

Practical SOAR Examples from the Field: Part 2

SEC1372

Richard Hampshire
Matthew Bennett



Forward- looking statements

This presentation may contain forward-looking statements regarding future events, plans or the expected financial performance of our company, including our expectations regarding our products, technology, strategy, customers, markets, acquisitions and investments. These statements reflect management's current expectations, estimates and assumptions based on the information currently available to us. These forward-looking statements are not guarantees of future performance and involve significant risks, uncertainties and other factors that may cause our actual results, performance or achievements to be materially different from results, performance or achievements expressed or implied by the forward-looking statements contained in this presentation.

For additional information about factors that could cause actual results to differ materially from those described in the forward-looking statements made in this presentation, please refer to our periodic reports and other filings with the SEC, including the risk factors identified in our most recent quarterly reports on Form 10-Q and annual reports on Form 10-K, copies of which may be obtained by visiting the Splunk Investor Relations website at www.investors.splunk.com or the SEC's website at www.sec.gov. The forward-looking statements made in this presentation are made as of the time and date of this presentation. If reviewed after the initial presentation, even if made available by us, on our website or otherwise, it may not contain current or accurate information. We disclaim any obligation to update or revise any forward-looking statement based on new information, future events or otherwise, except as required by applicable law.

In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not be incorporated into any contract or other commitment. We undertake no obligation either to develop the features or functionalities described, in beta or in preview (used interchangeably), or to include any such feature or functionality in a future release.

Splunk, Splunk> and Turn Data Into Doing are trademarks and registered trademarks of Splunk Inc. in the United States and other countries. All other brand names, product names or trademarks belong to their respective owners.

© 2025 Splunk LLC. All rights reserved.

splunk>

.conf25

Overview

Session Breakdown and Goals

Breakdown

- Introductions
- SOAR Intro and Refresher
- Cases Studies
- Additional Tips
- Wrap Up / Questions

Goals

- Inspire you to confidently handle automation
- Show you more real-world playbooks from the field
- Leave you with more ideas, tips and tricks on creating your own playbooks

Introductions



Richard Hampshire

Security Architect - Professional Services
Splunk

rich@splunk.com





Matt Bennett

Co-Founder / Core Consultant
Hyperion3

matt@hyperion3.com.au



SOAR Intro / Refresher

Recap

SEC1579B - .conf24

Last year we covered:

- Best practice SOAR playbook implementation
- Playbook case studies from the field
- Bonus tips
- ... and more



<https://tinyurl.com/4r7afb9r>

Recap

SOAR

Implementation Methodology



.conf24 SEC1579B

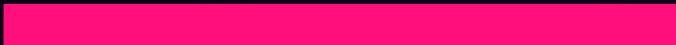
<https://tinyurl.com/4r7afb9r>



Recap

SOAR Integration - Where do the analysts work?

Many options, with no right or wrong way



Splunk SOAR



Splunk ES



3rd Party



Mission Control

.conf24 SEC1579B

<https://tinyurl.com/4r7afb9r>



Playbook Design Principles

Design Principles

What we've discovered

- Detailed / Low Level Designs used for traditional Splunk deployments do no work for SOAR
 - Automation is complex area and legacy design principals do not translate across
- Effort is better spent prototyping the solution and iterating / adjusting on the fly
 - Automation aligns better with a DevOps / CICD approach
- A high level design can still be used to set goals and establish guardrails
 - Provides the 'commander's intent'
- It's impossible to gather all requirements and permutations of the automation flow up front
 - Often the customer cannot provide this until they've seen the solution in action

Typical Legacy Approach



What's supposed to happen:

- Requirements are gathered and a design is created
- Customer reviews and if satisfied the design is approved and all playbook components are 'locked in'
- Playbook is developed in-line with the design
- Solution presented to customer, is tested, and signed off

Typical Legacy Approach



What usually actually happens..

- Process proceeds up until playbook development starts, during which:
 - Environment specifics are discovered that conflict with the design
 - New questions are raised that could not have been predicted previously
 - Additional requirements are raised from the customer
- Re-design is required, more time is needed, and often deadlock occurs
- And most importantly, more time is spent on the design than developing the playbook itself.

Best Practice Approach



How we've changed the approach

- Enhanced focus on discovery to draw out requirements, assumptions, and potential blockers
- High level design captures the general flow and overall intent - not every individual block / function
- Playbook developer then has freedom to innovate and develop the solution as long as it still reasonably meets the intent from the high level design.
- True up activity at the end uplifts documentation to reflect what was actually built

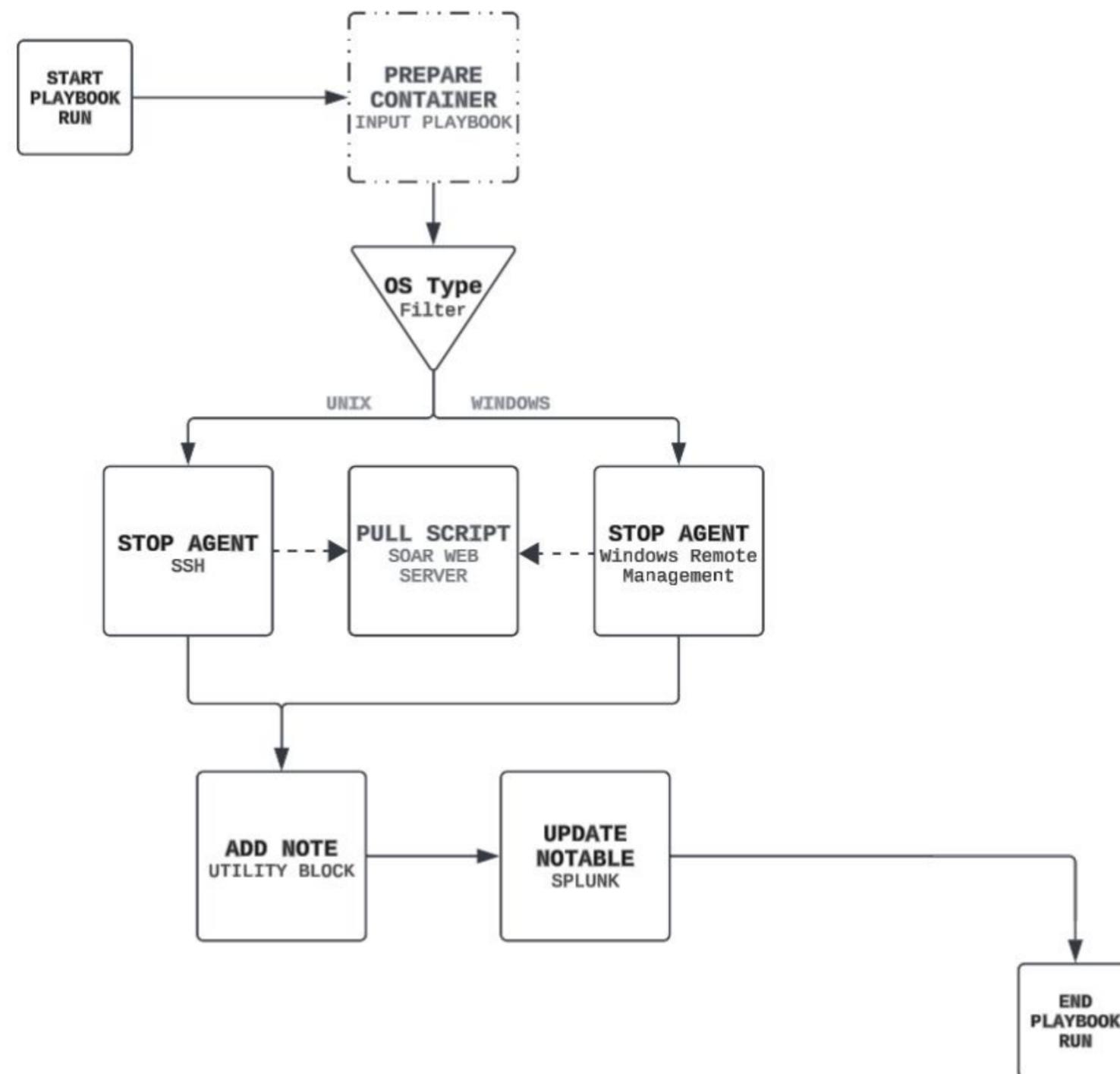
Case Study: Agent Management Playbook Design

Initial Design

Simple high level flow to determine the OS, stop the agent, add a note, and update the notable.

What more could you possibly need?

Required Apps:
SSH
Windows Remote Management
Splunk

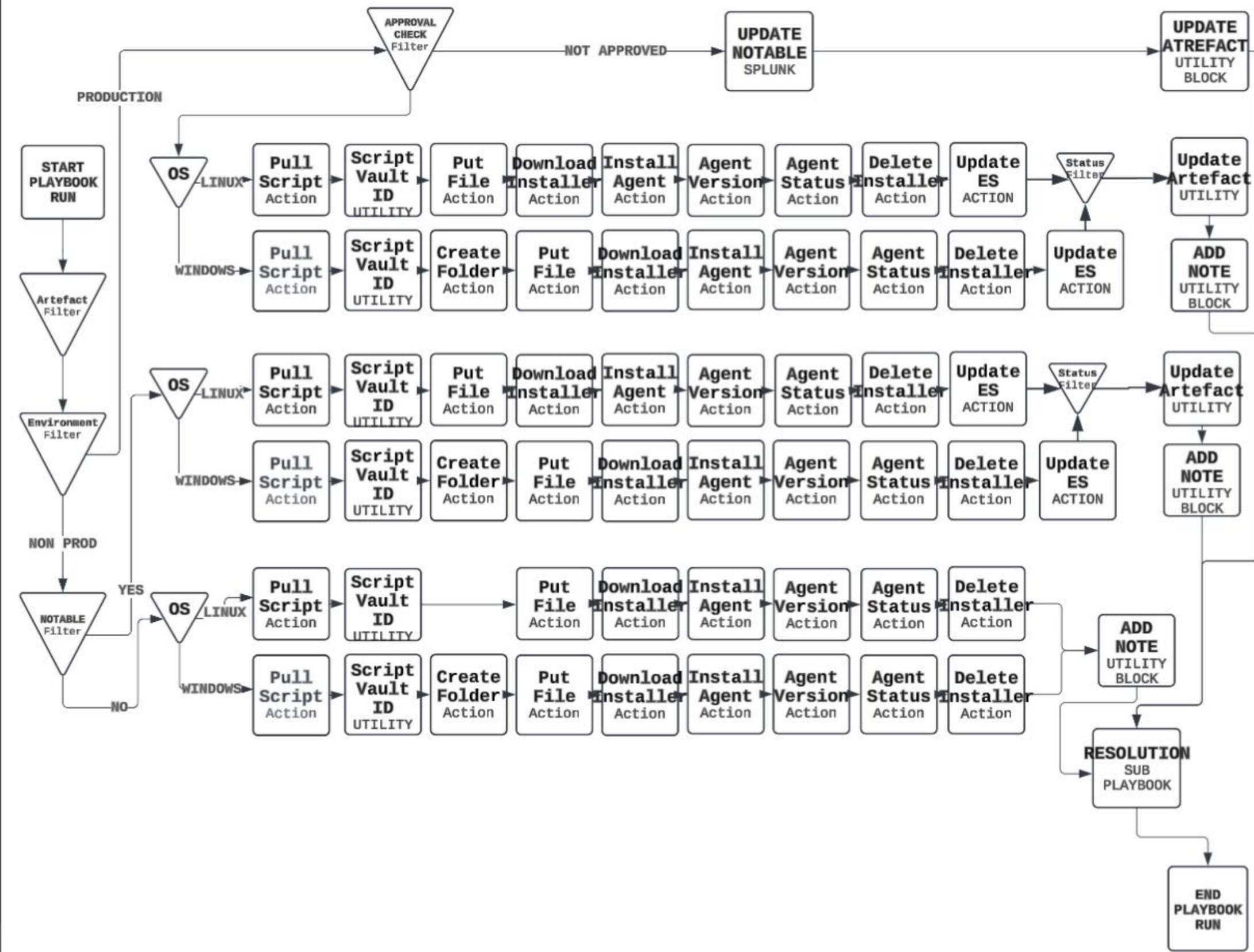


Final Design

Well...

Same playbook,
Same requirements,
Just a higher level of
detail unearthed once
development
commenced.

Required Apps:
SSH
Windows Remote Management
Splunk



Design Principles

Key Takeaways

- Legacy approach had been underway for over 24 months and completely stalled
 - Our new approach had customer design sign-off in < 1 week
- Employ high level design to capture the intent and establish automation goals / guardrails
- Low level / detailed designs are not practical to create in advance
 - Especially for complex automation - more time will be spent on design than build
- Perform a 'true-up' once the solution has been completed
 - Documentation as code will save time here

Playbook Case Studies

Custom ES Automation Workflow

Custom ES Automation Workflow

Using ES notable status changes to trigger automation

What and why?

- Requirement to selectively initiate automation under heavy constraints
 - Automation must be triggered by a user
 - User must be able to select some container and omit others
 - Automation 'requests' must be reviewed and approved before execution
 - Solution must provide different workflows for PROD and NON-PROD systems
 - PROD runs as per the above
 - NON-PROD hosts follow a different flow

Custom ES Automation Workflow

Using ES notable status changes to trigger automation

How?

- Correlation search creating notable events
- Custom notable statuses configured in ES
- Additional scheduled search monitoring for status change
- Send to SOAR adaptive response action

Analyst queue

Search findings & investigations

All time

Time Range: All time Clear All Save Apply

Zoom To Selection Zoom Out Deselect



4 selected Edit Assign to Me Add to Investigation

<input type="checkbox"/>	i	Title	Type	Status	Owner	Risk Object	Risk Score	Time	Disposition
<input checked="" type="checkbox"/>		Compliance Event	FINDING	New	unassigned	HYP3SVR655	1	Tue, 18 Mar 2025 10:50	Undetermined
<input checked="" type="checkbox"/>		Compliance Event	FINDING	New	unassigned	HYP3SVR542	1	Tue, 18 Mar 2025 10:50	Undetermined
<input checked="" type="checkbox"/>		Compliance Event	FINDING	New	unassigned	HYP3SVR204	1	Tue, 18 Mar 2025 10:45	Undetermined
<input checked="" type="checkbox"/>		Compliance Event	FINDING	New	unassigned	HYP3SVR193	1	Tue, 18 Mar 2025 10:45	Undetermined
<input type="checkbox"/>		Compliance Event	FINDING	New	unassigned	HYP3SVR687	1	Tue, 18 Mar 2025 10:40	Undetermined
<input type="checkbox"/>		Compliance Event	FINDING	New	unassigned	HYP3SVR609	1	Tue, 18 Mar 2025 10:40	Undetermined
<input type="checkbox"/>		Compliance Event	FINDING	New	unassigned	HYP3SVR400	1	Tue, 18 Mar 2025 10:35	Undetermined
<input type="checkbox"/>		Compliance Event	FINDING	New	unassigned	HYP3SVR627	1	Tue, 18 Mar 2025 10:30	Undetermined
<input type="checkbox"/>		Compliance Event	FINDING	New	unassigned	HYP3SVR16	1	Tue, 18 Mar 2025 10:30	Undetermined
<input type="checkbox"/>		Compliance Event	FINDING	New	unassigned	HYP3SVR17	1	Tue, 18 Mar 2025 10:25	Undetermined
<input type="checkbox"/>		Compliance Event	FINDING	New	unassigned	HYP3SVR638	1	Tue, 18 Mar 2025 10:25	Undetermined
<input type="checkbox"/>		Compliance Event	FINDING	New	unassigned	HYP3SVR727	1	Tue, 18 Mar 2025 10:25	Undetermined
<input type="checkbox"/>		Compliance Event	FINDING	New	unassigned	HYP3SVR161	1	Tue, 18 Mar 2025 10:20	Undetermined
<input type="checkbox"/>		Compliance Event	FINDING	New	unassigned	HYP3SVR171	1	Tue, 18 Mar 2025 10:20	Undetermined
<input type="checkbox"/>		Compliance Event	FINDING	New	unassigned	HYP3SVR602	1	Tue, 18 Mar 2025 10:20	Undetermined
<input type="checkbox"/>		Compliance Event	FINDING	New	unassigned	HYP3SVR979	1	Tue, 18 Mar 2025 10:10	Undetermined
<input type="checkbox"/>		Compliance Event	FINDING	New	unassigned	HYP3SVR445	1	Tue, 18 Mar 2025 10:10	Undetermined
<input type="checkbox"/>		Compliance Event	FINDING	New	unassigned	HYP3SVR592	1	Tue, 18 Mar 2025 10:10	Undetermined
<input type="checkbox"/>		Compliance Event	FINDING	New	unassigned	HYP3SVR91	1	Tue, 18 Mar 2025 10:05	Undetermined

Automatically detecting the status change

Required SPL

```
`notable`
| search status_label="Request Automation"
| eval rule_name=if(isnull(rule_name),source,rule_name)
| eval rule_title=if(isnull(rule_title),rule_name,rule_title)
| eval rule_description=if(isnull(rule_description),source,rule_description)
| eval security_domain=if(isnull(security_domain),source,security_domain)
| stats values(urgency) as urgency, values(status_label) as status_label,
values(status_description) as status_description, values(event_hash) as event_hash,
values(event_id) as event_id, values(orig_raw) as orig_raw, values(rule_name) as rule_name,
values(rule_id) as rule_id, values(status_logger) as status_logger count by owner
| table *
```

Automatically detecting the status change

Required SPL

```
`notable`
```

```
| search status_label="Request Automation"  
| eval rule_name=if(isnull(rule_name),source,rule_name)  
| eval rule_title=if(isnull(rule_title),rule_name,rule_title)  
| eval rule_description=if(isnull(rule_description),source,rule_description)  
| eval security_domain=if(isnull(security_domain),source,security_domain)  
| stats values(urgency) as urgency, values(status_label) as status_label,  
values(status_description) as status_description, values(event_hash) as event_hash,  
values(event_id) as event_id, values(orig_raw) as orig_raw, values(rule_name) as rule_name,  
values(rule_id) as rule_id, values(status_logger) as status_logger count by owner  
| table *
```

Native macro to return all notable events within the time window searched

Automatically detecting the status change

Required SPL

```
`notable`  
| search status_label="Request Automation"  
| eval rule_name=if(isnull(rule_name),source,rule_name)  
| eval rule_title=if(isnull(rule_title),rule_name,rule_title)  
| eval rule_description=if(isnull(rule_description),source,rule_description)  
| eval security_domain=if(isnull(security_domain),source,security_domain)  
| stats values(urgency) as urgency, values(status_label) as status_label,  
values(status_description) as status_description, values(event_hash) as event_hash,  
values(event_id) as event_id, values(orig_raw) as orig_raw, values(rule_name) as rule_name,  
values(rule_id) as rule_id, values(status_logger) as status_logger count by owner  
| table *
```

Filtering on the custom status created to return all notables pending approval

Automatically detecting the status change

Required SPL

```
`notable`
| search status_label="Request Automation"
| eval rule_name=if(isnull(rule_name),source,rule_name)
| eval rule_title=if(isnull(rule_title),rule_name,rule_title)
| eval rule_description=if(isnull(rule_description),source,rule_description)
| eval security_domain=if(isnull(security_domain),source,security_domain)
| stats values(urgency) as urgency, values(status_label) as status_label,
values(status_description) as status_description, values(event_hash) as event_hash,
values(event_id) as event_id, values(orig_raw) as orig_raw, values(rule_name) as rule_name,
values(rule_id) as rule_id, values(status_logger) as status_logger count by owner
| table *
```

Eval functions to set required fields for downstream automation

Automatically detecting the status change

Required SPL

```
`notable`  
| search status_label="Request Automation"  
| eval rule_name=if(isnull(rule_name),source,rule_name)  
| eval rule_title=if(isnull(rule_title),rule_name,rule_title)  
| eval rule_description=if(isnull(rule_description),source,rule_description)  
| eval security_domain=if(isnull(security_domain),source,security_domain)  
| stats values(urgency) as urgency, values(status_label) as status_label,  
values(status_description) as status_description, values(event_hash) as event_hash,  
values(event_id) as event_id, values(orig_raw) as orig_raw, values(rule_name) as rule_name,  
values(rule_id) as rule_id, values(status_logger) as status_logger count by owner  
| table *
```

Stats command to group notables by owner to support bulk approval request

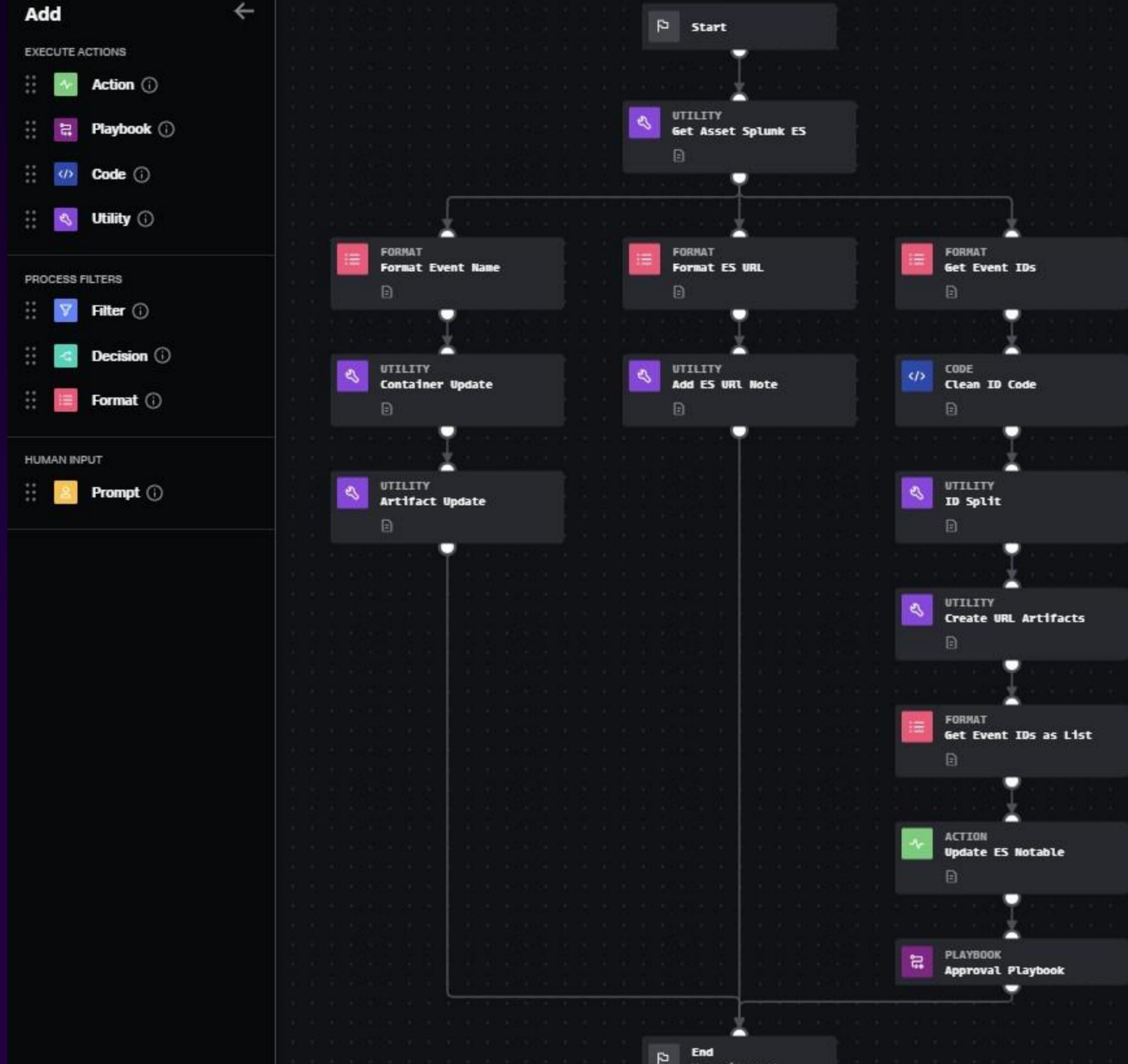
Automatically detecting the status change

Implementing the SPL

- Correlation Search scheduled to run every 5 minutes (* / 5 * * * *)
- Employ throttling on key fields
 - For this environment we used the 'host' field
- Run adaptive response 'Send to SOAR' with specific playbook and label selected
 - We used a specific label in SOAR named 'Approval' to store all approval containers

Notable Onboarding Playbook

Shown in part 1, Previously developed playbook used again for a new customer to solve a problem!

