HOW TO

STORMSHIELD ENDPOINT SECURITY

QUARANTINING A WORKSTATION WITH THE SES FIREWALL

Product concerned: SES

Date: November 29, 2018

Reference: ses-en-how_to_quarantine_workstation_from_network

Table of contents

In the documentation, Stormshield Endpoint Security is referred to in its short form: SES.

Quarantining a workstation with the SES firewall

This document applies to versions 7.2.11 and higher of Stormshield Endpoint Security.

In it, you will see an example of how to quarantine a workstation from the network in the event of an attack, such as a heap spray attempt for example.

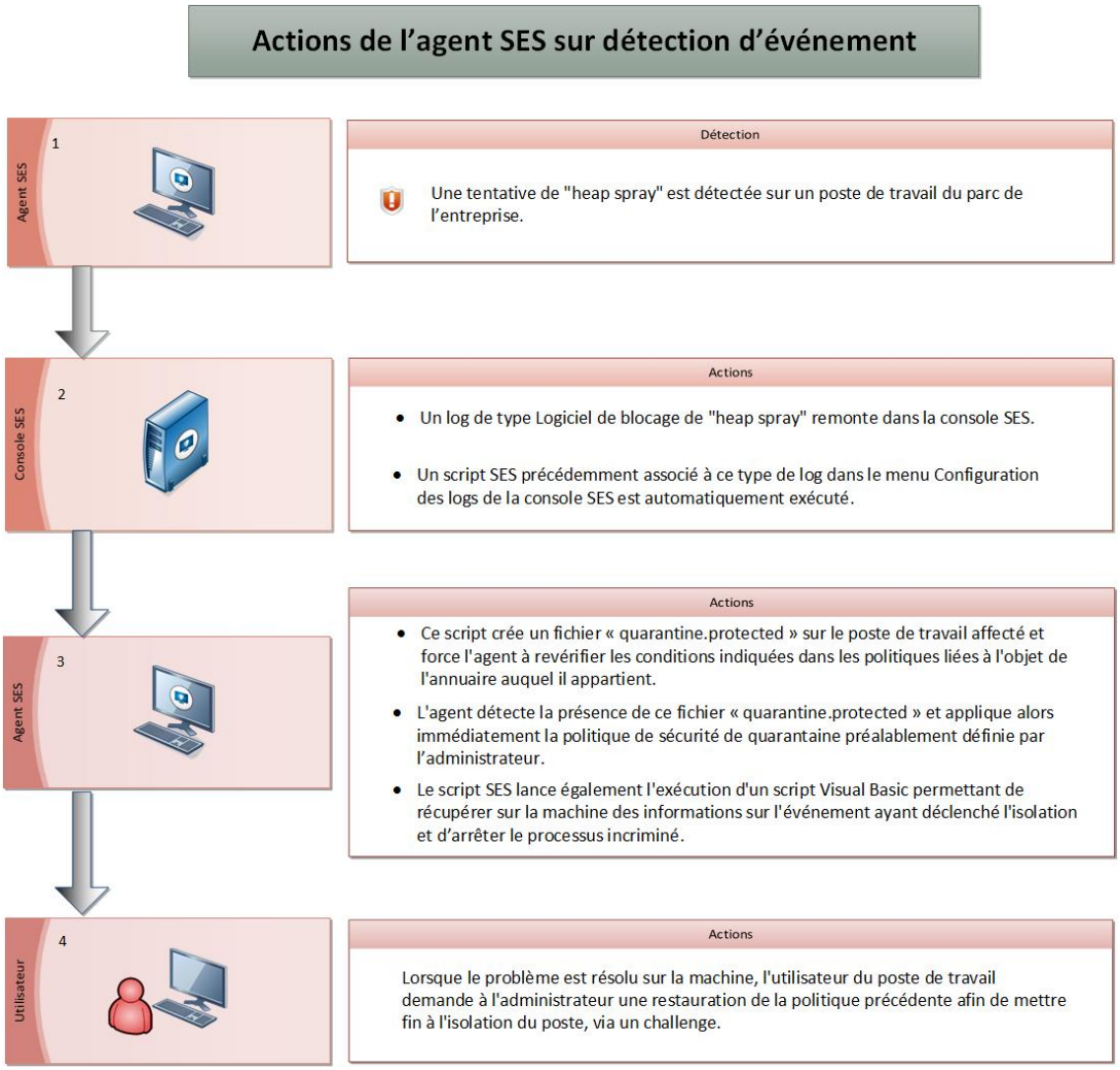
One of the ways to attain this objective is to use the automatic execution function that will perform a specified action whenever the SES agent detects a given event.

