

The background features several large, overlapping geometric shapes. In the top-left corner, there is a bright orange triangle. In the top-right, there is a grey trapezoid overlapping a black trapezoid. In the bottom-right, there is a large orange trapezoid overlapping a black trapezoid. A large, light grey trapezoid is positioned on the left side, partially overlapping the text.

JIGSAW24



Craig Hopkins

DevSecOps Engineer



Bob Bryden

DevSecOps Lead Architect

What is Endpoint Security Framework

- An API provided by Apple to allow monitoring of system events
- Not Kernel Extension
- Allow your app to receive notifications when an event has occurred or to authorise pending events
- Subscribe to Event Types (Notify/Auth)
- 100+ Notify Types
- 40+ Auth Types

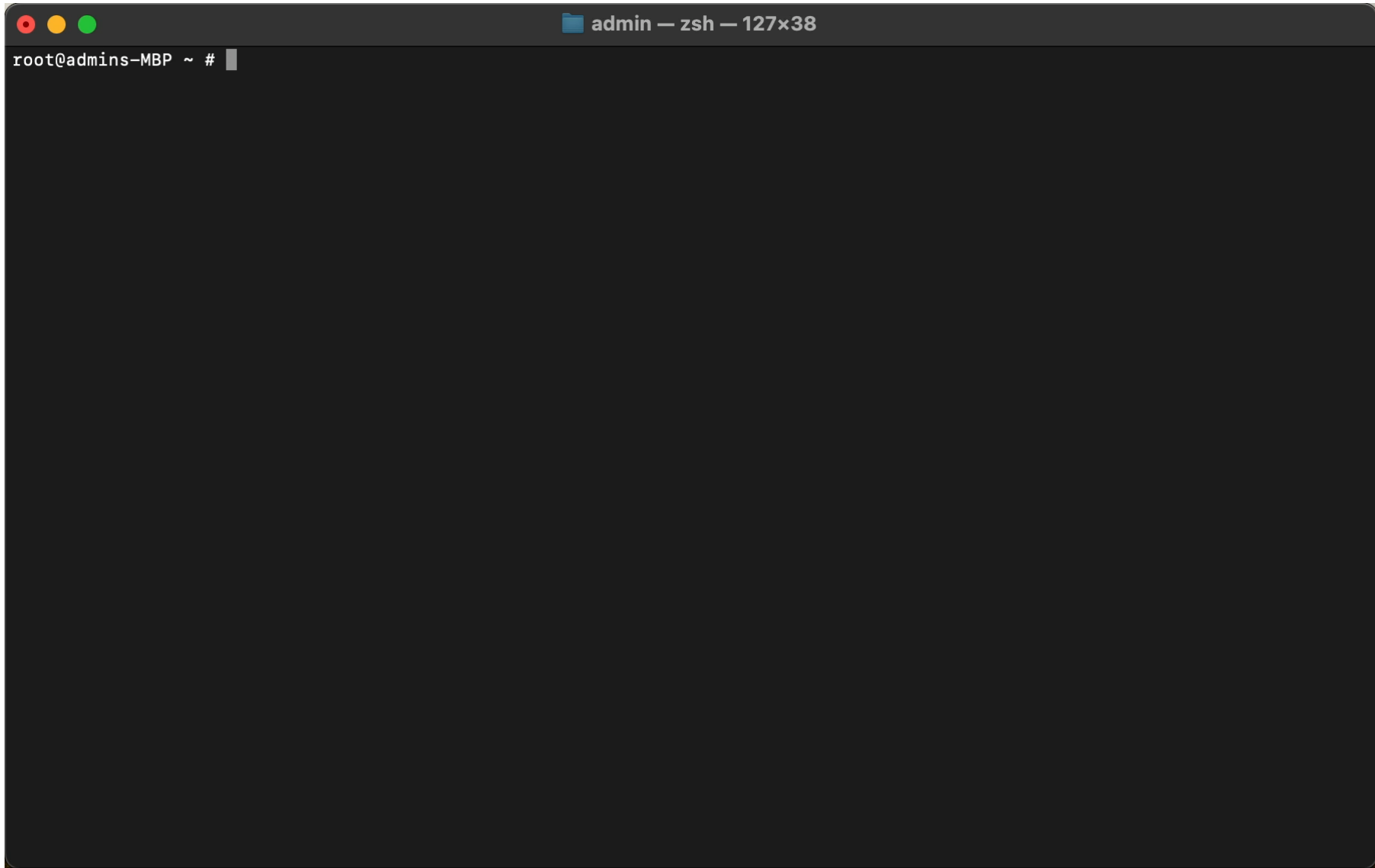


<https://developer.apple.com/documentation/endpointsecurity>

Why we wanted to use it

- Elevate24
- Centrally log events while elevated
- Ability to block some actions with more granular control than current MDM restrictions
- Needed a way to see events across multiple customers centrally

/usr/bin/eslogger open write

A terminal window with a dark background and light text. The title bar at the top reads "admin — zsh — 127x38". The prompt "root@admins-MBP ~ #" is visible at the top left of the terminal area, followed by a cursor. The rest of the terminal is empty.

```
admin — zsh — 127x38  
root@admins-MBP ~ #
```

```
{
  "event_type": "ES_EVENT_TYPE_NOTIFY_OPEN",
  "event": {
    "open": {
      "file": {
        "path": "/Users/craighopkins/Desktop/ShoppingList.rtf"
      }
    }
  },
  "global_seq_num": 1281,
  "time": "2024-05-01T13:46:34.779540157Z",
  "process": {
    "team_id": null,
    "executable": {
      "path": "/System/Library/CoreServices/Finder.app/Contents/MacOS/Finder"
    },
    "signing_id": "com.apple.finder",
