تكوين إعلامات البريد الإلكتروني مع البرامج النصية لتنبيهات IDS باستخدام مركز مراقبة CiscoWorks للأمان

المحتويات

المقدمة

تتوفر لدى "مراقبة الأمان" إمكانية إرسال إعلامات عبر البريد الإلكتروني عند تشغيل "قاعدة الأحداث". المتغيرات المضمنة التي يمكن إستخدامها ضمن إعلام البريد الإلكتروني لكل حدث لا تتضمن أشياء مثل معرف التوقيع، مصدر التنبيه ووجهته، وهكذا. يقدم هذا المستند إرشادات يمكنك إستخدامها لتكوين "مراقبة الأمان" لتضمين هذه المتغيرات (وغيرها) داخل رسالة إعلام البريد الإلكتروني.

المتطلبات الأساسية

المتطلبات

لا توجد متطلبات خاصة لهذا المستند.

المكونات المستخدمة

لا يقتصر هذا المستند على إصدارات برامج ومكونات مادية معينة. ومع ذلك، تأكد من إستخدام نص Perl المناسب بناء على ما تشغله إصدارات المستشعر في بيئتك.

الاصطلاحات

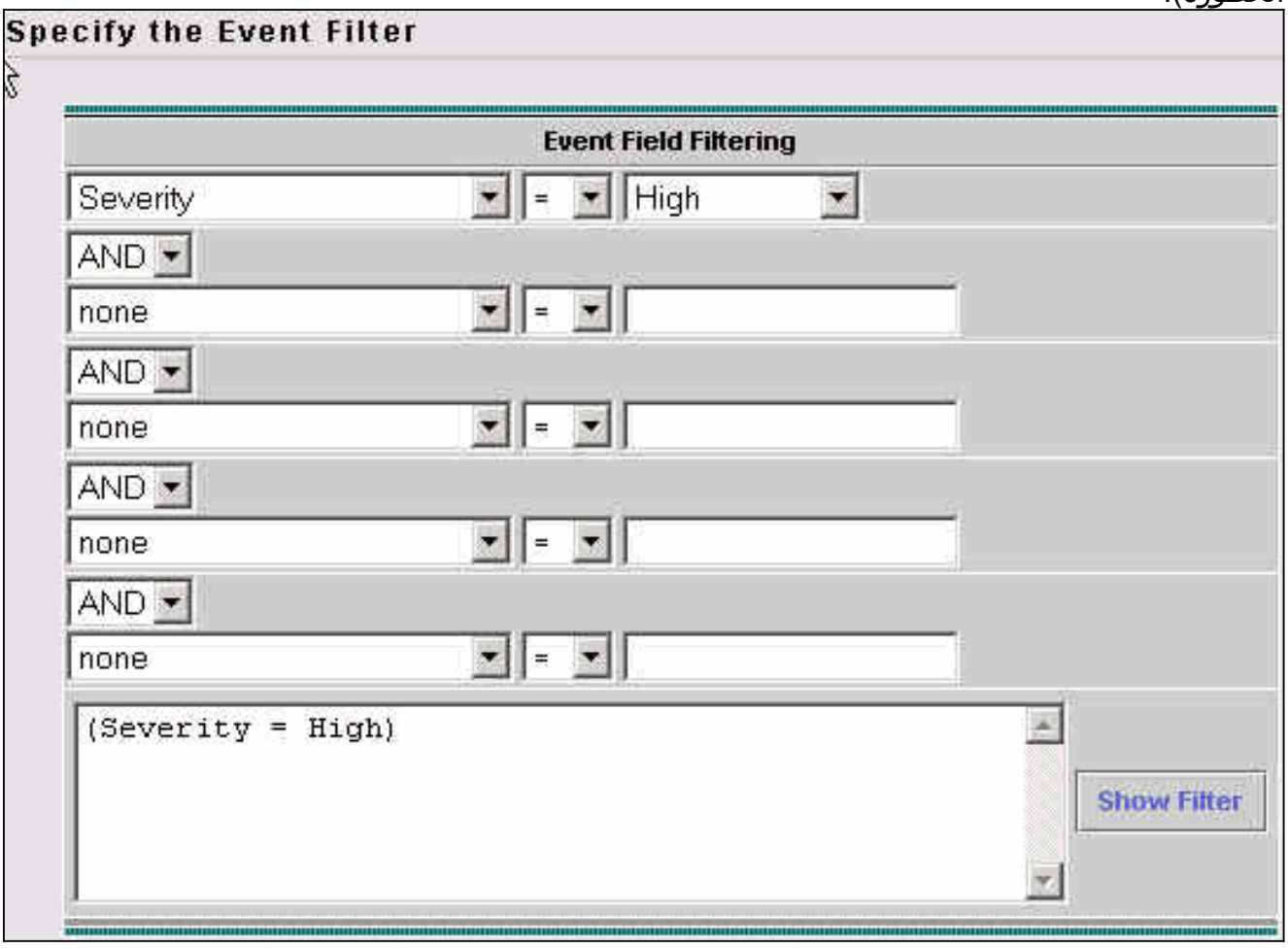
راجع اصطلاحات تلميحات Cisco التقنية للحصول على مزيد من المعلومات حول اصطلاحات المستندات.

