُذكِّر بأن المُفوَّض هو وكيل SNMP يعمل نيابةً عن أجهزة أخرى. وفي العادة تكون تلك الأجهزة الأخرى أجهزة خارجية بمعنى أنها لا تدعم بروتوكولات TCP/IP وSNMP. وفي بعض الحالات، قد يدعم النظام المفوِّض بروتوكول SNMP ولكنه بالرغم من ذلك يستخدم المُفوَّض ليقلل من تعاملاته مع نظم إدارة الشبكة.

يحتفظ النظام المُفوَّض بسياسة SNMP للوصول لكل جهاز يمثله. ولذا فهو يعرف أي الكائنات التي في MIB يمكن استخدامها لإدارة النظام المفوِّض (proxied system) ونمط الوصول الخاص بها.



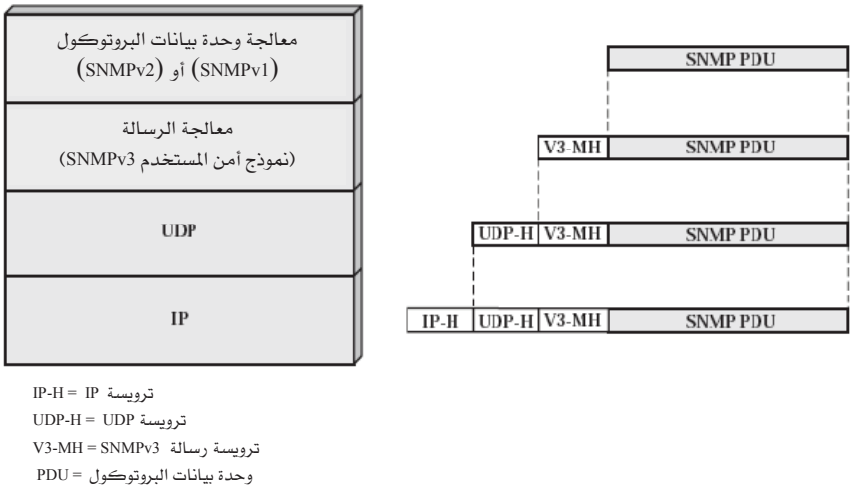
الشكل 8-4: المفاهيم الإدارية لبروتوكول SNMPv1.

8-3 الإصدار الثالث لبروتوكول إدارة الشبكات البسيط

في عام 1998 وضعت مجموعة عمل SNMPv3 – التابعة لـ فريق مهام هندسة الإنترنت (IETF) – مجموعة من معايير الإنترنت المقترحة، والمعروفة حالياً بمجموعة طلبات التعليقات من RFC 2570 حتى RFC 2576. تُحدد هذه المجموعة من الوثائق إطاراً لدمج السمات الأمنية في إمكانية شاملة تتضمن وظائف SNMPv1 أو SNMPv2. كما تُعَرِّف الوثائق أيضاً مجموعة محدّدة من الإمكانات لأمن الشبكات والتحكم في الوصول.

من المهم إدراك أن SNMPv3 ليس بديلاً عن أيٍّ من SNMPv1 أو SNMPv2 ولا لكليهما. حيث يُعرِّف SNMPv3 قدرات أمنية لاستخدامها مع SNMPv2 (المفضَّل) أو SNMPv1. بالإضافة إلى ذلك، يصف RFC 2571 بنية معمارية تُلائم جميع إصدارات SNMP الحالية والمستقبلية. كما يصف RFC 2575 وسيلة للتحكم في الوصول، والتي يقصد منها أن تعمل بشكلٍ مستقل عن الإمكانات الأساسية لـ SNMPv3. سنقوم في هذا الجزء بإلقاء نظرة شاملة وعمل دراسة مسحية عن الإمكانات المُعرَّفة في RFC 2570 حتى RFC 2576 .

يبيّن الشكل 8-5 العلاقة بين الإصدارات المختلفة من SNMP من خلال الصيغ المُستخدَمة. ويتم تبادل المعلومات بين محطة إدارة ووكيل عن طريق رسالة SNMP. وتتم المعالجة الأمنية على مستوى الرسالة؛ فعلى سبيل المثال، يُحدِّد SNMPv3 نموذجاً لأمن المُستخدِم ((User Security Model (USM) والذي يستخدم حقول ترويسة الرسالة. وتكون حمولة رسالة SNMP إما في شكل وحدة بيانات بروتوكول SNMPv1 (SNMPv1 PDU) أو وحدة بيانات بروتوكول SNMPv2 (SNMPv2 PDU). تُشير وحدة بيانات البروتوكول (PDU) إلى نوع من الإجراءات الإدارية (مثل استرداد أو تعيين قيم الكائنات المُدارة)، وتُشير أيضاً إلى قائمة بأسماء المتغيرات ذات العلاقة بذلك الإجراء.



الشكل 8-5: البنية المعمارية لبروتوكول SNMP.

تصف طلبات التعليقات من RFC 2570 حتى RFC 2576 بنية معمارية شاملة بالإضافة إلى السمات الأمنية والهياكل المحدّدة للرسائل، لكنها لا تُعرِّف صيغاً جديدة لوحدة بيانات البروتوكول (PDU) في SNMP. ولذلك، يجب استخدام صيغ PDU الموجودة في SNMPv1 أو SNMPv2 داخل البنية الجديدة. تتضمن تطبيقات

SNMPv3 السمات الأمنية والمعمارية المُعرَّفة في RFC 2570 حتى RFC 2576 إضافةً إلى صيغة PDU والوظائف المُعرَّفة في وثائق SNMPv2. وقد تم التعبير عن ذلك في RFC 2570 بالنص التالي: "يمكن اعتبار SNMPv3 على أنه SNMPv2 مع بعض الإمكانات الأمنية والإدارية المضافة."

فيما تبقى سنعرض في هذا الجزء أولاً مقدمة موجزة للبنية المعمارية الأساسية لبروتوكول SNMP المُعرَّفة في RFC 2571، ثم نتناول ميزات السرية والتوثيق التي يوفرها نموذج أمن المُستخدِم (USM) لبروتوكول SNMPv3. وأخيراً، سنناقش التحكم في الوصول ونموذج التحكم في الوصول المبنيّ على المشهد (View-based Access Control Model (VACM)).

8-3-1 البنية المعمارية لبروتوكول SNMP

تتكون البنية المعمارية لـ SNMP – كما هو معروض في RFC 2571 – من مجموعة كيانات SNMP موزَّعة ومتفاعلة. ويقوم كل كيان بتنفيذ جزء من إمكانات SNMP وقد يتصرف كوكيل أو مدير أو مزيج من الاثنين معاً. ويتكون كيان SNMP من مجموعة من الوحدات (modules) التي تتفاعل مع بعضها بعضاً لتوفير الخدمات. يمكن نمذجة هذه التفاعلات كمجموعة من العمليات الأوَّليَّة والمُعامِلات.

تعكس البنية المعرَّفة في RFC 2571 متطلباً رئيساً لتصميم SNMPv3 وهو: تصميم بنية معمارية مكونة من وحدات والتي سوف: (1) تسمح بالتنفيذ عبر مجموعة واسعة من البيئات التشغيلية، والتي قد يحتاج بعضها إلى استخدام القليل من الإمكانات غير المكلِّفة، في حين قد يدعم البعض الآخر منها ميزات إضافية لإدارة شبكات كبيرة؛ (2) تجعل من الممكن إحداث تقدم من حيث إقرار معيارية بعض أجزاء البنية المعمارية حتى لو لم يتم التوصل إلى اتفاق موحَّد على كافة الأجزاء؛ و(3) يمكنها استيعاب نماذج أمنية بديلة.

❖ كيان SNMP

يتضمَّن كل كيان SNMP مُحرّك SNMP واحداً. ويقوم محرك SNMP بتنفيذ وظائف إرسال الرسائل واستقبالها، وتوثيق الرسائل وتشفيرها وإزالة تشفيرها، والتحكم في الوصول إلى الكائنات المُدارة. ويتم توفير هذه الوظائف كخدمات لتطبيق واحد أو لعدد أكبر من التطبيقات والتي يتم تهيئتها بمحرك SNMP لتُشكِّل "**كيان** SNMP".

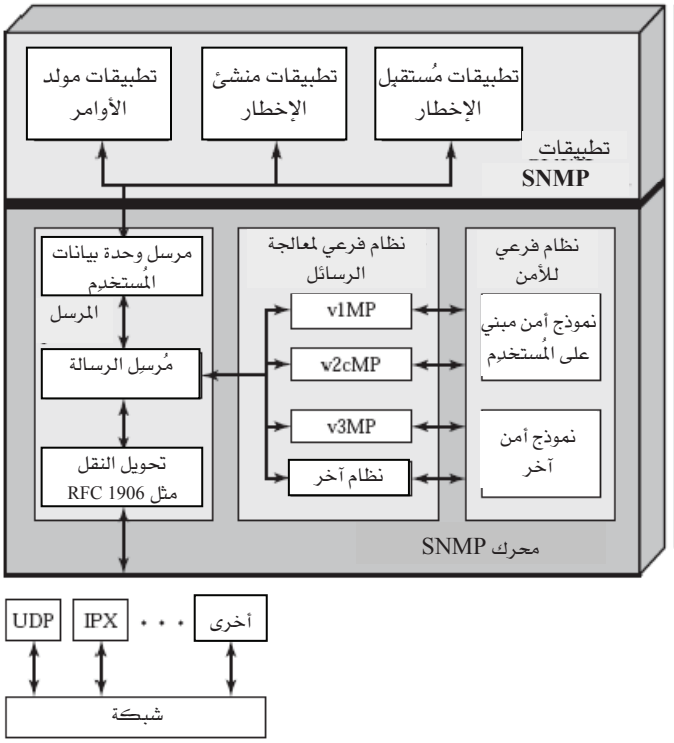
يُعرَّف كلٌّ من محرك SNMP والتطبيقات التي يدعمها بمجموعة من الوحدات المنفصلة. وتوفر هذه البنية المعمارية مزايا عدة كما يأتي:

أولاً: يتم تحديد دور كيان SNMP على أساس الوحدات المُنفَّذة في هذا الكيان. يحتاج وكيل SNMP إلى مجموعة معيّنة من الوحدات بينما يحتاج مدير SNMP إلى مجموعة مختلفة (مع وجود بعض التطابق) من الوحدات.

ثانياً: يسهل الهيكل التركيبي (modular structure) للمواصفات من عملية تعريف إصدارات مختلفة لكل وحدة. وهذا بدوره يجعل من الممكن: (1) تعريف إمكانات بديلة أو محسنَّة لجوانب معيّنة من SNMP دون الحاجة إلى إصدار جديد للمعيار القياسي بأكمله (مثلاً: SNMPv4)، و(2) تحديد واضح لاستراتيجيات التعايش والانتقال بين الإصدارات المختلفة (RFC 2576).

لفهم دور كل وحدة وعلاقتها بالوحدات الأخرى بشكلٍ أفضل، نحتاج إلى النظر إلى استخداماتها في مديري ووكلاء SNMP التقليديين. ويُستخدَم المصطلح "تقليدي" بمعنى نقي، ليؤكد على حقيقة أن تنفيذاً معيناً لا يلزم أن يكون مديراً أو وكيلاً خالصاً ولكن قد يكون لديه الوحدات التي تسمح للكيان بأداء كلٍّ من مهام الوكيل والمدير.

كيان SNMP



الشكل 8-6: مدير SNMP التقليدي.

يعتمد الشكل 8-6 على شكل في RFC 2571، ويمثل رسماً تخطيطياً لمدير SNMP تقليدي. يتفاعل مدير SNMP التقليدي مع وكلاء SNMP عن طريق إصدار أوامر Get وSet وباستقبال رسائل الإخطارات التلقائية (Trap)؛ كما يمكن أن يتفاعل أيضاً مع مديرين آخرين من خلال إصدار وحدات بيانات البروتوكول لطلب الإبلاغ (Inform Request PDUs) – والتي توفر تنبيهات – وكذلك تَلَقِّي رسائل وحدات بيانات البروتوكول كاستجابة للإبلاغ (Inform Response PDUs)، والتي تقر باستلام طلبات إبلاغ (Inform Requests). في مصطلحات SNMPv3، يتضمن مدير SNMP التقليدي ثلاث فئات من التطبيقات. تطبيقات مولِّد الأوامر (Command Generator Applications) والتي تراقب وتعالج بيانات الإدارة في الوكلاء البعيدين – باستخدام وحدات بيانات بروتوكول SNMPv1 و/أو SNMPv2

– ويشمل ذلك أوامر Get وGetNext وGetBulk وSet. أما تطبيق مُنشئ الإخطارات (Notification Originator Application) فيقوم بإرسال الرسائل غير المتزامنة. ويتم استخدام وحدة بيانات البروتوكول لطلب الإبلاغ (Inform Request PDU) لهذا التطبيق في حالة مدير تقليدي. ويقوم تطبيق مُستقبِل الإخطارات (Notification Receiver Application) بمعالجة ما يَرِد من الرسائل غير المتزامنة؛ والتي تشمل طلب الإبلاغ (InformRequest)، وإخطارات SNMPv2 التلقائية، ووحدات بيانات بروتوكول SNMPv1 للإخطارات التلقائية. ويستجيب هذا التطبيق (مُستقبِل الإخطارات) بإرسال وحدة بيانات البروتوكول للاستجابة (Response PDU) في حالة ورود وحدة بيانات البروتوكول لطلب الإبلاغ (InformRequest PDU).

تستفيد جميع التطبيقات المشروحة آنفاً من الخدمات التي يقدمها محرك SNMP لهذا الكيان. ويؤدي محرك SNMP مهمتين عامتين:

- يستقبل وحدات بيانات البروتوكول الصادرة من تطبيقات SNMP؛ ويقوم بالمعالجة الضرورية، بما في ذلك إدراج شفرات/ رموز التوثيق والتشفير؛ ومن ثمَّ يقوم بتغليف وحدات بيانات البروتوكول داخل رسائل للنقل.
- يستقبل رسائل SNMP الواردة من طبقة النقل؛ ويقوم بالمعالجة الضرورية بما في ذلك التوثيق وإزالة التشفير؛ ومن ثم يستخلص وحدات بيانات البروتوكول من الرسائل ويمررها إلى تطبيق SNMP المناسب.