    "original_ppid": 1,
    "ppid": 1,
    "start_time": "2024-05-01T09:35:33.079887Z",
    "is_platform_binary": true
  }
}
```

Requirements

- Apple Developer Account
- Requires com.apple.developer.endpoint-security.client entitlement granted to your developer account.
 - **Request via Apple Developer Account**
<https://developer.apple.com/contact/request/system-extension/>
 - **It is possible to test with SIP disabled - Don't do this!**
- A sample functional project is available here -
<https://github.com/Craighopkins12/EsfExample>
- The binary must have Full Disk Access



```
import EndpointSecurity

class ESFClient {
  var client: OpaquePointer?
  init() {
    // Create Client and event processor
    es_new_client(&client) { _, event in
      // This is where we process the event
      self.processEvent(event: event)
    }
    // Subscribe to Events
    let eventsToSubscribe = [ES_EVENT_TYPE_NOTIFY_OPEN, ES_EVENT_TYPE_AUTH_EXEC]
    es_subscribe(client!, eventsToSubscribe, UInt32(eventsToSubscribe.count))
  }
}
```



```
func processEvent(event: UnsafePointer<es_message_t>) {  
    // Depending on Event Type send to appropriate function  
    switch event.pointee.event_type {  
    case ES_EVENT_TYPE_NOTIFY_OPEN:  
        parseFileOpen(event: event)  
    case ES_EVENT_TYPE_AUTH_EXEC:  
        parseExecAuth(event: event)  
    default:  
        print ("Not Implemented")  
    }  
}
```

```
func parseFileOpen(event: UnsafePointer<es_message_t>) {  
    let filePath = convertString(event.pointee.event.open.file.pointee.path)  
    let processPath = convertString(event.pointee.process.pointee.executable.pointee.path)  
}
```

```
func convertString(_ token: es_string_token_t) -> String {  
    guard token.length > 0 else {  
        return ""  
    }  
    return String(cString: token.data)  
}
```