إجراء تكوين إعلام البريد الإلكتروني

أستخدم هذا الإجراء لتكوين إعلامات البريد الإلكتروني.

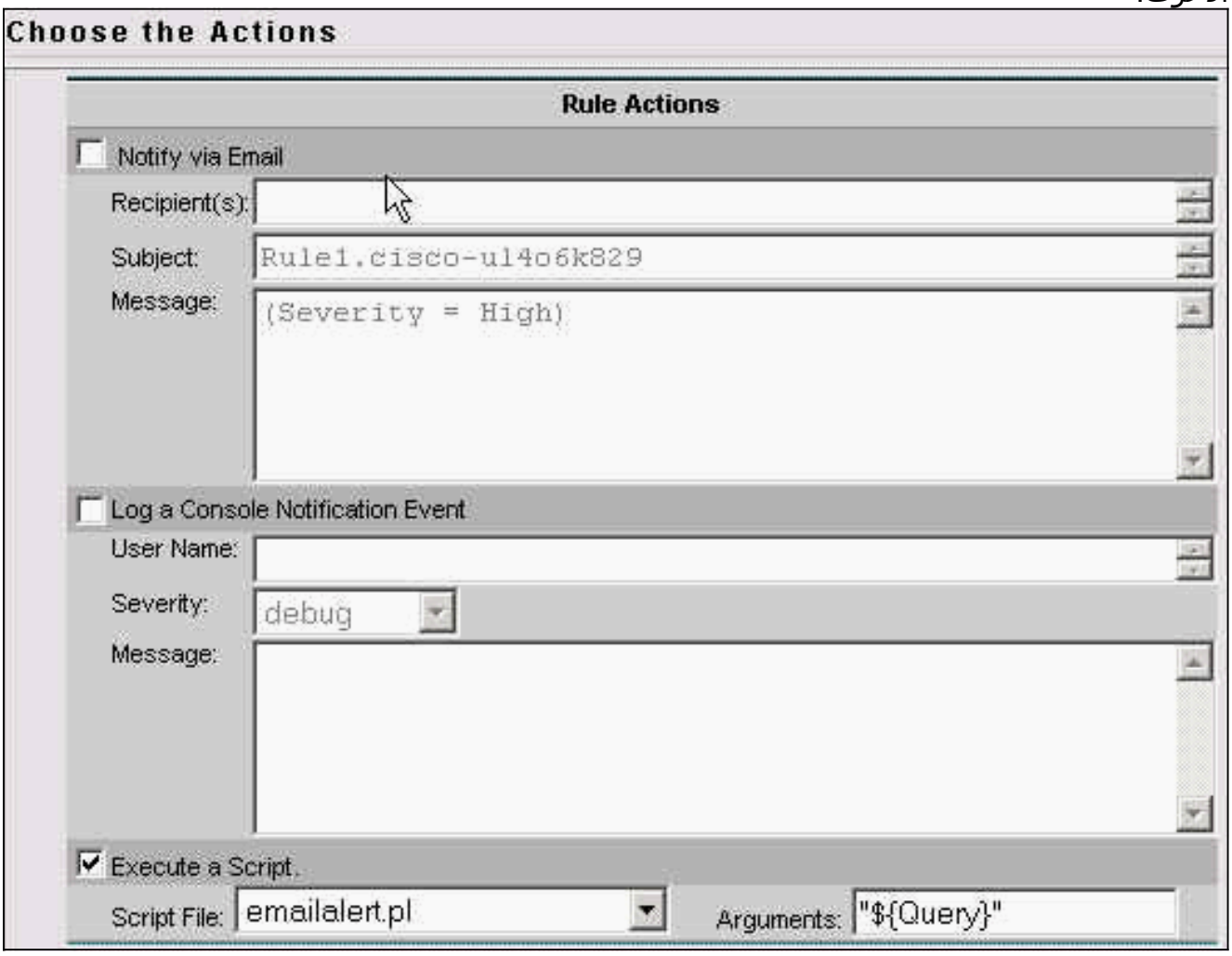
ملاحظة: لإرسال البريد الإلكتروني إلى عنوان البريد الإلكتروني الصحيح، تأكد من تغيير عنوان البريد الإلكتروني في البرنامج النصي.

1. انسخ أحد هذه البرامج النصية إلى الدليل $BASE\CSCOpx\MDC\etc\ids\scripts على خادم حل إدارة الأمان/الشبكة الخاصة الظاهرية (VPN). يسمح لك ذلك بتحديده لاحقا في العملية عند تعريفك لقاعدة حدث. حفظ البرنامج النصي على هيئة emailAlert.pl. **ملاحظة:** إذا كنت تستخدم اسما آخر، تأكد من الإشارة إلى هذا الاسم في "قاعدة الحدث" المعرفة في هذه الخطوات.بالنسبة لأجهزة إستشعار الإصدار x.3، أستخدم البرنامج النصي لأجهزة الاستشعار x.3بالنسبة لأجهزة إستشعار الإصدار x.4، أستخدم البرنامج النصي لأجهزة الاستشعار x.4لأجهزة إستشعار الإصدار x.5، أستخدم البرنامج النصي لأجهزة الاستشعار x.5إذا كان لديك مجموعة من إصدارات المستشعر، فإن Cisco توصيك بالترقية حتى تكون جميعها في مستوى الإصدار نفسه. وذلك لأنه يمكن تشغيل واحد فقط من هذه البرامج النصية في أي وقت.
2. يحتوي البرنامج النصي على تعليقات تشرح كل جزء وأي مدخل مطلوب. قم بشكل خاص بتعديل متغير `EmailRcpt$` (بالقرب من أعلى الملف) ليكون عنوان البريد الإلكتروني للشخص الذي سيقوم باستلام التنبيهات.
3. قم بتحديد قاعدة حدث داخل "مراقبة الأمان" لاستدعاء نص Perl جديد. من صفحة مراقبة الأمان الرئيسية، أختر **Admin > قواعد الحدث** وقم بإضافة حدث جديد.
4. في نافذة تحديد عامل تصفية الحدث، قم بإضافة عوامل التصفية التي تريد تشغيل تنبيه البريد الإلكتروني (في العينة هنا، يتم إرسال بريد إلكتروني لأي تنبيه عالي الخطورة).



