Automated Triage Collection at Scale in AWS

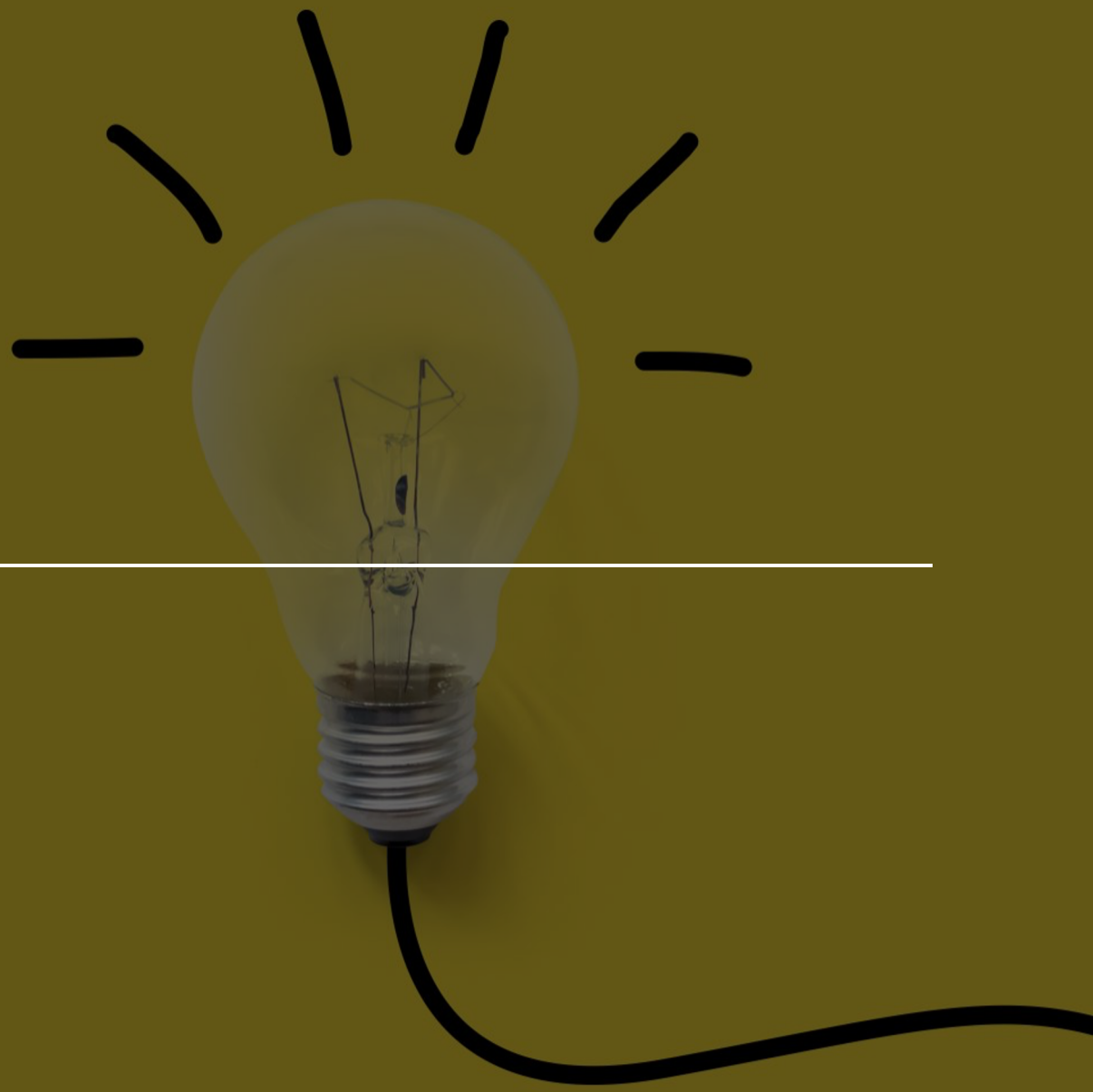
BSidesDFW 2021

11/06/2021

Ryan Tick - Manager, KPMG

Agenda

- Objectives
- Problem Statement
- Benefits of a Triage Collection Workflow
- Solution Deep Dive
- Pro Tips
- Further Learning



> whoami

- Manager at KPMG specializing in traditional IR and AWS DFIR automation
- ISC^2 (CISSP), SANS (GCFE, GCFA, GNFA, GREM) and AWS (Solution Architect Professional, Security Specialty, all Associate level) certified
- Previously a Senior Cloud Security Engineer for Goldman Sachs
- Presented at a lot of conferences (including AWS re:Invent)
- University of Notre Dame Grad

Objectives

Objectives

1	Create an automated workflow to allow for the triage collection of EC2 instances (Windows, Linux, and macOS) at scale in a same-account, cross-account, and cross-region scenario
2	Allow for the secure download of triage scripts from a private S3 bucket, and for the secure upload of output evidence packages to a private S3 bucket
3	Leverage the principle of least privilege and allow for an external ID (EID) when performing cross-account actions

