509.2

Microsoft Azure

© 2022 Pierre Lidome. All rights reserved to Pierre Lidome and/or SANS Institute.

PLEASE READ THE TERMS AND CONDITIONS OF THIS COURSEWARE LICENSE AGREEMENT ("CLA") CAREFULLY BEFORE USING ANY OF THE COURSEWARE ASSOCIATED WITH THE SANS COURSE. THIS IS A LEGAL AND ENFORCEABLE CONTRACT BETWEEN YOU (THE “USER”) AND SANS INSTITUTE FOR THE COURSEWARE. YOU AGREE THAT THIS AGREEMENT IS ENFORCEABLE LIKE ANY WRITTEN NEGOTIATED AGREEMENT SIGNED BY YOU.

With this CLA, SANS Institute hereby grants User a personal, non-exclusive license to use the Courseware subject to the terms of this agreement. Courseware includes all printed materials, including course books and lab workbooks, as well as any digital or other media, virtual machines, and/or data sets distributed by SANS Institute to User for use in the SANS class associated with the Courseware. User agrees that the CLA is the complete and exclusive statement of agreement between SANS Institute and you and that this CLA supersedes any oral or written proposal, agreement or other communication relating to the subject matter of this CLA.

BY ACCEPTING THIS COURSEWARE, USER AGREES TO BE BOUND BY THE TERMS OF THIS CLA. BY ACCEPTING THIS SOFTWARE, USER AGREES THAT ANY BREACH OF THE TERMS OF THIS CLA MAY CAUSE IRREPARABLE HARM AND SIGNIFICANT INJURY TO SANS INSTITUTE, AND THAT SANS INSTITUTE MAY ENFORCE THESE PROVISIONS BY INJUNCTION (WITHOUT THE NECESSITY OF POSTING BOND) SPECIFIC PERFORMANCE, OR OTHER EQUITABLE RELIEF.

If User does not agree, User may return the Courseware to SANS Institute for a full refund, if applicable.

User may not copy, reproduce, re-publish, distribute, display, modify or create derivative works based upon all or any portion of the Courseware, in any medium whether printed, electronic or otherwise, for any purpose, without the express prior written consent of SANS Institute. Additionally, User may not sell, rent, lease, trade, or otherwise transfer the Courseware in any way, shape, or form without the express written consent of SANS Institute.

If any provision of this CLA is declared unenforceable in any jurisdiction, then such provision shall be deemed to be severable from this CLA and shall not affect the remainder thereof. An amendment or addendum to this CLA may accompany this Courseware.

SANS acknowledges that any and all software and/or tools, graphics, images, tables, charts or graphs presented in this Courseware are the sole property of their respective trademark/registered/copyright owners, including:

AirDrop, AirPort, AirPort Time Capsule, Apple, Apple Remote Desktop, Apple TV, App Nap, Back to My Mac, Boot Camp, Cocoa, FaceTime, FileVault, Finder, FireWire, FireWire logo, iCal, iChat, iLife, iMac, iMessage, iPad, iPad Air, iPad Mini, iPhone, iPhoto, iPod, iPod classic, iPod shuffle, iPod nano, iPod touch, iTunes, iTunes logo, iWork, Keychain, Keynote, Mac, Mac Logo, MacBook, MacBook Air, MacBook Pro, Macintosh, Mac OS, Mac Pro, Numbers, OS X, Pages, Passbook, Retina, Safari, Siri, Spaces, Spotlight, There’s an app for that, Time Capsule, Time Machine, Touch ID, Xcode, Xserve, App Store, and iCloud are registered trademarks of Apple Inc.

PMP® and PMBOK® are registered trademarks of PMI.

SOF-ELK® is a registered trademark of Lewes Technology Consulting, LLC. Used with permission.

SIFT® is a registered trademark of Harbingers, LLC. Used with permission.

Governing Law: This Agreement shall be governed by the laws of the State of Maryland, USA.

All reference links are operational in the browser-based delivery of the electronic workbook.

Technet24

FOR509.2 Enterprise Cloud Forensics and Incident Response

Microsoft Azure

© 2022 Pierre Lidome | All Rights Reserved | Version H03_06

This page intentionally left blank.

FOR509.2: Microsoft Azure

Section 2.1: Understanding Azure

Section 2.2: VMs, Networking, and Storage

Section 2.3: Log Sources for IR

Section 2.4: Virtual Machine Logs

Section 2.5: In-Cloud IR

This page intentionally left blank.

© 2022 Pierre Lidome

Technet24

Microsoft Azure Roadmap

- 2.1: Understanding Azure
- 2.2: VMs, Network, and Storage
- 2.3: Log Sources for IR
- 2.4: Virtual Machine Logs
- 2.5: In-Cloud IR

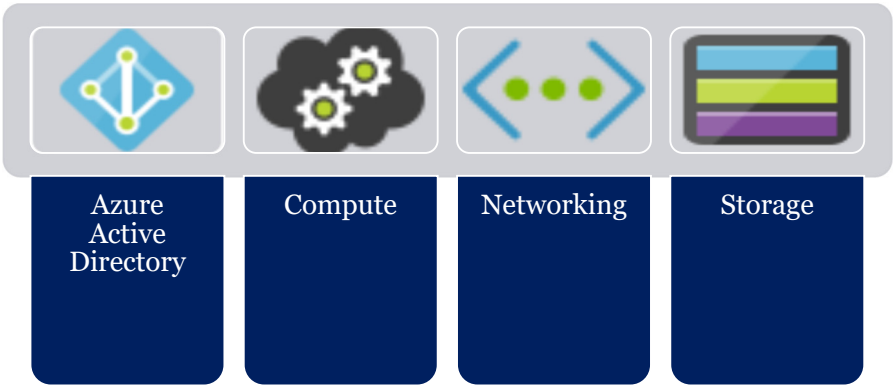
- Microsoft Azure
- Global Footprint
- Azure, Azure AD, Tenant
- Subscription
- Azure Resource Manager
- Resource Groups
- Key Resources
- Resource ID String
- Role Based Access Control
- MITRE ATT&CK for Azure
- Accessing Azure

This page intentionally left blank.

© 2022 Pierre Lidome

Key Azure Services for IR

Key services for incident response and forensics:



Azure is an ecosystem of resources that can be molded to achieve anything you want. It's a large ecosystem, so we will need to focus on the elements you are most likely to encounter during an incident response or forensic investigation. To that purpose, we will investigate where useful logs are stored, how to access them, and how to interpret them.

We will focus on four products: Azure Active Directory, compute, networking, and storage.

Before we can start, we must cover some Azure fundamentals, so everyone has a good foundation. We will discuss concepts such as tenants, subscriptions, Azure Resource Manager, role-based access control, and how to access the Microsoft cloud.

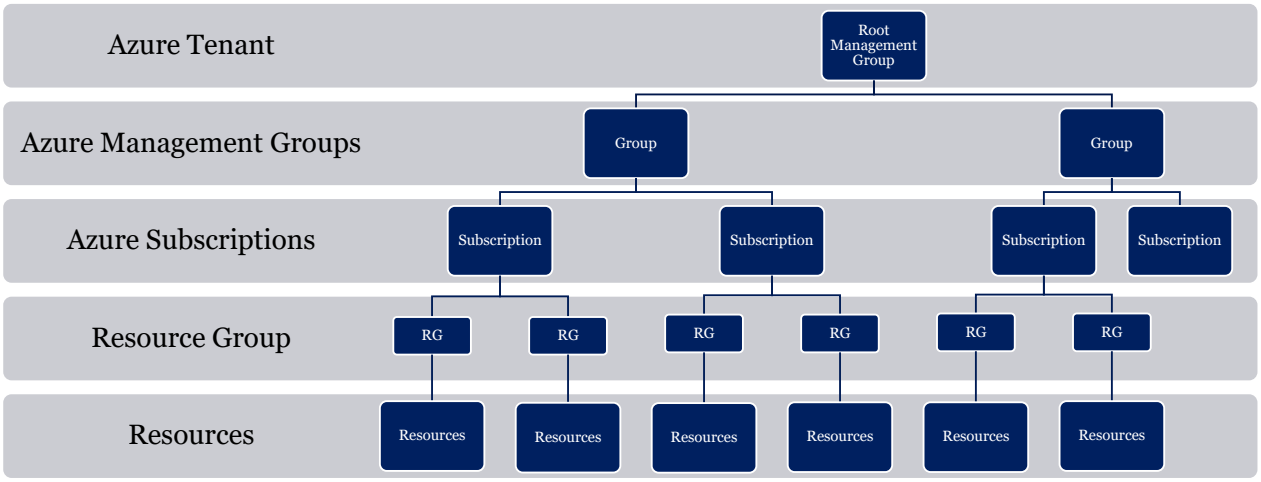
All icons courtesy of Microsoft Azure Cloud and AI Symbol/Icon Set.[1]

1. https://for509.com/azureicons

© 2022 Pierre Lidome

Technet24

Azure Structure



Management Groups are optional

SANS DFIR FOR509 | Enterprise Cloud Forensics and Incident Response 5

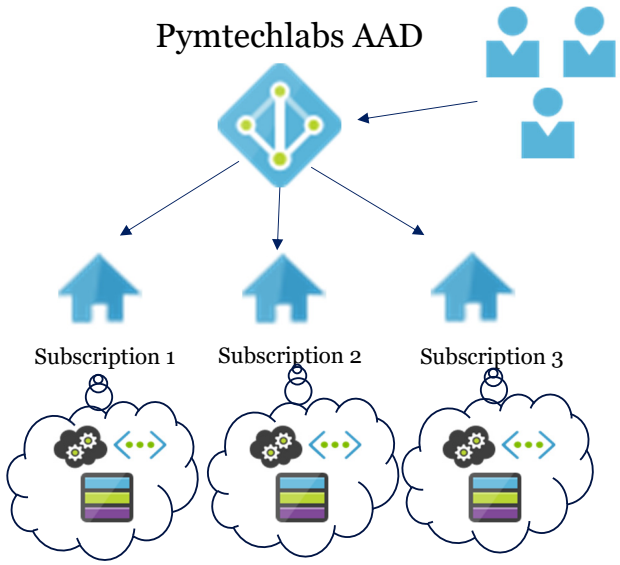
The Azure structure contains five main components:

1. **The Azure Tenant** represents an organization. It's associated with a dedicated Azure Active Directory instance which provides identity and access management.
2. **Management groups**[1] are useful to organize subscriptions. They are popular in enterprise accounts, as they help organize the large number of subscriptions these companies are likely to have. For example, you could create management groups to separate production subscriptions from non-production subscriptions. Another example is creating management groups for each corporate department. The benefit of management groups is the ability to create policies that impact all the subscriptions that belong to that management group at once. It's a lot easier than applying policies to each subscription, one at a time. That being said, management groups are an option, and they are not implemented by every company.
3. **Subscriptions** allows you to organize resources in Azure. Billing is often associated with each subscription (particularly when management groups aren't used). You can think of a subscription as a folder.
4. **Resource groups** are a method to keep related resources organized. You can think of a resource group as a container.
5. **Resources** are very varied in Azure. There are hundreds of possibilities. The most common are virtual machines, networking, and storage.

1. https://for509.com/management-groups

© 2022 Pierre Lidome

Subscriptions



- Tenants can have multiple Azure subscriptions
- They all share a single Azure Active Directory
- Users permissions are defined in AAD
- Subscriptions are a resource boundary
- Subscriptions have resource limitations, and higher CPU counts have to be requested

Now that we have our tenant setup, we need to define one or more subscriptions. As stated by Microsoft, "a subscription is an agreement with Microsoft to use one or more Microsoft cloud platforms or services, for which charges accrue based on either a per-user license fee or on cloud-based resource consumption."[1]

Companies will generally set up numerous subscriptions in order to keep track of charges incurred by different projects. This is a very important concept, as you will need permissions for each subscription that holds the resource(s) you are investigating.

Subscriptions have resource limitations, and higher limits need to be requested when needed. The first limitation you are likely to experience is a limit on the number of virtual CPUs (vCPUs). A higher limit can be requested via a ticket to the Microsoft helpdesk.

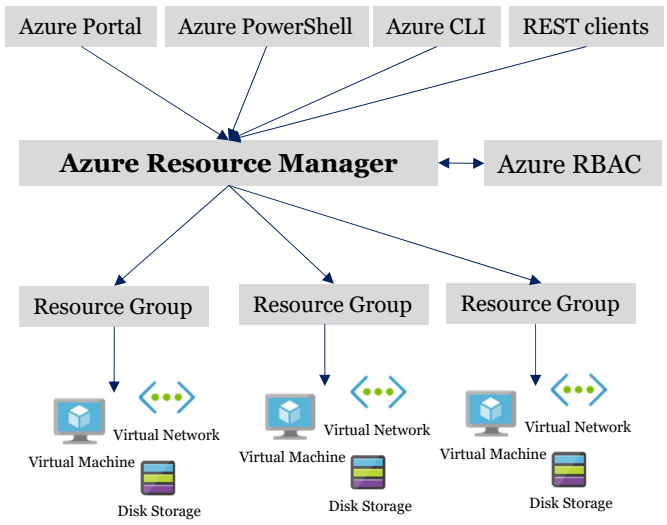
We will now discuss how resources are created and permissions to these resources managed.

1. https://for509.com/subscriptions

© 2022 Pierre Lidome

Technet24

Azure Resource Manager



- Underlying service for deploying and managing resources in Azure
- Consistent management layer: interfaces with Azure Portal, Azure PowerShell, Azure CLI, and REST clients
- Resource Manager template (**infrastructure as code**) Very similar to AWS Cloud Formation

Before we can discuss virtual networks, virtual machines, and storage, we need to understand how these are deployed and managed. The answer is the Azure Resource Manager, which provides a management layer that enables the creation, updating, and deletion of resources.[1]

The advantage of the Azure Resource Manager is that it can take instructions from many different interfaces: Azure Portal, Azure PowerShell, Azure CLI, and REST clients. Yet, the result will be the same irrespective of your choice of interface.

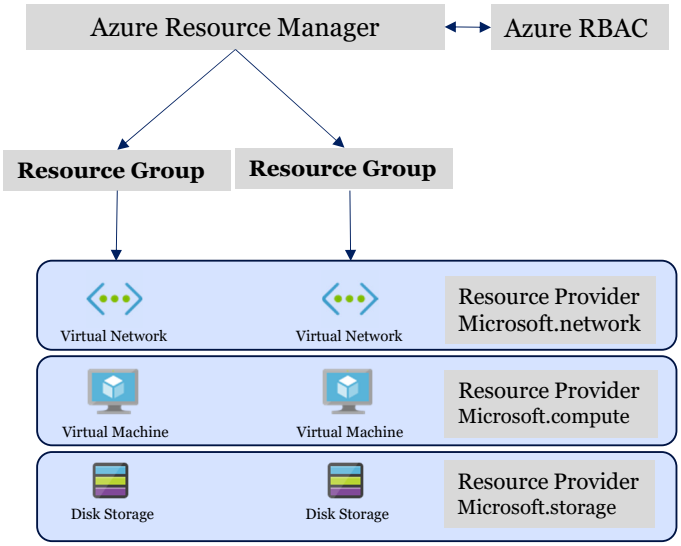
Azure Resource Manager also supports templates (in JSON format) that enable you to deploy resources consistently and repeatedly. This is sometimes referred to as "infrastructure as code." This is very similar to AWS CloudFormation.

Azure Resource Manager applies access controls based on Azure role-based access control (RBAC), which we will discuss shortly.

1. https://for509.com/resourcemanager

© 2022 Pierre Lidome

Resource Groups



- **Resource Groups**: Container that holds related resources
- **Resource Providers**: Service that supplies the resources you can deploy and manage through resource manager
  - Example: Microsoft.compute = VM resources
  - Microsoft.storage = Storage account resources
- **Resources**: VM, networking, storage accounts, etc.

Resources (virtual machines, networking, storage accounts, databases, etc.) should be grouped inside resource groups based on their purpose or commonality. In other words, a resource group is a container that holds related resources. You may have as many resource groups as you wish.

One of the main advantages of resource groups is that you can deploy, update, or delete all resources contained in a resource group at once. You can also set role-based access for each resource group.

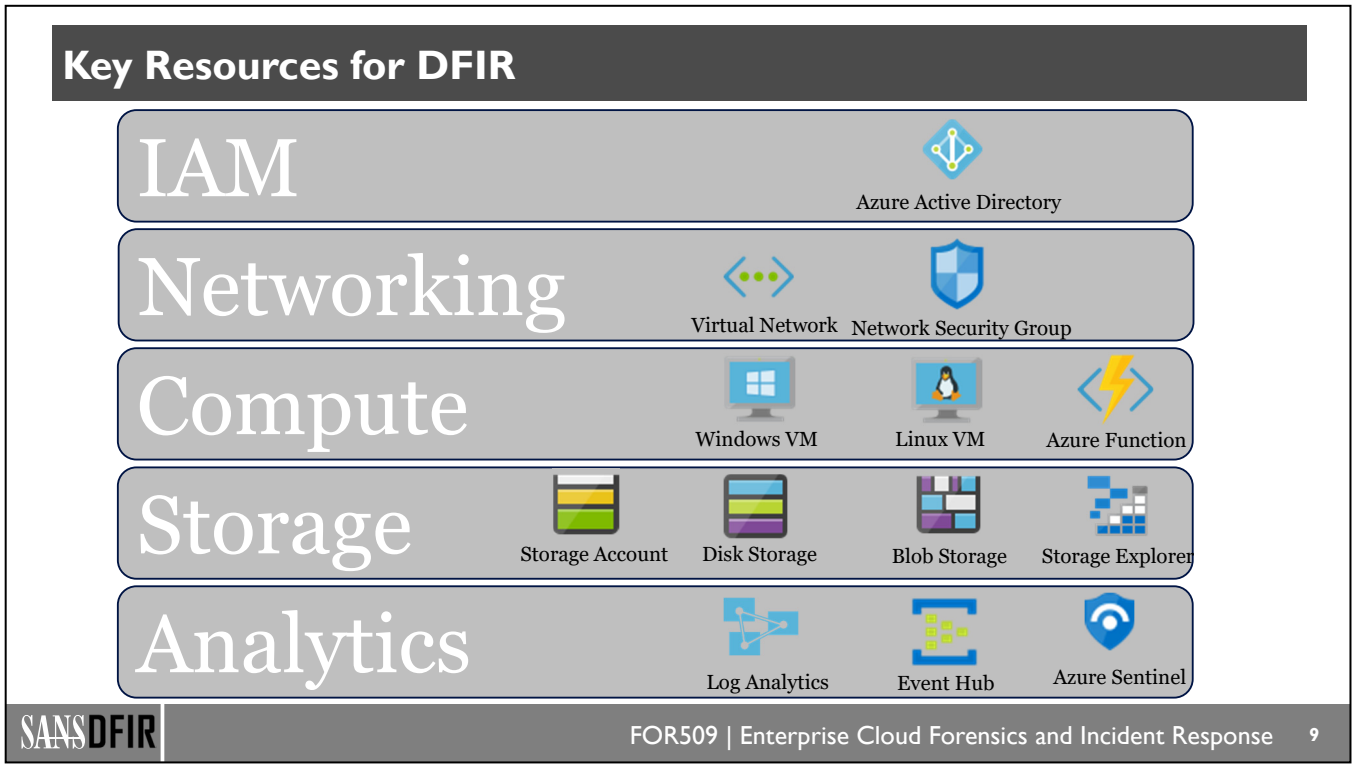
Investigative hint: Ideally you will be granted permissions at the subscription level. However, you could encounter a situation where you are only granted permission to a specific resource group. Be aware of this limitation, as you will have a very narrow view of the infrastructure being used. This is not a good situation, and you should attempt to get higher-level permissions to have a complete view for your investigation.

A resource provider is a service that supplies the resources you can deploy and manage through the Resource Manager. For example, if you want to deploy a virtual machine, the Microsoft.compute resource provider will be invoked. Similarly, a storage account will invoke the Microsoft.storage resource provider.

The two items you will interact with the most are resource groups and resources. It's important to know about Azure Resource Manager and resource providers to have a complete picture, but they perform their jobs in the background.

© 2022 Pierre Lidome

Technet24



Microsoft Azure offers hundreds of products.[1]

For the purposes of incident response and forensics, we will focus on just a few products:

- Identity and Access Management
  - Azure Active Directory: Identity management
- Networking
  - Virtual Networks: Provisioned private networks
  - Network Security Group: Filters network traffic to and from Azure resources
- Compute
  - Virtual Machines: Support for Windows and Linux virtual machines
  - Azure Functions: Serverless solution to implement compute-on-demand
- Storage
  - Disk Storage: Persistent storage from virtual machines
  - Blob Storage: REST-based object storage for unstructured data
  - Storage Account: Container for disk and blob storage. Storage account also contain queues, tables, and file shares.
  - Storage Explorer: View and interact with Azure storage resources. Storage Explorer is a graphical frontend for AzCopy, which we will discuss in more detail in the storage section later in the class.
- Analytics
  - Log Analytics: Collect, search, and visualize logs
  - Event Hubs: Real-time data ingestion service
  - Azure Sentinel: Cloud-native SIEM and intelligent security analytics

1. https://for509.com/azureproducts

© 2022 Pierre Lidome

Azure Resource ID Definitions

- Subscription ID
  "d841fb8e-c0c7-46fd-ad91-3689e704d1fd"
- Resource Group
  "Research"
- Provider
  "Microsoft.Compute"
- Virtual Machine
  "MiningVM"

These resource IDs are combined to create resource ID strings.

Now that we understand the hierarchy that defines an Azure resource, let's look at the internal notation.

- Subscription ID: Globally Unique Identifier (GUID) that belongs to your tenant
- Resource Group: User-generated name
- Provider: Name of the service that supplies the resources you can deploy and manage through Resource Manager
- Resource: Specified by its type and the name given by the user

In the next slide, we will see how Azure puts everything together in a single URI.

© 2022 Pierre Lidome

Technet24

Azure Resource ID Strings Examples

- Resource ID string for the VM
/subscriptions/d841fb8e-c0c7-46fd-ad91-3689e704d1fd/resourceGroups/Research/providers/Microsoft.Compute/virtualMachines/MiningVM

- Resource ID string for the OS disk
/subscriptions/d841fb8e-c0c7-46fd-ad91-3689e704d1fd/resourceGroups/Research/providers/Microsoft.Compute/disks/MiningVM_disk1_213aa18e15cb44a68812d435fff3c508

- Resource ID string for the network interface
/subscriptions/d841fb8e-c0c7-46fd-ad91-3689e704d1fd/resourceGroups/Research/providers/Microsoft.Network/networkInterfaces/miningvm106

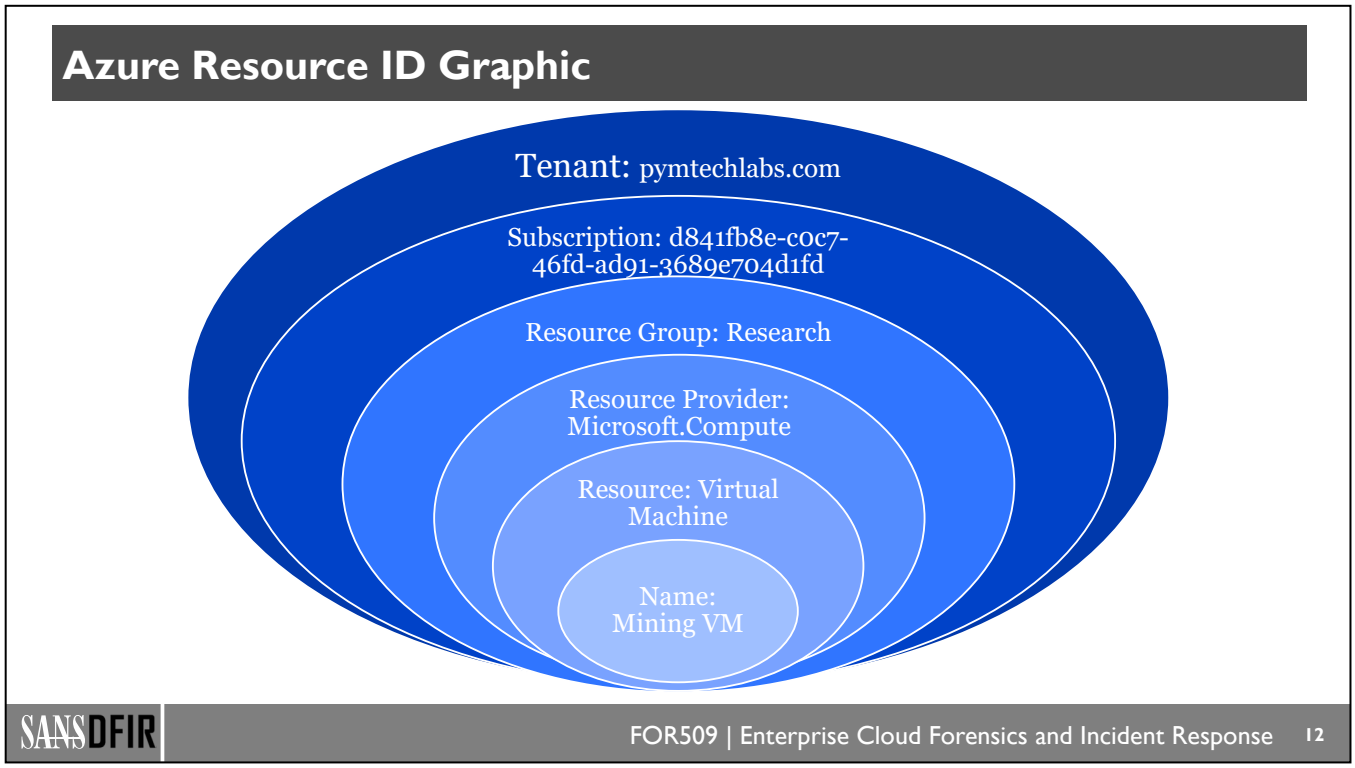
Every item in Azure has an associated Universal Resource Identifier (URI) that follows the format:

/subscription/<SubscriptionId>/resourceGroups/<resourcegroupname>/providers/<providername>/<resourceType>/<resourcename>

In the MiningVM example, we have a resource ID string for the VM itself, one for the OS disk, and one for the network interface. If we were to assign a public IP address to that VM, we would also have a resource ID string for that IP address.

This notation is important if you want to access Azure resources via the CLI or PowerShell.

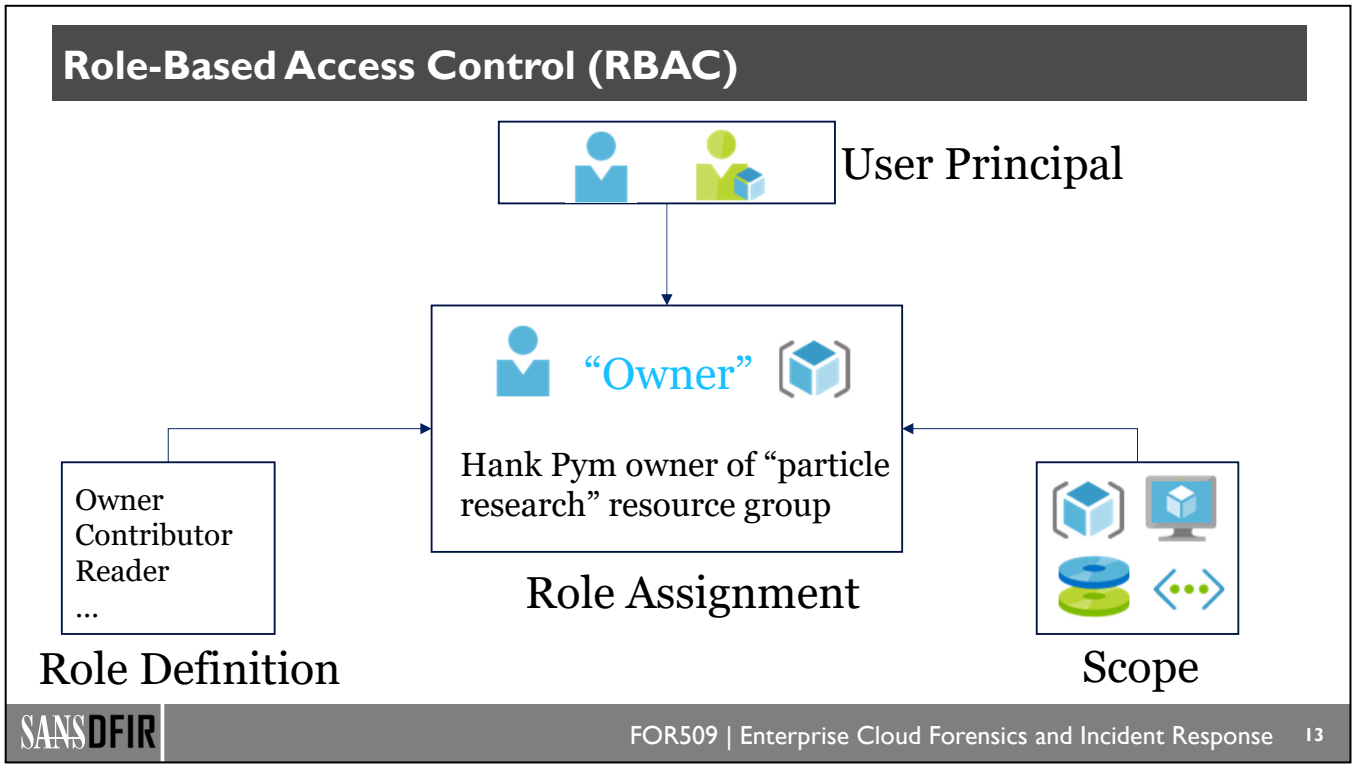
© 2022 Pierre Lidome



In this slide, you can see a graphical representation of the Azure Universal Resource Identifier (URI) for our virtual machine called “Mining VM.”