5. في نافذة إختيار العملية، حدد المربع لتنفيذ نص تنفيذي وحدد اسم النص التنفيذي من المربع المنسدل.

6. في قسم الوسيطات، أدخل **"{Query}$"** كما هو موضح هنا.**ملاحظة:** يجب إدخال هذا كما هو هنا تماما، بما في ذلك الاقتباسات المزدوجة. كما أنه حساس لحالة الأحرف.



7. عندما يتم تلقي تنبيه، كما هو معرف في عوامل تصفية الأحداث (في هذا المثال، تنبيه عالي الخطورة)، يتم إستدعاء البرنامج النصي المسمى EmailAlert.pl مع وسيطة `{Query}$`. يحتوي هذا على معلومات إضافية حول التنبيه. يحلل النص التنفيذي كل الحقول المنفصلة ويستخدم برنامج يسمى "blat" لإرسال بريد إلكتروني إلى المستخدم النهائي.
8. BLAT هو برنامج بريد إلكتروني مجاني يستخدم على أنظمة Windows لإرسال رسائل بريد إلكتروني من ملفات الدفعات أو برامج Perl النصية. يتم تضمينه كجزء من تثبيت VMS في دليل $BASE\CSCOpx\bin. للتحقق من إعدادات المسار، افتح نافذة موجه الأوامر على خادم VMS واكتب **blat**.إذا إستلمت خطأ ، فعليك إما نسخ ملف blat.exe إلى دليل winnt\system32، أو البحث عنه وفتحه من الدليل الذي يوجد به. لتثبيت هذا، قم بتشغيل:

```
blat -install
```

بمجرد تثبيت هذا البرنامج، يتم الانتهاء.

البرامج النصية

هذه هي البرامج النصية المشار إليها في الخطوة 1 من إجراء التكوين:

نص مستشعر x.3

أستخدم هذا البرنامج النصي لأجهزة إستشعار الإصدار x.3.