Problem Statement

FDC Are Expensive and Timely

On-Prem
≠
AWS

Rapidly Changing Environments at Scale

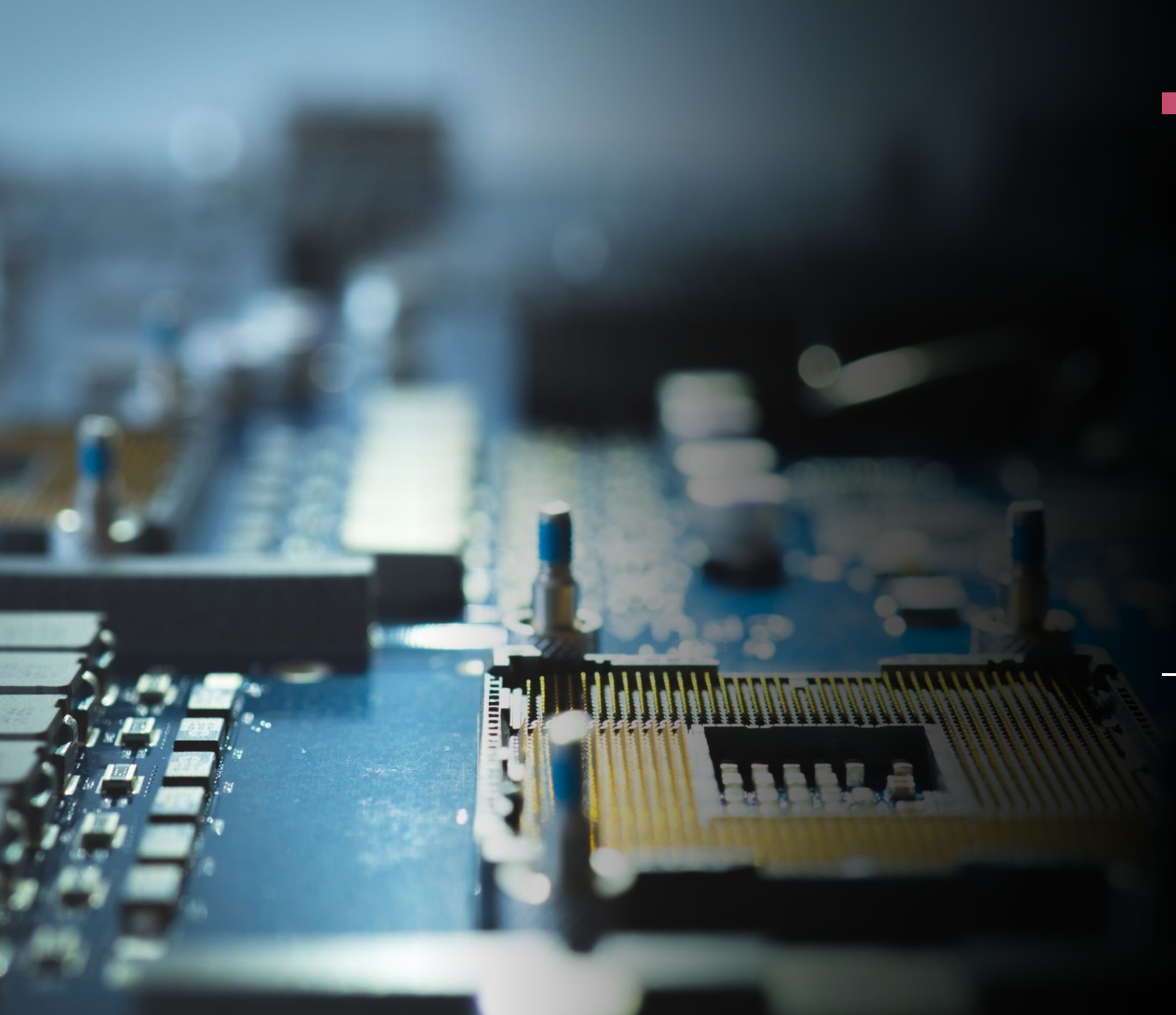
Rapidly Changing Environments at Scale

Benefits of a Triage Collection Workflow

Collect faster by collecting less

Standardized, auditable, and cloud-native approach

Can be done in parallel with a full disk collection



Solution Deep Dive

AWS Systems Manager (SSM)

- SSM allows for the automation of common and repetitive IT operations and management tasks. You also can execute Python, Bash, PowerShell, etc. scripts as part of a response action.
- Central to the functionality of our triage workflow.
- Requires an agent to be present on the target EC2 instance, and the target EC2 instance needs an instance profile with appropriate permissions to communicate with the SSM service.

Prerequisites

1. The target instance needs to have the SSM agent installed
   - Windows EC2 Amazon Machine Image (AMI) > November 2016 will have the agent pre-installed (e.g. Windows Server 2008-2012 R2 AMIs, 2016, and 2019)
   - Most Linux flavors will have the agent pre-installed (e.g. Amazon Linux, Amazon Linux 2, SUSE Linux Enterprise Server 12 and 15, and Ubuntu Server 16.04, 18.04, and 20.04)
   - The SSM Agent is preinstalled, by default, for macOS. This includes macOS 10.15.x (Catalina), macOS 10.14.x (Mojave), and macOS 11.x (BigSur).

Setting the Scene

UnauthorizedAccess:EC2/TorRelay

Finding ID: 0cb9b1ec82915cf247f759d34b2adce5

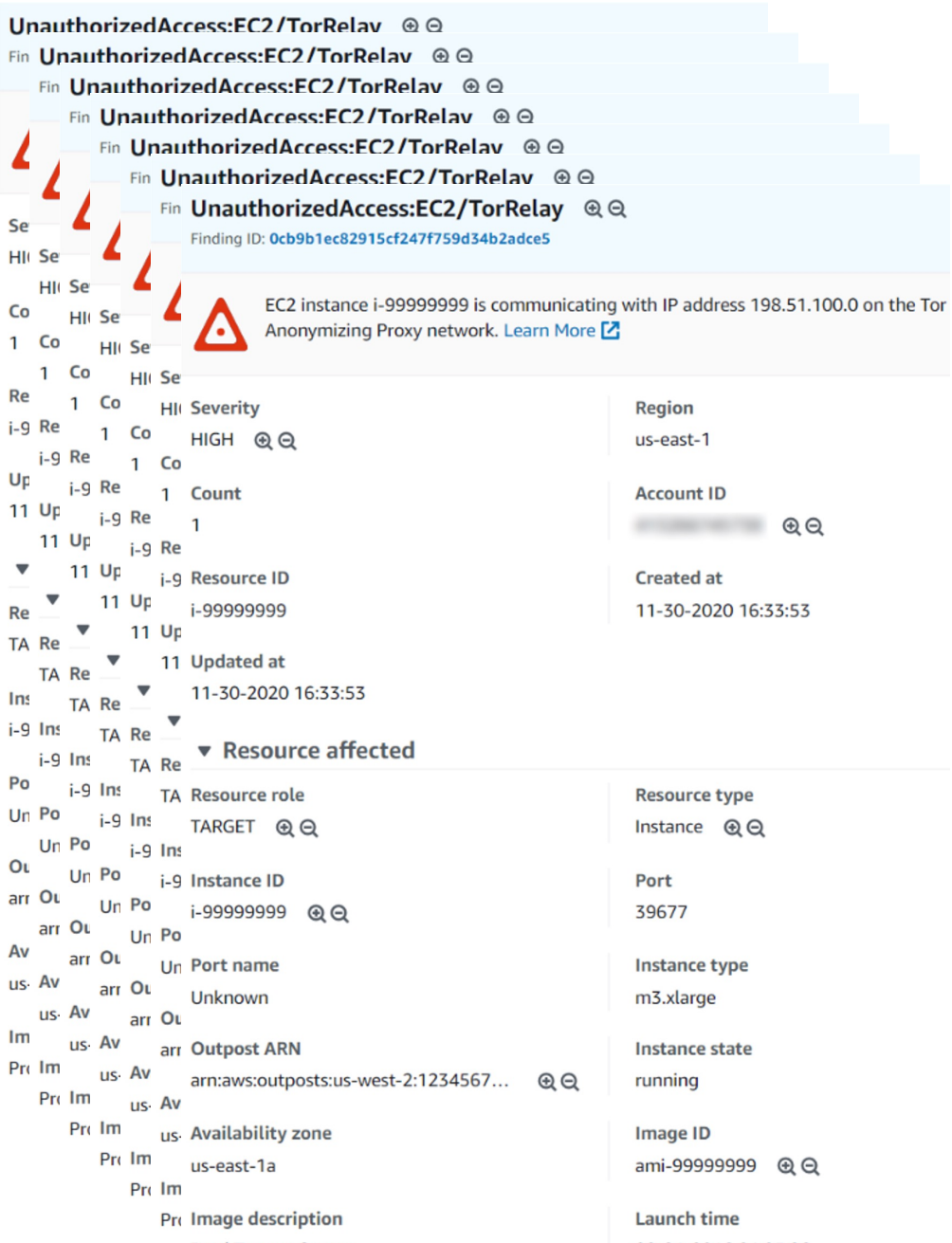
EC2 instance i-99999999 is communicating with IP address 198.51.100.0 on the Tor Anonymizing Proxy network. Learn More

Severity HIGH	**Region** us-east-1
Count 1	**Account ID** [redacted]
Resource ID i-99999999	**Created at** 11-30-2020 16:33:53
Updated at 11-30-2020 16:33:53	