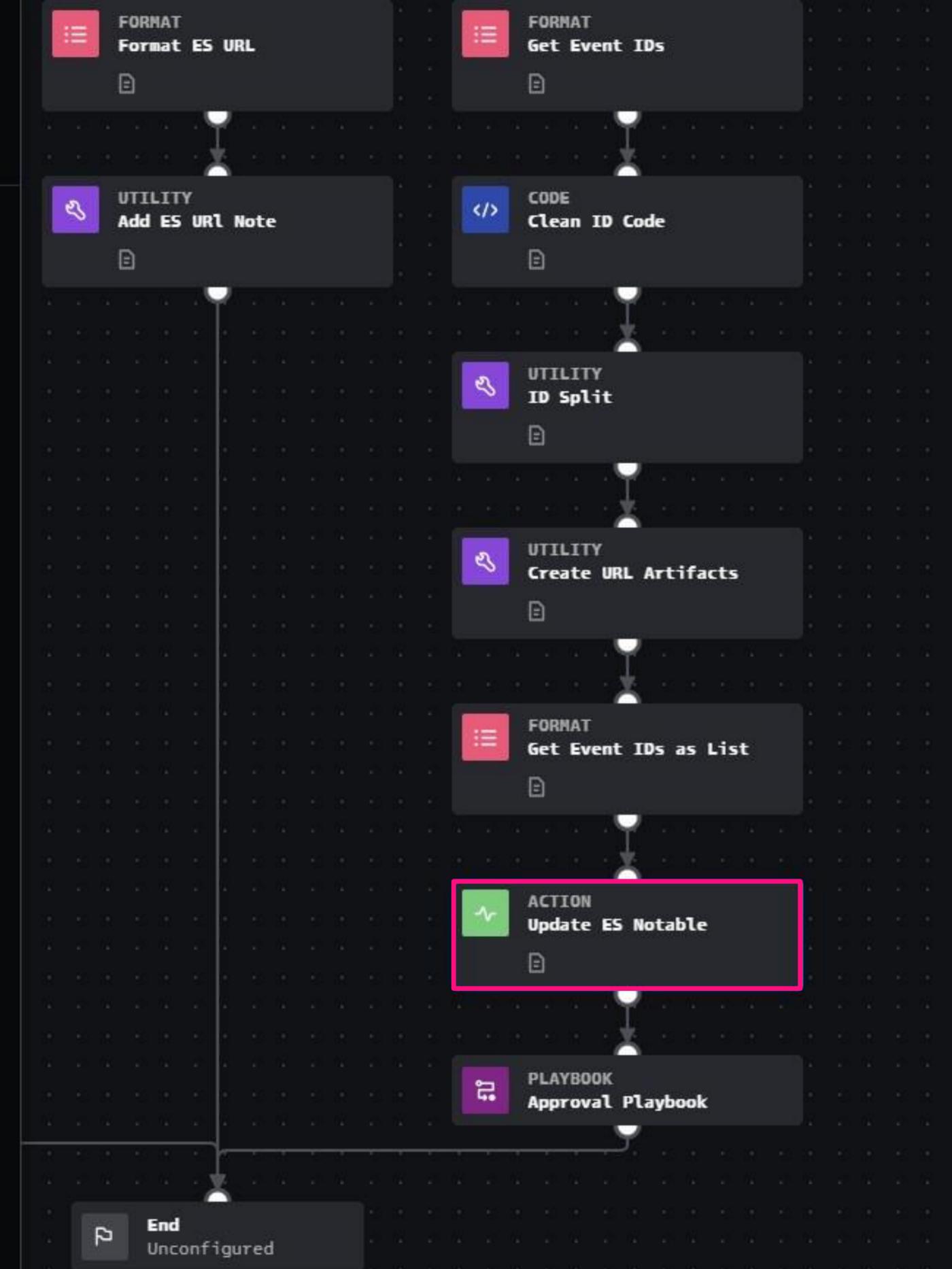


Notable Onboarding Playbook

Action Block:
Splunk Update Event

Update status to approval pending and add comment

The screenshot shows the configuration page for the 'Update ES Notable' action block. The title bar includes a back arrow, the text 'ACTION Update ES Notable update event · Splunk', and a close button. Below the title bar are tabs for 'Configure', 'Info', 'Stats', and 'Loop'. The main configuration area includes several fields: 'Asset' (splunkes), 'Inputs' (event_ids* with a value of get_event_ids_as_list:formatted_data.*), 'owner', 'status' (highlighted with an orange box, set to approval pending), 'integer_status' (numeric), 'urgency', and 'comment' (highlighted with a purple box, containing the text: 'Automation Request has been lodged, pending approval. SOAR event created: {0} Complete details can be found here: https://1.2.3.4/{1}').



Notable Onboarding Playbook

Playbook Block:

Run Approval Playbook as a sub playbook.

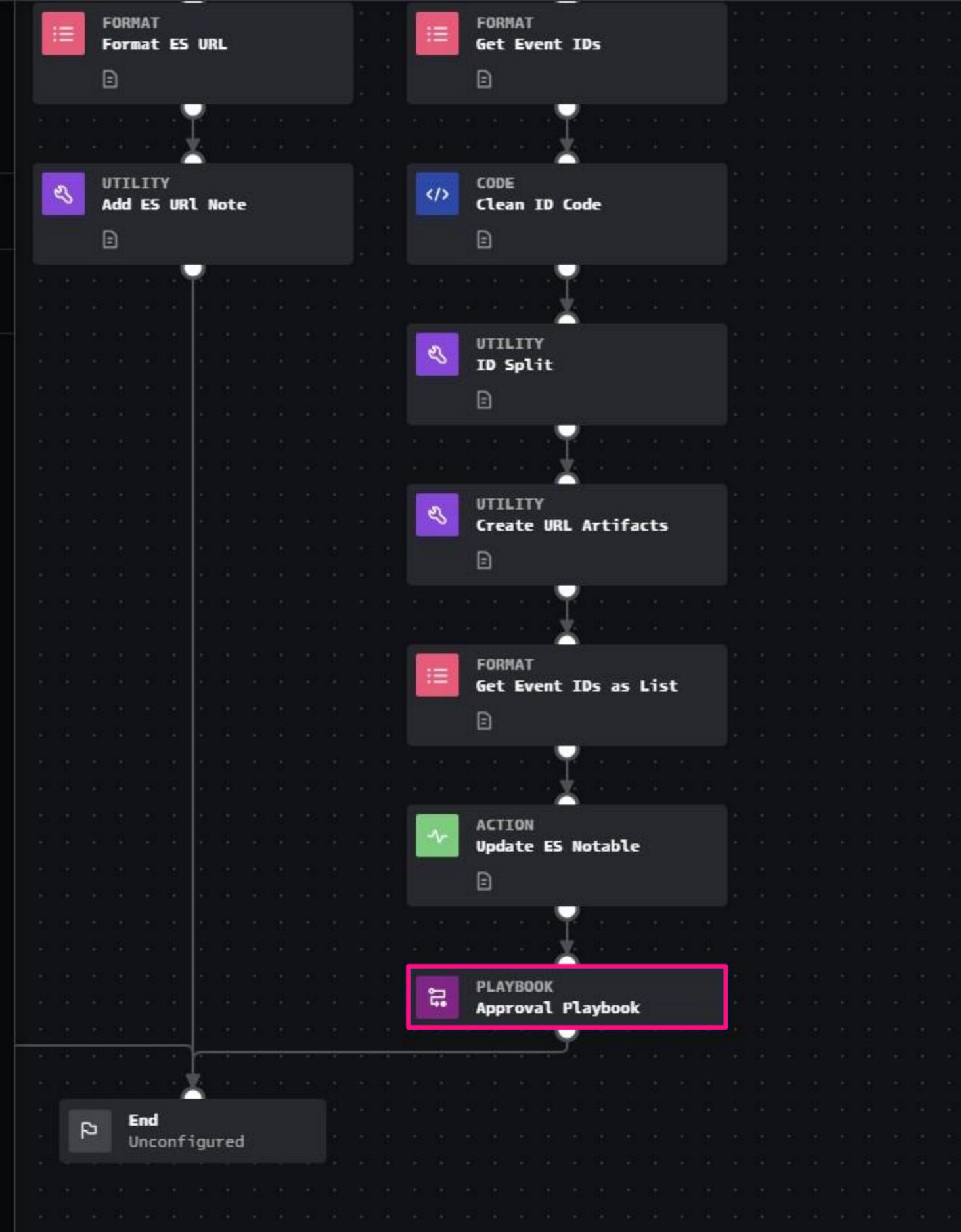
PLAYBOOK
Approval Playbook
Approval Manager - Example

Configure Info Stats Loop

Synchronous Off

> ADVANCED

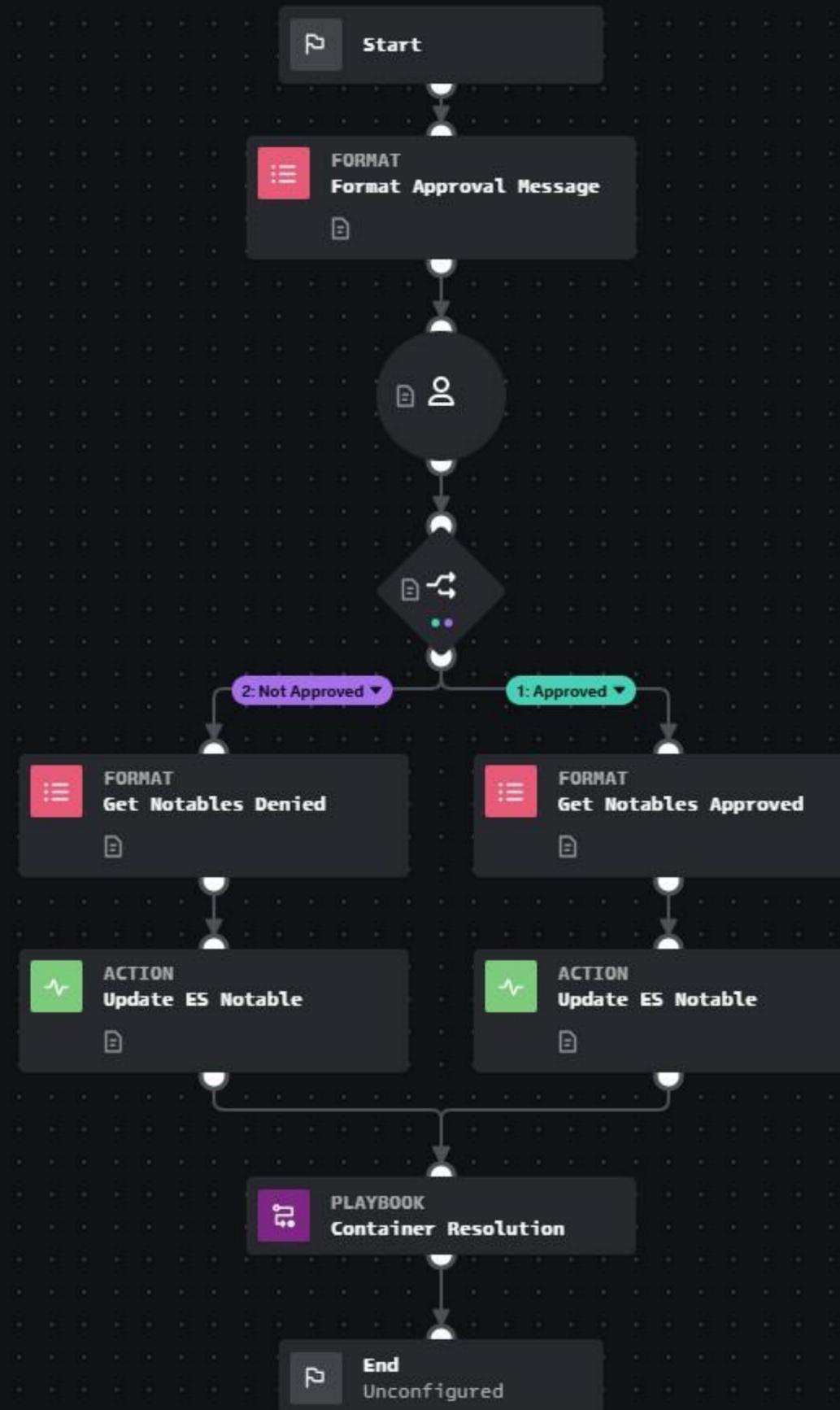
Done



Approval Playbook

New playbook created for
the customer

Simple example of using
prompts to capture an
approval for automation
and linking it back to ES!



Approval Playbook

Format Block:

Format custom message for the approver

FORMAT
Format Approval Message

Configure Info Stats

```
{1} has requested approval for automation on the following hosts  
  
{0}  
  
{1}'s comment
```

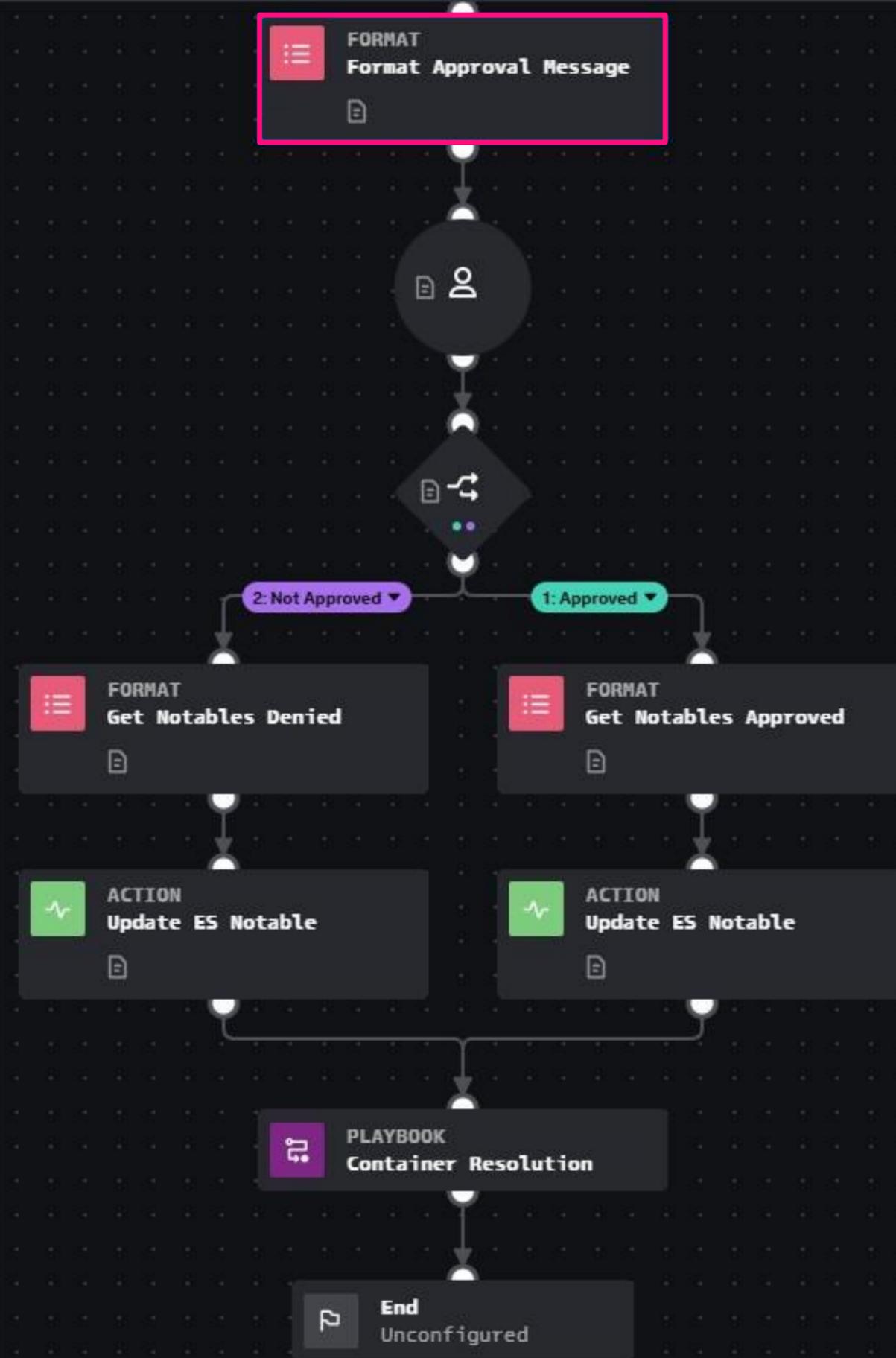
0 artifact*.cef.Endpoint > X

1 artifact*.cef.owner > X

2 artifact*.cef.Comment > X +

> ADVANCED

Done



Approval Playbook

Prompt Block:

Display formatted message
and request Yes/No input
from approver

PROMPT

Approval

Configure Info Stats

User or Role

Asset Owner

{0}

0 format_approval_message

QUESTION(S)

Question 1

Approve Splunk ES notable for endpoint agent

Response Type

Yes/No

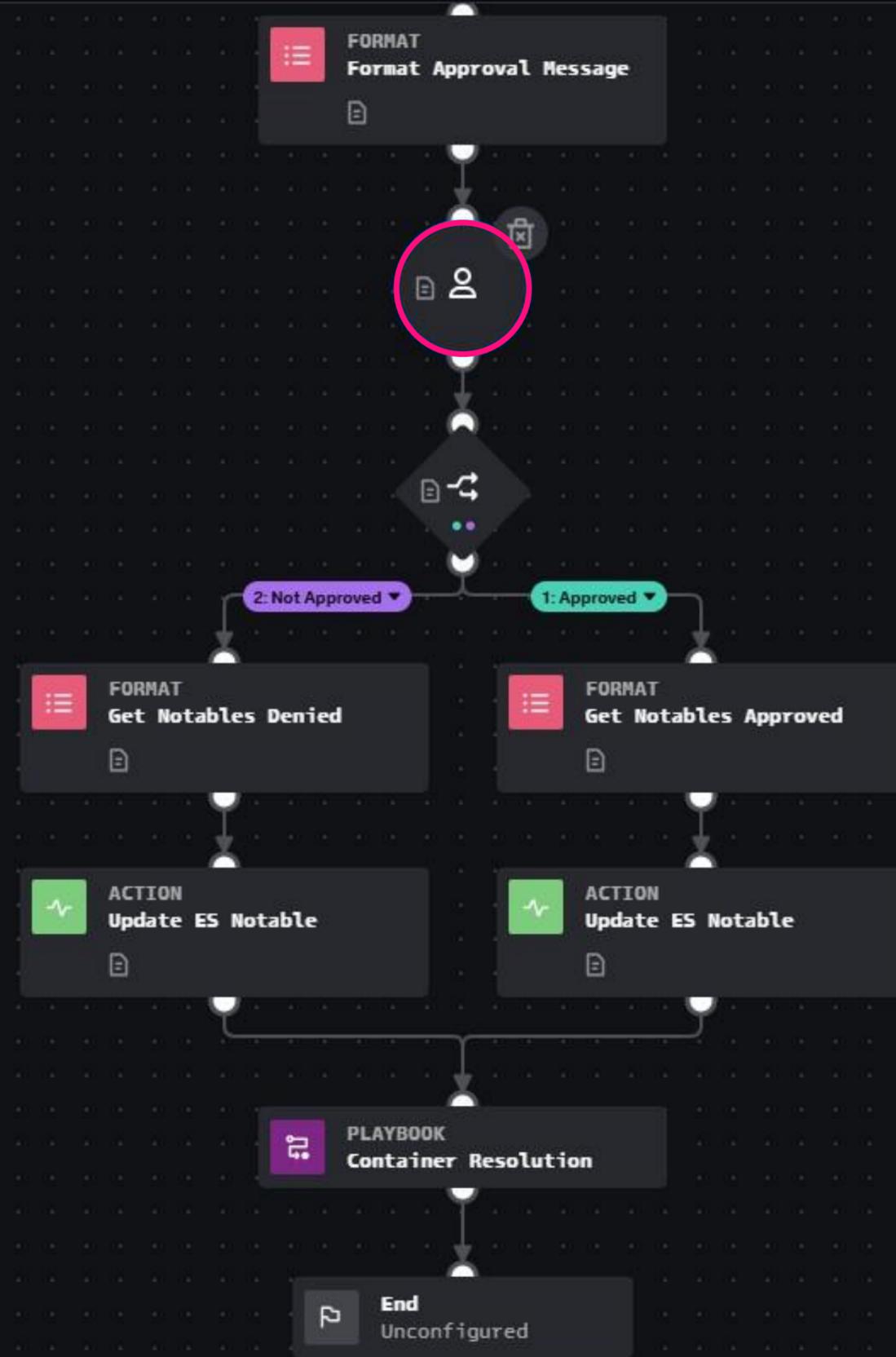
Delete

+ Question

Required response time (mins) 720

ADVANCED

Done



Approval Playbook

Decision Block:

Based off approval prompt, perform required actions

DECISION

Approved ✕

Configure Info Stats

CONDITIONS

If 1: Approved ✎

approval:action_result.summary.responses.(> ✕

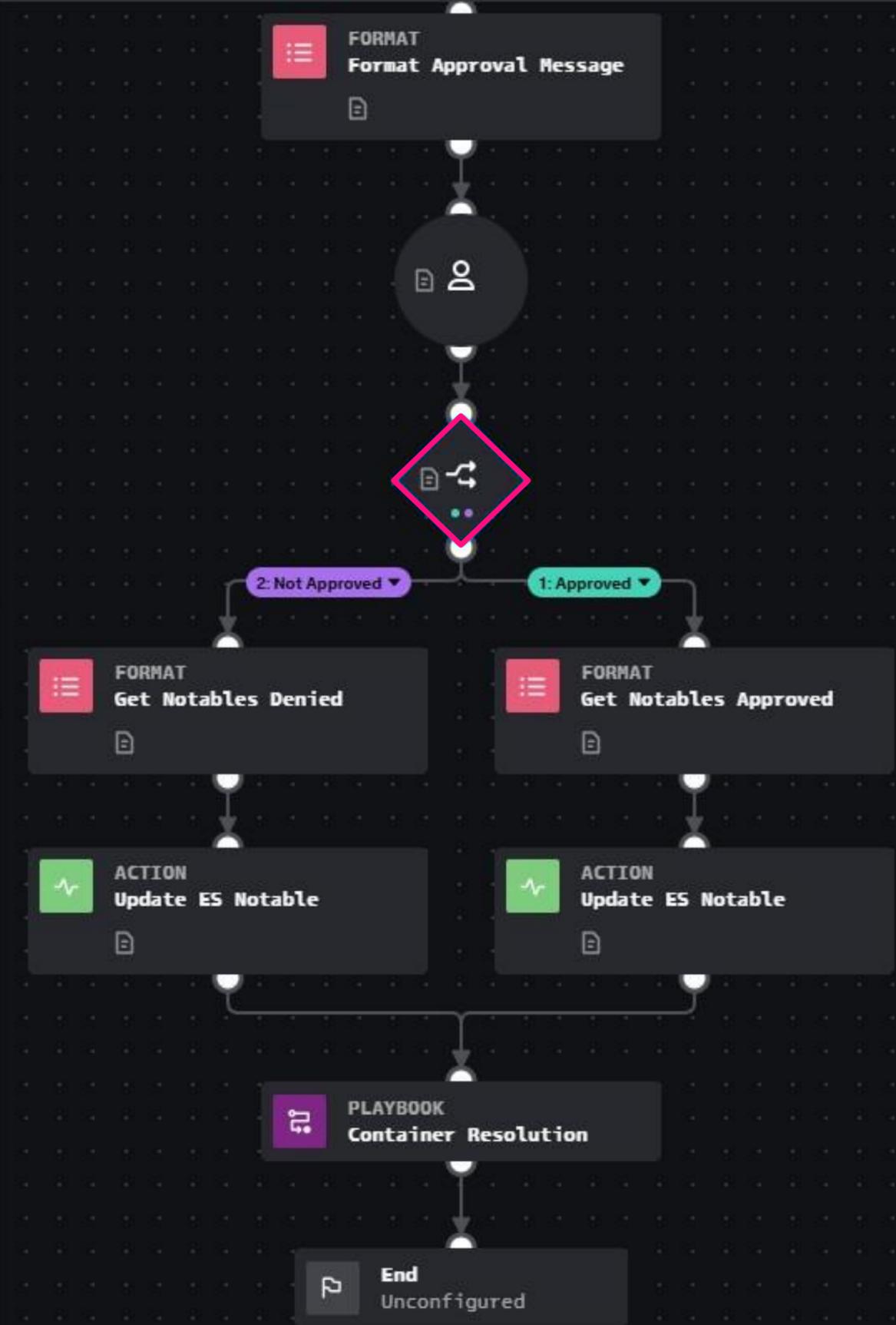
== ▼ Yes > +

Else 2: Not Approved ✎ ✕

+ Else If

> ADVANCED

Done



Approval Playbook

Format Block:

List each Notable ID as a
list

(See Part 1 on this)

FORMAT
Get Notables Approved

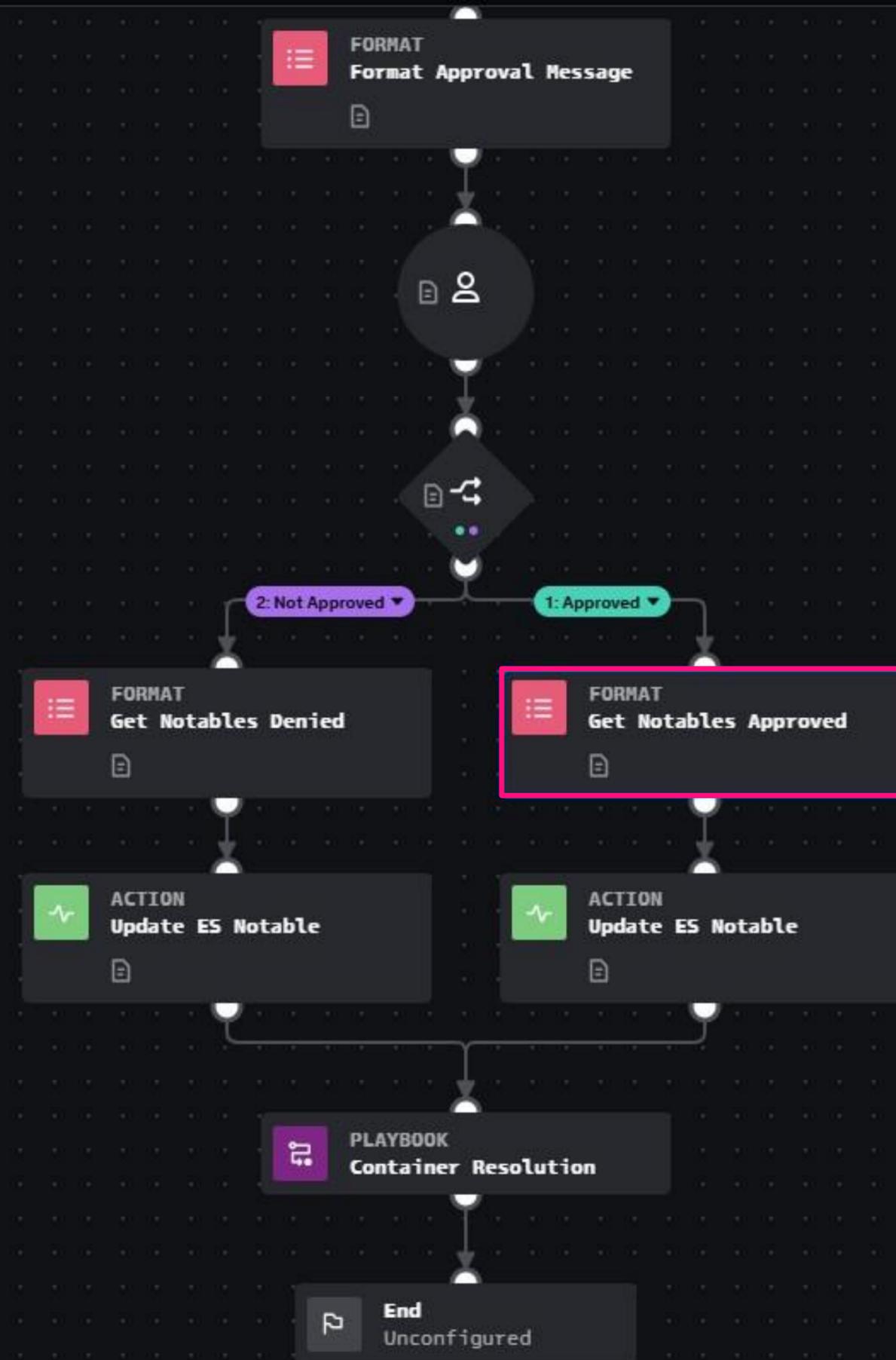
Configure Info Stats

```
%%  
{0}  
%%
```

0 artifact:* .cef.notable_id

ADVANCED

Done



Approval Playbook

Action Block:
Splunk Update Event

Change status to desired
state and provide message
to end user

ACTION Update ES Notable
update event · Splunk

Configure Info Stats Loop

Asset
splunkes

Inputs
event_ids* {0}
get_notables_approved:formatted_data.*

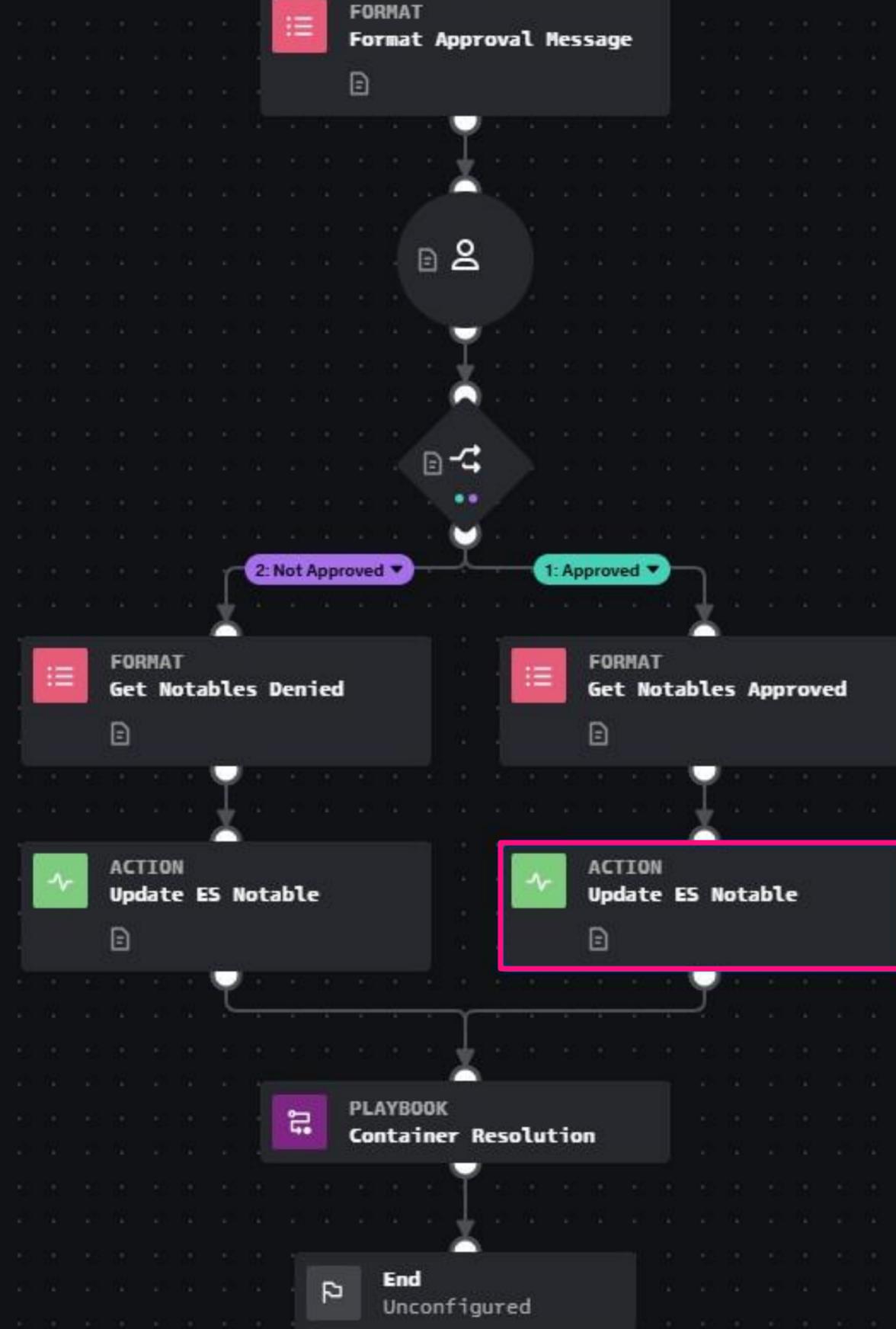
owner {0}

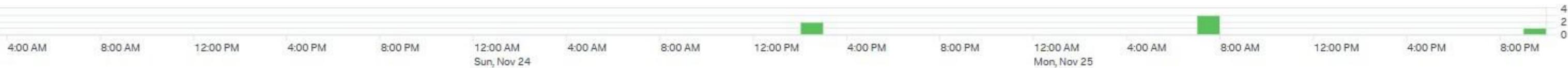
status {0}
automation approved

integer_status {0}
numeric

urgency {0}

comment {0}
Automation Request was approved.
Conduct automation actions as required.





Last refresh at 02:20 PM Auto-refresh off

Type	Time	Urgency	Status	Owner	Actions
FINDING	Mon, Nov 25, 2024 9:05 PM	Medium	Automation Approved	rich	⋮
FINDING	Mon, Nov 25, 2024 7:40 AM	Medium	Automation Approved	benno	⋮
FINDING	Mon, Nov 25, 2024 7:40 AM	Medium	Automation Approved	benno	⋮
FINDING	Mon, Nov 25, 2024 7:40 AM	Medium	Automation Approved	benno	⋮
FINDING	Sun, Nov 24, 2024 2:33 PM	Informational	Automation Approved	benno	⋮
FINDING	Sun, Nov 24, 2024 2:33 PM	Informational	Automation Approved	benno	⋮
FINDING	Fri, Nov 22, 2024 3:52 PM	High	Automation Denied	benno	⋮
FINDING	Fri, Nov 22, 2024 3:26 PM	Critical	Automation Approved	benno	⋮
FINDING	Fri, Nov 22, 2024 2:42 PM	High	Automation Approved	benno	⋮
FINDING	Fri, Nov 22, 2024 1:43 PM	Informational	Automation Approved	benno	⋮
FINDING	Fri, Nov 22, 2024 1:28 PM	Informational	Automation Approved	benno	⋮
FINDING	Fri, Nov 22, 2024 1:27 PM	Informational	Automation Approved	benno	⋮
FINDING	Fri, Nov 22, 2024 1:19 PM	Informational	Automation	benno	⋮

Automation Approval Playbook

Key Takeaways

- Thinking outside the box to create a custom solution to a complex set of requirements
- Leveraged the strengths of both ES and SOAR where each is more appropriate
 - Steered the solution away from a fixation on SOAR doing everything
- Final implementation allowed for bulk automation runs while still meeting customer approval process requirements

Remote Script / Command Execution

Remote Script and Command Execution

Using SOAR to invoke native OS level commands

What and why?

- Requirement to remediate various agents across the environment from SOAR
 - Logging, AV, EDR, VM, etc
 - Extant approach completely manual and no longer viable
- Solution needed to be able to:
 - Start/stop agents
 - Install/remove agents
 - Perform automated troubleshooting remotely, and capture results

Remote Script and Command Execution

Using SOAR to invoke native OS level commands

How?

- Custom scripts to do heavy lifting on OS side
 - WinRM App in SOAR using the Run Script function
 - SSH App in SOAR using the Execute Program function
- Leverage Native SOAR functions

Remote Script Example

Pushes a script to the remote endpoint and executes various scripted actions



Remote Script Example

Filter Block:

Checking if artifact is marked as old_artifact

Executing an adaptive response on an ES Notable re-opens the same container it originally created

FILTER

Artifact Filter

Configure Info Stats

CONDITIONS

If **1: OLD ARTIFACT**

artifact:*.cef.artifact_status

== old_artifact

If **2: NEW ARTIFACT**

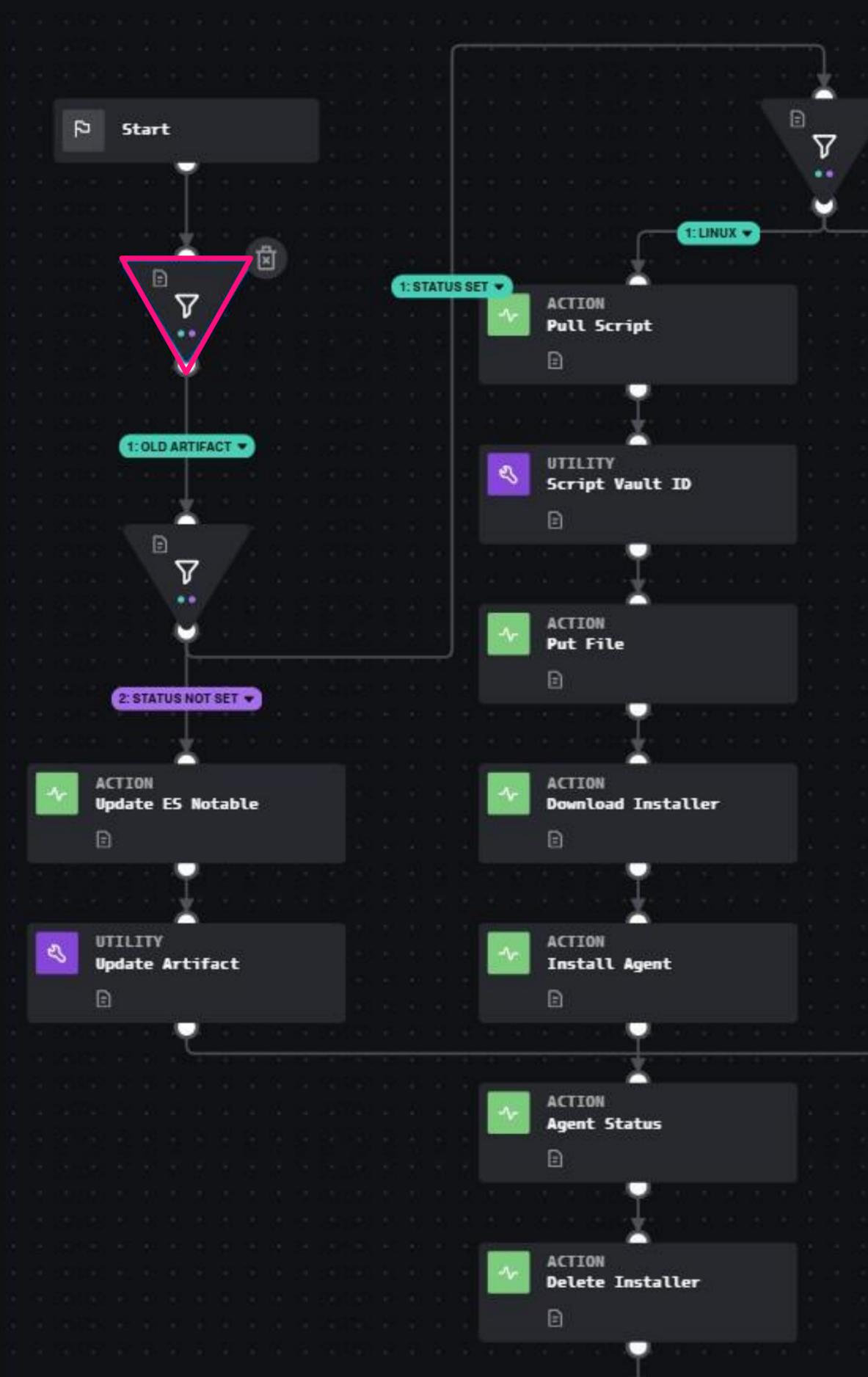
artifact:*.cef.artifact_status

== Select Value

+ Condition

> ADVANCED

Done



Remote Script Example

Filter Block:

Check if artifact contains Approved field under status_label

FILTER
Check Approval [Close]

Configure Info Stats

CONDITIONS

If **1: STATUS SET** [Edit]

artifact:*cef.status_label [Remove]

== Approved [Add]

If **2: STATUS NOT SET** [Edit]

