Forward-looking statements

This presentation may contain forward-looking statements that are subject to the safe harbors created under the Securities Act of 1933, as amended, and the Securities Exchange Act of 1934, as amended. All statements other than statements of historical facts are statements that could be deemed forward-looking statements. These statements are based on current expectations, estimates, forecasts, and projections about the industries in which we operate and the beliefs and assumptions of our management based on the information currently available to us. Words such as “expects,” “anticipates,” “targets,” “goals,” “projects,” “intends,” “plans,” “believes,” “momentum,” “seeks,” “estimates,” “continues,” “endeavors,” “strives,” “may,” variations of such words, and similar expressions are intended to identify such forward-looking statements. In addition, any statements that refer to (1) our goals, commitments, and programs; (2) our business plans, initiatives, and objectives; and (3) our assumptions and expectations, including our expectations regarding our financial performance, products, technology, strategy, customers, markets, acquisitions and investments are forward-looking statements. 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. Readers are cautioned that these forward-looking statements are only predictions and are subject to risks, uncertainties, and assumptions that are difficult to predict, including those identified in the “Risk Factors” section of Cisco’s most recent report on Form 10-Q filed on May 21, 2024 and its most recent report on Form 10-K filed on September 7, 2023. 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 Cisco or Splunk, on Cisco or Splunk’s website or otherwise, it may not contain current or accurate information. Cisco and Splunk undertake no obligation to revise or update any forward-looking statements for any reason, except as required by law.

In addition, any information about new products, features, functionality or 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 or be relied upon in making a purchasing decision. We undertake no commitment, promise or 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. The development, release, and timing of any features or functionality described for our products remains at our sole discretion.

splunk> .conf24

Practical SOAR Examples from the Field

SEC1579B

Overview

Session Breakdown and Goals

Breakdown

- Introductions
- Splunk® SOAR refresher
- Deployment methodology
- Playbook case studies + tips
- Wrap up / questions

Goals

- Inspire you to confidently handle security automation
- Provide a model to approach SOAR playbook development
- Show you real-world playbooks from the field
- Leave you with new founded ideas, tips and tricks on creating your own playbooks

Introductions

Richard Hampshire

Security Architect - Professional Services

Splunk

rich@splunk.com

Matthew Bennett

Managing Director / Core Consultant

Hyperion3

matt@hyperion3.com.au

.conf24

splunk>

SOAR Intro / Refresher

SOAR Intro / Refresher

Session Level: Advanced

- What is SOAR?
- Why do we need SOAR?
- What SOAR is not…

SOAR Implementation Methodology

SOAR Implementation

Challenges

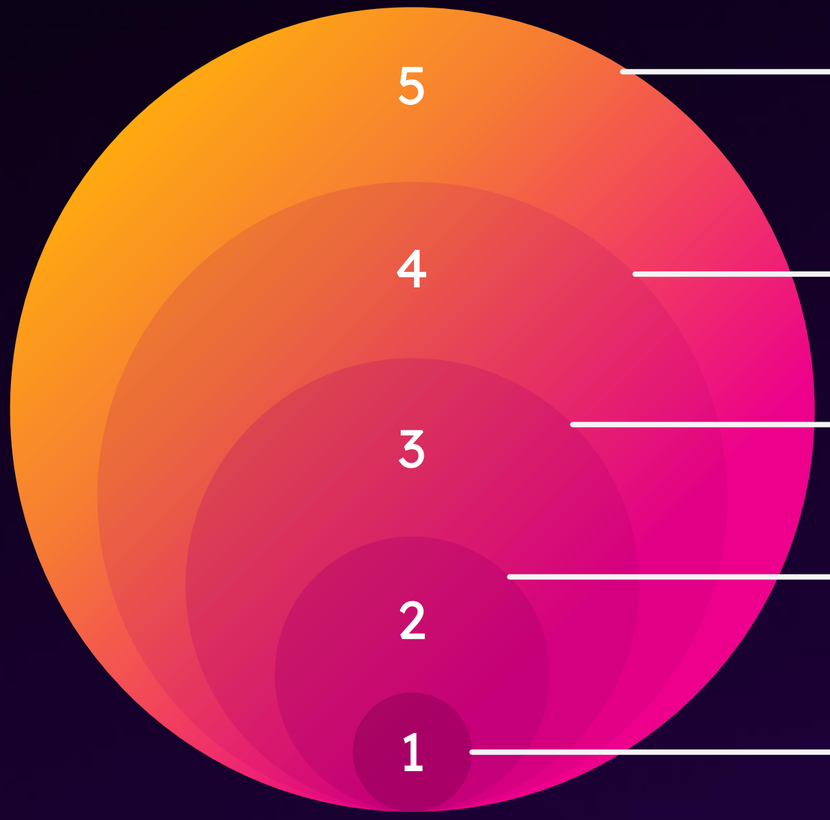
- The typical ‘all or nothing’ approach leads to a failed implementation
  - Shooting straight for complex automation of alert triage, investigation, determination, and response, all at once leads to immediate roadblocks and limits adoption of the platform
- Often SOCs that need SOAR the most are the ones too small or under resourced to invest in an automation platform
  - Common misconception that only large and mature SOCs will benefit from a SOAR
- Lack of defined processes erodes the value of a SOAR
  - You can’t automate something that doesn’t yet exist
- Gaps in key often skills undermine even a well approached deployment
  - Custom code is required for advanced playbook development, which many teams lack in-house

SOAR Implementation

Solutions

- Apply a staged approach to the deployment, with a focus on smaller achievable steps that provide immediate value to the SOC
  - 5 stages structured to minimise road-blocks and common challenges
- Integrate with systems under your control
  - Ignore the corporate Active Directory or core firewalls that you'll never get access to anyway
  - Start with the Splunk instance already under your control, and develop playbooks to automate investigations
- Build smaller, simpler automations
  - Something as basic as automatically logging a ticket / service request sounds boring,
    but once implemented the cumulative time saved is immense and reclaims time for actual analytics

Best Practice SOAR Implementation Methodology



5. Enterprise Systems
Active Directory, firewalls, email

4. External Services
Reputation lookup APIs

3. SOC Tooling
ITSM, EDR, scanning

2. Splunk Integrated
Ingest alerts & automate searches

1. Standalone Instance
Case management system

“How should I use SOAR?”

The consultants’ catch-all, “*It depends…*”

Examples of SOAR Integration

Where are your analysts going to work?

Splunk SOAR

- ES notables are sent to SOAR
- Other data sources fed in (email, cloud services etc)
- Automations and enrichment performed automatically or on demand

Splunk® Enterprise Security

- Splunk ES for day to day case management
- SOAR may be used ad hoc
- Automations and enrichment performed on demand

3rd Party System

- An external ticketing system being used
- Triage playbooks
- Centralisation of different systems

Playbook Case Studies

ES Notable Onboarding

Type 2 Splunk Integrated

Common for customers that are using SOAR as primary case management

- Links ES and SOAR cases together with pivot URLs and comments
- Runs automatically on any notables sent to SOAR

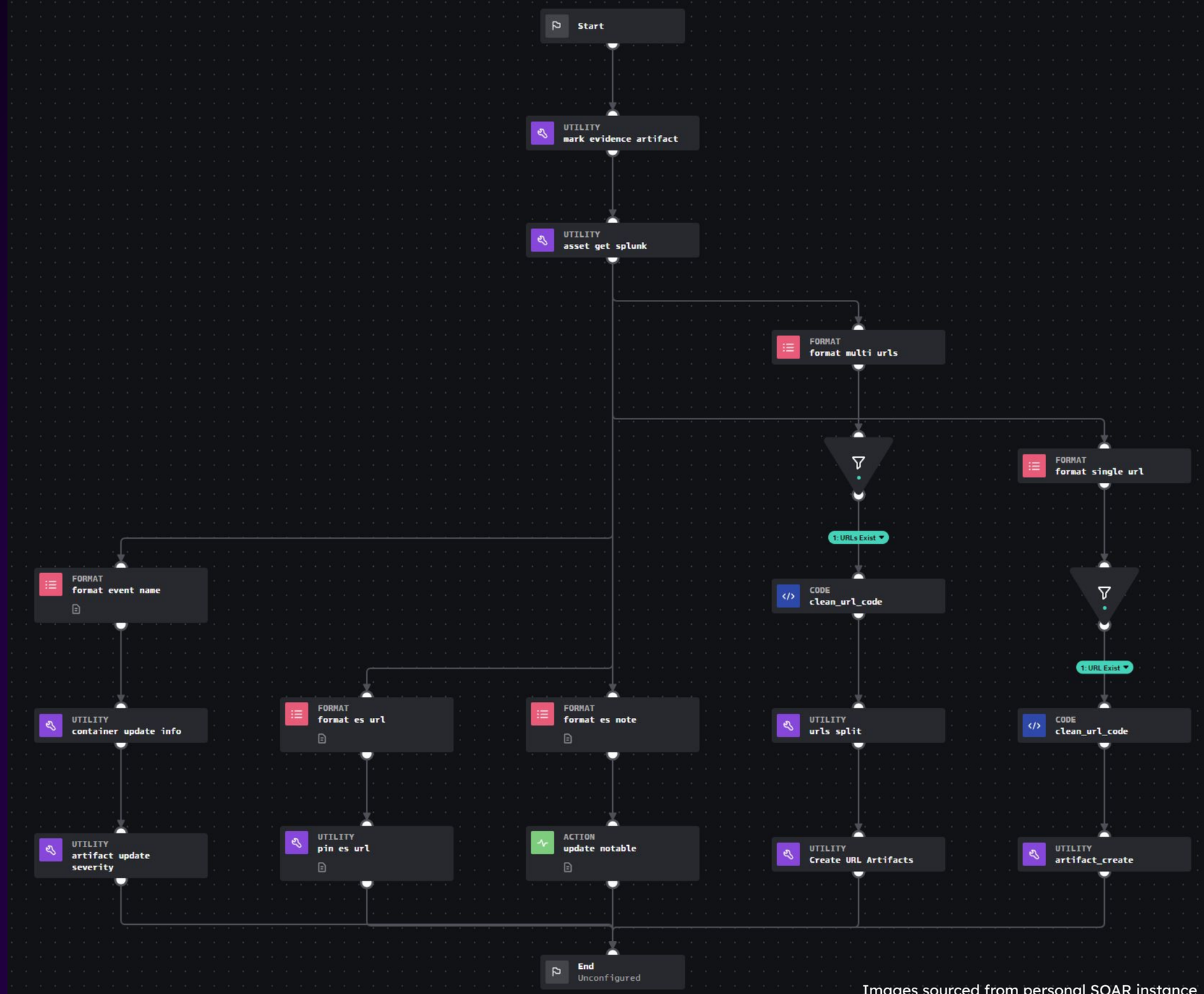
ES Notable Onboarding

Simple SPL using notable macro to gather and send notables to SOAR using scheduled searches

```
`notable`
| eval rule_name=if(isnull(rule_name),source,rule_name)
| eval rule_title=if(isnull(rule_title),rule_name,rule_title)
| `get_urgency`
| `risk_correlation`
| eval rule_description=if(isnull(rule_description),source,rule_description)
| eval security_domain=if(isnull(security_domain),source,security_domain)
| eval
es_pivot=https://splunkes.local:8000/en-US/app/SplunkEnterpriseSecuritySuite/incident_review?search=event_hash%3D+event_hash
| expandtoken
| table *
| sendalert sendtophantom param.phantom_server="SOAR" param.sensitivity="amber" param.severity="low" param.label="events"
```

ES Notable Onboarding

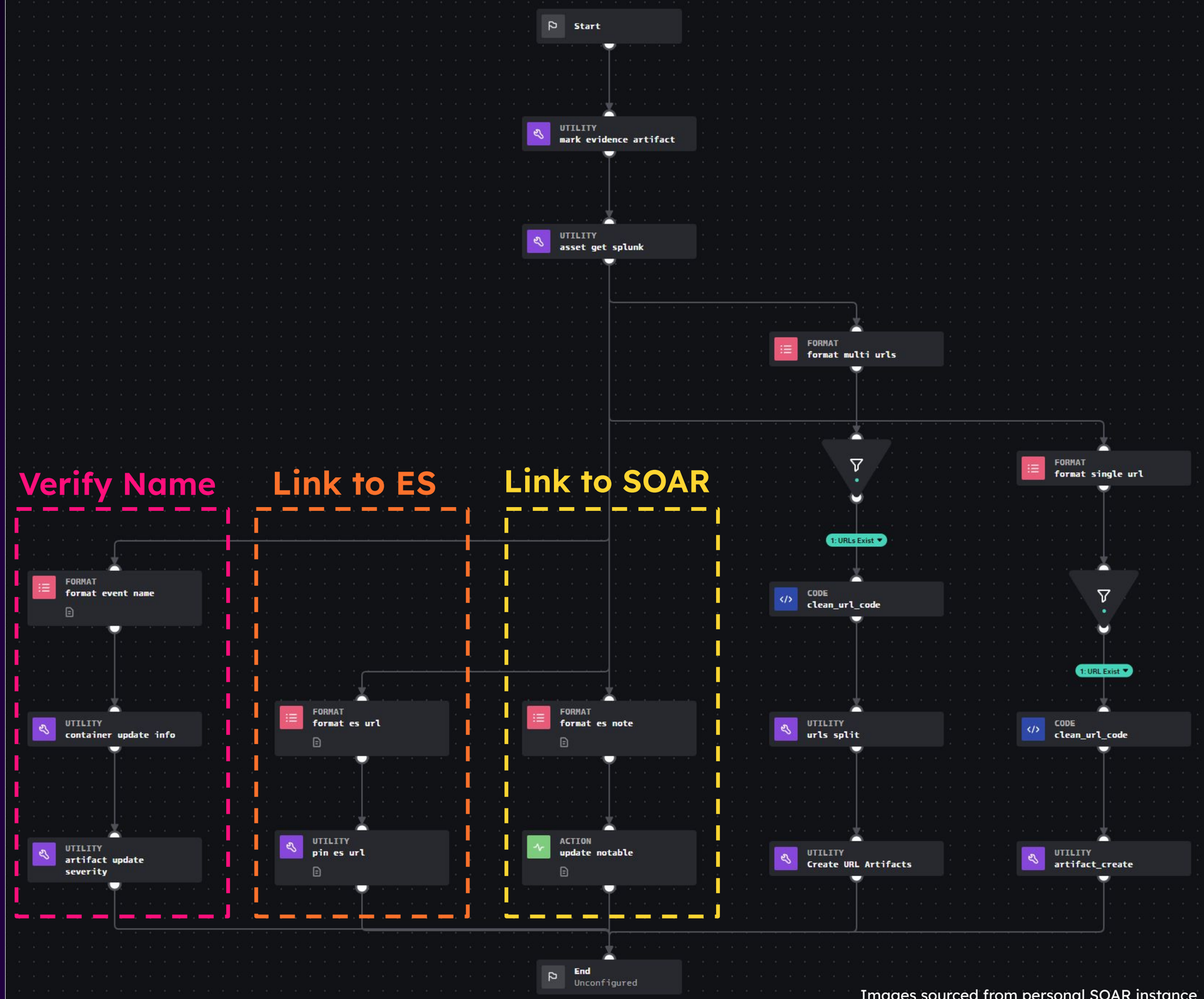
Example on how to link the ES notable and SOAR container together