© 2022 Pierre Lidome

Technet24



Azure role-based access control (RBAC) lets you manage who has access to what resource and what they can do with that resource. Azure RBAC is an authorization system built on Azure Resource Manager.

To control access to resources, you create role assignments. There are three elements to a role assignment:

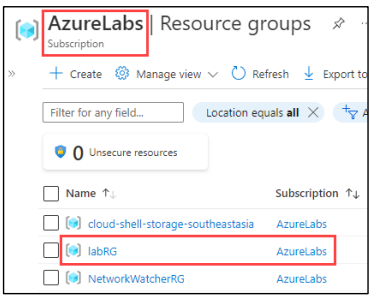
1. **Security Principal**
    - An object representing an entity such as a user or group, which can access the resource
2. **Role Definition**
    - A collection of permissions such as read, write, and delete
    - Azure has several built-in roles.[1] The most common are `Contributor`, `Owner`, and `Reader`
    - When performing an investigation, it's preferable to be granted `Owner` permission for maximum flexibility
3. **Scope**
    - Specify which role can access a resource or resource group
    - Scopes can be specified at four levels: management group, subscription, resource group, resource

There are many nuances to Azure RBAC, and this only covers a high-level overview that you are likely to encounter in your investigations. The Microsoft documentation should be referenced for an in-depth explanation of Azure RBAC.[2]

1. https://for509.com/azureroles
2. https://for509.com/azurerbac

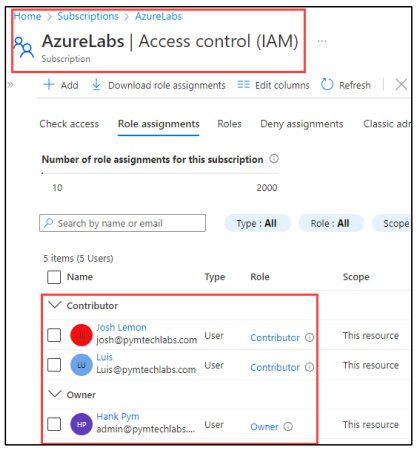
© 2022 Pierre Lidome

User Access Role Assignments

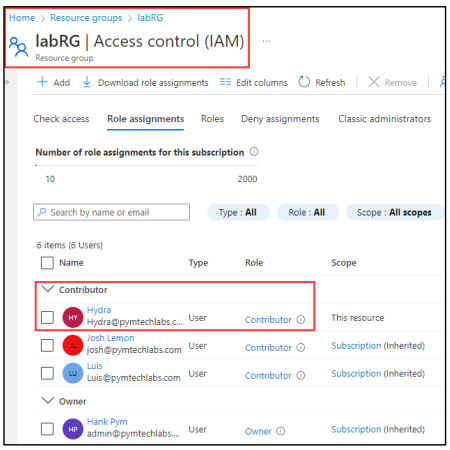


Subscription: AzureLabs
Resource Group: labRG

Hydra doesn't have access at the subscription level



Hydra is a contributor in the labRG resource group



How do we find role assignments for specific users?

Administrators may want to enumerate a user's access within a subscription. This would be simple if the user is granted access at the subscription level, but much more complicated if the access is granted at the resource group level. Further, a user could have limited access (or none at all) at a subscription level, yet have owner permission at a resource group level.

In the example shown on this slide, hydra has no access to the AzureLabs subscription, but has contributor access to the labRG resource group.

Going through every subscription and every resource group one-by-one would be very tedious. In the next slide, we will show you the `az` command to enumerate role assignments for a specific user.

© 2022 Pierre Lidome

Technet24

List User Role Assignments: hydra@pymtechlabs.com

1. Set the subscription

```
> az account set --subscription "AzureLabs"
```

2. Search for all role assignments

```
> az role assignment list --all --assignee hydra@pymtechlabs.com
```

3. Result

```
"principalId": "ddaca639-d9c8-4d65-9f30-c0dcc58f9574"
"principalName": "Hydra@pymtechlabs.com"
"principalType": "User"
"resourceGroup": "labRG"
"roleDefinition": "Contributor"

<other entries removed>
```


Enumerating role assignments can be done at the CLI level either with the `az` command or the `Get-AzRoleAssignment` PowerShell command.

The first step is to select a subscription. The second step is to issue the "`az role assignment list`" command with two parameters:

- `--all` to search recursively through all resource groups
- `--assignee` to specify the user

This process will need to be repeated for each subscription.

The result will show the role definition(s) assigned to the specified user for the subscription as well as each resource group under that subscription.

In our example, hydra@pymtechlabs.com is only assigned a role in the labRG resource group so we only have one result.

© 2022 Pierre Lidome

MITRE ATT&CK® and Azure

- Top attacks:
  - Obtaining credentials
  - Validating credentials via legacy protocols
  - Leveraging application tokens (OAuth)
  - Data acquisition (OneDrive, SharePoint, Exchange)
  - Data exfiltration (storage accounts, Graph API)
  - Cloud resource abuse (VMs for crypto)

- Resources
  - MITRE ATT&CK® Matrices: Office 365, Azure AD, SaaS, IaaS
  - Microsoft mapping of Azure security controls to MITRE ATT&CK®
  - Links in the notes

MITRE ATT&CK® matrices[1] detail the tactics and techniques used by threat actors. MITRE has a number of relevant matrices for Azure: Office 365, Azure AD, SaaS, IaaS.

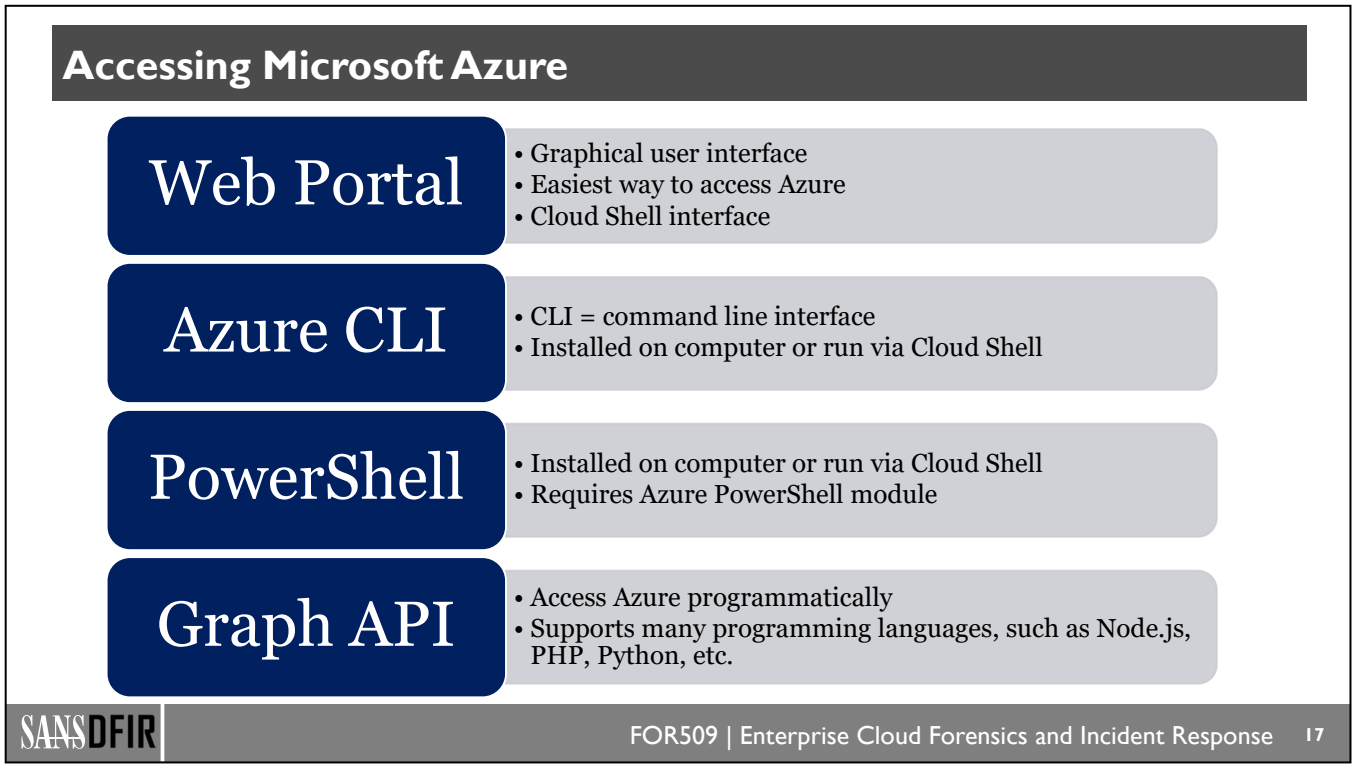
In addition, Microsoft mapped its built-in Azure security controls[2] against the attacks shown in the MITRE ATT&CK® matrices. This provides 48 controls that can be leveraged to protect your environment.[3]

From a practical point of view, threat actors focus on:
- Obtaining and verifying credentials. They frequently use legacy protocols (such as IMAP) to verify the validity of credentials.
  - These attacks can be mitigated by disabling legacy protocols and enforcing multifactor authentication.
- Obtaining and leveraging application tokens, as shown in Section 1 in the Graph API section.
- Acquiring data from OneDrive, SharePoint, and Exchange.
- Exfiltrating data via storage accounts or the Graph API.
- Abusing cloud resource (cryptocurrency mining, for example).

1. https://for509.com/mitre-cloud
2. https://for509.com/azure-attck-map
3. https://for509.com/azure-controls

© 2022 Pierre Lidome

Technet24



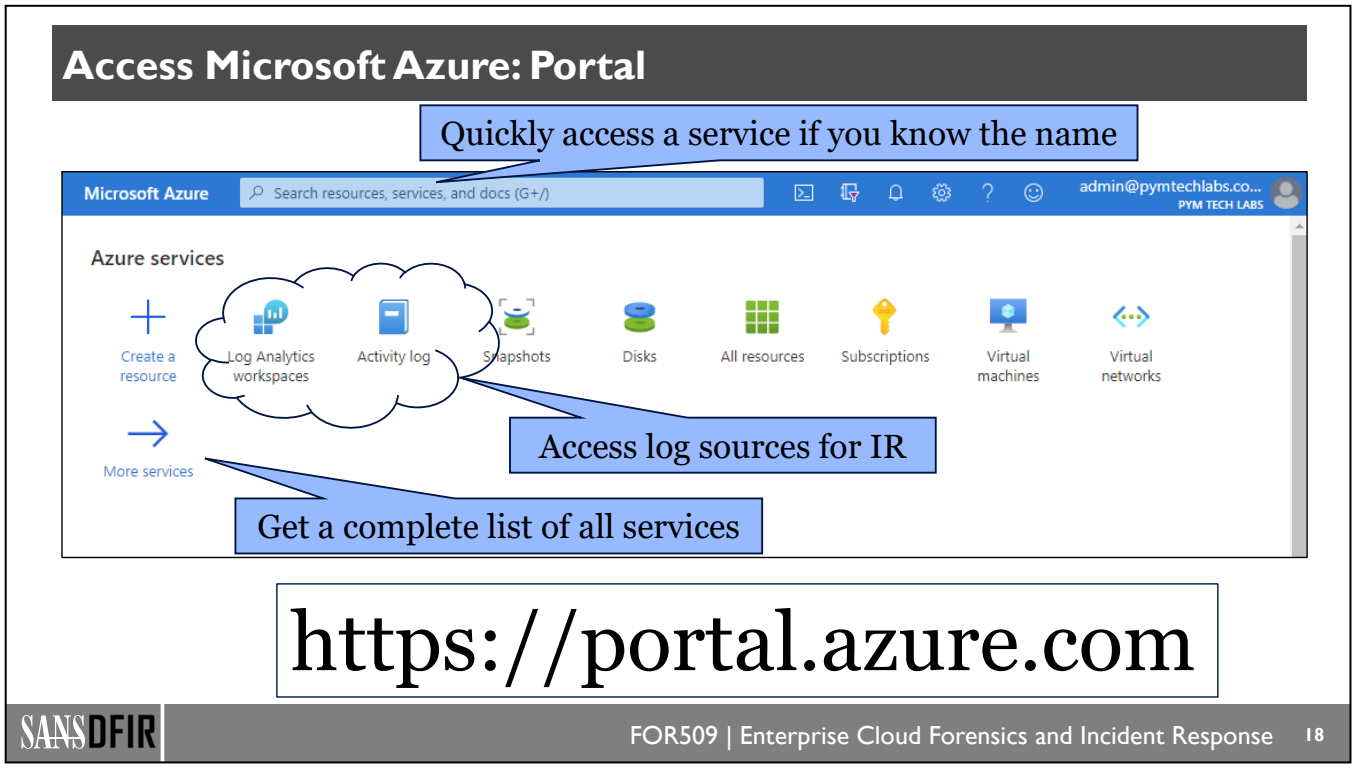
Before we move on to the next section and discuss the various log sources in Azure, we need to quickly review the four ways to access Azure:

1. Azure portal
2. Azure CLI
3. PowerShell
4. Graph API

Azure CLI and PowerShell can be installed on your computer or used via Cloud Shell. Cloud Shell is a command line interface available in the Azure portal.

The Microsoft Graph API is a RESTful web API that enables you to access Azure via your favorite programming language. It supports many different languages, including Node.js, PHP, Python, etc.

© 2022 Pierre Lidome



The Azure portal is a graphical user interface and is the most common and easiest way to access Microsoft Azure.[1]

For the purposes of this class, the most important services will be `Activity log` and `Log Analytics workspaces`. Other important services are `Resource groups`, `Virtual machines`, `Virtual networks`, `Disks`, `Snapshots`, and `Subscriptions`.

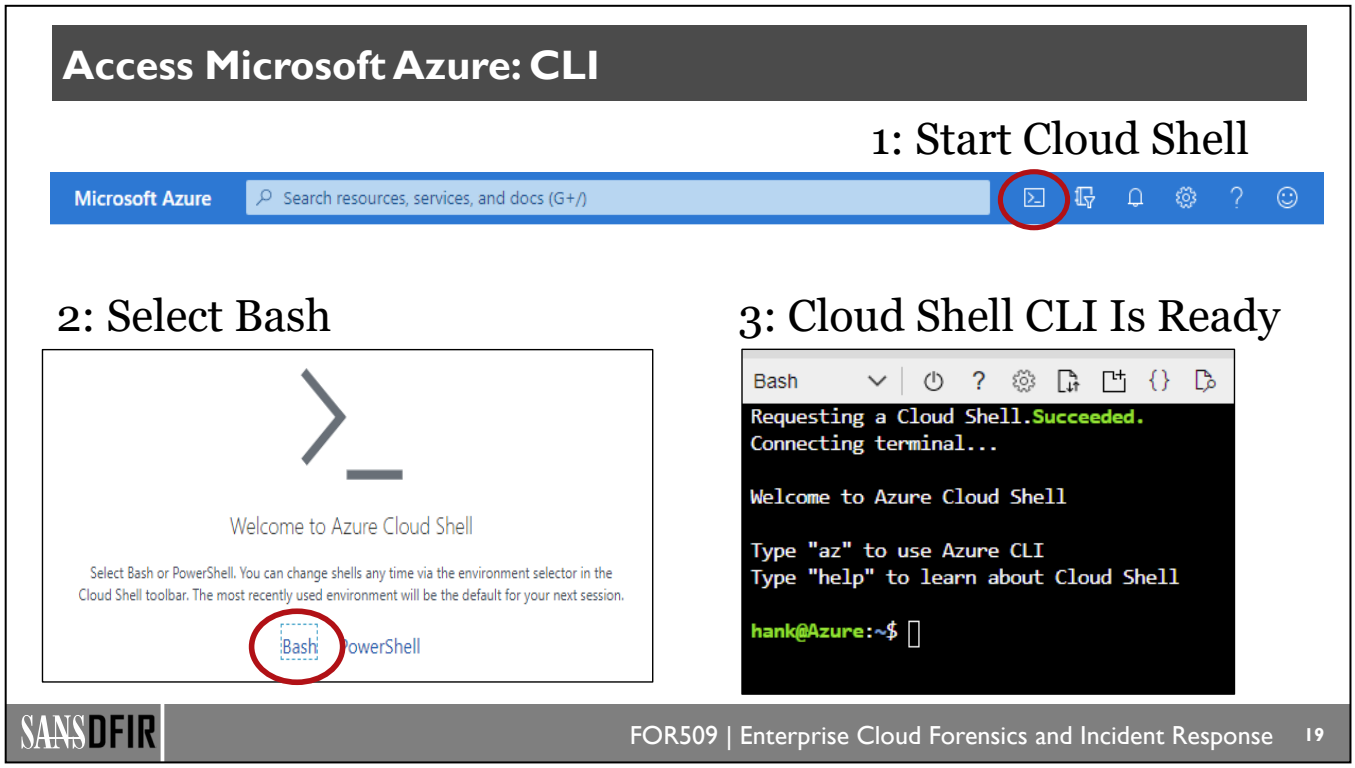
The service **`All resources`** is also very important, as it gives you a quick overall view of every resource provisioned in your subscription.

The top row will only show you the icons for the most recent services used. If you don't see what you are looking for, you can type it in the search box or select "More services" to get a complete list.

1. https://for509.com/portaloverview

© 2022 Pierre Lidome

Technet24



The Azure command line interface (CLI) is a set of commands used to create and manage Azure resources.[1] It can be installed on your computer[2] or run via Cloud Shell.[3] This slide shows Azure CLI running from Cloud Shell.

Microsoft provides the Azure CLI for Windows, macOS, and Linux. If you would rather not install software on your computer, you can use the same CLI commands in the Azure Cloud Shell Bash environment. The Azure Cloud Shell is an interactive shell for managing Azure resources. It provides a terminal window inside your browser. The terminal window is based on the Linux bash shell, so in addition to the Azure CLI commands, you also have access to a wealth of Linux commands.

When selecting the Cloud Shell in the Azure Portal, you will select the Bash environment if you wish to use CLI commands.[4] Another option is PowerShell, which we will discuss in the next slide.

Whether you choose to run the Azure CLI on your own computer or via the Cloud Shell, you now have access to the set of "az commands."

© 2022 Pierre Lidome

The first command you will need to issue is to authenticate yourself (not needed if you are using Cloud Shell, since you already authenticated to the Azure portal):

```
az logon
```

You can then list the subscription you have access to:

```
az account list
```

You now need to select the subscription that's appropriate for your investigation:

```
az account set –subscription "name of subscription"
```

1. https://for509.com/cli-intro
2. https://for509.com/cli-install
3. https://for509.com/cloudshell-intro
4. https://for509.com/bash

© 2022 Pierre Lidome

Technet24

DFIR Evidence in Cloud Shell

- When **Bash** Cloud Shell is used, a container is created in a storage account reserved for that purpose
- Evidence of issued commands can be found in the .bash_history file
- The .wget-hsts file can also contain valuable information
- Under the .azure directory, there are various log files, but the DFIR value seems to be limited

- When **PowerShell** Cloud Shell is used, there is no container created and, as such, no DFIR artifacts appear to be recorded
- However, the underlying actions will still be logged to the corresponding logs (audit log, sign-in log, resource log, etc.)

If a threat actor was to obtain the necessary cloud credentials and use Cloud Shell, would we be able to tell what they did?

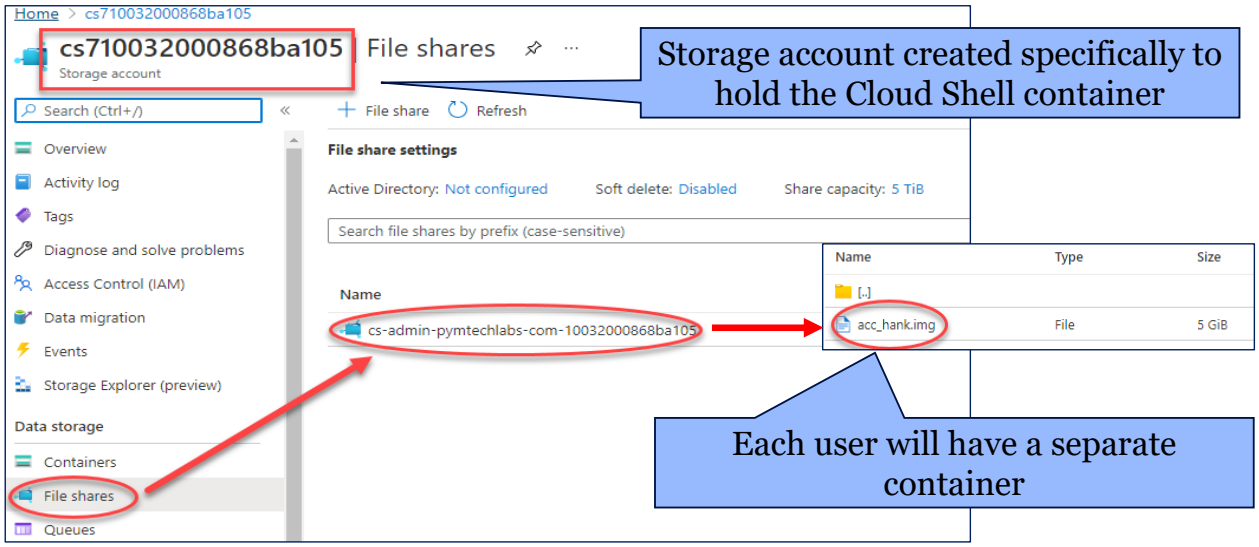
If they choose the Bash Cloud Shell, a storage account will be created to store the Linux image in a container. By downloading that container, we can perform traditional Linux forensics, which will lead us to the .bash_history file. While there is a risk that the threat actor deletes that file, it's worth looking for it. Other files such as .wget-hsts may be present if the threat actor used wget to retrieve files from the internet.

Unfortunately, if the threat actor uses the PowerShell Cloud Shell, there doesn't appear to be any DFIR artifacts.

© 2022 Pierre Lidome

Cloud Shell Storage Account

When Bash is selected, Azure requires a storage account to store the image



The storage account name will vary but should start with the letters “cs”. Under that storage account, select “File shares.” You will find a file share with a long name where the container is located. The container is a Linux image that’s named based on the user account that accessed Cloud Shell.

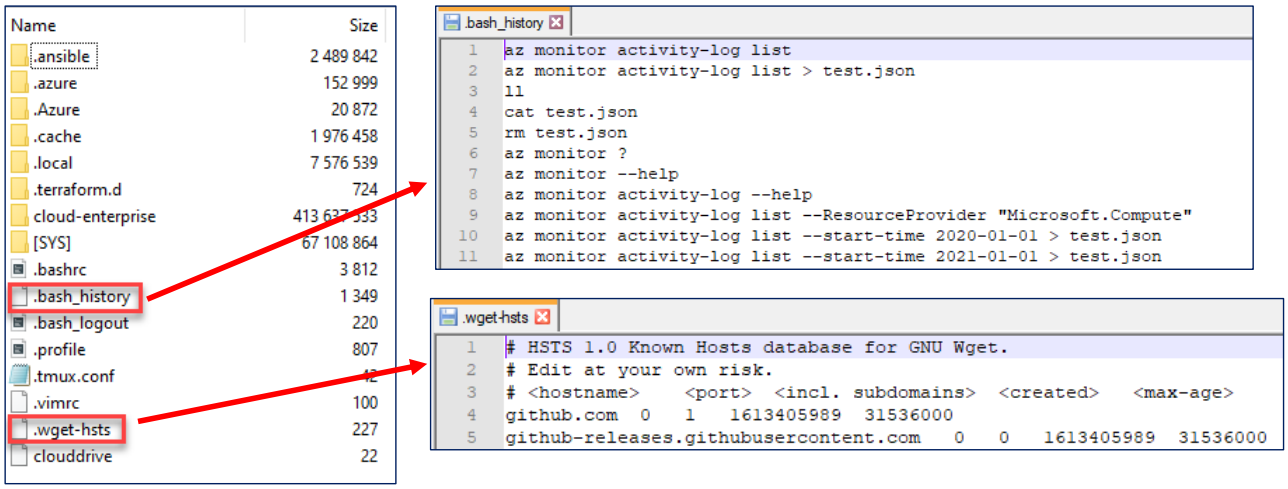
You will need to download that container in order to look for interesting artifacts. Later in the class we will discuss an application called Azure Storage Explorer, which is very helpful to download data from storage accounts.

© 2022 Pierre Lidome

Technet24

acc_hank.img

Inside the container we have .bash_history with the commands issued by the user.

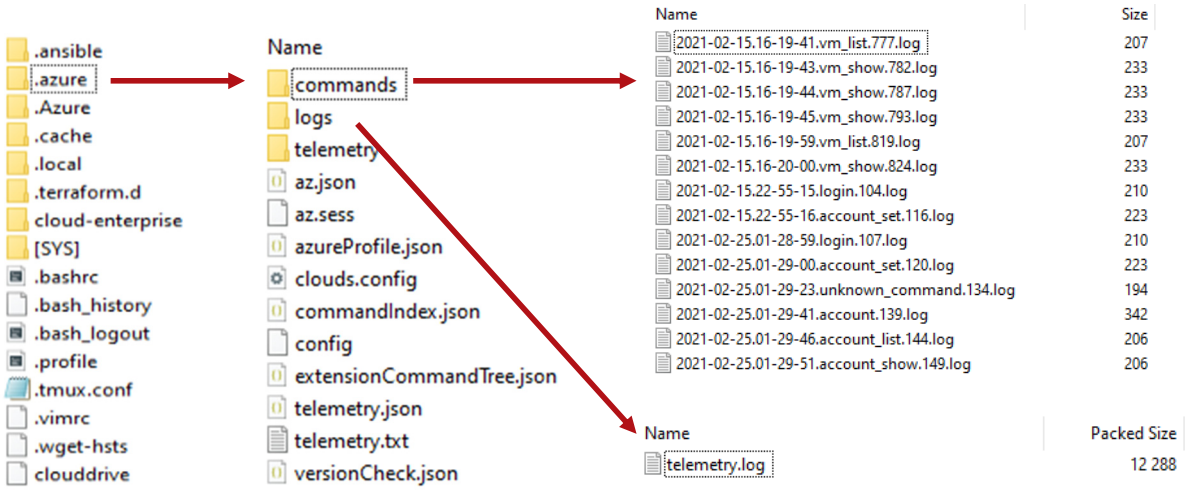


You can use 7-Zip to open the .img file. Inside you will find a number of files that investigators who are familiar with Linux will recognize. The `.bash_history` file will store the commands issued by the threat actor. If the threat actor used the `wget` command, there may be interesting information in the `.wget-hsts` file. Other Linux-type artifacts may also exist, depending on the actions taken by the threat actor.

© 2022 Pierre Lidome

Other Logs

Names look interesting but have limited forensic value



In the .azure directory, there are a number of subdirectories that look interesting. They may have some limited forensic value depending on your situation.

© 2022 Pierre Lidome

Technet24

Access Microsoft Azure: PowerShell

PowerShell on your computer

Step 1: Start a PowerShell terminal with Administrator permission

Step 2: Install Az module (if not already installed)

```
PS> Install-Module -Name Az -AllowClobber
```

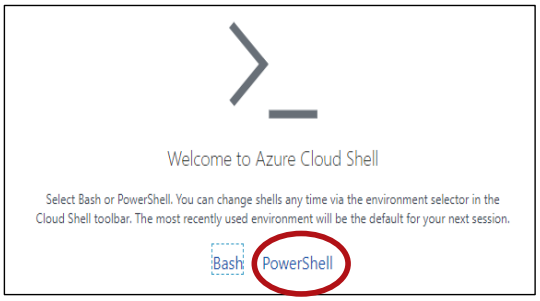
Step 3: Verify the module was installed and check the version (optional)

```
PS> Import-Module Az; Get-Module Az
```

Step 4: Connect to Azure

```
PS> Connect-AzAccount
```

PowerShell via Cloud Shell



Another way to interface with Azure is through PowerShell. Just like Azure CLI, you have the option to run PowerShell from your computer or run via Cloud Shell.[1]

To use PowerShell on your computer, you will need to install the Azure PowerShell module. Be sure to install the Az module and not the older AzureRM module, which is now deprecated.

Step 1: Start a PowerShell terminal with Administrator permission

Step 2: Install the Az module (if not already installed)
```
Install-Module -Name AZ -AllowClobber
```

Step 3: Verify the module was installed and check the version (optional)
```
Import-Module Az; Get-Module Az
```

Step 4: Connect to Azure
```
Connect-AzAccount
```

1. https://for509.com/powershell

© 2022 Pierre Lidome

FOR509.2: Microsoft Azure

Section 2.1: Understanding Azure

Section 2.2: VMs, Networking, and Storage

Section 2.3: Log Sources for IR

Section 2.4: Virtual Machine Logs

Section 2.5: In-Cloud IR

This page intentionally left blank.