يحتوي محرك SNMP في حالة المدير التقليدي على مُرسِل، ونظام فرعي لمعالجة الرسائل، ونظام فرعي للأمن. ويكون المُرسِل هنا هو مجرد مدير بسيط لحركة المرور. وبالنسبة لوحدات بيانات البروتوكول الصادرة، يستقبل المُرسِل وحدات بيانات البروتوكول من التطبيقات ويؤدي الوظائف الآتية: يحدِّد المُرسِل نوع المعالجة المطلوبة لكل وحدة بيانات (على سبيل المثال: SNMPv1، أو SNMPv2c، أو SNMPv3)، ثم يمرِّرها إلى الوحدة المناسبة لمعالجة الرسالة في النظام الفرعي لمعالجة الرسائل. بعد ذلك يقوم النظام الفرعي لمعالجة الرسائل بإرجاع رسالة تحتوي على وحدة بيانات البروتوكول تلك بالإضافة إلى ترويسة مناسبة للرسالة. بعدها يقوم المُرسِل بترحيل تلك الرسالة إلى طبقة النقل.

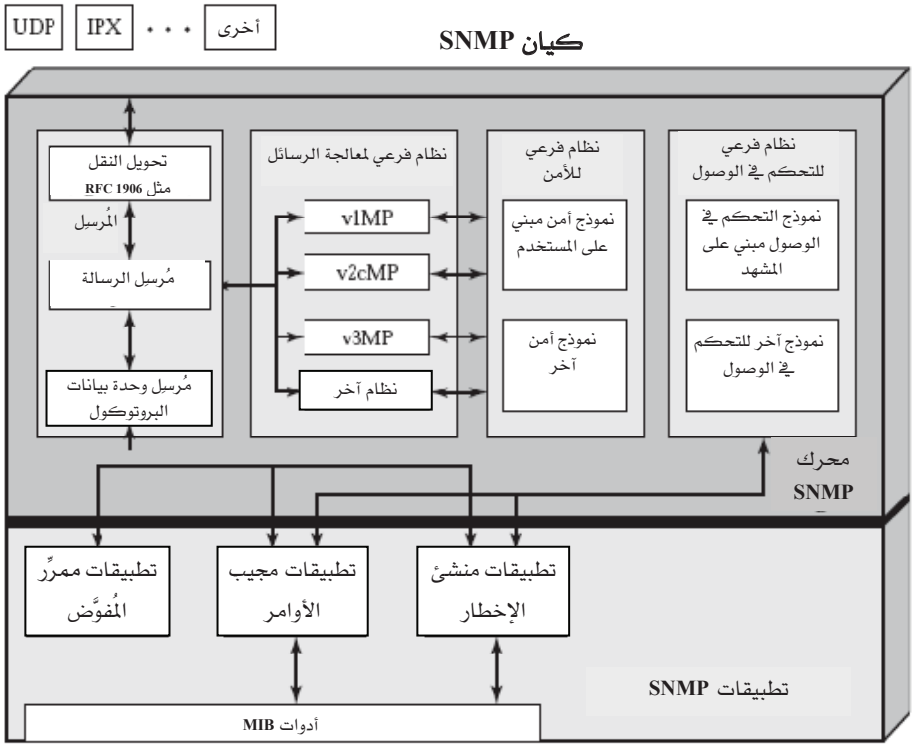
بالنسبة للرسائل الواردة، يستقبل المُرسِل الرسائل من طبقة النقل ويؤدي الوظائف الآتية. يقوم المُرسِل بتوجيه كل رسالة إلى الوحدة المناسبة لمعالجة الرسالة. بعد ذلك يعيد النظام الفرعي لمعالجة الرسائل وحدة بيانات البروتوكول الواردة بالرسالة، فيمررها المُرسِل إلى التطبيق المناسب.

يقوم النظام الفرعي لمعالجة الرسائل باستقبال PDUs الصادرة من المُرسِل وتجهيزها للنقل بتغليفها في ترويسة مناسبة ثم إعادتها إلى المُرسِل. أما بالنسبة للرسائل الواردة فيقوم النظام الفرعي لمعالجة الرسائل باستقبال الرسائل أيضاً من المُرسِل، ومعالجة ترويسة كل رسالة، ومن ثَمَّ إعادة PDU المرفقة بها إلى المُرسِل. وقد يدعم تطبيق النظام الفرعي لمعالجة الرسائل صيغة واحدة للرسائل موافقة لإصدار واحد من SNMP (SNMPv1، SNMPv2c، SNMPv3)، أو قد يحتوي النظام على عددٍ من الوحدات يدعم كلٌ منها إصداراً مختلفاً من SNMP.

أما النظام الفرعي للأمن فيقوم بأداء وظائف التشفير والتوثيق. يتم تمرير كل رسالة صادرة من النظام الفرعي لمعالجة الرسائل إلى النظام الفرعي للأمن. بناءً على الخدمات المطلوبة، قد يقوم النظام الفرعي للأمن بتشفير PDU المرفقة وربما بعض الحقول في ترويسة الرسالة، وقد يقوم بتوليد شفرة التوثيق وإدرجها في ترويسة الرسالة. ثم يتم إعادة الرسالة المُعالَجة إلى النظام الفرعي لمعالجة الرسائل. وبالمثل يتم تمرير كل رسالة واردة إلى النظام الفرعي للأمن من النظام الفرعي لمعالجة الرسائل، حيث يتحقق النظام الفرعي للأمن من رمز التوثيق – إذا تطلب الأمر ذلك – ويقوم بإزالة التشفير. ثم يقوم بإعادة الرسالة المُعالَجة إلى النظام الفرعي لمعالجة الرسائل. وقد يدعم تطبيق النظام الفرعي للأمن واحداً أو أكثر من نماذج الأمن المتميزة. وحتى الآن، يُعدّ نموذج الأمن الوحيد المُعرَّف هو نموذج أمن المُستخدِم (USM) لبروتوكول SNMPv3، والمُعرَّف في RFC 2574.

الشكل 8-7 – والمُعتمِد على شكل في RFC 2571 – عبارة عن رسم تخطيطي لوكيل SNMP تقليدي والذي قد يتضمن ثلاثة أنواع من التطبيقات. توفّر تطبيقات مجيب الأوامر (Command Responder) الوصول إلى بيانات الإدارة. تستجيب هذه

التطبيقات للطلبات الواردة من خلال استرداد و/أو تعيين قيم الكائنات المُدارة ومن ثمّ إصدار PDU للاستجابة. يقوم تطبيق مُنشئ الإخطارات بإصدار رسائل غير متزامنة؛ في حالة الوكيل التقليدي تُستخدم إخطارات SNMPv2 أو SNMPv1 التلقائية لهذا التطبيق. يعيد تطبيق ممرِّر المُفوَّض (Proxy Forwarder Application) توجيه الرسائل بين الكيانات.



الشكل 8-7: وكيل SNMP التقليدي.

يتضمن محرك SNMP لوكيل تقليدي كافة المكوِّنات الموجودة في مُحرّك SNMP للمدير التقليدي، بالإضافة إلى نظام فرعي للتحكم في الوصول. يوفر هذا النظام الفرعي خدمات الترخيص للتحكم في الوصول إلى قواعد معلومات الإدارة لقراءة وتعيين قِيم كائنات الإدارة. ويتم تنفيذ هذه الخدمات على أساس محتويات وحدات بيانات البروتوكول (PDUs). وقد يدعم تنفيذ معيَّن للنظام الفرعي للأمن نموذجاً واحداً أو عدة نماذج منفصلة للتحكم في الوصول. وحتى الآن يُعدّ نموذج الأمن الوحيد المُعرَّف هو نموذج التحكم في الوصول المبنيّ على المشهد (VACM) بالنسبة SNMPv3، والمعرَّف في RFC 2575.

لاحظ أنه يتم تنظيم المهام المتصلة بالأمن في نظامين فرعيين منفصلين وهما: نظام الأمن ونظام التحكم في الوصول. ويُعدُّ هذا مثالاً ممتازاً على التصاميم التركيبية الجيدة، لأن كلا النظامين الفرعيين يؤديان وظائف مختلفةً تماماً، ولذا فمن الطبيعي السماح للتوحيد القياسي في هذين المجالين بالمُضي قُدماً بشكلٍ مستقل. ويُعنَى النظام الفرعي للأمن بالخصوصية والتوثيق ويعمل على رسائل SNMP. أما النظام الفرعي للتحكم في الوصول فيهتم بالوصول المُرخّص به إلى معلومات الإدارة ويعمل على وحدات بيانات بروتوكول SNMP.

❖ المصطلحات

يُعرِّف الجدول 8-2 بإيجاز بعض المصطلحات التي تم عرضها في RFC 2571. ويقترن بكل كيان هوية فريدة للمحرك (snmpEngineID). وللتحكم في الوصول يمكن اعتبار أن كل كيان SNMP يقوم بإدارة عددٍ من سياقات المعلومات المُدارة (contexts of managed information)، ولكل سياق اسمٌ فريد (contextName) داخل ذلك الكيان. للتأكيد على أن هناك مديراً واحداً للسياقات داخل كيانٍ ما، يخصّص لكل كيان هوية محرك سياق فريدة (contextEngineID). ونظراً لوجود تناظر "واحد إلى واحد" بين محرك السياق ومحرك SNMP في هذا الكيان، فإن هوية محرك السياق (contextEngineID) تطابق قيمة هوية محرك SNMP (snmpEngineID). ويخضع التحكم في الوصول للسياق المحدّد الذي تمت فيه محاولة الوصول ولهوية المُستخدِم الذي طلب الوصول؛ يتم التعبير عن هذه الهوية الأخيرة بكيان رئيس (principal) والذي قد يكون فرداً أو تطبيقاً أو مجموعةً من الأفراد أو التطبيقات.

الجدول 8-2: مصطلحات SNMPv3.

snmpEngineId هوية فريدة لا لبس فيها لمحرك SNMP، فضلاً عن كيان SNMP الذي يتوافق مع ذلك المحرك. والذي يُعرَّف بجملة نصّية بصيغة سلسلة بايتات.
contextEngineID يُعرِّف على نحو فريد كيان SNMP الذي قد يحقق مثيلاً لسياق له اسم محدّد contextName.
contextName يعرّف سياقاً معيَّناً داخل محرك SNMP. وهذا يتم تمريره كمُعامِل إلى المرسل وإلى النظام الفرعي للتحكم في الوصول.
scopedPDU كتلة من البيانات التي تتألف من (contextEngineID) و(contextName) و وحدة بيانات بروتوكول SNMP (SNMP PDU). يتم تمريرها كمُعامِل من/إلى النظام الفرعي للأمن.
snmpMessageProcessingModel هوية فريدة لنموذج معالجة الرسالة من النظام الفرعي لمعالجة الرسائل. تتضمن القيم المحتملة SNMPv1 و SNMPv2c و SNMPv3. تُعرَّف هذه الهوية بجملة نصّية بصيغة عدد صحيح.
snmpSecurityModel هوية فريدة لنموذج أمني من النظام الفرعي للأمن. تتضمن القيم المحتملة SNMPv1 وSNMPv2c وUSM. وتُعرَّف بجملة نصية بصيغة عدد صحيح.
snmpSecurityLevel مستوى من الأمن الذي يمكن فيه إرسال رسائل SNMP أو يمكن من خلاله معالجة أية عمليات، أو التعبير عنها بمعايير توفير أو عدم توفير التوثيق و/أو الخصوصية. القيم البديلة هي (noAuthnoPriv) و(authNoPriv) و(authPriv). يتم تعريفه بواسطة بجملة نصية بصيغة عدد صحيح.
Principal الرئيس: الكيان الذي تُقدَّم الخدمات أو تتم المعالجة نيابةً عنه. ويمكن أن يكون الرئيس فرداً يعمل في دور محدّد؛ أو مجموعةً من الأفراد مع دور محدّد لكلٍّ منها؛ أو تطبيقاً أو مجموعةً من التطبيقات؛ أو تشكيلات من كلٍّ منها.
securityName سلسلة يمكن للإنسان قراءتها تمثل الرئيس. ويتم تمريرها كمُعامِل في جميع أوَّليَّات SNMP (المُرسِل، معالجة الرسائل، الأمن، التحكم في الوصول).

تتعلق بعض المصطلحات الأخرى ذات الأهمية بمعالجة الرسائل. ويحدِّد المصطلح snmpMessageProcessingModel صيغة الرسالة ورقم إصدار SNMP لمعالجة الرسالة. كما يحدِّد snmpSecurityModel النموذج الأمني الذي سيتم استخدامه. ويحدِّد المصطلح snmpSecurityLevel الخدمات الأمنية المطلوبة لهذه العملية تحديداً. ويمكن للمُستخدِم أن يطلب عملية التوثيق فقط أو التوثيق بالإضافة إلى الخصوصية (التشفير)، أو لا يطلب أيّاً منهما.

❖ تطبيقات SNMPv3

تم تعريف الخدمات بين الوحدات النمطية في كيان SNMP في RFCs في شكل عمليات أوَّليَّة ومُعامِلات. تحدد العمليات الأوَّليَّة الوظيفة التي سيتم تنفيذها، في حين تستخدم المُعامِلات لتمرير البيانات ومعلومات التحكم. ويمكن اعتبار هذه العمليات الأوَّليَّة والمُعامِلات أسلوباً رسمياً لتعريف خدمات SNMP. وتعتمد الصورة الفعلية للعمليات الأوَّليَّة على كيفية التنفيذ؛ مثال على ذلك طلب استدعاء إجراء (procedure call). في المناقشة التالية، سيكون من المفيد الإشارة إلى الشكل 8-8 والمبنيّ على شكل في RFC 2571 لمعرفة كيف تتواءم كل هذه العمليات الأوَّليَّة معاً. ويبيّن الشكل 8-8 (a) تسلسل الأحداث حيث يطلب أيٌّ من تطبيقي مولِّد الأوامر أو مُنشئ الإخطارات إرسال PDU، وما يتبع ذلك من كيفية إعادة الاجابة على هذا الطلب إلى ذلك التطبيق؛ وتقع هذه الأحداث عند نظام مدير. ويبيّن الشكل 8-8 (b) الأحداث المقابلة حين تقع عند نظام وكيل. كما يوضّح الشكل كيف تؤدي رسالة واردة إلى إرسال PDU المرفقة بالرسالة إلى تطبيق ما، وكيف تؤدي استجابة ذلك التطبيق إلى رسالة صادرة. لاحظ أن بعض الأسهم الموجودة في الرسم التخطيطي والتي تم وسمها باسم عملية أوَّليَّة تمثل استدعاء إجراء. وتمثل الأسهم غير الموسومة العودة من الإجراء، كما يشير التظليل إلى المطابقة بين الاستدعاء والعودة.