Images sourced from personal SOAR instance

ES Notable Onboarding

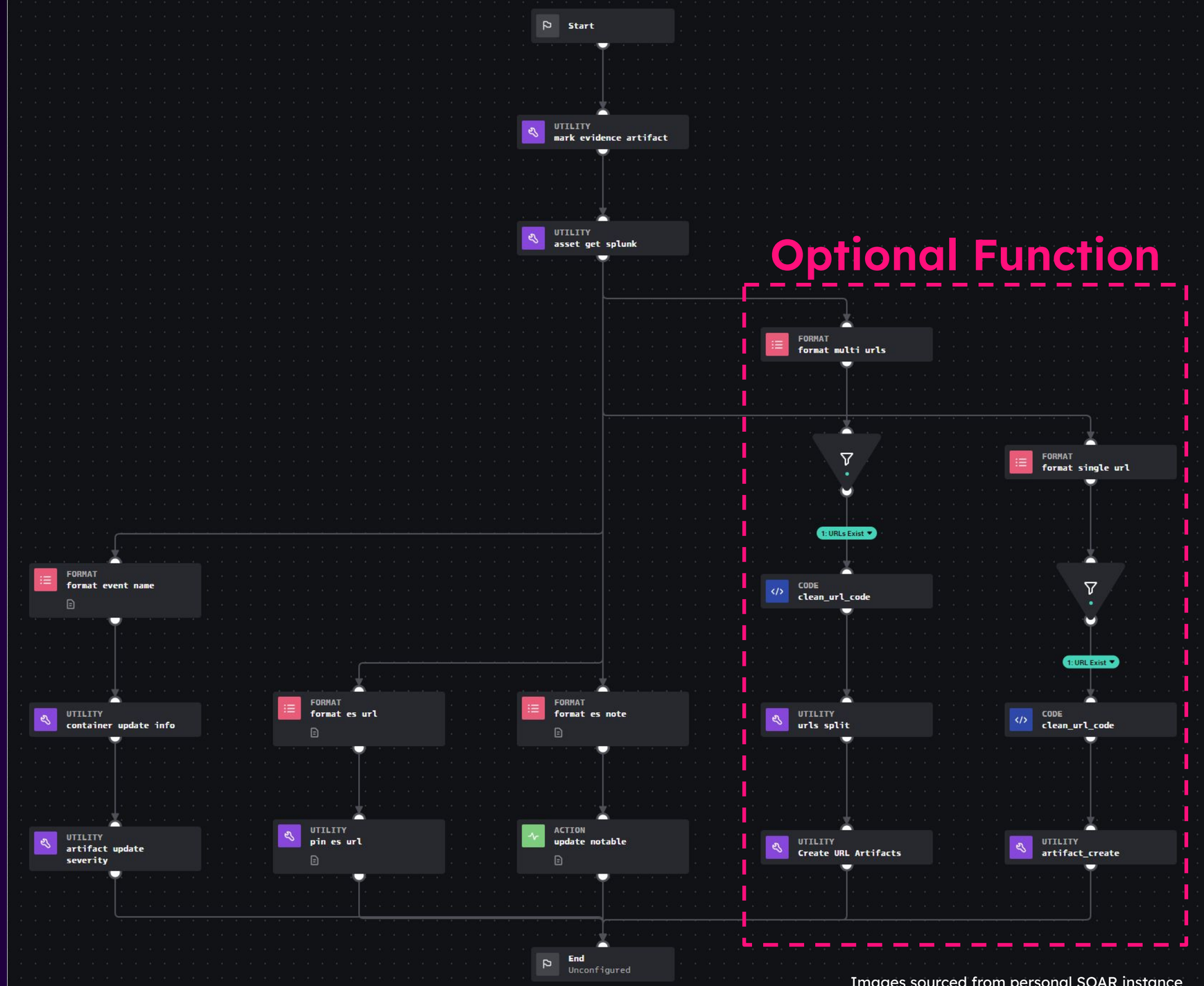
Linking parts



Images sourced from personal SOAR instance

ES Notable Onboarding

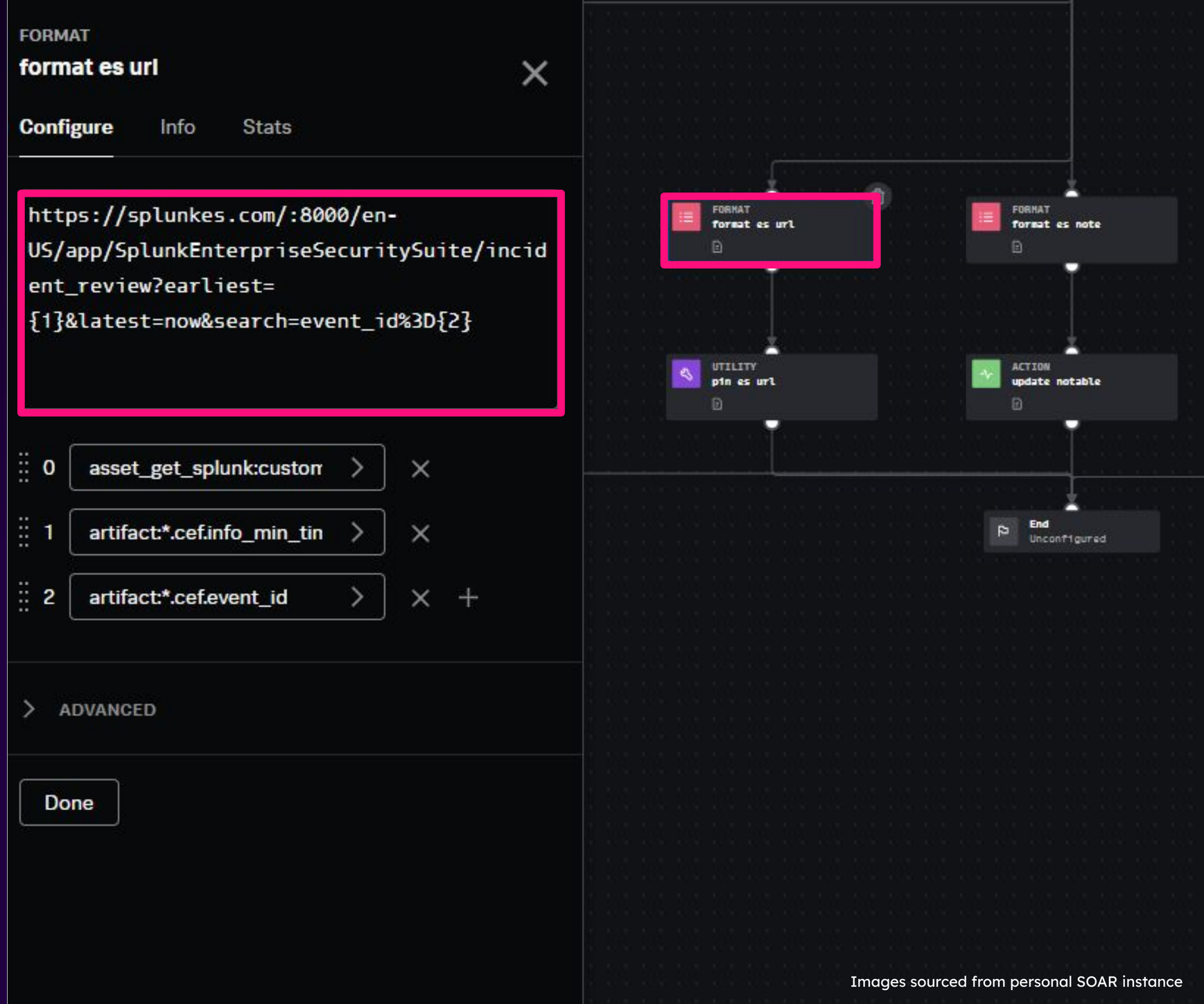
Optional side path created here to perform required URL cleaning.



Images sourced from personal SOAR instance

ES Notable Onboarding

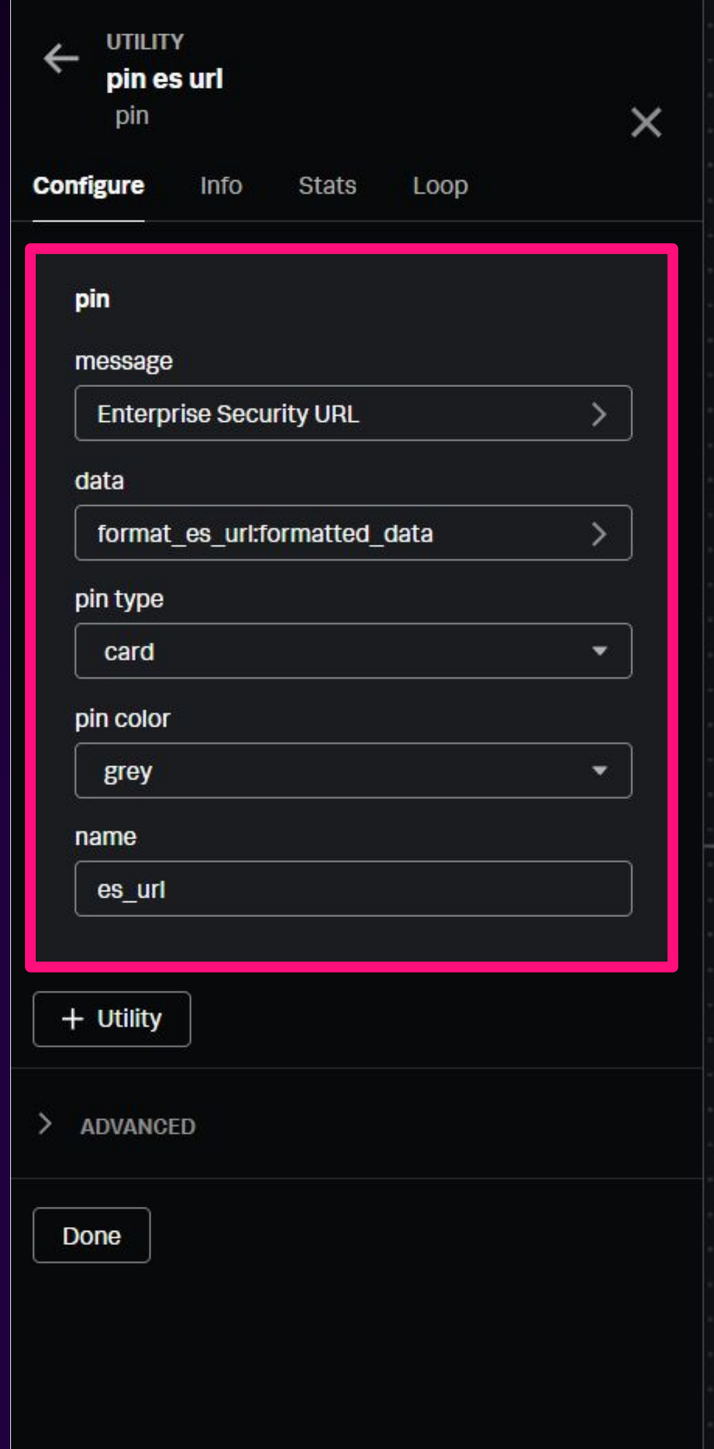
Format Block:
Creates a URL to your ES instance and injects ES EventID

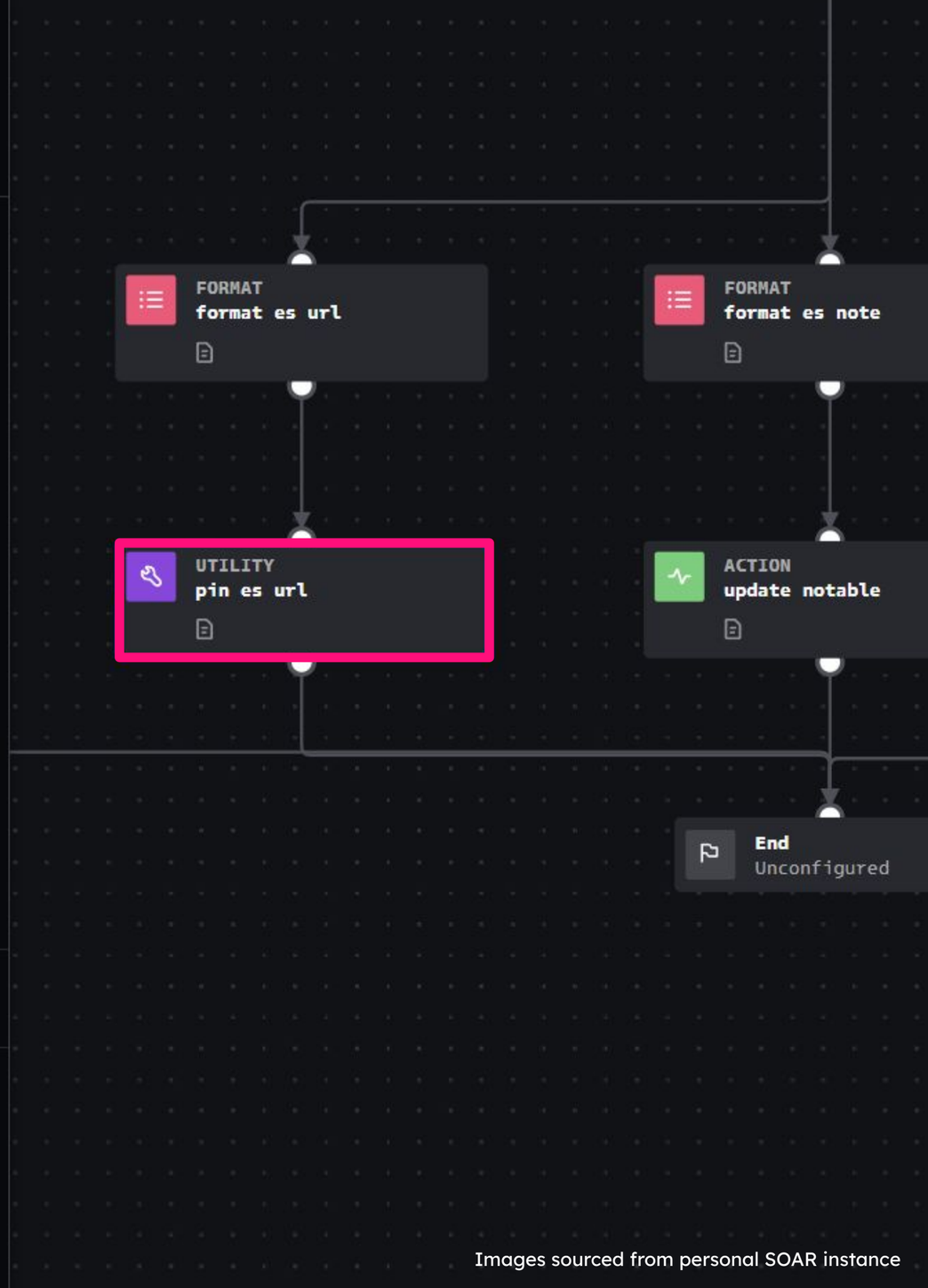


Images sourced from personal SOAR instance

ES Notable Onboarding

Utility Block:
Native SOAR utility to pin a message to the event with the ES pivot URL