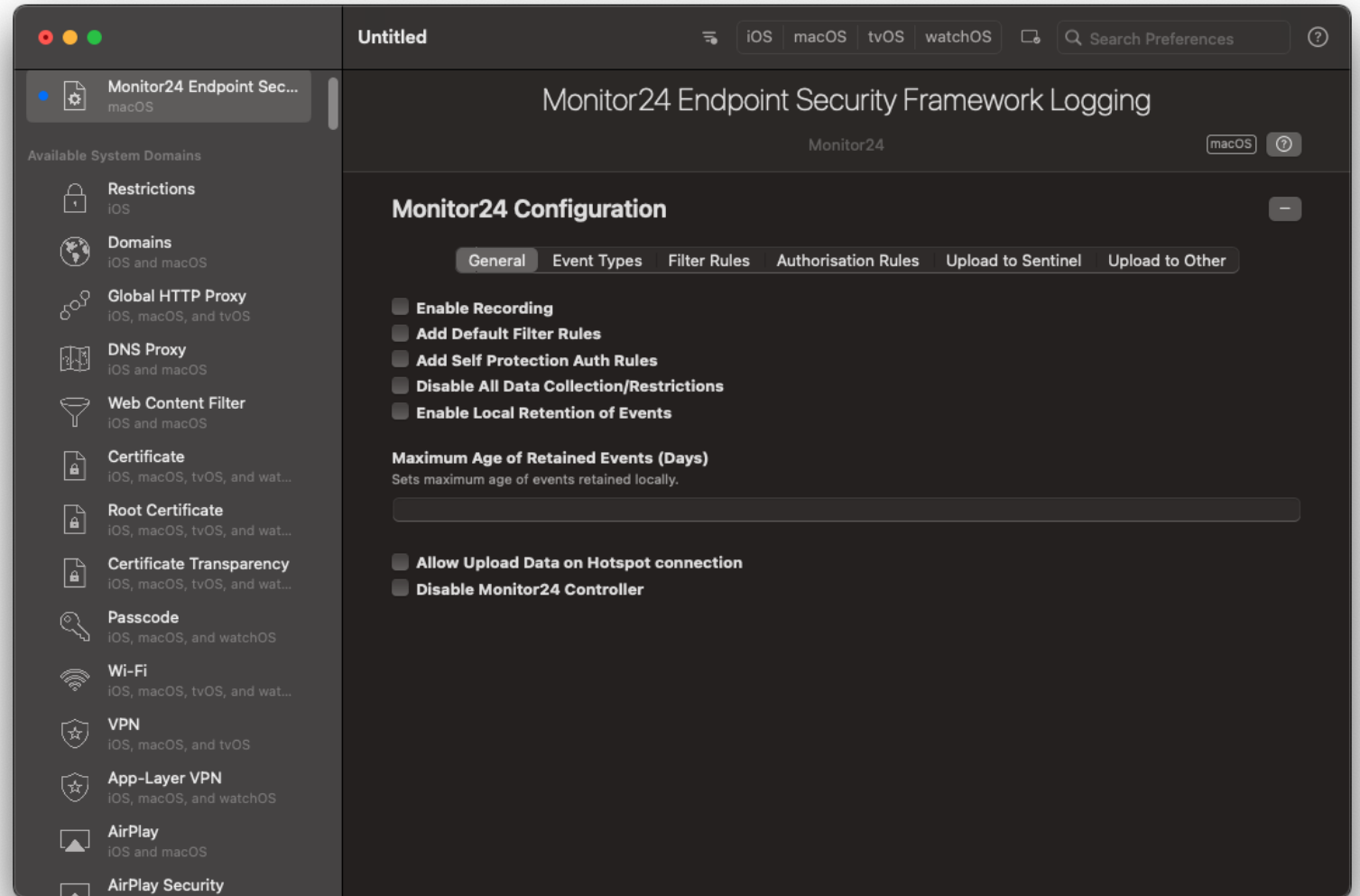
```
func parseExecAuth(event: UnsafePointer<es_message_t>) {
    var decision = ES_AUTH_RESULT_ALLOW
    let signingID = convertString(event.pointee.event.exec.target.pointee.signing_id)
    // Decide if action should be blocked
    if signingID == "com.unwantedApp.test" {
        decision = ES_AUTH_RESULT_DENY
    }
    // Get Arguments for launching the process
    let arguments = getProcessArguments(exec: event.pointee.event.exec)
    // Check the Arguments for blocking
    if (signingID == "com.apple.xpc.launchctl" && arguments.contains("bootout")) {
        decision = ES_AUTH_RESULT_DENY
    }
    // Apply the decision to the event
    es_respond_auth_result(client!, event, decision, false)
}
```