© 2022 Pierre Lidome

Technet24

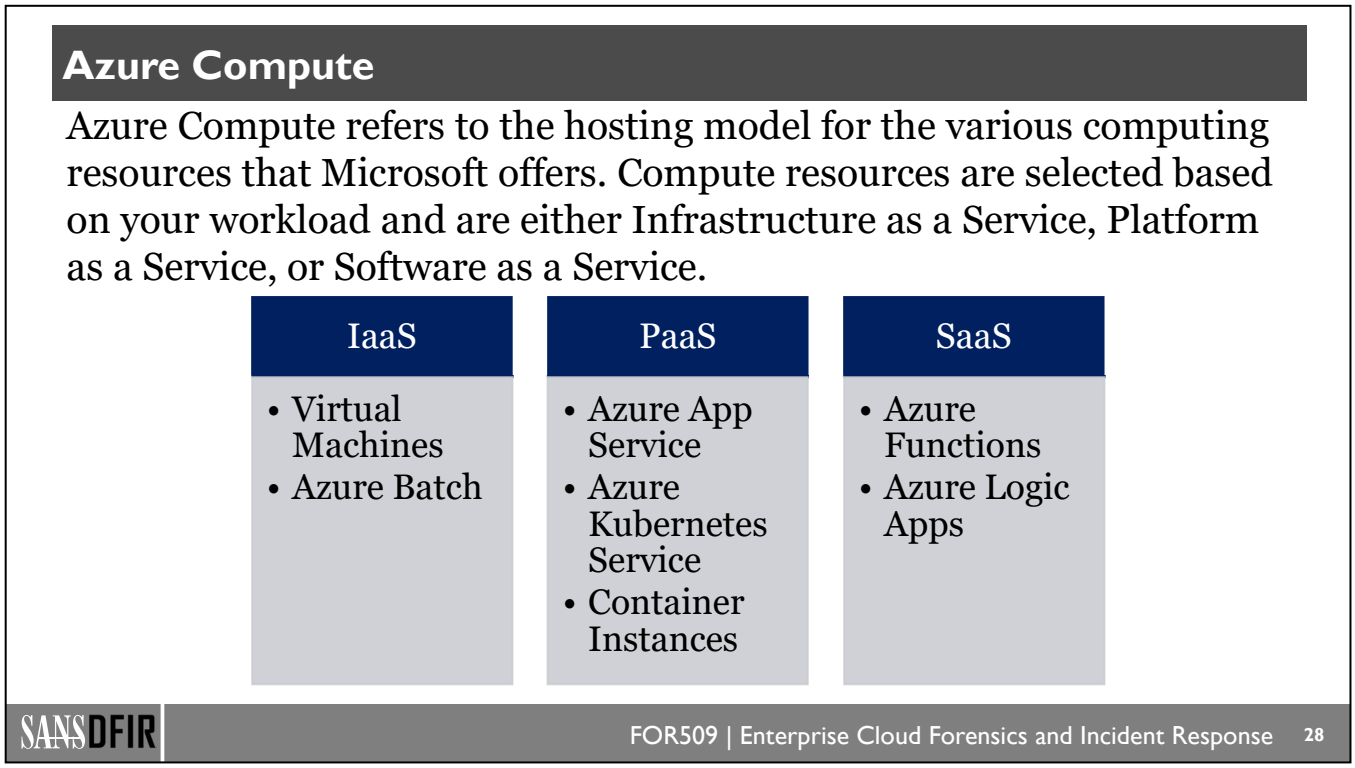
Microsoft Azure Roadmap

- 2.1: Understanding Azure
- 2.2: VMs, Network, and Storage
- 2.3: Log Sources for IR
- 2.4: Virtual Machine Logs
- 2.5: In-Cloud IR

- Azure Compute
- Virtual Machine Types
- Case Study: Crypto Mining VM
- Azure Virtual Networks
- Network Security Groups
- Virtual Appliances
- Storage
- **Lab 2.1**: Using SOF-ELK with Azure Logs

This page intentionally left blank.

© 2022 Pierre Lidome



The most visible resource of any cloud computing is the virtual machine (VM). However, there are many other kinds of resources available in Azure. Microsoft calls this category Azure Compute and offers several options based on your workload and application.[1]

Azure Compute resources are categorized based on their service model.

Infrastructure as a Service	**Description**
Virtual Machine	Typical server where you are responsible for everything
Azure Batch	Enables large-scale parallel and high-performance batch jobs with the ability to scale to tens, hundreds, or thousands of VMs

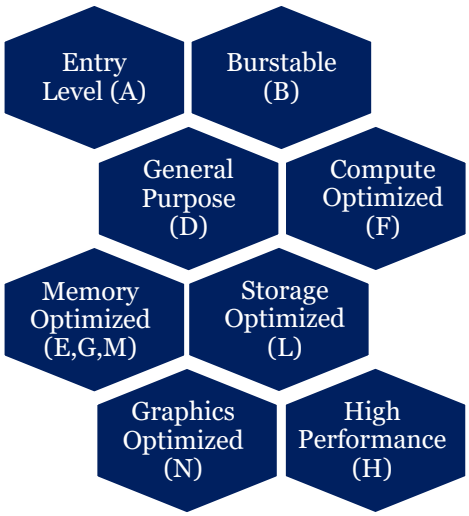
Platform as a Service	**Description**
Azure App Service	Managed service to host web apps, mobile app backends, RESTful APIs
Azure Kubernetes Service	Managed Kubernetes service for running containerized applications
Container Instances	Simple way to run a container in Azure without provisioning a VM

Software as a Service	**Description**
Azure Functions	Code executed in response to an event without concerns for the underlying platform or infrastructure
Azure Logic Apps	Logic apps are similar to functions for execute workflows instead of code

1. https://for509.com/azurecompute

© 2022 Pierre Lidome
Technet24

Virtual Machine Types



- Many VM types to meet workload and application requirements
- Some are only available in certain regions

Log features are the same for all VM types

Since you will spend most of your time analyzing VMs, let's examine the different classes of VMs offered by Azure. Azure offers a wide variety of different types of virtual machines. Some are only available in certain regions, and the virtual machines pricing web pages will show which ones are available in which region: Windows[1] and Linux.[2]

Series A: Entry level
- Series A VMs are entry-level machines suitable for development workloads, low-traffic website, micro services, etc.
- Naming examples: A1 v2, A2 v2, A4 v2, A8 v2

Series B: Burstable
- Series B VMs are a low-cost option that have the ability to burst to significantly higher CPU performance when the demand rises.
- Naming examples: B1S, B2S, B4MS, B12MB, B16MS, B20MS

Series D: General Purpose
- Series D VMs are optimized to meet the requirements of most production workloads. There are many variants in the D family, some of which emphasize certain features such as fast CPUs or fast disks.
- Naming examples: D2a v4, D2as v4, D2d v4, D2ds v4, D2s v4

Series F: Compute Optimized
- Series F VMs are optimized for compute-intensive workloads. They have a high CPU-to-memory ratio.
- Naming examples: F1, F1s, F2s v2

Series E, G, and M: Memory Optimized
- Series E, G, and M VMs are ideal for memory-intensive enterprise applications, such as database servers.
- Naming examples: E2a v4, E2as v4, E2ds v4, G1, G1s, M8ms, M208s

© 2022 Pierre Lidome

Series L: Storage Optimized
- Series L VMs feature high throughput, low latency, and directly mapped local NVMe storage.
- Naming examples: L8s v2, L4s

Series NC, NV, ND: Graphics Optimized
- Series NC, NV, and ND VMS have high-end GPUs and target applications such as visualization, deep learning, and predictive analytics.
- Naming examples: NC6, NC6s v2, NC4as T4, NV6, NV12s, NV4as v4, ND6s, ND40rs v2, ND96asr

Series H: High Performance Computing
- Series H VMs are designed for high-performance computing in applications such as financial risk modeling, seismic and reservoir simulation, and genomic research.
- Naming examples: H8, HB60rs, HB120rs v2, HC44rs

While you will encounter any combination of these VMs, logging and forensics analysis are performed the same way irrespective of the model.

1. https://for509.com/windowspricing
2. https://for509.com/linuxpricing

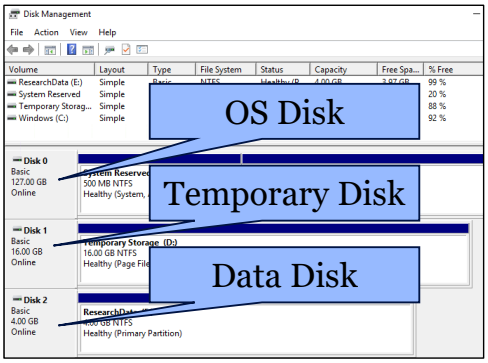
© 2022 Pierre Lidome

Technet24

VM Storage: Managed Disk

- Managed Disk
  - 99.999% availability
  - Azure manages backup and uptime
  - Size and performance are guaranteed based on selection
  - Can upgrade size and type
- Four types
  1. Standard HDD: slow but cheap
  2. Standard SSD: standard for production
  3. Premium SSD: fast and high performance
  4. Ultra disk: for most demanding and data intensive workloads

- Monthly cost based on:
  - Disk type and size
  - Snapshots
  - Outbound data transfers
  - Transactions



A managed disk[1] is very similar to the physical disk you have in your laptop or desktop. When configuring a managed disk, you select the size and the type: Standard HDD, Standard SSD, Premium SSD, or Ultra disk.[2]

However, there is a big difference that is very relevant to our investigation: **cost**.[3] Unlike the disk in your laptop or desktop, which has a one-time cost, Azure managed disks have a recurring cost. There are five components to the cost of a managed disk:

1. Disk type: SSD (various levels as previously indicated) or HDD.
2. Disk size: As expected, the larger the disk, the more it costs.
3. Snapshots: Billed based on the size of the snapshot; these are very important for our investigations.
4. Outbound data transfers: Transferring data out of Azure will incur billing for bandwidth usage.
5. Transactions: Billed for each I/O operation.

Most VMs will have two or more managed disks:

- OS disk, which is selected when the VM is created.
- Temporary disk used for short-term storage (example: page or swap files). Not all VM types have a temporary disk.
- One or more data disks, which are user created.

As a preview of the last section of the class, one method we use to perform our investigations is to snapshot the OS disk of the compromised machine, apply that snapshot to a new disk, and mount that new disk as a data disk on a fresh VM. This is why understanding managed disks is so important.

1. https://for509.com/diskintro
2. https://for509.com/disktypes
3. https://for509.com/diskprice

© 2022 Pierre Lidome

Case Study: Detecting a Crypto Mining VM

- Cloud crypto mining is a favorite action-on-objective for bad actors
- Only **N**-series VMs have GPUs
- Monitor logs for creation of **N**-series VMs
- Create workflow to notify subscription owner
- PowerShell script provided in the notes

```
PS> .\VmSize.ps1 (script provided in the notes & class GitHub)

VmID                                    VmSize
----                                    ------
f57b8807-a5bb-4e87-8f7d-d7ad0a3f0f80    Standard_NV4as_v4

088e4944-d5a9-49a4-b1ea-e927e2          Standard_D2s_v3
```

Are GPU-enabled VMs normal in this subscription?

The next section will have an in-depth discussion of the various Azure log sources. However, while we are on the topic of virtual machines, let's quickly discuss an idea to potentially detect the creation of crypto mining VMs. A bad actor whose action-on-objective might be to run a crypto miner will first need to access your Azure subscription. Perhaps they get extremely lucky and are able to compromise a VM that features a GPU. But most likely that won't be the case, and they will need to create an appropriate VM (or many such VMs).

As we saw in the last slide, GPU-enabled VMs belong to the **N**-series of VMs (NC, ND, and NV). The idea is to monitor the VM creation logs for this series of VMs. In the log, they will show up in a field called "vmSize" as "Standard_N*" where N* stands for the specific model name. An example would be "Standard_NV4as_v4".

While the log may reflect a legitimate creation of such a VM, a workflow could be designed to notify the subscription owner of this new specialized VM. It would then be up to the subscription owner to decide whether or not this is a legitimate VM.

If your organization doesn't normally use GPU-enabled VMs, this is a very simple early warning system. On the other hand, for an organization that dynamically builds and tears down a large number of these kinds of VMs, additional filters will need to be implemented.

This workflow can be implemented in two different manners, depending on the organization. If your organization imports Azure logs in a SIEM, writing rules for that condition would be the simplest method.[1]

© 2022 Pierre Lidome

Technet24

Here is an example rule for Splunk:

```
Index=ms_azure properties.hardwareProfile.vmSize=Standard_N* | dedup
properties.vmId | stats count by properties.hardwareProfile.vmSize
```

Example 30-day snapshot from a large corporation:

Properties.harwareProfile.vmSize	count
Standard_NV12s_v3	3867
Standard_NV6	474
Standard_NV12_Promo	22
Standard_NV12	7
Standard_NV6_Promo	7
Standard_NC24s_v3	6
Standard_NV32as_v4	4

Over 4,000 GPU-enabled VMs created in a 30-day period may seem suspicious. However, this is an example of a company that processes large amounts of data over short periods of time and therefore dynamically creates and tears down these VMs. The next step would be to add a filter and narrow down either by subscription or resource group to remove those that are authorized to create these GPU-enable VMs.

For organizations that don't have a SIEM, here is some PowerShell code that will extract the information from the log:[2]

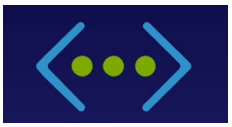
```
$results = get-azlog -ResourceProvider "Microsoft.Compute" -DetailedOutput
$results.Properties | foreach {$_} | foreach {
   $contents = $_.content
   if ($contents -and $contents.ContainsKey("responseBody")) {
      $fromjson=($contents.responseBody | ConvertFrom-Json)
      $newobj = New-Object psobject
      $newobj | Add-Member NoteProperty VmId $fromjson.properties.vmId
      $newobj | Add-Member NoteProperty Vmsize
$fromjson.properties.hardwareprofile.vmsize
      $newobj
      }
}
```

There are many other parameters that you can extract from the logs. This is a simple example to illustrate the possibilities. One downside of using PowerShell rather than a SIEM is that the script must be executed for each subscription.

1. https://for509.com/splunkautomation
2. Shoutout to Arjun Bhardwaj and Michael Getachew for their assistance with PowerShell scripting.

Azure Virtual Network (VNet)

- Required for communications between VMs and/or the internet.
  - Address space
    - Range of IP address available for the resources in that VNet
  - Subnet
    - Smaller network to facilitate resource grouping and security
  - Regions
    - Belongs to a single region and single subscription
    - Every resource on the VNet must be in the same region
  - Network Address Translation (NAT)
    - Azure public IP addresses are NATed
    - Virtual machines with a public IP only know their private IP



Significant consequence for DFIR, as logs will only show the machine's private IP

Azure Virtual Network (VNet) is the glue that allows other Azure resources to communicate with each other and with the internet.[1]

When creating a VNet, Azure will assign a set of RFC 1918 IP addresses, typically 10.0.0.0/24 for the first VNet you create. This private address space allows your VMs to communicate with each other. You can assign any public or private address range you want to your VNet. Creating a VNet doesn't incur any charges.

To communicate with the internet, you will need to assign your VM a public IP address. There is a recurring charge to use Azure's public IP addresses.

There are three ways to communicate with on-premises resources:

1. Point-to-site virtual private network (VPN): A connection between a VNet and a single computer.
2. Site-to-site VPN: A connection between your on-premise VPN device and an Azure VPN gateway deployed in the VNet.
3. Azure ExpressRoute: A connection between your premises and Azure through an ExpressRoute partner. This connection is private and doesn't go over the internet.

From an incident response point of view, it's very important to understand network topology and communication channels between Azure resources. We will discuss network security groups in the next slide.

1. https://for509.com/virtualnetwork

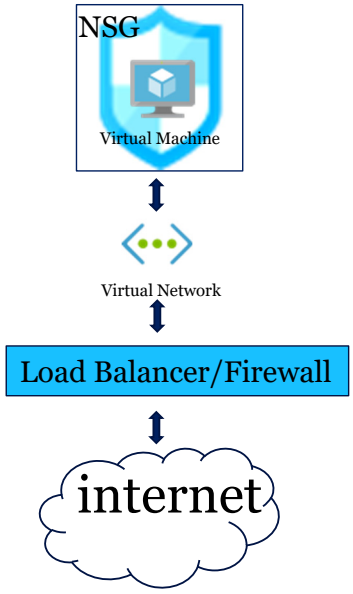
© 2022 Pierre Lidome

Technet24

Network Security Group

- Stateful packet filtering based on:
  1. Source IP
  2. Source Port
  3. Destination IP
  4. Destination Port
  5. Protocol
- For each rule, you must specify:
  1. Priority
  2. Action: Allow or Deny
- Flow logs are collected at **1-min** intervals

SSH and RDP are opened by default



To protect your VM, Azure will automatically create a network security group (NSG). The NSG allows you to control traffic in and out of the subnet. The NSG conducts a stateful inspection of the traffic based on the Source IP, Source Port, Destination IP, Destination Port, and Protocol.[1] This is to be considered a very basic firewall. Additional offerings are available from Azure and third parties for full-fledged firewalls that include application layer inspection.

NSG rules are read in order, based on the priority number assigned to each rule, from 100 to 4096 (excluding Azure-defined rules in the 65000 range). 100 is the highest priority rule, and 4096 is the lowest.

From an incident response point of view, it's important to understand the network topology and associated control. For example, you can imagine the scenario where you are told that a firewall rule was in place to block nefarious traffic, but your investigation reveals that the priority of that rule was too low to be effective and that a higher-priority rule was in fact allowing that traffic to the VM.

Critical to any investigation is the availability of network flow logs.[2] In Azure, flow logs are collected at 1-minute intervals which is not configurable.

The first 5GB of flow logs per month are free. Beyond that, they cost USD $0.50 per GB in addition to the storage account charges.

1. https://for509.com/nsg
2. https://for509.com/flowlogs

© 2022 Pierre Lidome

Network Virtual Appliance

Many devices can shape network traffic and can produce their own set of logs. Review the network topology before you start your investigation and look for the presence of network-shaping devices. These network virtual appliances are offered by both Microsoft Azure and third parties:

- Load Balancer
- Firewall
- Application Gateway
- VPN gateway
- WAN optimization appliance
- Virtual router

There are many other pieces of network infrastructure that can shape the traffic. These may have various impacts to your incident response or forensic investigation. You need to be aware of their presence so that you may investigate their logging capabilities, as they may provide further insight to your investigation.

Azure offers the following options:
- Azure Load Balancer, which will evenly distribute incoming network traffic across a groups of resources.
- Azure Firewall, which offers both stateful network and application-level filtering.
- Application Gateway, which will protect your application from common web vulnerabilities.
- VPN gateway, which connects your on-premise network to Azure.

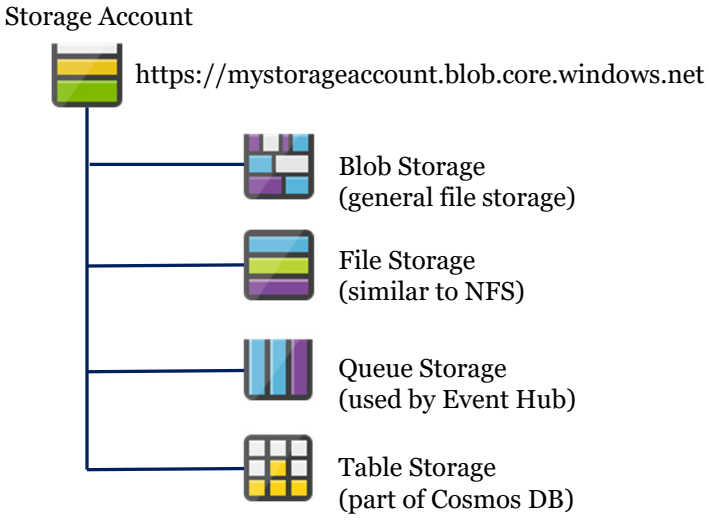
In addition, third-party vendors offer many network virtual appliances such as firewalls, WAN optimization, VPN access server, virtual router, etc. These offerings can be found on the Azure Marketplace.[1]

1. https://for509.com/marketplace

© 2022 Pierre Lidome

Technet24

Storage Account



- Single namespace for all storage resources; hence, all names must be unique across all of Azure
- Each storage service has its own web address
- Storage accounts can contain blob storage, file storage, queue storage, and table storage

Storage will be a key component of any investigation you conduct. In addition, many logs will only be retained if you create dedicated storage for them. There are four types of storage on Azure:[1]

1. Blob: Massively scalable object storage for unstructured data.
2. File: Simple, distributed, cross-platform filesystem.
3. Queue: Service to store and retrieve messages.
4. Table: Table storage is now part of Azure Cosmos DB.

We will focus on blob storage.[2]

Storage accounts provide a unique namespace for your data. As we discussed in the previous section, every resource in Azure has a unique resource ID. When you create a storage account, you are able to access your data directly using the URL, https://mystorageaccount.blob.core.windows.net, where "mystorageaccount" is the name you selected for your storage account.

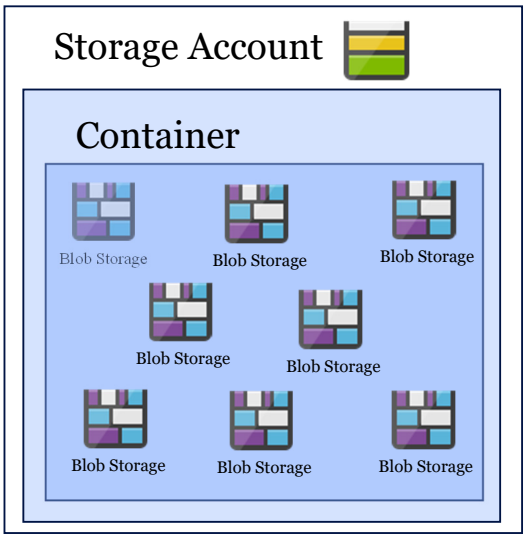
This will be a very useful feature when you need to download logs for analysis or safekeeping.

1. https://for509.com/storageintro
2. https://for509.com/blobintro

© 2022 Pierre Lidome

Storage: Blob

- Three levels to get to a blob:
  1. Storage account
  2. Container in the storage account
  3. Blob in a container
- Blob = Binary Large Object
  - Block Blob
    - Any file up to 4.75TB
  - Append Blob
    - Works well for logging where data is constantly appended
  - Page Blob
    - Used for virtual hard drives up to 8TB



Blob stands for Binary Large Object. It's a way for Azure to store an arbitrarily large amount of unstructured data. Any type of data can be stored in blobs, which is very convenient for videos, images, and particularly text. As you can imagine, for our purposes it's very convenient to store logs; large amounts of logs.

There are three types of blobs:

1. **Block blobs** to store text and binary data. Since block blobs are made up of blocks of data, they can be managed individually and can store up to 4.75TB of data. Microsoft is currently previewing block blobs up to 190.7TB.
2. **Append blobs** are optimized for append operations and are ideal for logging data.
3. **Page blobs** store random access files up to 8TB. They are usually used to store virtual hard drive files (VHD).

There are two characteristics that are very convenient with blobs:

1. With the correct permission, they can be accessed directly over the internet via HTTP or HTTPS using the following URL: http://mystorageaccount.blob.core.windows.net.
2. There are many methods to transfer data in or out of blobs. The two easiest ones are AzCopy[1] and Azure Storage Explorer[2] (which leverages AzCopy).

1. https://for509.com/azcopy
2. https://for509.com/storageexplorer

© 2022 Pierre Lidome

Technet24

Lab 2.1 Preview

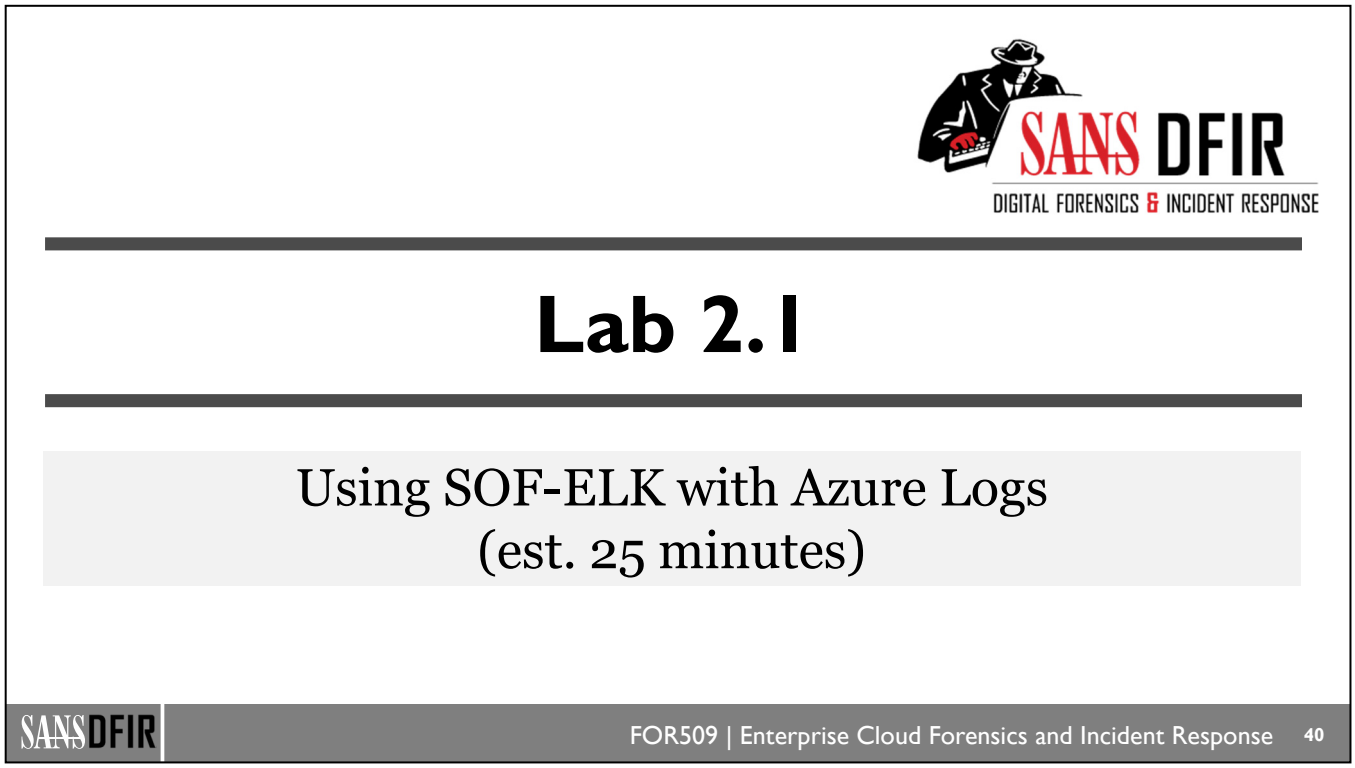
Lab 2.1 will explore the Azure logs to discover interesting information about the Pymtechlabs environment.

The following fields will be important in this lab:

- `ips`
- `resource_id`
- `user_name`
- `user_principal_name`
- `user_id`

This page intentionally left blank.

© 2022 Pierre Lidome



This page intentionally left blank.

© 2022 Pierre Lidome

Technet24

FOR509.2: Microsoft Azure

Section 2.1: Understanding Azure

Section 2.2: VMs, Networking, and Storage

Section 2.3: Log Sources for IR

Section 2.4: Virtual Machine Logs

Section 2.5: In-Cloud IR

This page intentionally left blank.

© 2022 Pierre Lidome

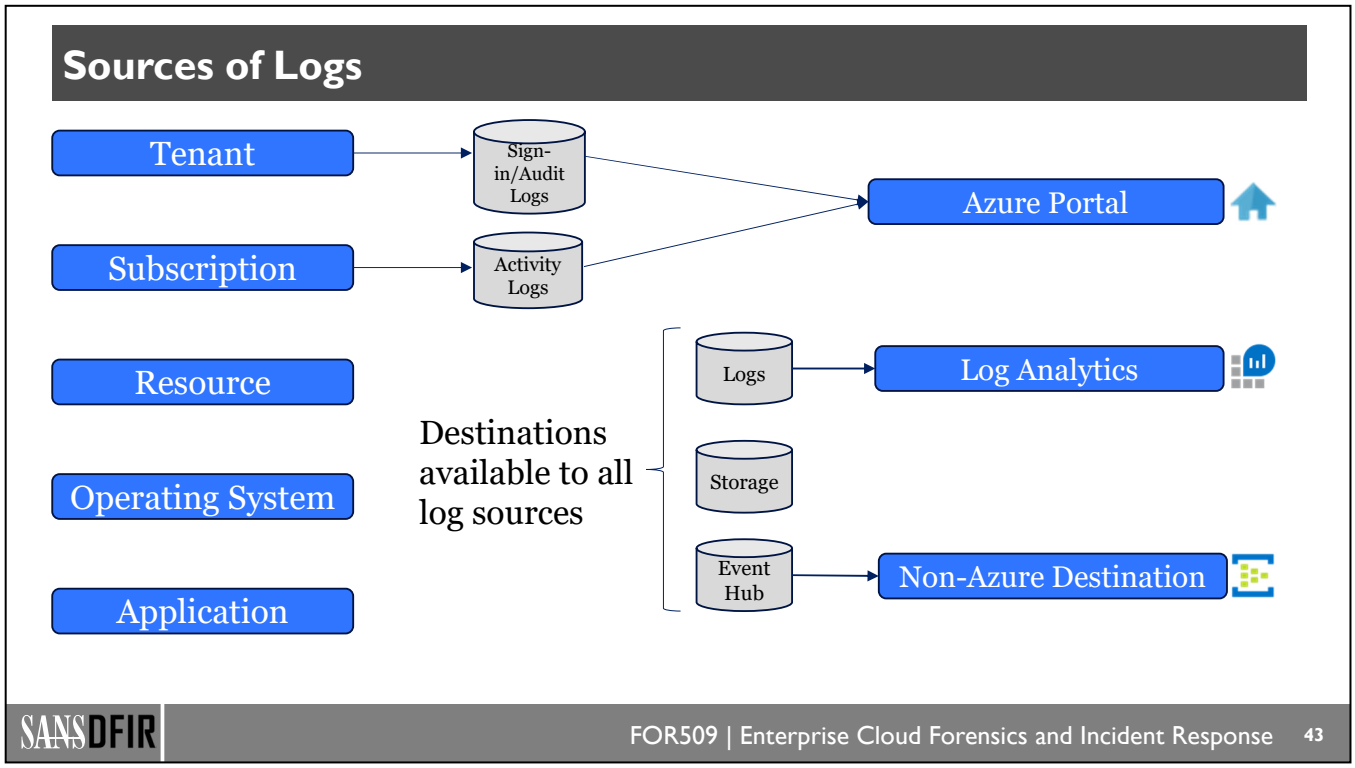
Microsoft Azure Roadmap

- 2.1: Understanding Azure
- 2.2: VMs, Network, and Storage
- 2.3: Log Sources for IR
- 2.4: Virtual Machine Logs
- 2.5: In-Cloud IR

- Sources of Logs
- Log Analytics Workspace
- Tenant Logs
- **Lab 2.2**: AAD Password Spray Attack
- Subscription Log
- **Lab 2.3**: Tracking Resource Creations
- NSG Flow Log
- Storage Logs
- **Lab 2.4**: Detecting Data Exfiltration

This page intentionally left blank.

