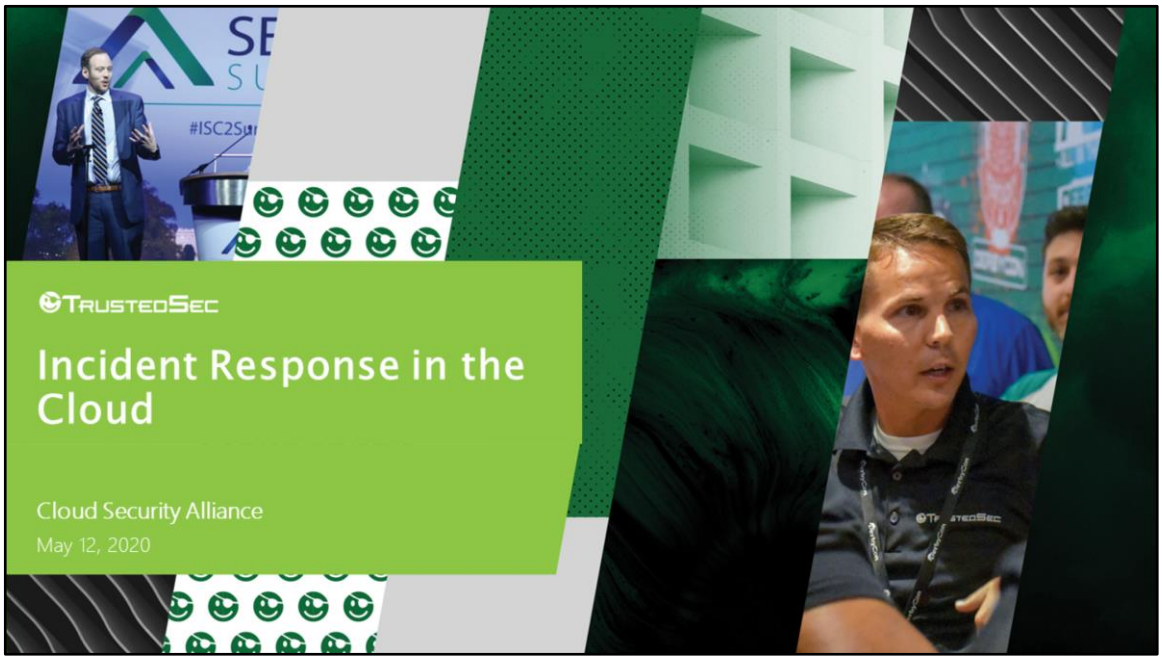

TrustedSec
Incident Response in the Cloud
Cloud Security Alliance
May 12, 2020

Tyler Hudak

Practice Lead, Incident Response

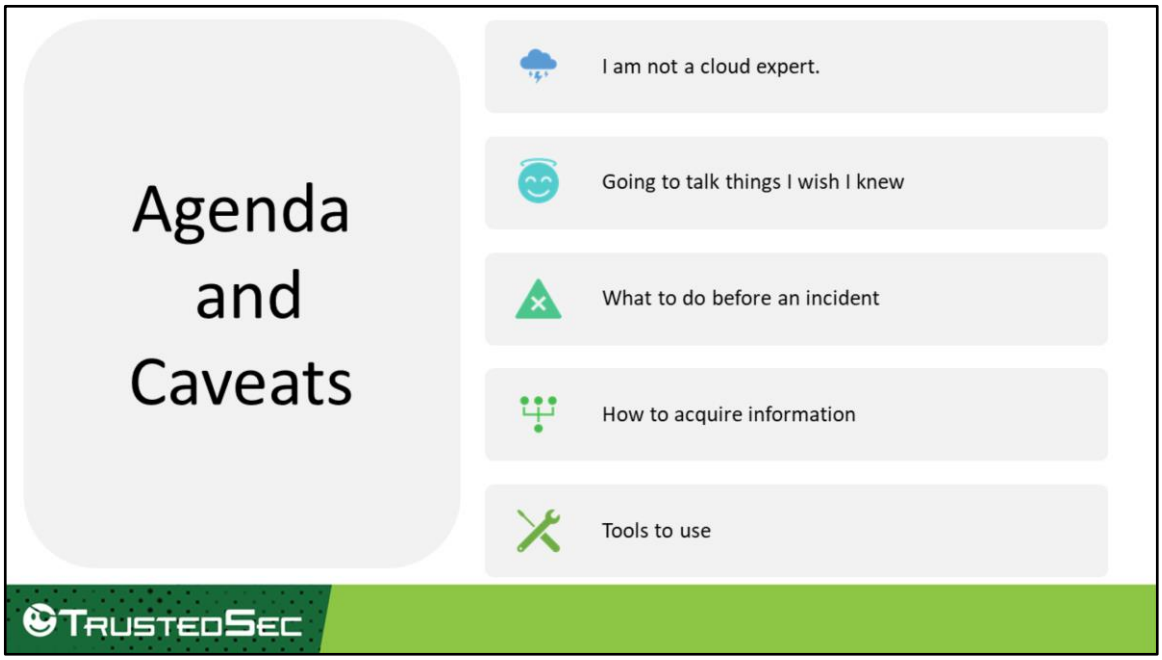
I have almost 20 years of experience in information security and have had multiple roles, including on the red team and in incident response (my current passion).

Huge geek and nerd.

Education & Certifications -
BS Computer Science, University of Akron, GIAC Certified Forensic Analyst (GCFA), GIAC Certified Forensic Examiner (GCFE)

tyler.hudak@trustedsec.com
@secshoggoth

TrustedSec



Full disclosure: I am not a cloud expert! I learn something new every time I get into the cloud environment to do a response (I even picked up a few things for this talk!) So, think of this more of an intro to IR talk and I'm going to talk through things I wish someone had told me when I started. This is going to include:

- What to do before an incident happens
- How to get information out of the cloud
- What tools you can use to make this easier

As organizations put more resources into the cloud, they must stay aware of the security of their systems and data. This includes not only locking systems, network, and data is the cloud down, but also being prepared for the inevitable incident response that will need to happen. The good news is that despite this being a relatively new environment for Information Security, many of the techniques and concepts of performing on-prem IR have stayed the same. The bad news is that the environment has become more complex, requiring additional specialized knowledge on how each cloud environment works.

IR is broken up into different phases and you could discuss each of these IR phases specific to the cloud in their own talk. Instead of talking about the entire IR process or even a specific phase, I'm going to talk about my own experiences with IR in the cloud, the common issues I've come across, and how to overcome those. The goal of this is to help those who are just starting to perform IR in the cloud get over that initial hump of how to get things done.

We're also only going to focus on the most popular players at this time – Amazon and Microsoft.

What's new?