أجهزة الاستشعار x.3

```
#!/usr/bin/perl
#**************************************************
**********
#
# FILE NAME : emailalert.pl
#
# DESCRIPTION : This file is a perl script that will be
executed as an
# action when an IDS-MC Event Rule triggers, and will
send an
# email to $EmailRcpt with additional alert parameters
(similar to
# (the functionality available with CSPM notifications
#
# NOTE: this script only works with 3.x sensors,
alarms from 4.0
# sensors are stored differently and cannot be
represented
# .in a similar format
#
# NOTE: check the "system" command in the script for
the correct
# format depending on whether you're using
IDSMC/SecMon
# v1.0 or v1.1, you may need the "-on" command-
.line option
#
# NOTE : This script takes the ${Query} keyword from
the
# triggered rule, extracts the set of alarms
that caused
# the rule to trigger. It then reads the last
alarm of
# this set, parses the individual alarm fields,
and
# calls the legacy script with the same set of
command
# .line arguments as CSPM
#
# The calling sequence of this script must be of the
:form
#
# "{emailalert.pl "${Query
#
:Where #
#
# Query}" - this is the query keyword}$"
dynamically
# .output by the rule when it triggers
# It MUST be wrapped in double quotes when
```

```
                                specifying it in the Arguments
                    .box on the Rule Actions panel         #
                                                           #
                                                           #
**********************************************************#
                                          ****************
                                                          ##
   The following are the only two variables that need ##
                              changing. $TempIDSFile can be any
 filename (doesn't have to exist), just make sure the ##
                                   directory that you specify
      exists. Make sure to use 2 backslashes for each ##
                              directory, the first backslash is
         so the Perl interpretor doesn't error on the ##
                                                  .pathname
                                                          ##
EmailRcpt is the person that is going to receive the$ ##
                                    email notifications. Also
       make sure you escape the @ symbol by putting a ##
                          backslash in front of it, otherwise
                        .you'll get a Perl syntax error ##
                                                          ##
                ;"TempIDSFile = "c:\\temp\\idsalert.txt$
                       ;"EmailRcpt = "nobody\@cisco.com$

                                                          ##
                                pull out command line arg ##
                                                          ##

                                    ;[whereClause = $ARGV[0$

                                                          ##
    extract all the alarms matching search expression ##
                                                          ##

                                    ;"tmpFile = "alarms.out$

     The following line will extract alarms from 1.0 ##
                                    IDSMC/SecMon database, if
  using 1.1 comment out the line below and un-comment ##
                                       the other system line
                                                .below it ##

                                V1.0 IDSMC/SecMon version ##
  ;(""\system("IdsAlarms -s\"$whereClause\" -f\"$tmpFile

                               .V1.1 IDSMC/SecMon version ##
          system("IdsAlarms -on -s\"$whereClause\" - ##
                                            ;(""\f\"$tmpFile
                                                          ##

                                open matching alarm output #

                          } ((if (!open(ALARM_FILE, $tmpFile
           ;"print "Could not open ", $tmpFile, "\n
                                               ;exit -1
                                                        {

                                       read to last line #

                                    } (<while (<ALARM_FILE
                                           ;_$ = line$
                                                        {
```

```
                                                clean up #

                                          ;(close(ALARM_FILE
                                           ;(unlink($tmpFile

                                                           ##
                                  split last line into fields ##
                                                           ##

                                    ;(fields = split(/,/, $line@

                                        ;[eventType = @fields[0$
                                         ;[recordId = @fields[1$
                      gmtTimestamp = 0; # need gmt time_t$
                   localTimestamp = 0; # need local time_t$
                                        ;[localDate = @fields[4$
                                        ;[localTime = @fields[5$
                                            ;[appId = @fields[6$
                                           ;[hostId = @fields[7$
                                            ;[orgId = @fields[8$
                                     ;[srcDirection = @fields[9$
                                   ;[destDirection = @fields[10$
                                        ;[severity = @fields[11$
                                           ;[sigId = @fields[12$
                                        ;[subSigId = @fields[13$
                                           ;"protocol = "TCP/IP$
                                         ;[srcAddr = @fields[15$
                                        ;[destAddr = @fields[16$
                                         ;[srcPort = @fields[17$
                                        ;[destPort = @fields[18$
                                     ;[routerAddr = @fields[19$
                                   ;[contextString = @fields[20$


                  ,Open temp file to write alert data into ##

open(OUT,">$TempIDSFile") || warn "Unable to open output
                                                    ;"file!\n

    Now write your email notification message. You're ##
                                    writing the following into
the temporary file for the moment, but this will then ##
                                   :be emailed. Use the format
                                                           ##
print (OUT "Your text with any variable name from the ##
                                               ;("list above \n
                                                           ##
Again, make sure you escape special characters with a ##
                     backslash (note the : in between $sigId
        (and $subSigId has a backslash in front of it ##

                                               ;("print(OUT "\n
         print(OUT "Received severity $severity alert at
                                    ;("$localDate $localTime\n
 print(OUT "Signature ID $sigId\:$subSigId from $srcAddr
                                               ;("to $destAddr\n
                                   ;("print(OUT "$contextString
                                                ;(close(OUT

then call "blat" to send contents of that file in the ##
                                    .body of an email message
    Blat is a freeware email program for WinNT/95, it ##
                                        comes with VMS in the
 BASE\CSCOpx\bin directory, make sure you install it$ ##
```

```
                                                :first by running
                                                              ##
    blat -install <SMTP server address> <source email ##
                                                <address>
                                                              ##
       For more help on blat, just type "blat" at the ##
                  command prompt on your VMS system (make
        sure it's in your path (feel free to move the ##
                  executable to c:\winnt\system32 BEFORE
   you run the install, that'll make sure your system ##
                                        .(can always find it

     system ("blat \"$TempIDSFile\" -t \"$EmailRcpt\" -s
                                  ;(""\\"Received IDS alert
```

نص مستشعر x.4

أستخدم هذا البرنامج النصي لأجهزة إستشعار الإصدار x.4.

