بررسی امنیتی پروتکل SNMP

فهرست

SNMPv1 & SNMPv2

این دو نسخه، نسخه‌های اولیهٔ پروتکل SNMP می‌باشند.

- **ساز و کارهای امنیتی**

روش احراز هویت استفاده شده در این نسخه‌ها community strings بوده است. بدین شکل که مدیر شبکه یک رشته کاراکتر را متناظر با یک سطح دسترسی به گره‌ها (node) در نظر گرفته و بوسیلهٔ آن با دستگاه‌ها تعامل برقرار می‌کند. این نام تعیین شده، یعنی community string، درواقع به منزلهٔ رمز عبور می‌باشد. هرکه رشتهٔ مذکور را در اختیار داشته باشد می‌تواند به گرهٔ مربوطه دسترسی یابد.

- **آسیب‌پذیری**

انتقال رشته بصورت رمزنگاری نشده در بستر انتقال، امکان packet sniffing را به فرد مهاجم می‌دهد. همچنین صحت و سلامت بسته‌های منتقل شده تأیید نمی‌شود.

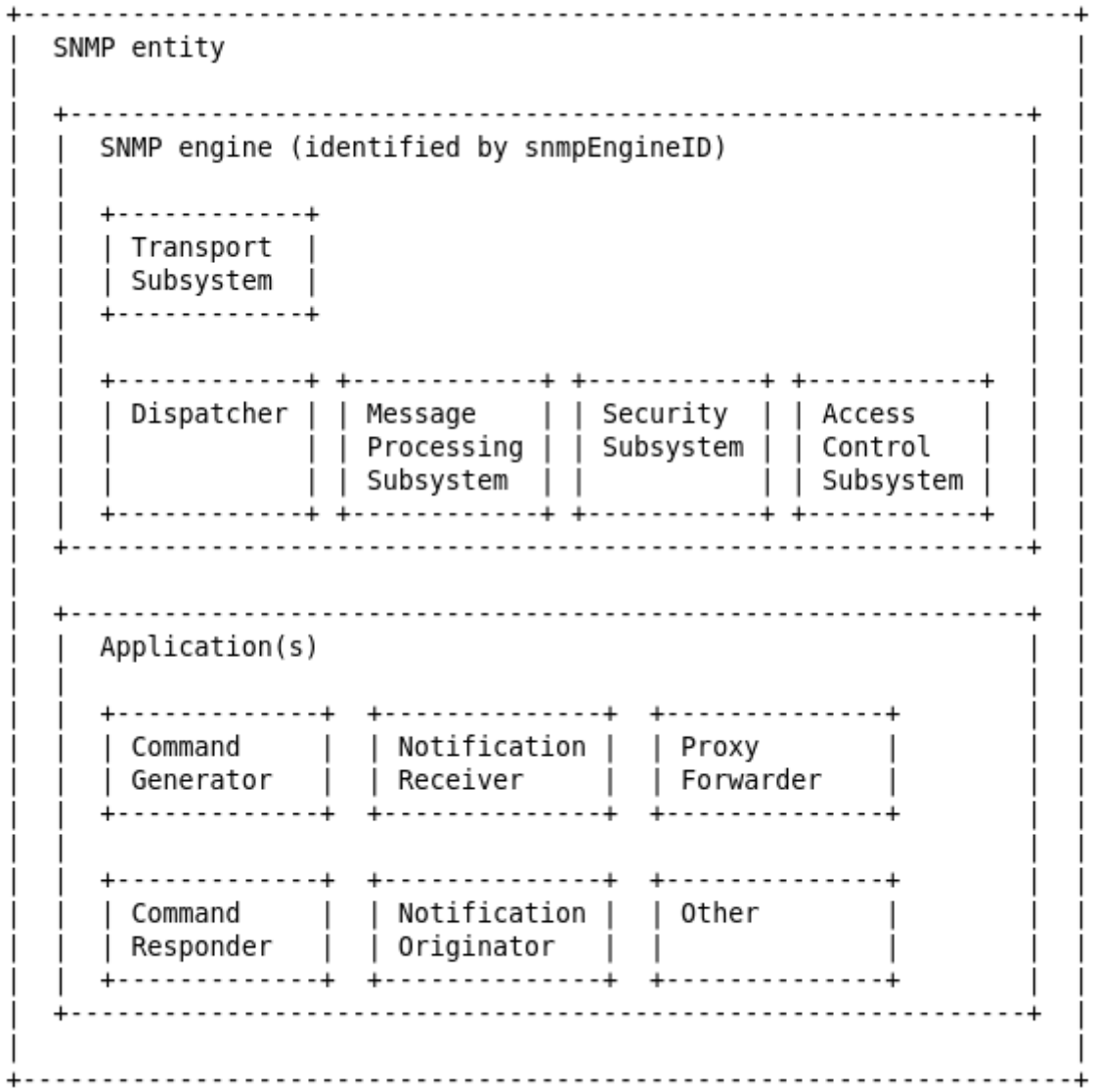
- **تهدیدات و حملات**
  - دزدین هویت مدیریت و ایفای نقش او، زیرا تنها ابزار شناسایی هویت community string می‌باشد.
  - گوش دادن به محتویات پیام‌های منتقل شده و به خطر انداختن حریم شخصی، به علت نبود رمزنگاری.
  - دستکاری بسته‌های ارسالی/دریافتی.