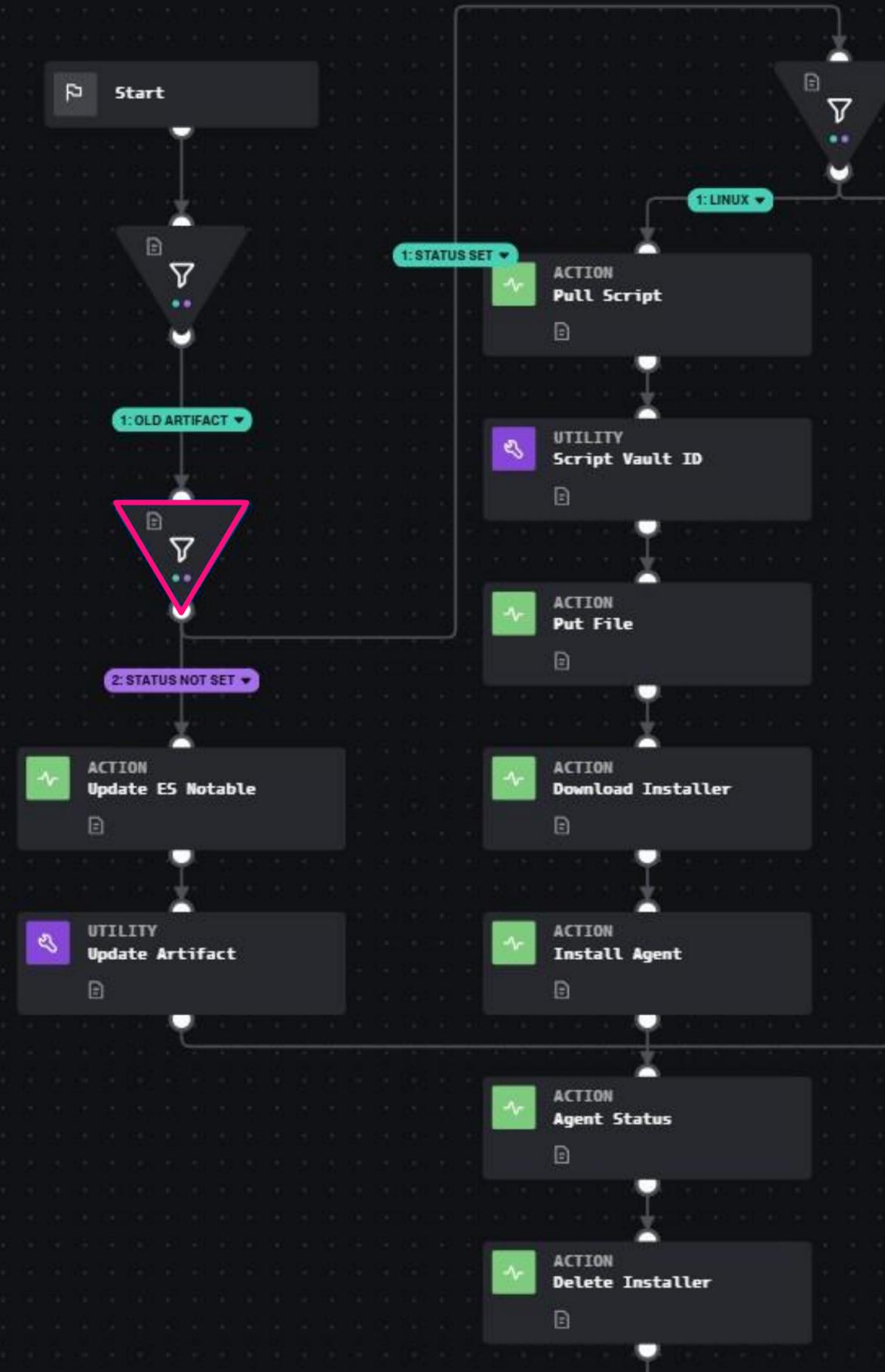
artifact:*cef.status_label [Remove]

!= Approved [Add]

+ Condition

> ADVANCED

Done



Remote Script Example

Action Block:
Update Splunk

Send comment to Notable event stating automation not run due to approval state

Asset
splunkes

Inputs
event_ids* ⓘ {0}
artifact:*.*.cef.event_id >

owner ⓘ {0}

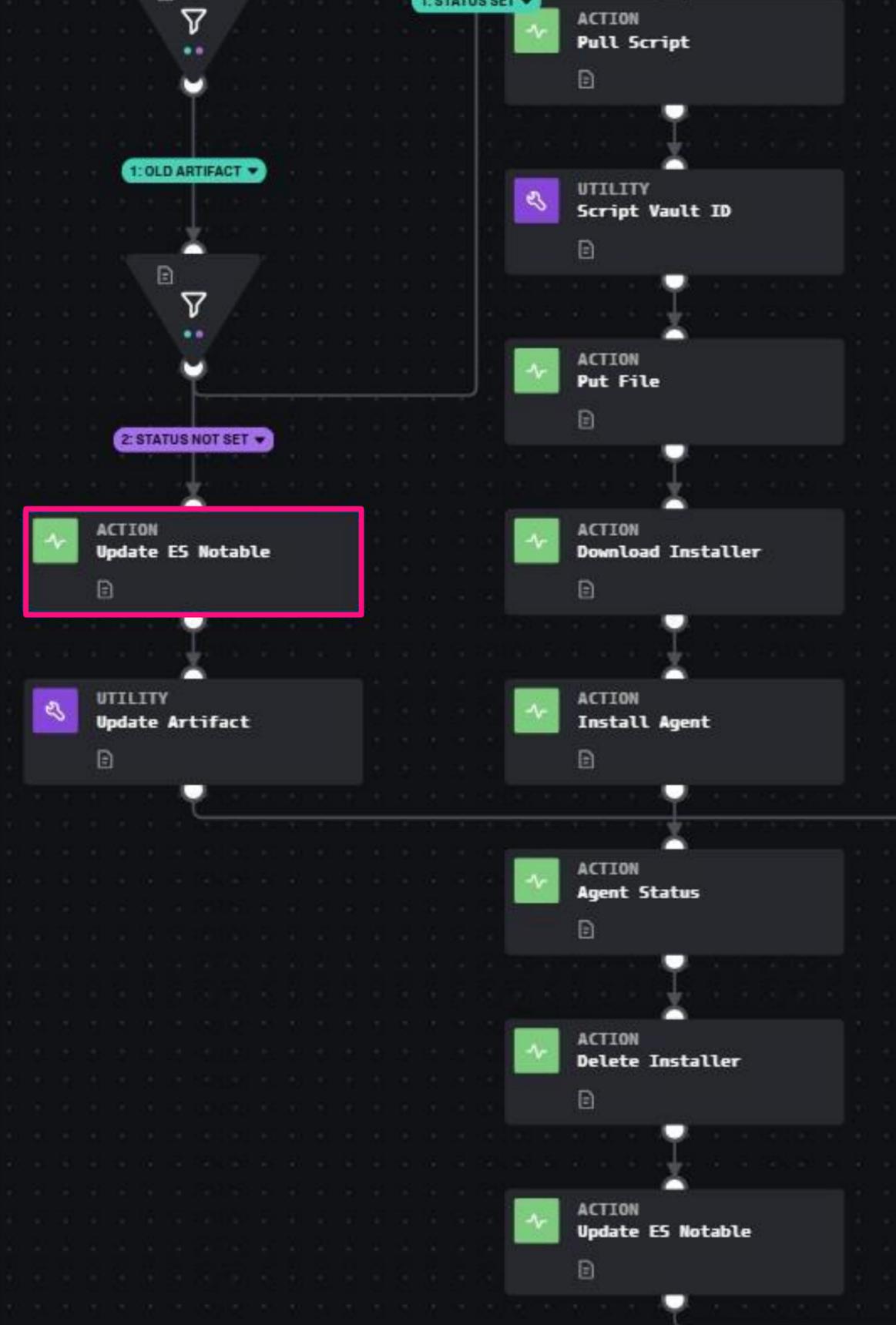
status ⓘ {0}

integer_status ⓘ
numeric >

urgency ⓘ {0}

comment ⓘ {0}

Current status of the notable is not acceptable. Please seek approval and have the Remediation Approved status set before running any of the endpoint agent playbooks.



Remote Script Example

Filter Block:

Check for OS to determine correct path

FILTER OS Type Filter

Configure Info Stats

CONDITIONS

If

1: LINUX

artifact:*cef.OS

==

Linux

If

2: WINDOWS

artifact:*cef.OS

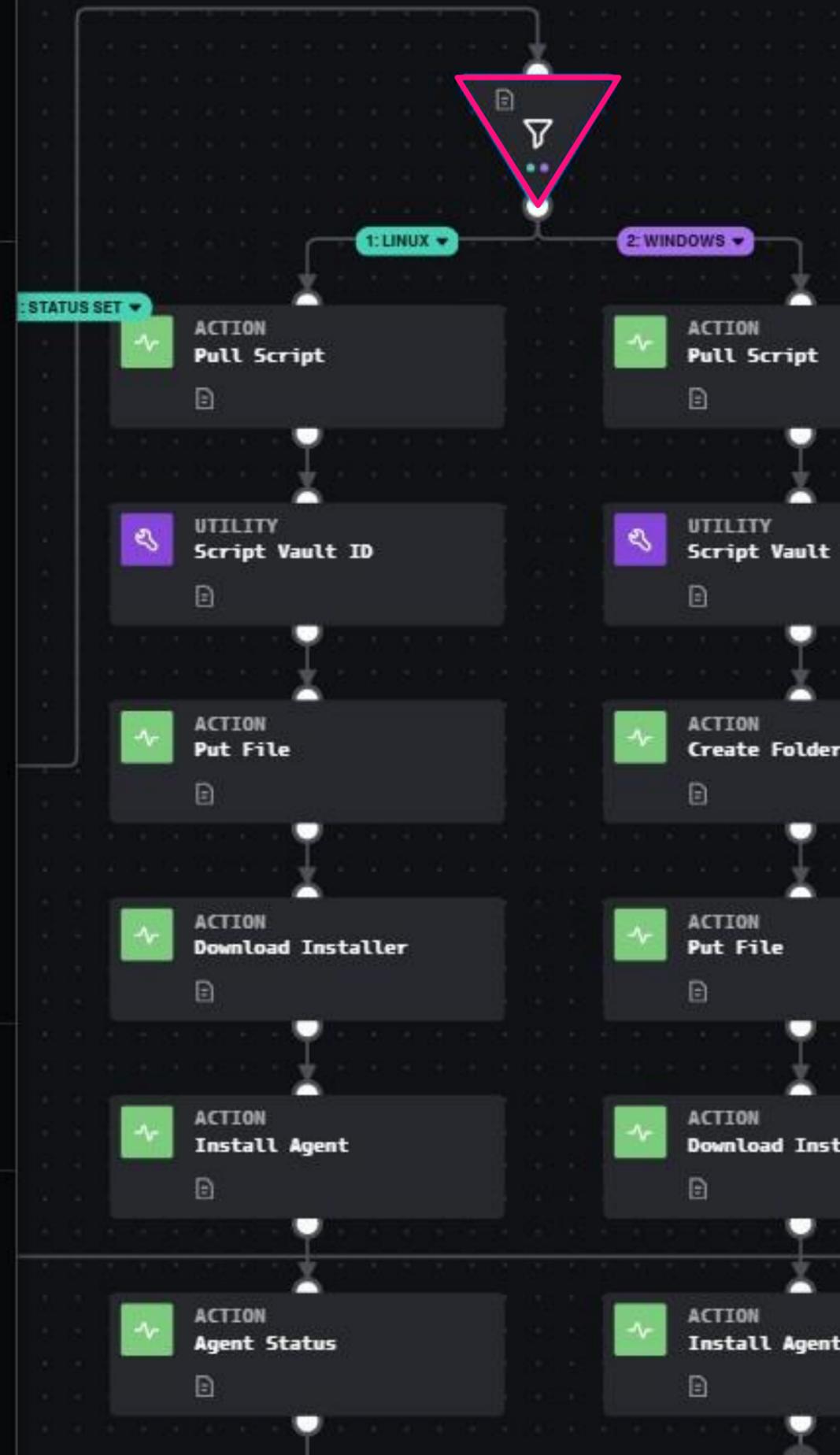
==

Windows

+ Condition

> ADVANCED

Done



Remote Script Example

Action Block: SSH Get File

Use the SSH App and the get file action to connect SOAR itself using 127.0.0.1 to get the script

← ACTION
Pull Script
get file · SSH

Configure Info Stats Loop

Asset
test123

Inputs

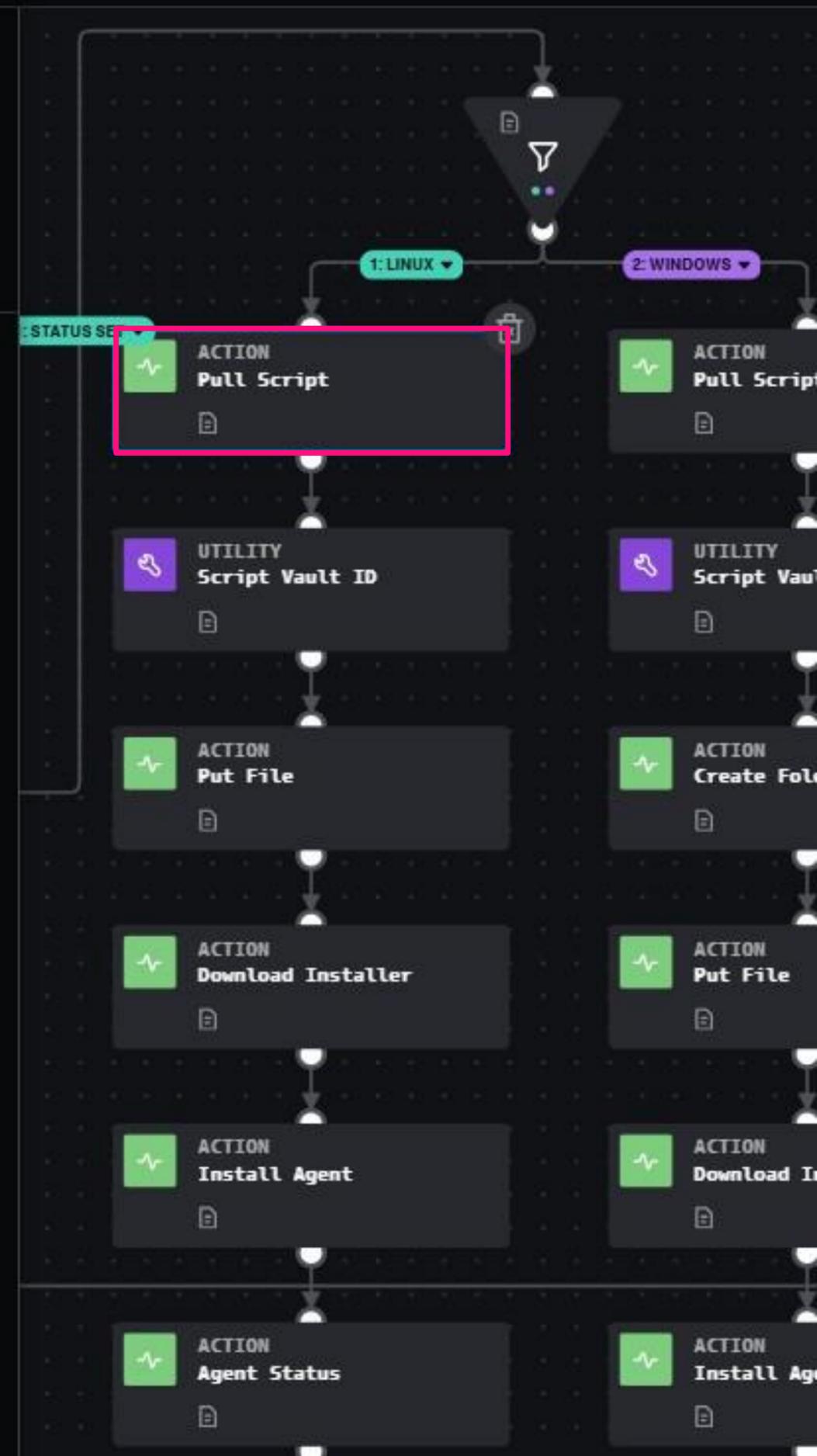
ip_hostname* ⓘ {0}

127.0.0.1 >

file_path* ⓘ {0}

/tmp/rhel/scipt_rhel.sh

0 Select Parameter > × +



Remote Script Example

Utility Block: Vault List

Add the previously pulled script into the containers file vault

UTILITY
Script Vault ID
vault_list

Configure Info Stats Loop

vault_list

container_id ⓘ

container:id >

vault_id ⓘ

artifact:*.cef.vault_id >

file_name ⓘ

script_rhel.sh >

> ADVANCED

Done

STATUS SET

ACTION
Pull Script

UTILITY
Script Vault ID

ACTION
Put File

ACTION
Download Installer

ACTION
Install Agent

ACTION
Agent Status

ACTION
Pull Script

UTILITY
Script Vault

ACTION
Create Fold

ACTION
Put File

ACTION
Download In

ACTION
Install Age

Remote Script Example

Action Block: SSH Put File

Now use SSH Put File action again to place the script on the target endpoint using the vault file.

← ACTION
Put File
put file · SSH ×

Configure Info Stats Loop

Asset
test123

Inputs

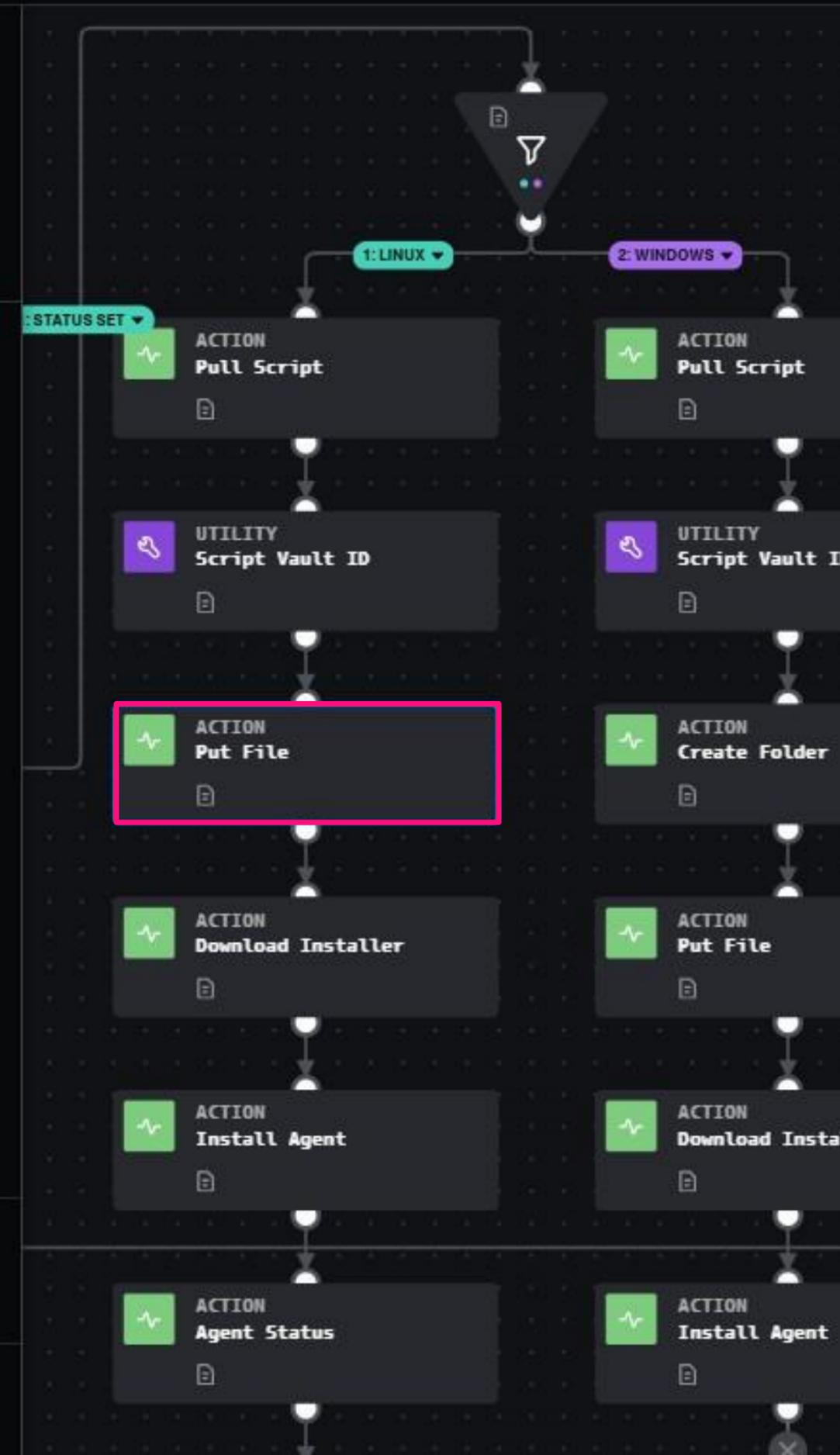
ip_hostname* ⓘ {0}
artifact:*.cef.IP >

vault_id* ⓘ {0}
vault_id >

file_destination* ⓘ {0}
/root/ >

> ADVANCED

Done



Remote Script Example

Action Block:
SSH Execute Program

Using bash execute the script file placed on target host.

Note in this instance its called a specific function within the script.