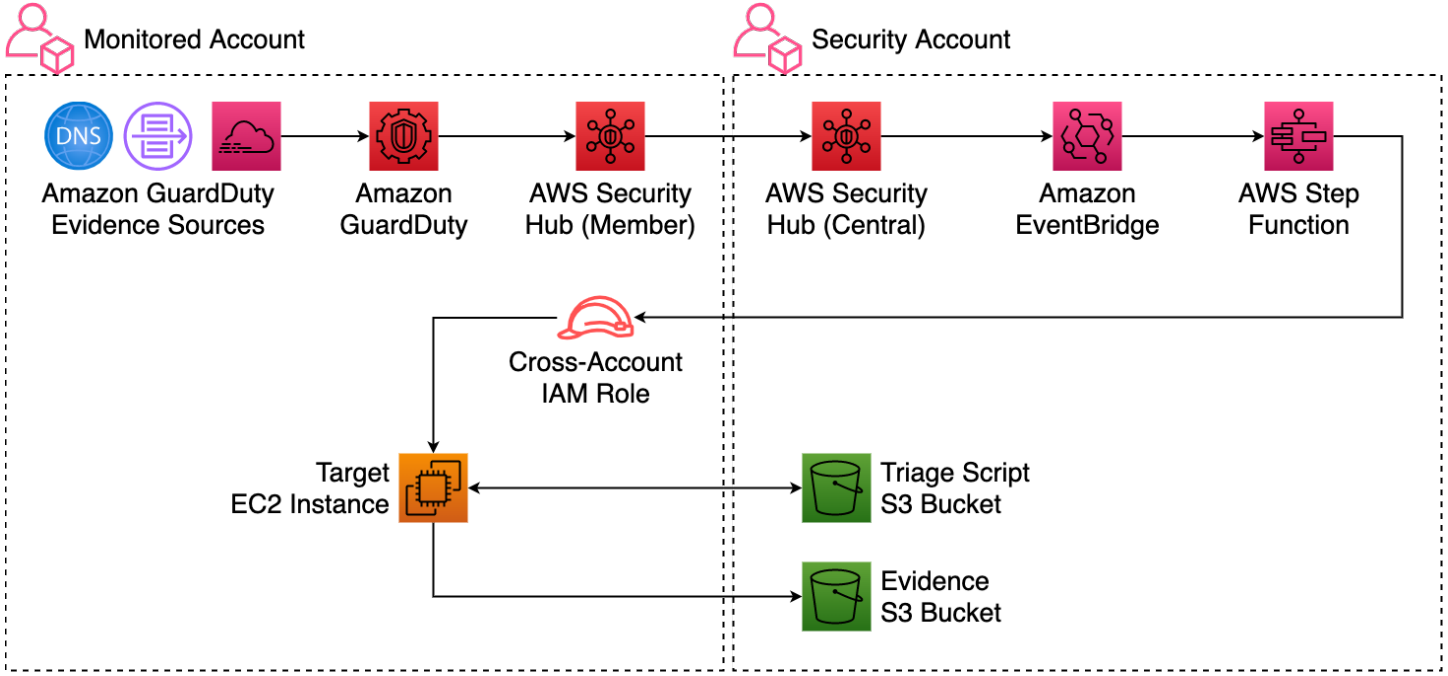
Resource affected

Resource role TARGET	**Resource type** Instance
Instance ID i-99999999	**Port** 39677
Port name Unknown	**Instance type** m3.xlarge
Outpost ARN arn:aws:outposts:us-west-2:1234567…	**Instance state** running
Availability zone us-east-1a	**Image ID** ami-99999999
Image description Prod Tomcat Server	**Launch time** 08-01-2016 21:05:06