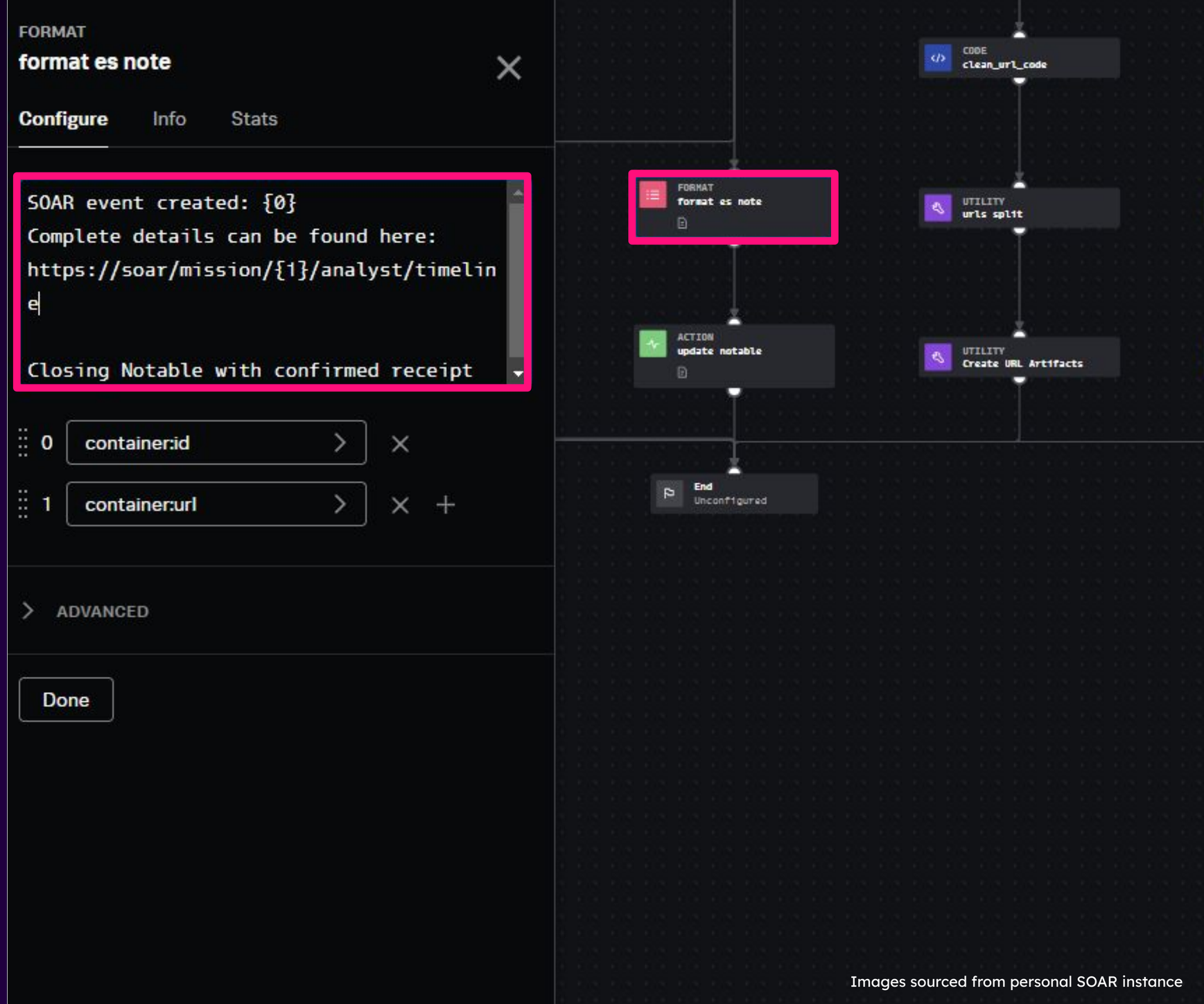




Images sourced from personal SOAR instance

ES Notable Onboarding

Format Block:
Creates a comment with another link to your SOAR instance with the matching container ID



Images sourced from personal SOAR instance

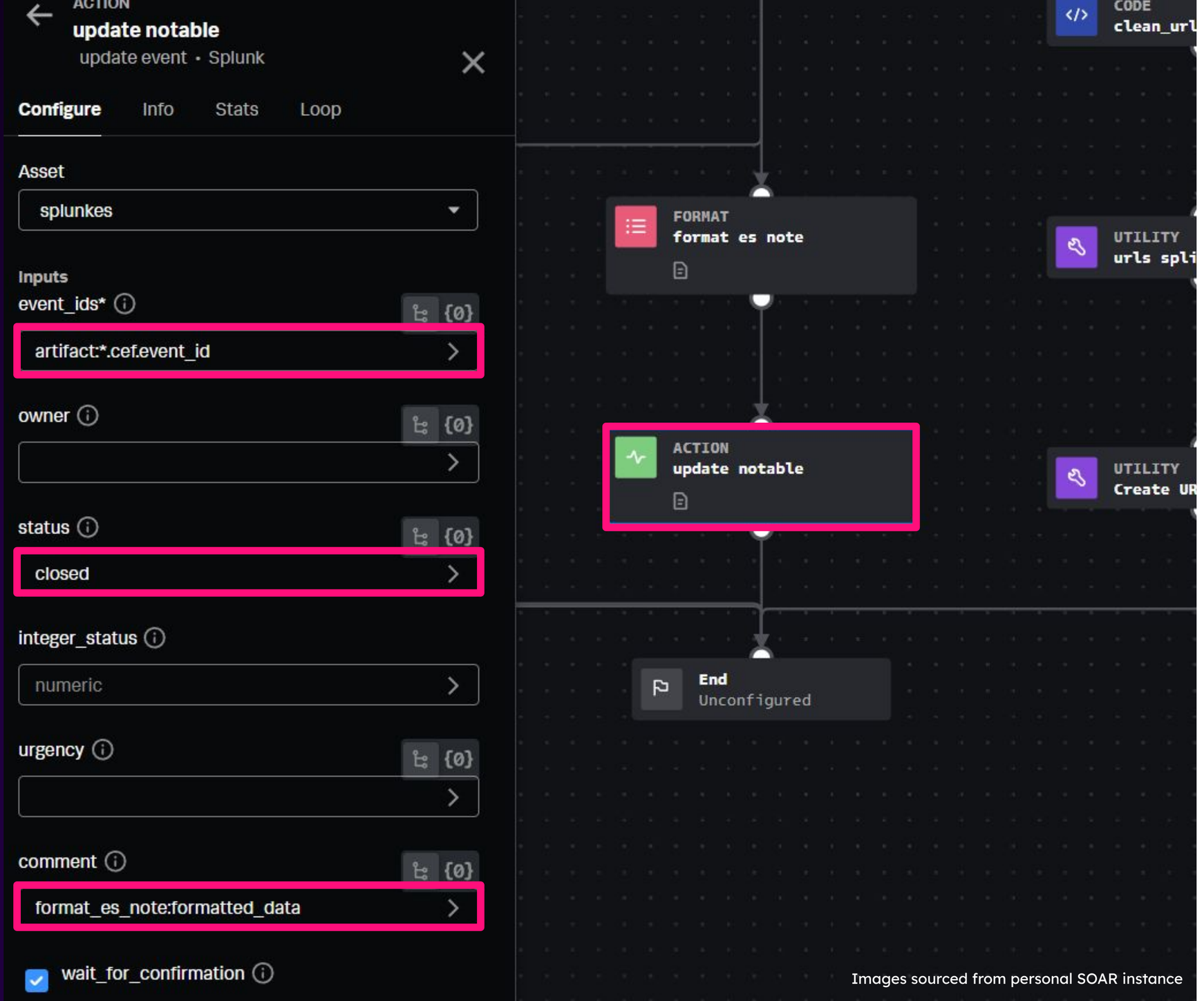
ES Notable Onboarding

Action Block:

Pulls the event_id from the notable event data

Sets to Closed

Uses format block as comment



Images sourced from personal SOAR instance

Automated User Investigative Playbook

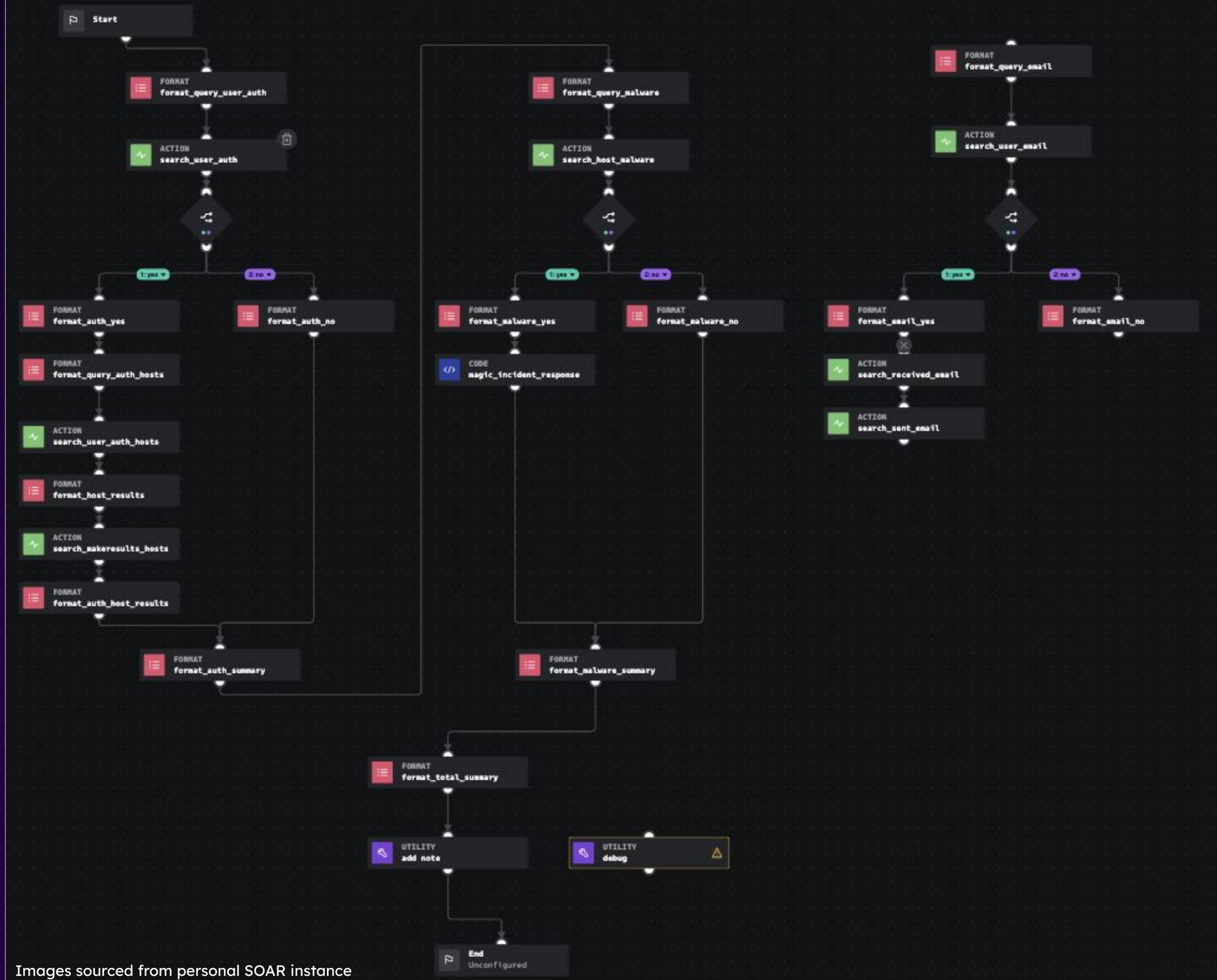
Type 2 Splunk Integrated

Common for customers to have automation goals blocked by the larger enterprise.
However integration with Splunk Enterprise is almost a given, so lets automate our investigations instead…

- Define / review questions normally asked during a user focused security investigation, and convert them into SPL for a Splunk search query
- Example: Has the user logged in within the last seven days?
  SPL: `| tstats count from datamodel Authentication where Authentication.user="$user$"`
  `| eval result = if(count > 0, "yes", "no") | fields result`
- Playbooks runs through a comprehensive set of questions with various forks and branches, retuning a summary of the users activity allowing the analyst to immediately start assessing user behaviour instead of running a ton a searches manually (huge saving in MTTR)
- Opens the door to auto-closing / escalating cases based on the responses returned

User Investigation Playbook

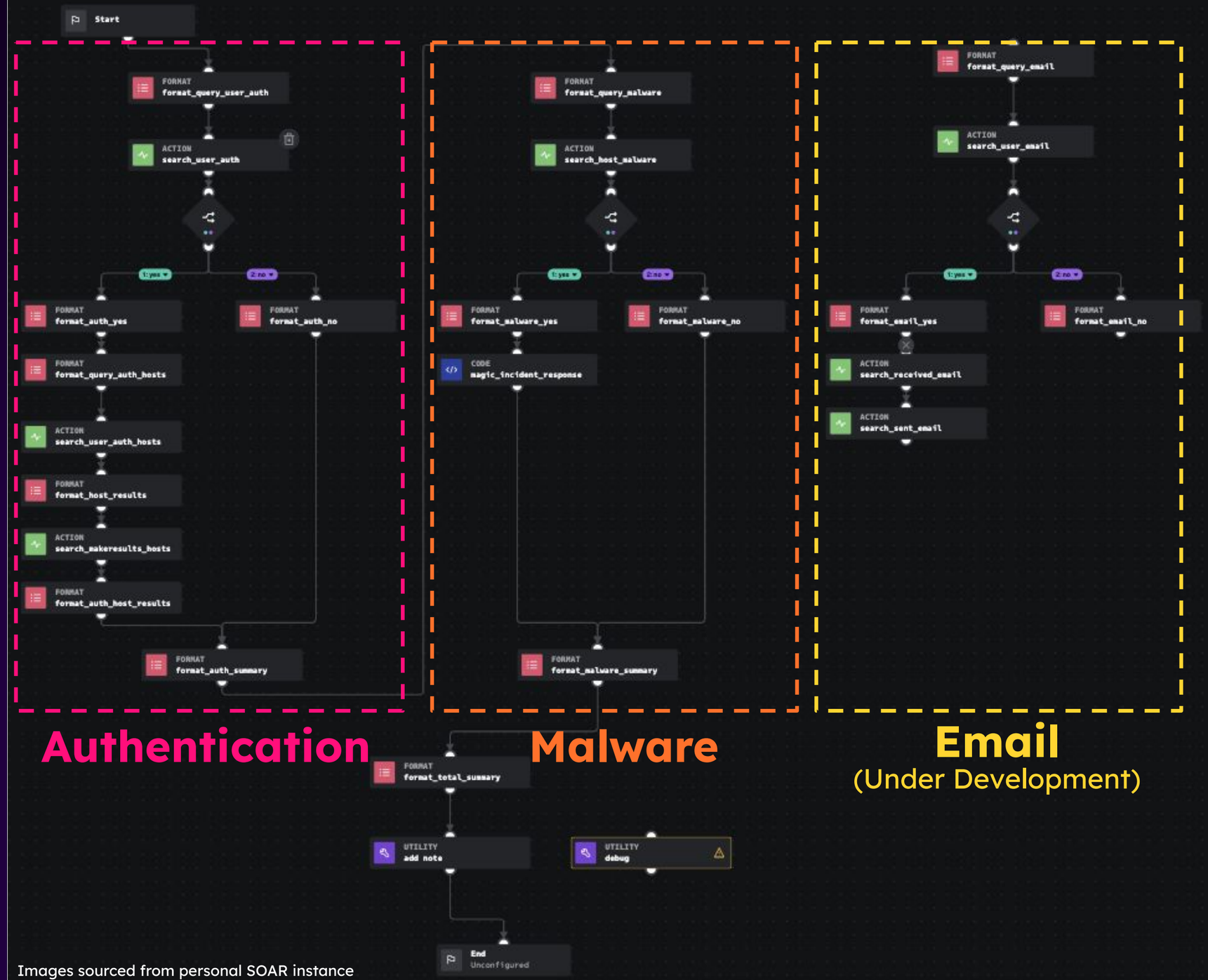
A short example to get you started...