← ACTION
Download Installer
execute program · SSH

Configure Info Stats Loop

Asset
test123

Inputs

ip_hostname* ⓘ {0}

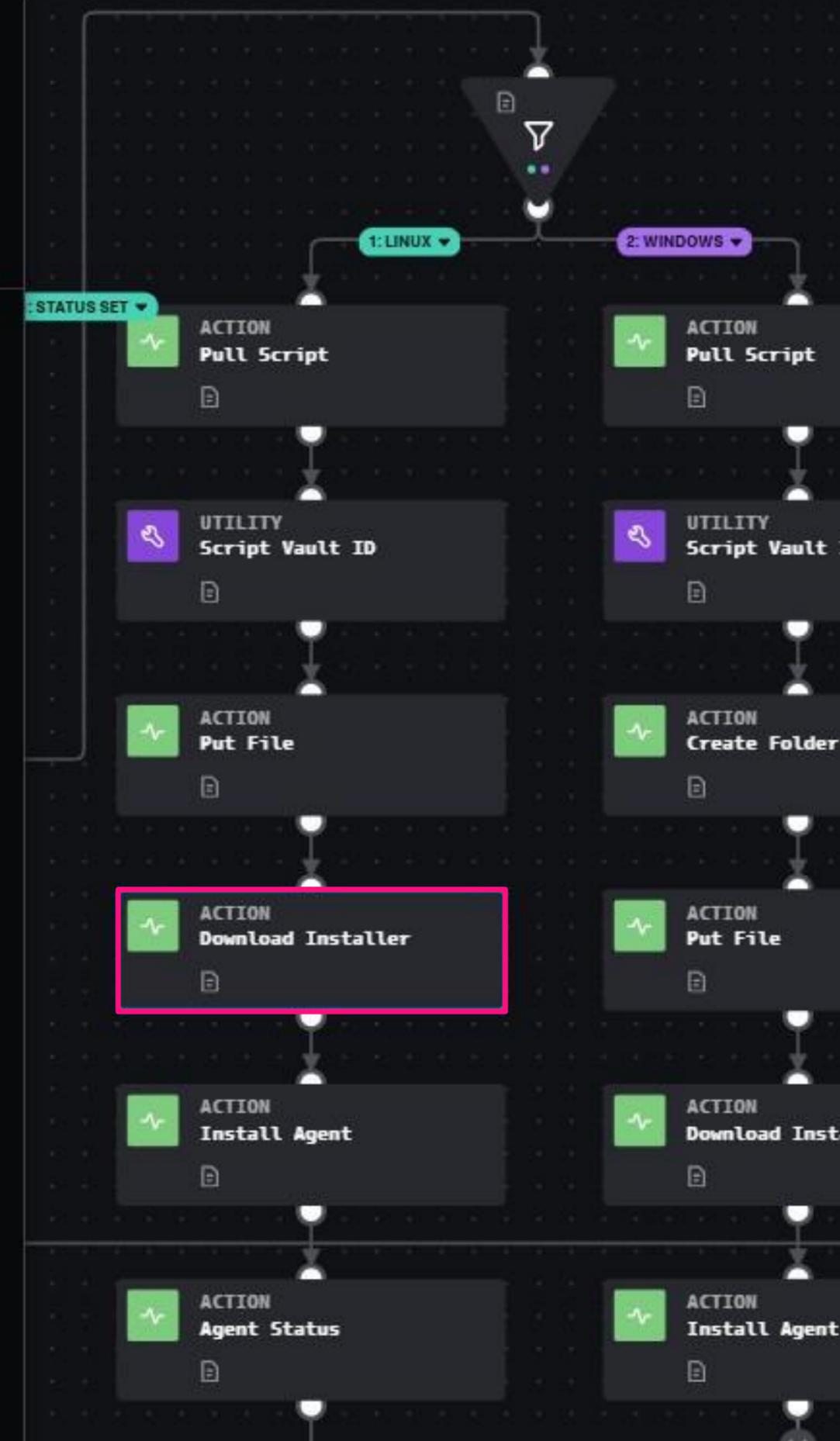
artifact:* .cef.IP >

command ⓘ {0}

```
bash script_rhel.sh agent_download /tmp/
```

+ Message Parameter

script_file ⓘ {0}



Remote Script Example

Action Block:
SSH Execute Program

Remove the script from the
end host

← ACTION
Delete Installer
execute program · SSH

Configure Info Stats Loop

Asset
test123

Inputs

ip_hostname* ⓘ {0}

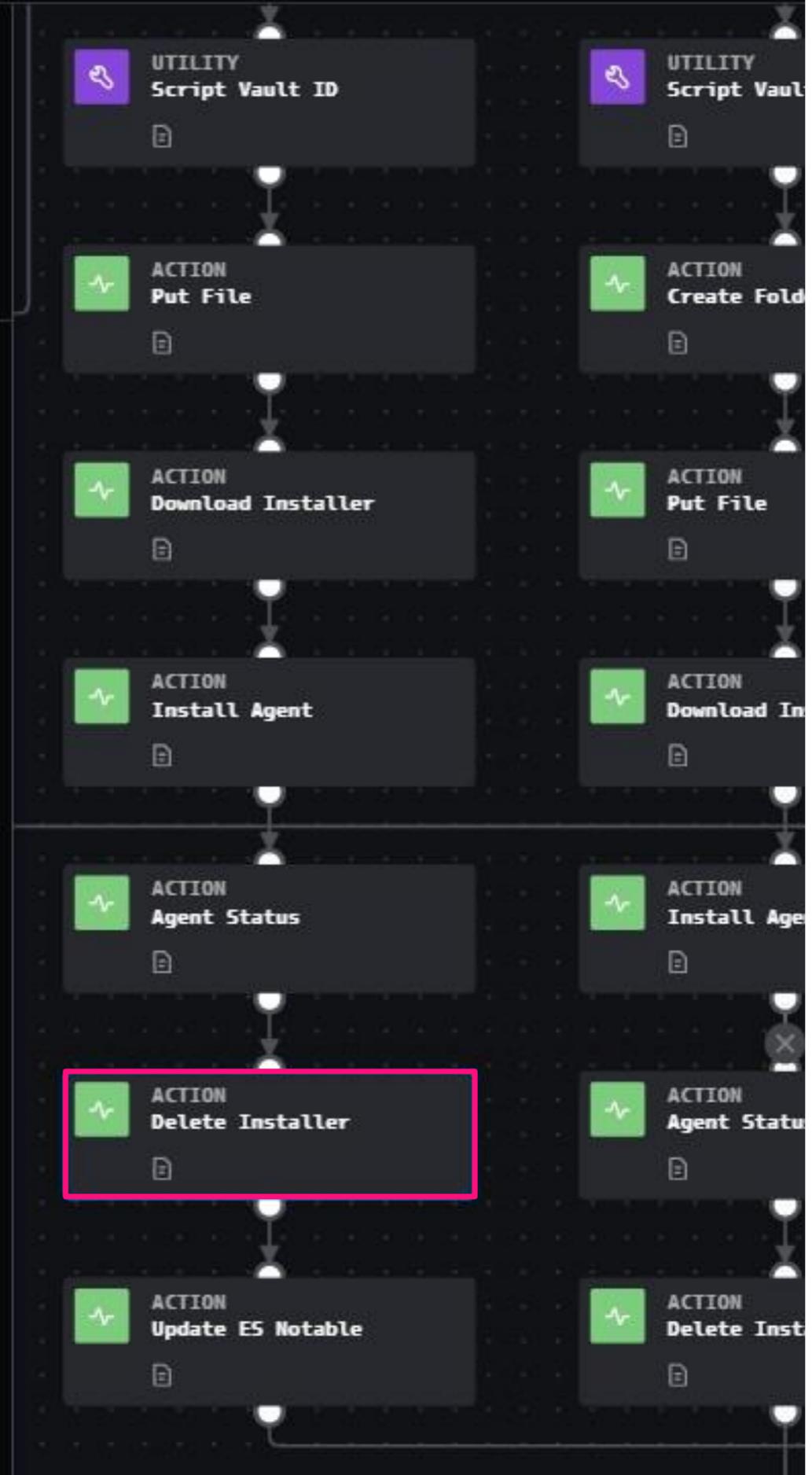
artifact:* .cef.IP >

command ⓘ {0}

```
rm -rf /tmp/script_installer/
```

+ Message Parameter

script file ⓘ

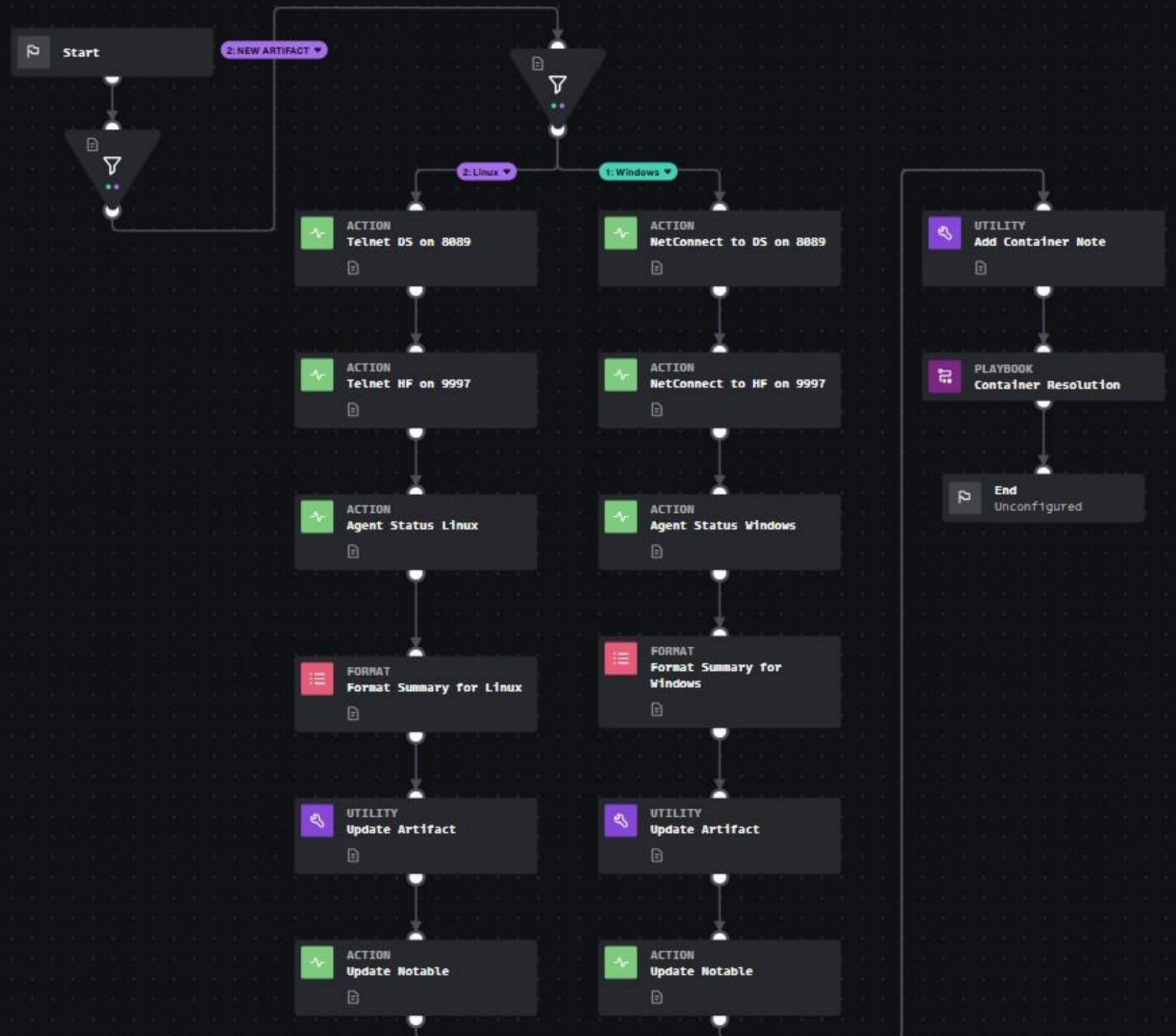


Remote Command Example

Runs commands based off determined host OS

Command results are returned to SOAR

Results formatted



Remote Command Example

Action Block:
SSH Execute Program

Run telnet command against IP and Port 8089 to test Splunk to Splunk

Results are automatically returned

← ACTION
Telnet DS on 8089
execute program · SSH

Configure Info Stats Loop

Asset
test123

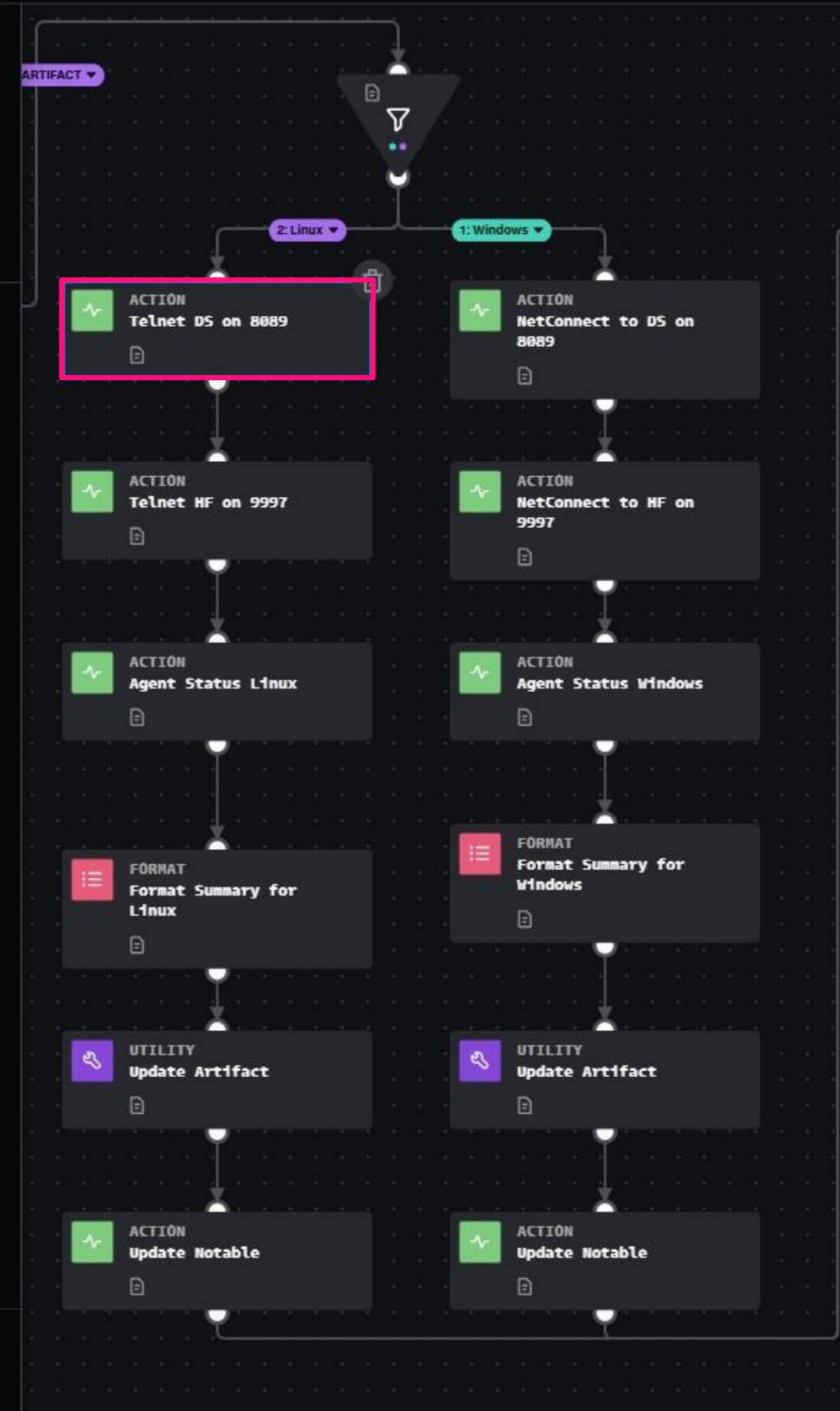
Inputs
ip_hostname* ⓘ
artifact:*.cef.IP

command ⓘ
echo "quit" | telnet 1.2.3.4 8089

script_file ⓘ

timeout ⓘ
numeric

> ADVANCED



Remote Command Example

Action Block:
SSH Execute Program

Run splunk status command

← ACTION
Agent Status Linux
execute program · SSH

Configure Info Stats Loop

Asset
test123

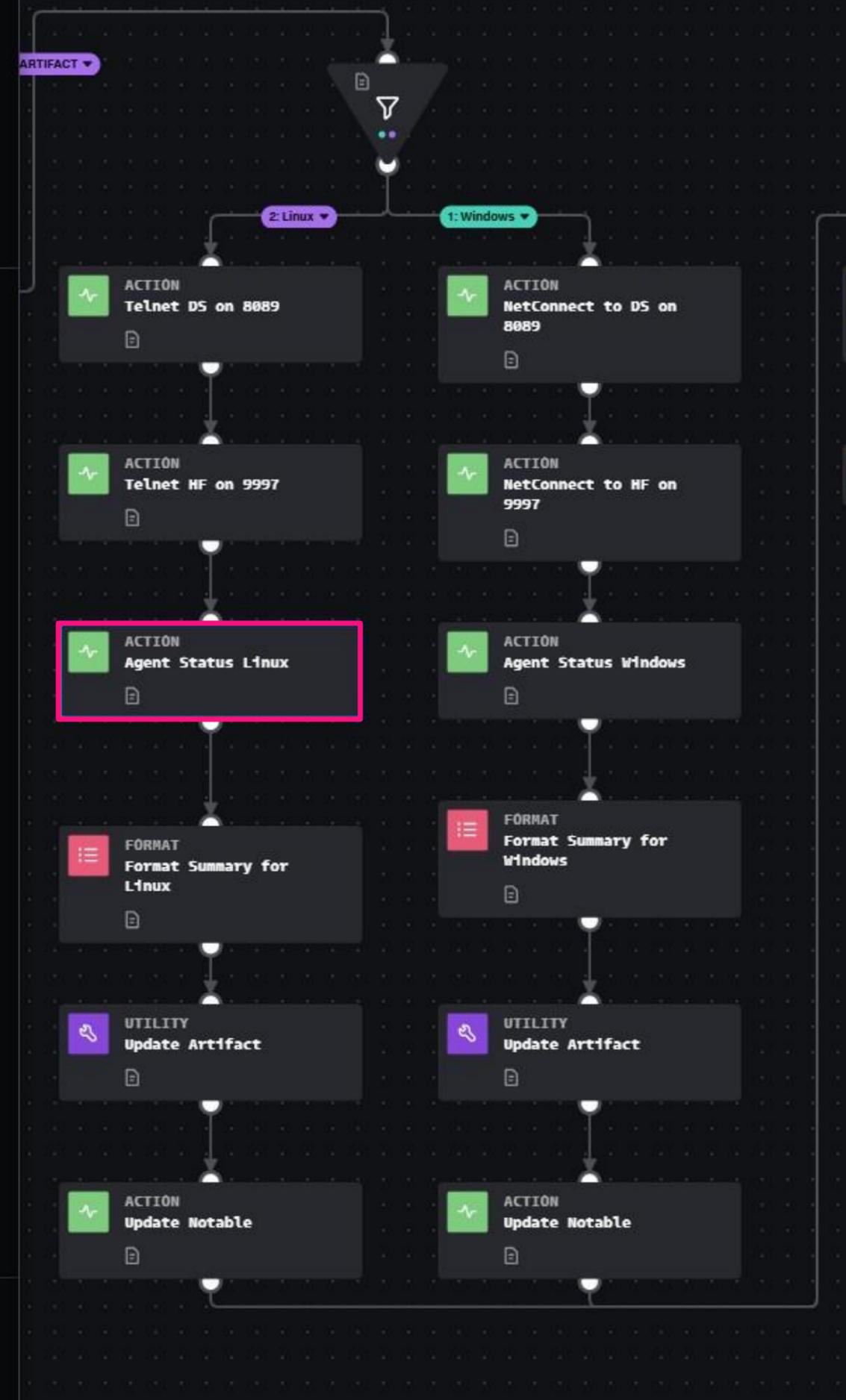
Inputs
ip_hostname* ⓘ {0}
artifact:*.cef.IP >

command ⓘ {0}
/opt/splunkforwarder/bin/splunk status >

script_file ⓘ {0}

timeout ⓘ
numeric >

> ADVANCED



Remote Command Example

Format Block:

Format a message using all the returned results from the previous action blocks

Note the drop none is selected

FORMAT Format Summary for Linux

Configure Info Stats

```
Telnet to Deployment Server Result  
{1}  
  
Telnet to Heavy Forwarder Result  
{2}
```

- 0 agent_status_linux:action
- 1 telnet_ds_on_8089:actio
- 2 telnet_hf_on_9997:actio