Setting the Scene



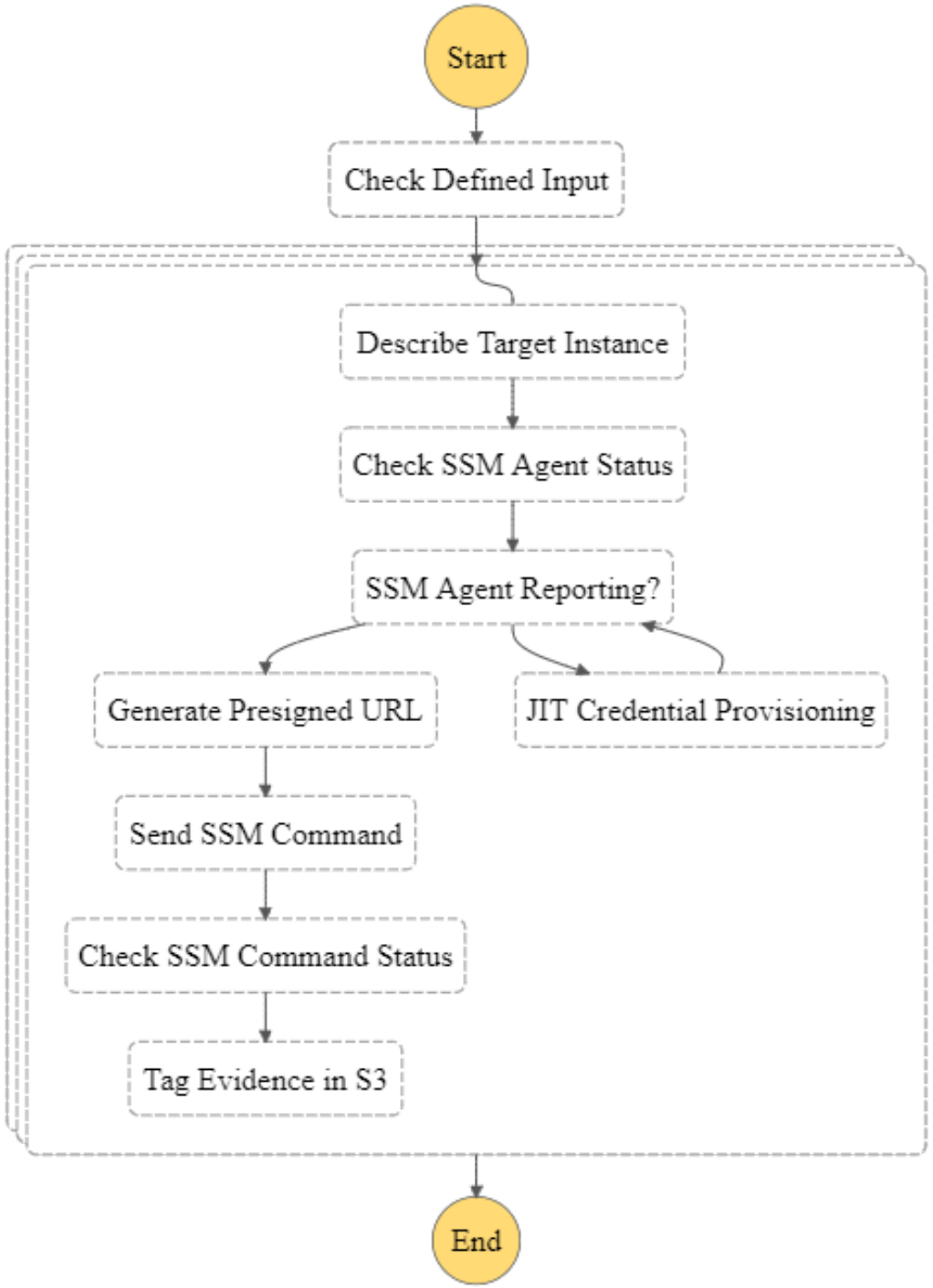
Workflow Diagram



Step Function

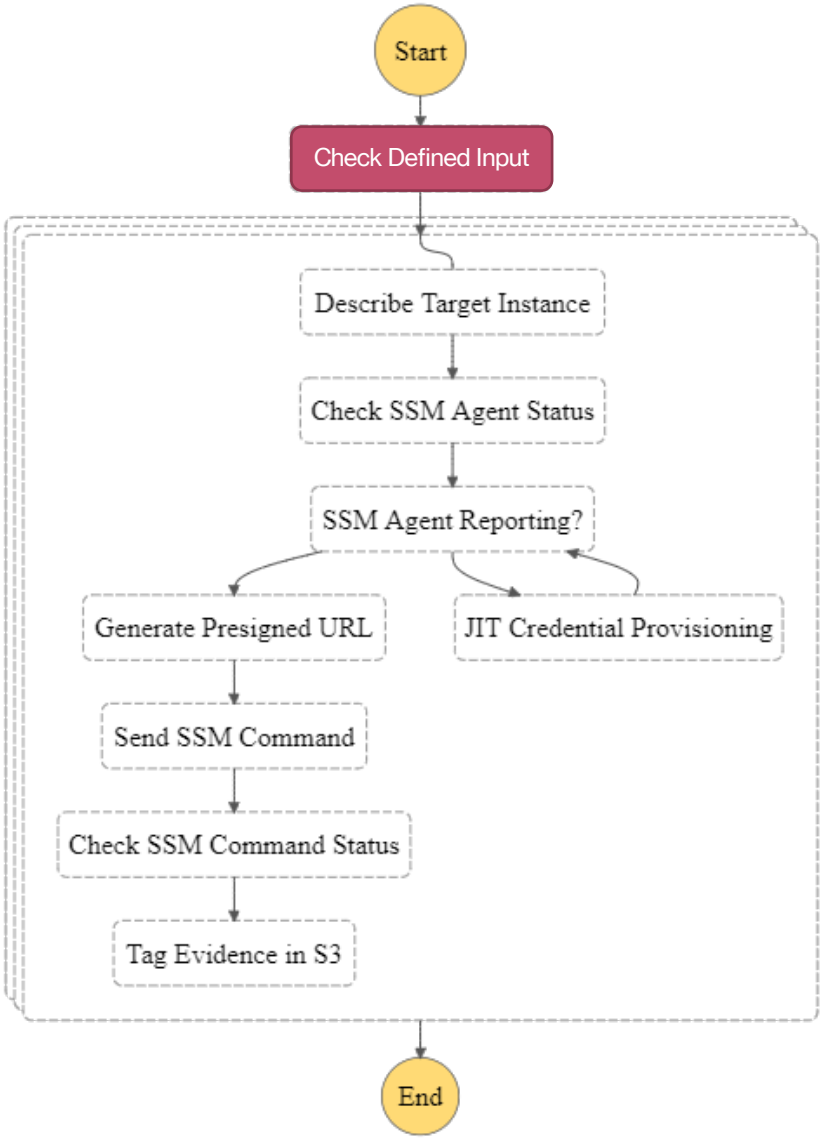
Inputs:

- ARN of Target Instance(s)
- Case ID



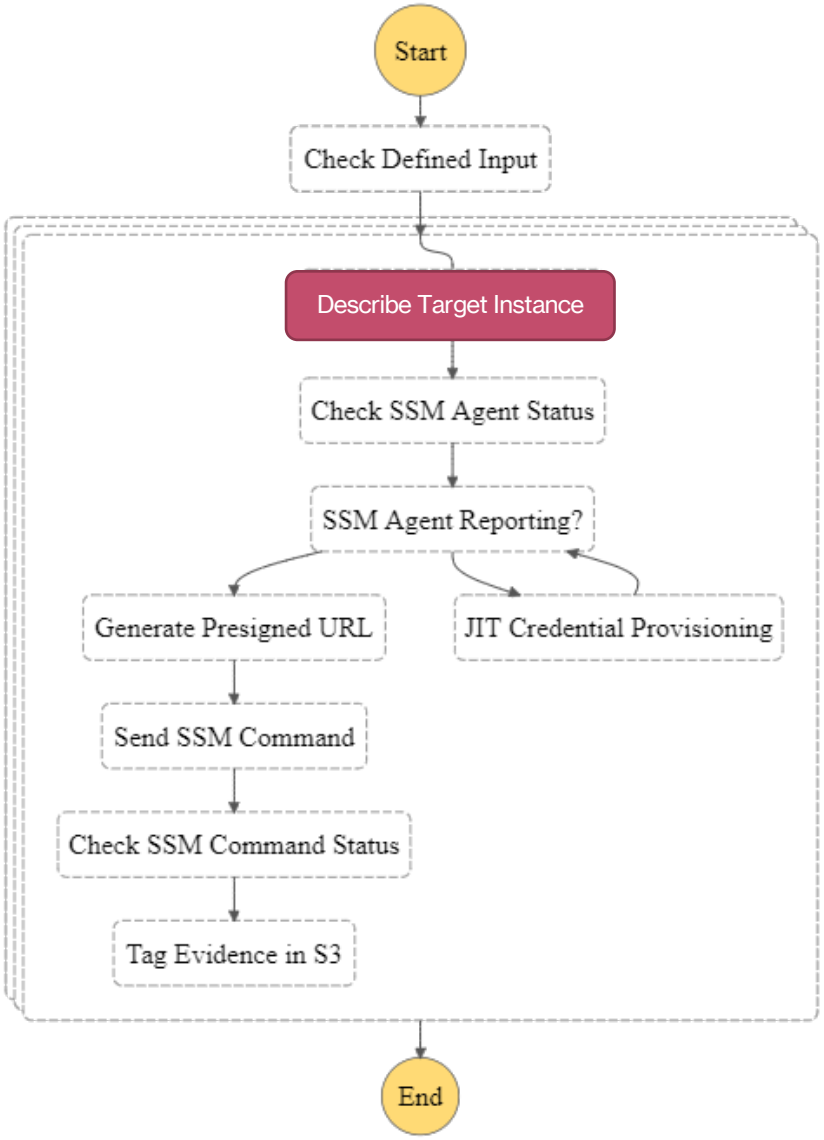
Check Defined Input

- This workflow can be triggered both automatically and manually
- Sanity checks input:
  - Account ID length of 12?
  - Instance ID valid?
  - Region valid?
  - Case ID unique?



