How can I identify and address a mail loop situation on the ESA?

Document ID: 118522

Contributed by Tomki Camp and Enrico Werner, Cisco TAC Engineers.

Oct 09, 2014

Contents

Introduction

This document describes how to identify a mail loop on the Email Security Appliance (ESA).

Background Information

Mail Loops can be indicated by messages with the same Message−ID that were injected more than 3 times. Mail Loops can cause symptoms of High CPU, slow delivery and overall performance issues. Normally message IDs injected more than once would indicate looping, but sometimes they are injected more than once because of problems, or it could be a sloppy spammer who keeps injecting the same spam message with the same Message−ID.

More typically a mail loop is caused by an email infrastructure problem which sends the same message or set of messages racing around your network from mail server to mail server endlessly. While these messages can keep themselves entertained in this way for a very long time, it's not a good thing for either your network bandwidth or the ESA processing cost incurred.

Solution

Identifying a mail loop, if you suspect that this may be the problem, is usually pretty easy though you'll need to eye−ball it.
Log into the command−line interface (CLI) of the system and issue one of these commands, or both as you find best benefits you:

```
grep "Subject" mail_logs
grep "Message-ID" mail_logs
```

Particularly for the search on Message−ID, if you see recurring instances of exactly the same ID then you will know that you have a mail loop. However sometimes this is not enough, because one of the mail servers rallying back the same message might be helpfully changing or removing the Message−ID header. So if you don't get anything identifiable with the Message−ID check go ahead and try the Subject check.

Assuming that you managed to find the looping message by the Message−ID you will also want to find out other information about the message and its parent connection (ICID). Given the Message−ID and a MID in

the same log line you can perform:

```
grep −e "MessageID_I_found" −e "MID 123456" mail_logs
```

Given the resultant output there you can find the relevant ICID and DCID and perform:

```
grep −e "MessageID_I_found" −e "MID 123456" −e "ICID 1234567" −e "DCID 2345767" mail_logs
```

Now you should have the complete connection − message transaction and can see where it came from and where it was delivered to (if that has already occured). Once you have identified the looping message, your next step is to get a look at the message so that you can fix the problem. Without fixing the cause of the loop, it's likely that this message and others will continue to loop or that the problem will soon reoccur.

Create a message filter similar to this one:

```
loganddrop_looper:
if(header("Message−ID") == "MessageID_I_found") {
   archive("looper");
   drop();
}
```

Now commit that change and issue this command to check out the message:

```
tail looper
```

With the information you can gain about the remote system by looking at the mail logs, and other information you can gain by looking at the message itself, you should be able to determine where your problem is.

How can you prevent mail loops from occurring?

In complex environments this can be difficult − understanding how mail flows in your environment and how a new networking change, either on the ESA or to another device, will affect that traffic is key. One common cause of run−away mail loops is the removal of the Received header. The ESA will automatically detect and halt a mail loop when it sees 100 Received headers in a message, but the ESA does allow for the removal of this header, which often lead to a bad mail loop. Unless there is a *really* good reason to, please do not turn off the Received header, or cause them to be removed.

Below is a filter example which can help either prevent or fix a mail loop:

```
External_Loop_Count:
if (header("X−ExtLoop1")) {
  if (header("X−ExtLoopCount2")) {
    if (header("X−ExtLoopCount3")) {
      if (header("X−ExtLoopCount4")) {
        if (header("X−ExtLoopCount5")) {
          if (header("X−ExtLoopCount6")) {
            if (header("X−ExtLoopCount7")) {
              if (header("X−ExtLoopCount8")) {
                if (header("X−ExtLoopCount9")) {
                   notify ('joe@example.com');
                   drop();
                }
              else {insert−header("X−ExtLoopCount9", "from
                   $RemoteIP");}}
            else {insert−header("X−ExtLoopCount8", "from $RemoteIP");}}
          else {insert−header("X−ExtLoopCount7", "from $RemoteIP");}}
        else {insert−header("X−ExtLoopCount6", "from $RemoteIP");}}
      else {insert−header("X−ExtLoopCount5", "from $RemoteIP");}}
    else {insert−header("X−ExtLoopCount4", "from $RemoteIP");}}
```

```
  else {insert-header("X-ExtLoopCount3", "from $RemoteIP");}}
else {insert-header("X-ExtLoopCount2", "from $RemoteIP");}}
else {insert-header("X-ExtLoop1", "1"); }
```

Updated: Oct 09, 2014 Document ID: 118522