يُعرِّف RFC 2573، بصورة عامة، الإجراءات المتَّبعة لكل نوع من أنواع التطبيقات عند إنشاء PDUs لإرسالها أو معالجة PDUs واردة. وفي جميع الحالات،

يتم تعريف الإجراءات من حيث التفاعل مع المُرسِل (Dispatcher) بواسطة العمليات الأوَّليَّة للمرسل (Dispatcher primitives).

يَستخدم تطبيق مولِّد الأوامر العمليات الأوَّليَّة للمُرسِل والمسماة sendPdu وResponsePdu. تزوِّد sendPdu المُرسِل بمعلومات حول الوِجهة المقصودة، ومُعامِلات الأمن، ووحدة بيانات البروتوكول المُعَدَّة فعلياً للإرسال. ثم يستدعي المُرسِل نموذج معالجة الرسائل، والذي يستدعي بدوره النموذج الأمني، لإعداد الرسالة. يقوم المُرسِل بتسليم الرسالة المُعَدَّة إلى طبقة النقل (مثلاً UDP) لإرسالها. في حالة فشل إعداد الرسالة، فإن القيمة الأوَّليَّة الراجعة من sendPdu، والمُحدَّدة سلفاً من قِبَل المُرسِل، هي مؤشر خطأ. في حال نجاح عملية إعداد الرسالة، يحدِّد المُرسِل قيمةً للمُعامِل sendPduHandle كهوية لوحدة بيانات البروتوكول تلك ويعيد تلك القيمة إلى مولِّد الأوامر. ويقوم مولِّد الأوامر بتخزين قيمة المُعامِل sendPduHandle بحيث يمكنه مطابقة الطلب الأصلي مع وحدة بيانات البروتوكول بالاستجابة التالية. يقوم المُرسل بتسليم كل PDU واردة إلى تطبيق مولِّد الأوامر الصحيح، باستخدام processResponsePdu.

يَستخدم تطبيق مجيب الأوامر أربعاً من عمليات المُرسل الأوَّليَّة: registerContextEngineID، unregisterContextEngineID، processPdu، returnResponsePdu. كما يستخدم عملية أوَّليَّة للنظام الفرعي للتحكم في الوصول هي isAccessAllowed.

تُمكِّن العملية الأوَّليَّة registerContextEngineID تطبيق مجيب الأوامر من ربط نفسه مع محرك SNMP من أجل معالجة بعض أنواع PDUs لمحرك السياق. وبمجرد تسجيل مجيب الأوامر، يتم إرسال كافة الرسائل التي تم استقبالها بشكلٍ غير متزامن والتي تحتوي على التركيبة المسجَّلة من هوية محرك السياق (contextEngineID) ونوع بيانات البروتوكول المدعوم (pduType) إلى مجيب الأوامر الذي تم تسجيله لدعم هذه التركيبة. ويمكن لمجيب الأوامر فك ارتباطه مع محرك SNMP باستخدام العملية الأوَّليَّة unregisterContextEngineID.

(a) مولد الأوامر أو منشئ الإخطار

(b) مجيب الأوامر

الشكل 8-8: تدفق SNMPv3.

يُسلِّم المُرسِل وحدة بيانات البروتوكول الواردة بكل طلب إلى تطبيق مجيب الأوامر المناسب باستخدام العملية الأوَّليَّة processPdu. ثم يقوم مجيب الأوامر بالخطوات الآتية:

- يفحص مجيب الأوامر محتويات PDU لهذا الطلب. ويجب أن يتطابق نوع العملية مع أحد الأنواع التي تم تسجيلها من قِبَل هذا التطبيق.
- يحدِّد مجيب الأوامر ما إذا كان الوصول مسموحاً لأداء عملية الإدارة المطلوبة من قِبَل تلك الـ PDU. ولهذا الغرض، يتم استدعاء العملية الأوَّليَّة isAccessAllowed. ويُشير مُعامِل securityModel إلى نموذج الأمن الذي سيستخدمه النظام الفرعي للتحكم في الوصول في الرد على هذا الاستدعاء. ويحدِّد النظام الفرعي للتحكم في الوصول ما إذا كان مسؤول الطلب (securityName) في المستوى الأمني (securityLevel) لديه الإذن لطلب عملية الإدارة (viewType) على كائن الإدارة (variabeName) في هذا السياق (contextName).
- إذا تم السماح بالوصول، يُنجز مجيب الأوامر عملية الإدارة ويجهز PDU للاستجابة. وفي حال فشل الوصول، يقوم مجيب الأوامر بإعداد PDU استجابة ملائمة للإشارة إلى ذلك الفشل.
- يستدعي مجيب الأوامر المُرسِل باستخدام العملية الأوَّليَّة returnResponsePdu لإرسال PDU الاستجابة.

يتبع تطبيق مُنشئ الإخطارات نفس الإجراءات العامة التي تستخدم في تطبيق مولِّد الأوامر. في حالة إرسال PDU لطلب إبلاغ، يتم استخدام العمليتين الأوليين sendPdu وprocessResponsePdu بنفس الطريقة كما في تطبيق مولِّد الأوامر. في حالة إرسال PDU لإخطار تلقائي، تُستخدم فقط العملية الأوَّليَّة sendPdu.

يتبع تطبيق مُستقبِل الإخطارات جزءاً محدّداً من الإجراءات العامة المتَّبعة في تطبيق مجيب الأوامر. ويجب أولاً على مُستقبِل الإخطارات التسجيل ليستقبل PDUs الخاصة بالإبلاغ و/أو الإخطار التلقائي. كلا النوعين من وحدات بيانات

البروتوكول يتم استقبالهما بواسطة العملية الأوَّليَّة processPdu. وبالنسبة لوحدة بيانات البروتوكول للإبلاغ، يتم الرد باستخدام العملية الأوَّليَّة returnResponsePdu.

يستفيد تطبيق مُمرِّر المُفوَّض من العمليات الأوَّليَّة لدى المُرسِل لإعادة توجيه رسائل SNMP. ويعالج مُمرِّر المُفوَّض أربعة أنواع أساسية من الرسائل:

- الرسائل التي تحتوي على أنواع وحدات بيانات البروتوكول من تطبيق مولِّد الأوامر. يحدِّد تطبيق مُمرِّر المُفوَّض المحرك المستهدف ليكون إما محرك SNMP المستهدف أو محرك SNMP الأقرب أو الذي يقع في نفس الاتجاه، ومن ثَمَّ يُرسِل وحدة بيانات البروتوكول المناسبة للطلب.
- الرسائل التي تحتوي على أنواع وحدات بيانات البروتوكول من تطبيق مُنشئ الإخطارات. ويحدِّد تطبيق مُمرِّر المُفوَّض محركات SNMP التي ينبغي أن تستلم الإخطار ثم يُرسِل وحدة أو وحدات بيانات البروتوكول المناسبة للإخطار.
- الرسائل التي تحتوي على وحدات بيانات البروتوكول من نوع استجابة. يحدِّد مُمرِّر المُفوَّض الطلب أو الإخطار الذي سبق توجيهه الذي يناظر هذه الاستجابة – إن وُجد – ثم يقوم بإرسال وحدة بيانات البروتوكول الملائمة كاستجابة.
- الرسائل التي تتضمن إشارة تقرير. تمثل وحدات بيانات البروتوكول للتقرير اتصالات SNMPv3 من محرك إلى محرك. ويحدِّد مُمرِّر المُفوَّض الطلب أو الإخطار الذي سبق توجيهه – إن وُجد – والذي يناظر إشارة هذا التقرير ويعيد توجيه إشارة التقرير مرة أخرى إلى بادئ الطلب أو الإخطار.

8-3-2 معالجة الرسالة ونموذج أمن المُستخدِم

تتضمن معالجة الرسائل نموذجاً عاماً لمعالجة الرسائل بالإضافة إلى نموذج محدَّد للأمن؛ ويوضّح الشكل 8-8 هذه العلاقة.

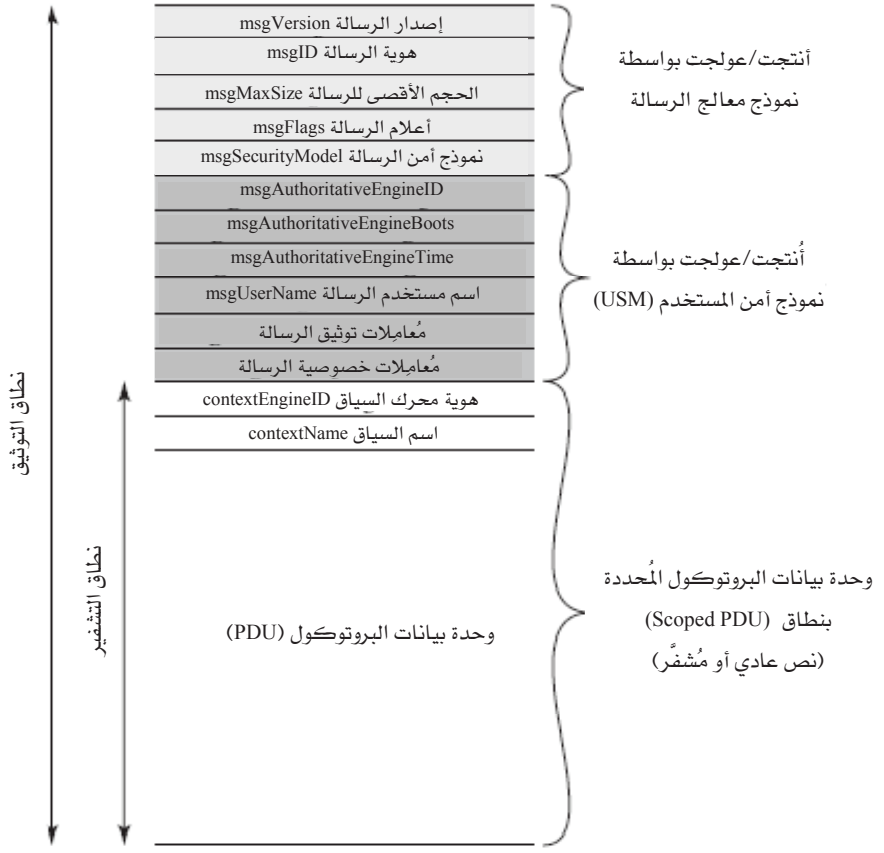
❖ نموذج معالجة الرسائل

يُعرِّف RFC 2572 نموذجاً عاماً لمعالجة الرسائل. وهذا النموذج مسؤول عن قبول وحدات بيانات البروتوكول من المُرسِل، وتغليفها في رسائل، والاستعانة بنموذج أمن المُستخدِم (USM) لإدراج المُعامِلات (البارامترات) الأمنية في ترويسة الرسالة. كما يقبل نموذج معالجة الرسالة أيضاً الرسائل الواردة، ويستعين بنموذج أمن المُستخدِم USM لمعالجة المُعامِلات الأمنية في ترويسة الرسالة، ثم يُسلِّم وحدة بيانات البروتوكول المغلَّفة إلى المرسل.

يوضّح الشكل 8-9 هيكل الرسالة. وتنتج الحقول الخمسة الأولى عن نموذج معالجة الرسالة للرسائل الصادرة ويتم معالجتها بواسطة نموذج معالجة الرسالة للرسائل الواردة. وتعرض الحقول الستة التالية مُعامِلات الأمن المُستخدَمة من قِبَل نموذج أمن المُستخدِم (USM). وأخيراً، تُشكل وحدة بيانات البروتوكول مع هوية محرك السياق (contextEngineID) واسم السياق (contextName) ما يمكن أن يسمى وحدة بيانات البروتوكول المُحدَّدة بنطاق (scoped PDU)، وتستخدم في معالجة وحدة بيانات البروتوكول.

الحقول الخمسة الأولى هي:

- رقم إصدار الرسالة (msgVersion): يأخذ القيمة 3 (أي SNMPv3).
- هوية الرسالة (msgID): مُعرِّف فريد يُستخدَم بين كيانَي SNMP لتنسيق رسائل الطلب والاستجابة، وبواسطة معالج الرسالة لتنسيق معالجة الرسالة بواسطة نماذج نظام فرعي مختلف داخل البنية. نطاق هذا المُعرِّف هو 0 إلى $(2^{31}-1)$.
- الحد الأقصى لحجم الرسالة (msgMaxSize): الحد الأقصى لحجم الرسائل المُعتمَد من قِبَل مُرسِل الرسالة بالبايتات (octets)، وتتراوح قيمته من 484 إلى $(2^{31}-1)$. هذا هو الحد الأقصى لحجم الجزء الذي يمكن للمرسل قبوله من محرك SNMP آخر (سواء كان رداً أو نوع رسالة آخر).



الشكل 8-9: صيغة رسالة SNMPv3 مع نموذج أمن المُستخدِم (USM).

- أعلام الرسالة (msgFlags): سلسلة من ثمانية بتات تحتوي على ثلاثة أعلام في البتات الثلاثة الأقل وزناً: reportableFlag، وprivFlag، وauthFlag. إذا كان العلم reportableFlag =1، فيجب إعادة PDU من نوع تقرير إلى المُرسِل في الحالات التي يمكن أن تتسبب في إنتاج PDU للتقرير؛ وعندما تكون قيمة العلم صفراً، يمكن ألا يتم إرسال PDU للتقرير. يتم وضع reportableFlag عند القيمة 1 بواسطة المُرسِل في كافة الرسائل التي

تحتوي على طلب (Get، Set) أو على طلب إبلاغ، بينما يوضع عند القيمة 0 للرسائل التي تحتوي على PDU للاستجابة أو للإخطار التلقائي، أو للتقرير. ويُعدّ reportableFlag بمثابة عامل ثانوي لتحديد متى يتم إرسال تقرير. ويُستخدم فقط في الحالات التي لا يمكن ترجمة الجزء الخاص بوحدة بيانات البروتوكول لهذه الرسالة (مثلاً عند فشل إزالة التشفير بسبب مفتاح غير صحيح). يتم تحديد قيم privFlag وauthFlag بواسطة المُرسِل ليشير إلى مستوى الأمن الذي تم تطبيقه على الرسالة. فحالة privFlag =1، تعني أنه قد تم تطبيق التشفير، أما حالة privFlag = 0، فتعني أنه قد تم تطبيق التوثيق. ويُسمح بجميع القيم لهذين العلمين باستثناء (privFlag = 1، وauthFlag = 0)؛ أي أن التشفير بدون التوثيق غير مسموح به.