Images sourced from personal SOAR instance

User Investigation Playbook

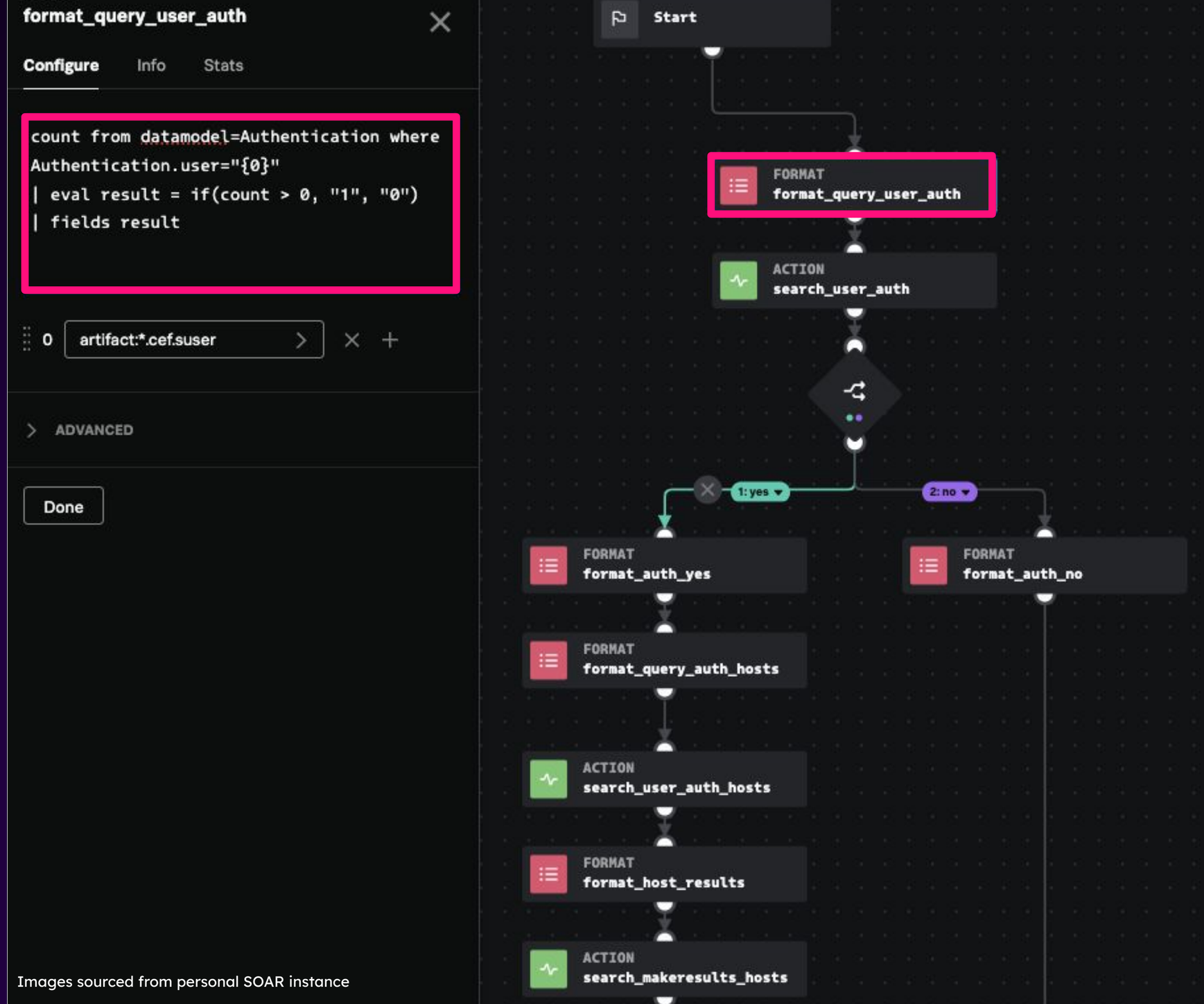
Structured with swimlanes for various lines of questioning



Images sourced from personal SOAR instance

User Investigation Playbook

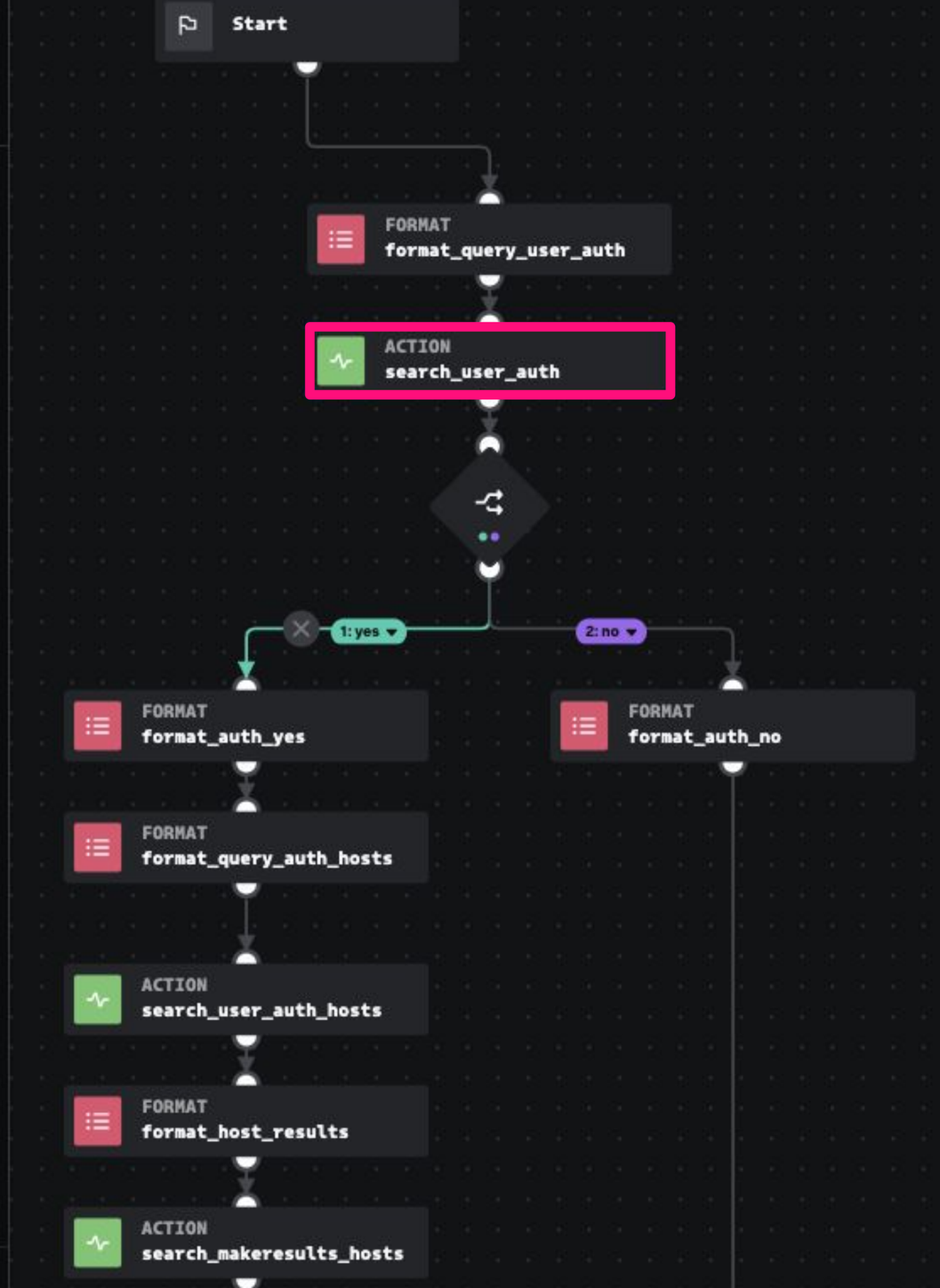
Format Block:
Create the user authentication query string



Images sourced from personal SOAR instance

User Investigation Playbook

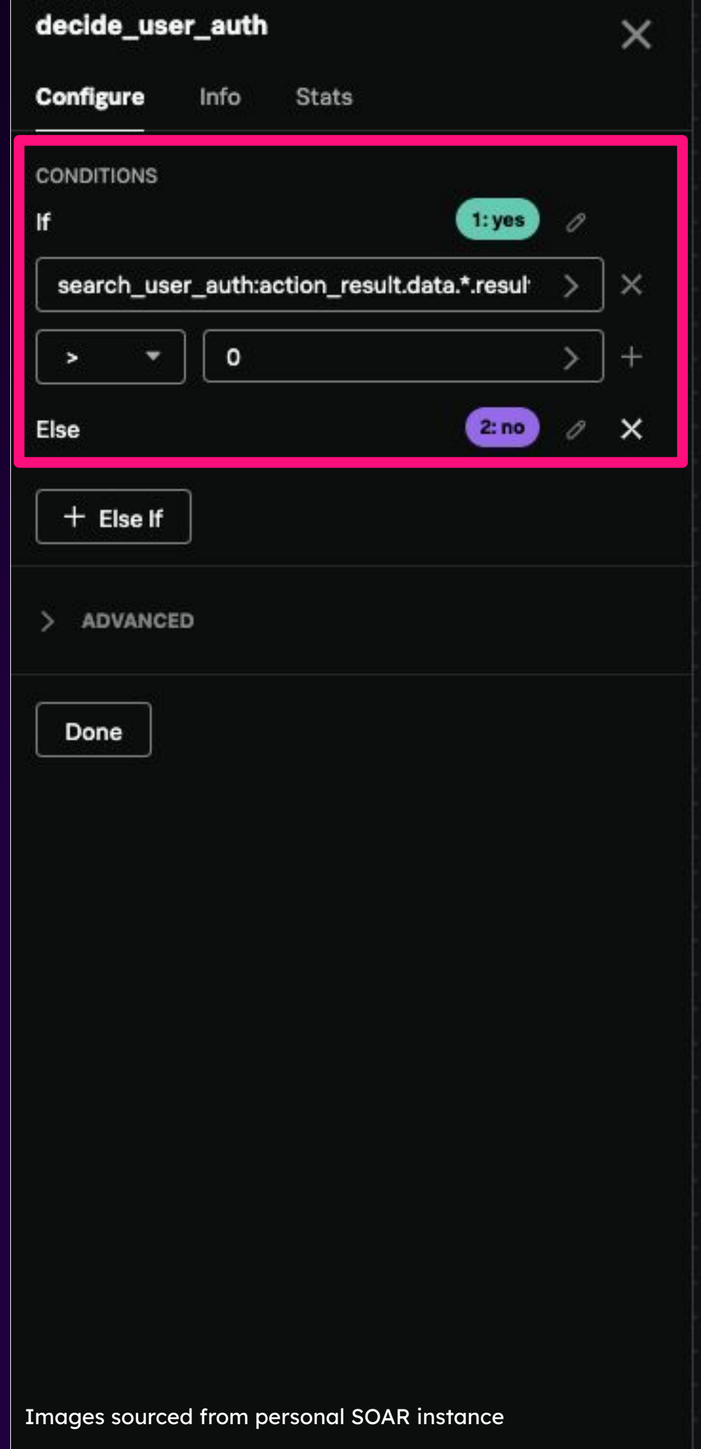
Action Block:
Execute the user authentication search

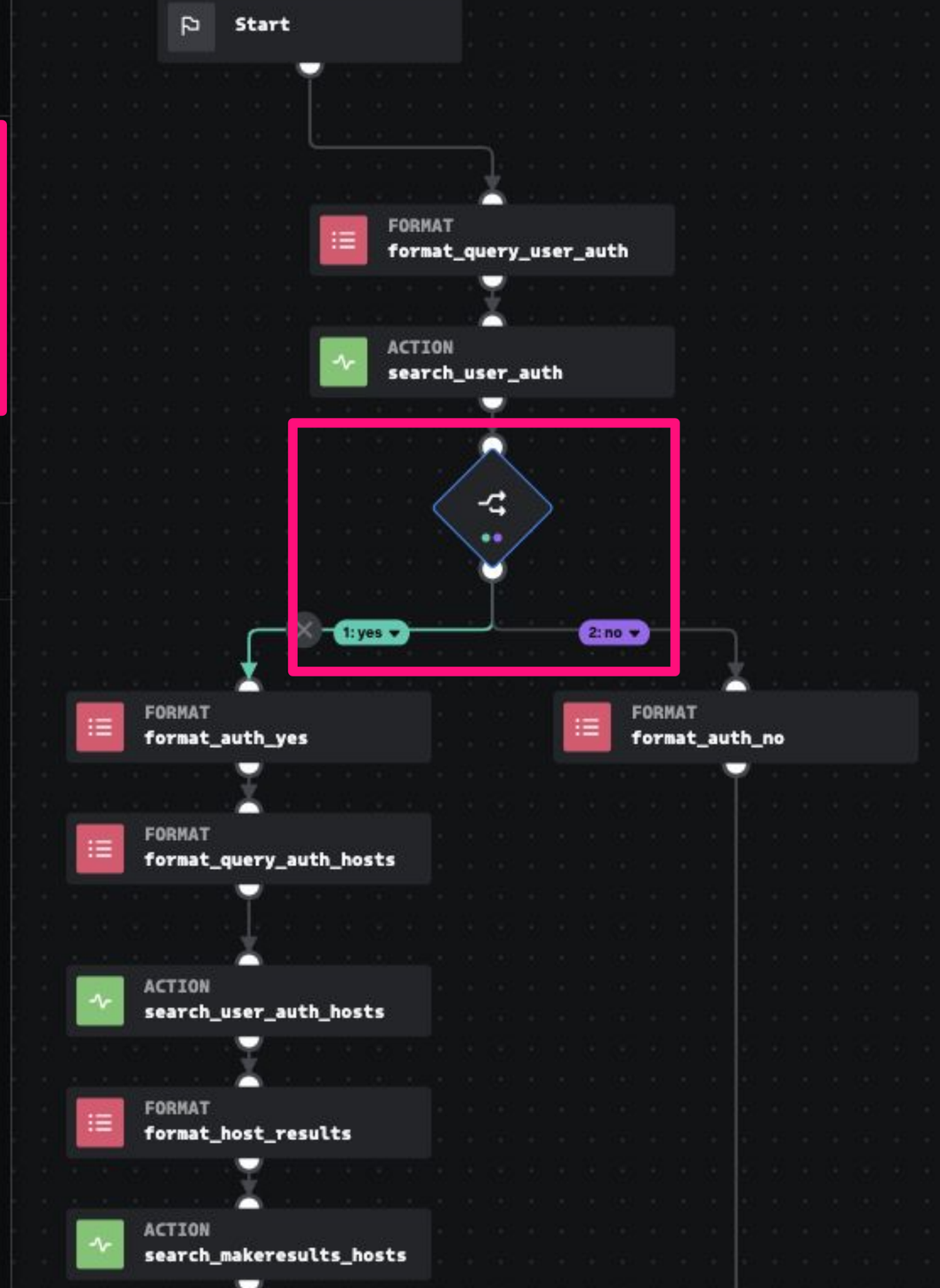


Images sourced from personal SOAR instance

User Investigation Playbook

Decision Block:
Use the search results to determine if the user has been active

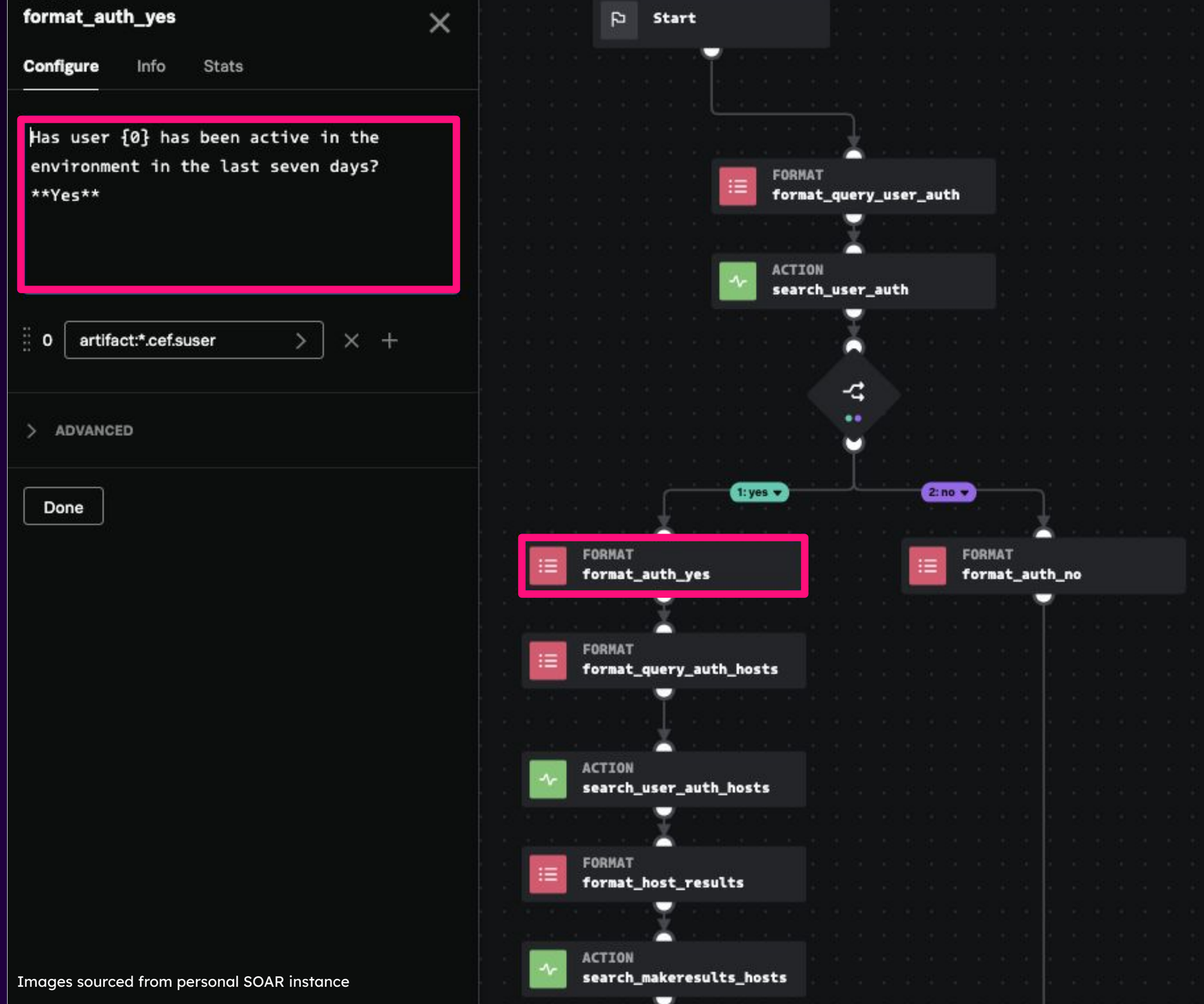




Images sourced from personal SOAR instance

User Investigation Playbook

Format Block:
Create a note with context for the analyst



Images sourced from personal SOAR instance

User Investigation Playbook

Playing it out in practice

Widgets Notes

Active User: admin

NOTES (1) Search notes

Auth Summary

General Note by **soar_local_admin** Apr 3, 2024 12:00 pm

Incident Summary

Has user admin has been active in the environment in the last seven days? **Yes**