The Good

- Some collection easier and faster
- Containment easier
- Everything in one (virtual) location
- Architecture relatively consistent
- Built-in automation

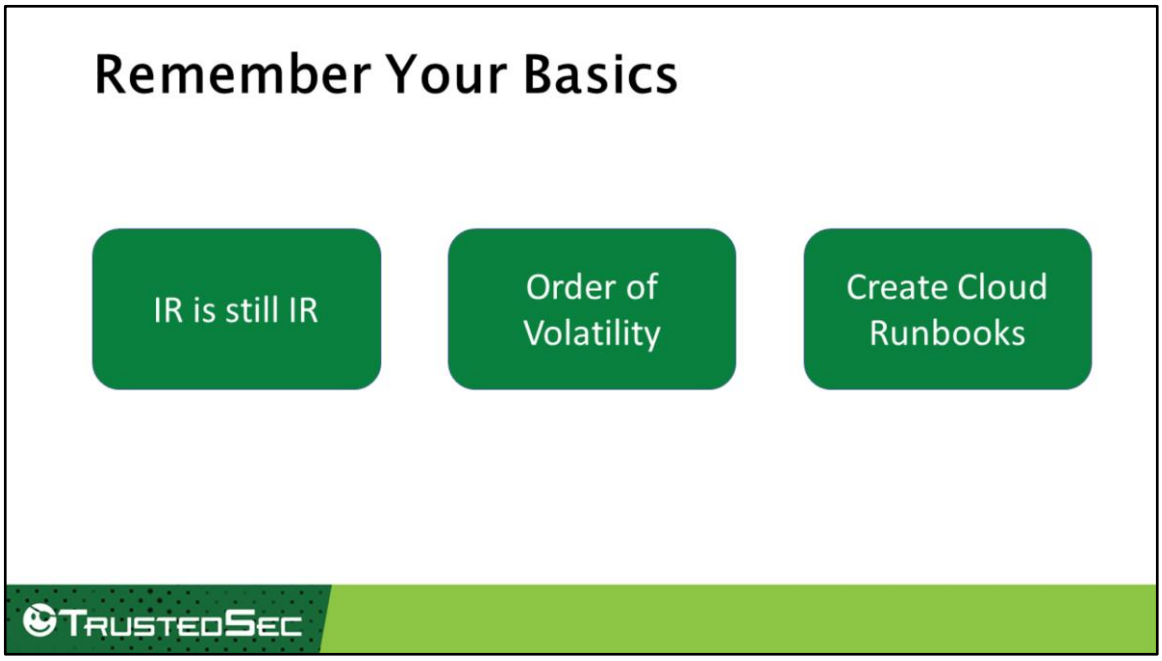
The Difficulties

- Missing some IR capabilities
- Containment can be more complication
- Every cloud provider is different
- Lots of unknowns
- Capabilities vary by license

TrustedSec

Before an Incident

TrustedSec



The big thing to remember when performing IR in the cloud is that its still IR. The basic incident response and forensic techniques and best practices still apply. In the end, you are still going to be investigating, containing, eradicating, and recovering systems and data. The only thing that has changed is the platform its on.

Don't forget the order of volatility when performing IR in the cloud. Fortunately, most cloud security services (e.g. Azure Sentinel or ATP) keep data for a while, but you will still need to grab that volatile data from systems.

Runbooks are IR procedures which describe what actions should be performed during an incident, and potentially how they are performed. If you don't have them, create them. If you don't have any that are specific to your cloud environments, create those! These help prevent "analysis paralysis" in the middle of an incident when an investigator hits a wall they haven't come across before.

Forensic Workstations

Deploy forensic analysis systems in the cloud

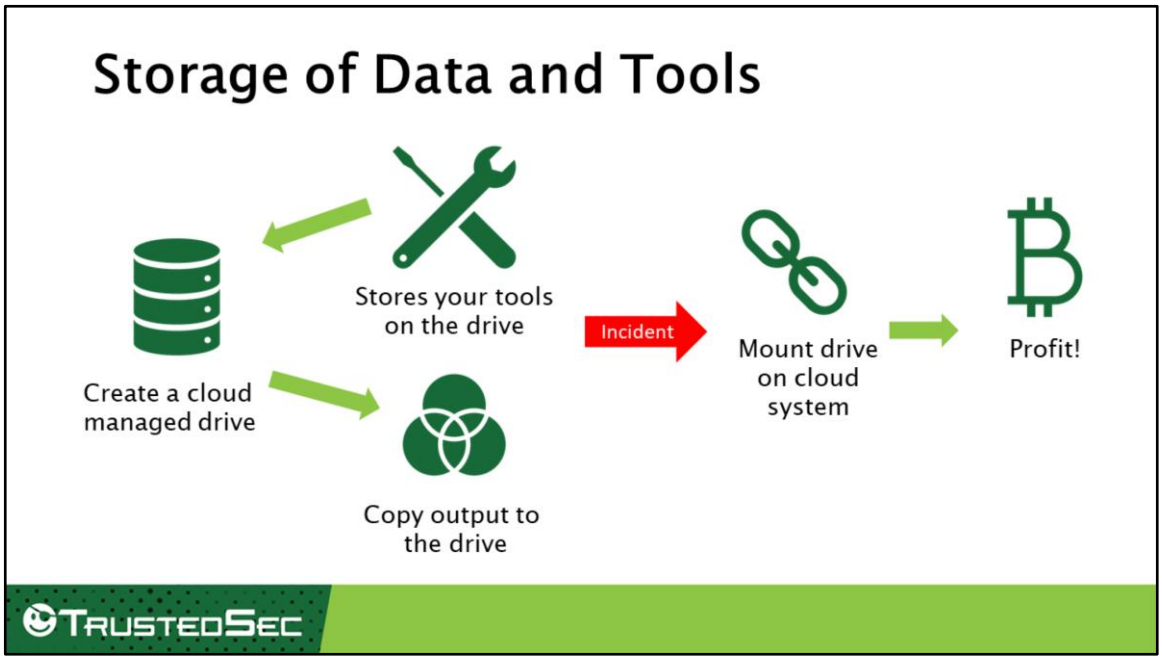
- ✓ Create one image, share out
- ✓ Clean environments each analysis
- ✓ Mount images directly to the workstation

- ✗ Don't upload on-prem images
- ✗ What if your Internet is cut off?

TrustedSec

Incidents can generate a lot of data, such as memory images and disks. While you'll still want to grab this data during your IR, you don't necessarily want to be downloading all of that from the cloud. Think about it – do you really want to be downloading potentially hundreds of GBs of data in order to analyze it? Of course not! So, use the cloud to your advantage.