All images in this document are for representational purposes only, actual products may differ.

Understanding the mechanism executing an action when an event is detected

Prerequisites: to allow the agent to detect heap spray attempts, buffer overflow protection must be enabled in the security policy implemented on the workstation (*System behavior* tab in the **System behavior control** menu in the security policy).

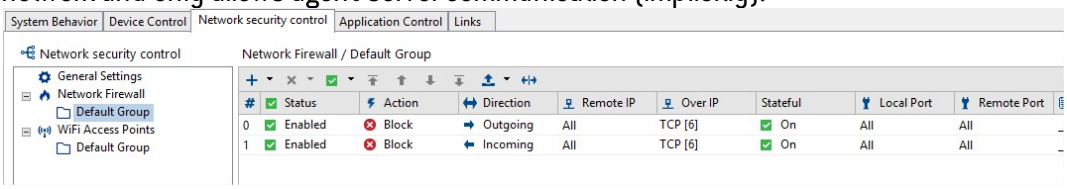
In brief, whenever the SES agent detects a given event, it will behave as follows:



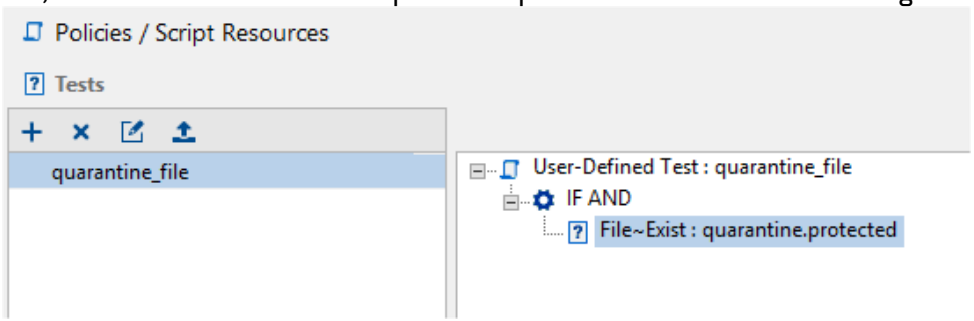
Refer to the rest of the document for the setup details of each stage. This serves as an example that you may adapt to similar situations.

Creating the security policy and test in script resources

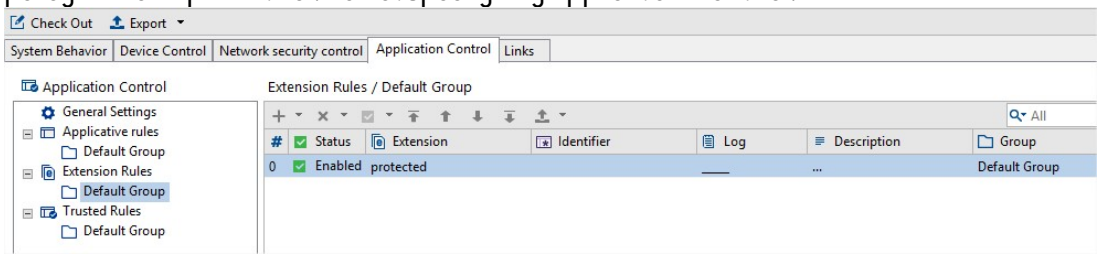
1. In the **Environment Manager > Policies > Security**, create a security policy named "quarantine". For example, the following policy isolates the workstation from the rest of the network and only allows agent/server communication (implicitly):



2. In the **Environment Manager > Policies > Script resources**, create a test named "quarantine_file", which will check whether a "quarantine.protected" file exists in the SES agent's folder.