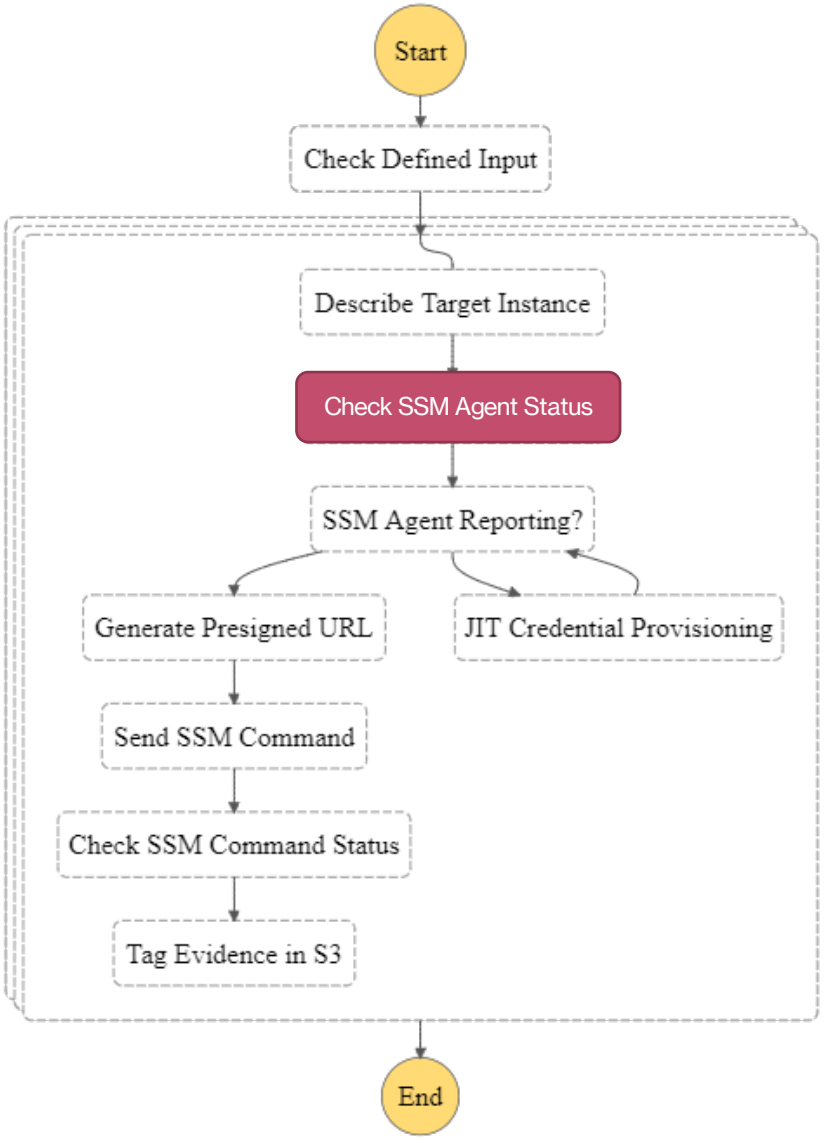
Describe Target Instance

- Returns metadata about the instance (public and private IP addresses, public and private DNS hostnames, volumes attached, creation date/time, OS type, etc.)
- Provides information at the end to the incident responder



Check SSM Status

- Is the instance reporting into the SSM service?
- If yes, move onto “Generate Presigned URL”
- If no, provision JIT credentials by modifying instance profile



SSM Agent Reporting?

- Assuming instance has the SSM agent installed – this is the one prerequisite
- Even if the agent is installed, the instance may not have the necessary permissions (instance profile) to allow it to communicate with the SSM service.
  1. Just-in-Time (JIT) access: modify the instance profile of the target EC2 instance on the fly to allow for the communication
  2. Existing connection: detect, alert on, and “auto-remediate” EC2 instances which don’t have this permission

Anonymous Write to Private S3 Bucket

Design Challenge

- How do I allow an EC2 instance without any AWS permission (e.g. does not have an instance profile) to write to a private S3 bucket?

Option 1

- Allow the triage script bucket to be publicly readable
- Allow the evidence bucket to be publicly writeable
- Monitor file uploads
- Worst case – a threat actor reads your triage scripts or writes files to your evidence bucket

Option 2

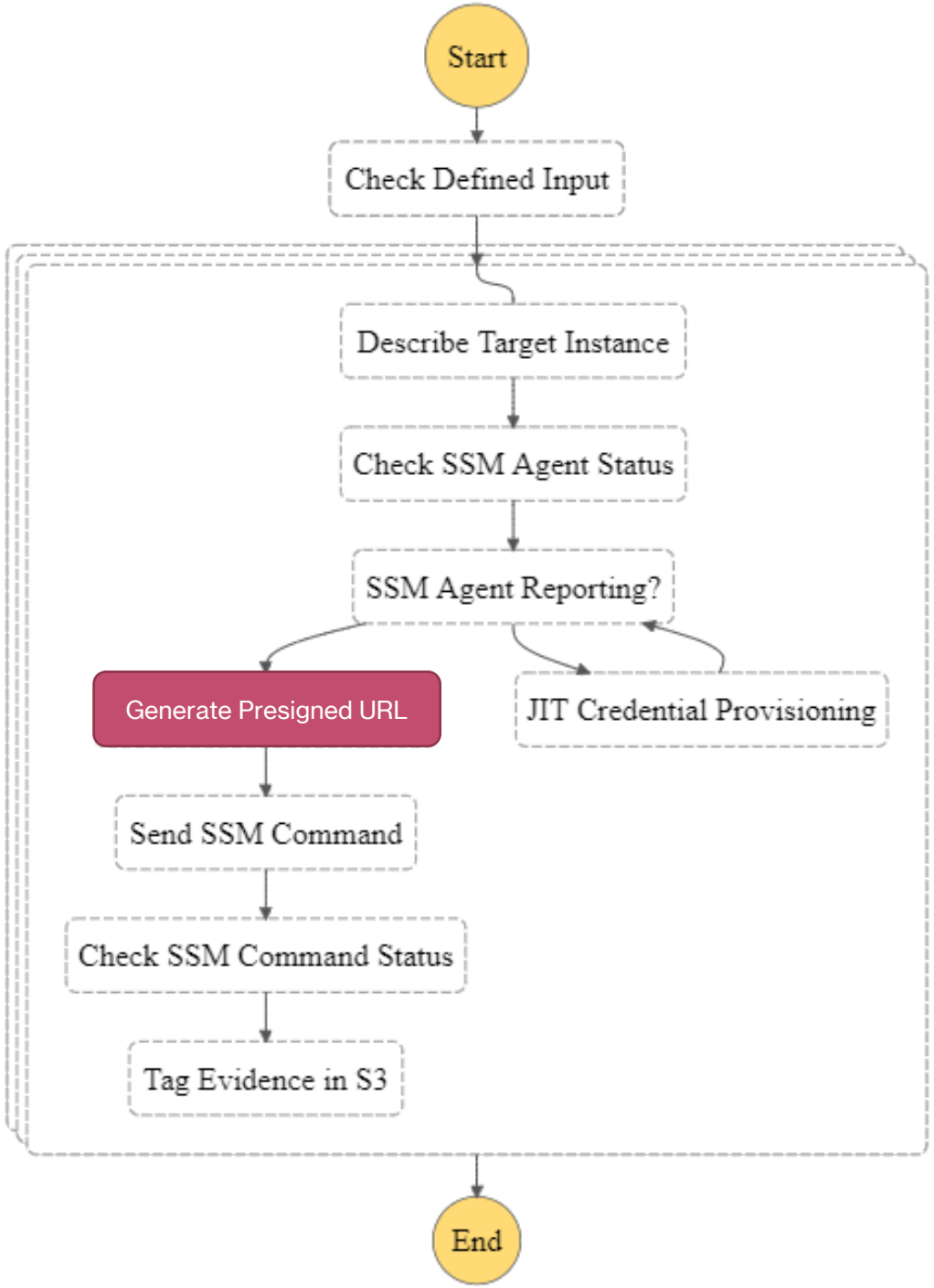
- Create a programmatic IAM user with permissions to only:
  - Read from the triage script bucket
  - Write to the evidence storage bucket
- Hard-code these keys in the triage collection scripts
- Rotate keys periodically and update your collection scripts with new keys