© 2022 Pierre Lidome
Technet24



To support your investigation, it's crucial to understand all the log sources in Azure. There are five sources of logs we will discuss in this section:[1]

1. Tenant logs
2. Subscription logs
3. Resource logs
4. Operating system logs
5. Application logs

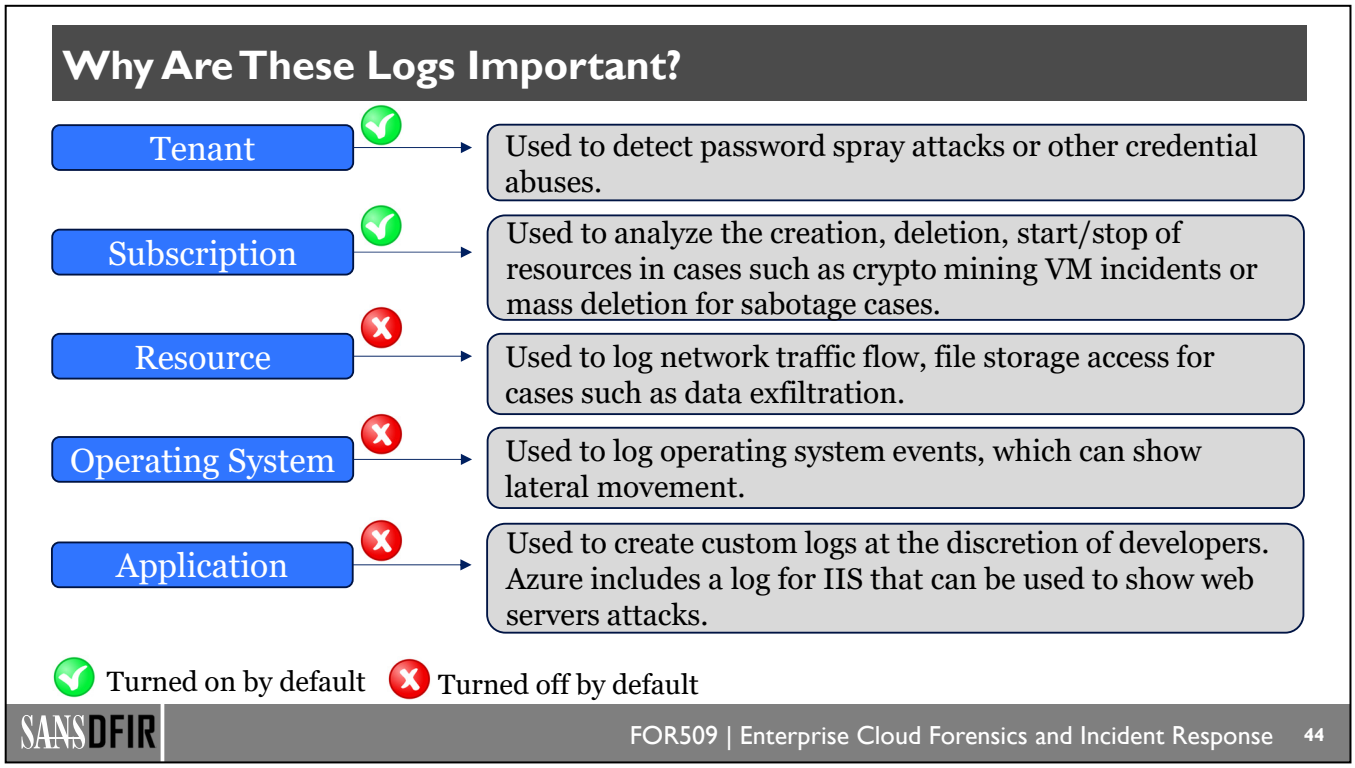
Where to find these logs and how to access them are the next pieces of the puzzle. Logs can be written to multiple locations at the same time, which means that there may be more than one way to consume the information.

While some log sources are automatically configured by Azure, that's not the case for most of them. In order to store logs, storage must be allocated, which implies a recurring cost as we previously discussed. Unfortunately, when called to perform incident response, you may find that the subscription owner didn't configure any of the optional log sources, leaving you with very little information to analyze.

Educating your team on the importance of configuring these log sources is really the first step in incident response. The ideal configuration is to continuously export these logs to a SIEM so that they are not stored where a bad actor may be able to delete them.

As we examine the five log sources mentioned above, we will discuss where to find them, how to configure them, and how to export them for analysis.

© 2022 Pierre Lidome



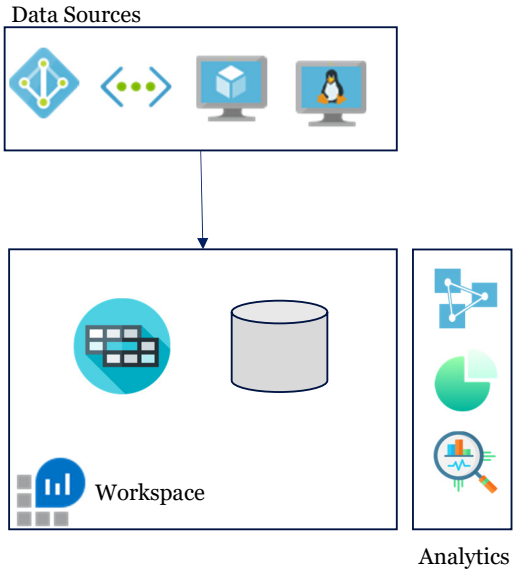
With so many logs available in Azure, it's easy to become overwhelmed. Investigations will normally make use of multiple logs, and these are just a few examples of the logs you could use in various DFIR situations.

© 2022 Pierre Lidome

Technet24

Log Analytics Workspace Overview

- Workspace
  - Similar to a data lake
- End-to-end analytics: combines multiple data sources
  - Azure resources
  - Subscription logs
  - Azure Active Directory
  - Non-Azure logs
- Data organized in a table
  - Each log source has its own table
- Designed for up to 6GB/min and 4TB/day



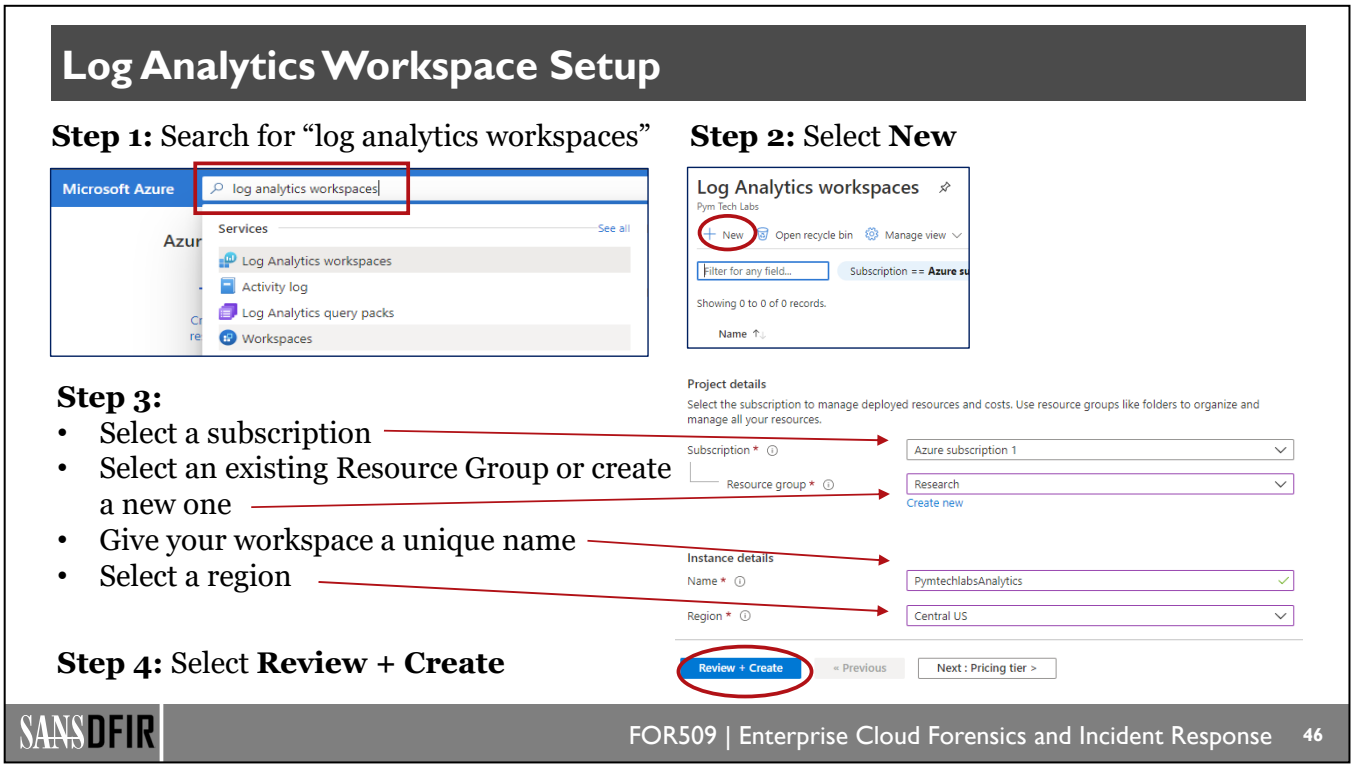
Microsoft will store some logs by default and make them available on the Azure portal. Others will only be retained if specialized storage has been created. Microsoft calls this specialized storage the Log Analytics workspace. The Log Analytics workspace collects and aggregates logs from various data sources—both Azure-based and non-Azure. The workspace is organized into tables, and each data source will create its own tables.

While you can create multiple workspaces to segregate logs, it's generally not needed, as a default workspace can accept logs at a rate up to 6GB/min with a maximum of 4TB/day. Customized workspaces can be created with greater limits if needed.

Access control is a key concern and can be customized based on your company's security policy. Azure offers many options, which are well documented in the references.

References:
https://for509.com/logdeployment
https://for509.com/logaccess

© 2022 Pierre Lidome



Before we dig into the logs, we need to create a Log Analytics workspace. The Log Analytics workspace can be created via the Azure portal,[1] Azure CLI,[2] or PowerShell.[3] You can reference the Microsoft documentation if you wish to create the workspace using the CLI or PowerShell. In this slide, we will use the Azure portal.

Step 1: Sign in to the Azure portal and search for the "log analytics workspaces" service in the search bar

Step 2: Select **New**

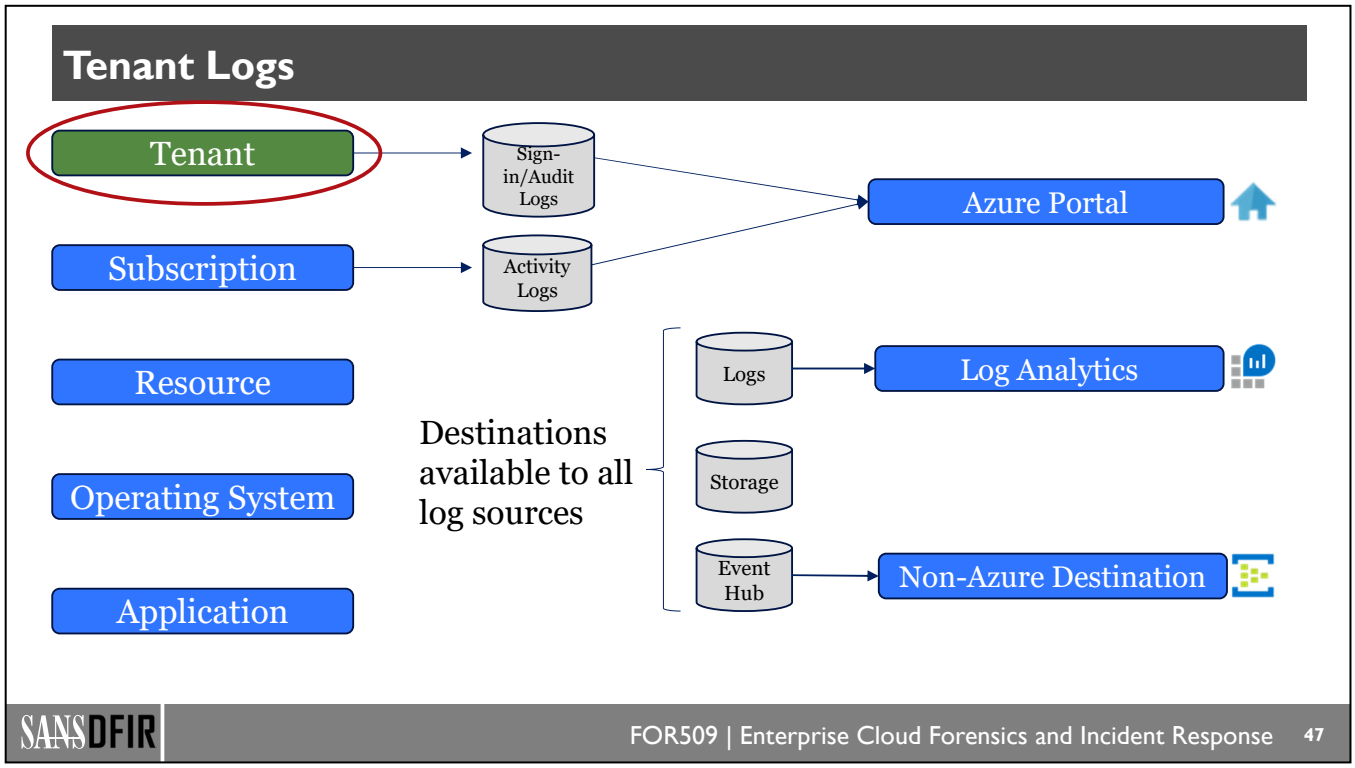
Step 3: Enter the information requested: Subscription, Resource Group, Name, and Region

Step 4: Select **Review + Create**

1. https://for509.com/law-portal
2. https://for509.com/law-cli
3. https://for509.com/law-powershell

© 2022 Pierre Lidome

Technet24



We will start our exploration of the Azure log sources with the tenant logs.

The tenant log contains information about operations conducted by tenant-wide services. This is where you will find the Azure Active Directory (AAD) log.[1] The AAD log contains audit logs, sign-in logs, and provisioning logs.[2]

1. https://for509.com/logs-aad
2. https://for509.com/logs-provisioning

© 2022 Pierre Lidome

Tenant Logs Agenda

- Sources of Logs
- Log Analytics Workspace
- Tenant Logs
- Lab 2.2: AAD Password Spray Attack
- Subscription Log
- Lab 2.3: Tracking Resource Creations
- NSG Flow Log
- Lab 2.4: Detecting Data Exfiltration

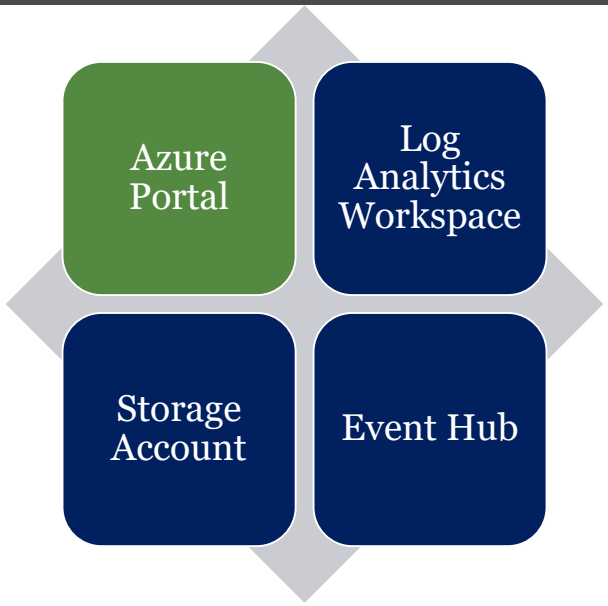
- Azure Portal
  - Sign-in logs
  - Audit log
- Log Analytics Workspace
- Storage Account
- Azure Storage Explorer
- Import into SOF-ELK
- Event Hubs
- Graph API

This page intentionally left blank.

© 2022 Pierre Lidome

Technet24

Tenant Logs: Azure Portal



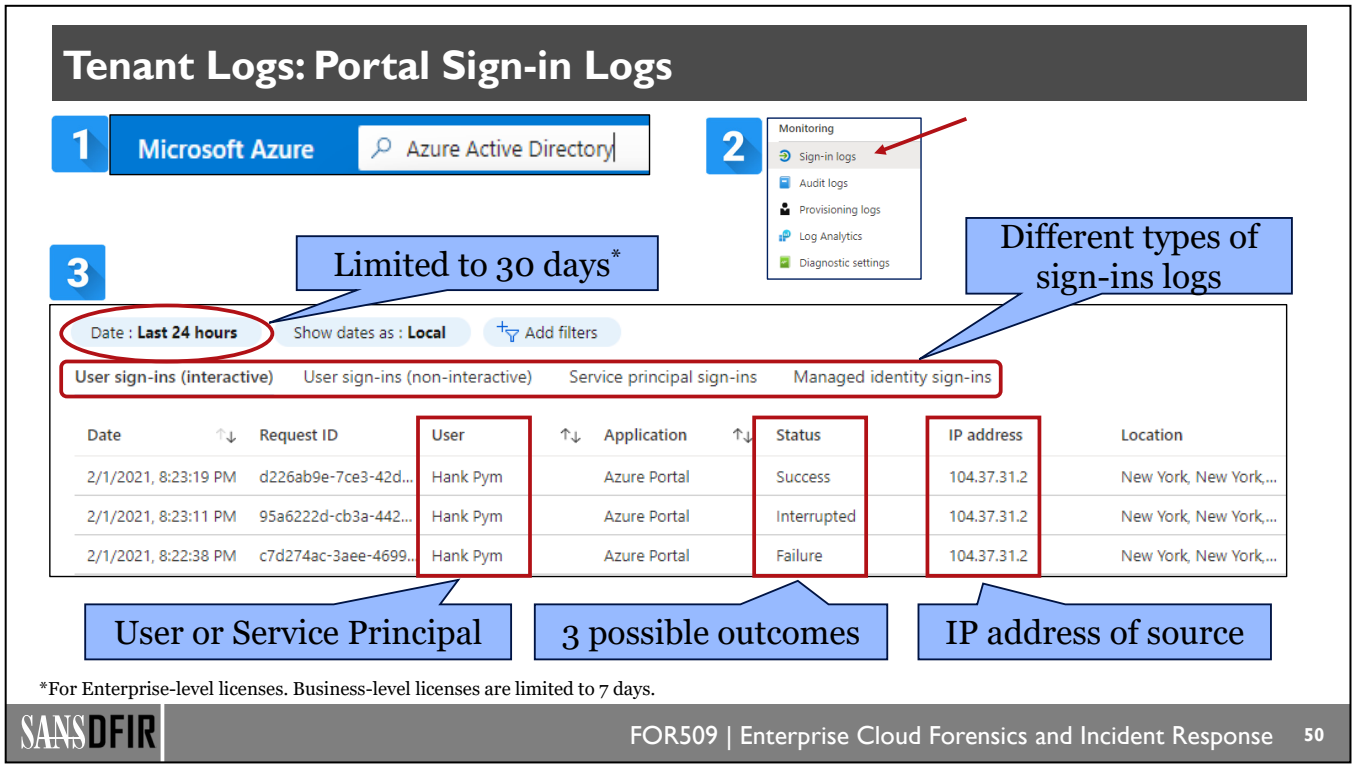
- Portal Sign-in Logs
- Description of Sign-in Logs fields
- Sign-in Logs Failed MFA Example
- Portal Audit Log

There are four actions you can take with these logs:
1. View them directly on the Azure portal.
2. Store them in a Log Analytics workspace.
3. Send them to a storage account for archival.
4. Send them to a SIEM by using the event hub.

These four options will be the similar for the other types of logs: subscription, resource, operating system, and application.

Note: Since there are multiple logs that contains sign-in information, we will refer to **logs** rather than log in this section.

© 2022 Pierre Lidome



The portal is a quick and easy way to check the sign-in log. Unfortunately, it's limited to the last 30 days.

The key fields are the date, user, status, and IP address. All the way to the right (not shown on the slide), there is a column that will tell you if multifactor authentication was used.

Notice at the top the four tabs for the different kinds of sign-ins: user, user non-interactive, service principal, managed identity.

You will see three possible statuses:

- Success
- Failure
- Interrupted

Success and failure are self-explanatory. Interrupted is due to the "Stay signed in?" window.[1] The documentation states that if a user "abandons the sign-in attempt," then the status of "Interrupted" will be recorded. Testing indicates that replying "No" may also generate the same status.

1. https://for509.com/staysignedin

© 2022 Pierre Lidome

Technet24

Sign-in Logs Fields (1)

ips	45.56.183.51
log.file.path	/logstash/azure/insights-logs-signinlogs.json
log.offset	543,475
result_signature	None
result_type	0
source_ip	45.56.183.51
tags	process_archive, filebeat, beats_input_codec_plain_applied, azure_json_signin_log, _geoip_lookup_failure
tenant_guid	7e325eda-7945-46d3-ac99-f0dcfeb4628e
type	azure
user_id	675be0f4-2486-4443-bef6-d37d9043ae99
user_name	Hank Pym
user_principal_name	admin@pymtechlabs.com

The event as recorded by Microsoft contains a large number of fields. When we import the data to SOF-ELK via the Logstash ingestion script, we filter the fields to import the most important ones only. This is an important step in order to maintain a high speed of ingestion and performance of SOF-ELK.

The next few slides will show the key fields. In SOF-ELK, you will see every one of these fields for each event.

Field	Description
ips/source_ip	IP address of the system accessing Azure
log.file.path	Name of the file ingested in SOF-ELK
tenant.guid	Unique identifier for your tenant
user_id	GUID of the login user
user_name	Display name of the login user
user_principal_name	Azure Active Directory name of the login user

© 2022 Pierre Lidome

Sign-in Logs Fields (2)

t useragent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.90 Safari/537.36 Edg/89.0.774.54
t useragentinfo.build	
t useragentinfo.device	Other
t useragentinfo.major	89
t useragentinfo.minor	0
t useragentinfo.name	Chrome
t useragentinfo.os	Windows
t useragentinfo.os_name	Windows
t useragentinfo.patch	4389

NOTE

The useragent field can be a very useful field to "fingerprint" the machines attacking your tenant.

This set of fields breaks down the browser user agent. This can be a very useful field in your investigation in order to "fingerprint" the machines trying to access your tenant.

© 2022 Pierre Lidome

Technet24

Sign-in Logs Fields (3)

NOTE

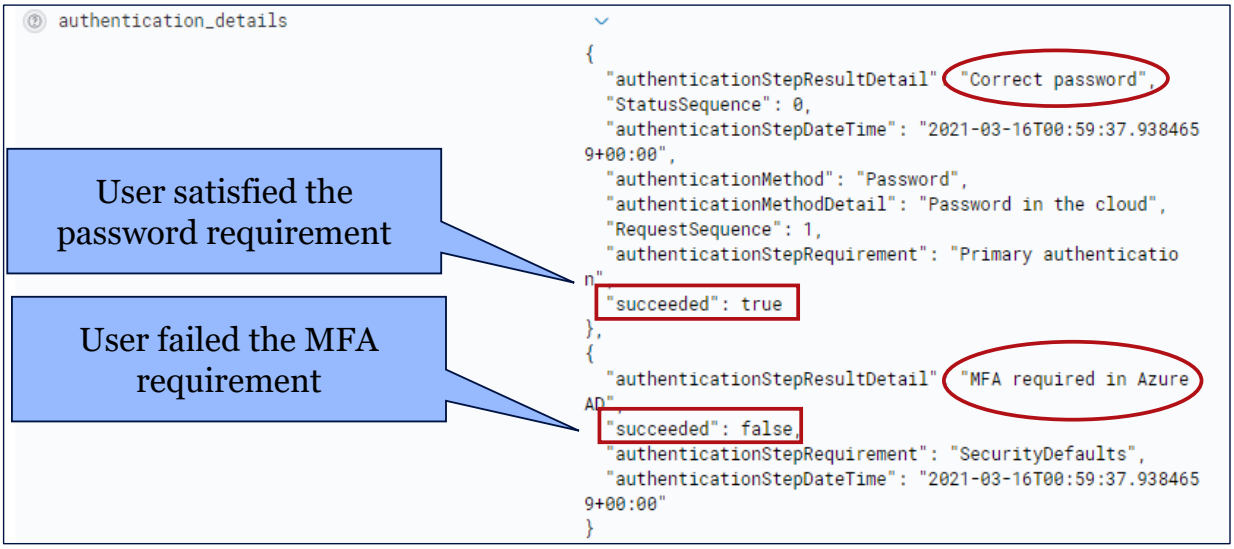
Two-factor authentication should be the standard given today's security threats

The authentication details field is quite interesting, as it provides information about the authentication method used for this specific login. In a tenant where two-factor authentication has been implemented, you will see if the MFA token was accepted.

An interesting search is for successful login with failed MFA. This may indicate that the user's credentials have been compromised, but the bad actor isn't able to fulfill the MFA requirements.

© 2022 Pierre Lidome

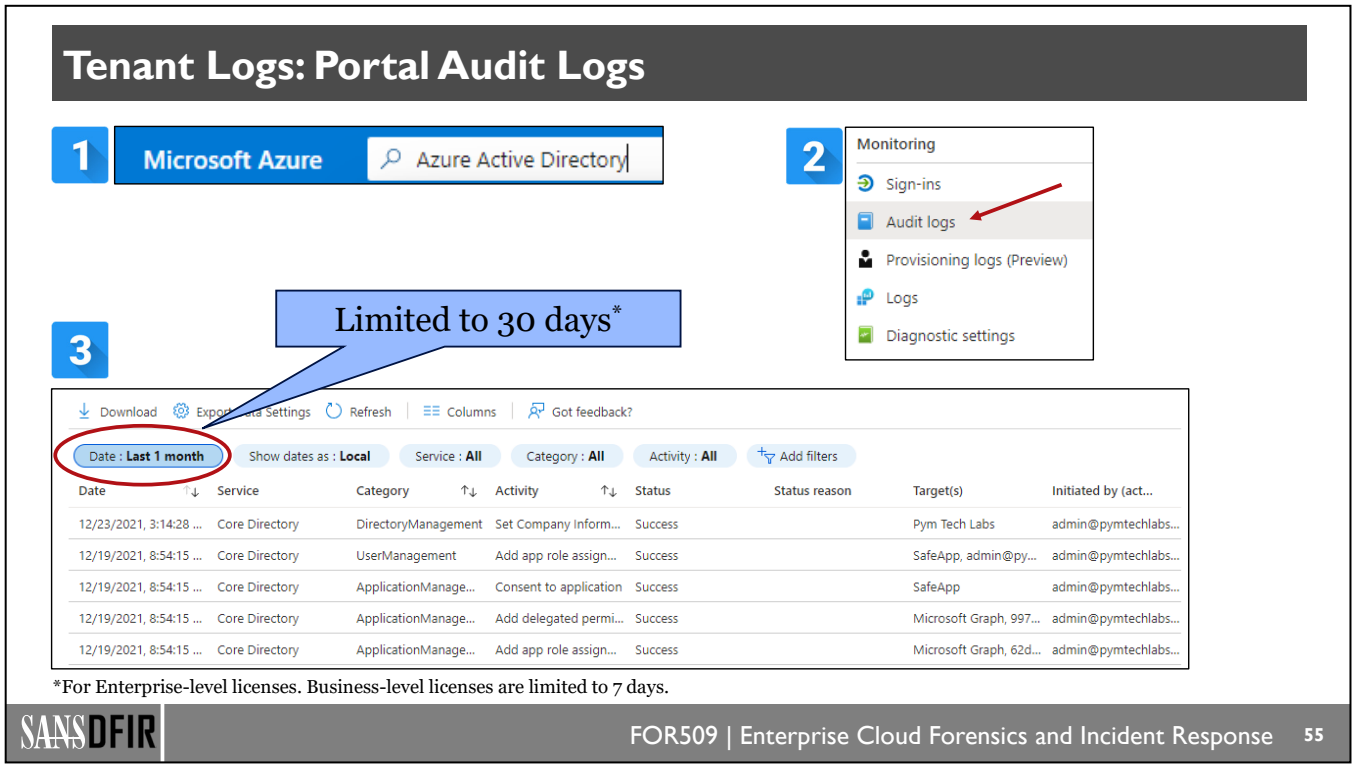
Sign-in Logs: Failed MFA Example



This is an example where the user satisfied the password requirement but failed the MFA requirement. While this will happen frequently due to user error, repeated failures on a large number of accounts coming from the same IP address should be cause for concern.