- نموذج أمن الرسالة (msgSecurityModel): هو مُعرِّف تتراوح قيمته من 0 إلى $(2^{31}-1)$ ويشير إلى نموذج الأمن الذي استخدمه المُرسِل لإعداد تلك الرسالة ومن ثَمّ نموذج الأمن الذي يجب أن يستخدمه المُستقبِل لمعالجة تلك الرسالة. القيم المحجوزة تشمل 1 لـ SNMPv1، و2 لـ SNMPv2c (SNMPv2 مع تجمُّع SNMPv1)، و3 لـ SNMPv3.

❖ نموذج أمن المُستخدِم (USM)

يعرِّف RFC 2574 نموذج أمن المُستخدِم (USM)، وهو يوفِّر خدمات التوثيق والخصوصية لـ SNMP. وعلى وجه التحديد، تم تصميم USM للتأمين ضد التهديدات الرئيسة الآتية:

- تعديل المعلومات: يمكن أن يغيِّر كيانٌ ما رسالة عابرة تم إنشاؤها بواسطة كيان مرخّص له، على نحو يتسبب في عمليات إدارة غير مرخّصة، بما في ذلك تغيير قيم الكائنات. ويتمثل جوهر هذا التهديد في إمكانية تغيير كيان غير مرخّص له أيّ مُعامِل من مُعامِلات الإدارة، بما في ذلك المُعامِلات ذات الصلة بالتهيئة والإجراءات والمحاسبة.

- انتحال هوية كيان آخر (masquerade): يمكن أن يقوم كيانٌ ما بمحاولة إجراء عمليات إدارة غير مرخّصة له بواسطة انتحال هوية كيان آخر مرخّص له.
- تعديل تدفق الرسائل: تم تصميم بروتوكول SNMP لكي يعمل على بروتوكول نقل لاتوصيلي. حيث يوجد خطر من إمكانية تأخير رسائل SNMP أو إعادة ترتيبها أو إعادة إرسالها لإجراء عمليات إدارة غير مرخّصة. فعلى سبيل المثال، يمكن أن يتم نسخ رسالة بدء تشغيل (reboot) جهاز ما ثم تكرارها فى وقتٍ لاحقٍ.
- إفشاء البيانات: يمكن لكيانٍ ما مراقبة التبادلات بين مدير ووكيل ومن ثَمَّ معرفة قيم الكائنات المُدارة ومعرفة الأحداث المُبلَّغ عنها. على سبيل المثال، قد يتمكن مهاجمٌ ما عن طريق مراقبة الأمر Set الذي يقوم بتغيير كلمات السر من معرفة كلمات السر الجديدة.

لا يهدف نموذج أمن المُستخدِم (USM) إلى التأمين ضد التهديدين الآتيين:

- حجب الخدمة: قد يمنع مهاجمٌ ما التبادلات بين مدير ووكيل.
- تحليل حركة المرور: قد يراقب مهاجمٌ ما النمط العام لحركة المرور بين المديرين والوكلاء.

يمكن تبرير عدم وجود إجراء مضاد لحالات حجب الخدمة بالسببين الآتيين:

1. يتعذر تمييز هجمات حجب الخدمة في كثير من الأحيان عن حالات فشل الشبكة والتي تواجهها جميع برامج إدارة الشبكة وتتعامل معها كأمر اعتيادي.
2. من المُرجح أن يعطل هجوم حجب الخدمة جميع أنواع التبادلات وهذا أمر يخص أمن الشبكة ككل، وليس خاصاً ببروتوكول إدارة الشبكة.

أما بالنسبة لتحليل حركة المرور، فإن كثيراً من أنماط حركة مرور إدارة الشبكة يمكن التنبؤ بها (فمثلاً، قد تُدار كيانات معيّنة بواسطة أوامر SNMP

الصادرة بانتظام من محطة إدارة واحدة أو من عدد من محطات الإدارة) ولذلك لا توجد ميزة كبيرة للحماية ضد مراقبة هذه الأنماط من حركة المرور.

❖ وظائف التشفير

تم تعريف وظيفتي تشفير لنموذج أمن المُستخدِم (USM) هما: التوثيق والتشفير. ولدعم هاتين الوظيفتين، يحتاج محرك SNMP إلى قيمتين: مفتاح خاص (privKey) ومفتاح توثيق (authKey). يتم الاحتفاظ بقيمتين منفصلتين لهذين المفتاحين للمُستخدِمين الآتيين:

- المستخدمون المحليون: وهم الكيانات الرئيسة (principals) بمحرك SNMP هذا والمرخّص لهم بعمليات الإدارة.
- المستخدمون البعيدون: وهم الكيانات الرئيسة (principals) بمحرك SNMP البعيد والمطلوب الاتصال بها.

تُعدُّ هذه القيم خصائص للمُستخدِم ويتم تخزينها لكل مُستخدِم ذي صلة. ولا يمكن الحصول بواسطة SNMP على قيم المفتاح الخاص (privKey) أو مفتاح التوثيق (authKey).

يسمح USM باستخدام أحد بروتوكولي التوثيق البديلة: HMAC-MD5-96 وHMAC-SHA-96. يستخدم HMAC – والذي تم شرحه في الفصل الثالث – دالة تحوير آمنة ومفتاحاً سرياً لإنتاج شفرة توثيق لرسالةٍ ما. بالنسبة لـ HMACMD5-96 يتم استخدام HMAC مع MD5 كدالة التحوير الأساسية. يتم استخدام مفتاح للتوثيق (authKey) مكوَّن من 128 بتاً (16 بايتاً) . كمُدخَل إلى خوارزمية HMAC. تنتج الخوارزمية مُخْرَجاً طوله 128 بتاً يتم بتره إلى 96 بتاً. وبالنسبة لـ -HMAC-SHA) (96، فإن دالة التحوير الأساسية هي SHA-1، وطول مفتاح التوثيق authKey 20 بايتاً، وتنتج الخوارزمية مُخْرَجاً طوله 20 بايتاً يتم بتره أيضاً إلى 12 بايتاً.

وللتشفير يستخدم USM نمط سلسلة كتل الشفرة (CBC) لخوارزمية DES. ويتم توفير مفتاح خاص (privKey) طوله 16 بايتاً كمُدخَل إلى بروتوكول التشفير.

وتُستخدم الـ 64 بتاً الأولى من قيمة privKey كمفتاح DES. ولأن معيار تشفير البيانات DES يتطلب مفتاحاً بطول 56 بتاً يتم تجاهل البت الأقل وزناً من كل بايت. ويتطلب نمط سلسلة كتل الشفرة مُتجَه تهيئة (IV) طوله 64 بتاً. تستخدم الـ 64 بتاً الأخيرة من المفتاح الخاص (privKey) لتوليد قيمة هذا المتَّجَه.

❖ المحركات السيادية وغير السيادية

في أي عملية انتقال رسالة بين كيانين، يتم تعيين أحدهما (المرسل أو المستقبل) كمحرك SNMP سيادي، وفقاً للقواعد الآتية:

- عندما تحتوي رسالة SNMP على حمولةٍ تتوقع رداً ما (على سبيل المثال: Get، أو GetNext،أو GetBulk، أو Set، أو PDU Inform)، يُعدُّ مُستقبِل مثل هذه الرسائل سيادياً.
- عندما تحتوي رسالة SNMP على حمولةٍ لا تتوقع رداً (على سبيل المثال وحدة بيانات البروتوكول لإخطار SNMPv2 تلقائي أو استجابة أو تقرير)، يُعدُّ مُرسِل مثل هذه الرسالة سيادياً.

وهكذا، بالنسبة للرسائل المُرسَلة نيابةً عن مولِّد للأوامر أو لرسائل الإبلاغ من مُنشئ الإخطارات، فإن المُستقبِل يكون سيادياً. وبالنسبة للرسائل المُرسَلة نيابةً عن مجيب الأوامر أو رسائل الإخطارات التلقائية من مُنشئ الإخطارات، فإن المُرسِل يكون سيادياً. يخدم هذا التعيين هدفين هما:

- يتم تحديد التوقيت للرسالة بواسطة ساعة يرعاها المحرك السيادي. فعندما يرسل محرك سيادي رسالة (إخطار تلقائي، استجابة، تقرير)، فإنها تحتوي على القيمة الحالية للساعة الخاصة به، وهكذا يمكن للمُستقبِل غير السيادي أن يتزامن مع تلك الساعة. وعندما يقوم محرك غير سيادي بإرسال رسالة (Get، GetNext، GetBulk، Set، Inform)، فإنها تتضمن تقديراته الحالية لقيمة الوقت عند الوجهة، مما يسمح للوجهة بتقييم ملاءمة توقيت الرسالة.

- تُمكِّن عملية توطين المفاتيح (key localization) – والتي سيرد وصفها لاحقاً – أحد الكيانات الرئيسة من امتلاك المفاتيح المخزنة في عدة محركات؛ يتم توطين هذه المفاتيح بالمحرك السيادي بحيث يكون الكيان الرئيس مسؤولاً عن مفتاح واحد مع تجنب المخاطر الأمنية لتخزين نسخ متعددة من نفس المفتاح في شبكة موزَّعة.

من الطبيعي أن يتم تعيين المُستقبل لمولِّد الأوامر ووحدات بيانات البروتوكول للإبلاغ كمحرك سيادي، ومن ثَمَّ مسؤول عن التحقق من توقيت الرسالة. فإذا حدث تأخير استجابة أو إخطار تلقائي أو تم إعادة إرسالهما يكون الضرر الناتج ضئيلاً. بيد أن وحدات بيانات البروتوكول لمُولِّد الأوامر – وإلى حدٍّ ما وحدات بيانات البروتوكول للإبلاغ – تؤدي إلى عمليات إدارة، مثل قراءة أو تغيير قيم كائنات MIB. لذا فإنه من المهم ضمان أن مثل تلك الوحدات لا يتم تأخيرها أو إعادة إرسالها، الأمر الذي قد يؤدي إلى تأثيراتٍ غير مرغوب فيها.

❖ مُعامِلات (بارامترات) رسائل USM

عندما يتم تمرير رسالة صادرة إلى USM بواسطة معالج الرسائل، يقوم USM بتعبئة المُعامِلات المتصلة بالأمن في ترويسة الرسالة. وعندما يتم تمرير رسالة واردة إلى USM بواسطة معالج الرسائل، يقوم USM بمعالجة القيم الواردة في تلك الحقول. والمُعامِلات المتصلة بالأمن هي كما يأتي:

- msgAuthoritativeEngineID: هو مُعرِّف محرك SNMP لمحرك SNMP السيادي المشارك في تبادل هذه الرسالة. وعلى هذا، فإن هذه القيمة تشير إلى المصدر لرسالة Trap أو Response أو Report، وإلى الوجهة لرسالة Get أو GetNext أو GetBulk أو Set أو Inform.
- msgAuthoritativeEngineBoots: هو قيمة المُعامِل snmpEngineBoots لمحرك SNMP السيادي المشارك في تبادل هذه الرسالة. أما قيمة الكائن snmpEngineBoots فهي عدد صحيح يقع بين 0 و(2^{31}-1) ويمثل عدد المرات

التي قام فيها محرك SNMP هذا بالتهيئة أو إعادة التهيئة منذ تهيئته أول مرة.

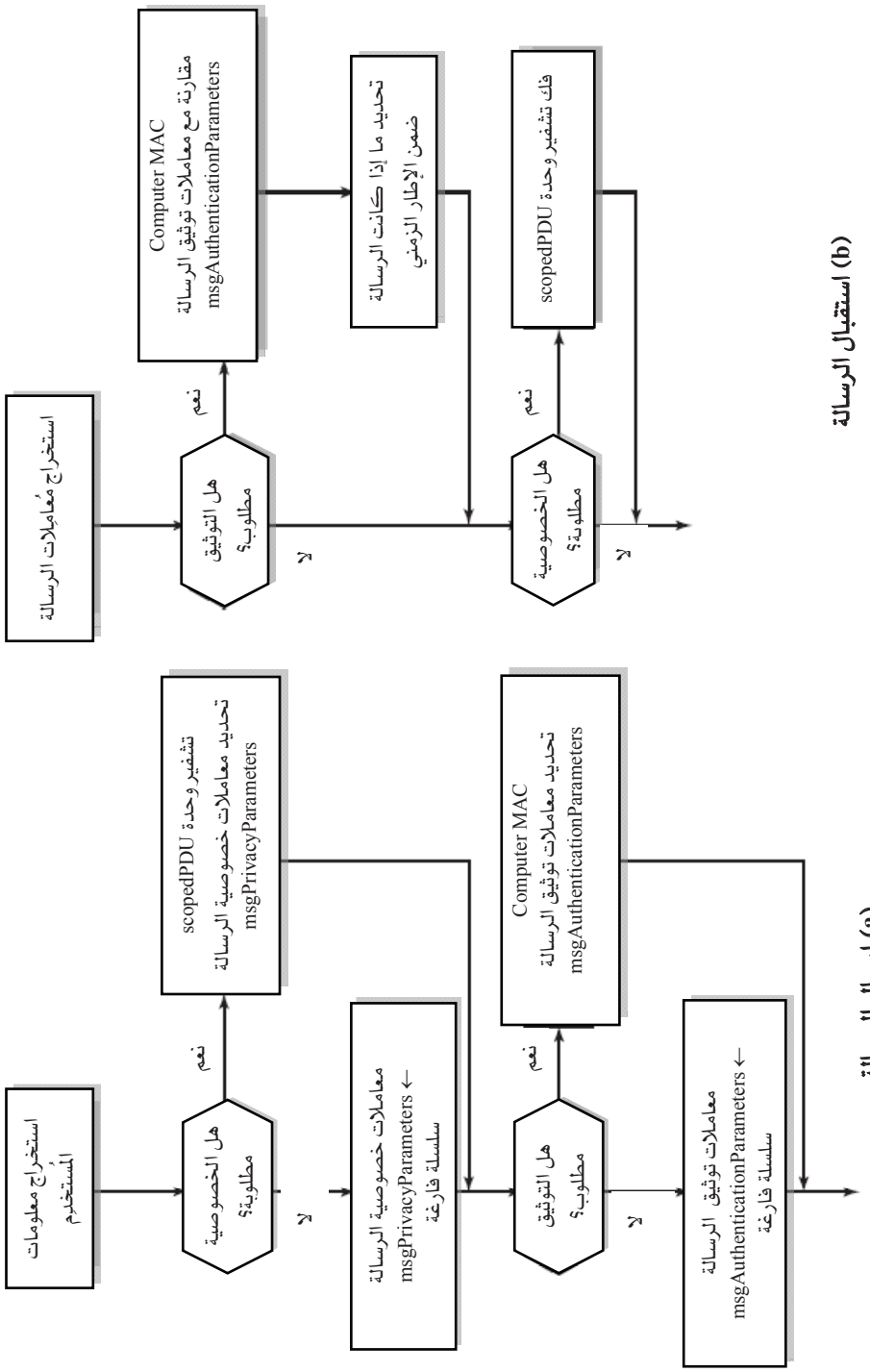
- msgAuthoritativeEngineTime: قيمة المُعامِل snmpEngineTime لمحرك SNMP السيادي المشارك في تبادل هذه الرسالة. أما قيمة الكائن snmpEngineTime فهي عدد صحيح يقع بين 0 و(2^{31}-1) ويمثل عدد الثواني منذ آخر إضافة قام بها محرك SNMP السيادي لقيمة هذا الكائن. يُعدُّ كل محرك SNMP سيادي مسؤولاً عن زيادة قيمة وقت محرك SNMP الخاص به (snmpEngineTime) مرة واحدة كل ثانية. يُعدُّ المحرك غير السيادي مسؤولاً عن زيادة القيمة التقديرية لمعامل snmpEngineTime الخاص بكل محرك سيادي بعيد يتواصل معه.
- msgUserName: هو اسم المُستخدِم (الكيان الرئيس) الذي يتم تبادل الرسالة نيابةً عنه.
- msgAuthenticationParameters: تكون قيمته لاشيء (null) في حالة عدم استخدام التوثيق لهذا التبادل. وإلا، يُعدُّ هذا مُعامِل توثيق. بالنسبة للتعريف الحالي لنموذج أمن المُستخدِم (USM)، مُعامِل التوثيق هو شفرة HMAC لتوثيق الرسالة.
- msgPrivacyParameters: تكون قيمته لاشيء (null) في حالة عدم كون هذا التبادل خصوصياً. وإلا، يُعدُّ هذا مُعامِل خصوصية. بالنسبة للتعريف الحالي لنموذج أمن المُستخدِم (USM)، تُستخدم قيمة مُعامِل الخصوصية لتحديد القيمة الأوَّليَّة (IV) في خوارزمية تسلسل كتل الشفرة (CBC) للمعيار القياسي لتشفير البيانات (DES).