ADVANCED

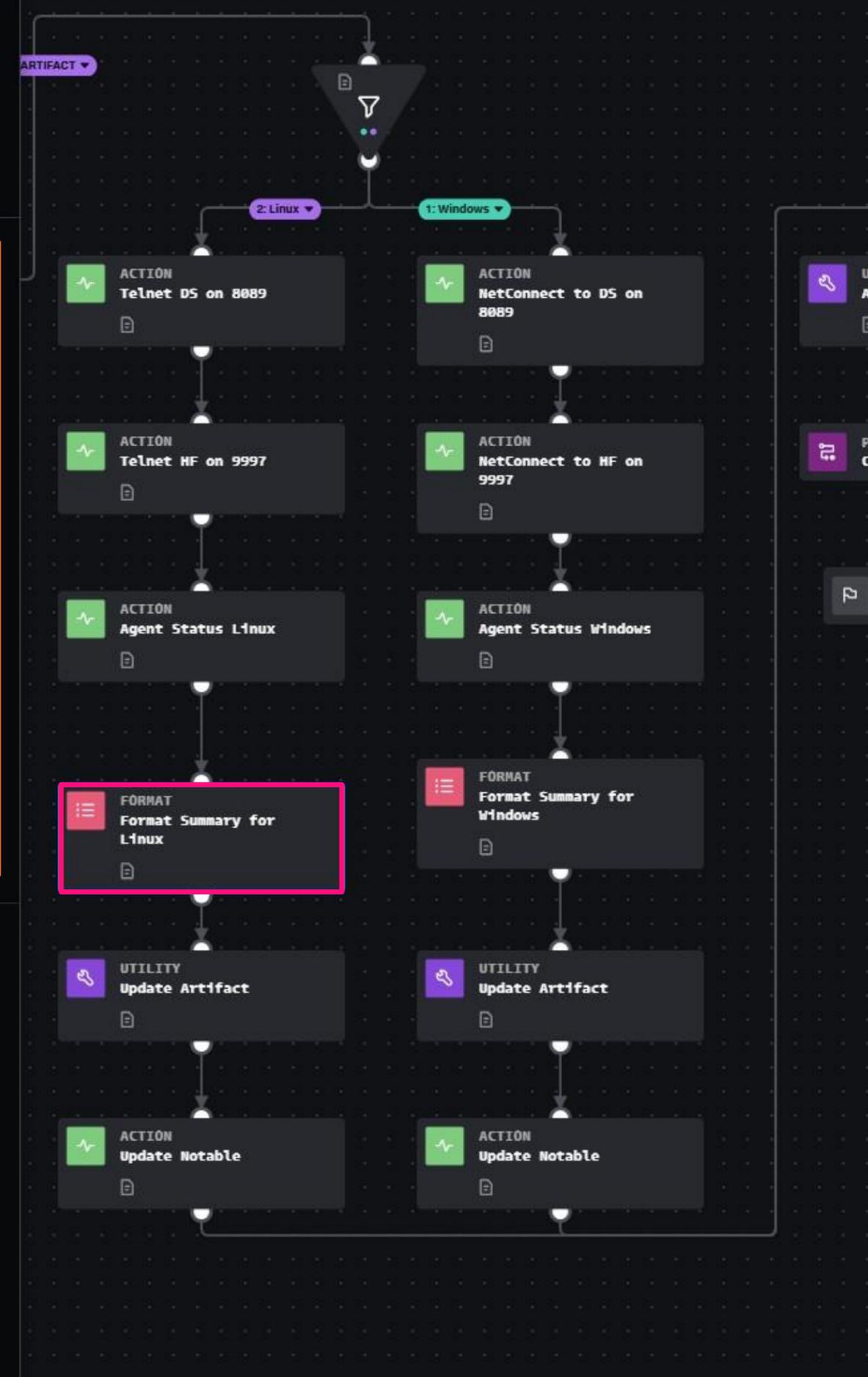
Scope ⓘ

Default

Delimiter ⓘ

,

Drop None ⓘ



Remote Command Example

Utility Block:
Artifact Update

Add custom field to artifact
for future use/ filtering

UTILITY
Update Artifact
artifact_update

Configure Info Stats Loop

artifact_update

artifact_id ⓘ

artifact:*id >

name ⓘ

artifact >

label ⓘ

events >

severity ⓘ

Medium >

cef_field ⓘ

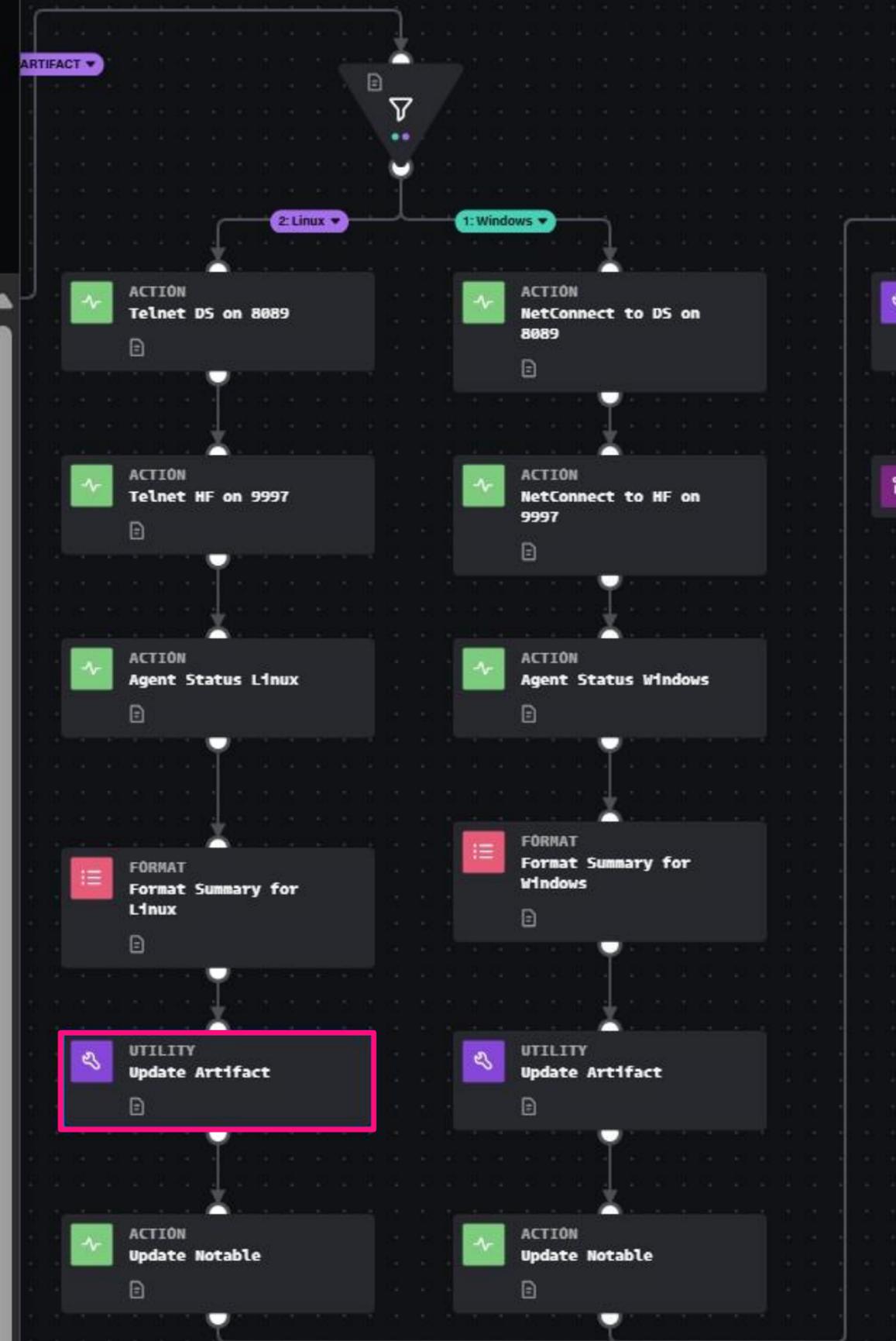
artifact_status >

cef_value ⓘ

old_artifact >

cef_data_type ⓘ

ip >



Remote Command Example

Action Block:
Splunk Update Event

Send formatted results to ES as a comment for the analyst to review

← ACTION
Update Notable
update event · Splunk

Configure Info Stats Loop

Inputs

event_ids* ⓘ {0}

artifact:*.cef.event_id >

owner ⓘ {0}

status ⓘ {0}

integer_status ⓘ

numeric >

urgency ⓘ {0}

comment ⓘ {0}

{0}



Remote Script / Command Execution

Key Takeaways

- Use the underlying Operating System to do the heavy lifting and return results to SOAR
- This initial phase was focused on compliance monitoring and troubleshooting agents
 - However the next phase will reuse this functionality to support automated investigations and incident response
- Consider how you might use these techniques with that in mind
 - Running native OS commands to gather evidence (ps, netstat, etc)
 - Running additional scripts to perform containment functions

Bonus Tips

Bonus Tip

Splunk Lookup Tables within* SOAR!

- *“SPL EVAL Functions within* SOAR”* was a fan favorite from our previous talk
 - See slide 40 of our previous talk
 - Lets extend on this, and use it to leverage lookup tables...
- *“This would be so much easier if I could just use a Splunk Lookup table instead”*
 - Anyone who’s ever had to write python code in SOAR for to manage a custom list...
- SOAR Custom Lists are very powerful, but have limited out of the box functionality
 - Assumes playbook developer will write a custom function in python to handle their data
- What if we cheat and get Splunk to do the heavy lifting for us?

Bonus Tip

Leverage Splunk Lookup Tables via SOAR

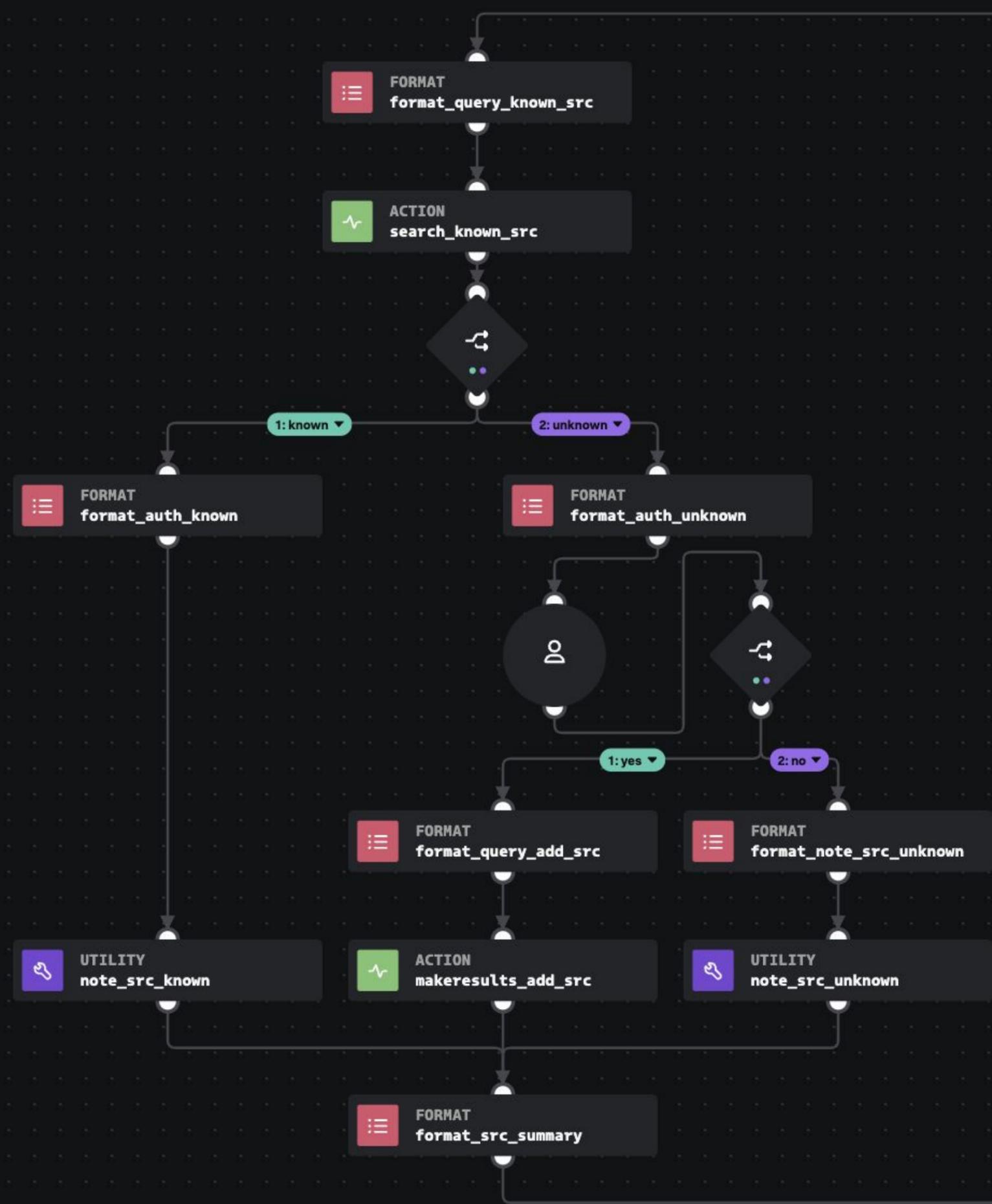
Additional branch for our user investigative playbook to check if their source IP(s) are known / trusted...



Bonus Tip

Leverage Splunk Lookup Tables via SOAR

Additional branch for our user investigative playbook to check if their source IP(s) are known / trusted...



Bonus Tip

Leverage Splunk Lookup Tables via SOAR

... with 'trust' based on a lookup table within Splunk, containing an IP address and a description.

The screenshot shows the Splunk interface for a search. The search bar contains 'inputlookup known_hosts.csv' and is highlighted with a red box. Below the search bar, the results are displayed in a table view. The table has two columns: 'description' and 'src'. The results are:

description	src
VPN Public IP - Melbourne	101.53.23.127
VPN Public IP - Sydney	101.53.17.127

Bonus Tip

Leverage Splunk Lookup Tables via SOAR

For this talk we'll use two demo containers to step through the example.

General Note by soar_local_admin 15 minutes ago

Container Summary:

ID: 1000

Created: Jun 30th at 2:55 am

Updated: Jun 30th at 2:55 am

ARTEFACT	CONTENT
----------	---------

user	admin
------	-------

src	101.53.23.127
-----	---------------

General Note by soar_local_admin 18 minutes ago

Container Summary:

ID: 1024

Created: Jun 30th at 2:56 am

Updated: Jun 30th at 2:56 am

ARTEFACT	CONTENT
----------	---------

user	pwny
------	------

src	1.156.14.224
-----	--------------

Bonus Tip

Leverage Splunk Lookup Tables via SOAR

Format Block:

Write your SPL to search for the users activity, and cross-check against the lookup table.

FORMAT
format_query_known_src

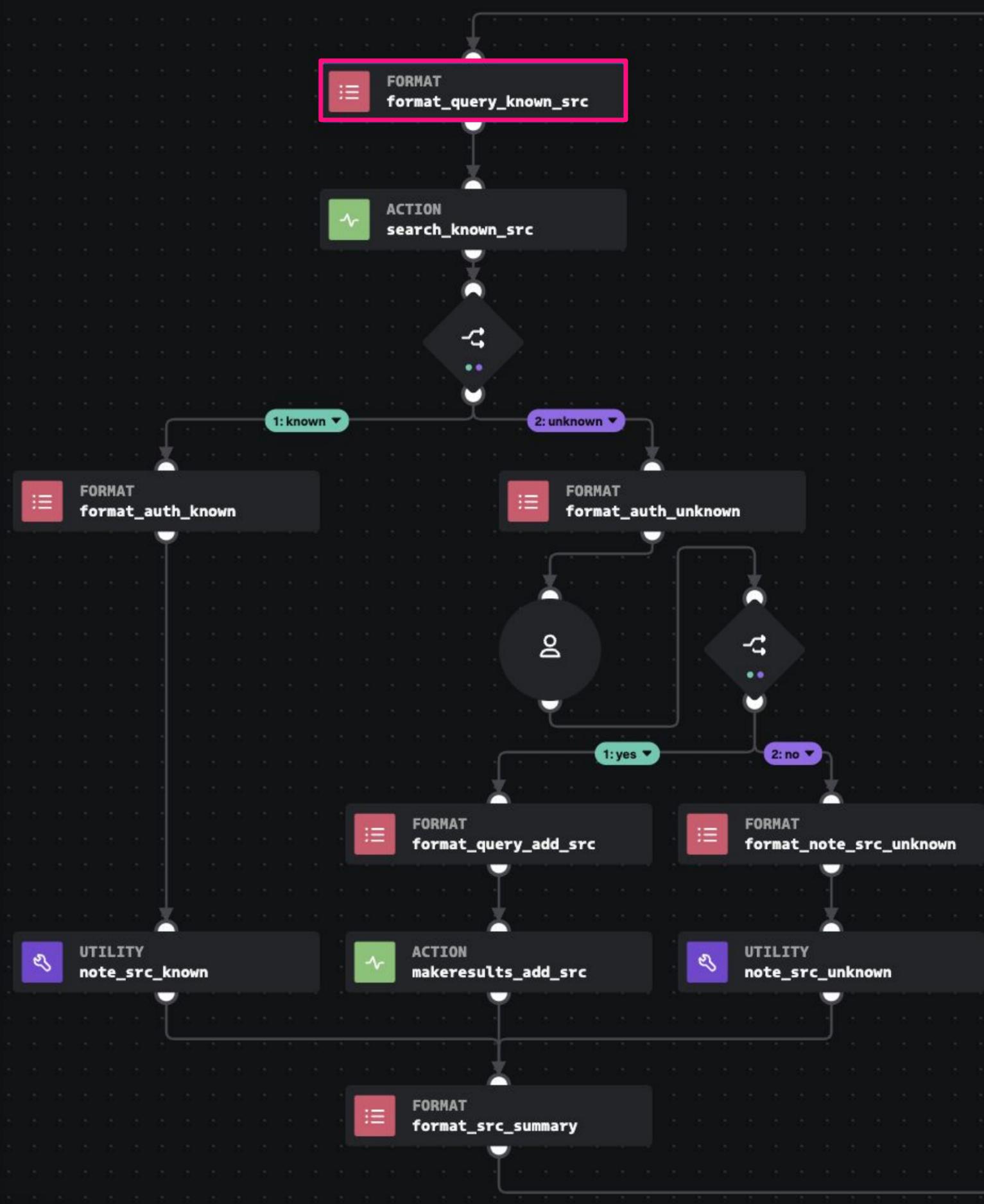
Configure Info Stats

```
count FROM datamodel=Authentication WHERE  
Authentication.user="{0}" AND  
Authentication.src="{1}"  
| lookup known_hosts.csv src AS  
Authentication.src  
| eval description =
```

0 artifact*.cef.suser > x
1 artifact*.cef.src > x +

> ADVANCED

Done



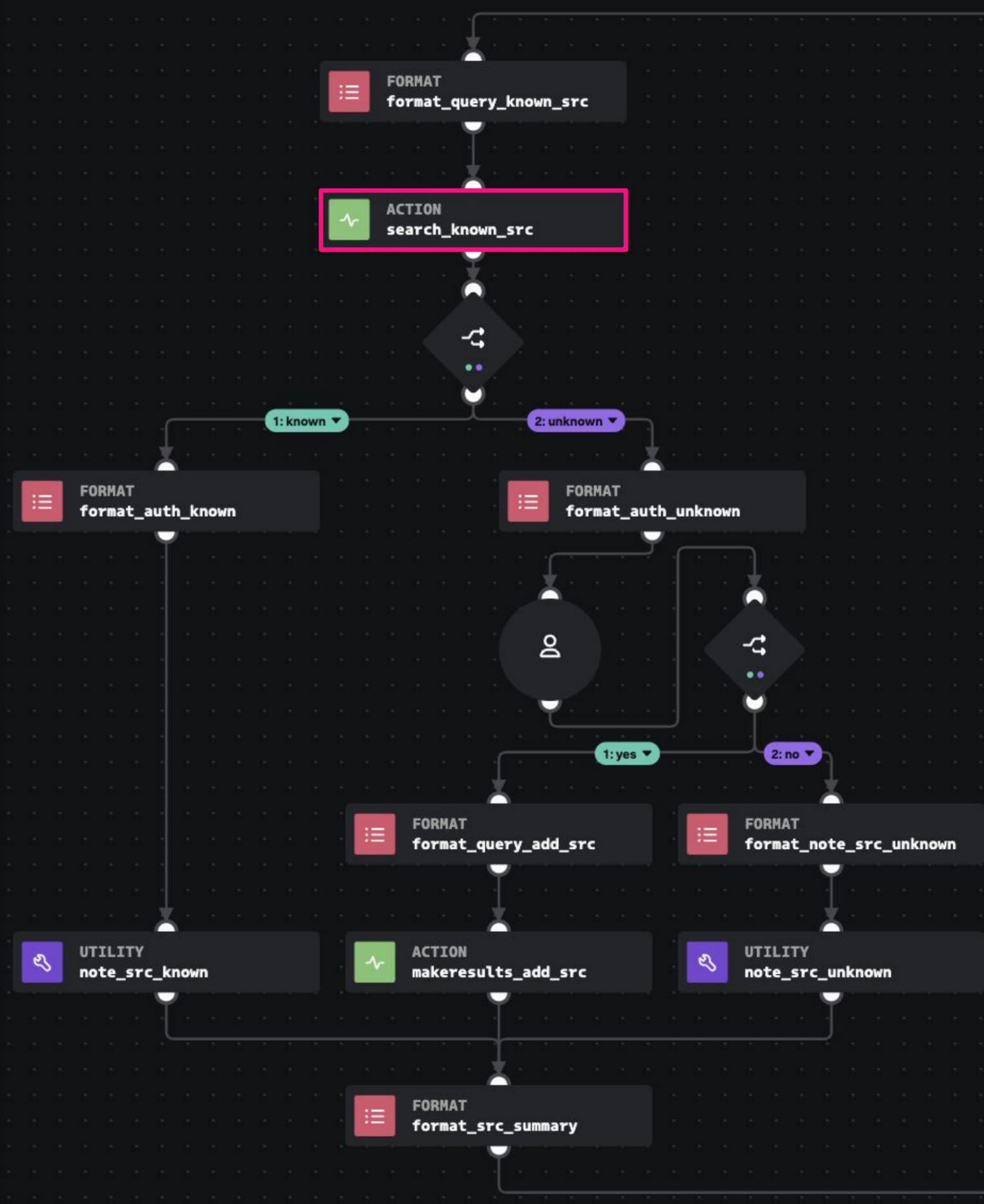
Bonus Tip

Leverage Splunk Lookup Tables via SOAR

Action Block:

Pass this SPL into an action block with a leading | tstats

The screenshot shows the configuration interface for the 'search_known_src' action block. The 'Inputs' section has 'command' set to '| tstats' and 'query*' set to 'format_query_known_src:formatted_data'. The 'display' field is empty. There are checkboxes for 'parse_only' and 'attach_result', both of which are unchecked. The 'start_time' and 'end_time' fields are empty. The 'search_mode' is set to 'smart'. The 'ADVANCED' section is collapsed. A 'Done' button is at the bottom.



Bonus Tip

Leverage Splunk Lookup Tables via SOAR

Searches for the two users from Splunk's perspective.

Note the result field - this creates a binary condition used by the upcoming decision block.