How many hosts has the user logged onto in the time period? **2**

What were the unique hosts involved? **gacrux.i-0920036c8ca91e501,mars.i-08e52f8b5a034012d**

Has Malware been detected on any of the hosts of interest (gacrux.i-0920036c8ca91e501,mars.i-08e52f8b5a034012d)? **No**

<End of Incident Summary>

Widgets Notes

Inactive User: pwny

NOTES (1) Search notes

Auth Summary

General Note by **soar_local_admin** Apr 3, 2024 12:11 pm

Incident Summary

Has user pwny has been active in the environment in the last seven days? **No**

<Skipping authentication based questions>

Has Malware been detected on any of the hosts of interest ()? **No**

<End of Incident Summary>

Images sourced from personal SOAR instance

Container Enrichment Playbook

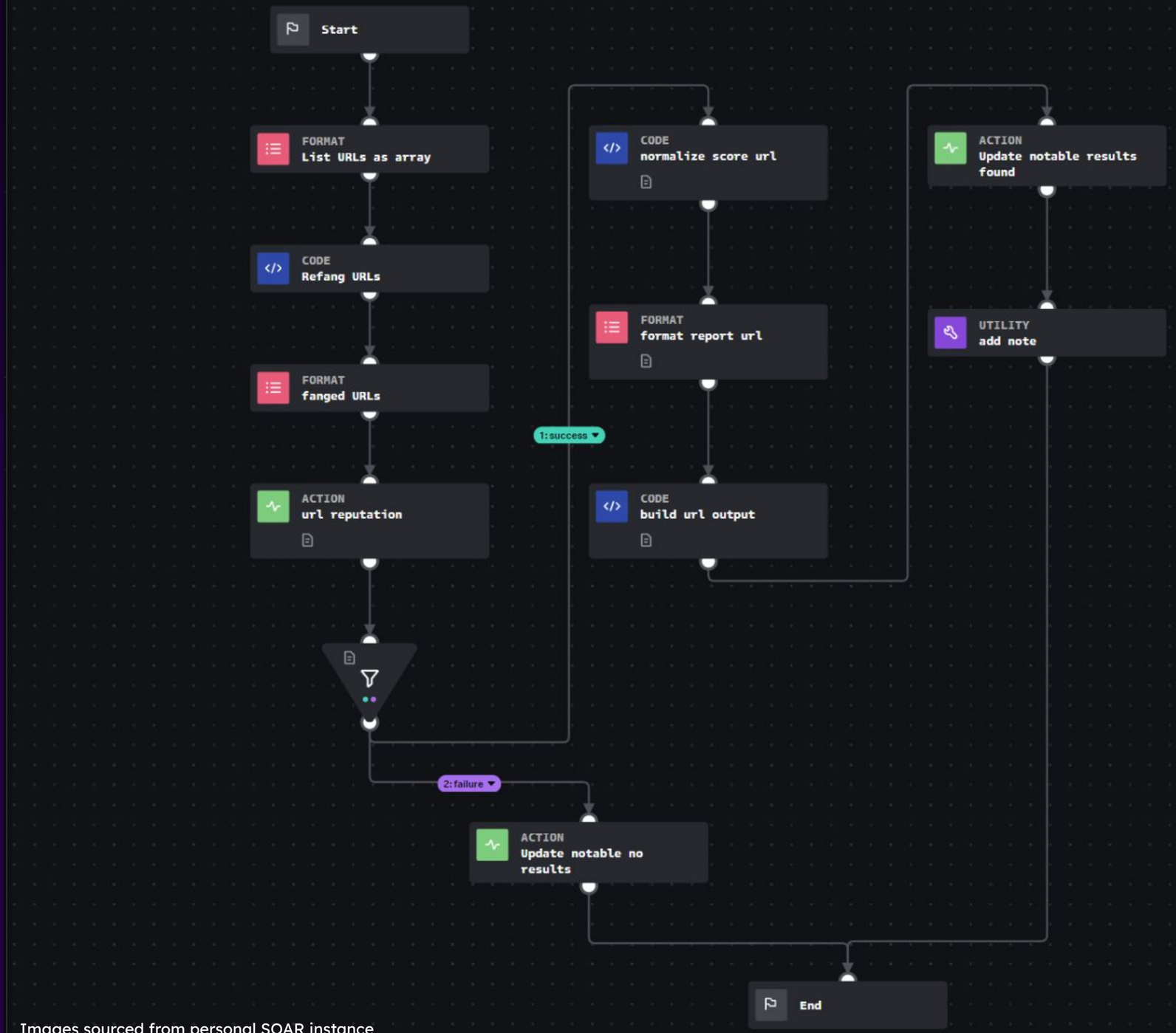
Type 4 External Services

Common for enrichment of IOCs such as URLs against internet based reputation services

- This example uses VirusTotal v3, can be swapped out to any service
- Gathers URLs from the container and runs a custom script to ‘re-fang’ an URL
- URLs are run against an action block

Enrichment Playbook

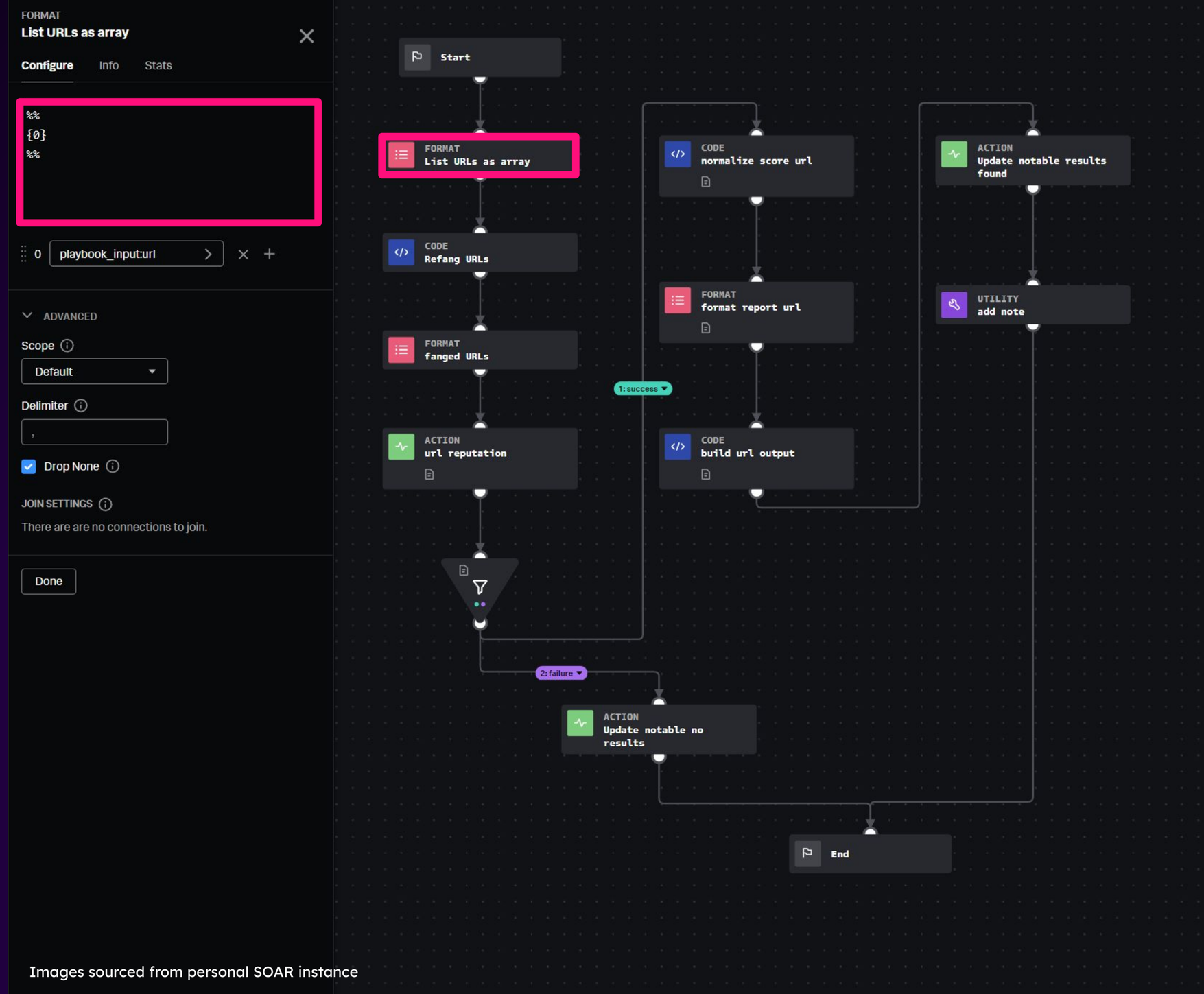
An example of leveraging threat reputation tools such as VirusTotal



Images sourced from personal SOAR instance

Enrichment Playbook

Format Block:
Leverage format block markup %% to list URLs as an array



Images sourced from personal SOAR instance

Enrichment Playbook

Code Block:
Using custom code to replace chars in a defanged URL

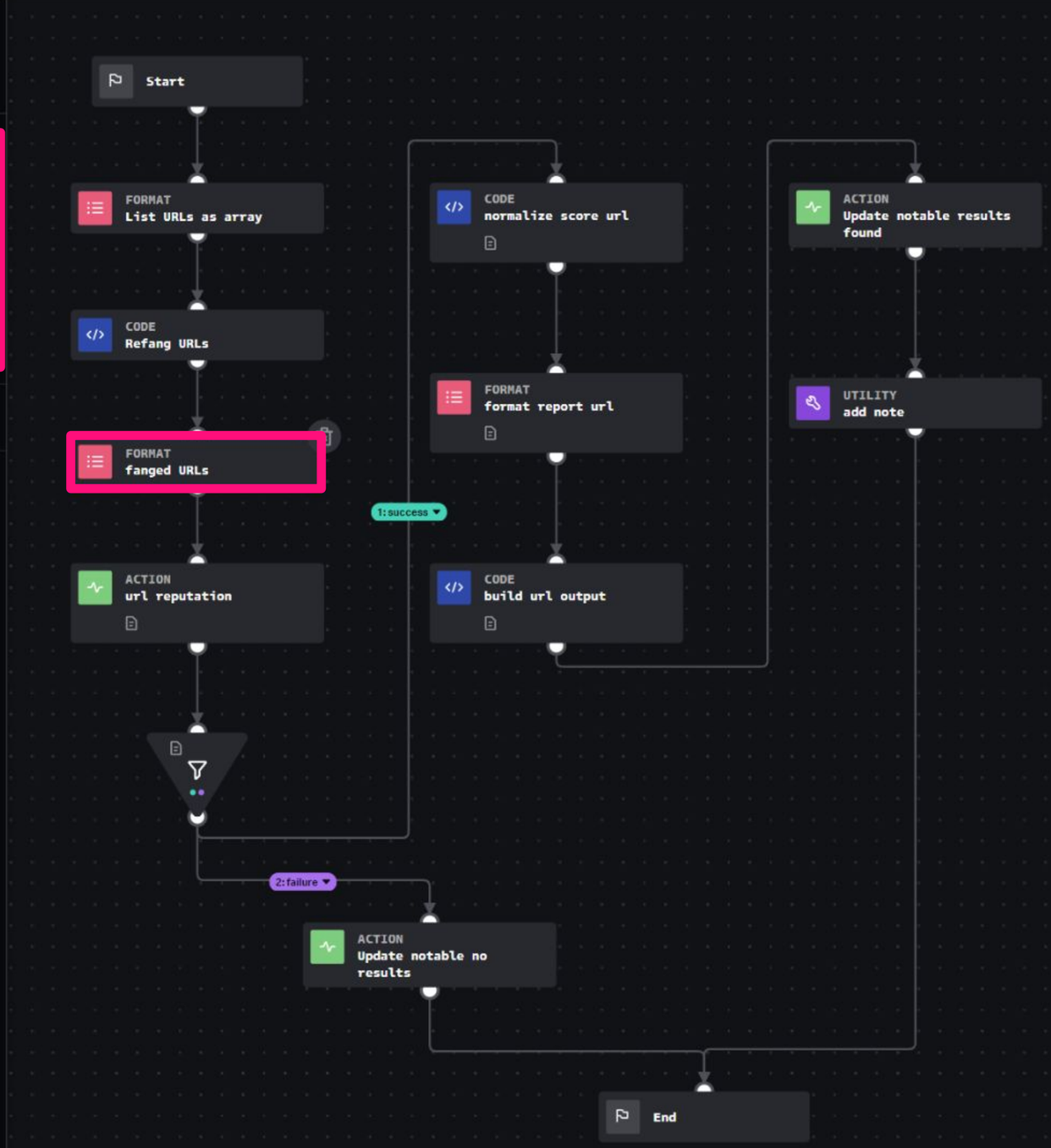
refang_urls Function Copy

```
@phantom.playbook_block()
def refang_urls(action=None, success=None, container=None, results=None, handle=None, filtered_artifacts=None, filtered_results=None, custom_func
    phantom.debug("refang_urls() called")

    list_urls_as_array__as_list = phantom.get_format_data(name="list_urls_as_array__as_list")

    refang_urls__refanged_url = None

    ################################################################################
    ## Custom Code Start
    ################################################################################

    def refang(defanged_urls):
        refanged_urls = []

    # interate over the list of urls
        for url in defanged_urls:
            if url == None or len(url) == 0:
                # skip empty urls
                continue

            phantom.debug("Before refang: {}".format(url))

            url = url.replace("hxxp", "http")
            url = url.replace("[.]", ".")
            url = url.replace("[at]", "@")
            url = url.replace("\\", "")

            phantom.debug("After refang: {}".format(url))

            refanged_urls.append(url)

        return refanged_urls

    phantom.debug(list_urls_as_array__as_list)
    refang_url__refanged_url = refang(list_urls_as_array__as_list)

    ################################################################################
    ## Custom Code End
    ################################################################################

    phantom.save_run_data(key="refang_urls:refanged_url", value=json.dumps(refang_urls__refanged_url))

    fanged_urls(container=container)

    return
```