© 2022 Pierre Lidome

Technet24



The other log available in the portal is the audit log. It's also limited to the last 30 days.

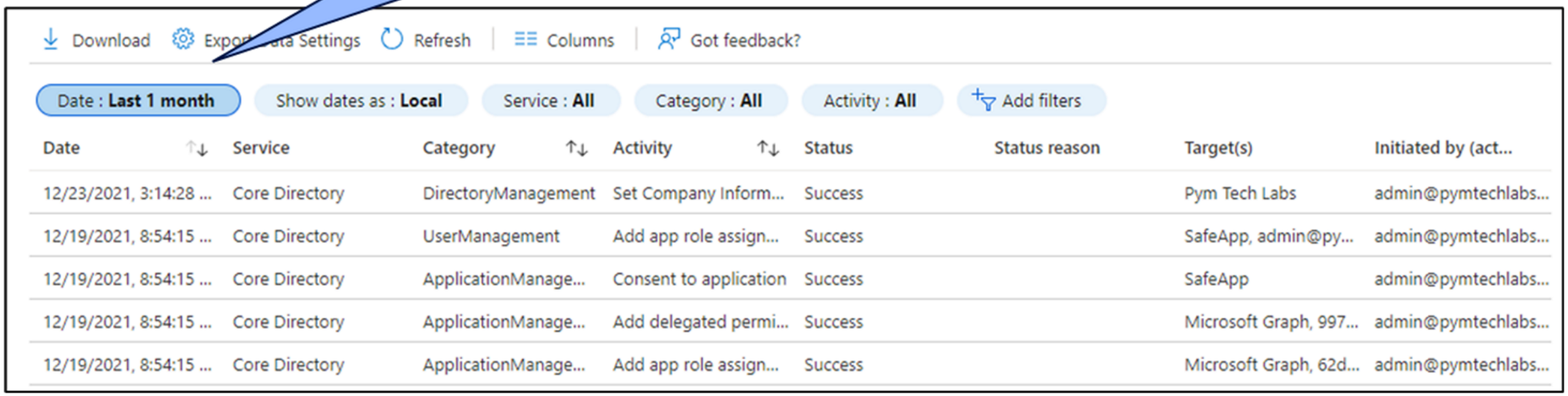
The audit log will show tenant-wide actions such as configuration changes to AAD.

Other than a quick check of recent events, the portal is far from the ideal place to check the tenant logs. A much better place is the Log Analytics workspace, which we will configure in the next slides.

© 2022 Pierre Lidome

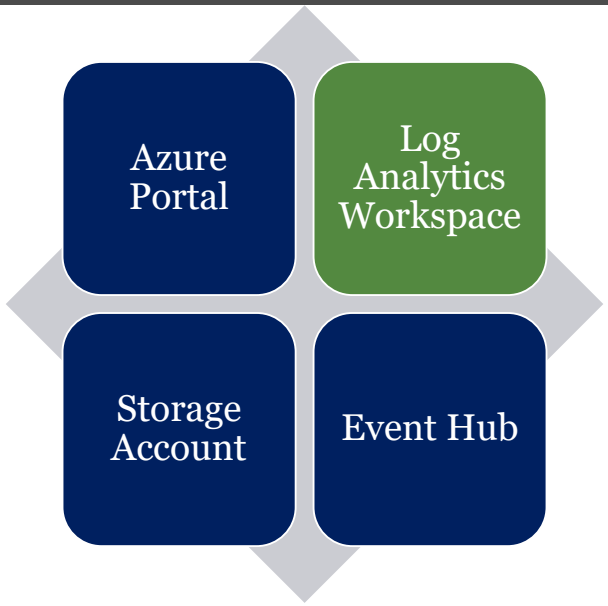
Technet24

Limited to 30 days(1)



© 2022 Pierre Lidome

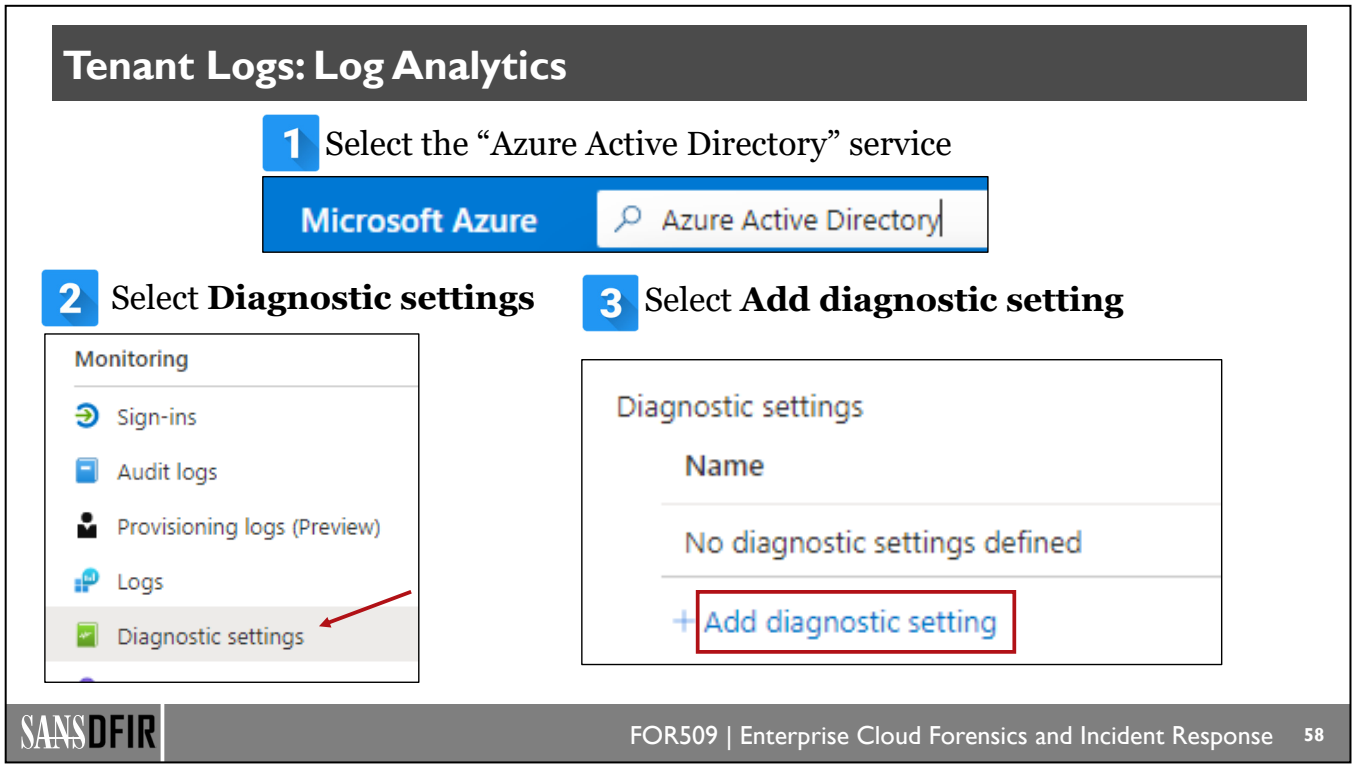
Tenant Logs: Log Analytics Workspace



- Log Analytics Setup
- Log Analytics Queries
- Log Analytics Examples

This page intentionally left blank.

© 2022 Pierre Lidome



While the portal is convenient for a quick search, you have seen that it's very limited. The real power is in the Log Analytics workspace.

In this section about the tenant logs, we will describe how to send the AAD logs to the Log Analytics workspace. In later slides, we will send other logs (such as subscription, resources, etc.) to the same Log Analytics workspace. This provides you a single location to see all your logs, which is very convenient.

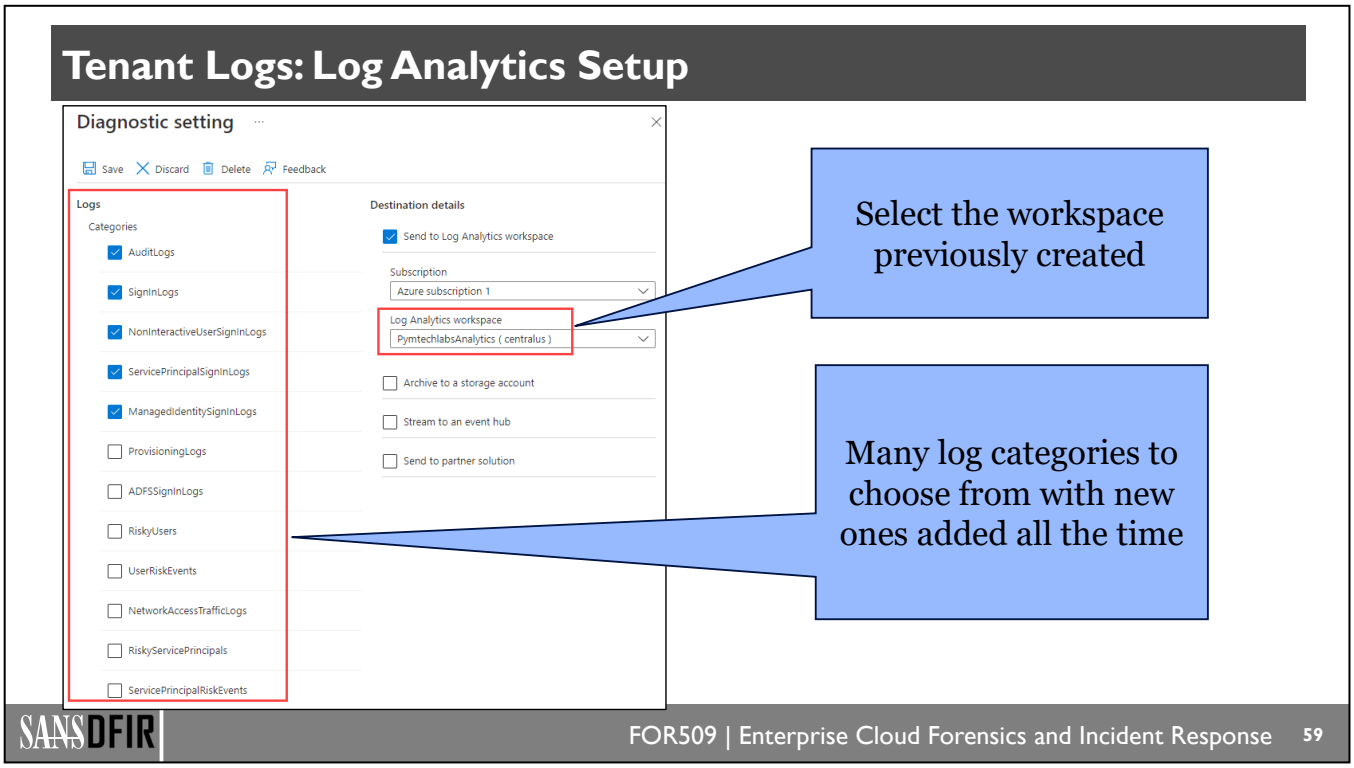
For the AAD logs, you will need to complete the following steps in the Azure portal:

- **Step 1**: Search for and select the "Azure Active Directory" service
- **Step 2**: On the left menu, select **Diagnostic settings**
- **Step 3**: Select **Add diagnostic setting**

The next screen will allow us to select our Log Analytics workspace.

These steps may also be completed via the Azure CLI or PowerShell, which you may find unnecessarily complicated compared to the Azure portal.

© 2022 Pierre Lidome
Technet24



In the second step, we need to select both the AAD logs we want and the Log Analytics workspace where we want to send the selected logs to. For this example, we are using the Log Analytics workspace we created in prior slides: PymtechlabsAnalytics.

Azure stores information in different logs:

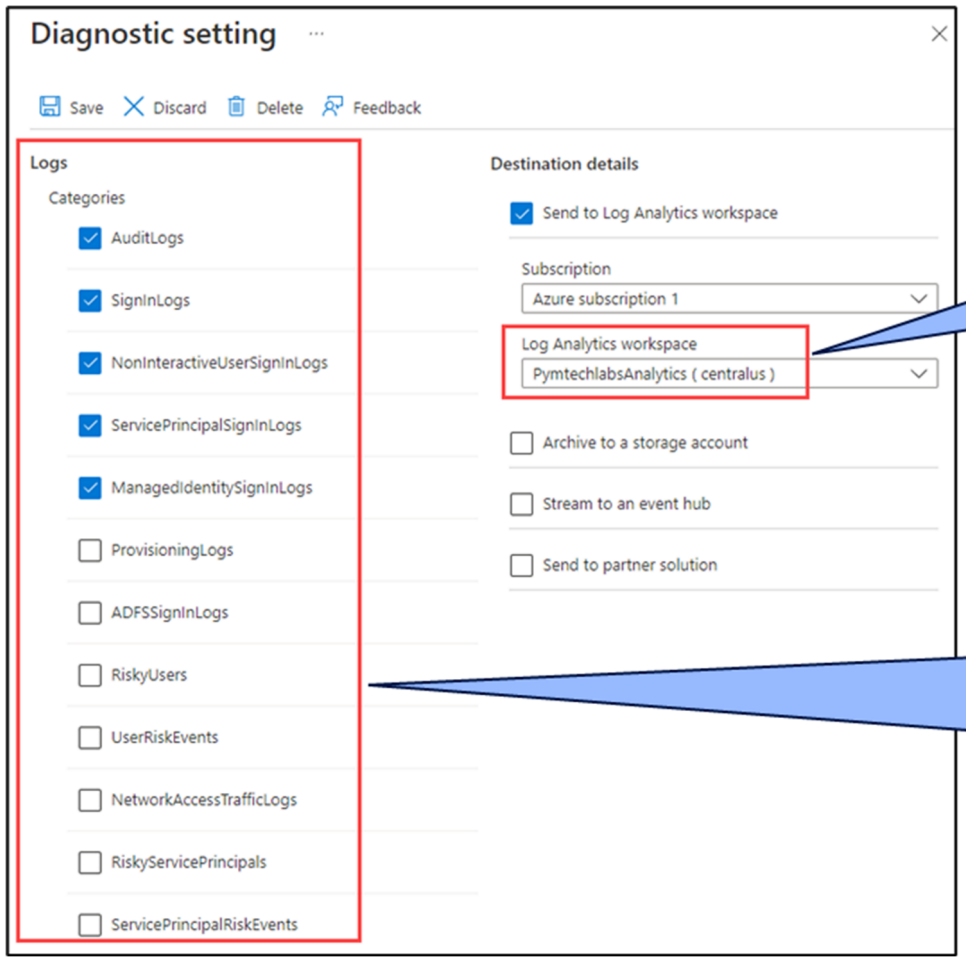
- **Audit logs** include adding or removing users, apps, groups, roles, and policies.
- **Sign-in logs** include usage of managed applications and user sign-in activities. These are further sub-divided into:
  - User sign-in
  - Non-interactive user sign-in
  - Service principal sign-in
  - Managed identity sign-in
- **Provisioning logs** include activity about users, groups, and roles that are provisioned by the Azure AD provisioning service.
- **ADFS Sign Logs** which are generated by the Active Directory Federation Service.
- Logs generated by Identity Protection for Azure AD:[1] **RiskyUsers**, **UserRiskEvents**, **RiskyServicePrincipals**, **ServicePrincipalRiskEvents**.
- **NetworkAccessTrafficLogs** is part of Identity and Network Access which contains Network Traffic Access logs. These logs can be leveraged for policy, risk and traffic management as well as to monitor user experience.

The list of log categories is constantly growing and should be checked frequently.

It might be tempting to save everything, but in a large organization the storage requirements may get very costly. An alternative solution is to store the logs in a storage account, as we will see on the next slide.

1. https://for509.com/identity-protection

© 2022 Pierre Lidome

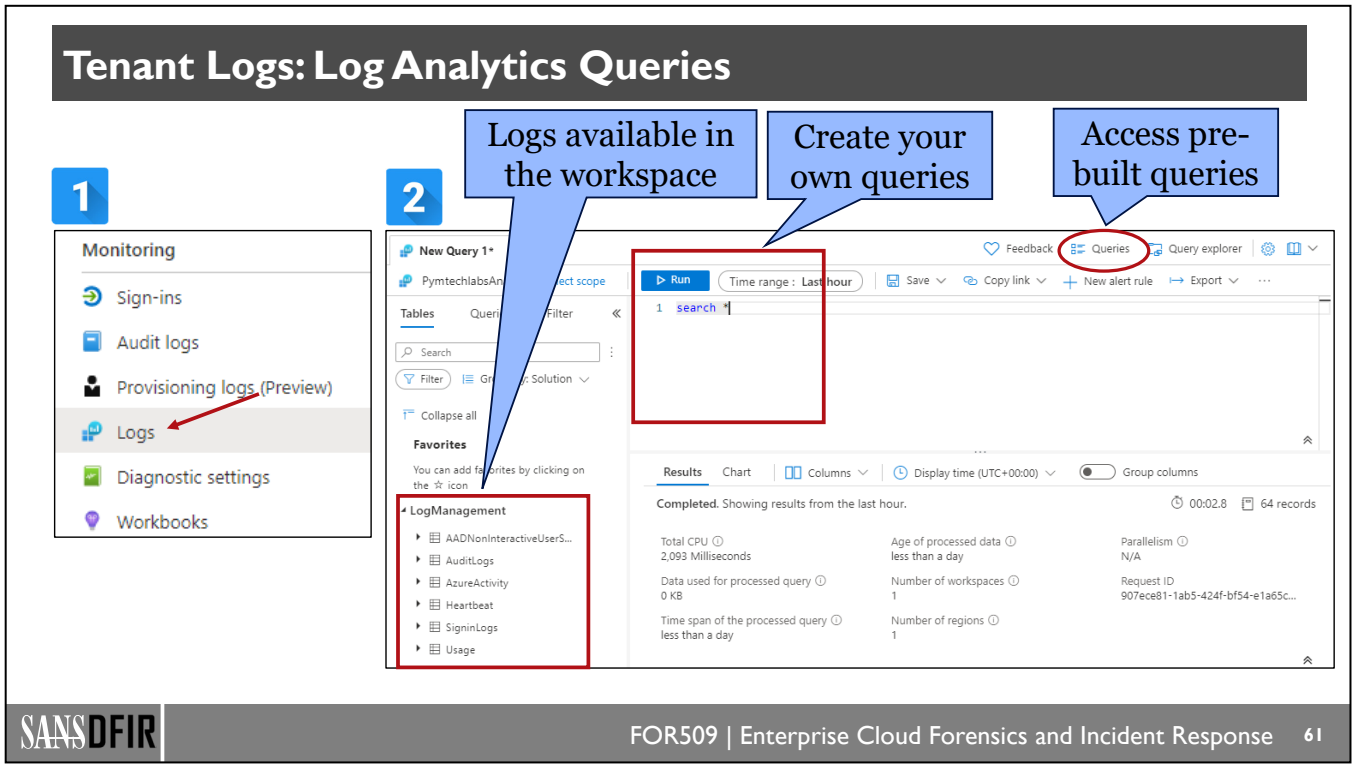


Select the workspace previously created

Many log categories to choose from with new ones added all the time

© 2022 Pierre Lidome

Technet24



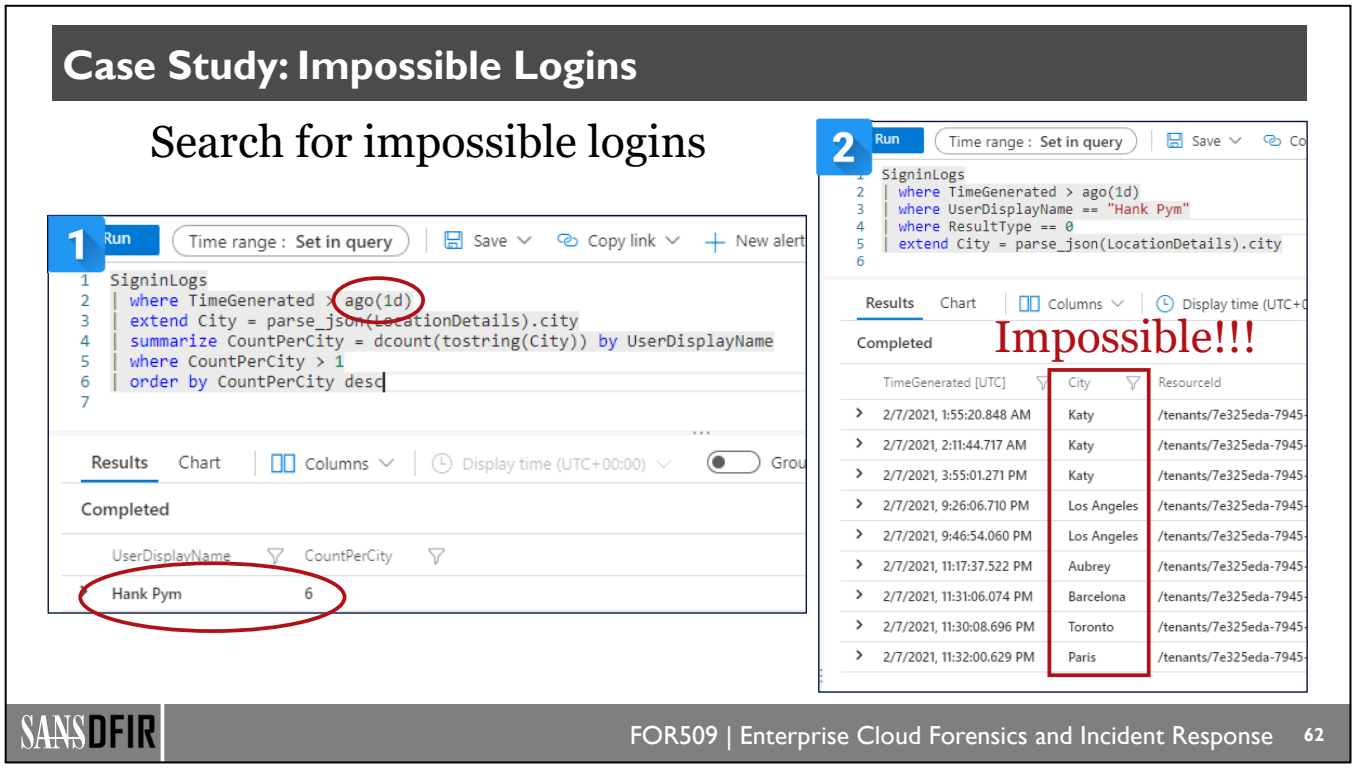
Now that we configured our AAD logs to be sent to the Log Analytics workspace. We can now query these logs using the Kusto Query Language (KQL), which is very similar to SQL.[1] Learning KQL is beyond the scope of this class, but here is a simple example:

```
SigninLogs
| where TimeGenerated > ago(1d)
| where ResultType == 0
```

The first line of the query specifies the log we want to search.
The second line limits the query to the last 24 hours.
The third line limits the query to successful logins.

KQL is a powerful language that can help you analyze and visualize your data easily.

1. https://for509.com/kql-overview



Let's try a KQL search to see if anyone has logged in from multiple cities too quickly (also known as an impossible login). As you can see from the first query, Hank Pym logged on from six different cities in the last 24 hours. That seems abnormal. The second query searches for the name of these cities and shows us the login times as well. Clearly, we may have an issue with Hank Pym's account.

1st Query:

```
SigninLogs
| where TimeGenerated > ago(1d)
| extend City = parse_json(LocationDetails).city
| summarize CountPerCity = dcount(tostring(City)) by UserDisplayName
| where CountPerCity > 1
| order by CountPerCity desc
```

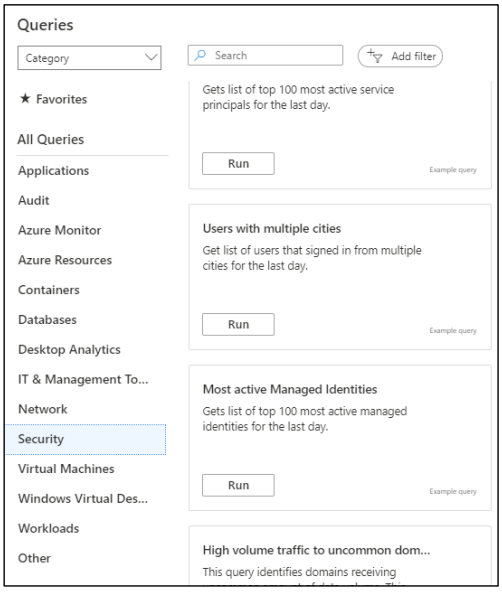
2nd Query:

```
SigninLogs
| where TimeGenerated > ago(1d)
| where UserDisplayName == "Hank Pym"
| where ResultType == 0
| extend City = parse_json(LocationDetails).city
```


© 2022 Pierre Lidome
Technet24

Tenant Logs: Log Analytics Pre-built Queries

Select **Queries** on the top right to access a large number of predefined queries across the entire Azure platform



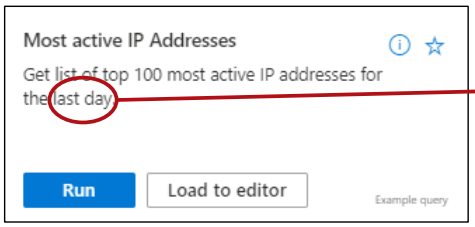
Microsoft provides a number of predefined queries for the various Azure resources. These serve as a great basis for developing your own queries. Obviously, you must first configure the data source to send its logs to the Log Analytics workspace for any of these queries to work.

More information about Log Analytics is available in the reference.

Reference:
https://for509.com/loganalyticsoverview

© 2022 Pierre Lidome

Tenant Logs: Log Analytics Pre-built Query Example




```
// Most active IP Addresses
// Get list of top 100 most active IP addresses for the last day.
AADNonInteractiveUserSignInLogs
| where TimeGenerated > ago(30d)
| summarize CountPerIPAddress = count() by IPAddress
| order by CountPerIPAddress desc
| take 100

// Most active IP Addresses
// Get list of top 100 most active IP addresses for the last day.
AADNonInteractiveUserSignInLogs
```

Results | Chart | Columns | Display time (UTC+00:00)

Completed

IPAddress	CountPerIPAddress
104.214.96.58	67
45.56.183.51	44
45.132.115.50	37
98.194.199.244	19

Queries can be run "as is" or easily modified (changed from 1 day to 30 days in this example)

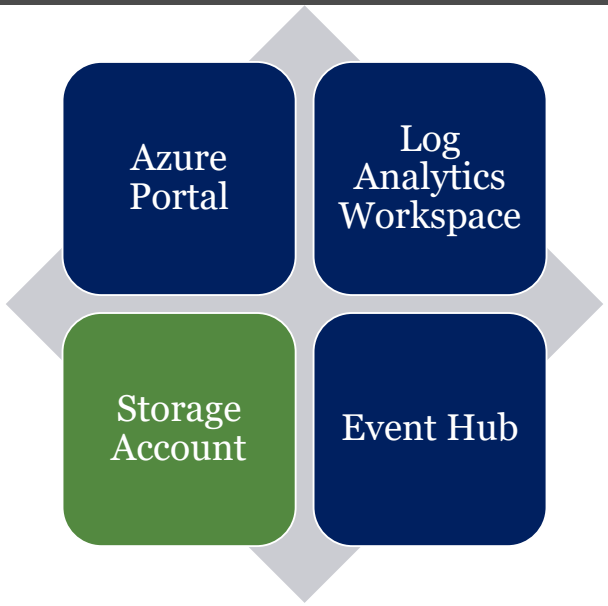
In this example, we selected a predefined query to search for the most active IP addresses. The query can be easily modified to meet your parameters. As an example, we changed this query from only looking at the last 24 hours to searching for the last 30 days.

Predefined queries are a great basis for creating your own queries.