Source Lookup - Admin

```
1 | tstats count FROM datamodel=Authentication WHERE Authentication.user="admin" AND Authentication.src="101.53.17.127"
2 | rename Authentication.src as src
3 | lookup known_hosts.csv src
4 | eval description = if(isnull(description),"unknown", description)
5 | eval result = if(description="unknown",0,1)
6 | table src, description, result
```

✓ 1 result (01/01/1970 00:00:00.000 to 09/07/2025 05:10:03.000) No Event Sampling Job

Events Patterns **Statistics (1)** Visualization

20 Per Page Format Preview

src	description	result
101.53.17.127	VPN Public IP - Sydney	1

Source Lookup - pwny

```
1 | tstats count FROM datamodel=Authentication WHERE Authentication.user="pwny" AND Authentication.src="1.156.14.224"
2 | rename Authentication.src AS src
3 | lookup known_hosts.csv src AS Authentication.src
4 | eval description = if(isnull(description),"unknown", description)
5 | eval result = if(description="unknown",0,1)
6 | table src, description, result
```

✓ 1 result (01/01/1970 00:00:00.000 to 09/07/2025 05:13:32.000) No Event Sampling Job

Events Patterns **Statistics (1)** Visualization

20 Per Page Format Preview

src	description	result
1.156.14.224	unknown	0

Bonus Tip

Leverage Splunk Lookup Tables via SOAR

Decision Block:

Choose a path based on the result returned

DECISION
decide_permitted_src X

Configure Info Stats

CONDITIONS

If 1: known ✎

format_query_known_src:formatted_data.* X

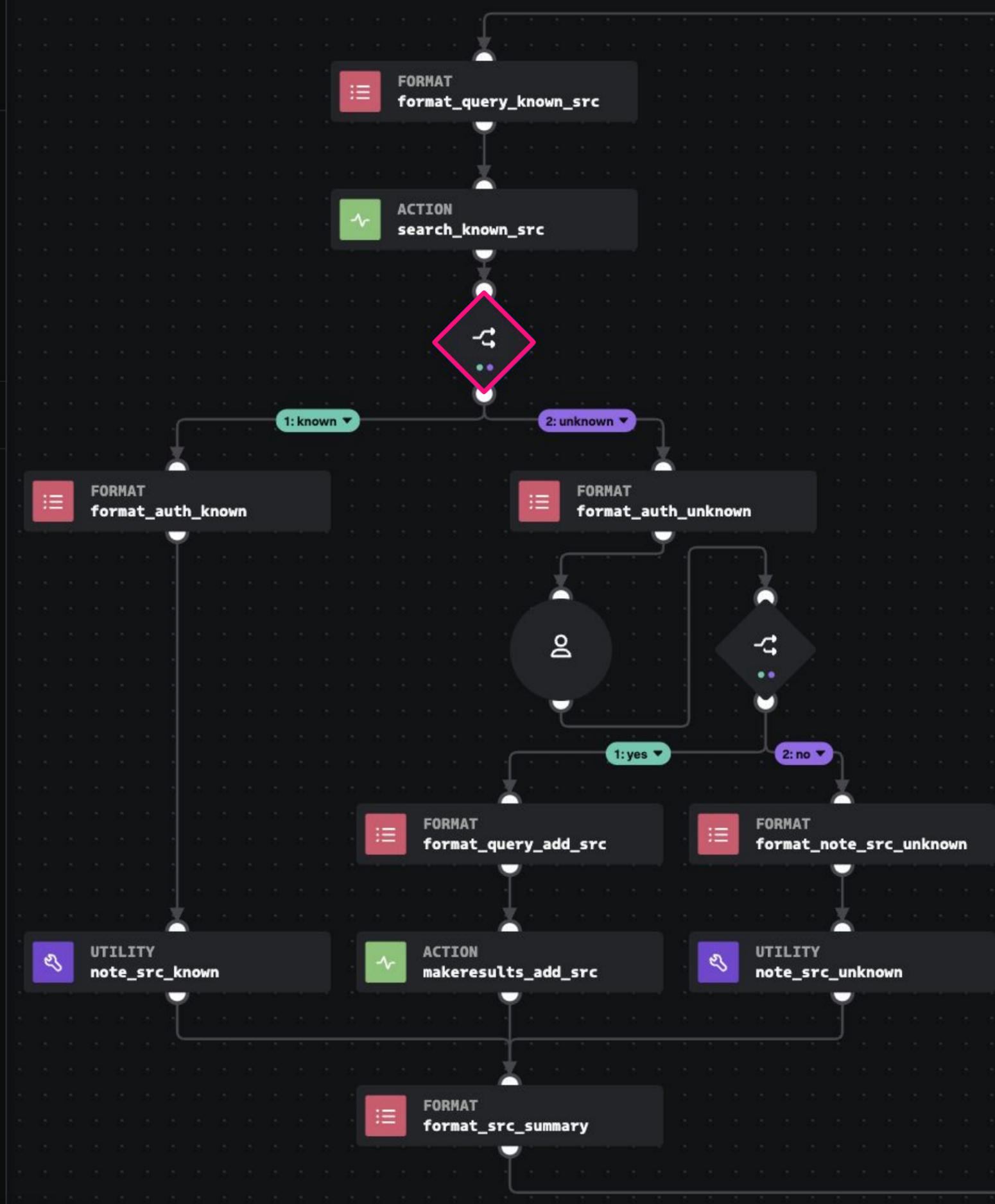
== 1 X

Else 2: unknown ✎

+ Else If

> ADVANCED

Done



Bonus Tip

Leverage Splunk Lookup Tables via SOAR

Prompt Block:

For unknown sources, have a user review and determine if the source is trusted.

PROMPT

prompt_add_known_src [X]

Configure Info Stats

User or Role

Event owner ⓘ

User '{0}' has accessed the network from an unknown source address '{1}'

0 artifact*.cef.src [X] [+]

QUESTION(S)

Question 1

Add new source to known host list?

Response Type

Yes/No

Delete

Question 2

Add a description of this source:

Response Type

Message

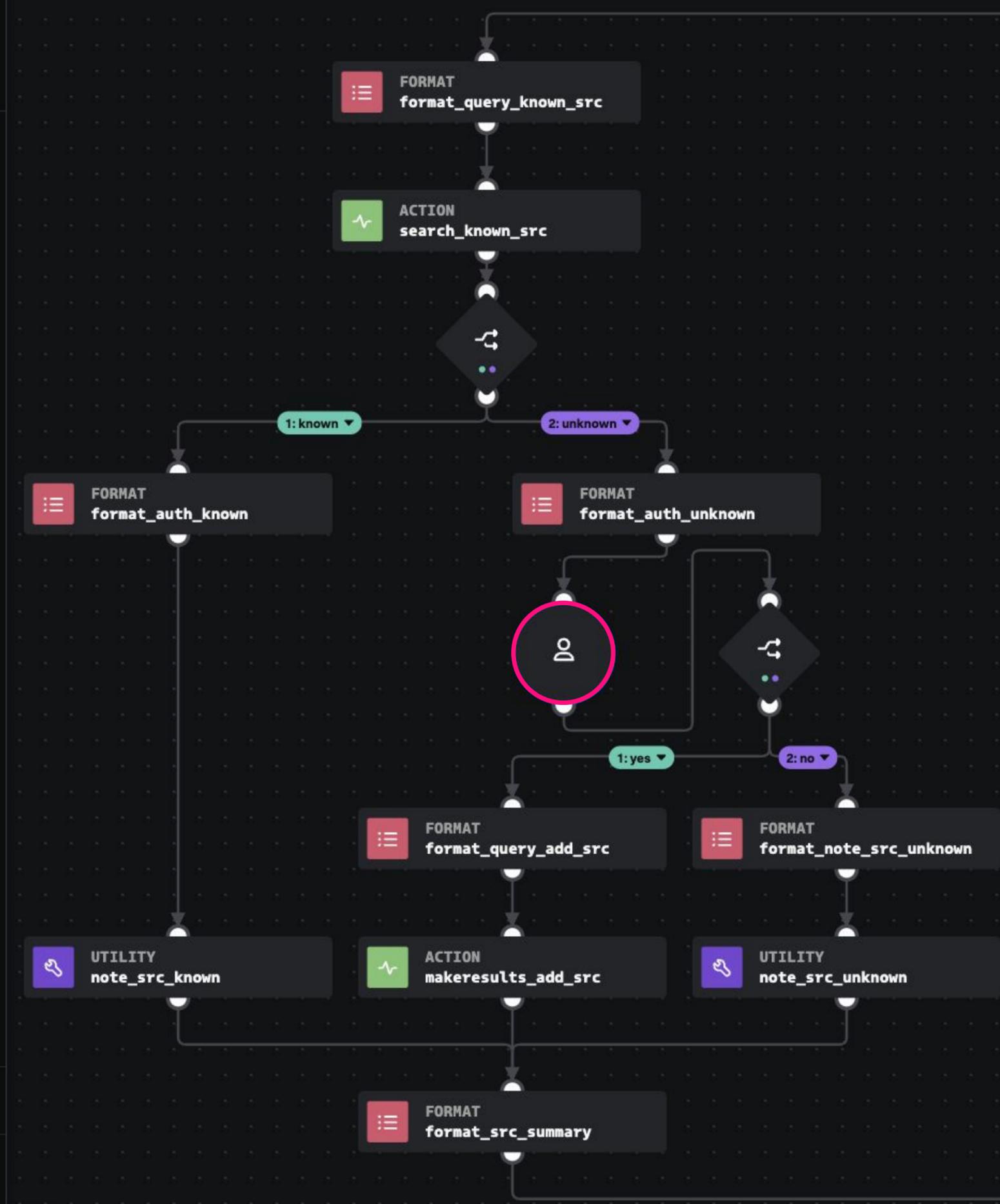
Delete

+ Question

Required response time (mins) 30

> ADVANCED

Done



Bonus Tip

Leverage Splunk Lookup Tables via SOAR

Decision Block:

Choose a path based on the prompt response

DECISION
decision 5

Configure Info Stats

CONDITIONS

If 1: yes

prompt_add_known_src:action_result.sumr

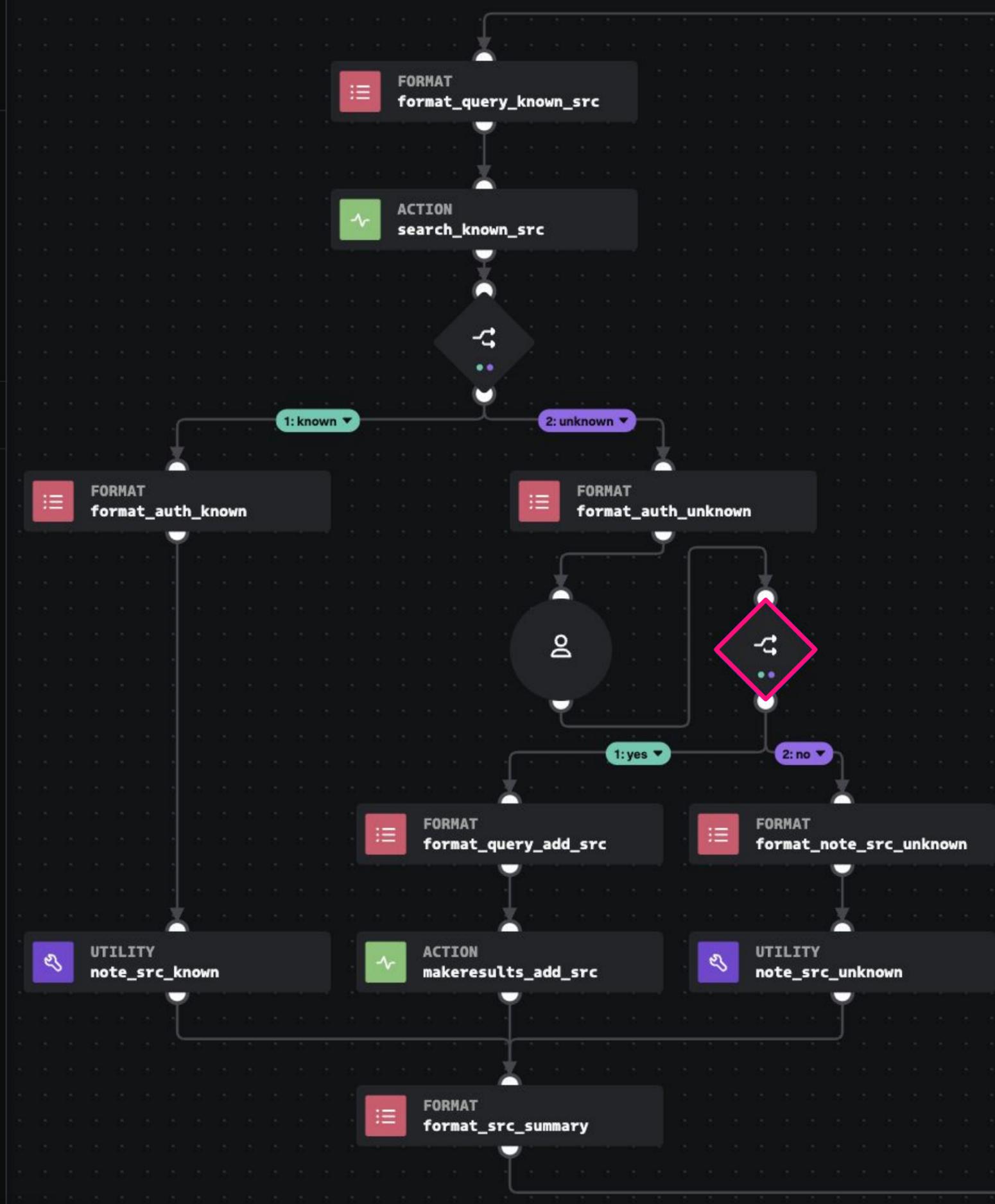
== Yes

Else 2: no

+ Else If

> ADVANCED

Done



Bonus Tip

Leverage Splunk Lookup Tables via SOAR

Format Block:

Write your SPL to create a search that will output into the lookup table.

FORMAT
format_query_add_src

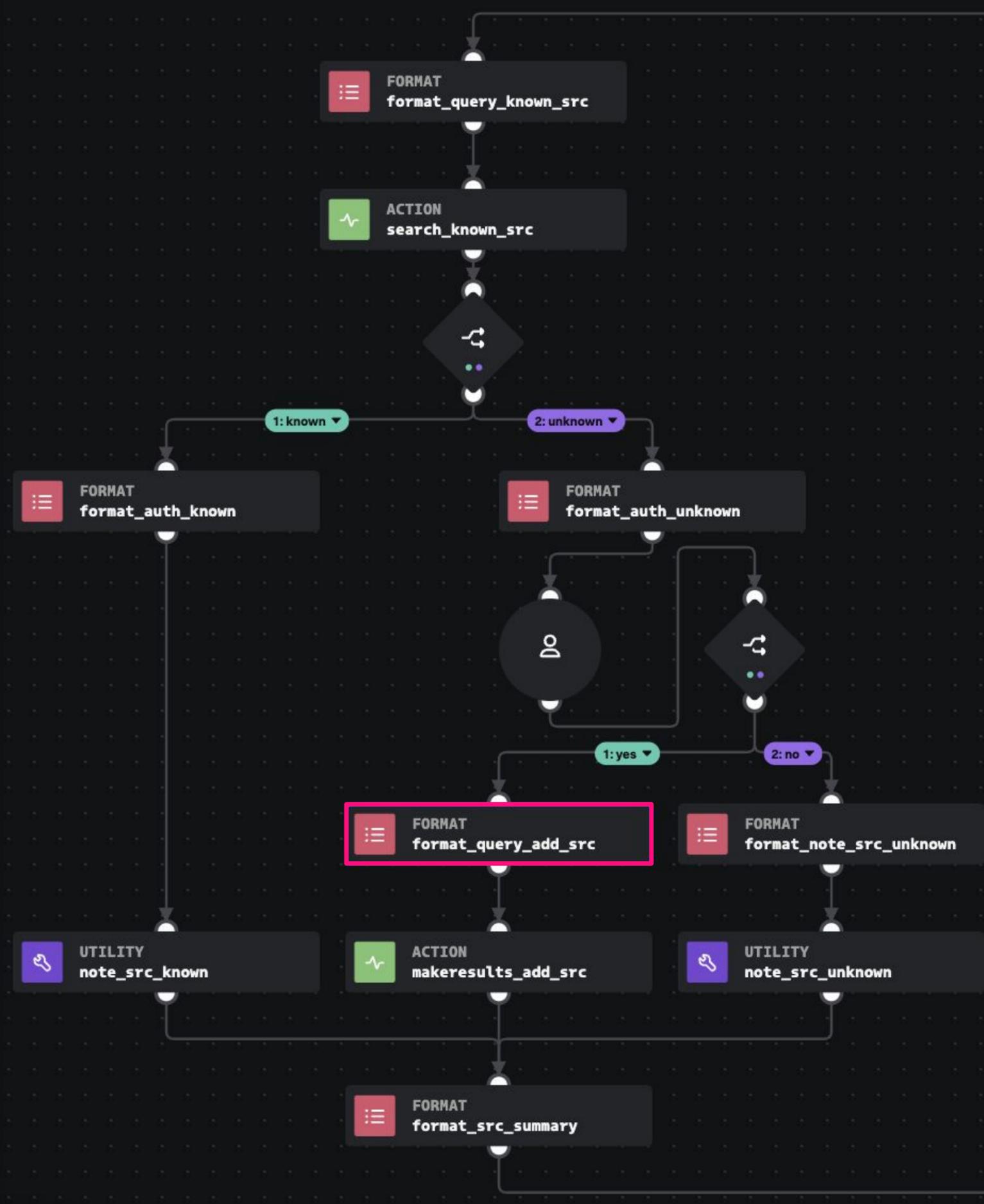
Configure Info Stats

```
| eval src="{0}", description="{1}"  
| table src, description  
| outputlookup append=true known_hosts.csv
```

0 artifact*.cef.src > x
1 prompt_add_known_src > x +

> ADVANCED

Done



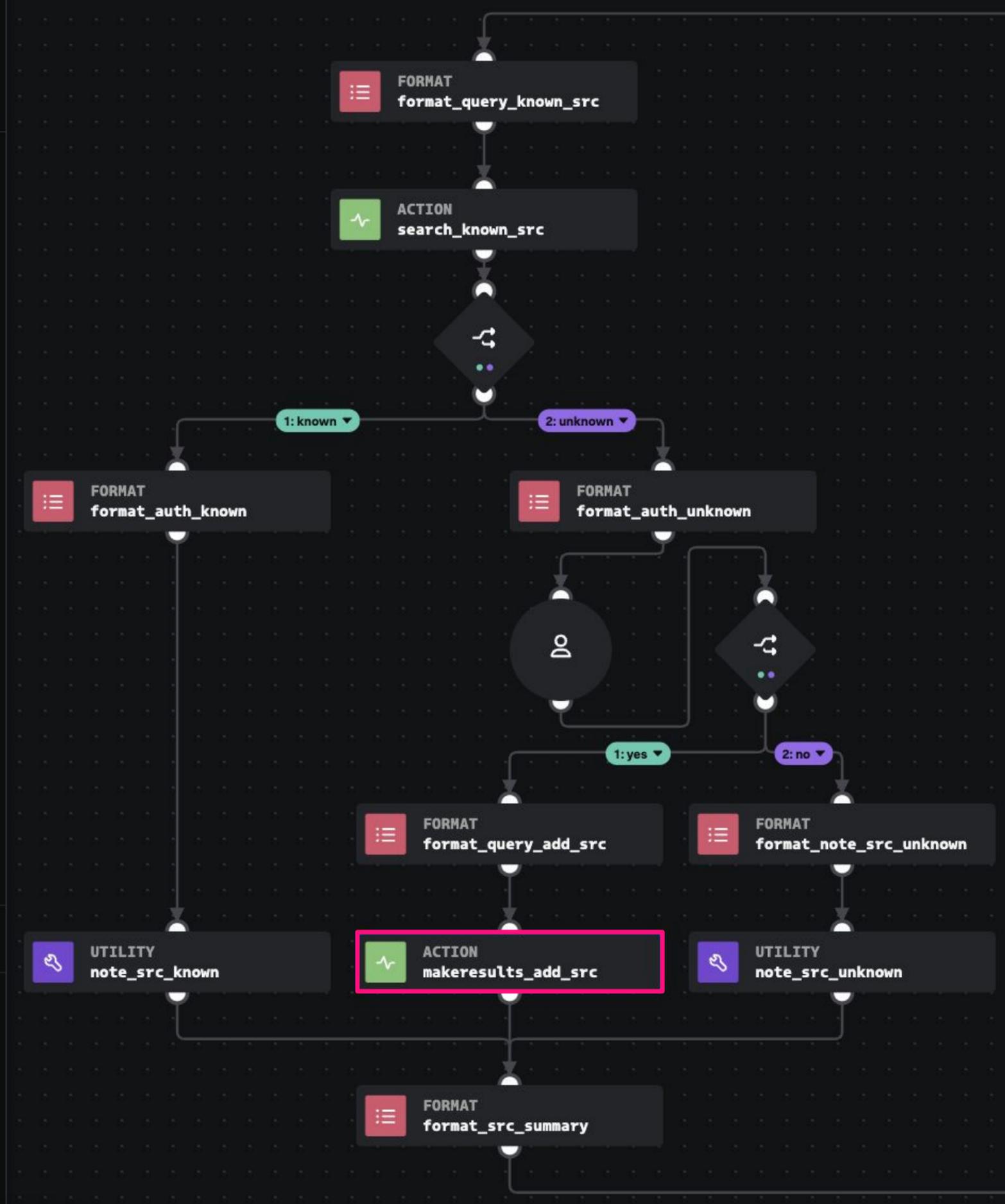
Bonus Tip

Leverage Splunk Lookup Tables via SOAR

Action Block:

Pass this SPL into an action block with a leading |
makeresults

The screenshot shows the configuration interface for the 'makeresults_add_src' action block. The 'command' field is set to '| makeresults' and the 'query*' field is set to 'search_known_src:action_result.parameter.que'. Other fields like 'display', 'parse_only', 'attach_result', 'start_time', 'end_time', and 'search_mode' are also visible.