أجهزة الاستشعار x.4

```
                                                usr/bin/perluse/!#
*****************************************#;Time::Local
                                ***************************
                                                             #
                                 FILE NAME : emailalert.pl #
                                                             #
 DESCRIPTION : This file is a perl script that will be #
                                                executed as an
   action when an IDS-MC Event Rule triggers, and will #
                                                    send an
  email to $EmailRcpt with additional alert parameters #
                                                 (similar to
  (the functionality available with CSPM notifications #
                                                             #
NOTE: this script only works with 4.x sensors. It will #
                                .not work with 3.x sensors #
                                                             #
   NOTES : This script takes the ${Query} keyword from #
                                                       the
triggered rule, extracts the set of alarms that caused #
  the rule to trigger. It then reads the last alarm of #
     this set, parses the individual alarm fields, and #
  calls the legacy script with the same set of command #
                                  .line arguments as CSPM #
                                                             #
    The calling sequence of this script must be of the #
                                                    :form
                                                             #
                                  "{emailalert.pl "${Query #
                                                             #
                                                  :Where #
                                                             #
    Query}" - this is the query keyword dynamically}$" #
                   .output by the rule when it triggers #
                    It MUST be wrapped in double quotes #
                    when specifying it in the Arguments #
                         .box on the Rule Actions panel #
                                                             #
                                                             #
*****************************************************#
```

```
                                                ****************
                                                              ##
  The following are the only two variables that need ##
                        changing. $TempIDSFile can be any
 filename (doesn't have to exist), just make sure the ##
                                directory that you specify
     exists. Make sure to use 2 backslashes for each ##
                        directory, the first backslash is
        so the Perl interpretor doesn't error on the ##
                                                .pathname
                                                              ##
EmailRcpt is the person that is going to receive the$ ##
                                   email notifications. Also
       make sure you escape the @ symbol by putting a ##
                        backslash in front of it, otherwise
                       .you'll get a Perl syntax error ##
                                                              ##


                ;"TempIDSFile = "c:\\temp\\idsalert.txt$
               ;"EmailRcpt = "yourname\@yourcompany.com$

   subroutine to add leading 0's to any date variable #
                                       .that's less than 10
                                             } sub add_zero
                                          ;_@ = (my ($var
                                       } (if ($var < 10
                                       var = "0" .$var$
                                                         {
                                             ;return $var
                                                         {

 subroutine to find one or more IP addresses within an #
                                XML tag (we can have multiple
          .(victims and/or attackers in one alert now #
                                       } sub find_addresses
                                            ;_@ = (my ($var
                                         ;() = my @addresses
                                              } (/if (m/$var
                                                 ;&$ = raw$
            } (/{while ($raw =~ m/(\d{1,3}\.){3}\d{1,3
                                         ;&$,push @addresses
                                                 ;'$ = raw$
                                                         {
                                ;(var = join(', ',@addresses$
                                                 ;return $var
                                                         {
                                                         {

                                  pull out command line arg #

                                     ;[whereClause = $ARGV[0$

     extract all the alarms matching search expression #

                                     ;"tmpFile = "alarms.out$

      .Extract the XML alert/event out of the database #

  ;(""\system("IdsAlarms -s\"$whereClause\" -f\"$tmpFile

                                  open matching alarm output #

                           } ((if (!open(ALARM_FILE, $tmpFile
                          ;"print "Could not open $tmpFile\n
```

```
                                                    ;exit -1
                                                           {

                                                read to last line #

                                             } (<while (<ALARM_FILE
                                                        ;_$ chomp
                                                  ;_$,push @logfile
                                                           {

                                                      clean up #

                                                  ;(close(ALARM_FILE
                                                   ;(unlink($tmpFile

                                      ,Open temp file to write alert data into #

                                                  ;("open(OUT,">$TempIDSFile

                                               split XML output into fields #

                                                ;(oneline = join('',@logfile$
                                               ;oneline =~ s/\<\/events\>//g$
                                   ;oneline =~ s/\<\/evAlert\>/\<\/evAlert\>,/g$
                                                ;(items = split(/,/,$oneline@

If you want to see the actual database query result in #
                                               the email, un-comment out the
                                      :(line below (useful for troubleshooting #
                                                   ;("print(OUT "$oneline\n #

                                             Loop until there's no more alerts #

                                                        } (foreach (@items

                                            } (/<\if (m/\<hostId\>(.*)\<\/hostId
                                                        ;hostid = $1$
                                                               {

                                                  } (/"(?*.)"=if (m/severity
                                                        ;sev = $1$
                                                               {

                                            } (/<\if (m/Zone\=".*"\>(.*)\<\/time
                                                          ;t = $1$
                                                } (/({if ($t =~ m/(.*)(\d{9
 sec,$min,$hour,$mday,$mon,$year,$wday,$yday,$isdst) =$)
                                                        ;(localtime($1

 .(Year is reported from 1900 onwards (eg. 2003 is 103 #
                                             ;year = $year + 1900$

Months start at 0 (January = 0, February = 1, etc), so #
                                                       .add 1
                                                ;mon = $mon + 1$

                                             ;(mon = add_zero ($mon$
                                           ;(mday = add_zero ($mday$
                                           ;(hour = add_zero ($hour$
                                             ;(min = add_zero ($min$
                                             ;(sec = add_zero ($sec$
                                                               {
                                                               {
```

```
                                } (/"(?*.)"=if (m/sigName
                                            ;SigName = $1$
                                                        {

                                  } (/"(?*.)"=if (m/sigId
                                              ;SigID = $1$
                                                        {

                                } (/"(?*.)"=if (m/subSigId
                                            ;SubSig = $1$
                                                        {

          ;"attackerstring = "\<attacker.*\<\/attacker$
 ((if ($attackerstring = find_addresses ($attackerstring
                                                        }
                                                        {

                ;"victimstring = "\<victim.*\<\/victim$
  } ((if ($victimstring = find_addresses ($victimstring
                                                        {

       } (/<\if (m/\<alertDetails\>(.*)\<\/alertDetails
                                     ;AlertDetails = $1$
                                                        {

                                            ;() = actions@
                 } (/<\if (m/\<actions\>(.*)\<\/actions
                                          ;rawaction = $1$
          } (/>\(?*.)<\(?*while ($rawaction =~ m/\<(\w
                                          ;'$ = rawaction$
                                       } ("if ($2 eq "true
                                         ;push @actions,$1
                                                        {
                                                        {
                                        } (if (@actions
                    ;(actiontaken = join(', ',@actions$
                                                        {
                                                        {
                                                 } else
                                    ;"actiontaken = "None$
                                                        {

    Now write your email notification message. You're ##
                                    writing the following into
the temporary file for the moment, but this will then ##
                                              .be emailed
                                                       ##
Again, make sure you escape special characters with a ##
                               backslash (note the : between
                               .(the SigID and the SubSig ##
                                                       ##
    Put your VMS servers IP address in the NSDB: line ##
                                    below to get a direct link
          .to the signature details within the email ##

  print(OUT "\n$hostid reported a $sev severity alert at
                  ;("$hour:$min:$sec on $mon/$mday/$year\n
 ;("print(OUT "Signature: $SigName \($SigID\:$SubSig\)\n
       print(OUT "Attacker: $attackerstring ---> Victim:
                                         ;("$victimstring\n
           ;("print(OUT "Alert details: $AlertDetails \n
            ;("print(OUT "Actions taken: $actiontaken \n
           print(OUT "NSDB: https\://<your VMS server IP
        ;("address>/vms/nsdb/html/expsig_$SigID.html\n\n
```

```
print(OUT "----------------------------------------
                                        ;("-------\n

                                                {

                                         ;(close(OUT

  Now call "blat" to send contents of the file in the ##
                                .body of an email message
    Blat is a freeware email program for WinNT/95, it ##
                                    comes with VMS in the
 BASE\CSCOpx\bin directory, make sure you install it$ ##
                                    :first by running
                                                      ##
    blat -install <SMTP server address> <source email ##
                                              <address
                                                      ##
       For more help on blat, just type "blat" at the ##
                 command prompt on your VMS system (make
        sure it's in your path (feel free to move the ##
                 executable to c:\winnt\system32 BEFORE
   you run the install, that'll make sure your system ##
                                  .(can always find it

     system ("blat \"$TempIDSFile\" -t \"$EmailRcpt\" -s
                                  ;(""\\"Received IDS alert
```