© 2022 Pierre Lidome

Technet24

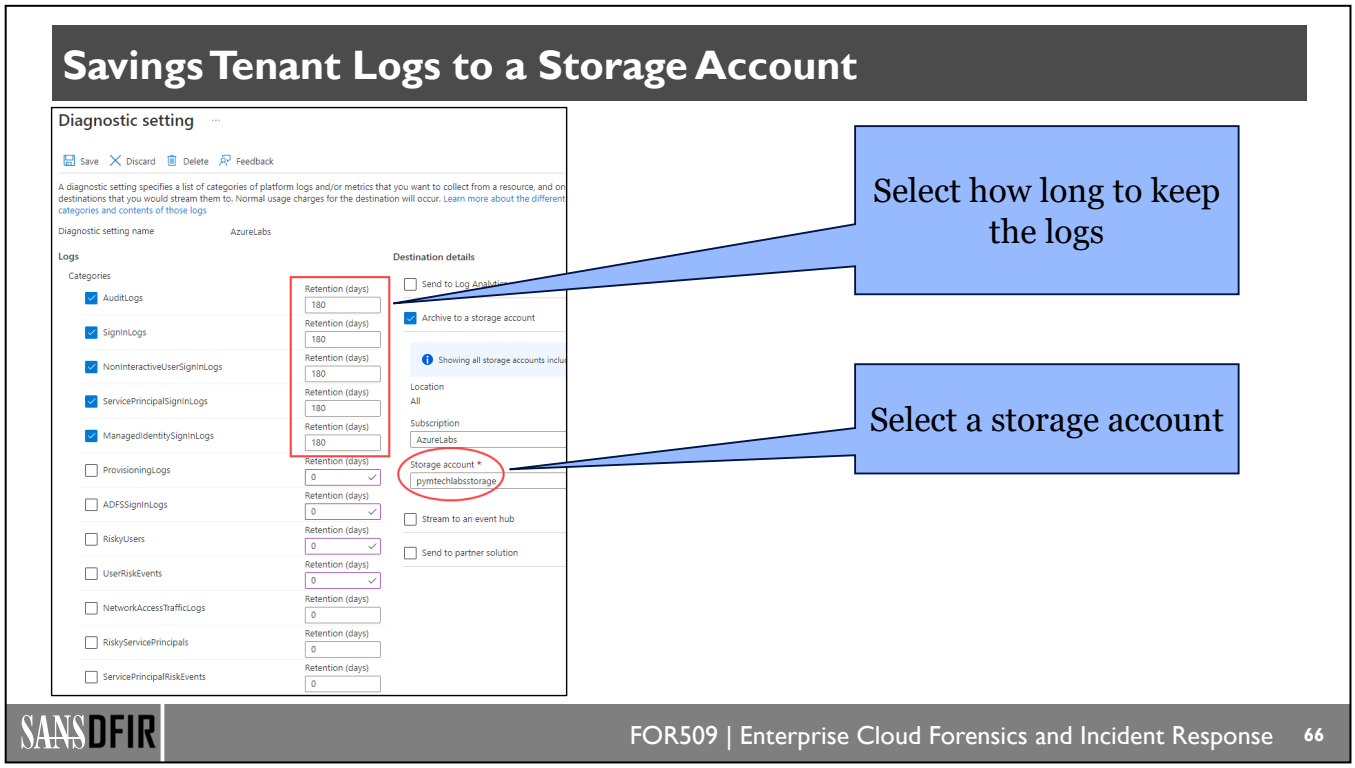
Tenant Logs: Storage Account



- Storage Account Setup
- Azure Storage Explorer
- Blobs
- Import Data to SOF-ELK

This page intentionally left blank.

© 2022 Pierre Lidome



The Log Analytics workspace is a great way to store and view the logs. However, you may want to save the logs for an extended period of time and export them to other tools. To achieve that goal, you will need to export the logs to a storage account.

Going back to the diagnostic setting, we configure the tenant logs to be sent to a storage account called *pymtechlabslogstorage*.

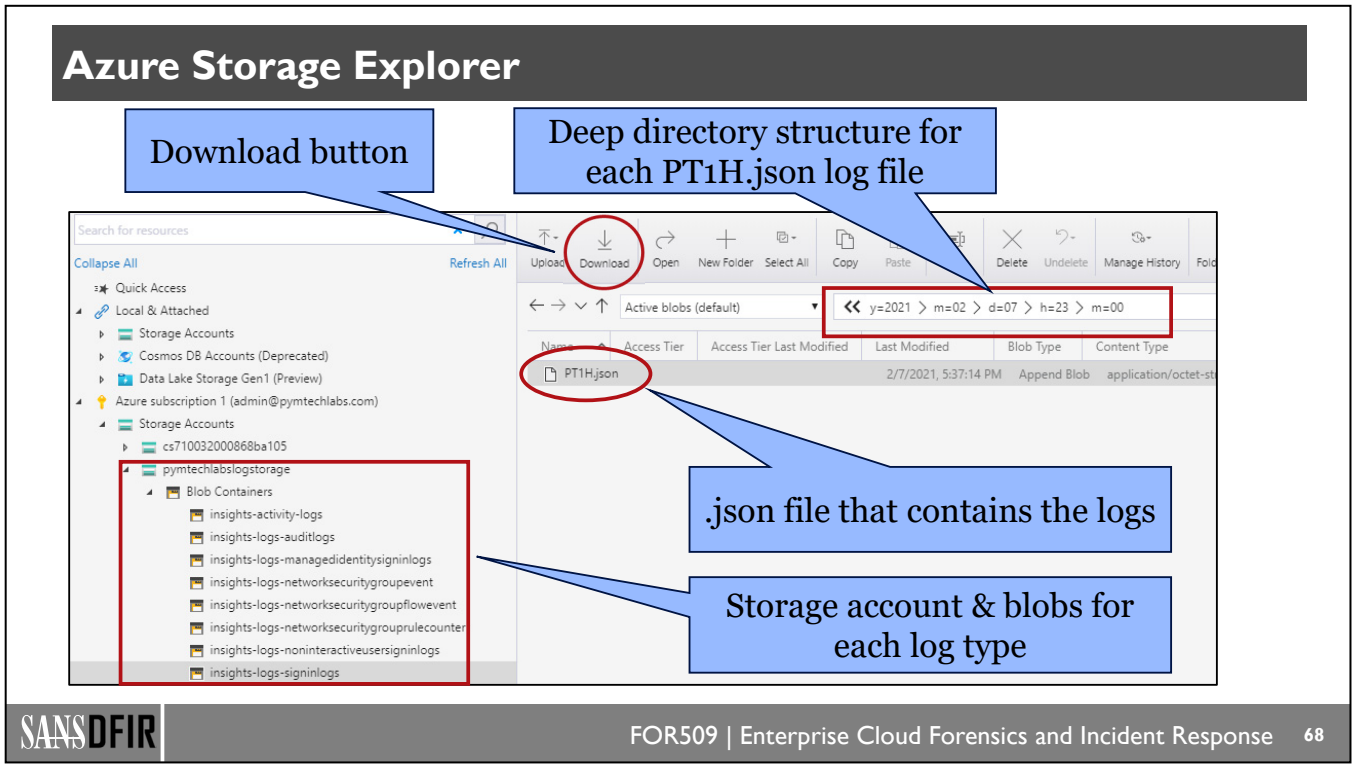
One of the great features of storage accounts is the ability to specify a retention period. As expected, you will be billed based on the quantity of data stored in the storage account. You will need to balance which logs you wish to retain, how long to retain them for, and the cost of the storage. Each organization will have a different answer.

Once the logs are in the storage account, you may access them in different ways. Programmers will use their favorite programming language to access the data via API. An easier way is to use a tool such as Azure Storage Explorer, which provides a GUI to see the storage accounts and the blobs that store the logs.

© 2022 Pierre Lidome

Technet24

© 2022 Pierre Lidome



Azure Storage Explorer is the simplest way to access the blobs within the storage account and download the data. Azure Storage Explorer is a free Microsoft application that you will need to install on your computer.[1]

We will use Azure Storage Explorer to export many kinds of logs. As a preview, here are the names of some of the logs you may encounter. Because there are so many, we are only listing a few here:

Log	**Name**
Tenant	insights-logs-auditlogs
	insights-logs-managedidentitysigninlogs
	insights-logs-noninteractiveusersigninlogs
	insights-logs-signinlogs
Subscription	insights-activity-logs
Network Watcher	insights-logs-networksecuritygroupevent
	insights-logs-networksecuritygrouprulecounter
NSG Flow	insights-logs-networksecuritygroupflowevent

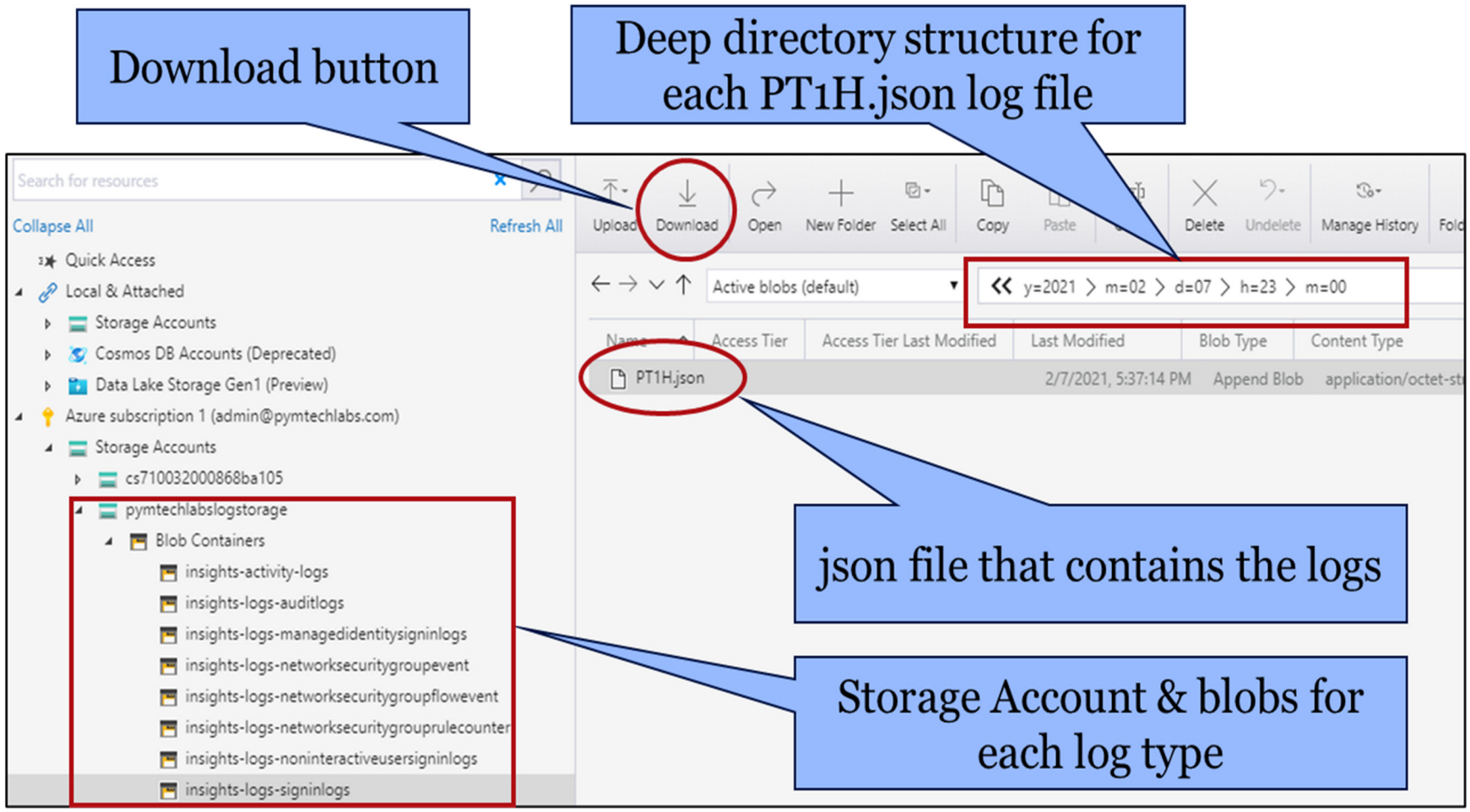
The schema for these logs is documented in the second reference.[2]

Most logs will be stored in the storage account as blobs. Operating system logs will be stored as tables.

1. https://for509.com/storageexplorer
2. https://for509.com/schema-activitylog

© 2022 Pierre Lidome

Technet24



© 2022 Pierre Lidome

Tenant Logs: Storage Blobs

- Separate JSON log files stored for each hour

```
tenantId=7e325eda-7945-46d3-ac99-f0dcfeb4628e\
  y=2021\m=03\d=20\h=15\m=00\PT1H.json
  y=2021\m=03\d=20\h=18\m=00\PT1H.json
  y=2021\m=03\d=26\h=01\m=00\PT1H.json
  y=2021\m=03\d=27\h=11\m=00\PT1H.json
```

- PT1H.json file only created when there is log data
- Minute field is always set to 00

Unfortunately, this is where things get a bit complicated. The logs are stored in a series of files called PT1H.json, broken down under the following hierarchy:

```
tenantId=<tenant id>\
  y=<year>\
    m=<month>\
      d=<day>\
        h=<hour>\
          m=00\
              PT1H.json
```

As you can imagine, this means you could have hundreds or even thousands of files.

© 2022 Pierre Lidome

Technet24

Tenant Logs: Import into SOF-ELK

Steps to import logs into SOF-ELK

1. Download the blob containers
   a) Azure Storage Explorer, then transfer to SOF-ELK VM
   b) Python script directly on SOF-ELK VM (see notes)
2. Combine all PT1H.json files into a single file

```
[elk_user@sof-elk]> find ./insights-logs-auditlogs/ -type f –name PT1H.json ⏎
                    -exec cat {} + |tee insights-logs-auditlogs.json
```

3. Copy file to Azure Logstash folder

```
[elk_user@sof-elk]> cp insights-logs-auditlogs.json /logstash/azure
```

4. Repeat for the different sign-in logs

Import these logs into SOF-ELK by following these steps:

1. Download the blob containers with option a) or b).
   a) Use Azure Storage Explorer and then transfer the logs to the SOF-ELK VM.
   b) Use the download_blobs.py script[1] directly in your SOF-ELK VM.
2. In your SOF-ELK VM, you will now have a large number of PT1H.json files located in a deeply nested directory structure. You will need to combine them into a single file. Assuming that the top directory is called `insights-logs-auditlogs`, run the command:

```
find ./insights-logs-auditlogs/ -type f –name PT1H.json ⏎
          -exec cat {} + |tee insights-logs-auditlogs.json
```

3. You will now have a single file called `insights-logs-auditlogs.json` that combines every PT1H.json file. Copy it to the appropriate Logstash folder:

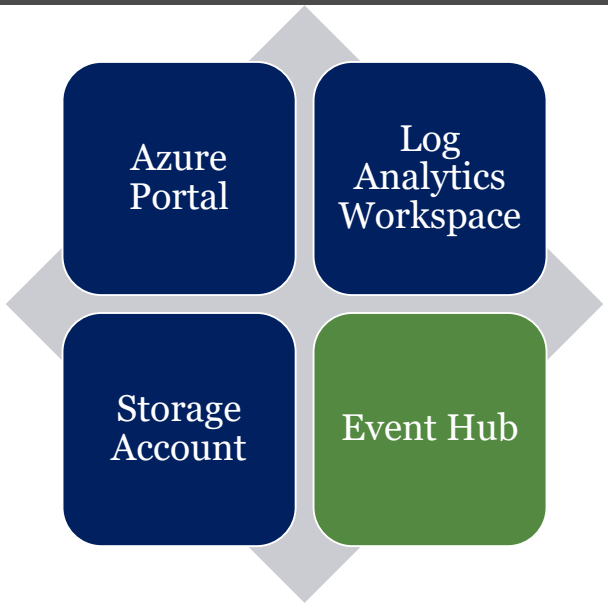
```
cp insights-logs-auditlogs.json /logstash/azure
```

4. ELK will now process the JSON file, and within a few minutes you will be able to see the logs in Kibana.

1. The Python script is courtesy of Quick Programming Tips: https://for509.com/blobpythonscript

© 2022 Pierre Lidome

Tenant Logs: Event Hub



- Event Hubs
- Graph API

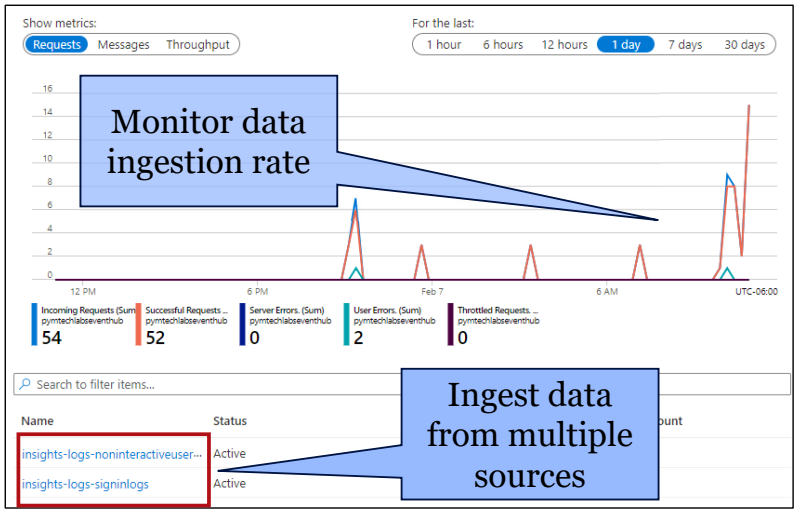
This page intentionally left blank.

© 2022 Pierre Lidome

Technet24

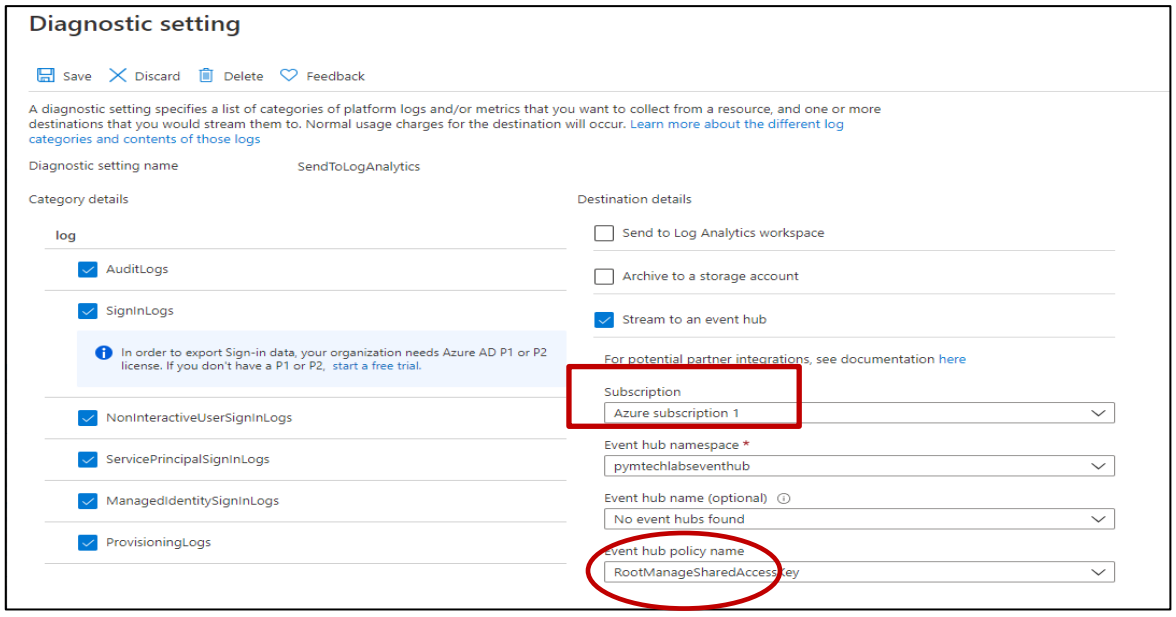
Event Hubs Stream

- Event hubs are real-time data ingestion services
- Capable of ingesting millions of events per second
- Support for multiple sources of data
- Check for their existence



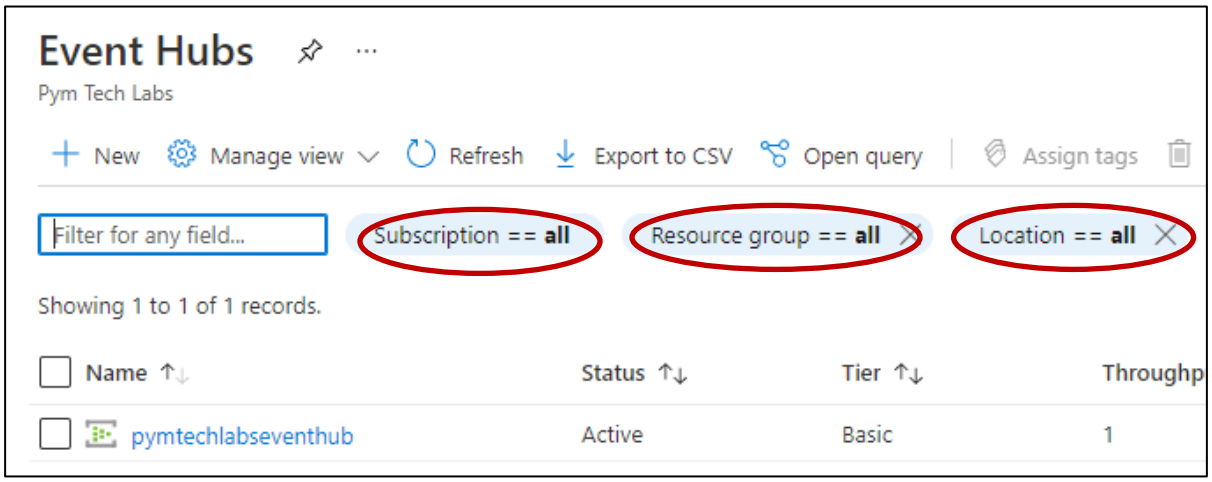
If you want to send your logs to a non-Azure destination, the event hub is a real-time data ingestion service.[1] It's capable of ingesting millions of events per second and creating a dynamic data pipeline to feed an application such as a SIEM.

The configuration to send logs to an event hub follows the same process as the one for the storage account. First, you will need to create the event hub. Second, you will specify the event hub namespace in the diagnostic settings as shown below.[2]



© 2022 Pierre Lidome

Using an event hub is beyond the scope of this course. However, as part of your investigation, it's important to look for such a configuration, as it may point to the existence of a log repository that your client may have forgotten to tell you about.



1. https://for509.com/eventhubs-overview
2. https://for509.com/eventhubs

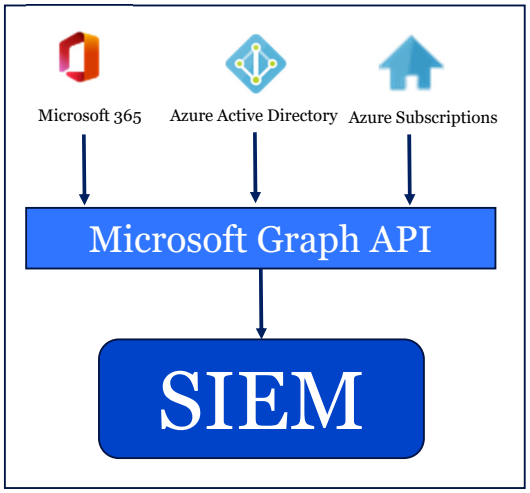
© 2022 Pierre Lidome

Technet24

Graph API

- An alternative method to send logs to an external SIEM is to use the Graph API
- Highly customizable and granular
- Important to discuss with SIEM team which logs they choose to import
- SIEM team should provide name of indexes for each type of logs

Sample architecture



The Microsoft Graph API is a very powerful way to retrieve data from Azure and Microsoft 365. Many programming languages and platform are supported.[1]

Like the event hubs, it's important that you know about this feature so you can communicate with the SIEM team and understand what logs they choose to import.

A company may choose to use event hubs and/or the Graph API, as many different architectures are possible. Because the Graph API pre-dates event hubs, you are very likely to find that many SIEMs obtain their data from Azure in this manner.

1. https://for509.com/graph

© 2022 Pierre Lidome

Lab 2.2 Preview

Lab 2.2 will explore Azure Active Directory logs to search for anomalous activity. In this lab, we will visualize a password spray attack.

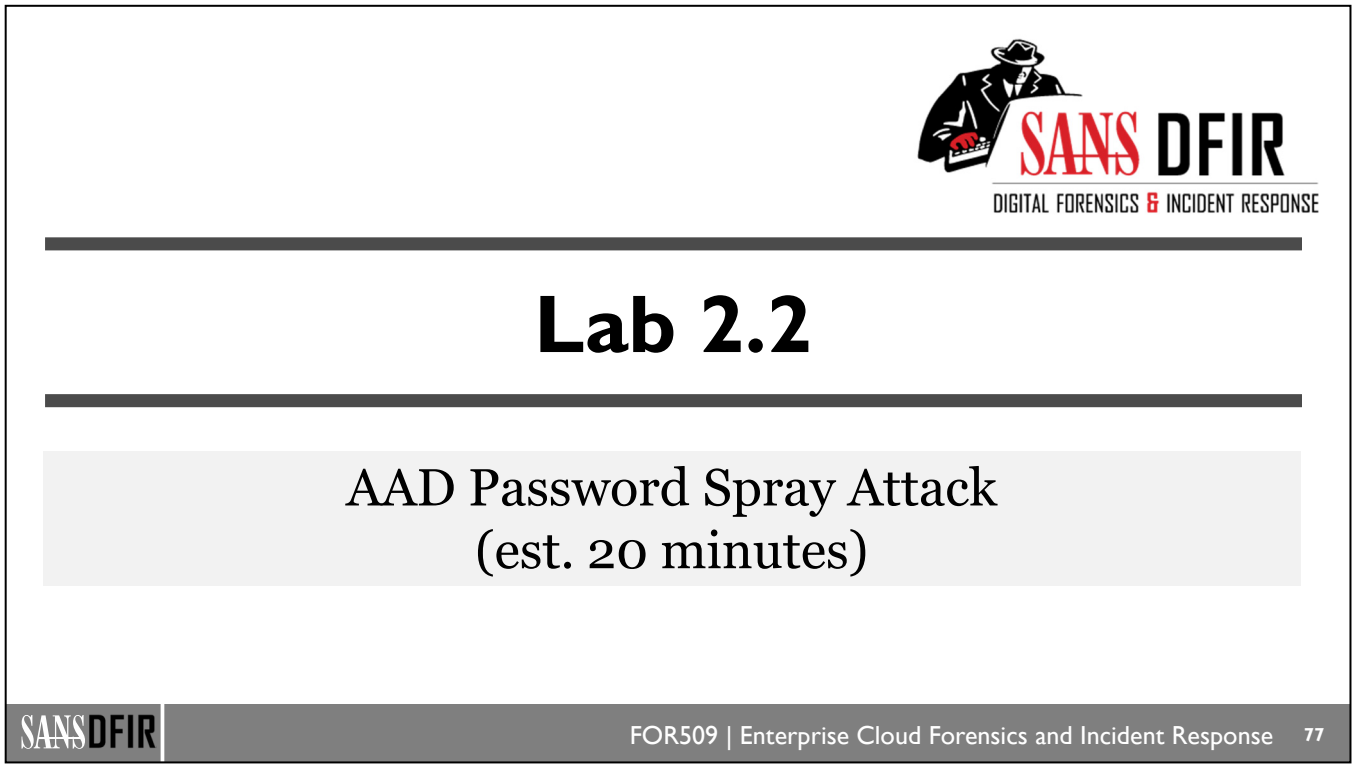
The following fields will be important in this lab:

- `authentication_details.authenticationStepResultDetail`
- `result_description`
- `result_type`
- `user_principal_name`

This page intentionally left blank.

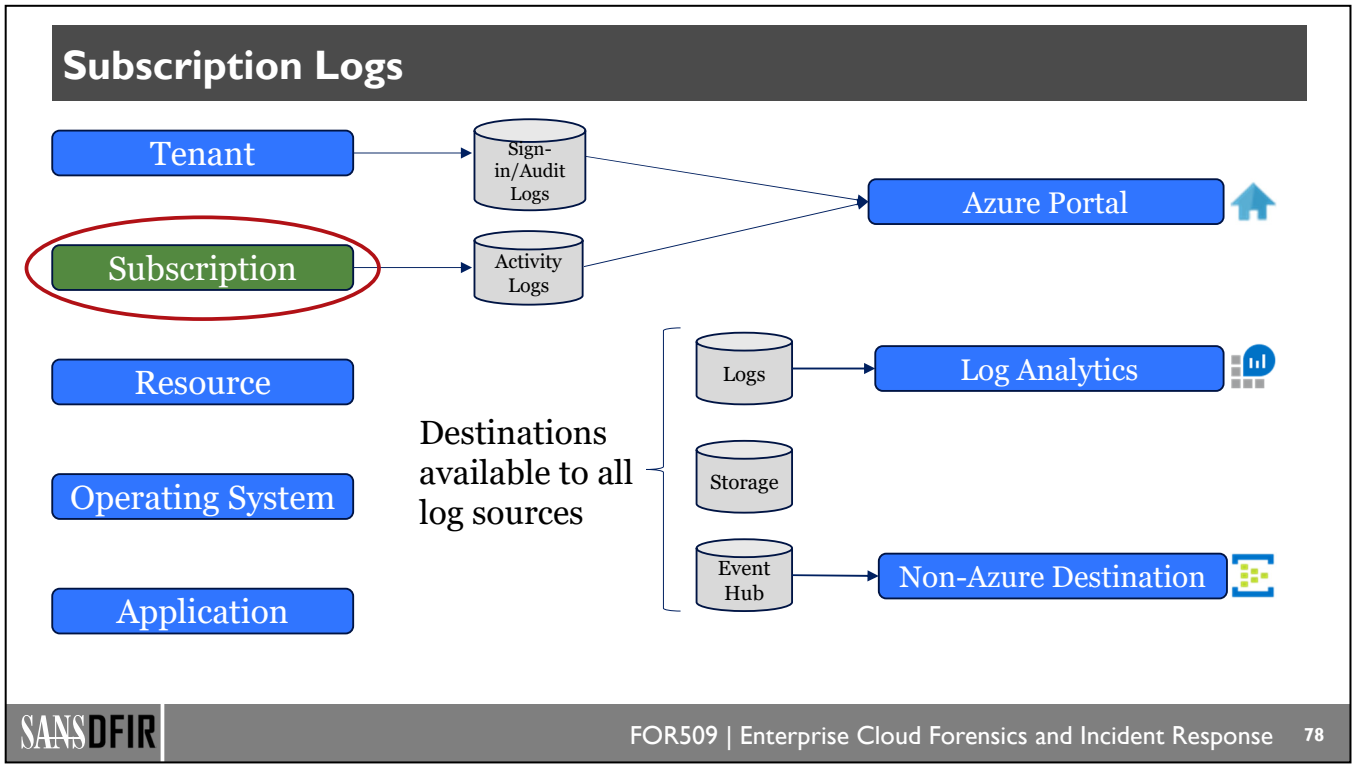
© 2022 Pierre Lidome

Technet24



This page intentionally left blank.