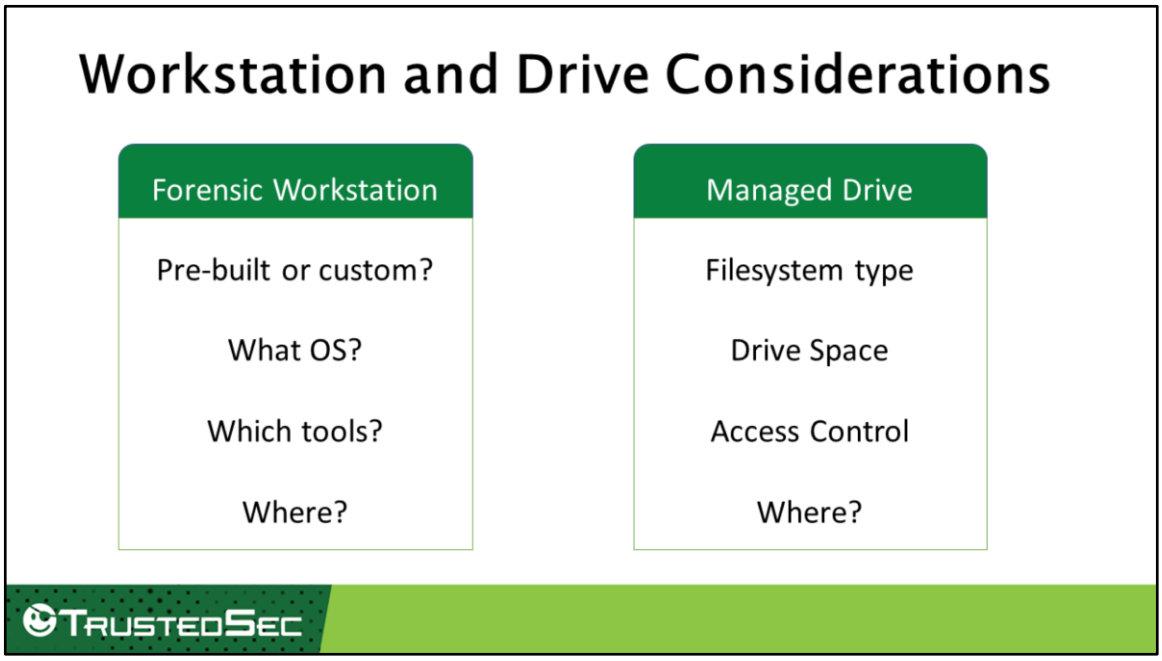
The best way is to create a forensic workstation in your cloud environment. This is just a fancy way of saying you should pre-deploy one or more images in your cloud environment that you can use for forensic analysis of your cloud systems.



As we stated, during IR you are going to be gathering a lot of data for analysis. Some will be disk images, but others like memory images and volatile data, will need some place to go. Additionally, you'll need to store your tools somewhere that is accessible from your cloud systems. One of the easiest ways to do this is to create a data and tool drive in your cloud instance.

Azure and AWS allow you to create a managed drive in the cloud – essentially a disk volume that you can share with any cloud system in your instance. By creating a managed drive, formatting it with the appropriate file system, and storing your tools on it, it becomes a shared drive that can be added to any cloud system you need to perform analysis on.

This means that when you have an incident, you can attach the drive to that system, run your data collection tools and store their output to a locally connected drive on the system. There is no need to try to access tools or store data over the network, which would likely be much slower. It also prevents you from needing to store data on the local drive, which could remove forensic data.



When creating forensic workstations and managed drives there are a few considerations that should be kept.

For workstations:

- Do you want a pre-built workstation, or do you want to build one on your own? Each cloud marketplace has some forensic workstations that can be used for free or purchase, and there are multiple forensics distros that could be used. Using a pre-built/ISO is nice because you don't have to build or maintain anything. However, it might not be in the right OS for what you need and might not have the right tools. Building a workstation will make sure you have everything you want.

- What OS do you want to use? Your choice will be dependent upon which tools you want to use and your (and your team's) comfort level with that OS. Of course, there's nothing to say you can't build multiple workstations.

- What tools do you want installed? The answer to this question will be heavily influenced by the previous two questions, but its good to keep in mind which tools you want readily available especially if some will have special licensing

requirements.

- Finally, where do you want the workstation? Some cloud instances restrict cloud systems to specific regions so you might need to plan for multiple workstations to go across your entire cloud infrastructure.

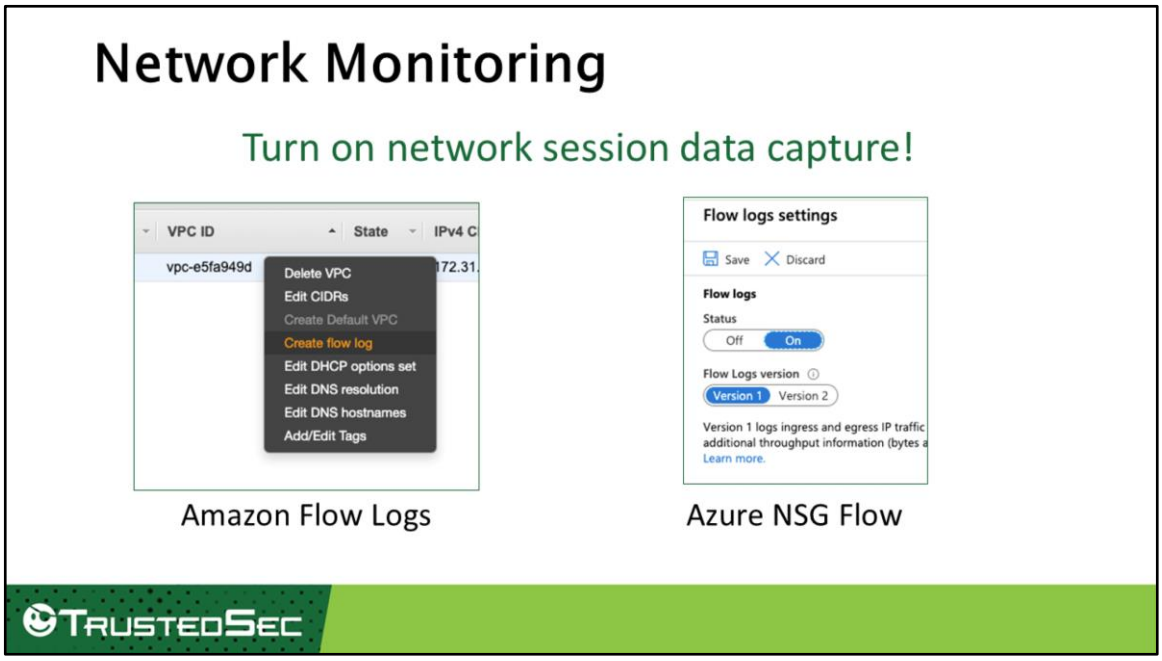
For the managed drive:

- What filesystem will it be formatted with? You want to be sure you format the drive with a filesystem that will be accessible by your systems. Filesystem file size limitations also need to be taken into consideration. For example, the FAT32 file system is easily mountable on both Windows and Linux, but it has a file size limitation of 4GB. That might sound like a lot, but what if you need to acquire 32GB of a memory image and your tool doesn't know how to split it up? NTFS, on the other hand, has a 16TB file size limitation but may not be writable on some Linux systems.

- The amount of drive space needs to be considered as well. You want to have enough space for your tools and their output. Remember that some tools – like memory acquisition tools – will generate VERY large files.

- Who can access it? If possible, restrict access to the managed drive and only open it up as necessary. This will help keep the integrity of your data.

- Finally, the same "where" question for the forensic workstation applies to the managed drive.



Just like in-on-prep infrastructures, network monitoring is extremely important to determining network-based activity. AWS and Azure provide the capability to capture network flow logs, similar to NetFlow. These logs will describe source and destination addresses and ports, protocol, packet and byte counts, and times of the connection. You won't get full introspection into the traffic though.