Images sourced from personal SOAR instance

Enrichment Playbook

Format Block:
The now ‘fanged’ URLs are gathered into an array again for sending to the action block below



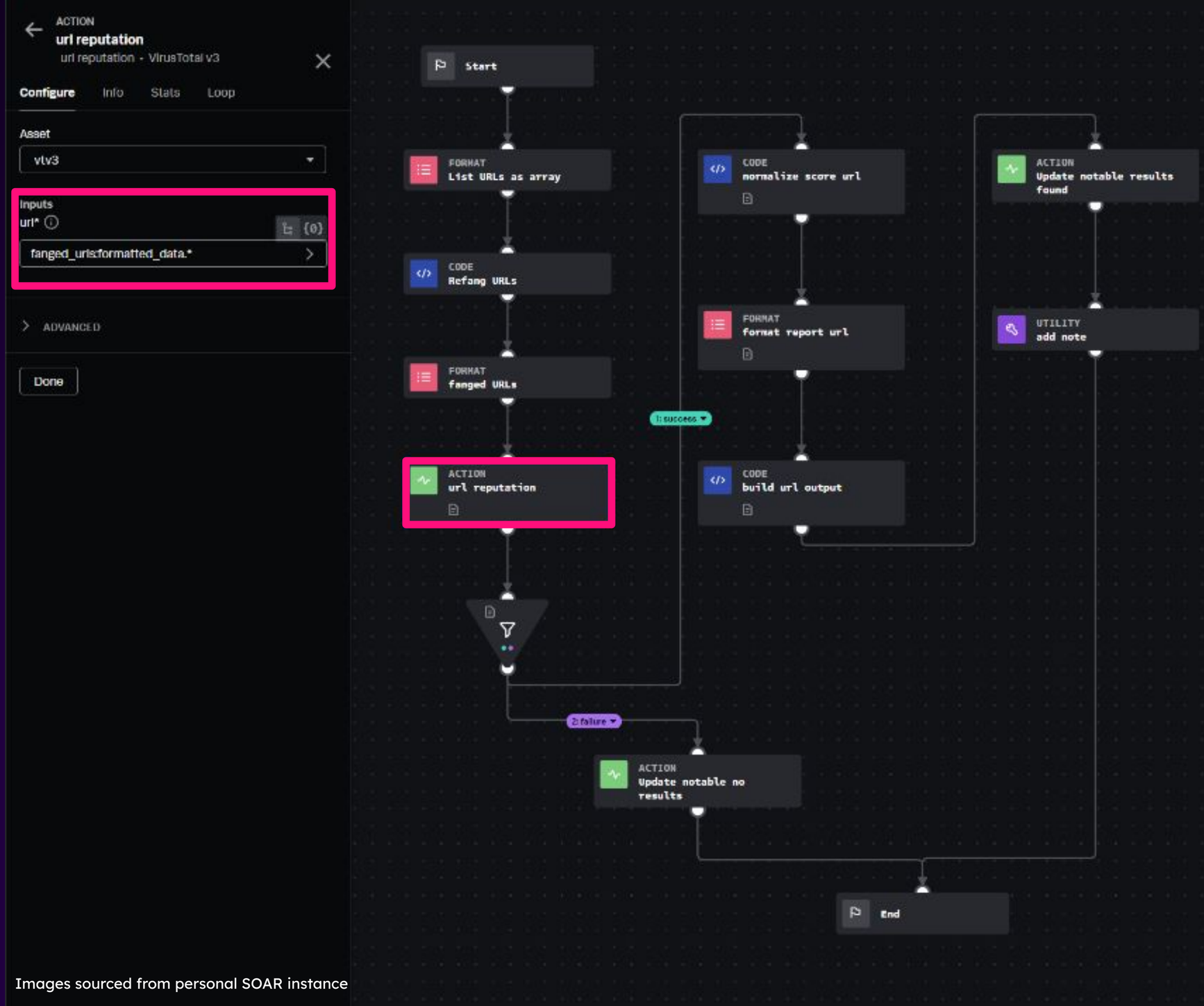
Images sourced from personal SOAR instance

Enrichment Playbook

Action Block:
Note the input uses formatted_data.*

Why?

This makes the action block execute for each item in the array



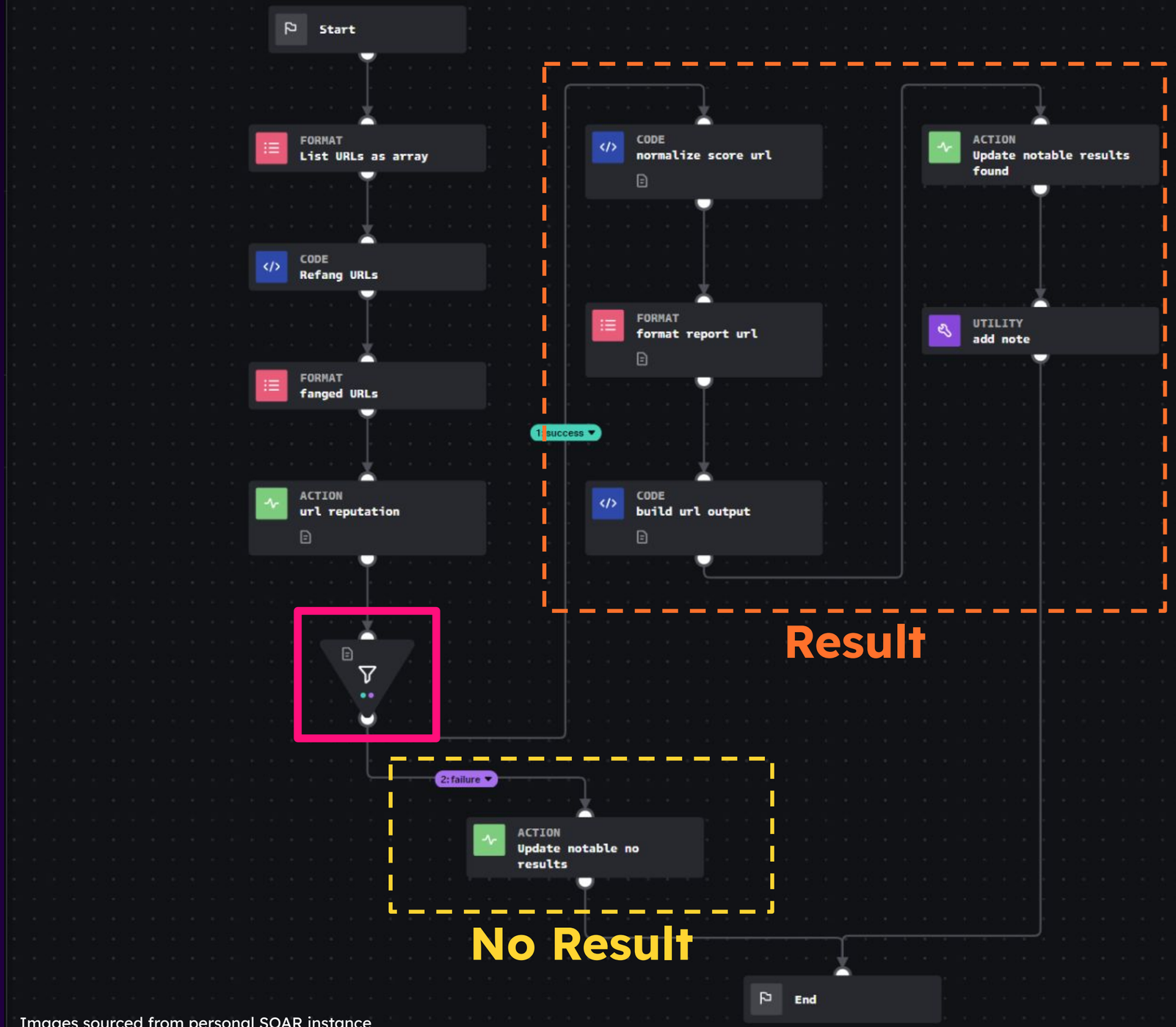
Images sourced from personal SOAR instance

Enrichment Playbook

Simple filtering now occurs to determine if a result was found

If successful (result): continue on path and build out as desired

If unsuccessful (no result): Update ES and finish



Images sourced from personal SOAR instance

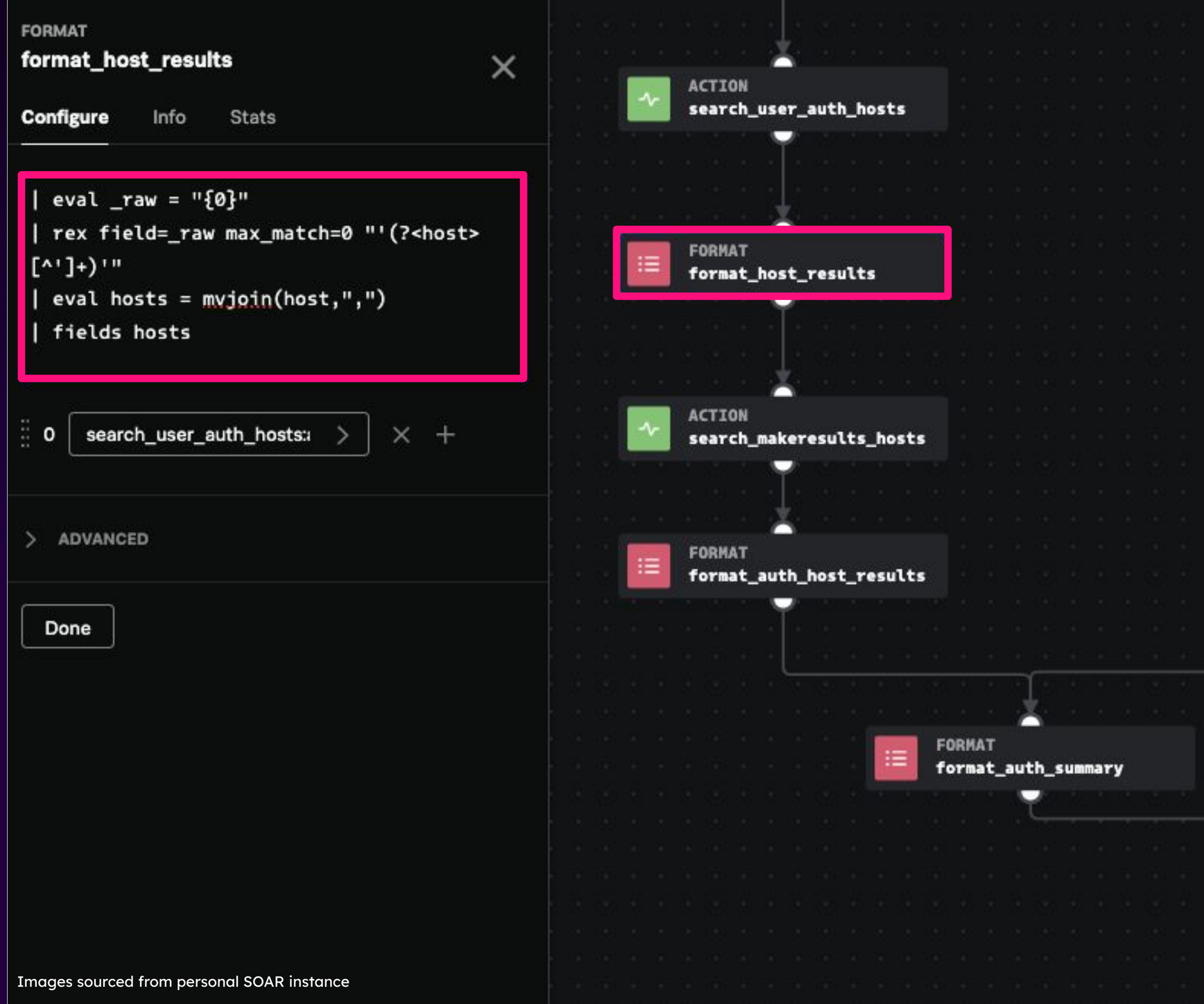
Bonus Tip

SPL EVAL Functions within* SOAR!

- “*This would be so much easier if I could could just use SPL instead*”
  - Anyone who’s ever had to write a custom function in SOAR for basic string manipulation…
- Unfortunately SOAR can’t natively leverage SPL, and has limited built-in functionality for handling manipulation of data within the platform
  - 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?

SOAR SPL

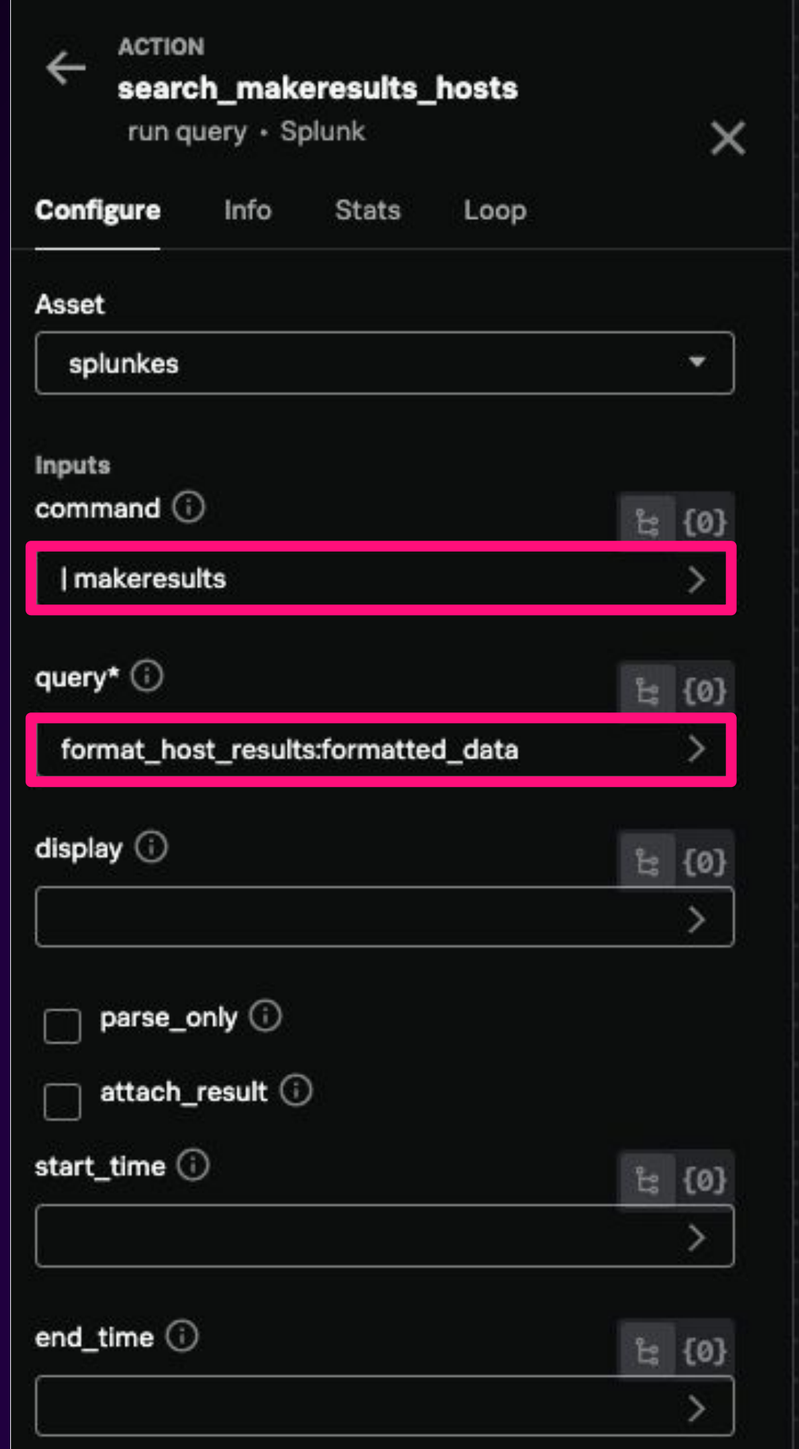
Format Block:
Write your SPL to reformat your data, with input inserted into _raw as a variable