ويلخص الشكل 8-10 طريقة عمل نموذج أمن المُستخدِم (USM). بالنسبة لنقل الرسالة، يتم إجراء التشفير أولاً، إذا لزم الأمر. يتم تشفير وحدة بيانات البروتوكول المُحدَّدة بنطاق ووضعها في حمولة الرسالة، وتعيين قيمة مُعامِل خصوصية الرسالة (msgPrivacyParameters) إلى القيمة المطلوبة لتوليد القيمة الأوَّليَّة IV، ثم يتم تنفيذ التوثيق إذا دعت الحاجة. يتم استخدام الرسالة بأكملها بما في ذلك وحدة بيانات

البروتوكول المُحدَّدة بنطاق كمُدخَل لـ HMAC، ثم توضع شفرة التوثيق الناتجة في مُعامِل توثيق الرسالة (msgAuthenticationParameters). بالنسبة للرسائل الواردة يتم تنفيذ التوثيق أولاً إذا دعت الحاجة. يفحص USM أولاً شفرة التوثيق الواردة مقارنةً مع شفرة توثيق يقوم بحسابها؛ فإذا تطابقت القيمتان، يفرض أن الرسالة موثَّقة (أي آتية من المصدر المفترض ولم تتغير أثناء الإرسال). ثم يقوم USM بالتحقق من كون الرسالة وصلت في مدى زمني صحيح، كما سنوضّح لاحقاً. إذا لم يكن توقيت الرسالة مناسباً، يتم تجاهلها على اعتبار أنها غير موثَّقة. وأخيراً، إذا كان قد تم تشفير وحدة بيانات البروتوكول، يقوم USM بإزالة التشفير واستعادة النص الأصلي.

❖ آليات التوقيت في USM

يتضمن USM مجموعة من آليات التوقيت للوقاية ضد تأخير أو إعادة إرسال الرسالة. ويجب على كل محرك SNMP يمكنه العمل في أي وقت كمحرك سيادي أن يحتفظ بالكائنين snmpEngineBoots وsnmpEngineTime اللذين يقومان بتعريف وقته المحلي. عند أول تثبيت لمحرك SNMP، يتم تصفير قيمة هذين الكائنين. بعدها تتم زيادة قيمة snmpEngineTime مرة واحدة كل ثانية. إذا وصلت قيمة snmpEngineTime في أي وقت إلى الحد الأقصى (2^{31}-1)، يتم زيادة قيمة snmpEngineBoots كما لو أن النظام قد أعيد تشغيله، ويعاد تصفير قيمة snmpEngineTime وزيادة قيمته كل ثانية مرة أخرى. باستخدام آلية تزامن، يقوم كل محرك غير سيادي بالاحتفاظ بتقدير لقيم الوقت لكل محرك سيادي يتواصل معه. وتوضع هذه القيم المُقدَّرة في كل رسالة صادرة لتُمكِّن المحرك السيادي المُستقبِل للرسالة من تحديد ما إذا كان توقيت الرسالة الواردة مناسباً أو لا.



الشكل 8-10: معالجة رسالة USM.

تعمل آلية التزامن بالطريقة الآتية. يحتفظ محرك غير سيادي بنسخة محلية من متغيرات ثلاث خاصة بكل محرك SNMP سيادي معروف لهذا المحرك:

- snmpEngineBoots: أحدث قيمة للمُعامِل snmpEngineBoots الخاص بالمحرك السيادي البعيد.
- snmpEngineTime: قيمة تقديرية من هذا المحرك للمُعامِل snmpEngineTime الخاص بالمحرك السيادي البعيد. وتتم مزامنة هذه القيمة مع المحرك السيادي البعيد بواسطة عملية التزامن التي سنوضحها فيما بعد. تتم زيادة هذه القيمة بين أحداث المزامنة بشكلٍ منطقي مرة كل ثانية للحفاظ على تزامن غير دقيق مع المحرك السيادي البعيد.
- latestReceivedEngineTime: أعلى قيمة للمُعامِل msgAuthoritativeEngineTime التي تلقاها هذا المحرك من المحرك السيادي البعيد؛ ويتم تحديث هذه القيمة كلما وردت قيمة أكبر لها. والغرض من هذا المتغير هو الحماية ضد هجوم إعادة الإرسال الذي قد يمنع تقدم قيمة snmpEngineTime التقديرية لمحرك SNMP غير السيادي.

يتم الحفاظ على مجموعة واحدة من هذه المتغيرات الثلاث لكل محرك سيادي بعيد معروف لهذا المحرك. وتتم المحافظة بشكلٍ منطقي على القيم في نوع من ذاكرة التخزين المخبأة (Cache) والمعنونة بواسطة مُعرِّف محرك SNMP الفريد لكل محرك سيادي بعيد (snmpEngineID).

ولتمكين المحركات غير السيادية من الحفاظ على تزامن الوقت، يُدرِج كل محرك سيادي القيم الحالية للتشغيل (boot) والوقت الخاصة به، وكذلك قيمة مُعرِّف محرك SNMP (snmpEngineID)، بكل رسالة استجابة أو تقرير أو إخطار صادرة في الحقول msgAuthoritativeEngineBoots، وmsgAuthoritativeEngineTime، وmsgAuthoritativeEngineID. إذا كانت الرسالة موثَّقة وكانت ضمن الإطار الزمني المضبوط، يقوم المحرك غير السيادي المُستلِم بتحديث المتغيرات المحلية

snmpEngineBoots، وsnmpEngineTime، وlatestReceivedEngineTime لذلك المحرك البعيد وفقاً للقواعد الآتية:

1. يتم التحديث إذا تحقق أحد الشرطين الآتيين على الأقل:
   - msgAuthoritativeEngineBoots > snmpEngineBoots
     أي عدد مرات التشغيل لمحرك سيادي > عدد مرات التشغيل لمحرك SNMP
   - msgAuthoritativeEngineBoots = snmpEngineBoots AND msgAuthoritativeEngineTime > latestReceivedEngineTime
     أي عدد مرات التشغيل لمحرك سيادي = عدد مرات التشغيل لمحرك SNMP، ووقت محرك سيادي > آخر وقت محرك تم استقباله

ينص الشرط الأول على وجوب التحديث إذا زادت قيمة مُعامِل التشغيل (msgAuthoritativeEngineBoots) من قِبَل المحرك السيادي منذ آخر تحديث. وينص الشرط الثاني على أنه إذا لم تتم زيادة قيمة معامل التشغيل، فيجب التحديث إذا كانت قيمة وقت المحرك الواردة أكبر من آخر قيمة لوقت المحرك تم تلقيها. وتكون قيمة الوقت الواردة للمحرك أقل من آخر قيمة وقت للمحرك تم تلقيها في حالة وصول رسالتين بترتيب عكسي؛ وهذا أمر وارد الحدوث. كما يمكن حدوث ذلك أيضاً في حالة هجوم إعادة الإرسال. في كلتا الحالتين، لن يقوم المحرك المتلقي بالتحديث.

2. إذا وجب التحديث، يتم إجراء التغييرات الآتية:
   - msgAuthoritativeEngineBoots → snmpEngineBoots
   - msgAuthoritativeEngineTime → snmpEngineTime
   - msgAuthoritativeEngineTime → latestReceivedEngineTime

إذا قلبنا المنطق، فسوف نرى أنه في حالة msgAuthoritativeEngineBoots < snmpEngineBoots، لا يتم التحديث. مثل هذه الرسالة تُعدُّ غير موثَّقة ويجب تجاهلها. وفي حالة msgAuthoritativeEngineBoot = snmpEngineBoots، ولكن في

نفس الوقت latestReceivedEngineTime > msgAuthoritativeEngineTime، لا يتم التحديث أيضاً. في هذه الحالة، قد تكون الرسالة موثّقة إلا أنها قد تكون بترتيب خاطئ، وفي هذه الحالة يكون تحديث snmpEngineTime غير مبرَّر.

لاحظ أن توظيف التزامن يتم فقط إذا كانت خدمة التوثيق مُستخدَمة في هذه الرسالة وأنه قد تم توثيق الرسالة باستخدام HMAC. هذا التقييد ضروري لأن نطاق التوثيق يشمل msgAuthoritativeEngineID، وmsgAuthoritativeEngineBoots، وmsgAuthoritativeEngineTime، ومن ثَمَّ يمكن ضمان كون هذه القيم صالحة.

ينص SNMPv3 على أنه يجب استلام الرسالة ضمن إطار زمني مقبول لتجنب التأخير وهجمات إعادة الإرسال. ويجب اختيار الإطار الزمني صغيراً بقدر الإمكان بالنظر إلى دقة الساعات المستعملة والتأخير المصاحب للاتصال ذهاباً وإياباً ومعدل مزامنة الساعات. وفي حال تعيين إطار زمني صغير جداً، سوف يتم رفض الرسائل الموثَّقة على اعتبار أنها غير موثَّقة،ومن ناحية أخرى فإن تعيين إطار زمني كبير يجعل الرسائل أكثر عرضة للتأخيرات الناتجة عن هجمات خبيثة.

سندرس الحالة الأكثر أهمية المتعلقة بمُستقبِل سيادي (يختلف اختبار التوقيت قليلاً في حالة المُستقبِل غير السيادي). مع كل رسالة واردة موثَّقة وقيمة مُعامِلها msgAuthoritativeEngineID مساوية لقيمة snmpEngineID لهذا المحرك، يقوم المحرك بمقارنة قيمة المُعامِلين msgAuthoritativeEngineBoots وmsgAuthoritativeEngineTime بالرسالة الواردة مع قيمة المُعامِلين snmpEngineBoots وsnmpEngineTime اللتين يحتفظ بهما هذا المحرك. تعتبر الرسالة الواردة خارج الإطار الزمني إذا تحقق أي من الشروط الآتية:

- snmpEngineBoots = 2^{31} - 1
- snmpEngineBoots ≠ msgAuthoritativeEngineBoots
- قيمة msgAuthoritativeEngineTime تختلف عن قيمة snmpEngineTime بأكثر من ± 150 ثانية

ينص الشرط الأول على أنه إذا تم تخزين snmpEngineBoots عند قيمته القصوى، فلا يمكن اعتبار أي رسالة واردة رسالةً موثقة. وينص الشرط الثاني على أنه يجب أن يكون للرسالة وقت تشغيل مُساوياً لوقت تشغيل المحرك المحلي؛ على سبيل المثال، في حالة إعادة تشغيل المحرك المحلي وعدم قيام المحرك البعيد بالمزامنة مع المحرك المحلي منذ إعادة التشغيل، تُعدُّ الرسائل من ذلك المحرك البعيد غير موثَّقة. وينص الشرط الأخير على أن الوقت المُتضمَّن في الرسالة الواردة يجب أن يكون أكبر من الوقت المحلي ناقص 150 ثانية وأقل من الوقت المحلى زائد 150 ثانية.

في حالة اعتبار رسالةٍ ما على أنها خارج الإطار الزمني، تُعدُّ الرسالة غير موثَّقة، ويتم إرجاع إشارة خطأ خارج إطار الوقت (notInTimeWindow) إلى الوحدة المتصلة (calling module).

ومرة أخرى، كما هو الحال مع التزامن، فإن التحقق من مناسبة الوقت يتم فقط في حالة استخدام خدمة التوثيق وعندما تكون الرسالة موثَّقة، مما يضمن صلاحية حقول ترويسة الرسالة.

❖ توطين المفاتيح (Key Localization)

المطلب الرئيس لاستخدام خدمات التوثيق والخصوصية في SNMPv3 هو أنه لا بد لأي اتصال بين كيان رئيس (principal) على محرك غير سيادي ومحرك سيادي بعيد من وجود مفتاحين سريين مُشتركين لعمليتي التوثيق والخصوصية، يُمكِّن هذان المفتاحان مُستخدِماً ما بمحرك غير سيادي (في العادة نظام إدارة) من استخدام التوثيق والخصوصية مع الأنظمة السيادية البعيدة التي يديرها المُستخدِم (في العادة أنظمة الوكلاء). يوفِّر المرجع RFC 2574 مبادئ توجيهية لإنشاء تلك المفاتيح وتحديثها وإدارتها.