Amazon does this through Flow Logs. To turn them on, go into your Virtual Private Cloud (VPC) interface, right click on the VPC and select "Create flow log". Logs can be sent to an S3 bucket or your CloudWatch logs. Full PCAP may be captured through VPC Traffic Mirroring (https://aws.amazon.com/blogs/aws/new-vpc-traffic-mirroring/) but there is an additional cost and it is only possible using the AWS Nitro systems.

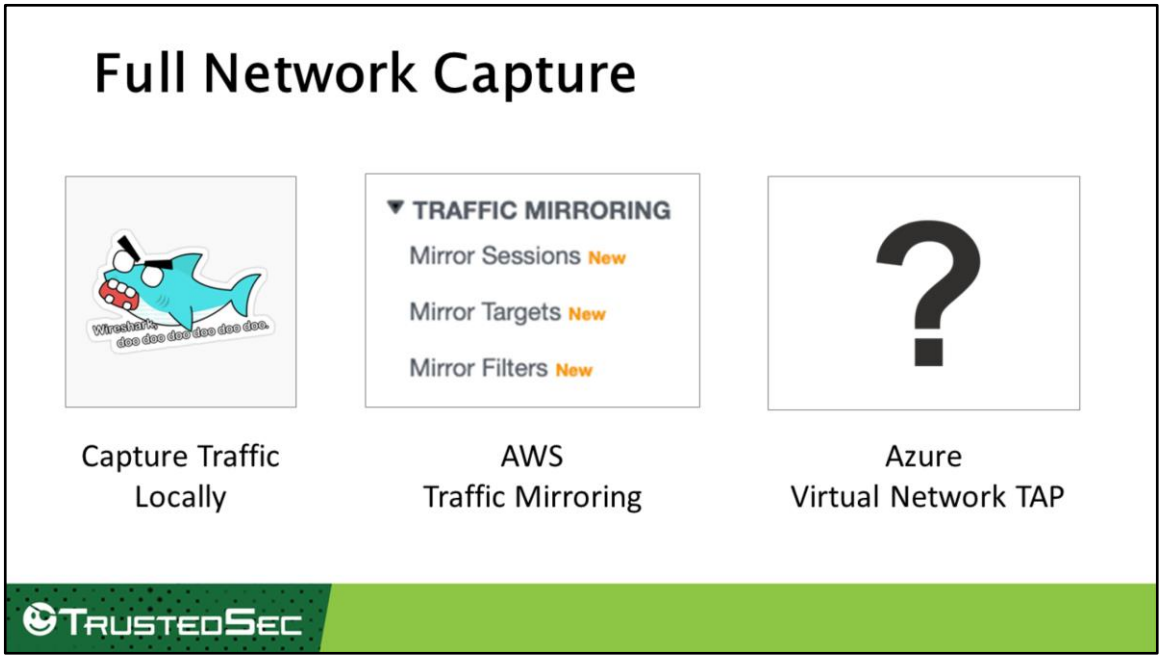
Microsoft has created Network Security Group Flow to do the same. To turn this on, go into the Network Watcher interface and select Logs->NSG flow logs. Click on the name of the NSG you wish to turn flow logs on for and change status to on. To access the flow logs, go into the storage account that you configured for NSG flow logs, select Blob service, Containers, the insight-logs entry, and then drill down until you find PT1H.json. Select the … next to the name and choose download.

(https://docs.microsoft.com/en-us/azure/network-watcher/network-watcher-nsg-flow-logging-portal#download-flow-log)

Azure also offers Traffic Analytics which provides additional visualizations and analysis of your flow logs. However, this requires a Log Analytics Workspace, which adds additional costs and is not available everywhere.

Both Azure and AWS also allow configuration through command line tools and APIs.

https://docs.microsoft.com/en-us/azure/network-watcher/network-watcher-nsg-flow-logging-overview
https://docs.microsoft.com/en-us/azure/network-watcher/network-watcher-nsg-flow-logging-portal#download-flow-log
https://docs.microsoft.com/en-us/azure/network-watcher/network-watcher-nsg-flow-logging-cli
https://docs.microsoft.com/en-us/azure/network-watcher/traffic-analytics
https://docs.aws.amazon.com/vpc/latest/userguide/flow-logs.html
https://docs.microsoft.com/en-us/azure/network-watcher/traffic-analytics



Both cloud environments allow some methods of capturing full network traffic of data.

Remember, there is always the option of running tcpdump, windump, or wireshark on a host system and capturing network traffic to and from a system that way. (BTW that sticker is from https://www.redbubble.com/i/sticker/angry-zombie-wireshark-by-shortstack/36407775.EJUG5 – go buy it. I have no affiliation with it)

AWS allows Traffic Mirroring in the VPC (Virtual Private Cloud). This service creates a copy of the traffic within a VPC and send it to another VM, such as a network monitoring solution (e.g. Zeek) you set up in your cloud. You can find the interface for this by into your VPC region interface, scrolling down and selecting Traffic Mirroring. Follow the documentation at https://docs.aws.amazon.com/vpc/latest/mirroring/traffic-mirroring-getting-started.html for how to turn it on.

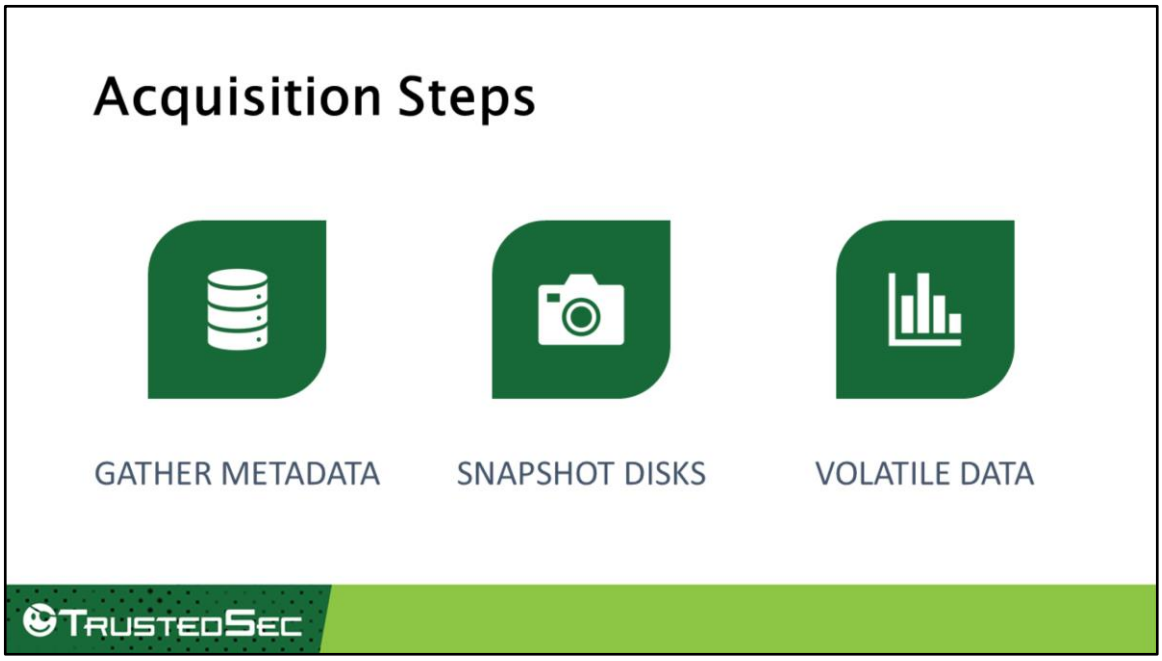
Microsoft is also ”previewing” virtual network TAP in Azure which provides full packet capture, but as far as I have seen it has not been released yet. (https://docs.microsoft.com/en-us/azure/virtual-network/virtual-network-tap-

overview)