Option 3

- Leverage pre-signed URLs to generate temporary credentials that allow for the secure read of a file (from the triage scripts bucket) or the secure write of a file (to the evidence bucket)

- May seem like a clear winner, but has some design considerations

Generate Pre-signed URL



Pre-signed URL - Definition

- Allows for temporary read (GET) or write (PUT) access to a file within an S3 bucket
  - A pre-signed URL is generated by an AWS identity (user or role) that has access to the target object. The generated URL is then given to the unauthorized user.
  - During creation, the exact name of the target file to read (e.g. the triage collection script filename) or the name of the file that you’re going to upload (e.g. output evidence file) is needed
  - A pre-signed URL remains valid for a limited period (seconds to up to 7 days)

```python
import logging
import boto3
from botocore.exceptions import ClientError


def create_presigned_url(bucket_name, object_name, expiration=3600):
    """Generate a presigned URL to share an S3 object

    :param bucket_name: string
    :param object_name: string
    :param expiration: Time in seconds for the presigned URL to remain valid
    :return: Presigned URL as string. If error, returns None.
    """

    # Generate a presigned URL for the S3 object
    s3_client = boto3.client('s3')
    try:
        response = s3_client.generate_presigned_url('get_object',
                                                    Params={'Bucket': bucket_name,
                                                            'Key': object_name},
                                                    ExpiresIn=expiration)
    except ClientError as e:
        logging.error(e)
        return None

    # The response contains the presigned URL
    return response
```

Example code taken directly from: https://boto3.amazonaws.com/v1/documentation/api/latest/guide/s3-presigned-urls.html

Anatomy of a pre-signed URL

- https://s3-bucket-name.s3.amazonaws.com/FileToDownload.exe?AWSAccessKeyId=HJAW8JJTAKOOL93J11JA&Signature=J9AEcNnA3UQEshaS1HAEj12EQDfe30%3D&x-amz-security-token=VGhpcyBpcyBhbiBleGFtcGxlIHRva2VuIC0gbm90IGFuIGFjdHVhbCB0b2tlbi4gSWYgeW91J2QgbGlrZSB0byBnZW5lcmF0ZSB5b3VyIG93biwgcGxlYXNlIGdvIGxvb2sgb25saW5lIGluc3RlYWQgb2YgdHJ5aW5nIHRvIHN0ZWFsIHRoZSBvbmUgaW4gdGhpcyBwcmVzZW50YXRpb24K&Expires=1636229155

Anatomy of a pre-signed URL

- https://s3-bucket-name.s3.amazonaws.com/FileToDownload.exe?AWSAccessKeyId=HJAW8JJTAKOOL93J11JA&Signature=J9AEcNnA3UQEshaS1HAEj12EQDfe30%3D&x-amz-security-token=VGhpcyBpcyBhbiBleGFtcGxlIHRva2VuIC0gbm90IGFuIGFjdHVhbCB0b2tlbi4gSWYgeW91J2QgbGlrZSB0byBnZW5lcmF0ZSB5b3VyIG93biwgcGxlYXNlIGdvIGxvb2sgb25saW5lIGluc3RlYWQgb2YgdHJ5aW5nIHRvIHN0ZWFsIHRoZSBvbmUgaW4gdGhpcyBwcmVzZW50YXRpb24K&Expires=1636229155

Anatomy of a pre-signed URL

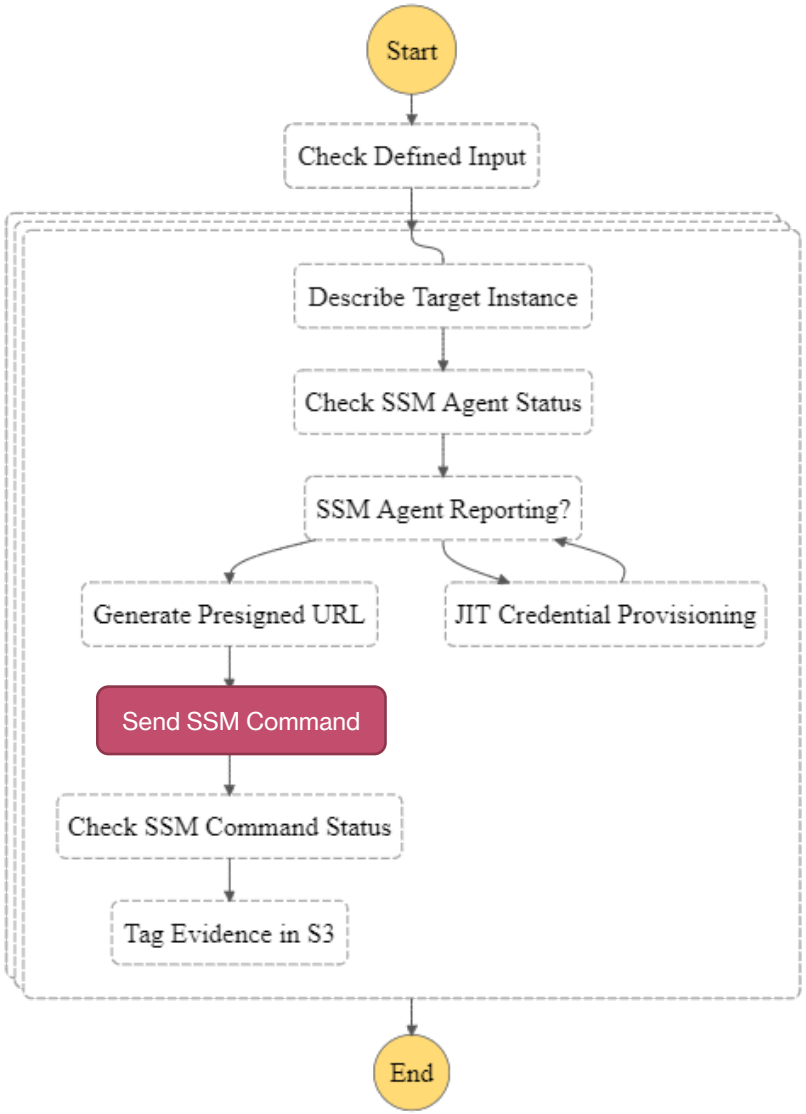
- https://s3-bucket-name.s3.amazonaws.com/FileToDownload.exe?AWSAccessKeyId=HJAW8JJTAKOOL93J11JA&Signature=J9AEcNnA3UQEshaS1HAEj12EQDfe30%3D&x-amz-security-token=VGhpcyBpcyBhbiBleGFtcGxlIHRva2VuIC0gbm90IGFuIGFjdHVhbCB0b2tlbi4gSWYgeW91J2QgbGlrZSB0byBnZW5lcmF0ZSB5b3VyIG93biwgcGxlYXNlIGdvIGxvb2sgb25saW5lIGluc3RlYWQgb2YgdHJ5aW5nIHRvIHN0ZWFsIHRoZSBvbmUgaW4gdGhpcyBwcmVzZW50YXRpb24K&Expires=1636229155

Pre-signed URL Use Case

- For each target instance, we need two pre-signed URLs: one to allow us to read the appropriate triage collection script, and another to output the evidence file created by the triage collection script to S3.