نص مستشعر x.5

أستخدم هذا البرنامج النصي لأجهزة إستشعار الإصدار x.5.

أجهزة الاستشعار x.5

```
                                        usr/bin/perl/!#
                                       ;use Time::Local

**************************************************#
                                       ***************
                                                      #
                          FILE NAME   : emailalertv5.pl  #
                                                      #
DESCRIPTION : This file is a perl script that will be  #
                                           executed as an
       action when an IDS-MC Event Rule               #
                                  triggers, and will send an
    email to $EmailRcpt with additional               #
                                alert parameters (similar to
  the functionality available with CSPM               #
                                           (notifications
                                                      #
  NOTE: this script only works with 5.x               #
                                              .sensors
                                                      #
 NOTES       : This script takes the ${Query} keyword  #
                                                 from the
    triggered rule, extracts the set of               #
                                        alarms that caused
the rule to trigger.  It then reads the               #
                                         last alarm of
  this set, parses the individual alarm               #
                                          fields, and
```

```
  calls the legacy script with the same                #
                                                set of command
               .line arguments as CSPM                 #
                                                       #
   The calling sequence of this script                 #
                                     :must be of the form
                                                       #
             "{emailalert.pl "${Query                  #
                                                       #
                                   :Where              #
                                                       #
    Query}"  - this is the query}$"                    #
                                      keyword dynamically
   output by the rule                                  #
                                        .when it triggers
 It MUST be wrapped in                                 #
                                            double quotes
 when specifying it in                                 #
                                            the Arguments
       box on the Rule                                 #
                                          .Actions panel
                                                       #
                                                       #
**************************************************#
                                           ***********
                                                      ##
  The following are the only two variables that need ##
                       changing.  $TempIDSFile can be any
 filename (doesn't have to exist), just make sure the ##
                                directory that you specify
    exists.  Make sure to use 2 backslashes for each ##
                         directory, the first backslash is
        so the Perl interpretor doesn't error on the ##
                                                .pathname
                                                      ##
EmailRcpt is the person that is going to receive the$ ##
                                email notifications.  Also
      make sure you escape the @ symbol by putting a ##
                      backslash in front of it, otherwise
                       .you'll get a Perl syntax error ##
                                                      ##

              ;"TempIDSFile = "c:\\temp\\idsalert.txt$
                   ;"EmailRcpt = "gfullage\@cisco.com$

   subroutine to add leading 0's to any date variable #
                                      .that's less than 10
                                            } sub add_zero
                                       ;_@ = (my ($var
                                     } (if ($var < 10
                                  var = "0" .$var$
                                                     {
                                       ;return $var
                                                    {

 subroutine to find one or more IP addresses within an #
                             XML tag (we can have multiple
          .(victims and/or attackers in one alert now #
                                      } sub find_addresses
                                         ;_@ = (my ($var
                                       ;() = my @addresses
                                            } (/if (m/$var
                                            ;&$ = raw$
        } (/{while ($raw =~ m/(\d{1,3}\.){3}\d{1,3
```

```
                            ;&$,push @addresses
                                     ;'$ = raw$
                                                {
                        ;(var = join(', ',@addresses$
                                      ;return $var
                                                  {
                                                    {

                                pull out command line arg #

                                   ;[whereClause = $ARGV[0$

     extract all the alarms matching search expression #

                                   ;"tmpFile = "alarms.out$

     .Extract the XML alert/event  out of the database #

                 system("IdsAlarms -os -s\"$whereClause\" -
                                        ;(""\f\"$tmpFile

                                open matching alarm output #

                          } ((if (!open(ALARM_FILE, $tmpFile
                        ;"print "Could not open $tmpFile\n
                                             ;exit -1
                                                  {

                                      read to last line #

                                   } (<while (<ALARM_FILE
                                            ;_$ chomp
                                     ;_$,push @logfile
                                                  {

                                               clean up #

                                    ;(close(ALARM_FILE
                                     ;(unlink($tmpFile

                    ,Open temp file to write alert data into #

                                   ;("open(OUT,">$TempIDSFile

                                split XML output into fields #

                                   ;(oneline = join('',@logfile$
                             ;oneline =~ s/\<\/sd\:events\>//g$
                                                 oneline =~$