SNMPv3

نسخهٔ سوم پروتکل SNMP تغییرات گسترده‌ای را در زمینهٔ امکانات امنیتی جهت اصلاح نسخه‌های پیشین ارائه داده است.

- **معماری SNMP**

ابتدا لازم است کمی دربارهٔ معماری پروتکل بدانیم. این معماری به صورت ماژولار طراحی شده است تا وظایف هر بخش مستقل از دیگری بوده و نیز امکان ایجاد تغییرات لازم در آینده فراهم شود:



همانطور که مشاهده می‌شود دو بخش کلی SNMP Engine و Applications وجود دارد.

برنامه‌ها (Applications) از سرویس‌های فراهم شده توسط SNMP Engine استفاده می‌کنند.

مسئولیت هر بخش بطور خلاصه:

Dispatcher: توزیع پیام‌ها

Message Processing Subsystem: پردازش پیام‌ها

Security Subsystem: برقراری امنیت

Access Control Subsystem: مدیریت دسترسی‌های درخواستی به Object ها

Transport Subsystem: برقراری امنیت به کمک لایه‌های امنیتی خارجی

هر یک از زیرسیستم‌های اشاره شده خود دارای یک یا چند مدل می‌باشند. در اینجا توجه خود را به زیرسیستم‌های امنیتی معطوف می‌کنیم.

- **ساز و کارهای امنیتی**
  - **USM**

مدل USM یا User-based Security Model مدل امنیتی پیشفرض نسخهٔ سوم در زیرسیستم Security است که جهت دستیابی به اهداف امنیتی خود سرویس‌های ذیل را پشتیبانی می‌کند:

1. حفظ تمامیت داده‌ها:

   این امر به کمک پروتکل‌های احراز هویت HMAC-MD5-96 و HMAC-MD5-96 که به ترتیب از توابع هش MD5 و SHA استفاده میکنند صورت میگیرد. روش رمزنگاری نیز HMAC می‌باشد، که از هش کردن داده با بهره‌گیری از کلیدی مشخص استفاده میکند. همچنین پروتکل پیشرفته‌تر HMAC-SHA-2 نیز معرفی شده است که از خانواده هش‌های SHA-2 بهره می‌گیرد.

2. احراز هویت مبداء:

   بطور مشابه با استفاده از پروتکل‌های ذکر شده امکان اعتبار سنجی هویت ارسال‌کننده توسط دریافت‌کننده فراهم می‌شود.

3. محرمانگی اطلاعات:

   این مدل به کمک پروتکل‌های CBC-DES و AES-128 محرمانگی داده‌ها را حفظ می‌کند.

4. حفاظت در برابر تأخیرهای زمانی طولانی و پاسخ‌های نامرتبط:
   بخشی از پروسهٔ احراز هویت که کنترل میکند آیا یک پیام جدید است یا خیر و از تاخیرهای بیش از حد تعیین شده جلوگیری میکند.