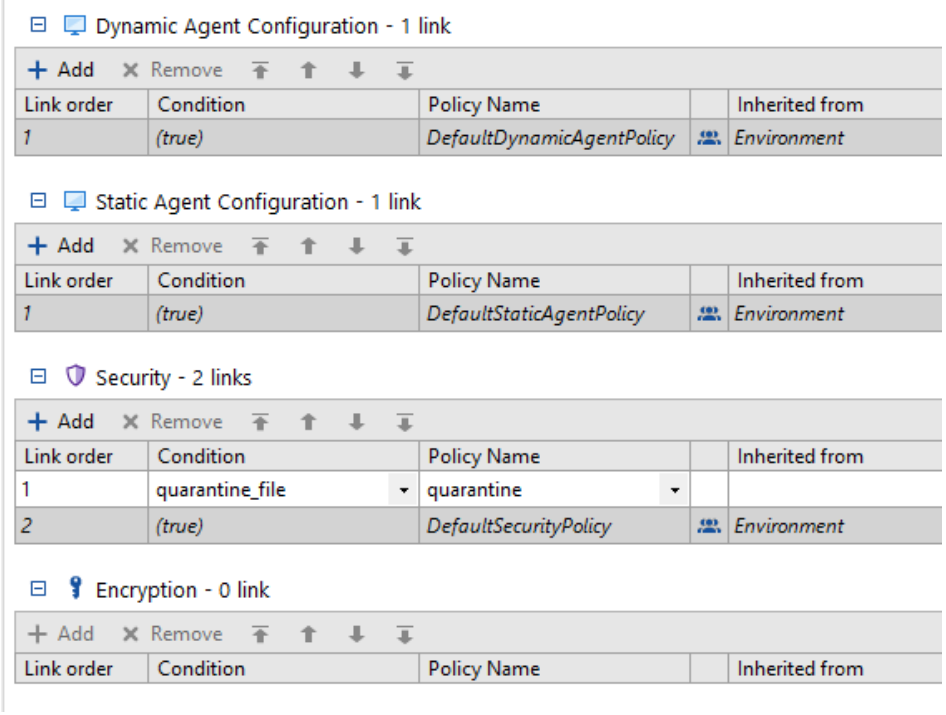


3. This "quarantine.protected" file must be protected so that the user or third-party applications would not be able to delete or modify it. To do so, create an extension rule in the security policy named "quarantine". Do not specify any application identifier.



Stormshield Endpoint Security will therefore protect all files with the ".protected" extension.

4. Go to your **Environment** and in the *Policies linked* tab, under **Security**, add "quarantine_file" as a condition and "quarantine" as a policy. Therefore, if the SES agent detects the "quarantine.protected" file on the workstation, it will immediately apply the workstation's quarantine policy.



Creating the Visual Basic script and downloading it on the workstation

In our example of a workstation quarantine, we use a Visual Basic script that makes it possible to retrieve information regarding the event that triggered the quarantine and to shut down the relevant process involved. It will be run by the SES workstation quarantine script, as described in the following section.

> **NOTE**
> The use of this VB script is not compulsory for the actual quarantining of the workstation, but it allows gaining control over the issue on the workstation as quickly as possible. It is only used here as an example.

SES environment variables relating to the event in question will automatically be communicated to the VB script or to any program run via the **Run process** menu in an SES script. For more information on such variables, refer to the section *Actions based on detection of events* in the *Activity monitoring > Log Manager* chapter of the *Stormshield Endpoint Security administration guide*.

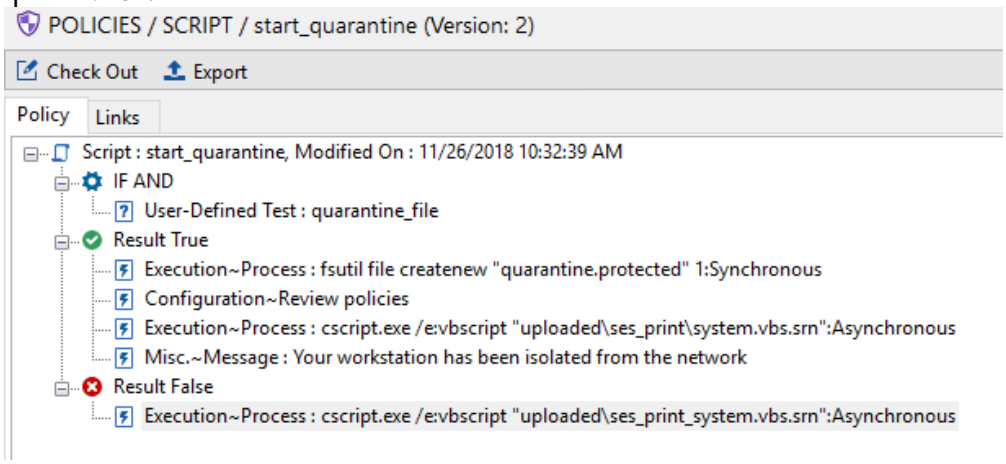
1. Create a VB script and name it "ses_print_system.vbs".
2. Copy the contents of the sample script given in the section Sample Visual Basic script. This sample makes it possible to retrieve information about the event in question based on SES environment variables and organizes them in a "quarantine.log" log file by adding the date and time before them. The process in question on the machine can also be shut down using this script.