- Important to note that pre-signed URLs do not accept wildcards (*) during creation. You need to specify the exact filename.

Send SSM Command

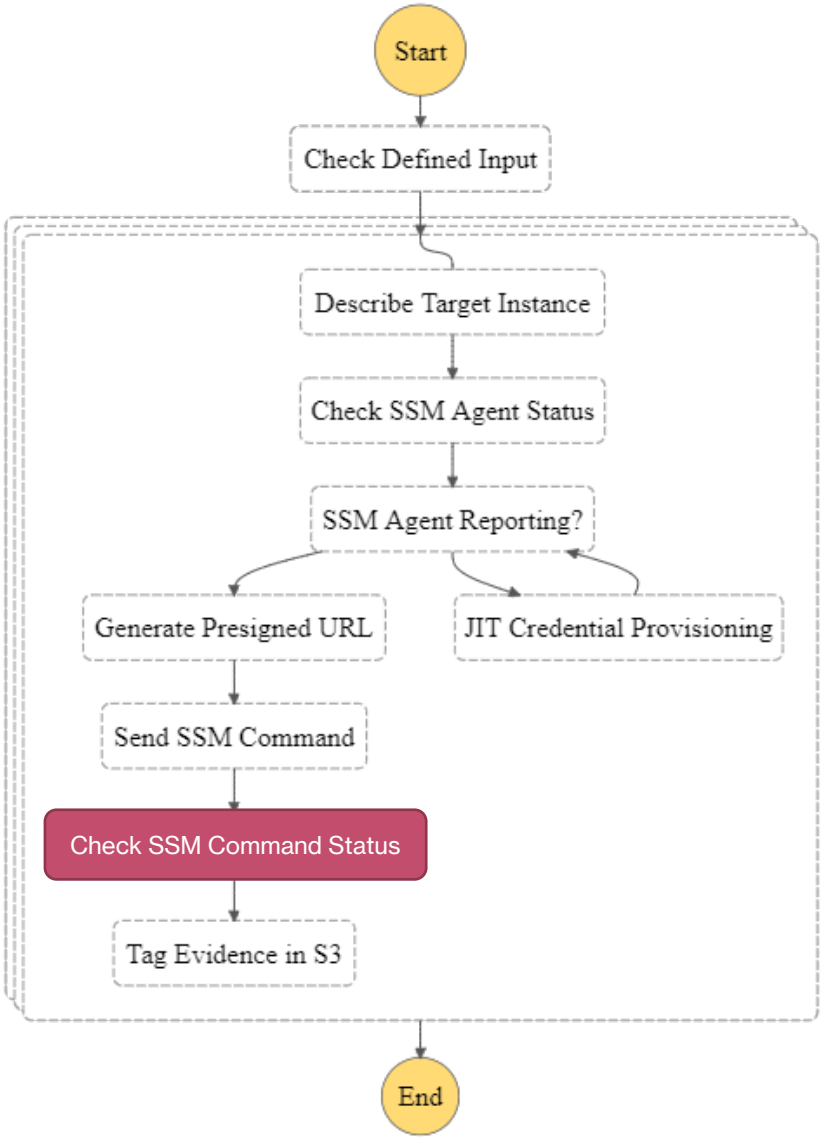


Executing Commands Through SSM

- AWS-RunPowerShellScript (Windows) or AWS-RunShellScript (Linux and macOS): Specify the PowerShell commands or Shell commands to execute on a target instance
- AWS-RunRemoteScript (Windows, Linux, or macOS): Execute scripts stored in a remote location (public and private GitHub or S3) on a target instance
  - Allows DFIR professional to update a script, without having to know a lot about the cloud

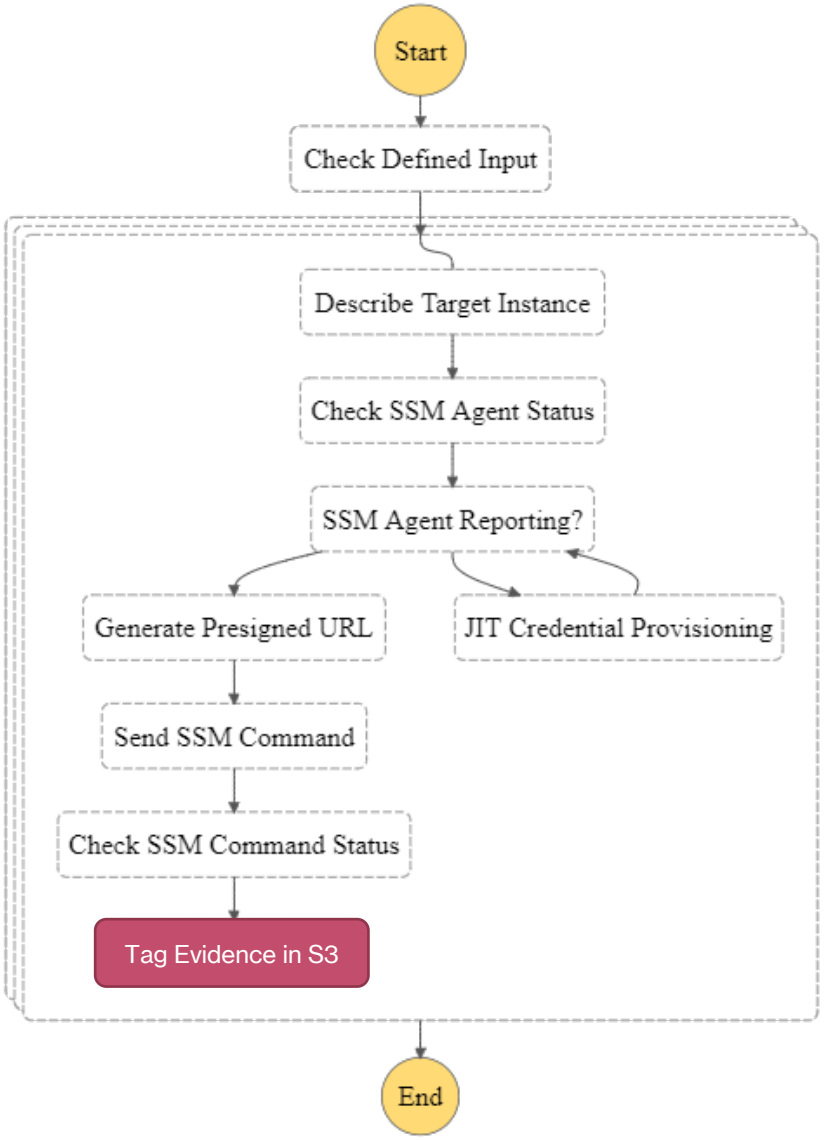
Check SSM Command Status

- Just because a command was sent via SSM, doesn't mean the command executed successfully.