Monitor24

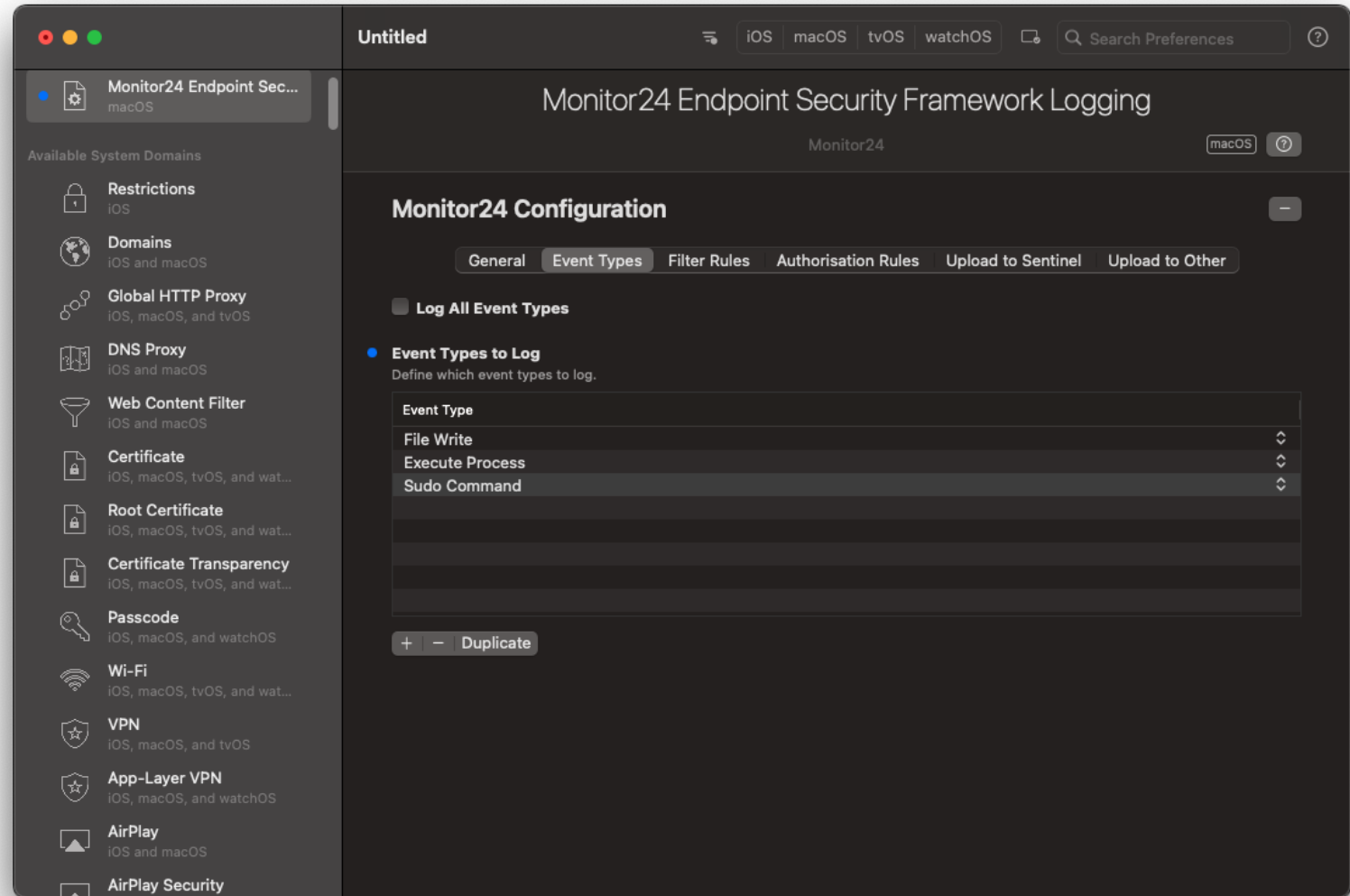
- Background Application
- Interacts with the Endpoint Security Framework
- Configurable via Configuration Profile
- Records events of selected type with filter options
- Authorisation Rules support blocking actions
- Events can be stored locally or uploaded to Logging Platform/SIEM
- Release as free tool on GitHub (Currently Beta)