همچنین کنترل می‌شود که هر درخواست با پاسخ مربوط به آن متناظر شود و پاسخ‌های نامربوط در نظر گرفته نشوند. این عمل به کمک متغیر msgID که شناسهٔ منحصر بفرد هر پیام است انجام می‌گیرد.

- **TSM**

یک مدل دیگر در زیرسیستم Security، طراحی شده جهت امکان استفاده از پروتکل‌های SSH TLS و DTLS. این مدل به علت ضعف‌هایی در USM مانند فرآیند Discovery (توضیح بیشتر در بخش آسیب‌پذیری و حملات، فرآیند Discovery) اغلب ترجیح داده می‌شود.

- **VACM**

VACM یا View-based Access Control Model که مدلی در زیرسیستم Access Control است با تنظیم سطح دسترسی و تعیین بخش‌هایی خاص از MIB جهت خواندن یا نوشتن از درخواست‌های نامربوط کاربران جلوگیری می‌کند.

- **آسیب‌پذیری‌ها و حملات**

- **فرآیند Discovery**

شاید بتوان اصلی‌ترین نقطهٔ آسیب‌پذیر نسخهٔ سوم پروتکل SNMP را فرآیند Discovery آن دانست.
هر موتور SNMP دارای یک شناسه بنام snmpEngineID می‌باشد. این شناسه منحصر بفرد و به عبارتی نمایانگر هویت هر گره است. برای آنکه مدیر شبکه با یک گره ارتباط برقرار کند ابتدا باید snmpEngineID آن را بداند.

بدین ترتیب نخستین فرمان ارسالی مدیر، فرمان درخواست این شناسه است. بدنبال آن پیامی **رمز نشده** از سمت گره حاوی این مشخصه ارسال می‌شود. به علت رمز نشدن، این پیام و لذا مقدار snmpEngineID توسط فرد نفوذگر قابل رؤیت است.

همچنین پیام ارسالی احراز هویت نیز نمی‌شود، لذا در صورت دستکاری شناسه، مدیر این تفاوت را متوجه نخواهد شد.

نکتهٔ بسیار حائز اهمیت آن است که کلیدهای رمزنگاری و احراز هویت کاربر حاصل ترکیبی از رمزعبور (تعیین شده توسط مدیر) و **snmpEngineID** می‌باشند!

سناریوی مقابل را متصور شویم. فرض کنیم یک کلید رمزنگاری لو رفته است که می‌دانیم مرتبط با کدام snmpeEngineID است. حال آلیس را بعنوان مدیر شبکه و باب را یک میزبان (Host) معمولی در نظر بگیریم. فرد نفوذگر میان این دو قرار دارد.

نخست مهاجم فرمان Discovery آلیس را بدون اعمال به باب انتقال می‌دهد. باب یک پاسخ حاوی snmpeEngineID خود میفرستد. حال کافی است مهاجم شناسهٔ بسته عبوری را با شناسهٔ متناظر کلید لو رفته جایگزین کند و پاسخ را عبور دهد. این امر ممکن است چرا که برای فرآیند Discovery نه رمزنگاری و نه احراز هویت اعمال می‌شود.

بدین ترتیب مهاجم نه تنها قادر است با در دست داشتن کلید لو رفتهٔ خود فرمان‌های بعدی آلیس به باب (و برعکس) را **بخواند**، بلکه می‌تواند **نقش** باب را ایفا کرده و اطلاعات غلط مخابره نماید.

- **EngineID**

همانطور که در بخش قبل گفته شد این شناسه نقش بسیار مهمی در پروتکل ایفا میکند. چرا که به منزلهٔ هویت هر گره بوده و نیز در الگوریتم‌های تولید کلیدهای احراز هویت و رمزنگاری، یکی از دو ورودی به کار رفته است. لذا در صورت افشا شدن خطراتی متوجه مدیر شبکه خواهد شد.این شناسه یک مقدار هگزادسیمال است که ساختار آن از موارد ذیل تشکیل شده است:(تمامی ارقام در مبنای ۱۶ می‌باشند.)