Tag Evidence in S3

- For the output Zip file in S3, tag the file with the metadata gathered from “Describe Target Instance” state

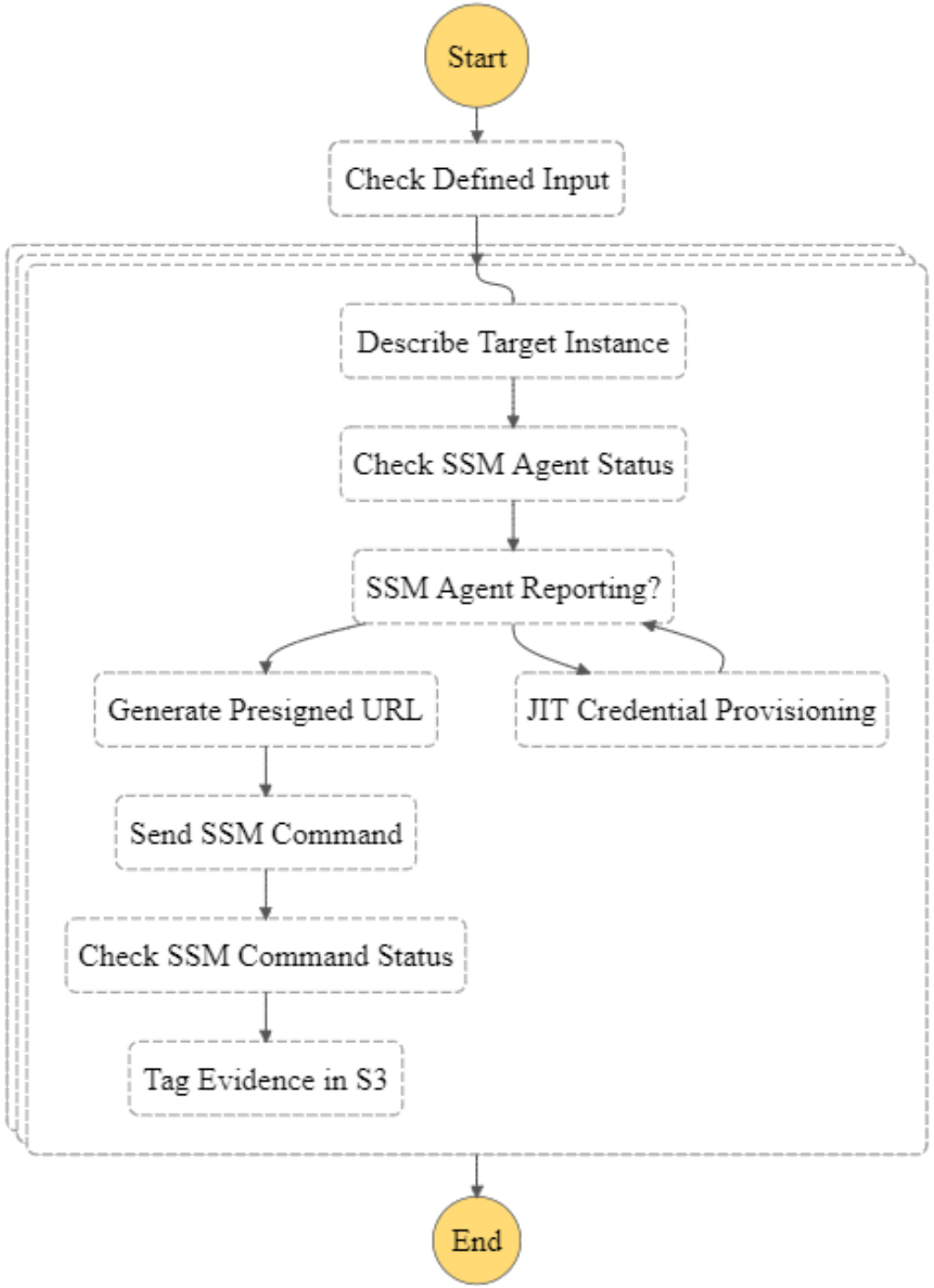


Example Output

Name	Size	Packed Size	Modified	Mode	User
Misc_Info	2 823 995	2 827 264	2021-08-17 12:42	0rwxr-xr-x	root
var	301 661	304 128	2021-08-17 12:42	0rwxr-xr-x	root
System_Info	77 319	80 384	2021-08-17 12:42	0rwxr-xr-x	root
Network_Info	12 719	14 336	2021-08-17 12:42	0rwxr-xr-x	root
etc	12 651	17 920	2021-08-17 12:42	0rwxr-xr-x	root
User_Account_Info_Activity	5 015	6 144	2021-08-17 12:42	0rwxr-xr-x	root
Suspicious_Files	1 243	2 048	2021-08-17 12:42	0rwxr-xr-x	root
ip-10-0-1-36-Collection-Log.txt	11 186	11 264	2021-08-17 12:42	0rw-r--r--	root

Name	Size	Packed Size	Modified
Extracted_Artifacts	127 030 368	23 069 276	
Extracted_MFT	309 329 920	27 580 715	
user_account_output.txt	16	18	2021-08-17 17:43
systeminfo_output.txt	3 347	1 115	2021-08-17 17:43
arp_output.txt	490	174	2021-08-17 17:43
ipconfig_output.txt	1 364	565	2021-08-17 17:43
nbrstat_c_output.txt	37	37	2021-08-17 17:43
nbtstat_S_output.txt	37	37	2021-08-17 17:43
net_sessions_output.txt	420	148	2021-08-17 17:43
net_file_output.txt	2 981	477	2021-08-17 17:43
net_share_output.txt	271	128	2021-08-17 17:43
netstat_output.txt	358	176	2021-08-17 17:43
autoruns_output.csv	825 306	178 336	2021-08-17 17:44
tmp.log	40 874	40 874	2021-08-17 17:44

Step Function



Pro Tips

Deployment and Testing Considerations

- Consider a multi-region deployment to create a highly available solution
- Roll out this solution in phases:
  1. Single test account (same-account, cross-region workflows)
  2. Start to onboard additional test accounts (via cross-account role with EID)
  3. Slowly shift additional accounts into the scope of solution
  4. Leverage the Amazon EventBridge to automate further
     - Native integration (e.g. GuardDuty presents a finding, and this solution runs)
     - Custom integration (e.g. EDR feeds into Security Hub, and this solution runs)

Security Considerations

- AWS Systems Manager (SSM) is a powerful tool. If misused, it can do more harm than good as most commands execute as root/Administrator.
- Consider locking down what users/roles have permissions to use SSM, update SSM documents, run commands, etc. via service control policies (SCPs) or permission boundaries.
- Create custom detections to alert when SSM is run or when someone attempts to use SSM

Additional Considerations

- Investigate Map states in Step Functions to allow for parallel execution
- Review the pros and cons of x86 vs. ARM Lambda types. If ARM Lambda types fit your use-case, AWS quotes ~20% better performance at a ~20% reduced cost (link)
- Leverage exponential backoffs/retries to fix certain errors
- Allows you to collect instance store data (EBS snapshots do not)
- Test and train

Limitations

- It may not be realistic for an organization to expect 100% of EC2 instances to have the SSM agent pre-installed.
- Helpful to pair this solution with the full disk collection solution to better ensure full coverage.

Further Learning

Additional Resources

- AWS Digital Forensics Automation (full disk collections and memory collections of potentially compromised EC2 instances): talk and slides
- Road to New Reality – Automating Incident Response in the Cloud (automated evidence processing and analysis): talk and slides
- AWS re:Invent SEC306: Automated Forensic Artifact Collection (custom AWS CloudTrail detections, enrichment, and auto-remediation at the AWS account level): talk and slides

Thank You

LinkedIn: /ryan-tick
Email: ryan.g.tick@gmail.com
Twitter: @tracer_tick
Discord: tracer_tick#4012