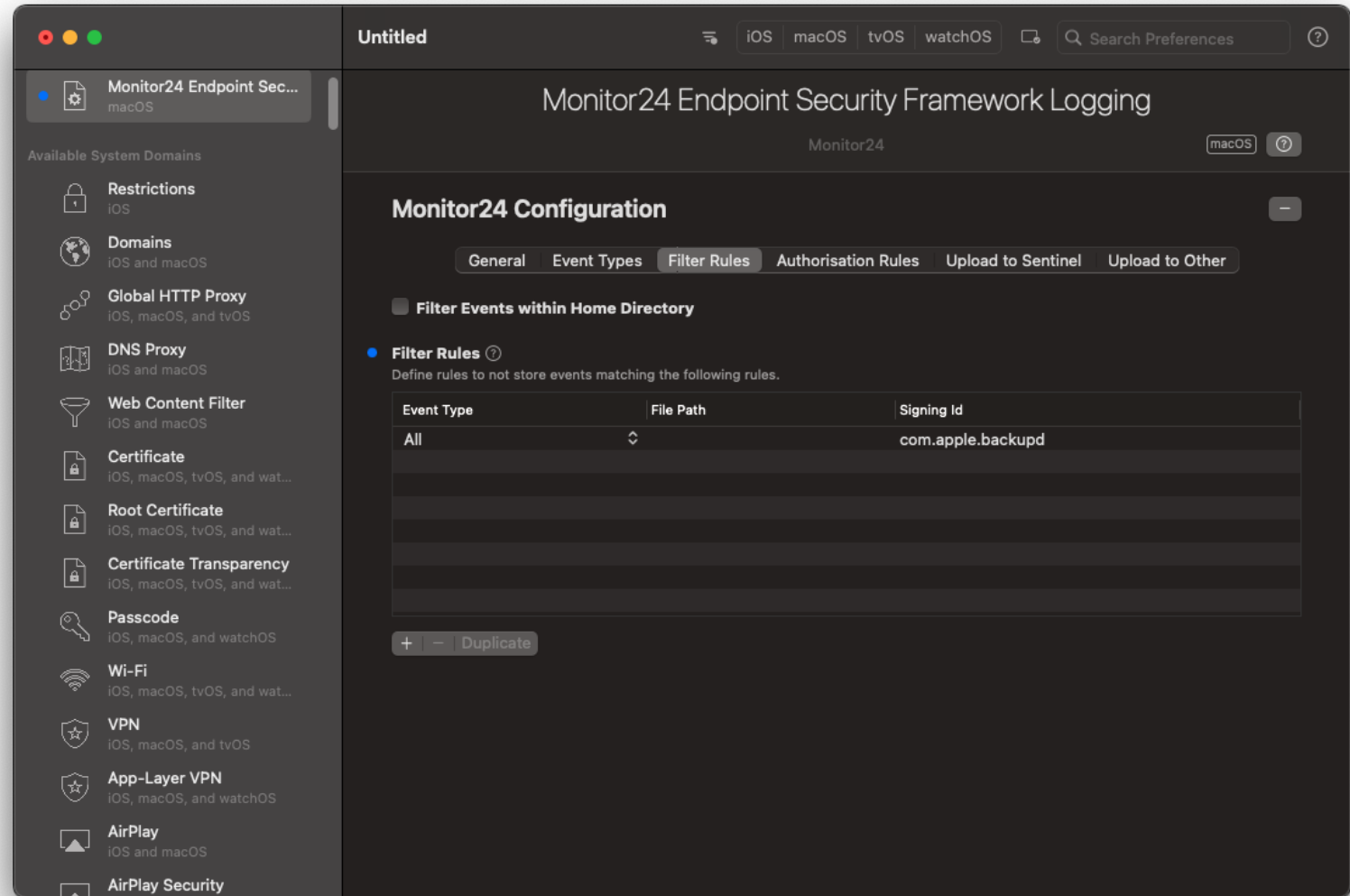
Monitor24 - Configuration



Monitor24 - Configuration



Monitor24 - Configuration



Monitor24 - Configuration

The screenshot shows the macOS System Preferences window for 'Monitor24 Endpoint Security Framework Logging'. The left sidebar lists various system domains, with 'Restrictions' selected. The main pane displays the configuration for 'File Operation Rules' and 'Execute Process Rules'.