لتبسيط عبء إدارة المفاتيح على الكيانات الرئيسة (principals)، يُطلب فقط من كل كيان رئيس الاحتفاظ بمفتاح توثيق واحد ومفتاح تشفير واحد. ولا يتم

تخزين هذه المفاتيح في MIB ولا يمكن الوصول إليها عبر SNMP. وفيما يلي سنتناول أولاً أسلوب توليد هذه المفاتيح من كلمة السر. ثم نستعرض مفهوم توطين المفاتيح، والذي يُمكِّن الكيان الرئيس من مشاركة مفتاح توثيق وتشفير فريد مع كل محرك بعيد مع الاحتفاظ فقط بمفتاح توثيق وتشفير واحد محلياً.

يتطلب المُستخدِم مفتاحاً خاصاً مُكوَّناً من 16 بايتاً ومفتاح توثيق مُكوَّناً من 16 أو 20 بايتاً. بالنسبة للمفاتيح التي يملكها مُستخدِمون من البشر، من المرغوب فيه أن يكون المُستخدِم قادراً على توظيف كلمة سر مقروءة بشرياً بدلاً من مفتاح مُكوَّن من سلسلة من البتات. ويُعرِّف المرجع RFC 2574 خوارزمية لاستخلاص مفتاح بطول 16 أو 20 بايتاً من كلمة السر للمُستخدِم. ولا يضع (USM) قيوداً على كلمة السر نفسها، ولكن ينبغي لسياسات الإدارة المحلية أن تفرض على المُستخدِمين استعمال كلمات سر لا يمكن تخمينها بسهولة.

يتم استخلاص المفتاح من كلمة السر كما يأتي:

- اتخاذ كلمة السر الخاصة بالمُستخدِم كمُدخل وتوليد سلسلة بطول 2^{20} بايتاً (1,048,576 بايتاً) بتكرار قيمة كلمة السر عدة مرات حسب الضرورة، وبتر القيمة الأخيرة إذا دعت الضرورة، لتشكيل سلسلة الحروف digest0. على سبيل المثال، إذا كانت كلمة السر مكونة من 8 أحرف (2^3 بايتات) فسوف تُسلسل مع نفسها 2^{17} مرة لتشكيل digest0.
- في حال الرغبة في استعمال مفتاح مكوَّن من 16 بايتاً، تُستخدم دالة التحوير MD5 لتشكيل digest1 من digest0. أما في حال مفتاح مكوَّن من 20 بايتاً، تُستخدم دالة التحوير SHA-1 لتشكيل digest1 من digest0. ويكون الناتج هنا هو مفتاح المُستخدِم.

من مزايا هذا الأسلوب أنه يبطئ إلى حدٍّ كبير هجوم القاموس الاستقصائي (brute-force dictionary attack)، والذي يقوم فيه الخَصْم بتجربة كلمات سر مختلفة كثيرة محتملة لتوليد المفتاح من كل واحدة من تلك الكلمات، ومن ثَمَّ اختبار ما إذا كان المفتاح الناتج يعمل مع بيانات التوثيق أو التشفير المتاحة له. فعلى

سبيل المثال، لو أن مهاجماً ما اعترض رسالة موثَّقة، يمكنه أن يحاول توليد قيمة HMAC مع مختلف مفاتيح المُستخدِم الممكنة. وفي حالة حدوث مطابقة، يمكن للمهاجم افتراض أنه تم اكتشاف كلمة السر. فالعملية المكونة من خطوتين والتي تم ذكرها تؤدي إلى زيادة كبيرة في الوقت الذي قد يستغرقه مثل هذا الهجوم.

ميزة أخرى لهذا الأسلوب هو أنه يعزل مفاتيح المُستخدِم عن نوعية نظام إدارة الشبكة (NMS). فلا يحتاج نظام إدارة الشبكة إلى تخزين قيم مفاتيح المُستخدِم. بدلاً من ذلك، يتم توليد مفتاح المُستخدِم من كلمة السر عند الحاجة. ويسرد [BLUM97b] الاعتبارات التالية التي تحفّز استخدام طريقة كلمة السر بشكلٍ مستقل عن نظام إدارة الشبكة:

- إذا أُريد للمفتاح أن يُخزّن بدلاً من أن يتم توليده من كلمة السر، فإن أحد البدائل هو الاحتفاظ بمستودع مركزي للمفاتيح السرية. ولكن هذا يؤثر سلباً على الاعتمادية الكلية للنظام كما يمكن أن يجعل عملية اكتشاف الأخطاء وإصلاحها أمراً مستحيلاً إذا كان المستودع نفسه لا يمكن الوصول إليه عند الحاجة.
- ومن جهة أخرى، إذا تم الحفاظ على عدة نسخ من المستودع، فإن هذا يهدد الأمن الشامل بتوفير مواطن اقتحام أكثر للمهاجمين المحتملين.
- في حالة استخدام مستودع مركزي أو عِدة نسخ منه، يجب المحافظة عليه في مواقع آمنة. قد يقلل ذلك من فرصة إقامة "موقع متقدم" أثناء التصدي للهجوم (أي، التدخل للإصلاح عندما تتعطل قطاعات غير متوقعة من الشبكة أو عندما لا يمكن الوصول إليها لفترات من الزمن لا يمكن التنبؤ بها).

ويمكن استخدام كلمة سر واحدة لتوليد مفتاح واحد يُستخدم لكلٍّ من عمليتي التشفير والتوثيق. الأسلوب الأكثر أماناً هو استخدام كلمتي سر، إحداهما لإنشاء مفتاح توثيق والأخرى لإنشاء مفتاح تشفير مختلف.

تم تعريف المفتاح المحلي في المرجع RFC 2574 على أنه مفتاح سري مشترك بين مُستخدِم ومحرك SNMP سيادي واحد. والهدف هو احتفاظ المُستخدِم بمفتاح واحد فقط (أو مفتاحين اثنين عند الحاجة لكلٍّ من التوثيق والخصوصية) ومن ثَمَّ يحتاج فقط لتذكُّر كلمة سر واحدة (أو اثنتين). تختلف المعلومات السرية الفعلية المشتركة بين مُستخدِم معين وكل محرك من محركات SNMP السيادية. ويُشار إلى العملية التي يتم بواسطتها تحويل مفتاح مُستخدِم واحد إلى عدة مفاتيح فريدة (واحد لكل محرك SNMP بعيد)، بعملية توطين المفتاح. يستعرض [BLUM97a] الدافع وراء هذه الاستراتيجية، والتي سنلخصها هنا.

يمكننا تحديد الأهداف الآتية لعملية إدارة المفاتيح:

- لكل نظام وكيل SNMP في شبكة موزعة مفتاحٌ فريدٌ خاصٌ بكل مُستخدِم مرخّص له بإدارة هذا الوكيل. إذا تم الترخيص لعدة مُستخدِمين كمديرين، يكون لدى الوكيل مفتاح توثيق فريد ومفتاح تشفير فريد لكل مُستخدِم. وهكذا، إذا تم اختراق مفتاح أحد المُستخدِمين، فلا يؤثر ذلك على مفاتيح المُستخدِمين الآخرين.
- تختلف مفاتيح مُستخدِم واحد بالنسبة للوكلاء المختلفين. وهكذا، إذا تعرض وكيل ما للاختراق، فإن مفاتيح المُستخدِم لهذا الوكيل فقط هي التي تتعرض للخطر بمعزلٍ عن مفاتيحه الأخرى مع الوكلاء الآخرين.
- يمكن إدارة الشبكة من أي نقطة فيها، بغض النظر عن توافر نظام إدارة الشبكة مسبق التهيئة (NMS). ويسمح ذلك للمُستخدِم بأداء وظائف إدارية من أي محطة إدارة. ويتم توفير هذه الإمكانية بواسطة خوارزمية "كلمة السر إلى مفتاح" المشروحة سابقاً.

يمكننا أيضاً تعريف الأمور الآتية كأشياء ينبغي تجنبها:

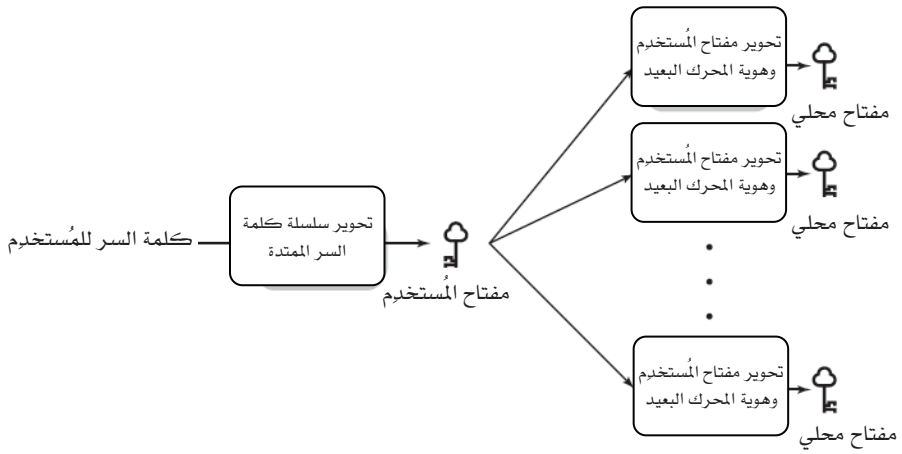
- على المُستخدِم تذكُّر (أو بمعنى آخر إدارة) عدد كبير من المفاتيح، وهو عدد يتنامى مع إضافة وكلاء جدد لإدارتهم.

- الخَصْم الذي يكتشف مفتاحاً لوكيل واحد يعد قادراً حينئذ على انتحال هوية أي وكيل آخر بالنسبة لأي مُستخدِم، أو أي مُستخدِم بالنسبة لأي وكيل آخر.

استجابةً للأهداف والاعتبارات السابقة، يتم باستخدام دالة أحادية الاتجاه غير قابلة للعكس (أي دالة تحوير آمنة) لتحويل مفتاح مُستخدِم واحد إلى عدة مفاتيح محلية مختلفة لمحركات موثَّقة مختلفة (وكلاء مختلفين).

ويتم هذا الإجراء كما يأتي:

- تُشكَّل السلسلة digest2 عن طريق سَلسَلة digest1 (تم وصفه سابقاً) مع قيمة snmpEngineID للمحرك السيادي مع digest1.
- إذا كان الهدف مفتاحاً مكوناً من 16 بايتاً، تُستخدم دالة التحوير MD5 مع digest2. وإذا كان المفتاح المرغوب بطول 20 بايتاً، تُستخدم دالة التحوير SHA-1 مع digest2. ويكون المُخرَج هو المفتاح المحلي للمُستخدِم.



الشكل 8-11: توطين المفاتيح.

يمكن بعدها تهيئة المفتاح المحلي الناتج على نظام الوكيل بطريقةٍ آمنة. ونظراً للطبيعة أحادية الاتجاه لدالتي MD5 وSHA-1، فإنه من غير المعقول (شبه المستحيل) لخَصْمٍ ما اكتشاف مفتاح المُستخدِم حتى لو نجح هذا الخَصْم في اكتشاف مفتاح محلي. يلخص الشكل 8-11 عملية توطين المفاتيح.

8-3-3 التحكم في الوصول المبنيّ على المشهد

التحكم في الوصول هو وظيفة أمنية تُؤدَّى على مستوى وحدة بيانات البروتوكول (PDU). تُعَرِّف وثيقة التحكم في الوصول آلياتٍ لتحديد ما إذا كان ينبغي السماح لكيان رئيس (principal) بعيد بالوصول لكائن مُدار في قاعدة معلومات إدارة (MIB) محلية. ويمكن تصور وتحديد آليات متعددة للتحكم في الوصول. تُعرِّف وثائق SNMPv3 نموذجاً للتحكم في الوصول المُعتمِد على المشهد (VACM). يستفيد VACM نفسه من MIB والتي تُعرِّف سياسة التحكم في الوصول لهذا الوكيل وتجعل من الممكن استخدام التهيئة عن بُعد.

يتسم نموذج VACM بخاصتين مهمتين:

- يحدِّد VACM ما إذا كان ينبغي السماح لكيان رئيس (principal) بعيد بالوصول إلى كائن مُدار في MIB محلية.
- يستفيد VACM من قاعدة معلومات إدارة MIB والتي:
  - تُعرِّف سياسة التحكم في الوصول إلى ذلك الوكيل.
  - تجعل التهيئة عن بُعد ممكنة الاستخدام.

❖ عناصر نموذج VACM

يُعَرِّف المرجع RFC 2575 خمسة عناصر تؤلِّف النموذج VACM: المجموعات (groups)، مستوى الأمن، السياقات (contexts)، ومشاهد MIB، وسياسة الوصول.

تم تعريف "المجموعة" كسلّة مكونة من صفر أو أكثر من الثنائي <securityModel, securityName> (أي <نموذج أمن، واسم أمن>) التي يمكن نيابةً

عنها الوصول إلى كائنات إدارة SNMP. يشير اسم الأمن securityName إلى كيان رئيس (principal) ، مع العلم أن جميع الرؤساء في مجموعة معيّنة لهم نفس حقوق الوصول. ويخصّص لكل مجموعة اسم فريد groupName. يُستخدم مفهوم المجموعة كأداة مفيدة لتصنيف المديرين فيما يتعلق بحقوق الوصول. فعلى سبيل المثال، قد يكون لجميع مديري المستوى الأعلى سلة واحدة من حقوق الوصول، بينما قد يكون لمديري المستوى المتوسط سلة مختلفة من حقوق الوصول.

يمكن لأي تشكيلة معيّنة من قيم securityModel وsecurityName أن تنتمي إلى مجموعة واحدة على الأكثر. أي أنه لهذا الوكيل ، لا يمكن أن يدرج أي كيان رئيس (principal) محدَّد له اتصالات محميَّة بنموذج أمن محدَّد (securityModel) إلا في مجموعة واحدة فقط.

تختلف حقوق الوصول لمجموعةٍ ما استناداً إلى مستوى الأمن للرسالة التي تحتوي على الطلب. على سبيل المثال، يمكن أن يسمح وكيلٌ ما بحق القراءة فقط لطلب مُرسَل في رسالة غير موثَّقة ولكن للسماح لها بحق الكتابة قد تحتاج إلى توثيق أولاً. وعلاوة على ذلك، في حالة بعض الكائنات الحساسة، قد يطلب الوكيل أن يتم إرسال الطلب والاستجابة له باستخدام خدمة الخصوصية.