- بایت اول آن ثابت و برابر 80 می‌باشد.
- ۳ بایت بعدی شمارهٔ مخصوص شرکت است. بطور مثال شمارهٔ متناظر با برنامهٔ Net-SNMP برابر 8072 است (در مبنای ۱۰) که معادل آن در مبنای ۱۶ می‌شود: 1f88
- بایت پنجم که یکی از موارد زیر است مشخص میکند چگونه باقی‌ماندهٔ بایت‌ها مقدار دهی شوند:
  - 0: رزرو شده، بدون استفاده
  - 1: آی‌پی نسخهٔ ۴، (۴ بایت)
  - 2: آی‌پی نسخهٔ ۶، (۱۶ بایت)
  - 3: آدرس فیزیکی (Mac Address)، (۶ بایت)
  - 4: متن، تعیین شده توسط مدیر

- 5: بایت، تعیین شده توسط مدیر
- local engine :6
- 7-127: رزرو شده، بدون استفاده
- 128-255: وابسته به شرکت مرتبط

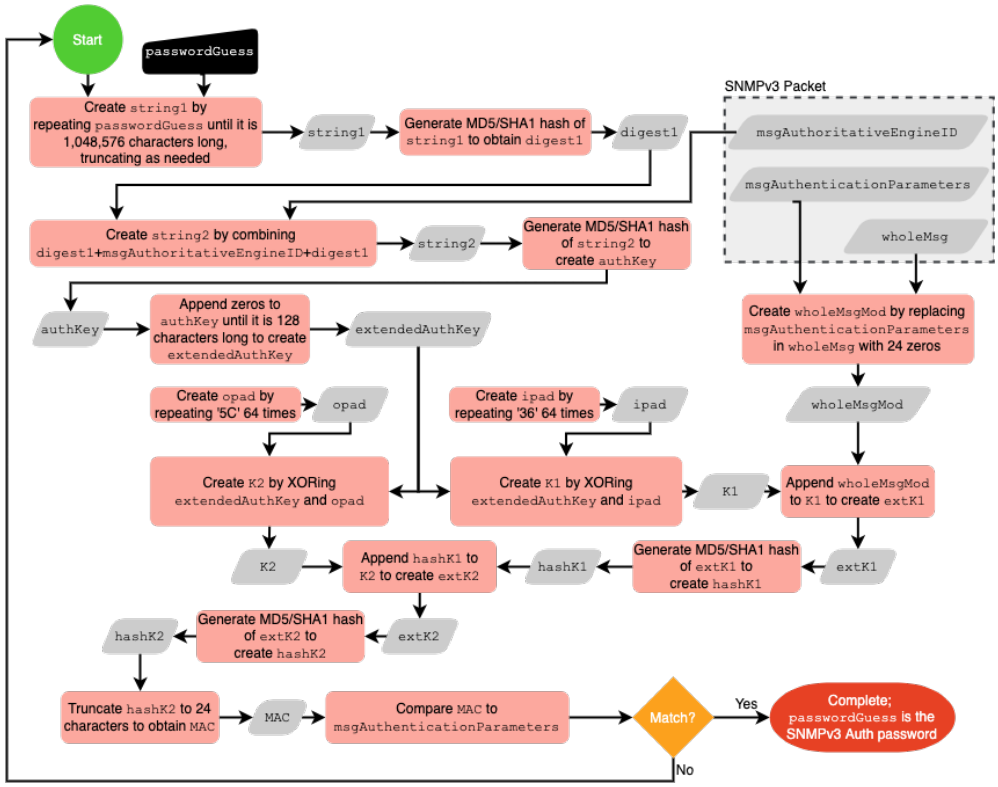
چنانچه بایت پنجم به بعد توسط مواردی چون آی‌پی، آدرس فیزیکی و یا متن تعریف شود چند ایراد بوجود خواهد آمد:

- با یک حد حساب شده در مورد شرکت سازنده و پارامتر بکار رفته (یعنی آی‌پی یا ...) شناسهٔ EngineID قابل شناسایی خواهد شد.
- طی فرآیند Discovery که شناسه قابل رؤیت است می‌توان به اطلاعاتی چون آی‌پی و آدرس فیزیکی دستگاه که برای ساخت شناسه استفاده شده است پی برد.
- خصیصهٔ منحصر به فرد بودن شناسه دچار اشکال می‌شود. بطور مثال هنگام استفاده از یک متن یکسان برای دو گره و یا زمان تغییر آی‌پی دستگاه.