© 2022 Pierre Lidome



This page intentionally left blank.

© 2022 Pierre Lidome

Technet24

Subscription Log Agenda

- Sources of Logs
- Log Analytics Workspace
- Tenant Logs
- Lab 2.2: AAD Password Spray Attack
- Subscription Log
- Lab 2.3: Tracking Resource Creations
- NSG Flow Log
- Lab 2.4: Detecting Data Exfiltration

- The activity log schema
- Viewing the activity log in the portal
- Searching the activity log in a Log Analytics workspace
- Activity log examples
- Sending the activity log to storage or to an event hub
- Importing the activity log into SOF-ELK

The subscription log contains information about operations conducted by tenant-wide services.[1] This is where you will find information about resources being created, modified, or deleted.

You will find the subscription logs under the "Activity log" service.

Just like the tenant logs, there are four actions you can take with these logs:
1. View them directly on the Azure portal.
2. Store them in a Log Analytics workspace.
3. Send them to a storage account for archival.
4. Send them to a SIEM by using the event hub.

1. https://for509.com/activitylog

© 2022 Pierre Lidome

Subscription Log Schema (1)

- Activity log events will be generated for many activities. In this example we are **creating a new VM**.
- The key elements of the log entry are shown on the next slide:
  - `ResourceId` will uniquely identify the VM we created.
  - `OperationName` shows that we created a VM (MICROSOFT.COMPUTE/VIRTUALMACHINES/WRITE).
  - `ResultType` and `ResultSignature` show if we were successful.
  - `CallerIpAddress` shows the IP address of the principal who made the request.
  - `CorrelationId` allows us to track the different sub-operations.
  - `Claims` will show the principal who made the request and its authentication.
- Many additional fields will be present but may be of limited use to our investigation.

The activity log contains valuable information for our investigations. It's important to understand the key fields in that log. In this example, we have a log entry from the creation of a virtual machine. The next slides will show the most important fields common to the various types of entries.

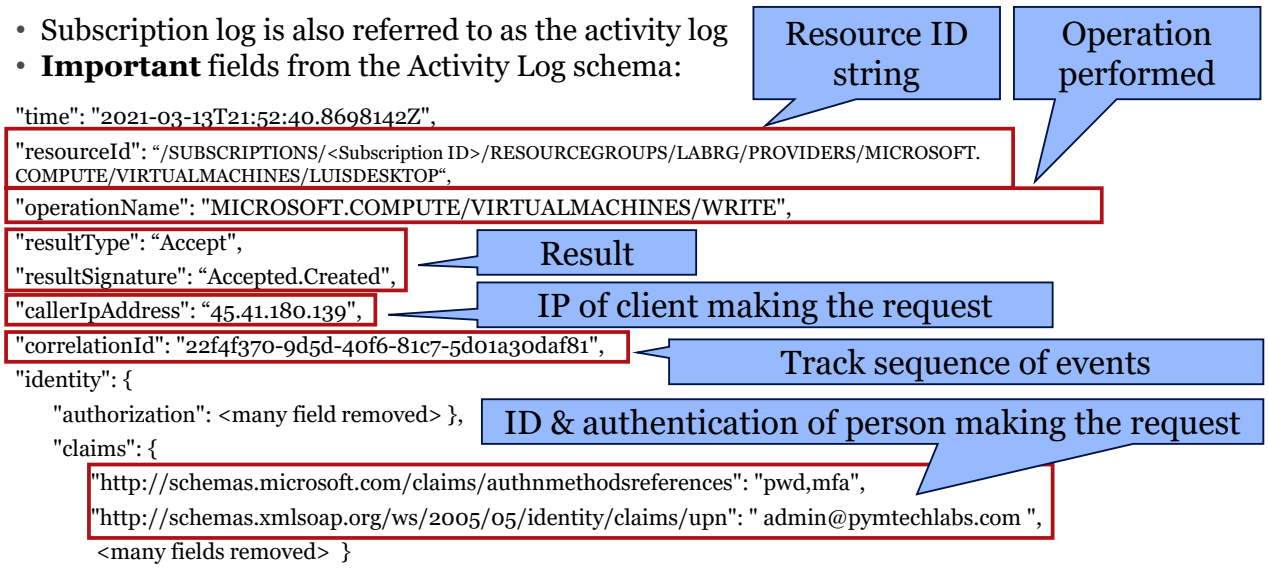
Entries from the various providers (compute, network, etc.) will have the same schema but may populate fields differently. For example, when creating a virtual machine, the compute provider will populate a field called "responseBody" that contains an interesting piece of information, as we will see in Lab 2.3.

© 2022 Pierre Lidome

Technet24

Subscription Log Schema (2)

- Subscription log is also referred to as the activity log
- **Important** fields from the Activity Log schema:



The subscription log is often referred to as the activity log. The complete schema is quite long, and many fields duplicate the same information. In this slide, we are highlighting a few important fields:

- **resourceId** is the string of the resource being added/modified/deleted.
- **operationName** is the name of the provider making the change and the nature of the change. In this example, we are changing a security rule in the network security group.
- **resultType** and **resultSignature** contain nearly identical information with the result of the operation.
- **callerIpAddress** is the IP address of the client making the request.
- **correlationId** is a unique GUID that is used to track the sequence of events that make the operation (example in the next slide).
- **Identity** contains an authorization sub-field and a claims sub-field. In the claims sub-field, you will find information about the person who is making the change.

Notice how the schema has multiple levels of nested fields. This will be a challenge to import in SOF-ELK, as SOF-ELK uses a flat schema. We will discuss later some of the compromises that have to be made.

© 2022 Pierre Lidome

Subscription Log: correlationId

The correlationId field is found in nearly all Azure logs. It's very useful to track the sequence of events that make up an operation

correlation_guid : "22f4f370-9d5d-40f6-81c7-5d01a30daf81"

Time ▲	action	result_signature
> 2021-03-13 14:22:23.974 +00:00	Microsoft.Resources/deployments/write	Started.
> 2021-03-13 14:22:25.434 +00:00	Microsoft.Resources/deployments/write	Accepted.Created
> 2021-03-13 14:22:26.474 +00:00	Microsoft.Network/networkSecurityGroups/write	Started.
> 2021-03-13 14:22:26.474 +00:00	Microsoft.Network/publicIpAddresses/write	Started.
> 2021-03-13 14:22:26.479 +00:00	Microsoft.Compute/register/action	Started.
> 2021-03-13 14:22:26.814 +00:00	Microsoft.Compute/register/action	Succeeded.OK

Using SOF-ELK, we can filter our data to only show the events with the same correlationId:

```
correlation_guid : "22f4f370-9d5d-40f6-81c7-5d01a30daf81"
```

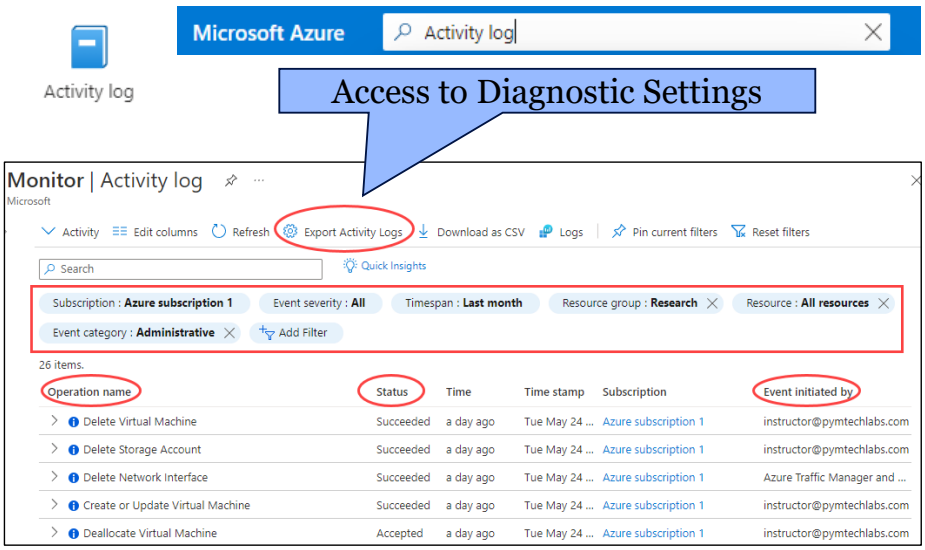
This filter will show us the operations that Azure performed to complete the tasks, which in this case was to create a new VM.

Note: The operations shown above are a small subset of the 28 operations that Azure performed to create a new VM.

© 2022 Pierre Lidome

Technet24

Subscription Log: Portal



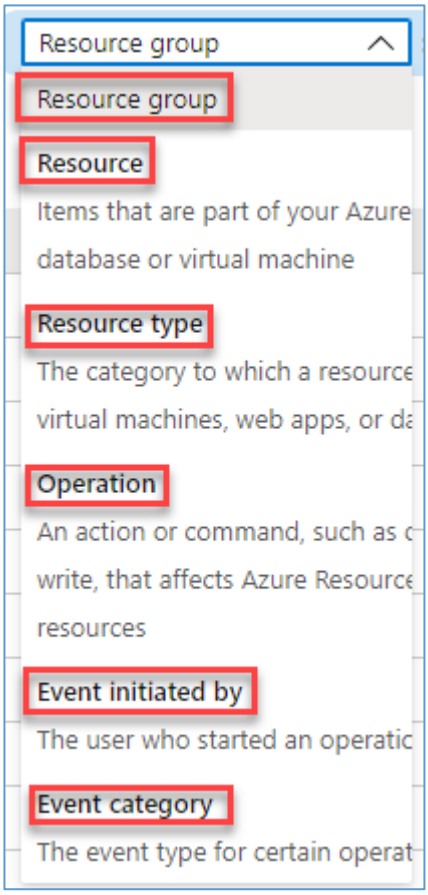
- Search the activity log by specifying the subscription, event severity, and the time frame
- Options for further filtering
- Results include the operation name, status, and userid of person who initiated the event
- Access diagnostic settings via the "Export Activity Logs" menu

Just like the AAD log, the portal is a quick and easy way to get an overview. However, as you will see in the next few slides, it's not a practical way to get detailed information.

In the portal, you need to select:
1. The subscription you wish to query.
2. The event severity: Critical, Error, Warning, or Informational.
3. The timeframe.
4. Optionally, additional filters are available, as shown in the picture on the right.

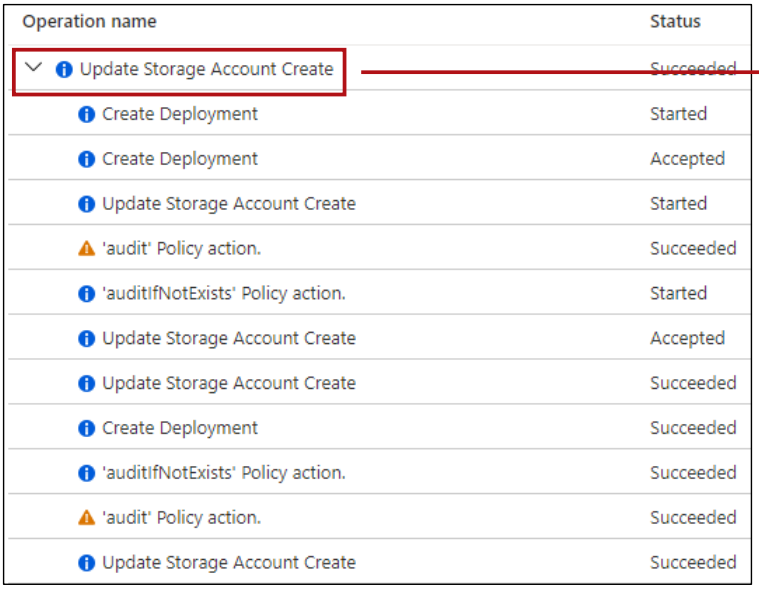
The access to the Diagnostic Settings menu is via the "Export Activity Logs" option on the top menu bar. There is a "Diagnostic Settings" option on the left menu bar (not shown in the slide screenshot); however, it currently doesn't lead to the intended destination.

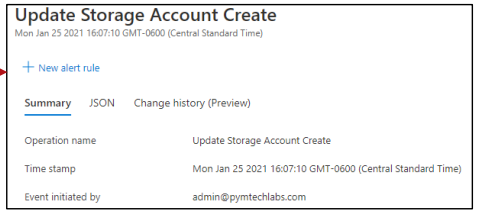
Additional Filters:



© 2022 Pierre Lidome

Subscription Log: Portal Details





- Each operation is a series of steps that are recorded in the log
- Clicking on the name of the operation will provide more details

In this example, we are creating a new storage account. Notice how this single action is made up of numerous smaller steps. To see the smaller steps, click on the arrow itself rather than the text.

If you click on the text of the operation, you will get details about who initiated it and when.

Further, by selecting the JSON tab, you will see a large number of details regarding the operation. Unfortunately, there is no ability to export the JSON from the portal other than copy/paste, which is not practical for a large number of events.

The same data can be queried programmatically using the `Get-AzLog` PowerShell cmdlet.[1] Alternatively, you can use the CLI with the command `az monitor activity-log`.[2]

However, as we saw with the AAD logs, using the Log Analytics workspace is a much better way to visualize the information in the portal.

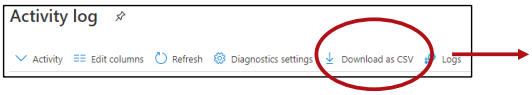
1. https://for509.com/monitor-powershell
2. https://for509.com/monitor-cli

© 2022 Pierre Lidome

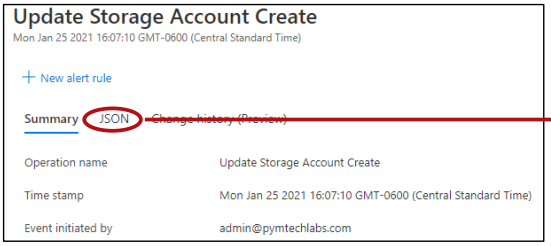
Technet24

Subscription Log: Log Entry Details

- CSV file doesn't include all the details that are part of the schema
- JSON entry has all the details but not useful inside the portal

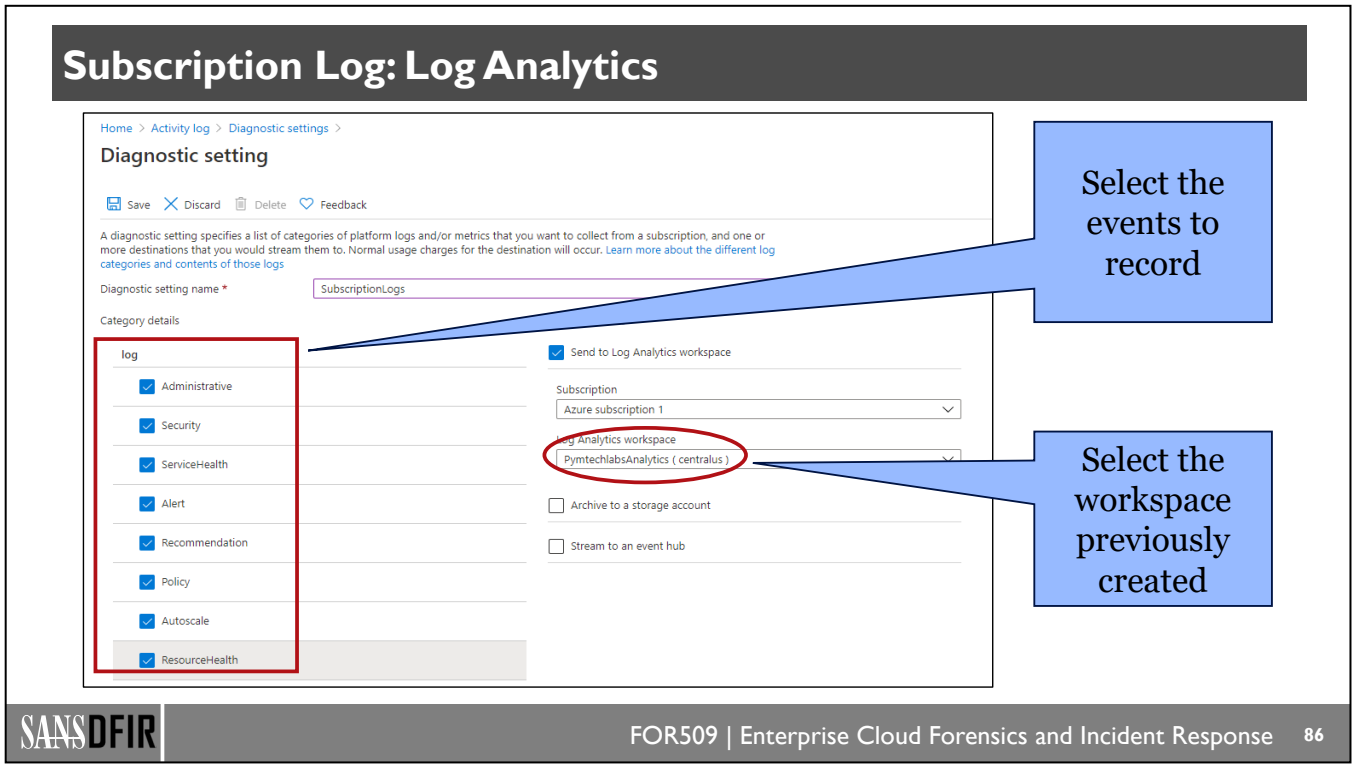


	A	B	C	D	E	F	G	H	I	J	K
1	Correlatio	Operation	Status	Event cate	Level	Time	Subscripti	Event initi	Resource	Resource	Resource
2	ad9c72ee-	Update St	Succeeded	Administr	Informati	2021-01-2	d841fb8e-	admin@p	Microsoft.	FileShare	/subscriptions/
3	ad9c72ee-	Create De	Started	Administr	Informati	2021-01-2	d841fb8e-	admin@p	Microsoft.	FileShare	/subscriptions/
4	ad9c72ee-	Create De	Accepted	Administr	Informati	2021-01-2	d841fb8e-	admin@p	Microsoft.	FileShare	/subscriptions/
5	ad9c72ee-	Update St	Started	Administr	Informati	2021-01-2	d841fb8e-	admin@p	Microsoft.	FileShare	/subscriptions/



You may have noticed that the portal provides an option to export the data into a CSV file. While this appears to be a good solution, upon closer examination you will find that only the high-level information is exported. When you compare the data in the CSV file to the data in the JSON, you will see that all the detailed information is missing. If you need to export the data to a file, we will show you how to write it to a storage account just like we did for the AAD logs.

© 2022 Pierre Lidome



While the portal may be fine for a quick look at the subscription logs, it's much better to set up a Log Analytics workspace. By sending the subscription logs to the same Log Analytics workspace as the AAD log, we can get a more comprehensive view of the events in our Azure tenant.

The setup process is identical to the one for the tenant logs: select the log categories you wish to save and the Log Analytics workspace to send them to.

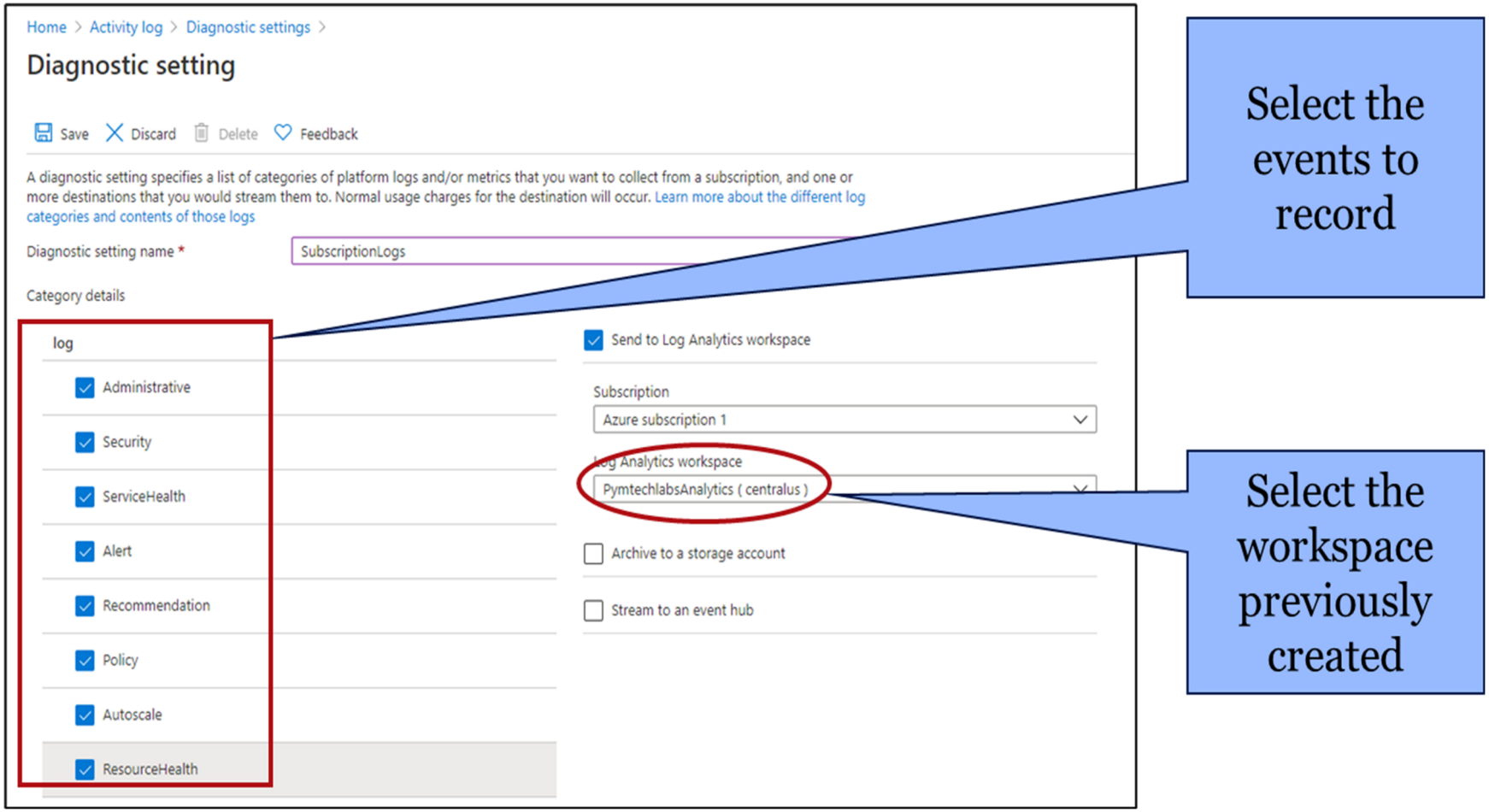
Subscription log categories are:

- **Administrative**: Actions performed through the Resource Manager, such as resource creation, update, and deletion.
- **Security**: Alerts generated by Azure Security Center.
- **Service Health**: Service health incidents such as Azure service downtime.
- **Alert**: Triggering of previously configured alerts, such as CPU utilization exceeding a specific threshold.
- **Recommendation**: Recommendations from the Azure Advisor.
- **Policy**: Policy events such as *audit* and *deny* per policies established in the subscription.
- **Autoscale**: Events related to the operation of the autoscale engine based on settings defined in your subscription.
- **Resource Health**: Events regarding the health of your resource with a possible status of *Available*, *Unavailable*, *Degraded*, or *Unknown*.

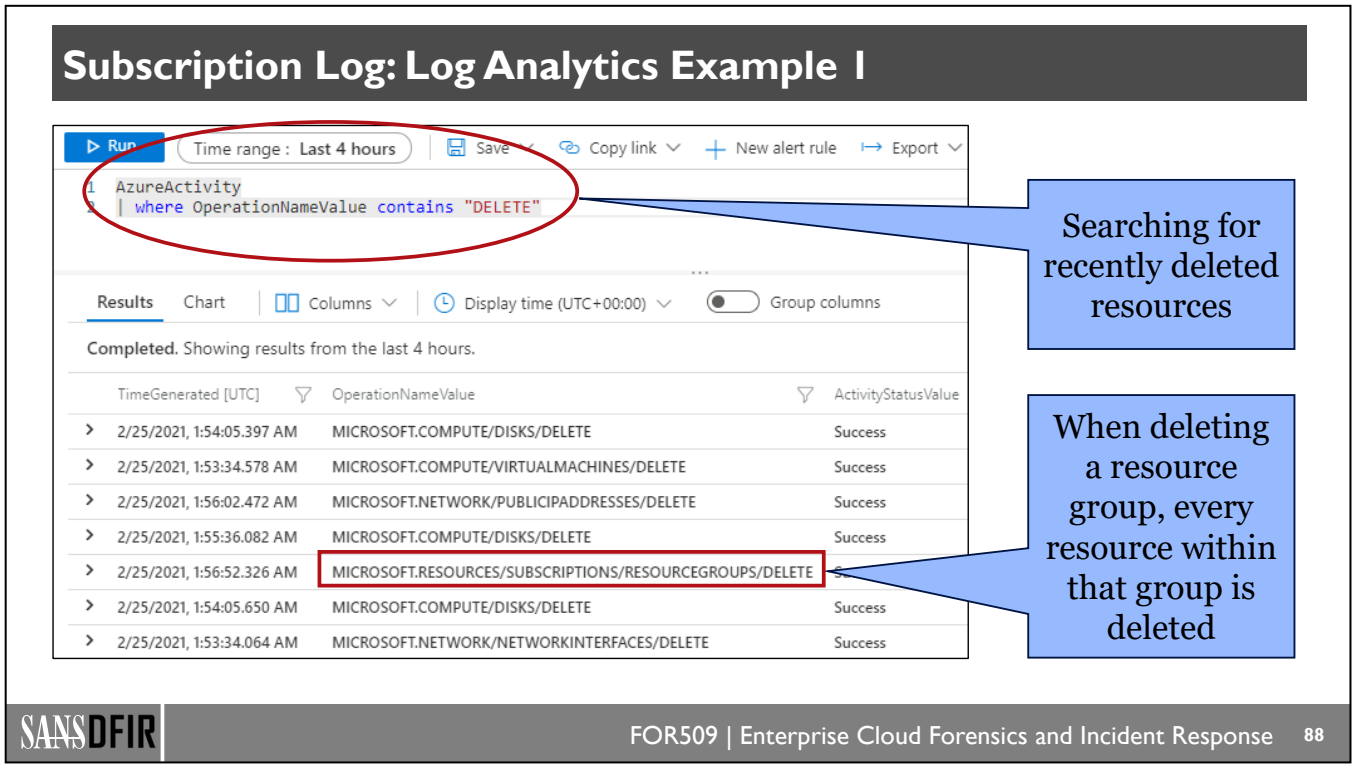
Many of these categories will only show up if the corresponding settings have been enabled in your subscription. From a practical point of view, you will mostly see administrative log entries.

As with the tenant logs, it might be tempting to save everything, but in a large organization the storage requirements may get very costly.

© 2022 Pierre Lidome
Technet24



© 2022 Pierre Lidome



Now that we have configured our Log Analytics workspace, let's look at an example query.

The table that contains the subscription logs is called *AzureActivity*. Consider the scenario where a bad actor deletes a large number of resources to cover their activity. You could write the following query:

```
AzureActivity
| where OperationNameValue contains "DELETE"
```

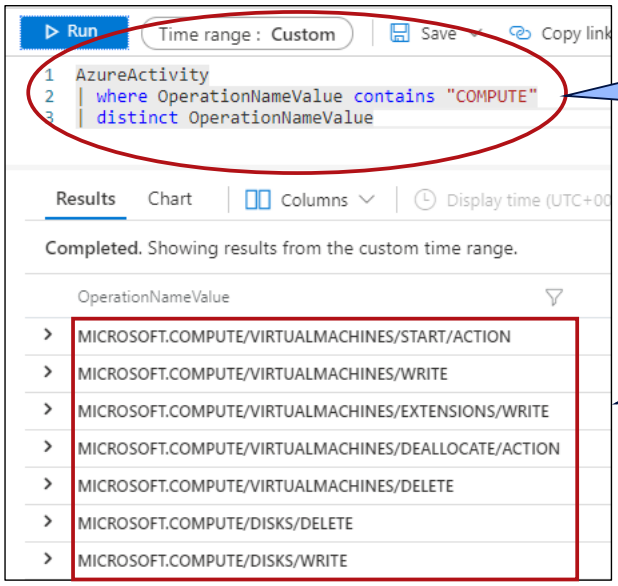
Observe that every element that constitutes a virtual machine is being deleted: the disk, VM itself, IP address, and network interface. This actually doesn't happen if you delete just the VM. The reason everything is being deleted is because the resource group that contains the VM is being deleted.

Deleting a resource group is a very effective way to delete a large number of resources; hence, it's important to carefully consider who has this level of access.