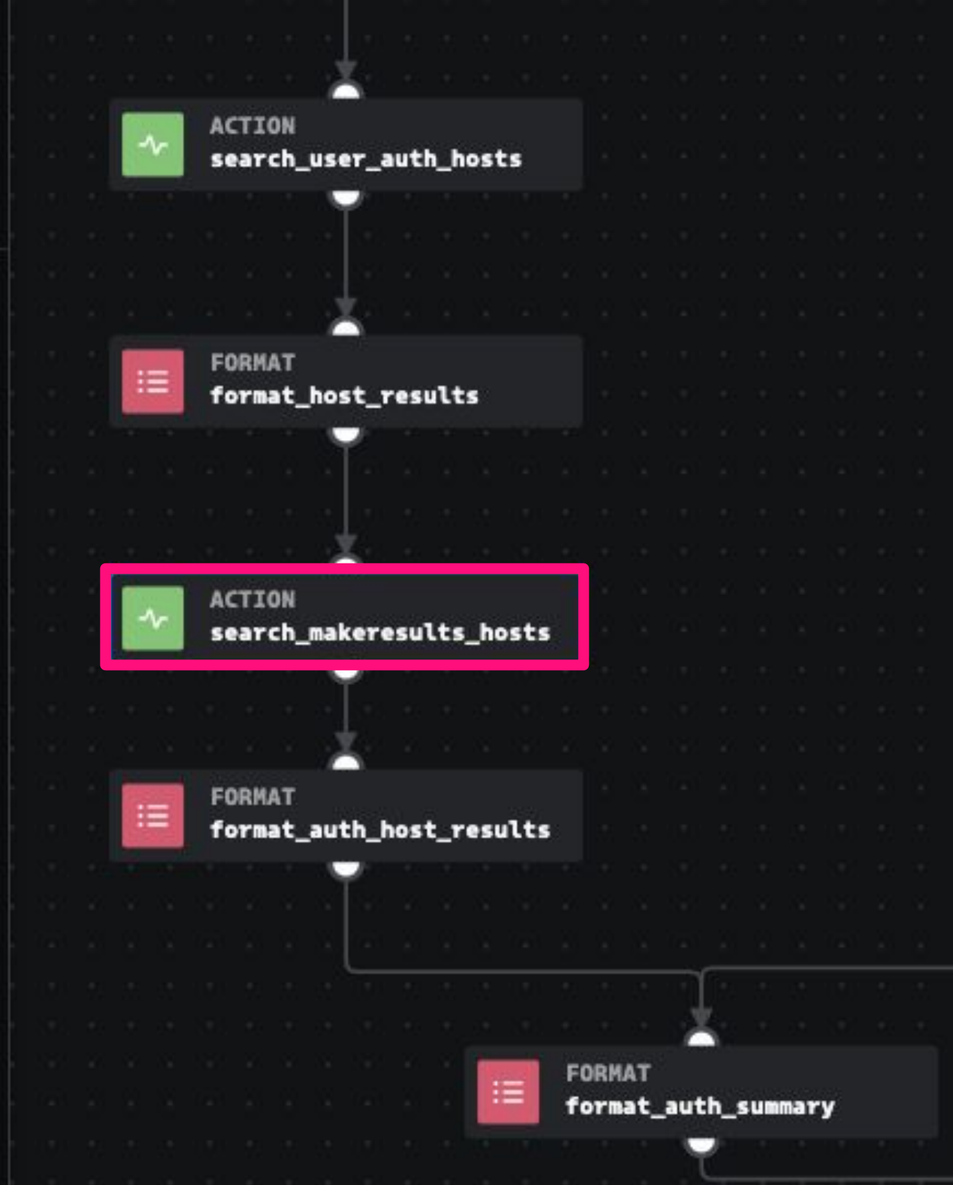


Images sourced from personal SOAR instance

SOAR SPL

Action Block:
Pass this SPL into an action block with a leading | makeresults and return the result



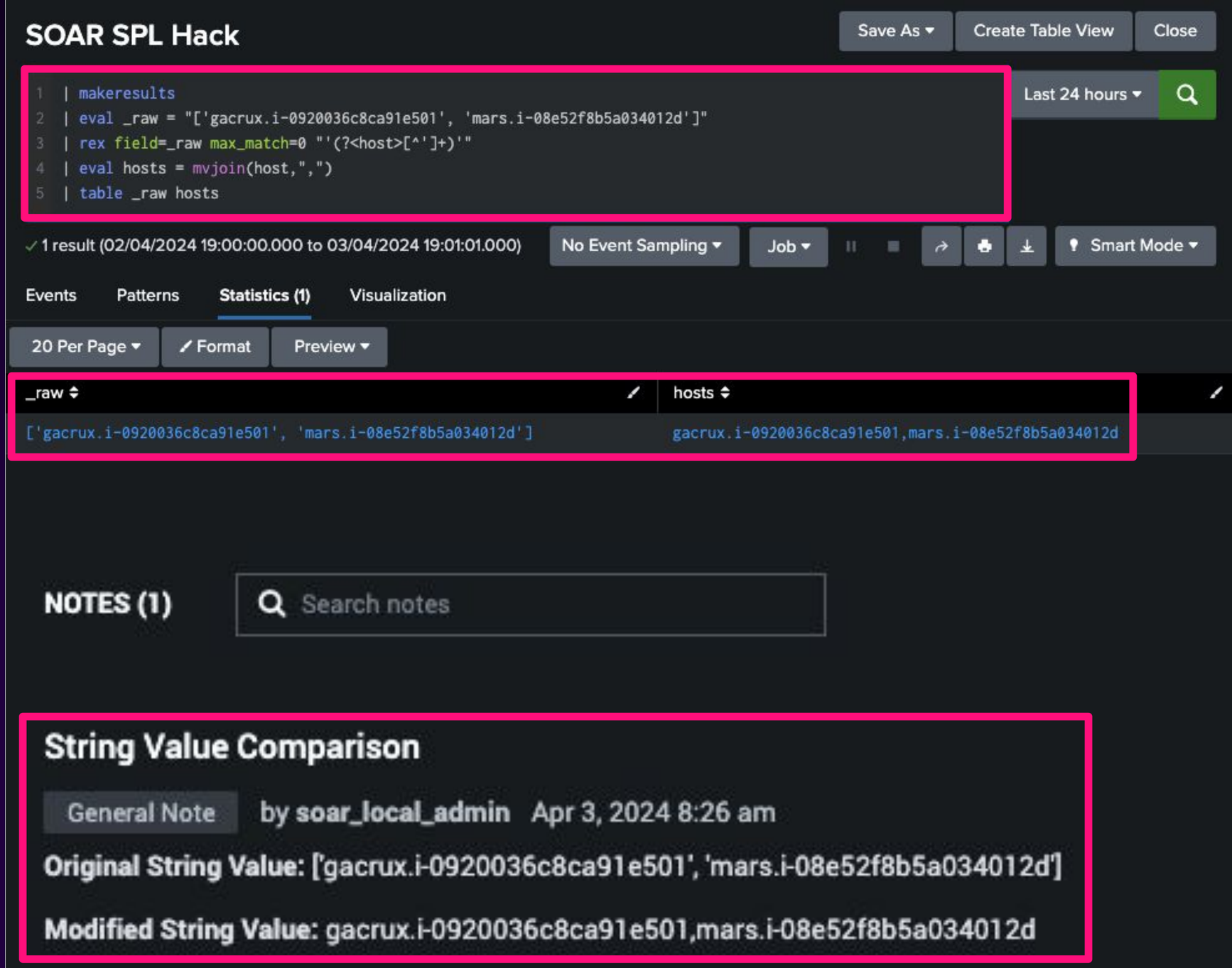


Images sourced from personal SOAR instance

SOAR SPL

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

String updated with zero custom code!

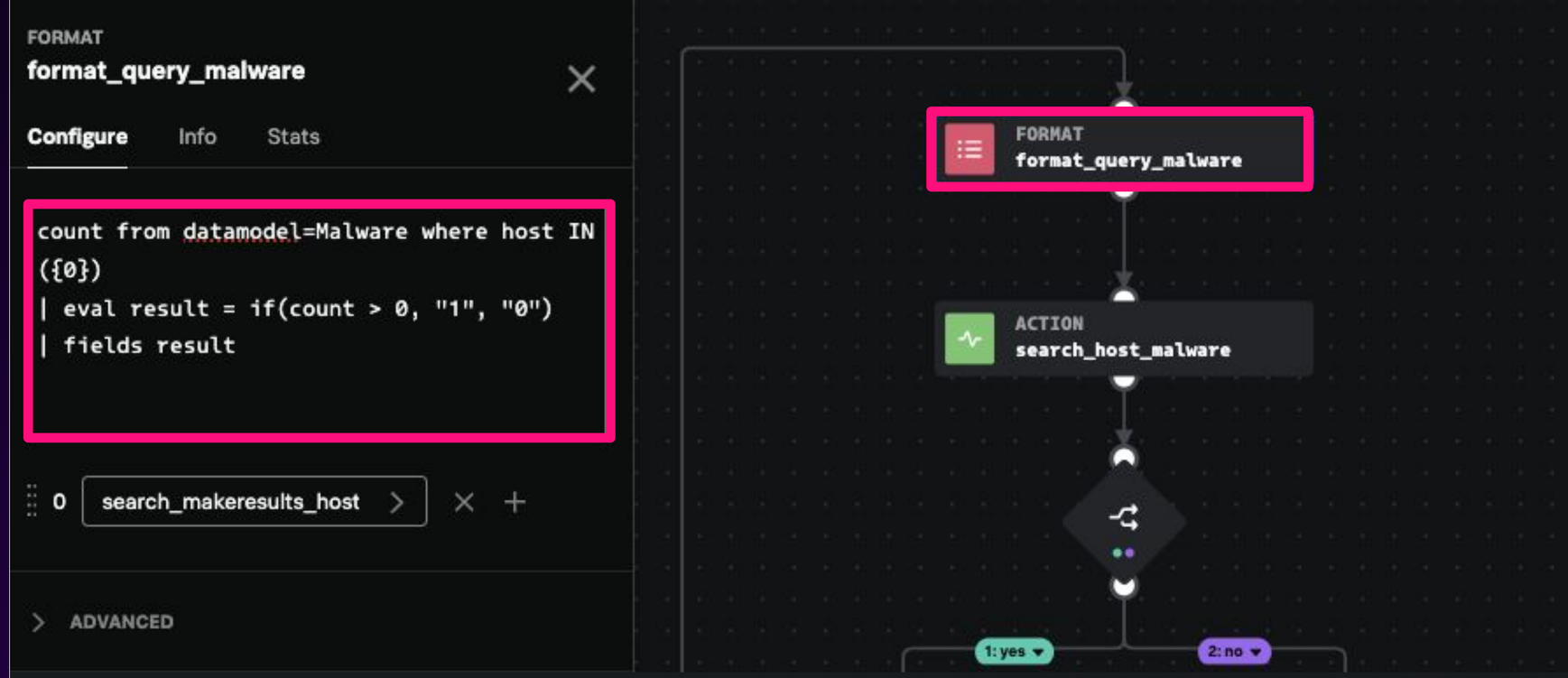


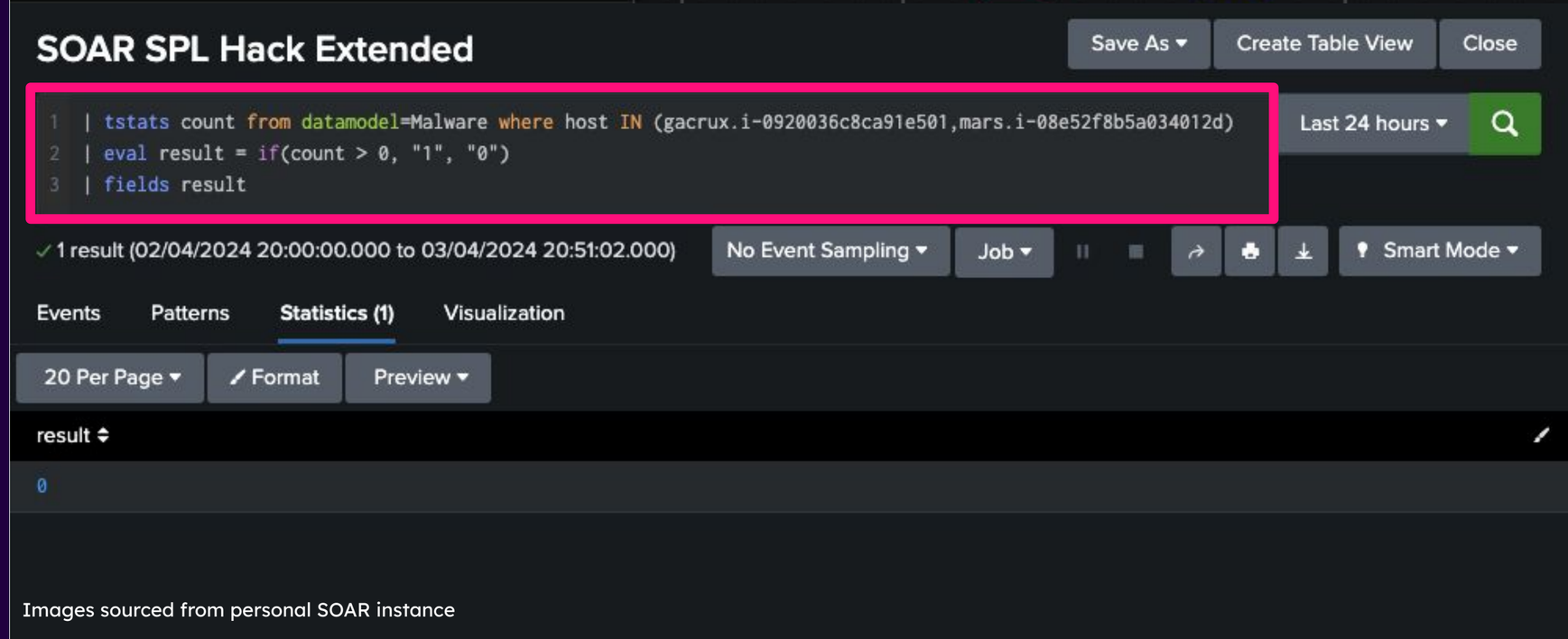
Images sourced from personal SOAR instance