3. In the SES console, add the VB script in **Environment Manager > Policies > File deployment** in order to transfer the file to the agents. For further information on file transfers, refer to the section *Transferring files towards the agents* in the chapter on *Scripts* in the *Stormshield Endpoint Security administration guide*.
4. Apply changes to the environment.

Creating quarantine start and end scripts

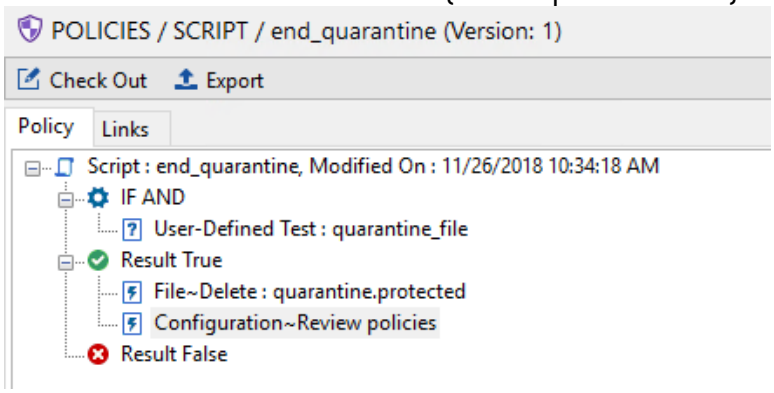
You will be creating two scripts in **Environment Manager > Policies > Script**: "start_quarantine" and "end_quarantine".

- The "start_quarantine" script allows the activation of the workstation quarantine. If the "quarantine.protected" file does not already exist on the workstation, the script:
  - will create the "quarantine.protected" file on the infected workstation in the agent's folder. The extension rule defined earlier will protect it from any attempts to modify or delete it.
  - will force the agent to immediately re-check the policies that it needs to apply according to the conditions defined in **Policies linked** ("Review policies" action) and therefore quarantine the workstation.
  - will run the Visual Basic script described in the previous section. Do not forget the VB script's "srn" extension, which Stormshield Endpoint Security adds during the file transfer.
  - will show the user a notification message to indicate that his workstation has been quarantined.



If the "quarantine.protected" file already exists on the workstation, this means that the workstation is already quarantined. The script will then run the Visual Basic script described in the previous section in order to retrieve information about the event that set off the quarantine.

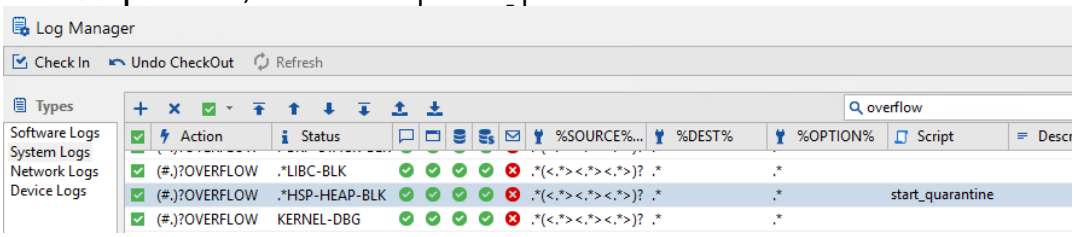
- The "end_quarantine" script makes it possible to stop quarantining the workstation once the issue has been resolved. The script will:
  - delete the "quarantine.protected" file on the infected workstation.
  - force the agent to immediately re-check the policies that it needs to apply according to the conditions defined in **Policies linked** ("Review policies" action).