File Operation Rules

Rules to restrict file operations. Allow rules can be used in conjunction with block rules to allow a process to execute under the specified conditions. File open operations can be restricted to read only access where required.

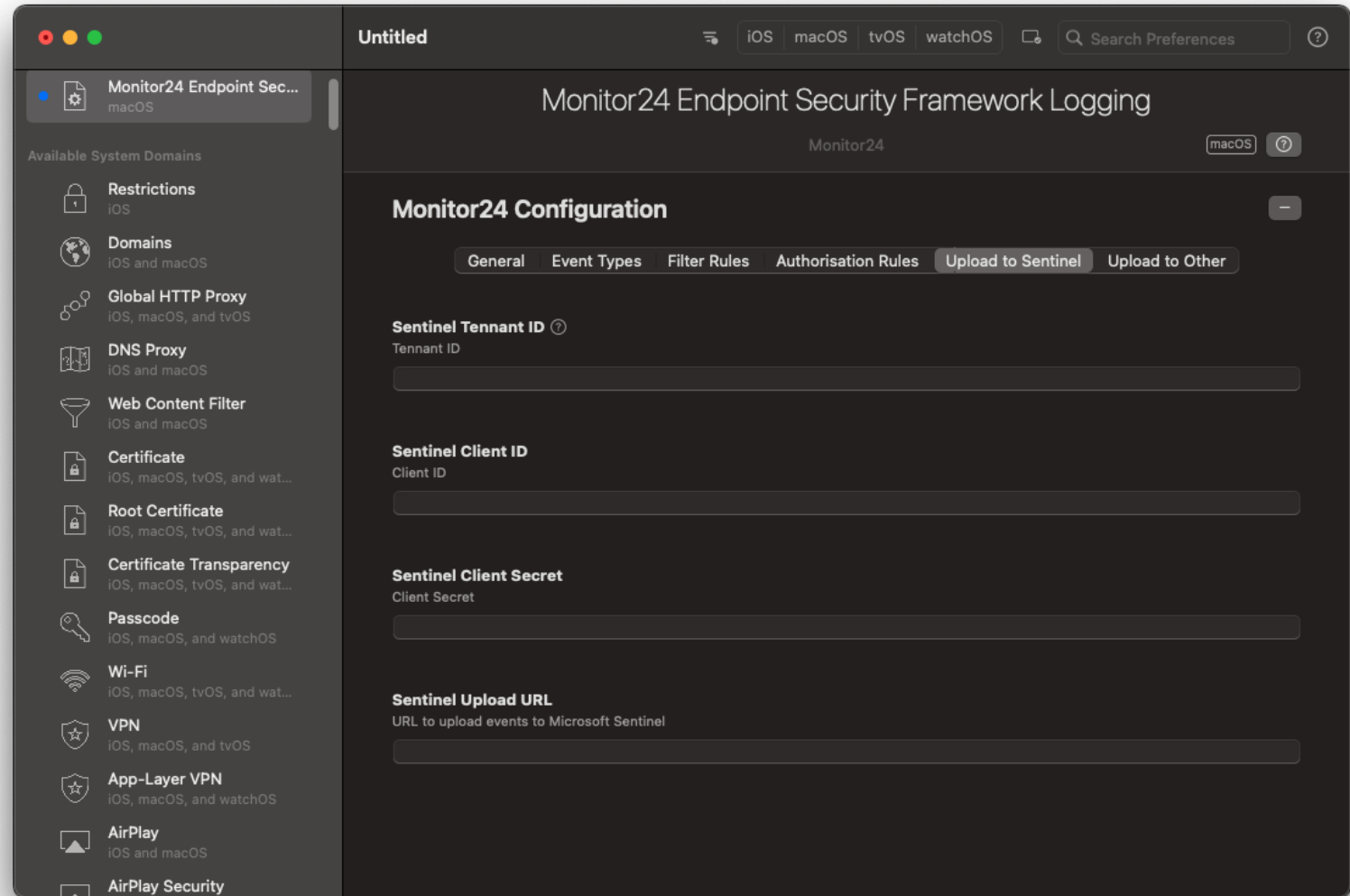
File Operation	Action	File Target Path	Signing ID of Process	Username
File Open	Read Only (Fil...	/ReadOnly.txt		
File Open	Block	/Secret.txt		
File Delete	Block	/Important.txt		