SOAR SPL

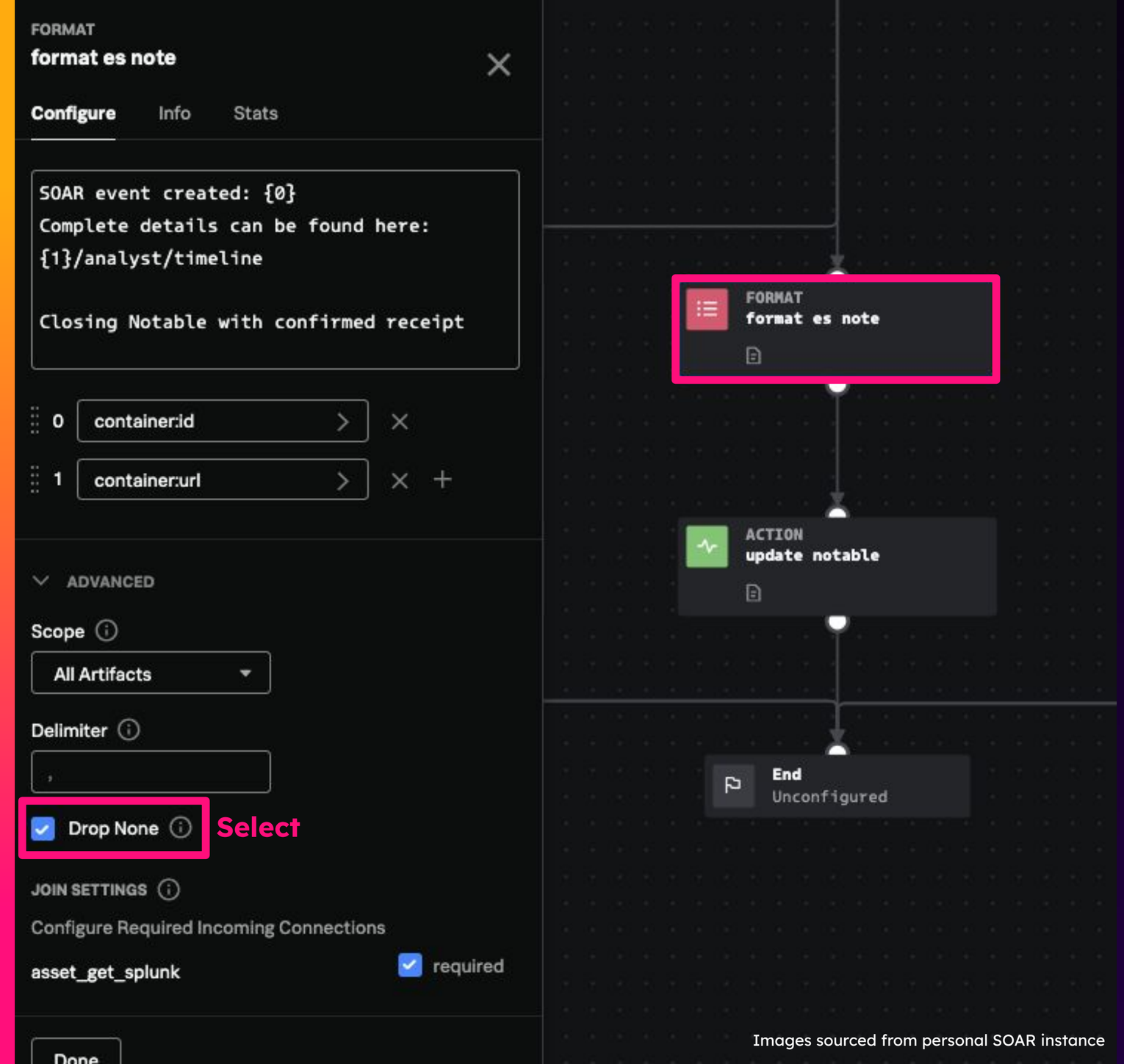
Not just for pretty presentation!

Results from a search are cleaned up and used to power the next search in the investigation





Images sourced from personal SOAR instance



Images sourced from personal SOAR instance

Bonus Tip

Drop None

Depending on how you've configured the data path ie *artifact:*.cef.url* can return

`"url, None, None, None, None, None"`

Select Drop None to clean this up

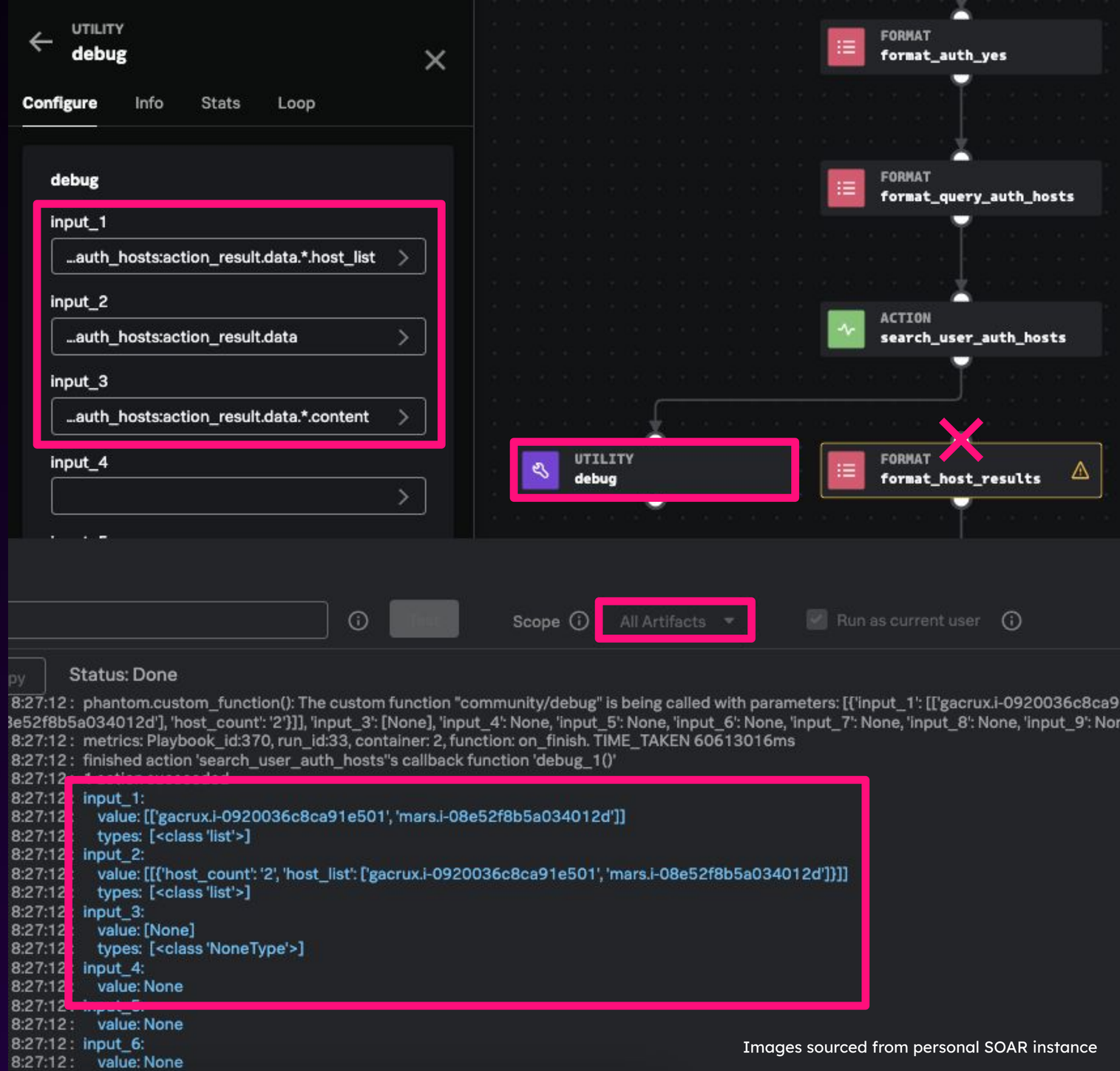
This is NOT selected by default

Bonus Tip

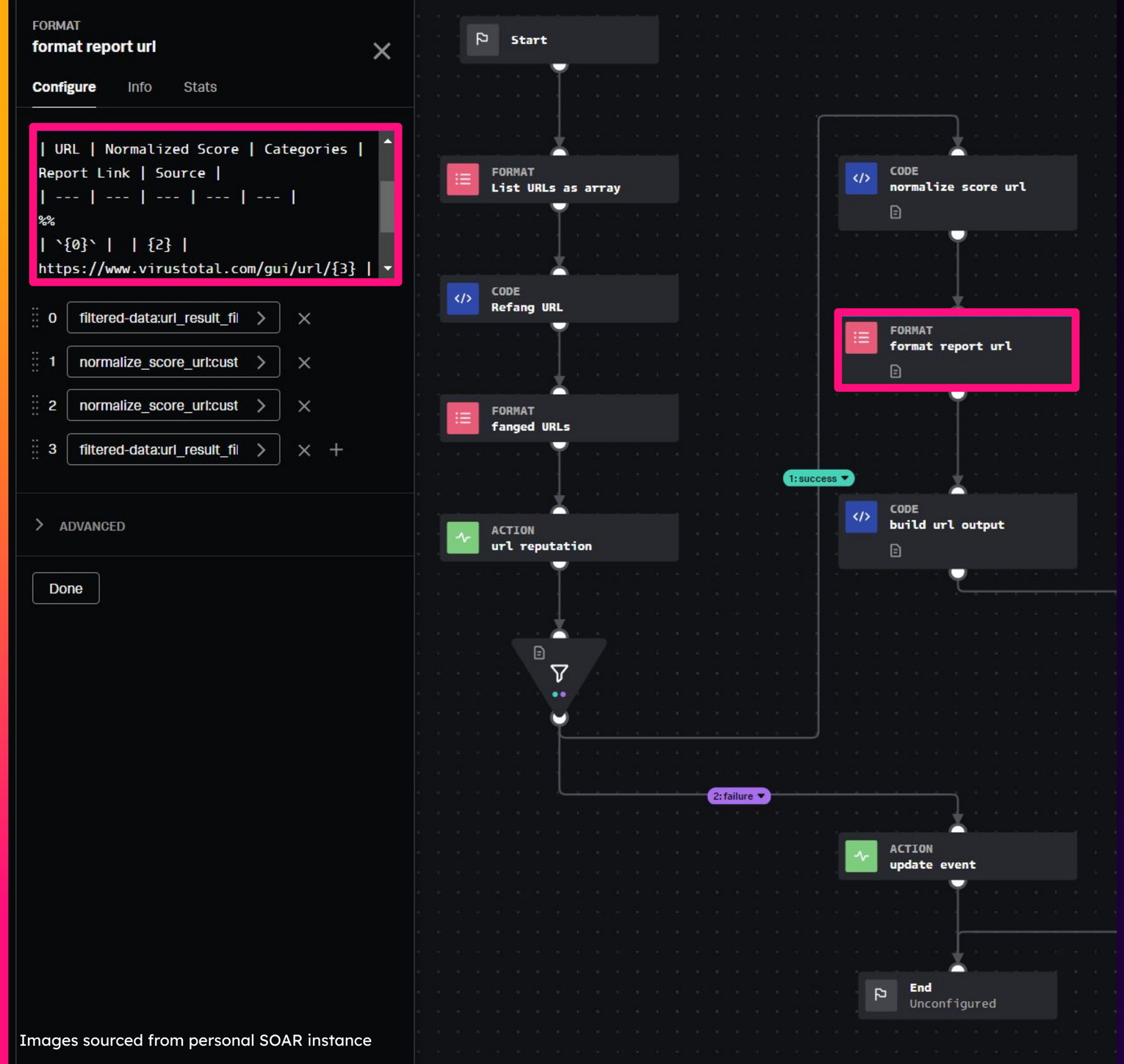
Debugging

Use the Playbook Debugger window to monitor a playbook as it executes

- Provide an artifact ID and always set the scope to ‘All Artifacts’
- Insert Utility Debug blocks to review action block output
- Disconnect downstream blocks to speed up execution
- Use phantom.debug() in custom code blocks for even more control



Images sourced from personal SOAR instance



Images sourced from personal SOAR instance

Bonus Tip

Formatting Arrays

Format block has inbuilt markup for creating headings, tables and more

See docs for more information

https://docs.splunk.com/Documentation/SOAR/current/Playbook/VPEFormatBlock#Example_of_defining_a_template

Bonus Tip

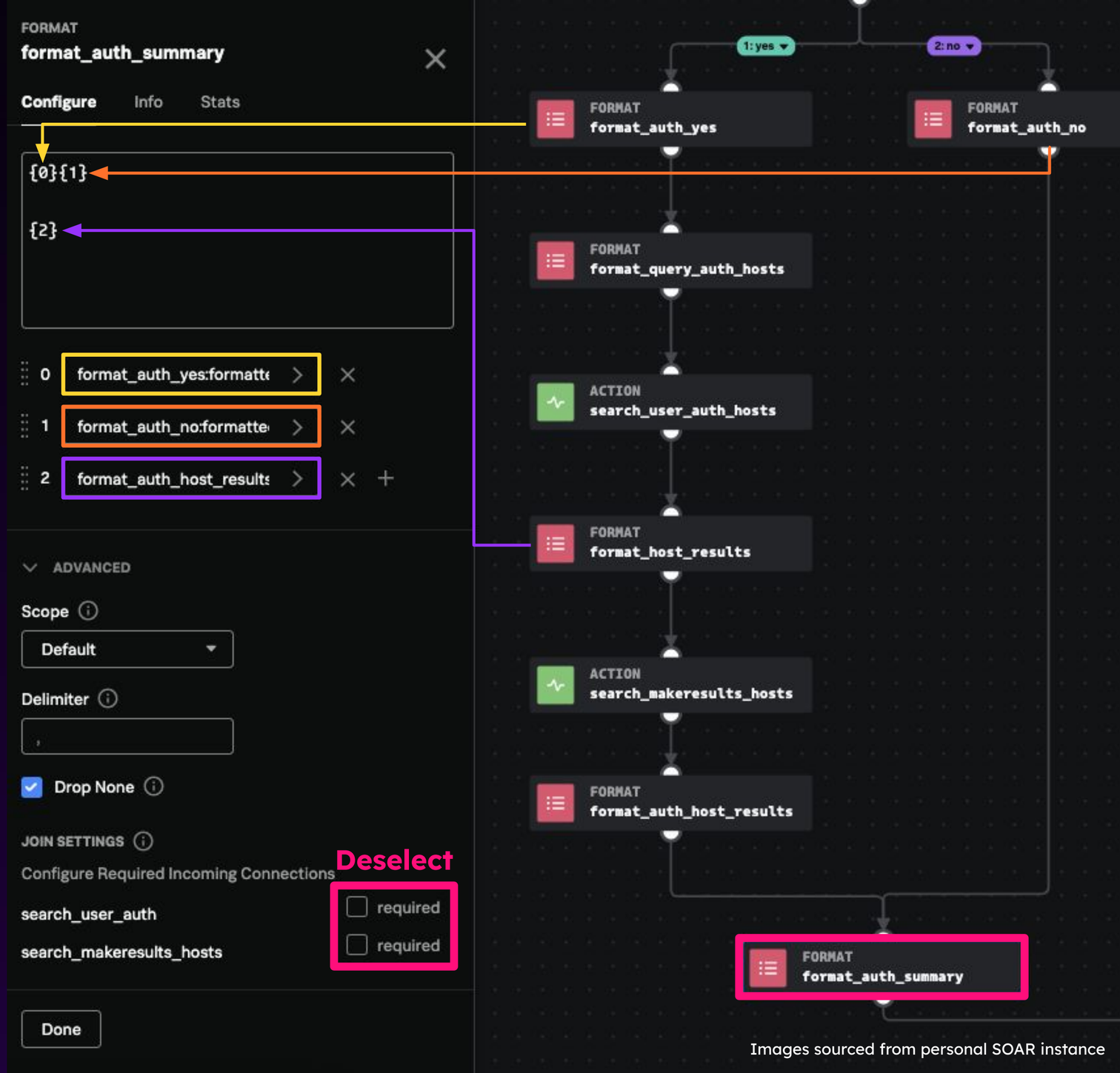
Joining Branches

An absolute must-know is how SOAR handles the joining of two branches

- By default SOAR expects both branches to execute and will wait indefinitely - deselect ‘required’ to fix

You can also use a join to perform advanced formatting of your content

- Configuring two inputs from different branches acts like an OR statement allowing for dynamic results



Images sourced from personal SOAR instance

Final Development Tips

Driving Effective Automation

- Leverage documented SOC processes, work instructions and SOPs
- Consult T1, T2 and T3 analysts on their day-to-day repetitive tasks
- Leverage the Community Playbooks

Don’t discard small automations, small tasks that appear minor or not worth the effort in the micro can add up to huge tangible savings in the macro.

Aim for small wins first.

Where to go next?

All of these example playbooks and more are available in a public Git repository

https://github.com/MattHyp3/Hyperion3_SOAR

Hyperion3 SOAR
Git Repo

Questions!

Open floor to any
and all questions

Thank you