Assigning quarantine start and end scripts

"start_quarantine" script

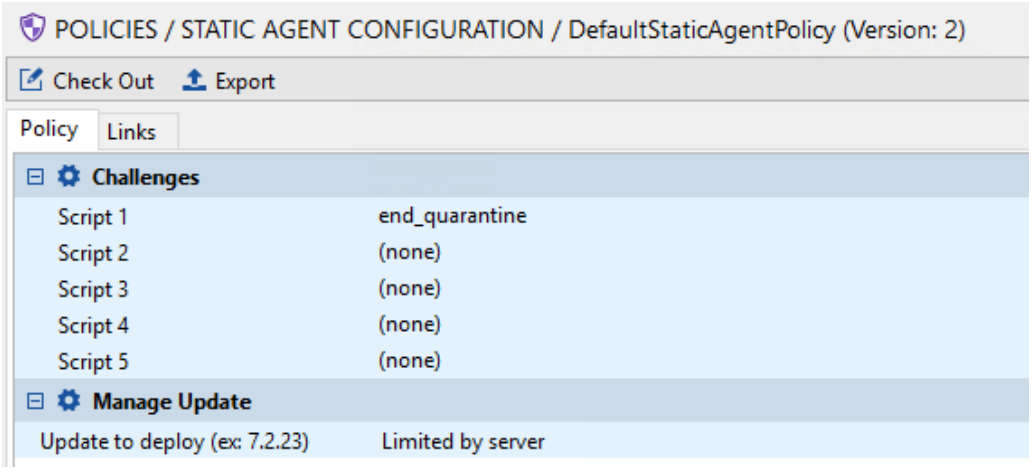
1. In **Environment Manager** > **Log Manager**, select **System logs**.
2. Locate the log HSP_HEAP_BLK (for the example we are describing here).
3. In the **Script** column, select the script "start_quarantine":



As such, every time a heap spray attempt is detected and a log is recorded in the SES console, the script will be run and the affected workstation will be immediately quarantined.

"end_quarantine" script

1. In **Environment Manager** > **Policies** > **Static agent configuration** > **Challenges**, select the "end_quarantine" script:



2. Once the issue has been resolved on the workstation, ask the user to provide an action code (right-click on the Stormshield icon on the workstation, **Other operations** > **Challenges**).
3. In the **Tools** > **Generate challenge** menu in the SES console, select the script in **Action type** and keep the default value, which is **Until reboot**.
4. Give the generated authorization code to the user and restart his workstation, which will be able to access the network again.

For more information on managing challenges, refer to the section *Editing a Static Agent Configuration Policy* in the chapter *SES agent configuration* in the *Stormshield Endpoint Security administration guide*.

Sample Visual Basic script

The sample VB script provided on the following page is used in the workstation quarantine procedure illustrated in this document. The use of this script, based on SES environment variables, is not compulsory. For more information on such variables, refer to the section *Actions based on detection of events* in the *Activity monitoring > Log Manager* chapter of the *Stormshield Endpoint Security administration guide*.

```
'print SES Environment Variables in a file
option explicit
Const ForWriting = 2
Const ForAppending = 8
Const vbsInterpreter = "cscript.exe"
Dim oFso
Dim oshell
Dim oShellEnv
Dim oNewFileLog
Dim regex, matchs, match
Dim processName

'initialize variables
set oFso = CreateObject("Scripting.FileSystemObject")
set oShell = WScript.CreateObject("WScript.Shell")
set oShellEnv = oShell.Environment("Process")
Set regex = New RegExp

With regex
               .Pattern = "[^\\]+\.exe"
               .Global = True
               .IgnoreCase = True
End With
'log file
Set oNewfilelog = oFso.OpenTextFile("quarantine.log", ForAppending, True)

'write data in the log file
oNewfilelog.WriteLine(vbCrLf & Date() & " " & TimeValue(Time) & vbCrLf)
oNewfilelog.WriteLine("Event generated by process : " & oShell.ExpandEnvironmentStrings("%SES_Source_Path%") &
vbCrLf)
oNewfilelog.WriteLine("Attack destination is : " & oShell.ExpandEnvironmentStrings("%SES_Destination_Path%") &
vbCrLf)

'extracting process name from SES_Source_Path

Set matchs = regex.Execute(oShell.ExpandEnvironmentStrings("%SES_Source_Path%"))
processName = ""
For Each match In matchs
If processName = "" Then
processName = match.Value
End If
Next
oNewfilelog.WriteLine("Will now try to kill " & processName & vbCrLf)
'kill process
oShell.Run "Taskkill /F /IM " & processName, , True
'clean up

Set oFso = Nothing
Set oShell = Nothing
Set oShellEnv = Nothing
Set oNewFileLog = Nothing
Set processName = Nothing
WScript.Quit
```

STORMSHIELD

documentation@stormshield.eu