يُعرَّف "سياق MIB" بمجموعة فرعية لبعض مثيلات (instances) الكائنات في الـ MIB المحلية باسم معين. توفر السياقات طريقة مفيدة لتجميع الكائنات في مجموعات لها سياسات مختلفة للوصول.

يتصل مفهوم السياق بعملية التحكم في الوصول. عندما تتفاعل محطة إدارة مع وكيل للوصول إلى معلومات الإدارة لدى الوكيل ، حينئذ يتم التفاعل بين الكيان الرئيس (principal) للإدارة ومحرك SNMP للوكيل، ويتم التعبير عن امتيازات التحكم في الوصول بمشهد MIB ينطبق على هذا الكيان الرئيس وذاك السياق. وتتمتع السياقات بالخصائص الرئيسة الآتية:

- قد يحتوي كيان SNMP – المُعرَّف بشكلٍ فريد بواسطة هوية محرك السياق contextEngineID – على أكثر من سياق واحد.

- قد يظهر الكائن أو مثيله في أكثر من سياق واحد.
- لتعريف مثيل كائن – في حال وجود سياقات متعددة – يجب تعريف اسم السياق contextName، وهوية محرك السياق contextEngineID الخاص به، بالإضافة إلى نوع الكائن ومثيله.

كثيراً ما نود تقييد وصول مجموعة معيّنة إلى كائنات بمجموعة فرعية في الوكيل. ولتحقيق هذا الهدف، يكون الوصول إلى سياق بواسطة مشهد MIB (VACM)، والذي يُعرِّف مجموعة معيّنة من الكائنات المُدارة (واختيارياً مثيلات كائنات محدَّدة). يستخدم VACM تقنية قوية ومرنة لتعريف مشاهد MIB، استناداً إلى مفاهيم مشهد الأشجار الفرعية ومشهد العائلات. ويُعرَّف مشهد MIB عن طريق مجموعة أو عائلة من الأشجار الفرعية، حيث يمكن إدراج أيٍّ من تلك الأشجار الفرعية أو استبعادها من المشهد.

يتم تنظيم الكائنات المُدارة في قاعدة بيانات محلية في تسلسل هرمي، أو شجري، استناداً إلى مُعرّفات الكائنات. تضم هذه القاعدة المحلية للبيانات مجموعة فرعية من جميع أنواع الكائنات المُعرَّفة وفقاً لمعيار الإنترنت الخاص ببنية معلومات الإدارة (Structure of Management Information (SMI)) ويتضمن مثيلات للكائنات التي تتفق مُعرّفاتها مع اصطلاحات SMI.

يتضمن SNMPv3 مفهوم الشجرة الفرعية. الشجرة الفرعية ببساطة هي عقدة في التسلسل الهرمي للتسمية والخاص بقاعدة معلومات الإدارة MIB بالإضافة إلى جميع العناصر المندرجة تحتها. ويمكن تعريف شجرة فرعية بشكلٍ اصطلاحي (رسمي) باعتبارها المجموعة التي تضم جميع الكائنات ومثيلات الكائنات التي لديها بادئة ASN.1 مشتركة في أسمائها. تُعدُّ البادئة المشتركة الأطول لجميع المثيلات في الشجرة الفرعية هي هوية الكائن (object identifier) للعقدة الأصل بتلك الشجرة الفرعية.

ترتبط ثلاثة مشاهد MIB مع كل مُدخَل في جدول وصول VACM (vacmAccessTable)، مشهد لكل وصول قراءة وكتابة وإخطار. ويتألَّف كل

مشهد MIB من مجموعة من مشاهد الشجرات الفرعية. ويتم تحديد كل مشهد شجرة فرعية في مشهد MIB من حيث كونه مضمَّناً أو مستبعداً. أي أن مشهد MIB إما يتضمن أو يستبعد كل مثيلات الكائنات الواردة في تلك الشجرة الفرعية. وبالإضافة لذلك، يتم تعريف قناع للمشهد لتقليل كمية معلومات التهيئة عندما يتطلب الأمر تحكماً دقيقاً وتفصيلياً للوصول (مثل التحكم في الوصول على مستوى مثيل الكائن).

يُمكِّن VACM من تهيئة محرك SNMP لفرض مجموعة معيّنة من حقوق الوصول، والتي تُشكِّل سياسة وصول. يعتمد تحديد الوصول على العوامل الآتية:

- يقدم الكيان الرئيس طلباً للوصول. يجعل VACM من الممكن للوكيل أن يخصّص امتيازات وصول مختلفة لمُستخدِمين مختلفين. فعلى سبيل المثال، قد تُمنح سلطات واسعة لتغيير عناصر في قاعدة معلومات الإدارة MIB المحلية لنظام مدير مسؤول عن تهيئة شبكة بأكملها، بينما قد يكون لأحد مديري المستويات الوسطى مسؤولية مراقبة الشبكة مع حق الوصول للقراءة فقط، وعلاوة على ذلك قد يكون الوصول مقصوراً على مجموعة فرعية فقط من MIB المحلية. وكما تمت مناقشته سابقاً، يتم تعيين الكيانات الرئيسة للمجموعات ويتم تحديد سياسة الوصول للمجموعة.
- مستوى الأمان الذي أُرسِل به الطلب في رسالة SNMP. حيث يطلب الوكيل عادةً استخدام التوثيق للرسائل التي تحتوي على طلب تعيين قيم (أي عملية كتابة).
- نموذج الأمن المُستخدَم في معالجة رسالة الطلب. إذا ما تم تنفيذ عدة نماذج للأمن في وكيلٍ ما، يمكن تهيئة الوكيل ليوفر مستويات مختلفة من الوصول للطلبات المُقدَّمة من رسائل تمت معالجتها بواسطة نماذج أمن مختلفة. فعلى سبيل المثال، قد تكون بعض العناصر قابلة للوصول إليها إذا كان طلب الرسالة قد أتى عبر نموذج أمن المُستخدِم (USM)، ولكن غير قابلة للوصول إليها إذا كان نموذج الأمن SNMPv1.
- سياق MIB للطلب.

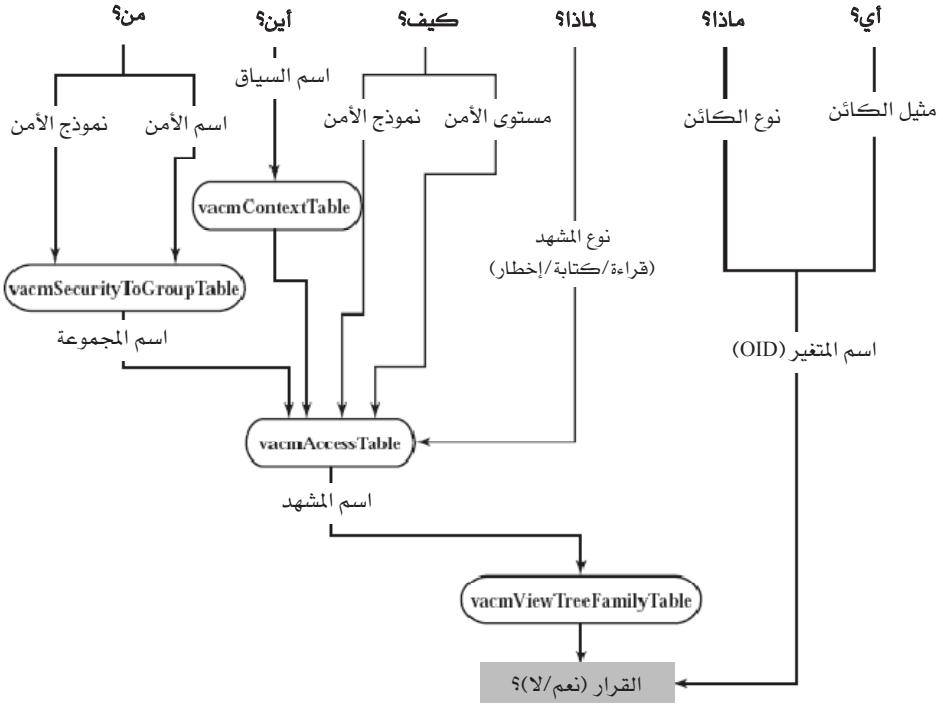
- مثيل الكائن المحدَّد المطلوب الوصول إليه. تمتلك بعض الكائنات معلومات أكثر خطورة أو حساسية مقارنةً بغيرها، ولذلك يجب أن تعتمد سياسة الوصول على المثيل المحدَّد المطلوب.
- **نوع** الوصول المطلوب (قراءة، كتابة، إخطار). فالقراءة والكتابة والإخطار عمليات إدارة مختلفة، ويمكن تطبيق سياسات مختلفة للتحكم في الوصول لكلٍّ منها.

❖ معالجة التحكم في الوصول

يقوم تطبيق SNMP باستدعاء نموذج التحكم في الوصول المبنيّ على المشهد (VACM) عن طريق isAccessAllowed، مع المُدخلات نموذج الأمن (securityModel) واسم الأمن (securityName) ومستوى الأمن (securityLevel) ونوع المشهد (viewType) واسم السياق (contextName) واسم المتغير (variableName). ونحتاج لجميع قيم هذه المُعامِلات لاتخاذ قرار التحكم في الوصول. أي أن النظام الفرعي للتحكم في الوصول قد تم تعريفه كأداة عالية المرونة لتهيئة التحكم في الوصول لدى الوكيل ، عن طريق تقسيم مكوِّنات قرار التحكم في الوصول إلى ستة متغيرات منفصلة.

يعرض الشكل 8-12 – المقتبس من شكل في مقالة طلب تعليقات RFC2575 – طريقة مفيدة للنظر إلى متغيرات الإدخال ويبيّن كيف أن الجداول المتنوعة في قاعدة معلومات الإدارة في نموذج VACM تؤثر في اتخاذ قرار التحكم في الوصول.

- مَنْ (who): وهو عبارة عن مزيج المُعامِلين securityModel وsecurityName، والذي يُعرِّف "مَنْ" لهذه العملية، كما يحدِّد هوية الكيان الرئيس الذي تكون اتصالاته محميّة بنموذج أمن (securityModel) معين. وينتمي هذه المزيج إلى مجموعة واحدة على الأكثر في محرك SNMP. أما الجدول vacmSecurityToGroupTable فيُعرِّف اسم المجموعة (groupName) لكل مزيجٍ من المُعامِلين securityModel وsecurityName.



الشكل 8-12: منطق نموذج التحكم في الوصول المبنيّ على المشهد (VACM).

- أين (where): يحدِّد اسم السياق (contextName) "أين" يمكن العثور على كائن الإدارة المطلوب. ويحتوي جدول سياق VACM (vacmContextTable) على قائمة بأسماء السياقات (contextNames) التي يتم التعرف عليها.
- كيف (how): وهو عبارة عن مزيج من المُعامِلين securityModel وsecurityLevel والذي يُعرِّف كيف تمت حماية الطلب الوارد أو PDU للإخطار. ويحدِّد المزيج الثلاثي "مَنْ"، و"أين"، و"كيف" صفاً واحداً على الأكثر يقابل هذا المزيج في جدول وصول VACM (vacmAccessTable).
- لماذا (why): يحدِّد مُعامل نوع المشهد (viewType) لماذا تم طلب الوصول لعملية قراءة أو كتابة أو إخطار. يحتوي الصف المُختار في جدول وصول VACM (vacmAccessTable) على اسم مشهد واحد (viewName) لقاعدة

معلومات الإدارة لكل نوع من هذه الأنواع الثلاثة من العملية، ويستخدم مُعامِل نوع المشهد (viewType) لاختيار اسم مشهد (viewName) محدَّد، والذي يختار بدوره مشهد MIB المناسب من جدول مشهد شجرة عائلة VACM (vacmViewTreeFamilyTable).

- ماذا (what): يُستخدم المُعامِل variableName كمُعرِّف كائن والذي تحدد بادئته نوعية هذا الكائن كما تحدد لاحقته مثيلاً معيناً. وتُشير نوعية الكائن إلى ماذا تكون نوعية معلومات الإدارة المطلوبة.
- ماهية (which): يشير مثيل الكائن إلى ماهية العنصر المحدَّد المطلوب من المعلومات.

وأخيراً، تتم مقارنة اسم المتغير (variableName) مع مشهد MIB الذي تم التوصل إليه. وإذا تطابق اسم المتغير مع عنصر ضمن مشهد MIB، يتم منح الوصول.

❖ الدوافع وراء مفاهيم VACM

يبدو أن المفاهيم التي تؤلِّف VACM تؤدِّي إلى تعريف معقد نوعاً ما للتحكم في الوصول. أما دوافع تقديم هذه المفاهيم فهي توضيح العلاقات المشاركة في معلومات إدارة الوصول والتقليل إلى أدنى حدّ من متطلبات التخزين والمعالجة لدى الوكيل. ولفهم تلك الدوافع، لاحظ أنه في SNMPv1 يُستخدَم مفهوم التجمُّع لتمثيل معلومات الأمن الآتية:

- هوية الكيان الطالب (محطة الإدارة).
- هوية الكيان المُنفِّذ (وكيل يمثل نفسه أو ينوب عن كيان مُفوِّض).
- هوية موقع معلومات الإدارة المراد الوصول إليها (وكيل أو كيان مُفوِّض).
- معلومات التوثيق.
- معلومات التحكم في الوصول (الترخيص بإجراء العملية المطلوبة).
- معلومات مشهد MIB.

وينتج عن دمج كل تلك المفاهيم في متغير واحد فقدان المرونة والفاعلية. يوفّر VACM نفس المعلومات الأمنية باستخدام متغيرات مميزة لكل عنصر. ويُعدُّ ذلك تحسناً جوهرياً مقارنةً بـ SNMPv1 . حيث يؤدي فصل هذه المفاهيم المتعددة إلى إمكانية تعيين كل متغير بشكلٍ مستقل.

4-8 توصيات للمطالعة

يتضمن المرجع [STAL99] دراسة شاملة ومفصَّلة عن SNMP وSNMPv2 وSNMPv3؛ كما يقدم نظرة عامة على تقنية إدارة الشبكة.

[STAL99] Stallings, W. *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2*. Reading, MA: Addison-Wesley, 1999.