Execute Process Rules

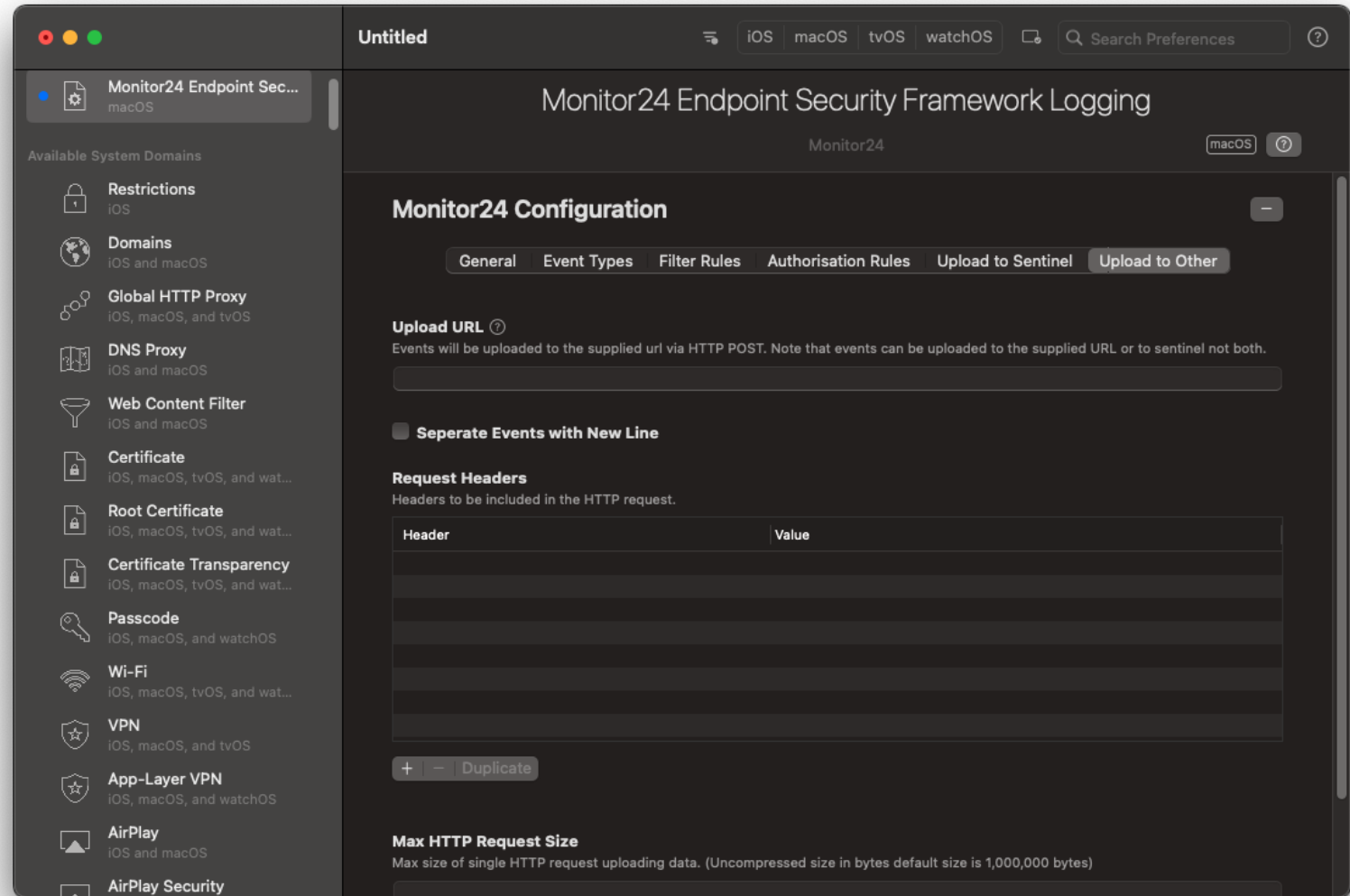
Rules to restrict execution of processes. Allow rules can be used in conjunction with block rules to allow a process to execute under the specified conditions. Multiple arguments can be specified by separating them with a comma.