Bonus Tip

Leverage Splunk Lookup Tables via SOAR

Equivalent SPL within Splunk, plus SOAR output via a note:

Data has now been written in a Splunk lookup table from SOAR with zero custom code!

The screenshot shows the Splunk interface for updating a lookup table from SOAR. The top navigation bar includes 'splunk>enterprise', 'Apps', 'Administrator', 'Messages', 'Settings', 'Activity', 'Help', and 'Find'. Below this is a secondary navigation bar with 'Search', 'Analytics', 'Datasets', 'Reports', 'Alerts', and 'Dashboards', along with a 'Search & Reporting' button. The main heading is 'Update lookup from SOAR', with 'Save As', 'Create Table View', and 'Close' buttons. A code editor contains the following SPL:

```
1 | makeresults
2 | eval src="1.156.14.224", description="Hyperion3 Mobile Hotspot"
3 | table src, description
4 | outputlookup append=true known_hosts.csv
```

Below the code editor, it shows '1 result (01/01/1970 00:00:00.000 to 09/07/2025 08:27:11.000)' and 'No Event Sampling'. The 'Statistics (1)' tab is active, showing a table with columns 'src' and 'description'. The table contains one row:

src	description
1.156.14.224	Hyperion3 Mobile Hotspot

Below the table, there is a 'NOTES (1)' section with a search bar. A note is displayed, created by 'soar_local_admin' on 'Jul 10, 2025 4:57 am'. The note content is:

Container Summary:

User 'pwny'
Source '1.156.14.224'
Add to lookup 'yes'
Lookup Description 'Hyperion Mobile Hotspot'

[Show Less](#)

Bonus Tip

Leverage Splunk Lookup Tables via SOAR

Contents of the updated lookup table, showing the new entry from SOAR.

The screenshot shows the Splunk interface for an updated lookup table. The search bar at the top is highlighted in pink and contains the text '1 | inputlookup known_hosts.csv'. Below the search bar, the interface shows '3 results' for the time range '01/01/1970 00:00:00.000 to 09/07/2025 08:33:05.000'. The table below has two columns: 'description' and 'src'. The row 'Hyperion3 Mobile Hotspot' with the IP address '1.156.14.224' is highlighted in orange.

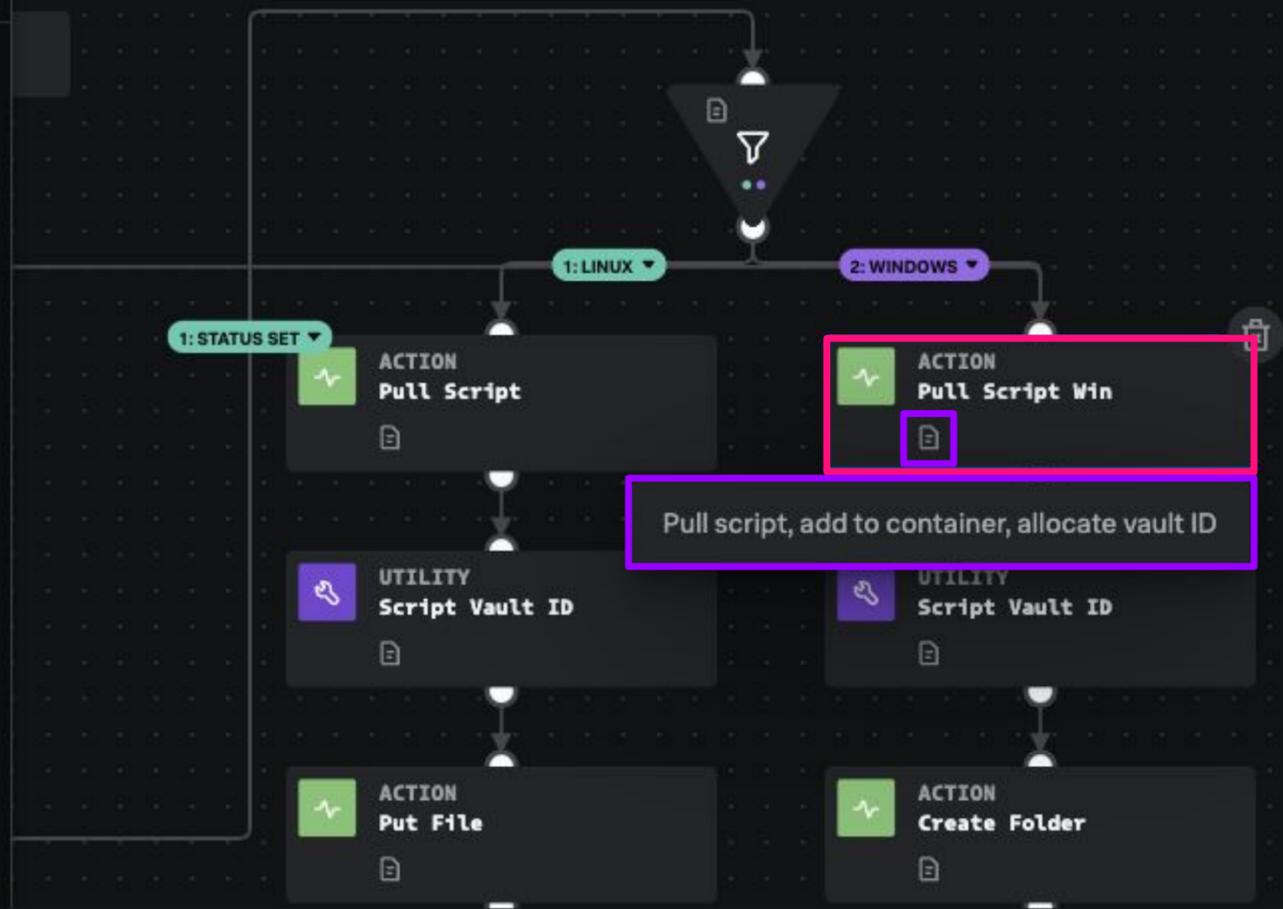
description	src
VPN Public IP - Melbourne	101.53.23.127
VPN Public IP - Sydney	101.53.17.127
Hyperion3 Mobile Hotspot	1.156.14.224

Custom Name
 Pull Script Win

Description (code comment)
 This action block will pull the Splunk Universal Forwarder script from the SOAR machine and add as a file to the container and allocate a vault ID.
 Script type: powershell
 Default path: /tmp/win/script_win.ps1
 The Security Engineering team are responsible for maintaining of the install script. See <<REDACTED>> as the key point of contact for details or troubleshooting.

Notes (block tooltip)
 Pull script, add to container, allocate vault ID

pull_script_win Function Copy



```

91 @phantom.playbook_block()
92 def pull_script_win(action=None, success=None, container=None, results=None, handle=None, filtered_artifacts=None, filtered_results=None)
93     phantom.debug("pull_script_win() called")
94
95     # phantom.debug('Action: {0} {1}'.format(action['name'], ('SUCCEEDED' if success else 'FAILED')))
96
97     #####
98     # This action block will pull the Splunk Universal Forwarder script from the SOAR
99     # machine and add as a file to the container and allocate a vault ID.
100    #
101    # Script type: powershell
102    #
103    # Default path: /tmp/win/script_win.ps1
104    #
105    # The Security Engineering team are responsible for maintaining of the install
106    # script.
107    # See <<REDACTED>> as the key point of contact for details or troubleshooting.
108    #
109    #####
110
111    parameters = []
112
  
```

Bonus Tip

Documentation as Code

Save time by documenting automation as you build!

- Set a custom name to each block, this defines the function name
- Add a detailed description to each block, this becomes living documentation within the code
- Configure a note, this provides a tooltip to assist development / code reuse / debugging

Bonus Tip

Floating Debugger

Keep a 'floating debugger' close to wherever you are actively developing a playbook.

Name, description, and notes as per good coding practices.

UTILITY
floating debugger
debug

Configure Info Stats Loop

Custom Name
floating debugger

Description (code comment)
Floating debugger to attach inline for debugging

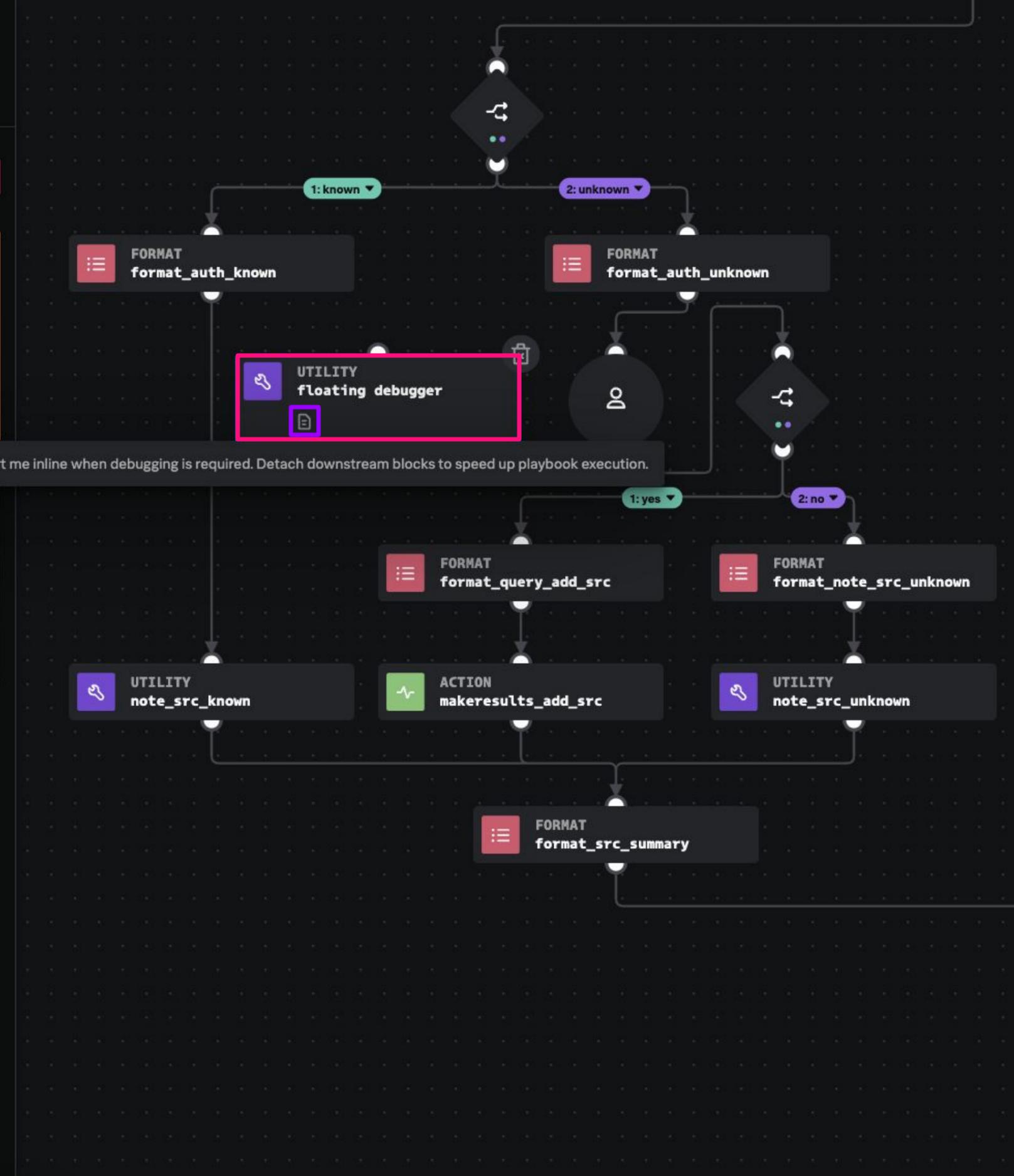
Notes (block tooltip)
Insert me inline when debugging is required.
Detach downstream blocks to speed up playbook execution.

[View Custom Function](#)

Repo
community

Description
Print debug messages with the type and value of 0-10 different inputs. This is useful for checking the values of input data or the outputs of other playbook blocks.

Inputs
input_1 (*)
input_2 (*)
input_3 (*)
input_4 (*)
input_5 (*)
input_6 (*)
input_7 (*)
input_8 (*)
input_9 (*)
input_10 (*)



Bonus Tip

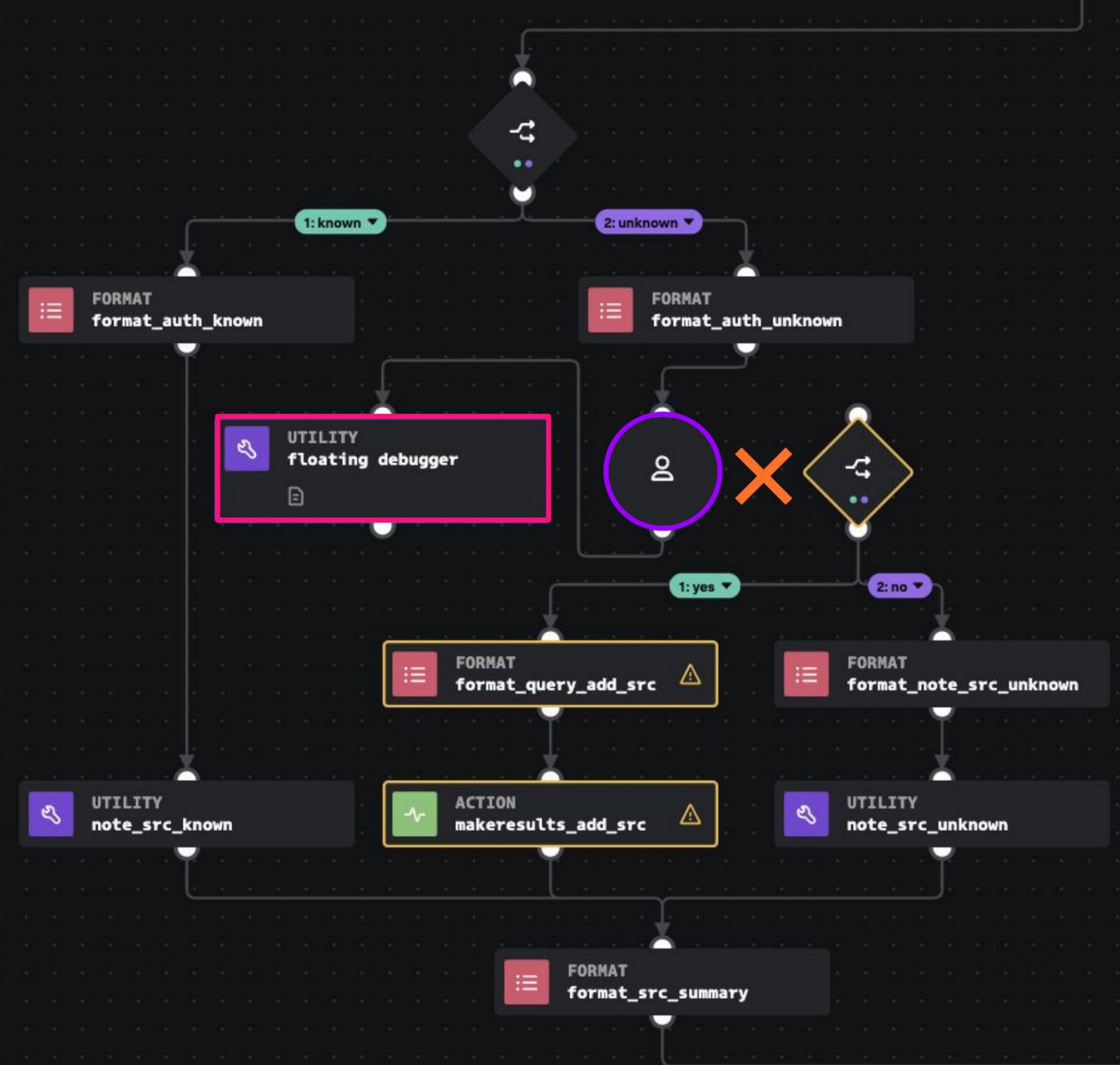
Floating Debugger

- Connect in-line when running the playbook debugger
- Disconnect downstream blocks to speedup execution
- Add up inputs in bulk in order to locate the fields relevant to your playbook

The screenshot shows the 'floating debugger' configuration window. At the top, there are tabs for 'Configure', 'Info', 'Stats', and 'Loop'. Below these, the configuration is organized into 'input' fields:

- input_1: `_add_known_src:action_result.status`
- input_2: `_src:action_result.parameter.message`
- input_3: `_src:action_result.summary.responses.0`
- input_4: `_src:action_result.summary.responses.1`
- input_5: (empty)
- input_6: (empty)
- input_7: (empty)
- input_8: (empty)
- input_9: (empty)
- input_10: (empty)

At the bottom, there is an 'ADVANCED' section with a 'Done' button.



FORMAT
format_auth_known

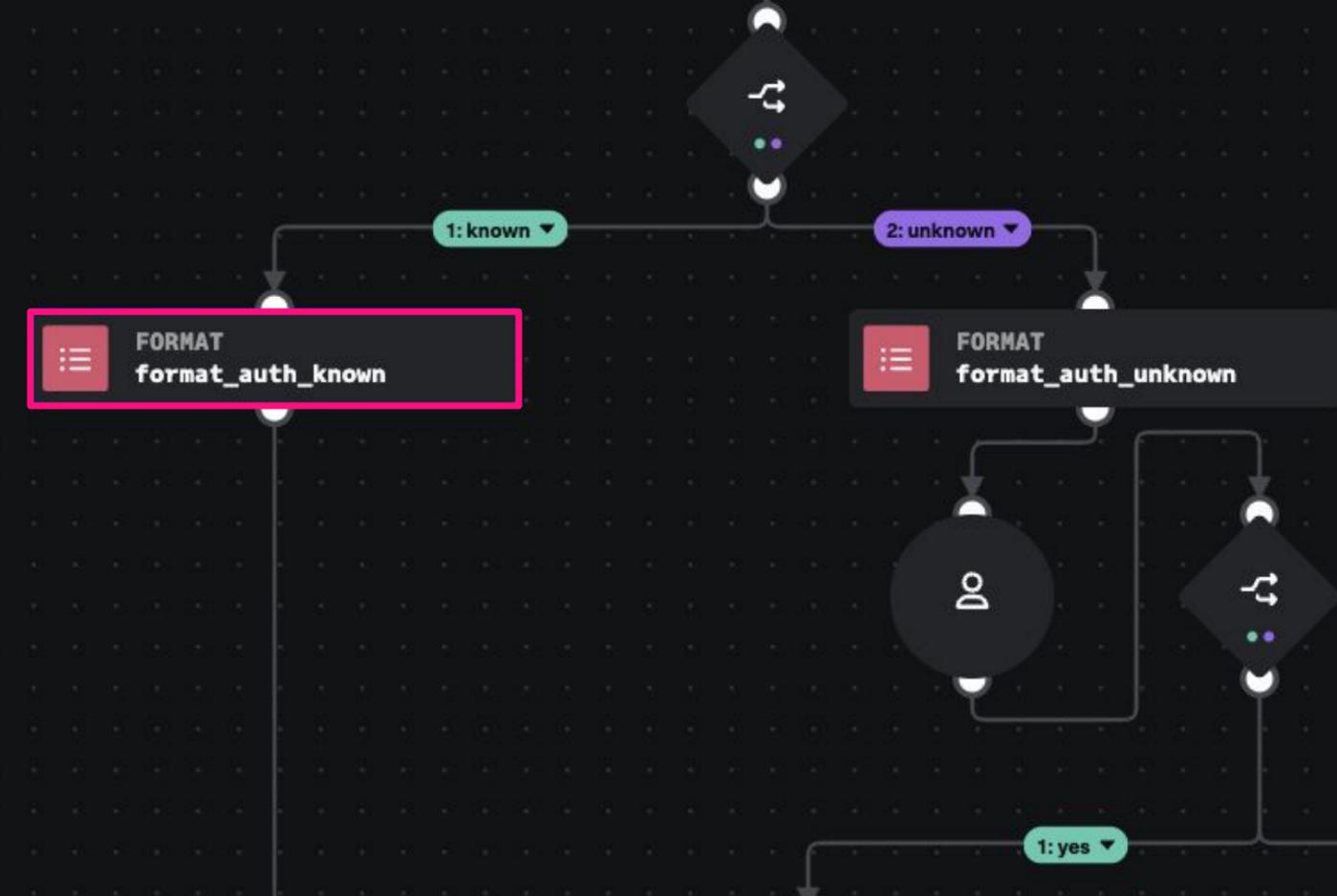
Configure Info Stats

****Container Summary:****

ID: {0}
user	source
{1}	{2}

- 0 container.id
- 1 artifact*.cef.suser
- 2 artifact*.cef.src

ADVANCED



Widgets Notes

NOTES (1) Search notes

Show All Notes by Newest

Note title

Write Preview

****Container Summary:****
ID: 1024

USER	SOURCE
pwny	1.156.14.224

Cancel Save

MANAGE WIDGETS

Markdown supported

- **Strong**** Strong
- *Emphasis** Emphasis
- ~~~~Strike~~~~ Strikethrough
- [Text Link](URL) Text Link

Widgets Notes

NOTES (1) Search notes

Note title

Write Preview

Container Summary:
ID: 1024

USER SOURCE
pwny 1.156.14.224

Bonus Tip

Markdown Formatting

- Format blocks support markdown input, allowing for text formatting / tables
- The same applies for notes, where you can even insert images!
- Use the inbuilt markdown cheatsheet to get started
- Use note preview to test and adjust

Wrap Up

Today we covered

- Refresher of some Part 1 content
- Example of out of the box thinking
- Example of remote code/script execution
- Tips and good practice

Watch Part 1 if you missed it!

Hyperion3 SOAR Git Repo

https://github.com/MattHyp3/Hyperion3_SOAR



Questions?

Mic is open!

Or, come and see us afterwards /
reach out to us at:

rich@splunk.com

matt@hyperion3.com.au



Thank you