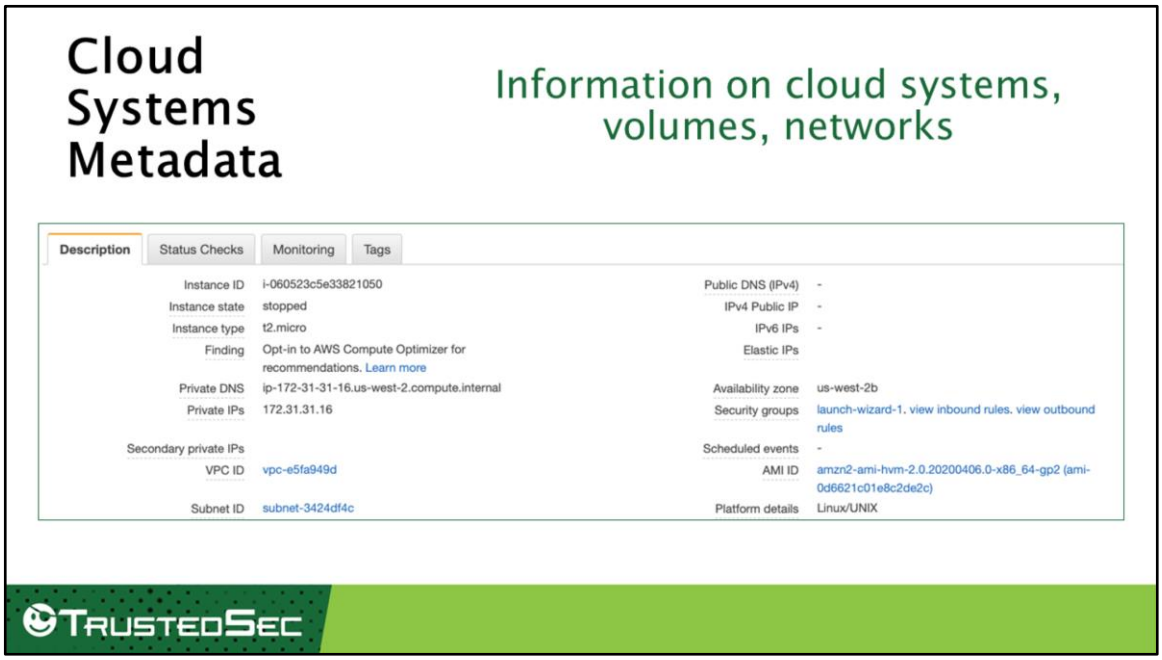
https://docs.aws.amazon.com/vpc/latest/mirroring/what-is-traffic-mirroring.html

Data Acquisition

TrustedSec

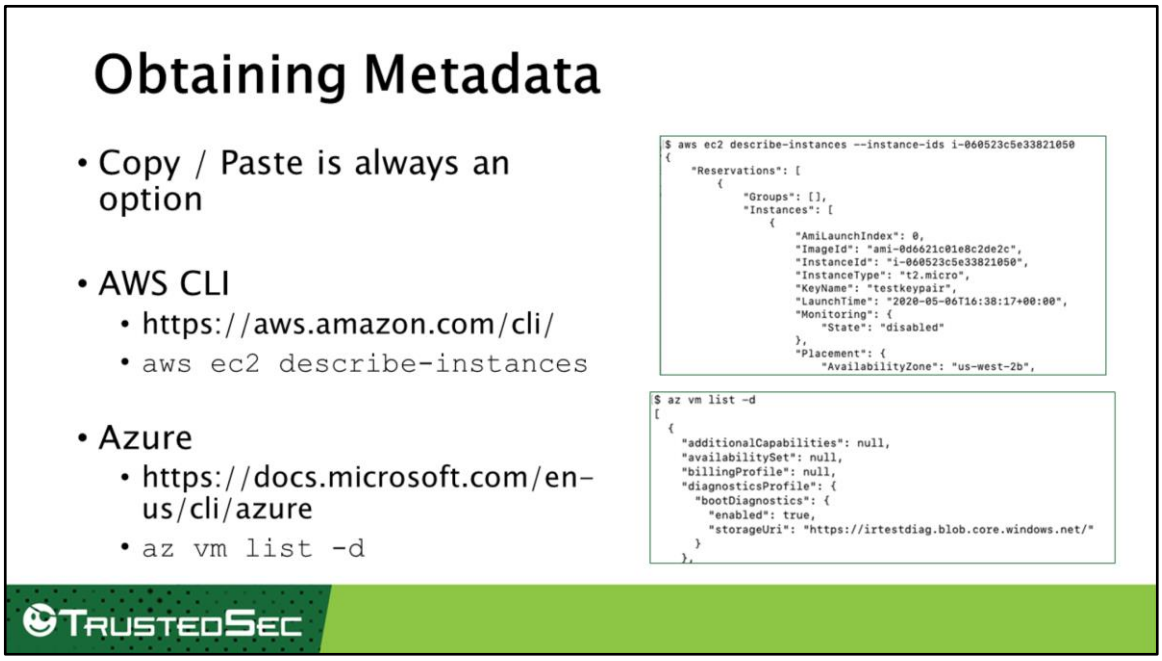


Once an incident happens in the cloud, there are a few things you'll want to do right away: gather the metadata of the hosts, snapshot the disks, and gather volatile information from the affected systems.



Organizations could have dozens to hundreds of cloud systems in their VPCs. Having to perform analysis on only a few of them could get confusing, especially since a lot of systems have numerical IDs associated with them, may have multiple volumes attached, and could be in their own private networks. Therefore, one of the first things you should grab in an incident is the metadata for the systems you are responding to.