                ;s/\<\/sd\:evIdsAlert\>/\<\/sd\:evIdsAlert\>,/g
                                   ;(items = split(/,/,$oneline@

If you want to see the actual database query result in #
                                   the email, un-comment out the
                    :(line below (useful for troubleshooting #
                                      ;("print(OUT "$oneline\n #

                             Loop until there's no more alerts #

                                        } (foreach (@items
                             } (/<\unless ($_ =~ /\<\/env\:Body

                } (/<\if (m/\<sd\:hostId\>(.*)\<\/sd\:hostId
                                      ;hostid = $1$
```

```
                                                    {

                            } (/"(?*.)"=if (m/severity
                                                ;sev = $1$
                                                    {

            } (/<\if (m/Zone\=".*"\>(.*)\<\/sd\:time
                                                ;t = $1$
                        } (/({if ($t =~ m/(.*)(\d{9

 sec,$min,$hour,$mday,$mon,$year,$wday,$yday,$isdst) =$)
                                                ;(localtime($1

  Year is reported from 1900 onwards (eg. 2003 #
                                                    .(is 103
                                ;year = $year + 1900$

 Months start at 0 (January = 0, February = 1, #
                                                .etc), so add 1
                                    ;mon = $mon + 1$

                              ;(mon = add_zero ($mon$
                                  ;(mday = add_zero ($mday$
                                  ;(hour = add_zero ($hour$
                                    ;(min = add_zero ($min$
                                    ;(sec = add_zero ($sec$
                                                     {
                                                       {

                        } (/"(?*.)"=if (m/description
                                        ;SigName = $1$
                                                    {

                                 } (/"(?*.)"=if (m/\ id
                                          ;SigID = $1$
                                                    {

  } (/<\if (m/\<cid\:subsigId\>(.*)\<\/cid\:subsigId
                                        ;SubSig = $1$
                                                    {

                                                   if
\(m/\<cid\:riskRatingValue\>(.*)\<\/cid\:riskRatingValue
                                                    } (/<
                                                ;RR = $1$
                                                    {

} (/<\if (m/\<cid\:interface\>(.*)\<\/cid\:interface
                                            ;Intf = $1$
                                                    {

                                       attackerstring =$
                            ;""\<sd\:attacker.*\<\/sd\:attacker
                    if ($attackerstring = find_addresses
                                       } (((($attackerstring
                                                    {

     ;"victimstring = "\<sd\:target.*\<\/sd\:target$
 ((if ($victimstring = find_addresses ($victimstring
                                                         }
                                                    {

                                                   if
 } (/<\(m/\<cid\:alertDetails\>(.*)\<\/cid\:alertDetails
```

```
                                ;AlertDetails = $1$
                                                    {

                                          ;() = actions@
     } (/<\if (m/\<sd\:actions\>(.*)\<\/sd\:actions
                                       ;rawaction = $1$
 } (/>\(?*.)<\(?*while ($rawaction =~ m/\<\w*?:(\w
                                ;'$ = rawaction$

                             } ("if ($2 eq "true
                            ;push @actions,$1
                                                 {
                                                    {
                                    } (if (@actions
            ;(actiontaken = join(', ',@actions$
                                                    {
                                                      {
                                                  } else
                          ;"actiontaken = "None$
                                                    {

    Now write your email notification message. You're ##
                                  writing the following into
the temporary file for the moment, but this will then ##
                                             .be emailed
                                                        ##
Again, make sure you escape special characters with a ##
                              backslash (note the : between
                              .(the SigID and the SubSig ##
                                                        ##
    Put your VMS servers IP address in the NSDB: line ##
                                  below to get a direct link
           .to the signature details within the email ##

 print(OUT "\n$hostid reported a $sev severity alert
             ;("at $hour:$min:$sec on $mon/$mday/$year\n
                        print(OUT "Signature: $SigName
                                    ;("\($SigID\:$SubSig\)\n
 print(OUT "Attacker: $attackerstring  --->  Victim:
                                          ;("$victimstring\n
       ;("print(OUT "Alert details: $AlertDetails \n
;("print(OUT "Risk Rating: $RR,  Interface: $Intf \n
        ;("print(OUT "Actions taken: $actiontaken \n
                        print(OUT "NSDB: https\://sec-
              ;("srv/vms/nsdb/html/expsig_$SigID.html\n\n
print(OUT "----------------------------------------
                                       ;("----------\n

                                                    {
                                                       {

                                                ;(close(OUT

  Now call "blat" to send contents of the file in the ##
                                  .body of an email message
    Blat is a freeware email program for WinNT/95, it ##
                                       comes with VMS in the
 BASE\CSCOpx\bin directory, make sure you install it$ ##
                                          :first by running
                                                        ##
blat -install <SMTP server address> <source email     ##
                                                 <address
                                                        ##
       For more help on blat, just type "blat" at the ##
```

```
command prompt on your VMS system (make
sure it's in your path (feel free to move the ##
executable to c:\winnt\system32 BEFORE
you run the install, that'll make sure your system ##
.(can always find it

system ("blat \"$TempIDSFile\" -t \"$EmailRcpt\" -s
;(""\\"Received IDS alert
```

التحقق من الصحة

لا يوجد حاليًا إجراء للتحقق من صحة هذا التكوين.