راه‌حل ایده‌آل استفاده از امکانات نرم‌افزار است که می‌تواند شناسه را به صورت شبه-تصادفی تولید کند. (رجوع شود به بخش پیکربندی، Net-SNMP)

- **حملهٔ Brute Force**

اجرای حملهٔ دیکشنری بر روی پروتکل احراز هویت و همچنین رمزنگاری DES به علت کوتاه بودن طول کلید آن.
تصویر صفحهٔ بعد روال کارکرد پروتکل‌های احراز هویت را به خوبی نشان می‌دهد:



توضیح تصویر: بطور خلاصه مشاهده میکنیم که authkey از ترکیب رمزعبور کاربر و snmpEngineID بدست می‌آید. این کلید به همراه دو کلید ثابت دیگر ipad و opad کلیدهای HMAC این الگوریتم را تشکیل می‌دهند.

جهت انجام حملهٔ Brute Force می‌توان از برنامهٔ ذیل استفاده نمود:

https://github.com/applied-risk/snmpv3brute

نحوهٔ استفاده:

این برنامه می‌تواند یک پرونده با پسوند pcap حاوی ارتباطات SNMP، به همراه یک لیستِ رمز (Password List) از کاربر دریافت کرده و روال Brute Force را برای بدست آوردن رمزعبور مدیر اجرا کند.

آرگومان‌های برنامه:

-h= راهنمای برنامه

-a = تعیین الگوریتم مورد استفاده که md5 و یا sha می‌باشد

-w = لیستِ رمز

-W = ورود رمزها به صورت تکی

-p = پروندهٔ pcap مورد نظر

-m = وارد کردن اطلاعات ورودی الگوریتم به صورت دستی

مثال:

```
snmpv3brute.py -w wordlist.txt -p foo.pcapng
```

```
snmpv3brute.py -W password1 password2 password 3 -p foo.pcapng
```

```
snmpv3brute.py -W password1 password2 -p foo.pcapng -w wordlist.txt
```

```
snmpv3brute.py -m <msgAuthoriativeEngineID> <msgAuthenticationParameters>
<msgWhole> -w wordlist.txt
```

```
snmpv3brute.py -m 80001f888056417b0bd201d85d00000000
a34b57081ff0cef821e4da43
3081dc020103301002043cabfa64020205c0040103020103043f303d041180001f88805
6417b0bd201d85d0000000002010102020 0a20409736e6d705f75736572040ca34b570
81ff0cef821e4da430408bec2e5f547aaa89c048183dfe158807f83a660d37264c7f397a8
a42c237988ee829c52b003f6d772df683c51acb56bb327a36ee590e1d65c9466e9d18a4
8e80539e5fff12006d2fba6bc61756956285b84bafe773b6359d2273db3b6e49f89a6609
a86ac5f440d4bfa55b17af5a81db1fa0030402bba9befad240addc41d9b394d0fb2c4a3f5
ffde3730485cdaf6
```

پیکربندی

- **نکات کلی**
  - همانطور که در بخش‌های قبل ذکر شد، نسخه‌های ۱ و ۲ پروتکل از امنیت بسیار پایینی برخوردار هستند و استفاده از نسخهٔ ۳ به همراه ساز و کارهای امنیتی متناظر آن پیشنهاد می‌شود. اما در صورت استفاده از نسخه‌های ابتدایی چند نکته باید مورد توجه قرار گیرد:

- عدم استفاده از community string های پیشفرض: تنظیمات پیشفرض رایج رشته‌ها بصورت public جهت دسترسی فقط-خواندن و private جهت دسترسی خواندن-نوشتن می‌باشد. به سبب آگاهی قبلی از این رشته‌ها تغییر ندادن این تنظیمات به فرد نفوذگر امکان سوءاستفاده از شبکه را می‌دهد.
- استفاده از رشته‌های قوی و تغییر متناوب آن‌ها: مانند تمامی رمزعبور ها لازم است از community string های با طول زیاد و متشکل از کاراکترهای متنوع استفاده کرد. ضروری است این رشته‌ها در بازه‌های منظم و همینطور مطابق با سیاست‌های امنیت شبکه تعویض کردند. بطور مثال زمان تعویض مدیر شبکه.
- چه در نسخه‌های ابتدای و چه در نسخهٔ سوم باتوجه به مدل‌های بررسی شده تنظیمات امنیتی به حد نیاز انجام گیرد. بطور مثال اعمال پروتکل‌های رمزنگاری و احراز هویت و استفاده از قابلیت‌های مدل VACM مانند محدود کردن کاربران به OID های خاص.

- **Net-SNMP**

این بستهٔ نرم‌افزاری متشکل از برنامه‌های گوناگون در دو گروه Applications و Daemons می‌باشد که امکان استفاده از پروتکل SNMP را فراهم می‌کند.

Net-SNMP علاوه بر پشتیبانی از نسخه‌های ۱ و ۲ پروتکل، از SNMPv3 بهمراه USM، DTLS و TLS نیز پشتیبانی قدرت‌مندی به همراه دارد.

این برنامه همچنین پشتیبانی جزئی نسخهٔ ۳ برای SSH و استفادهٔ آزمایشی از Kerberos را دربر دارد.

این پروژه سرویس snmpd را ارائه می‌دهد که به منظور دریافت و تحلیل درخواست‌ها، جمع‌آوری اطلاعات مربوطه، انجام اعمال درخواست شده و نهایتاً ارسال پاسخ، در پس‌زمینه اجرا می‌شود.

حال به دستورات مربوط به پیکربندی غلط (Misconfiguration) در پروندهٔ تنظیمات snmpd می‌پردازیم:

1. agentaddress: این دستور یک لیست از آدرس‌هایی که برای دریافت درخواست به آن‌ها گوش میدهد تعریف میکند. مقدار پیشفرض آن گوش دادن بر روی UDP، پورت ۱۶۱ و **تمامی** رابط‌های IPv4 است.

2. maxGetbulkRepeats: تعیین کنندهٔ ماکزیمم تعداد پاسخ‌های مجاز برای متغیری واحد در یک درخواست getbulk. برابر قرار دادن با صفر مقدار پیشرفض آن را فعال می‌کند و ۱- مقدار نامحدود را. از آنجا که حافظهٔ مورد نیاز از قبل تخصیص می‌یابد، قرار دادن مقدار نامحدود در یک شبکهٔ نامطئن، ناامن می‌باشد.
تنظیم پیشفرض **نامحدود** می‌باشد.

3. maxGetbulkResponses: تعیین کنندهٔ ماکزیمم تعداد پاسخ‌های مجاز برای یک درخواست getbulk. برابر قرار دادن با صفر مقدار پیشرفض آن را فعال می‌کند و ۱- مقدار نامحدود را.
از آنجا که حافظهٔ مورد نیاز از قبل تخصیص می‌یابد، قرار دادن مقدار نامحدود در یک شبکهٔ نامطئن، ناامن می‌باشد.
تنظیم پیشفرض ۱۰۰ می‌باشد.

4. engineID: پروژهٔ Net-SNMP بطور پیشفرض برای تولید منحصر بفرد شناسهٔ EngineID از زمان لحظه‌ای سیستم و یک عدد تصادفی استفاده می‌کند.
به کمک دستور *engineID* می‌توان تولید شناسه را بر پایهٔ متن و با استفاده از دستور engineIDType بر پایهٔ آی‌پی و یا آدرس فیزیکی انجام داد.
همانطور که در بخش آسیب‌پذیری‌ها و حملات اشاره شد در صورت تنظیم دستی احتیاط لازم باید در نظر گرفته شود.