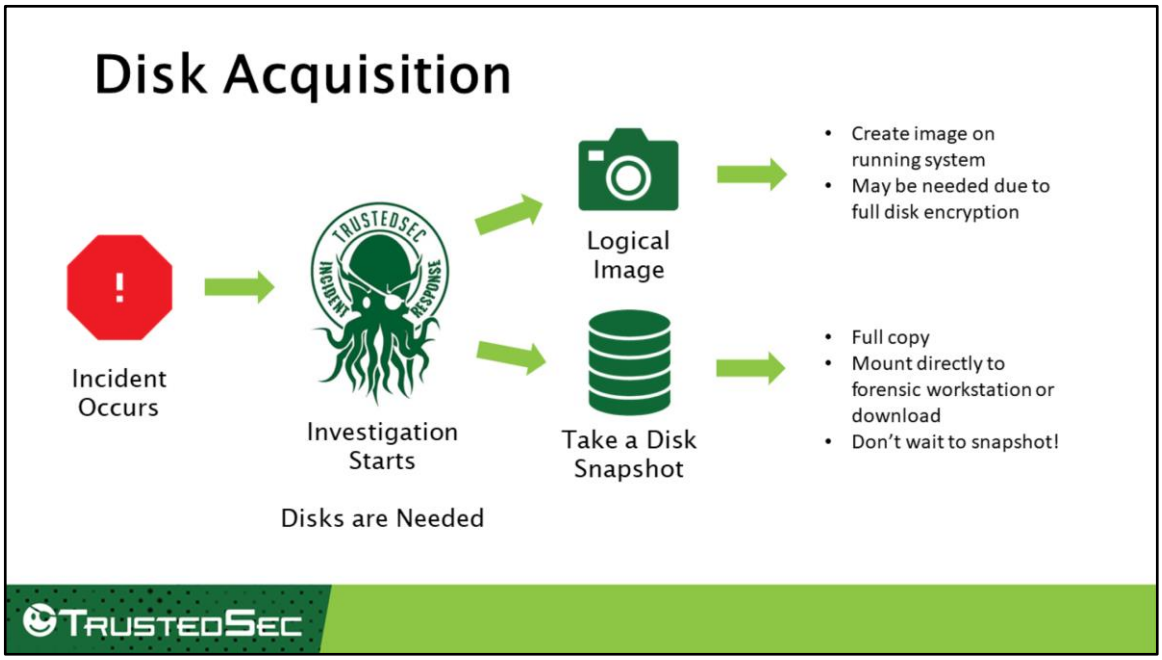
Obtaining the metadata in the early stages of the incident will allow investigators to ensure they are responding to the right systems and will ensure that data is captured before it is forgotten about. Additionally, some other response tasks might need this information to properly pull back the data.



There are multiple techniques to obtain the metadata of cloud systems. The easiest, although likely most tedious, is to just copy and paste it from the interface. Outside of that, there are a few API interfaces we can use to obtain the data.

For Amazon AWS, the aws command line tools (https://aws.amazon.com/cli/) provide a method to obtain metadata. This can be accomplished by running the command *aws ec2 describe-instances --instance-ids <YOUR INSTANCE IDS>* which then displays the instance metadata in JSON.

For Azure, the az command line tools can be used to obtain metadata for VMs. This can be done by running the *az vm list* command.

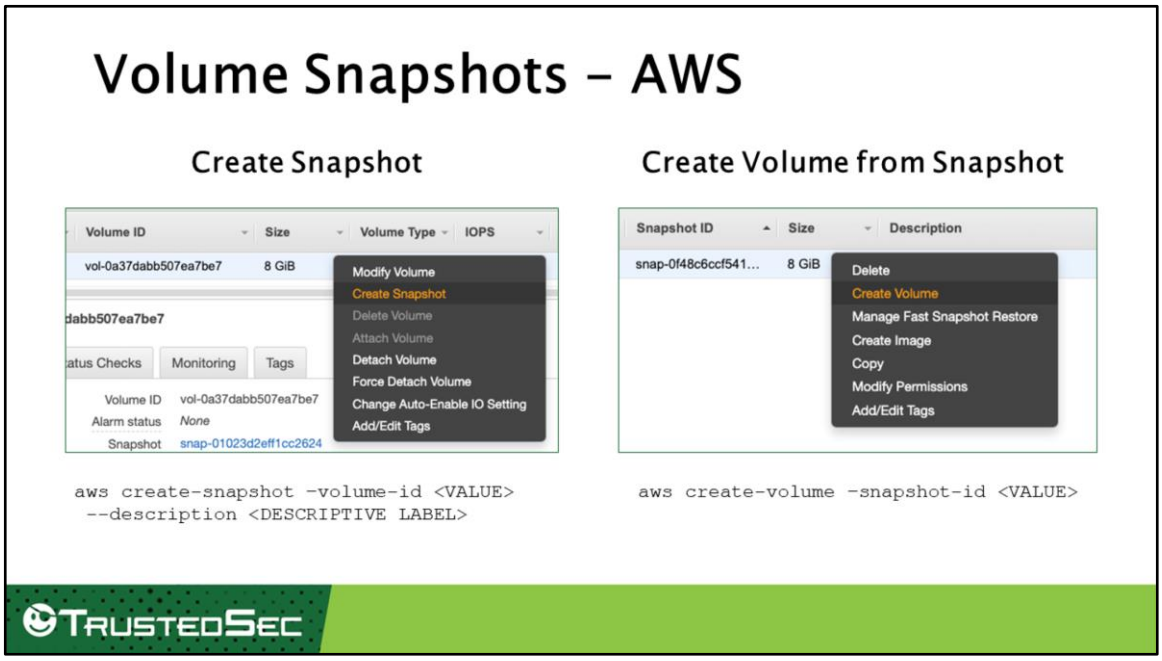


As with all IR investigations, eventually the disks of the cloud systems will be needed for analysis. The good news is that this is one area in which the cloud makes this easier!

One potential method to capture the disk is to perform a logical image. This is where you utilize software to image the drive while it is live. One potential reason for needing to do this is if the disk has full disk encryption, such as with BitLocker. Software such as FTK Imager allows you to do this easily. Just make sure you image it to your forensic drive and not to the local drove.

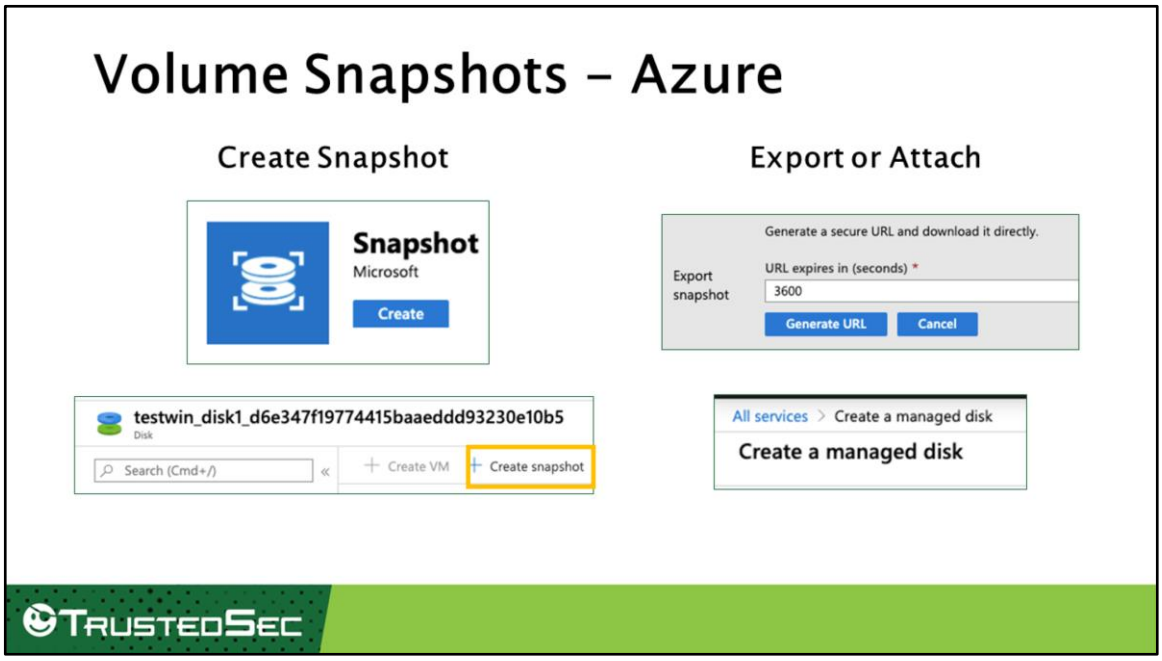
Most cloud providers (including Azure and AWS) provide a way to snapshot a disk. This creates a full copy of the disk that can be shared out to investigators and mounted directly in their forensic workstations! It has little to no impact to the volumes themselves, so high criticality systems can remain available.