Action	Process Path	Signing Id of the Process	SigningID	Team ID	Match Arguments Containing
Block		com.apple.xpc.launchd			com.jigsaw24.Monitor24
Block				XXXXXXX	

Monitor24 - Configuration



Monitor24 - Configuration



Start Monitor

Stop Monitor

Clear Databases

Get Events in DB

Get Events in Archive

Filter Events

TimeStamp

Event Type

Usern...

Process Path

Signing ID

Apple...

Target

TimeStamp	Event Type	Usern...	Process Path	Signing ID	Apple...	Target

Events in DB 42

Events in Archive 0



Monitor24 - Upload Data

^ Sudo Sessions

Sudo Sessions - 30 Days

Search

TimeGenerated	User	Serial_Number	eventtype	Command_Ran_As	Command_Ran
16/05/2024, 16:43:21.000	ldonnelly	HFXP2L44MD	Process:Auth:Exec	root	sudo echo <result>3</result>
16/05/2024, 16:43:20.000	ldonnelly	HFXP2L44MD	Process:Auth:Exec	root	sudo -u ldonnelly defaults -currentHost read
16/05/2024, 16:43:19.000	ldonnelly	HFXP2L44MD	Process:Auth:Exec	root	/usr/bin/sudo /usr/local/bin/sentinelctl versi
16/05/2024, 16:43:02.000	ldonnelly	HFXP2L44MD	Process:Auth:Exec	root	sudo jamf recon
16/05/2024, 16:39:42.000	ldonnelly	HFXP2L44MD	Process:Auth:Exec	root	sudo jamf policy
16/05/2024, 16:38:32.000	ldonnelly	HFXP2L44MD	Process:Auth:Exec	root	sudo jamf policy
15/05/2024, 16:06:52.000	craighopkins	GDXKYHXQGQ	Process:Auth:Exec	root	sudo -s
03/05/2024, 15:20:58.000	craighopkins	GDXKYHXQGQ	Process:Auth:Exec	root	sudo -s
03/05/2024, 15:20:55.000	craighopkins	GDXKYHXQGQ	Process:Auth:Exec	root	sudo 0s
02/05/2024, 17:41:40.000	craighopkins	GDXKYHXQGQ	Process:Auth:Exec	root	sudo -s
24/04/2024, 16:27:33.000	craighopkins	GDXKYHXQGQ	Process:Auth:Exec	root	sudo -s

Serial Number	Count
HFXP2L44MD	19
FVFDMSUAQ05F	7
GDXKYHXQGQ	5
FVEZ7IADLYWJ	2

Back to Elevate24



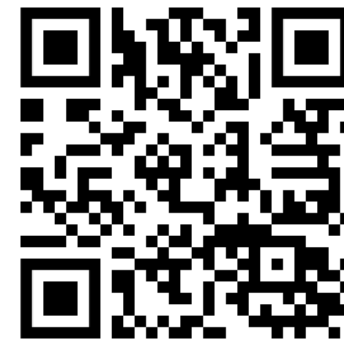
Thank you

Useful resources

- Apple documentation - <https://developer.apple.com/documentation/endpointsecurity>
- Red Canary Mac Monitor - <https://github.com/redcanaryco/mac-monitor>
- Crescendo - <https://github.com/SuprHackerSteve/Crescendo>



Example Code Project
[https://github.com/
Craighopkins12/EsfExample](https://github.com/Craighopkins12/EsfExample)



Monitor24 (beta)
[https://github.com/
Jigsaw24/Monitor24](https://github.com/Jigsaw24/Monitor24)