5-8 مصادر للمعلومات على الويب

- موقع ويب SNMPv3: تحتفظ به "جامعة براونشفايغ التقنية". وهو يوفر وصلات إلى المراجع ومسودات الإنترنت ونسخاً من التوضيحات والتغييرات المقترحة التي أُرسِلت بواسطة الفريق العامل ووصلات تشعبية إلى البائعين مع تطبيقات SNMPv3.
- موقع The Simple Web: تحتفظ به جامعة تفينتي. وهو مصدر جيد للمعلومات المتعلقة بـ SNMP، بما في ذلك المؤشرات إلى كثير من تطبيقات النطاق المفتوح وقوائم من الكتب والمقالات.

6-8 مصطلحات رئيسة

Access Policy	سياسة الوصول، سياسة النفاذ
Agent	وكيل
Community	التجمُّع
Community Name	اسم التجمُّع
Key Localization	توطين المفاتيح
Management Information Base (MIB)	قاعدة معلومات الإدارة
Management Station	محطة الإدارة
Message Processing Model	نموذج معالجة الرسالة
Network Management Proxy	مُفوَّض إدارة الشبكة
Simple Network Management Protocol (SNMP)	بروتوكول إدارة الشبكة البسيط
User Security Model	نموذج أمن المُستخدِم
View-based Access Control Model (VACM)	نموذج التحكم في الوصول المبنيّ على المشهد

8-7 أسئلة للمراجعة ومسائل

8-7-1 أسئلة المراجعة

8-1 ما الزوايا التي يمكن من جانبها اعتبار البنية الهيكلية لإدارة الشبكة متكاملة؟

8-2 ما العناصر الرئيسة في نموذج SNMP؟

8-3 ما المقصود بـ MIB؟

8-4 ما القدرات الأساسية أو الأوامر التي يتم توفيرها في SNMPv1؟

8-5 ما وظيفة مُفوَّض SNMP؟

8-6 وضّح باختصار مفهوم التجمُّع في SNMPv1.

8-7 ما العلاقة بين SNMPv1 وSNMPv2 وSNMPv3؟

8-8 ما التهديدات التي تم تصميم USM لمواجهتها؟

8-9 ما الفرق بين محرك سيادي (authoritative) ومحرك غير سيادي (nonauthoritative)؟

8-10 ما المقصود بتوطين المفاتيح؟

8-11 اسرد وعرّف باختصار العناصر التي تؤلّف VACM.

8-7-2 مسائل

8-1 يعرّف SNMPv1 نوع بيانات يُشار إليه كمقياس (gauge) ويتضمن الشرح الآتي دلالات هذا النوع:

> يُمثّل هذا النوع عدداً صحيحاً غير سالب، والذي قد يزيد أو ينقص،وتغلق (latches) قيمة المقياس عند القيمة القصوى، وقد حدد هذا المعيار القيمة القصوى للمقاييس بـ $2^{32} - 1 = 4294967295$.

وللأسف، لم يتم تعريف المقصود بالكلمة latches في المعيار، وأدى ذلك إلى تفسيرين مختلفين. أزال معيار SNMPv2 الالتباس بالتعريف الآتي:

> تتخذ قيمة المقياس الحد الأقصى عندما تكون المعلومات التي يجري نمذجتها أكبر من أو تساوي تلك القيمة القصوى؛ أما إذا انخفضت المعلومات التي يجري نمذجتها فيما بعد إلى أقل من قيمة الحد الأقصى، ينقص المقياس أيضاً.

a. ما التفسير البديل لذلك؟

b. ناقش إيجابيات وسلبيات كلا التفسيرين.

2-8 في SNMPv1، أي كائن يتم تعريفه في MIB يكون له فئة وصول يمكن تعيين قيمتها لأيٍّ من القيم الآتية: "القراءة فقط"، أو "القراءة والكتابة"، أو "الكتابة فقط"، أو "ممنوع الوصول". ويتم إنجاز القراءة عن طريق Get أو عن طريق Trap، ويتم إنجاز الكتابة عن طريق Set. وللكتابة فقط، قد يكون الكائن متاحاً لعمليتي Get وTrap، ولكن طريقة العمل هنا تعتمد على كيفية التنفيذ العملي. وتحدد فئة وصول MIB الحد الأقصى للوصول الذي يمكن أن يُسمح به لأحد الكائنات، ولكن في تسهيلات تجمُّع SNMPv1، يمكن لأسلوب الوصول أن يقصر هذا الوصول على تجمعات لها خصائص معيّنة. وفي الجدول الآتي قم بملء كل مُدخَل موضحاً الوصول المسموح به.

فئة وصول MIB	نمط وصول SNMP	
	قراءة فقط	قراءة وكتابة
قراءة فقط		
قراءة وكتابة		
كتابة فقط		
ممنوع الوصول		

3-8 a. يوضّح RFC 2574 أنه بالنسبة لمحرك غير سيادي (nonauthoritative)، يتم تعيين قيم msgAuthoritativeEngineBoots وmsgAuthoritativeEngineTime في ترويسة رسالة صادرة فقط إذا كانت الرسالة يتم توثيقها بواسطة المُستقبِل السيادي. لماذا يعدّ هذا التقييد ذا معنى؟

b. ومع ذلك، بالنسبة لرسالة استجابة من محرك سيادي، يتم تعيين قيم msgAuthoritativeEngineBoots وmsgAuthoritativeEngineTime دائماً في ترويسة الرسالة الصادرة. اشرح أسباب ذلك.

4-8 يحدِّد RFC 2574 أن مزامنة الساعة (تحديث الساعة المحلية بناءً على القيم الواردة) يحدث قبل التحقق من الإطار الزمني (التحقق من أن الرسالة الواردة في التوقيت المناسب). وهذا يعني أنه قد يقوم محرك غير سيادي بتحديث القيمة التقديرية لساعة

المحرك السيادي في حال كانت الرسالة المُستلَمة موثَّقة حتى إذا لم تكن الرسالة ضمن الإطار الزمني. كان هناك جدل مستمر حول ذلك في قائمة مراسلات SNMPv3 منذ صدور طلب التعليقات، ولكن يبدو أن الصياغة في المعيار لم تتغير حتى كتابة هذه السطور. ومن المفيد أن ننظر في الآثار المترتبة على ذلك. بافتراض التعريفات الآتية:

MAEB = msgAuthoritativeEngineBoots
MAET = msgAuthoritativeEngineTime
LRET = latestReceivedEngineTime
SET = local notion of snmpEngineTime for the remote authoritative engine
SEB = local notion of snmpEngineBoots for the remote authoritative engine

ثم لنفترض محركاً غير سيادي يستقبل رسالة بحيث

(MAEB = SEB) AND [LRET < MAET < (SET-150)]

ثم تتوافر الشروط لحصول تحديث الساعة، ومن ثَمَّ يتم التحديث:

SET := MAET; LRET := MAET

الآن، عندما نصل إلى اختبار الإطار الزمني، لدينا

(MAEB = SEB) AND (MAET = SET)

لذلك نعلن أن الرسالة في التوقيت المناسب. ومع ذلك، افترض أنه كان علينا أولاً القيام باختبار إطار الوقت. أكان علينا إعلان أن الرسالة في التوقيت المناسب أو ليست في التوقيت المناسب؟

5-8 في النسخة الأصلية المنشورة لمواصفات نموذج أمن المُستخدِم (USM) رقم RFC 2274، والتي تناقش مزامنة الساعة وإجراءات التحقق من الإطار الزمني، ورد التعليق الآتي: "لاحظ أن هذا الإجراء لا يسمح بالمزامنة التلقائية للوقت إذا كان محرك SNMP غير السيادي لديه حالة حقيقية من عدم التزامن أدت إلى أن يتأخر محرك SNMP سيادي أكثر من 150 ثانية عن محرك SNMP غير السيادي." وقد تم إسقاط هذا البيان من النسخة المعدلة رقم RFC 2574 بعد أن أوضح مؤلف هذا الكتاب لمجموعة العمل أن البيان ليس صحيحاً دائماً، وذكر المثال الذي جاء في المسألة 8-4 للتدليل على ذلك.

6-8 يفترض SNMPv3 أن هناك بعض الوسائل الآمنة لتسليم المفاتيح المحلية إلى أنظمة (وكيل) موثَّقة. تُعدُّ عملية التسليم الآمن هذه خارج نطاق SNMPv3؛ حيث يمكن أن يكون التسليم إما يدوياً أو بواسطة بروتوكول آمن آخر. وبمجرد أن يتم تسليم مفتاح أوَّليّ (أو زوج من المفاتيح للتوثيق والخصوصية) إلى وكيل ما، فإن SNMPv3 يوفر آلية لتحديث المفاتيح بشكلٍ آمن. ومن المرغوب فيه أن يتم تغيير المفاتيح من وقت لآخر لتعزيز

الأمن. يمكن لأحد المُستخدِمين الشروع في عملية تغيير المفتاح، عن طريق طلب كلمة سر جديدة وإدخالها. وبدلاً من ذلك، يمكن لنظام إدارة شبكة (NMS) بدء العملية بواسطة طلب كلمة سر جديدة. وفي كلتا الحالتين، يتم تحديث مفتاح المُستخدِم في NMS. حينئذ يمكن لنظام إدارة شبكة (NMS) اعتماد مفتاح محلي لكل وكيل اتصال.

يجب أن يتصل NMS بعد ذلك بشكلٍ آمن بكل وكيل ليحثه على تحديث مفتاحه المحلي. ومن الواضح أن NMS لا يمكنه ببساطة إرسال المفتاح في نص عادي عبر الشبكة. وهنا يوجد خياران:

- تشفير المفتاح الجديد باستخدام المفتاح القديم باعتباره مفتاح التشفير.
- استخدام نوع من الدوال أحادية الاتجاه لتوليد قيمة جديدة للمفتاح من قيمة المفتاح القديم. إجراء عملية أو-الحصرية (XOR) لهذه القيمة مع قيمة المفتاح الجديد وإرسال النتيجة إلى الوكيل. يستطيع الوكيل حينئذ إجراء عملية أو-الحصرية (XOR) لهذه النتيجة الواردة مع المفتاح القديم لإنتاج المفتاح الجديد.

يستخدم SNMPv3 نسخة معدلة من الطريقة الثانية. فما هي ميزة هذا النهج على الأول؟

7-8 ينطوي نهج SNMPv3 على استخدام كائن تغيير المفتاح (KeyChange) في نظام MIB المستهدف. ويقوم الكيان الرئيس البعيد أو نظام إدارة الشبكة (NMS) بضبط هذا الكائن، والذي يستخدم تلقائياً من قِبَل الوكيل لتحديث المفتاح المناظر. وتتم الخوارزمية على مرحلتين، تعمل واحدة منها في المحرك الطالب والأخرى في محرك الوكيل البعيد. تبدأ العملية عندما يرغب الطالب في تحديث مفتاح موجود (keyOld) إلى قيمة جديدة (keyNew). عندها يقوم الطالب بتنفيذ الخطوات الآتية:

1. توليد قيمة عشوائية إما من مولِّد أرقام شبه عشوائي أو مولِّد أرقام عشوائي حقيقي.
2. حساب

$$digest = Hash (keyOld \parallel random)$$

حيث دالة التحوير (Hash) هي إما MD5 أو SHA-1، بالاعتماد على ما إذا كان المطلوب مفتاح من 16 بايتاً أو 20 بايتاً، والرمز || يمثل الوصل.

3. حساب

$$delta = digest \oplus keyNew$$

حيث ⊕ هي عملية أو - الحصرية (XOR).

4. إرسال قيمة المعامل protocolKeyChange إلى الوكيل في أمر تعيين (Set) لتحديث مثيل كائن (KeyChange) في وكيل MIB.

ما الذي يجب أن يفعله الوكيل مع القيمة الواردة لتحديث المفتاح؟

8-8 الإجراء الأبسط من ذلك المبيَّن في المسألة السابقة يكون بإجراء عملية أو - الحصرية (XOR) بين قيمة المفتاح القديم (keyOld) وقيمته الجديدة (keyNew) وإرسال تلك القيمة. ثم يُجري المُستلِم عملية أو - الحصرية (XOR) بين القيمة التي تم تلقيها و(keyOld) لإنتاج المفتاح الجديد (keyNew). ولأن المهاجم لا يعرف (keyOld)، فإنه لا يستطيع استنتاج قيمة (keyNew). ما هي مزايا استخدام الرقم العشوائي ودالة التحوير الآمنة أحادية الاتجاه للمسألة 8-7 مقارنةً مع هذا النهج ؟

الباب الثالث

أمن نظام الحاسب

يتناول الباب الثالث المسائل الأمنية على مستوى النظام، بما في ذلك التهديد الذي يُمثله المتسللون والإجراءات المضادة للتسلل وللفيروسات وكذلك استخدام الجدران النارية والأنظمة الموثوقة.

خريطة الطريق للجزء الثالث

الفصل التاسع : المتسللون

يبحث الفصل التاسع مجموعة متنوعة من التهديدات المتعلقة بالخدمة أو الوصول إلى المعلومات التي تنجم عن قراصنة يستغلون نقاط الضعف في أنظمة الحاسب المبنية على الشبكات. يبدأ الفصل بمناقشةٍ لأنواع الهجمات التي يمكن أن يقوم بها المستخدمون غير المصرّح لهم، أو المتسللون، ويحلل الطرق المختلفة للوقاية من تلك الهجمات واكتشافها. كما يغطي هذا الفصل المسألة ذات الصلة والمتعلقة بإدارة كلمات السر.

الفصل العاشر: البرمجيات الخبيثة

يتناول الفصل العاشر التهديدات البرمجية للنظم، مع التركيز بشكلٍ خاص على الفيروسات والديدان. ويبدأ الفصل بمسح لمختلف أنواع البرامجيات الخبيثة، ويخص – بنظرة أعمق – طبيعة الفيروسات والديدان. بعد ذلك يتناول الفصل موضوع الإجراءات المضادة، وأخيراً يناقش الهجمات الموزّعة للحرمان من الخدمة.

الفصل الحادي عشر : الجدران النارية

يُعدُّ استخدام جدار ناري أحد الوسائل القياسية لحماية الأصول المحلية لنظام الحاسب من خطر التهديدات الخارجية. يتناول الفصل الحادي عشر مبادىء تصميم الجدار الناري ويستعرض تقنيات محددة في هذا الصدد. كما يغطي ذلك الفصل المسألة ذات الصلة والمتعلقة بالنظم الموثوقة.