Since volume snapshots are so easy to do and provide a point-in-time view of the system, you should do these as soon as you suspect an incident has occurred. In fact, add it to your cloud response runbooks to perform early in any investigation.



To take a volume snapshot in Amazon AWS, go to the volume console page, right-click on the volume and create snapshot. Alternatively, you can create a snapshot using the CLI tools and the *aws create-snapshot* command. When you are ready to mount the snapshot, right click on it and select Create Volume or utilize the *aws create-volume* command. That volume can then be mounted onto another system and imaged there and downloaded if necessary.

https://docs.aws.amazon.com/cli/latest/reference/ec2/create-snapshot.html
https://fireoakstrategies.com/create-and-export-an-aws-ec2-volume-image/
https://docs.aws.amazon.com/cli/latest/reference/ec2/create-volume.html



To create a snapshot of a disk in Azure, to go your Azure portal page, select “Create a resource”, then snapshot. After that, fill out the information for the snapshot name, region, and source disk and then select Create Snapshot. The snapshot should show up in your snapshot resources on the portal page. Alternatively, you can go into an individual disk’s resource page and click Create Snapshot.

You can then download the snapshot as a VHD by clicking on the Export button in the snapshot resource page. This generates an expiring URL directly to the disk.

To attach a snapshot to a VM, you need to search for “Managed disk” in the marketplace and fill out the form to include the details of your snapshot. Once the disk is created, you can attach it as a data disk to any VM.

https://docs.microsoft.com/en-us/azure/virtual-machines/windows/snapshot-copy-managed-disk
https://docs.microsoft.com/en-us/azure/virtual-machines/windows/managed-disks-overview
https://docs.microsoft.com/en-us/azure/virtual-machines/windows/create-vm-specialized-portal

Volatile Information

Live Response Collection

- Collect memory, volatile data as normal
- Take advantage of forensic drive
- KAPE
  - https://www.kroll.com/en/services/cyber-risk/investigate-and-respond/kroll-artifact-parser-extractor-kape
- Margarita Shotgun
  - AWS Linux remote memory collection
  - https://github.com/ThreatResponse/margaritashotgun

TrustedSec

After you've gathered the system metadata and snapshot'd the disks, you need to gather volatile information. Volatile information includes items like memory acquisition, process listings, network connections, and other information that may be lost when the system is turned off. Since this is going to be performed on the suspect systems themselves, you can run whatever tools you normally would to do so.

One generally recommended tool is KAPE, the Kroll Artifact Parser and Extractor, which has become popular in the last year for collecting volatile information and triage files from systems.

If you are using AWS, check out Margarita Shotgun. This tool automates the memory collection on AWS Linux instances.

Within AWS, there are many additional places that data can be obtained and used for IR or forensic purposes.

CloudTrail – CloudTrail is an Amazon service that records events and API activity within your cloud. This allows you to see which user IDs, access keys and IP addresses were making modifications to the cloud environment itself. By default, 90 days of events are kept. You can also export all events to an S3 bucket for more analysis.

CloudWatch – This service records OS, application, and performance metrics for all instances. These can be used to discover spikes in activity which could be related to timeframes of interest in an investigation.

AWS Config – This service that monitors configuration changes and keeps a history of the configurations. While this must be set up prior, if this data exists it can provide some information as to what was changed during an AWS configuration compromise.

S3 Data Storage Access Logs – Provides access logs for S3 buckets within AWS.

Note that some of these need to be configuration before any incident so the logs are

kept. Additionally, some of these services can be configured to alert once specific criteria is met, which could be used to detect anomalies or attacks.

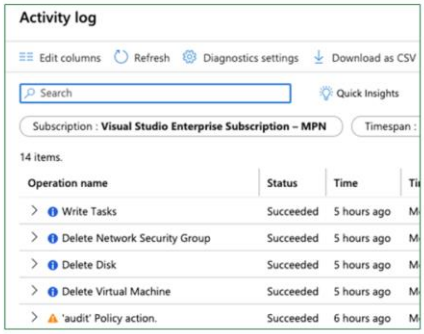
https://docs.aws.amazon.com/AmazonS3/latest/user-guide/server-access-logging.html

Other Data to Grab – Azure

Security Center

DESCRIPTION	COUNT	DETECTED BY
Security incident detected	1	Microsoft
Security incident detected	1	Microsoft
Security incident detected	1	Microsoft
Potential SQL Injection	1	Microsoft
Modified system binary discovered in du...	1	Microsoft
Successful RDP brute force attack	1	Microsoft

Activity Logs



TrustedSec

Azure seems to have one place for security information: Security Center - a goldmine of security information. From here, Microsoft provides continuous monitoring and alerting of the cloud resources, and any attack will likely have something related to it in here. Any that takes place in Azure should immediately utilize Security Center and look at any alerts that have been generated for any assets in the cloud. One section that needs to be highlighted are the Security Alerts.

The Security alerts are correlated detections from all the cloud environment events that are going into Security Center. While the free tier does not include this ability (outside of some minimal detections), there is a free trial to test out the extended analysis and events.

The Azure Activity logs provide information on what has been done within Azure – similar to AWS' CloudTrail. This can be used by investigators to determine if or what modifications an attacker has made to the environment. The Activity Logs can be found by searching for "Activity Log" in the Azure services portal.

https://docs.microsoft.com/en-us/azure/security-center/
https://docs.microsoft.com/en-us/azure/security-center/security-center-alerts-

overview
https://docs.microsoft.com/en-us/azure/azure-monitor/platform/activity-log-view

The good news for those performing incident response is that there are a number of tools available that help automate or analyze the data.