5. در SNMPv3 می‌توان توسط پروتکل‌های TLS و DTLS تونل‌سازی نمود. TLS بر روی TCP اجرا می‌شود و DTLS معادل UDP آن می‌باشد. درحالی که پروتکل‌های بر پایهٔ TCP برای شبکه‌های پایدار کارآمد می‌باشند اما در شبکه‌های ناپایدار (و یا حتی با قطعی متوسط، ۲۰-۳۰ درصد از دست رفتگی بسته) بسرعت مشکل‌زا می‌شوند. در نتیجه در انتخاب TLS یا DTLS متناسب با محیط شبکه باید تصمیم گیری نمود.

6. proc و procfix: پروسهٔ مورد نظر را تحت نظر میگیرد (monitor)، در صورتی که برنامه بیش از تعداد مشخص شده در حال اجرا باشد یک trap به مدیر ارسال کرده و یا بر روی پروسه دستور تعیین شده‌ای (مانند خاموش کردن یا راه اندازی مجدد) اجرا می‌کند.
بطور مثال اگر برنامهٔ sendmail به تعداد بالایی در حال اجرا باشد می‌توان آن را متوقف کرد و یا می‌توان اطمینان حاصل کرد که تنها یک پروسهٔ snmpd در حال اجرا است.
به طریق مشابه به کمک دستورهای swap ، load ، disk و سایر گزینه‌های نظارتی می‌توان فشار وارده بر سیستم را کاهش داده و منابع را مدیریت نمود.

7. authtrapenable: تصمیم می‌گیرد که در صورت شکست خوردن احراز هویت اقدام به ارسال trap کند یا خیر. مقدار ۱ ارسال را فعال کرده و مقدار ۲ ارسال را غیر فعال می‌کند. توجه به این نکته حائز اهمیت است که که آبجکت متناظر این دستور در حالت عادی دسترسی خواندن-نوشتن دارد اما تغییر دادن آبجکت با استفاده از این دستور آن را فقط-خواندن خواهد کرد. لذا در این صورت توسط درخواست‌های SET تغییر نمی‌کند.
این گزینه بطور پیشفرض **غیرفعال** می‌باشد.

- **Cisco**

محصولات Cisco تمهیدات ذیل را جهت برقراری امنیت در حوزهٔ پروتکل SNMP پیشنهاد میکند:

- در صورت استفاده از نسخه‌های ۱ و ۲، بهره‌گیری مناسب از Community String ها باتوجه نکات ذکر شده در بخش‌های قبلی.
- استفادهٔ مناسب از سطوح دسترسی SNMP مانند خواندن-نوشتن و یا فقط-خواندن.
- استفاده از ACL ها به منظور محدود کردن آدرس‌های درخواستی.
- بهره‌گیری از SNMP View ها جهت محدود کردن دسترسی‌ها به بخش‌های مورد نظر از MIB.
- بکارگیری نسخهٔ سوم همراه با پروتکل‌های امنیتی احراز هویت و رمزنگاری داده‌ها.
- پیاده‌سازی قابلیت MPP (مخفف Management Plane Protection) که اجازه می‌دهد ترافیک پروتکل SNMP تنها بر روی رابط مشخص شده جاری شود.

فرهنگ واژه

احراز هویت: Authentication

رشته: String

گره: Node

دستگاه: Device

تحت نظر گیری: Monitoring

پیکربندی: Configuration

میزبان: Host

پرونده: File

رابط: Interface

مراجع

- مفاهیم کلی پروتکل:

https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol

- معماری SNMP:

https://tools.ietf.org/html/rfc3411

https://tools.ietf.org/html/rfc5590

- مدل USM:

https://tools.ietf.org/html/rfc3414

- حملهٔ Discovery:

https://www.usenix.org/system/files/conference/woot12/woot12-final14.pdf

- EngineID:

https://tools.ietf.org/html/rfc3411
https://tools.ietf.org/html/rfc5343

- حملهٔ Brute Force:

https://applied-risk.com/resources/brute-forcing-snmpv3-authentication

https://github.com/applied-risk/snmpv3brute

- Net-SNMP:

snmpd.conf manpage
http://www.net-snmp.org/wiki/index.php/Tutorials

- Cisco:

https://www.cisco.com/c/en/us/support/docs/ip/access-lists/13608-21.html#anc54
https://www.cisco.com/c/en/us/td/docs/ios/security/configuration/guide/
sec_mgmt_plane_prot.html