استكشاف الأخطاء وإصلاحها

اتبع هذه التعليمات لاستكشاف أخطاء عملية التكوين لديك وإصلاحها.

1. قم بتشغيل هذا الأمر من موجه أوامر للتحقق من أن الخيار يعمل بشكل صحيح:

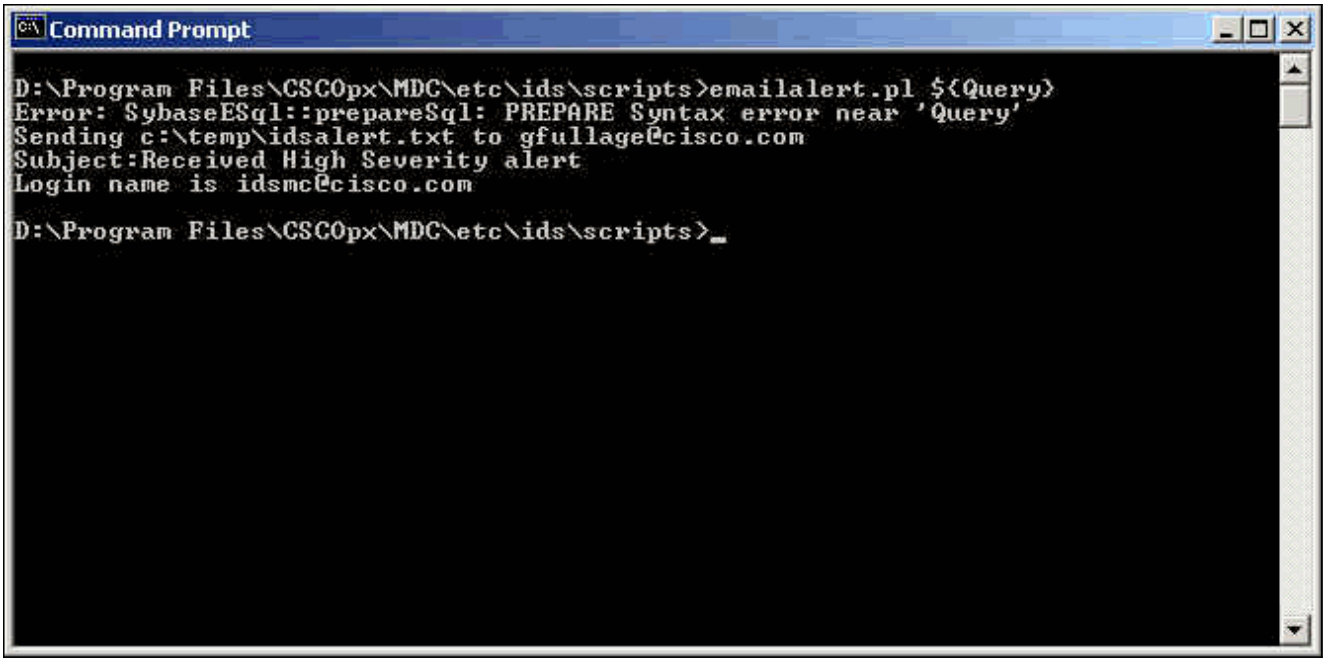
   blat

   <filename> هو المسار الكامل لأي ملف نصي على نظام VMS. إذا كان المستخدم الذي يتم توجيه البرنامج النصي للبريد الإلكتروني إليه يتلقى هذا الملف في نص رسالة بريد إلكتروني، فهذا يعني أنك تعرف أن Blat يعمل.
2. إذا لم يتم تلقي أي بريد إلكتروني بعد تشغيل تنبيه، حاول تشغيل البرنامج النصي ل Perl من نافذة مطالبة الأوامر.يسلط ذلك الضوء على أي مشاكل متعلقة بالمسار أو حرف Perl. للقيام بذلك، افتح موجه أوامر وأدخل:

   cd Program Files/CSCOpx/MDC/etc/ids/scripts<

   {emailalert.pl ${Query<

   من المحتمل أن تتلقى خطأ Sybase، مماثل لهذا المثال. ويرجع هذا إلى حقيقة أن المعلمة {Query}$ التي تمر بها لا تحتوي على معلومات بالفعل، بخلاف عندما تمر من "مراقبة الأمان".



بخلاف رؤية هذا الخطأ، يتم تشغيل البرنامج النصي بشكل صحيح وإرسال بريد إلكتروني. أية معلمات تنبيه ضمن نص البريد الإلكتروني تكون فارغة. إذا إستلمت أي أخطاء في المسار أو Perl، فإنه يلزم إصلاحها قبل إرسال بريد إلكتروني.

معلومات ذات صلة

- صفحة دعم منع التسلل الآمن من Cisco
- الدعم التقني والمستندات - Cisco Systems

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