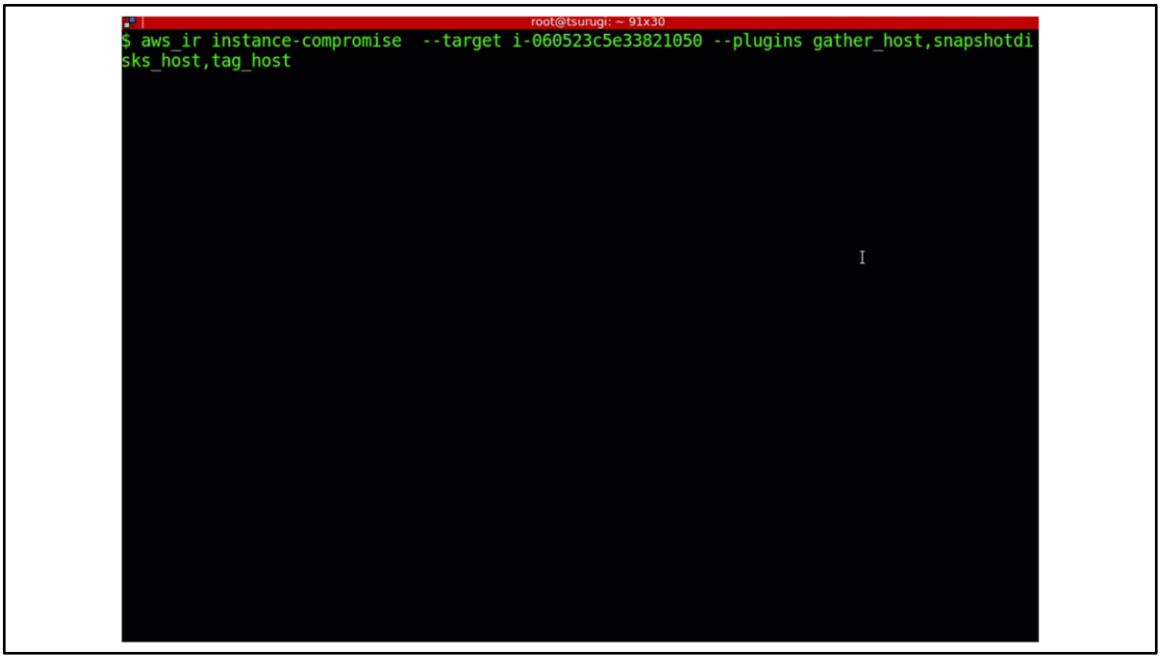
Much of what responders want to do should be scripted when necessary. This allows us the commands to be automated, run consistently, and without error. Fortunately, most cloud services have robust command line interfaces (CLI) that allow scripting. Some tasks in the cloud cannot be performed outside of the CLI. Therefore, it is almost imperative that you have these tools installed if you are going to be performing IR against the cloud.

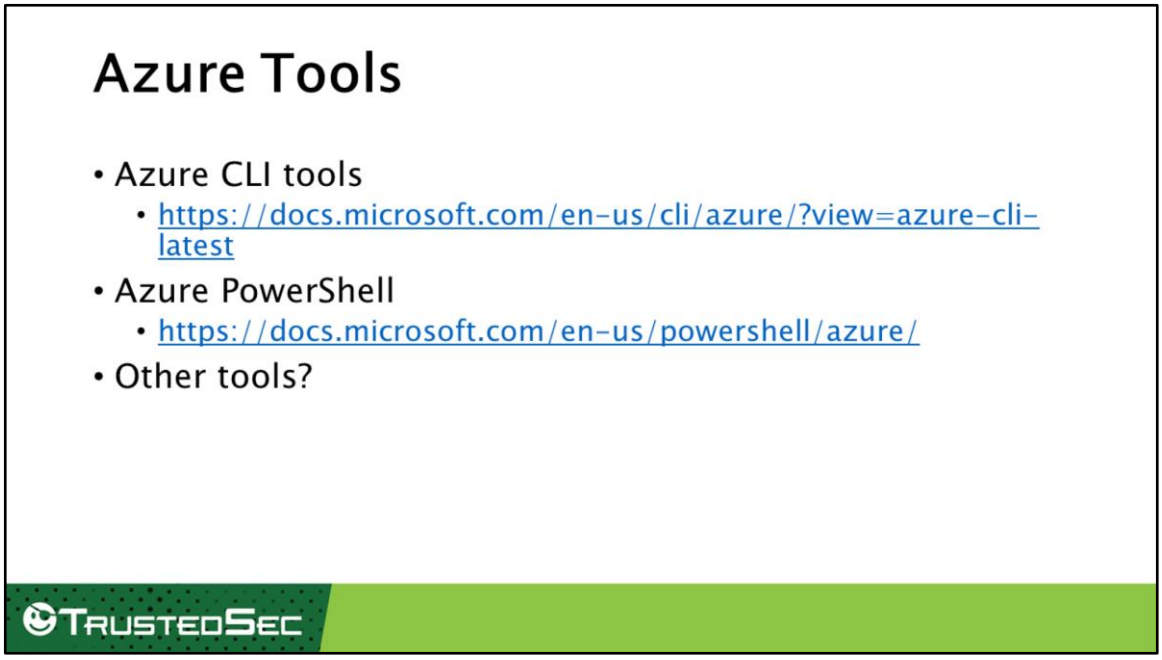
The AWS CLI is located at https://aws.amazon.com/cli/ and is a command line program that performs the tasks within the AWS cloud. Make sure to configure it prior to use (https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-configure.html).

Much of the data that comes back from CLI/APIs (especially AWS) is in JSON format. While you could write your own tool to parse it, sometimes you just need to do it quickly. That's where jq comes in. JQ is a powerful command line JSON processor. There is a little bit of a learning curve, so be sure to check out the tutorial at

https://stedolan.github.io/jq/tutorial/ and some examples at https://shapeshed.com/jq-json/.

We want to make our lives as easy as possible and that's where automation comes in. You can always write your own automation, but for AWS this has already been done with the aws_ir project (https://github.com/ThreatResponse/aws_ir)!


root@tsurugi: ~ 91x30
$ aws_ir instance-compromise --target i-060523c5e33821050 --plugins gather_host,snapshotdisks_host,tag_host



Similar to AWS, there are multiple Azure toolsets available to assist in analysis.

Microsoft has released multiple ways to interface with Azure outside of the web console. First is the Azure command line tools which are available for use on Windows, Mac, and Linux. These provide a command line interface into your Azure environment to be able to obtain most any information. Just make sure to run *az configure* prior to doing anything so your system is set up.

Since this is Microsoft, there is of course a PowerShell interface into Azure. This API is VERY powerful and is great for scripting tasks out.

CLI Tool Warning

Notice something?

```
$ ls .aws
config          credentials
$
$ ls .azure
accessTokens.json       clouds.config           telemetry
az.json                 commands                telemetry.txt
az.sess                 config
azureProfile.json       logs
```

TrustedSec

When I was testing the cloud command line tools out, I noticed something odd. When I ran the tools, I noticed that after I configured them, it would allow me to execute commands without prompting for creds again, regardless of my IP. This is because you are utilizing access keys and tokens to provide access into your cloud environment.

What does this mean? Well, if I were an attacker and compromised your (or your cloud admins) systems, I would immediately go after these files. Then I would have admin level access into your cloud environments.

To note – I have not fully tested this out so there may be additional protections in place, you may be limited by the API, and I do not know how MFA factors into this. Regardless, be careful.

Take Away

- Prepare beforehand
  - Turn on network monitoring
  - Set up forensic workstations and managed drives
  - Create Runbooks
- Data Acquisition
  - Gather metadata
  - Take volume snapshots
  - Gather Volatile Information
  - Other Cloud Logs


Questions?
The information security industry is constantly evolving—keep up by following TrustedSec's active social media, blogs, podcasts, and webinars.
TrustedSec
Follow Us
@TRUSTEDSEC