© 2022 Pierre Lidome

Technet24

Subscription Log: Log Analytics Example 2



Search for all operations associated with virtual machines by filtering for the COMPUTE resource provider

The "distinct" statement gives us a summary of the operations performed in the subscription. At this point we don't know how many resources are involved

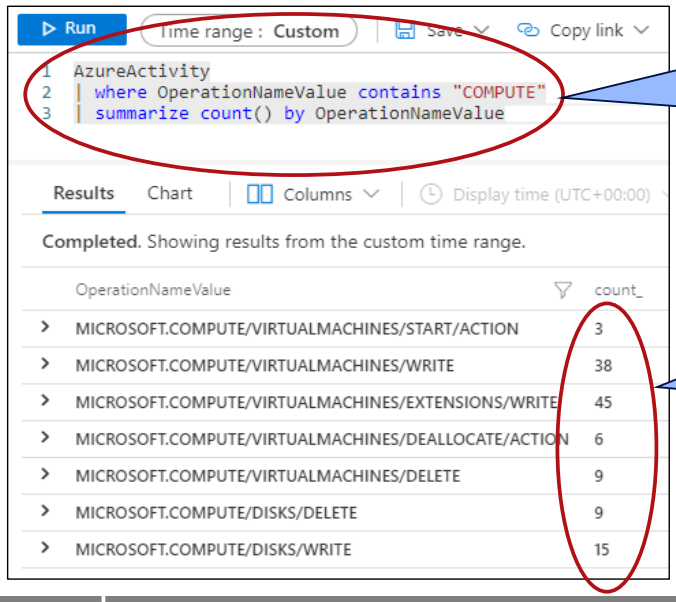
Another interesting query would be to look for operations associated with virtual machines. You may remember that the resource provider responsible for virtual machines is called COMPUTE. So, to find the unique operations performed on virtual machines, you could write the following query:

```
AzureActivity
| where OperationNameValue contains "COMPUTE"
| distinct OperationNameValue
```

The "distinct" statement gives us the unique values. This is a great way to limit the output to a reasonable number of lines and get an overall view of the activity in the subscription.

© 2022 Pierre Lidome

Subscription Log: Log Analytics Example 3



Count the operations associated with the COMPUTE resource provider to get a better idea of the activities in the subscription

The "summarize count() by" statement gives us a count of each operation similar to an Excel pivot table

To understand the scope of the resources being impacted, we can count the number of operations like you would do in an Excel pivot table. The following query will do just that:

```
AzureActivity
| where OperationNameValue contains "COMPUTE"
| summarize count() by OperationNameValue
```

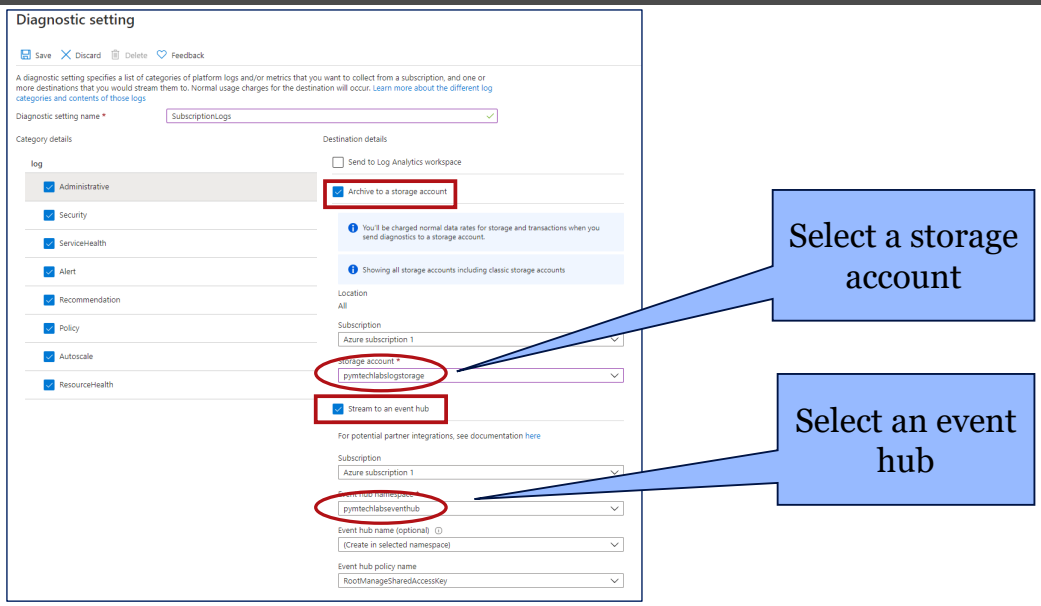
This example takes place over a number of days and shows the owner of the subscription creating, starting, stopping, and eventually deleting a number of virtual machines and associated disks. The output is truncated to fit in the slide.

As you can see, the Log Analytics workspace combined with the KQL is a powerful tool to analyze your log data.

© 2022 Pierre Lidome

Technet24

Subscription Log: Storage Account and Event Hub

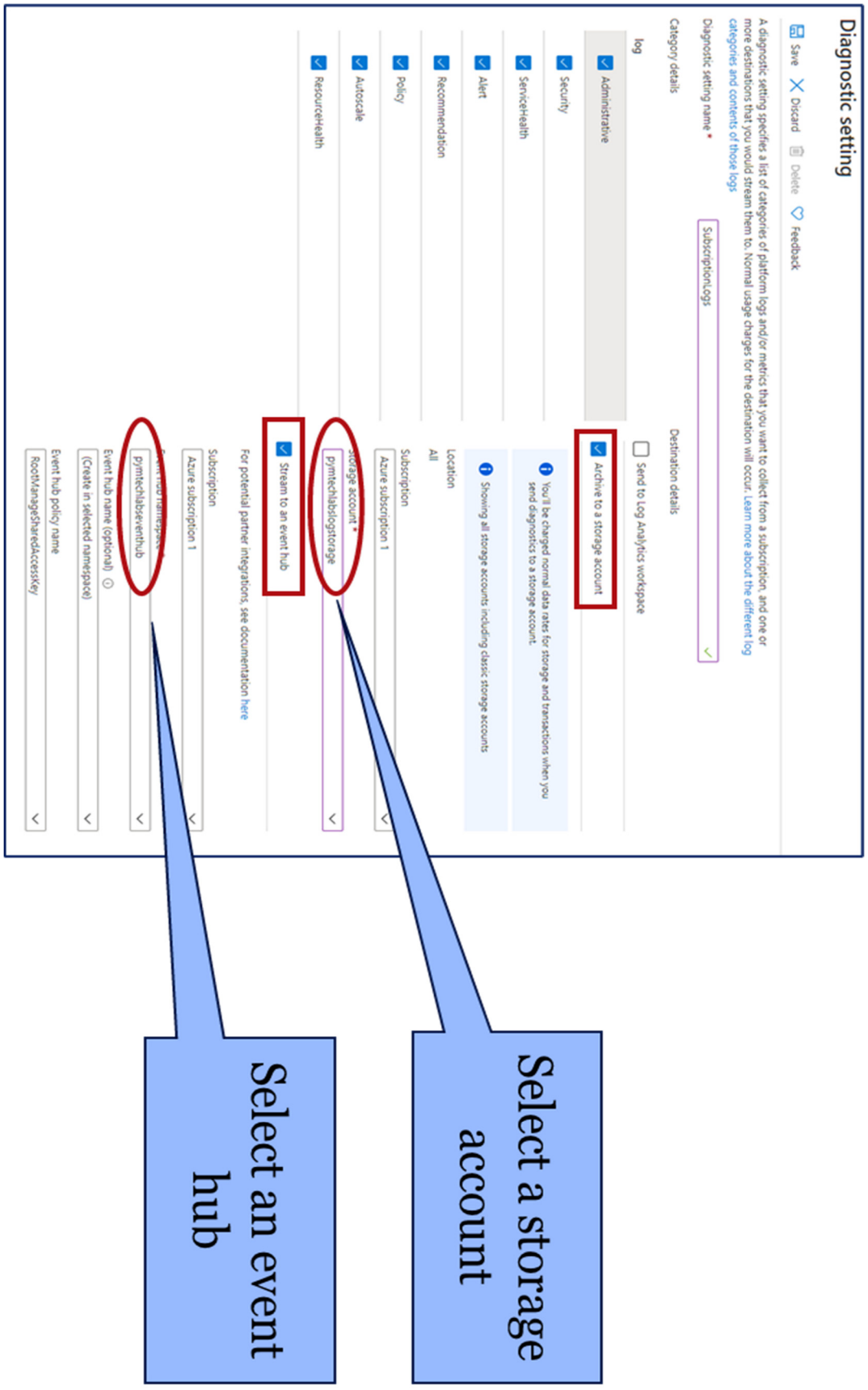


The diagnostic settings are configured the exact same way as tenant logs, specifically Azure Active Directory logs. All you have to do is specify the storage account and the event hub.

In this example, we will store the subscription logs to the storage account called pymtechlabslogstorage and stream the logs to the event hub called pymtechlabseventhub.

The log entries will be found in a blob called insights-activity-logs. As you may remember from one of our earlier slides, a series of files called PT1H.json will be created in a very deep directory structure. By using Azure Storage Explorer, you can download that directory structure and use the scripts provided in the SOF-ELK distribution to combine these files in a single one.

© 2022 Pierre Lidome



© 2022 Pierre Lidome

Technet24

Subscription Log: Import into SOF-ELK

- The process is the same as the tenant logs.
- Due to the flat nature of the schema in SOF-ELK, some of the fields have to be mapped to different names.
- There are too many fields in the activity log schema, so we only mapped some of them. You may choose to create your own mapping.

Activity Log Schema	SOF-ELK
resourceId	resource_id = scope
operationName	operation_name = action
resultType / resultSignature	result_type / result_signature
callerIpAddress	source_ip
correlationId	correlation_guid
Identity / claims	Partially mapped
responseBody	response_body

Some fields are duplicated because they show up in multiple sections of the event entry

You can import the activity log the same way you imported the tenant logs. You should name your file `insights-activity-logs.json` so it's easily identifiable in SOF-ELK.

The nested nature of the activity log file is a challenge when you're mapping fields to import into SOF-ELK's flat structure. As a result, some of the data is imported into two separate fields.

At this time, the identity and claims sections are partially mapped. They contain a lot of information, and we only mapped the fields we believe to be the most important.

The Logstash parser is part of the public SOF-ELK distribution, and everyone is welcome to contribute and improve the mapping.

© 2022 Pierre Lidome

Lab 2.3 Preview

Lab 2.3 will explore the Azure logs to track the creation of cloud resources. In this lab, we will see the creation of GPU-enabled VMs typically created for the purpose of crypto mining.

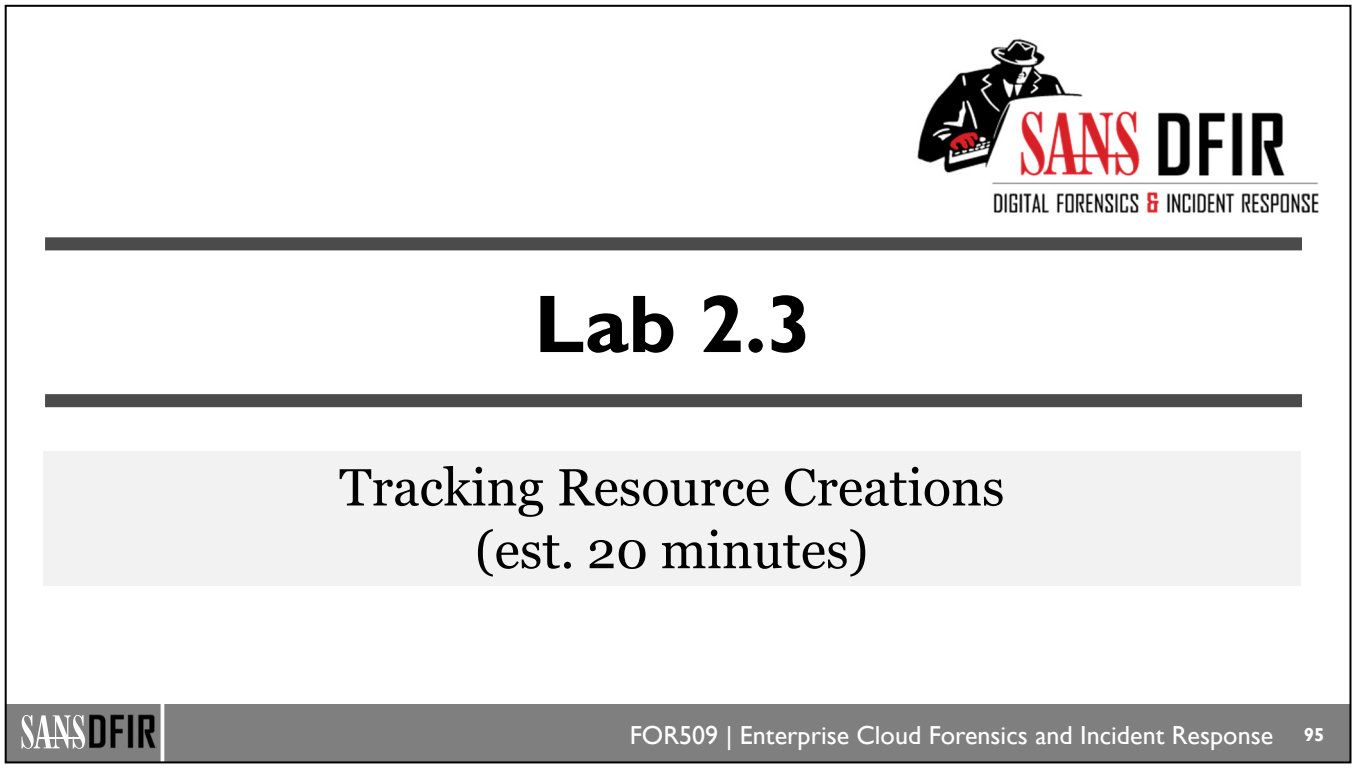
The following fields will be important in this lab:

- `correlation_guid`
- `operation_name`
- `resource_id`
- `response_body`
- `source_ip`
- `user_principal_name`

This page intentionally left blank.

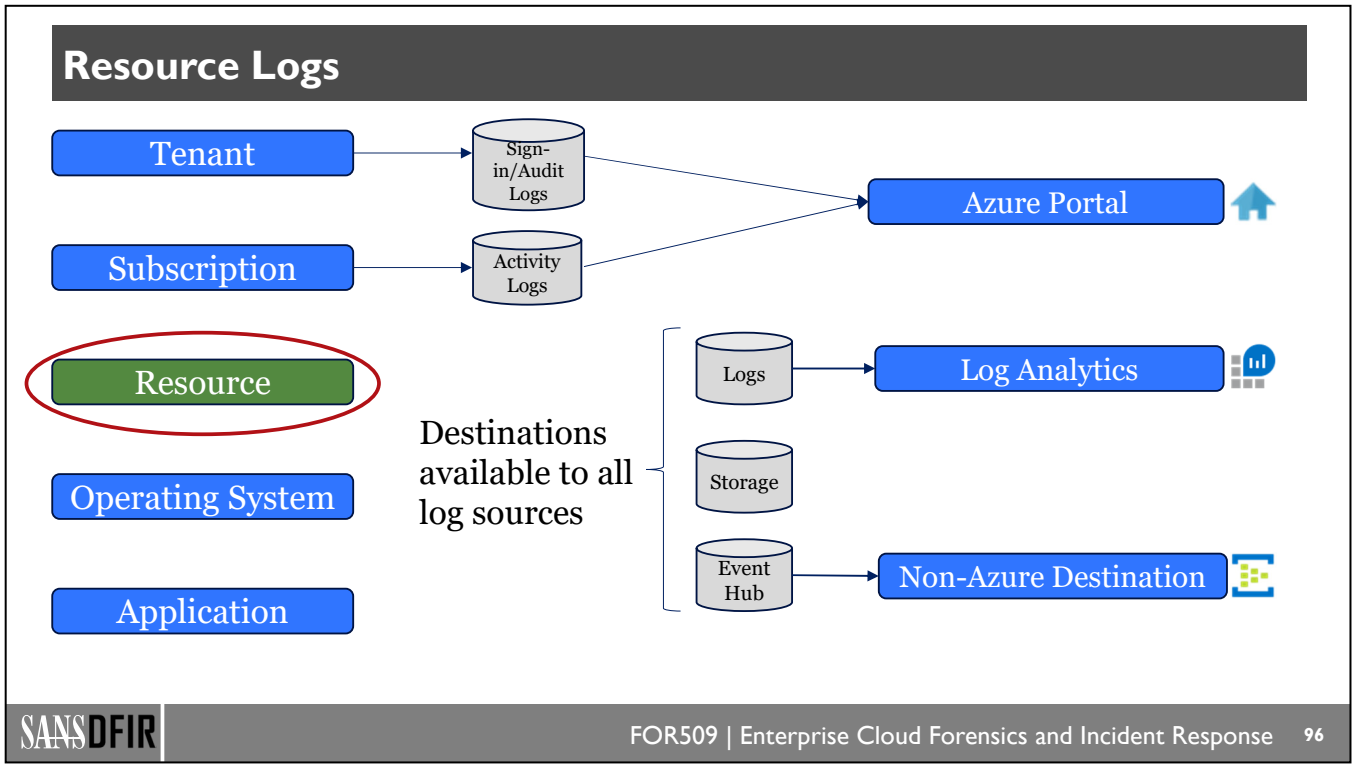
© 2022 Pierre Lidome

Technet24



This page intentionally left blank.

© 2022 Pierre Lidome



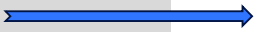
This page intentionally left blank.

© 2022 Pierre Lidome

Technet24

Resource Log Agenda

- Sources of Logs
- Log Analytics Workspace
- Tenant Logs
- Lab 2.2: AAD Password Spray Attack
- Subscription Log
- Lab 2.3: Tracking Resource Creations
- NSG Flow Log
- Storage Account Logs
- Lab 2.4: Detecting Data Exfiltration



- NSG Rules
- NSG Flow Log
- Configuring NSG
- Visualizing Network Traffic
- Importing NSG Logs into SOF-ELK
- Storage Account Logs
- Storage Account Keys
- Data Exfiltration
- Policies

Azure offers a large number of resources. Each one of these resources can generate one or more logs, if configured to do so. This means that there are potentially hundreds of log categories.[1]

For the purposes of incident response and forensics, we will focus on the most important resource log: the network security group (NSG) flow log.

The portal, Log Analytics workspace, storage, and event hub configuration options are the same as the tenant and subscription logs but are not as useful since they only provide metadata about the creation of the flow logs.

Virtual machines are the other key Azure resource. However, we will cover them in the next section, as their logs are more valuable when an agent is installed.

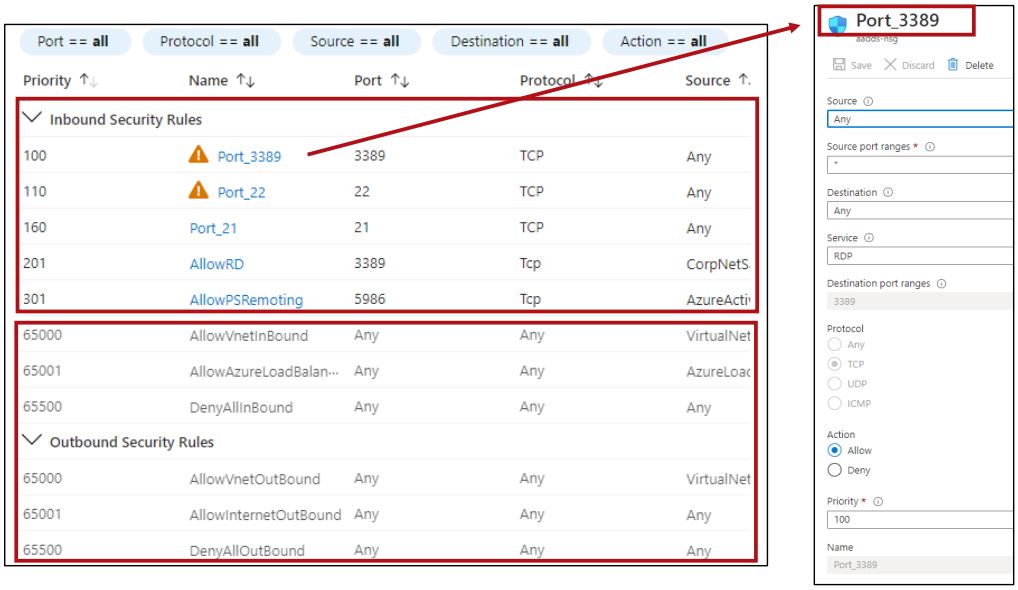
1. https://for509.com/resourcelogscategories

© 2022 Pierre Lidome

NSG Rules

Custom rules

Default rules



Azure network security groups are used to filter network traffic to and from Azure resources. The network security groups contain security rules that allow or deny inbound and/or outbound network traffic to/from Azure resources. For each rule, you can specify source and destination, port, and protocol.

Azure creates several default rules in each network security group. These have a priority of 65000 or higher. Speaking of priorities, rules are evaluated in order from the lowest number to the highest.

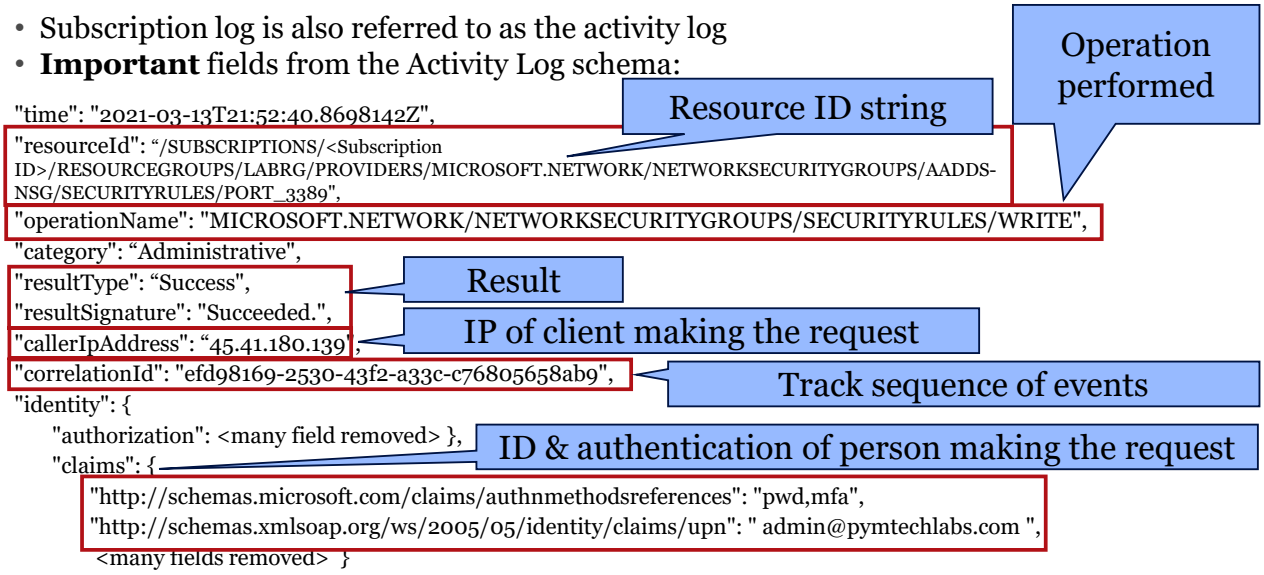
The example in this slide shows rules to open ports 3389 (RDP), 22 (SSH), and 21 (FTP) to the internet. All very bad ideas from a security perspective. In the next slide, we will look at the log entries generated from the creation of the rule for port 3389.

© 2022 Pierre Lidome

Technet24

NSG Rule Creation Log

- Subscription log is also referred to as the activity log
- **Important** fields from the Activity Log schema:



The subscription log is often referred to as the activity log. The complete schema is quite long, and many fields duplicate the same information. In this slide, we are highlighting a few important fields:

- **resourceId** is the string of the resource being added/modified/deleted.
- **operationName** is the name of the provider making the change and the nature of the change. In this example, we are changing a security rule in the network security group.
- **resultType** and **resultSignature** contain nearly identical information with the result of the operation.
- **callerIpAddress** is the IP address of the client making the request.
- **correlationId** is a unique GUID that is used to track the sequence of events that make the operation (example in the next slide).
- **Identity** contains an authorization sub-field and a claims sub-field. In the claims sub-field, you will find information about the person who is making the change.

Notice how the schema has multiple levels of nested fields. This will be a challenge to import in SOF-ELK, as SOF-ELK uses a flat schema. We will discuss later some of the compromises that have to be made.

© 2022 Pierre Lidome

NSG Rule Details

The details of the NSG rule is contained in the log: Port 3389 opened to all incoming IP addresses.


```
"requestbody":
  "properties":
        "description":"RDP port",
        "protocol":"*",
        "sourcePortRanges":null,
        "sourcePortRange":"*",
        "sourceAddressPrefix":"*",
        "destinationPortRanges":null,
        "destinationPortRange":"3389",
        "destinationAddressPrefix":"*",
        "access":"Allow",
        "priority":100,
        "direction":"Inbound",
        "id":"/subscriptions/beb0c5fa-418f-4240-aa63-
ff8cad3f1e1b/resourceGroups/labRG/providers/Microsoft.Network/networkSecurityGroups/aadds-
nsg/securityRules/Port_3389",
        "name":"Port_3389"
```


The log event showing the creation of the NSG rule will also list the details of the rule. In this case port 3389 is opened to the internet without any limitations.

Depending on your SIEM, this part of the log entry may or may not be included. SOF-ELK doesn't show this part of the log entry and you will need to look at the raw log.

© 2022 Pierre Lidome

Technet24

NSG Logs

"category": "NetworkSecurityGroupEvent",
"resourceId": "/SUBSCRIPTIONS/BEB0C5FA-418F-4240-AA63-FF8CAD3F1E1B/RESOURCEGROUPS/LABRG/PROVIDERS/MICROSOFT.NETWORK/NETWORKSECURITYGROUPS/AADDS-NSG",
"operationName": "NetworkSecurityGroupEvents",
"properties": {
"vnetResourceGuid": "{7CD158A6-B80F-40C4-BECF-BAD22571ED77}",
"subnetPrefix": "10.1.0.0/24",
"macAddress": "00-0D-3A-EC-A7-0B",
"primaryIPv4Address": "10.1.0.5",
"ruleName": "UserRule_Port_3389",
"direction": "In",
"priority": 100,
"type": "allow",
"conditions": {
"protocols": "6",
"sourcePortRange": "0-65535",
"destinationPortRange": "3389-3389",
"sourceIP": "0.0.0.0/0,0.0.0.0/0",
"destinationIP": "0.0.0.0/0,0.0.0.0/0"

Each log event shows the rule and whether the traffic was "allow(ed)" or "block(ed)"

The log doesn't capture information about the source IP

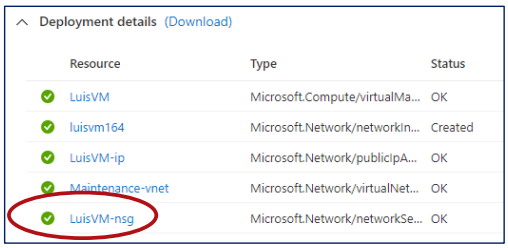
While there is the option to capture NSG logs to observe the activity of the NSG rules, that log has a major flaw. It doesn't capture the source IP of the traffic. Perhaps, there is more value in this log if you wish to assess the effectiveness of specific rules. For DFIR purposes, we will instead focus on NSG flow logs.

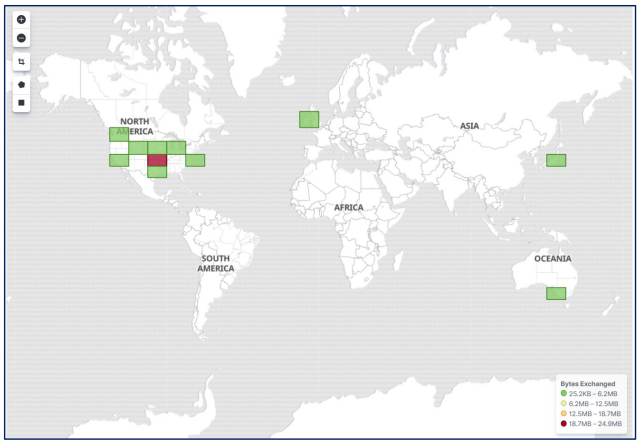
© 2022 Pierre Lidome

NSG Flow Log

- Network security group automatically created with every VM
- Enable log capture in Azure Network Watcher

- Import NSG logs in SIEM and create interesting visualizations





Now that we have created NSG rules, we can record the flow logs.

As previously mentioned, Azure automatically creates a network security group (NSG) when you create a virtual machine: *<name of machine>-nsg*.

These logs are part of the Azure Network Watcher[1] and have the following characteristics:
- They are captured at the transport layer (layer 4).
- They are collected at one-minute intervals.
- They are written in JSON format (like all other Azure logs).
- Each log record will include the network interface, 5-tuple information, and traffic decision (Allow or Deny).
- They are retained for 1 year.

The example graph shows the traffic observed after a VM was created in Azure and left running for 1 hour doing "nothing." The port for remote desktop (RDP) was immediately scanned for potential vulnerabilities. Clearly, setting up strong rules in your NSG is a must.

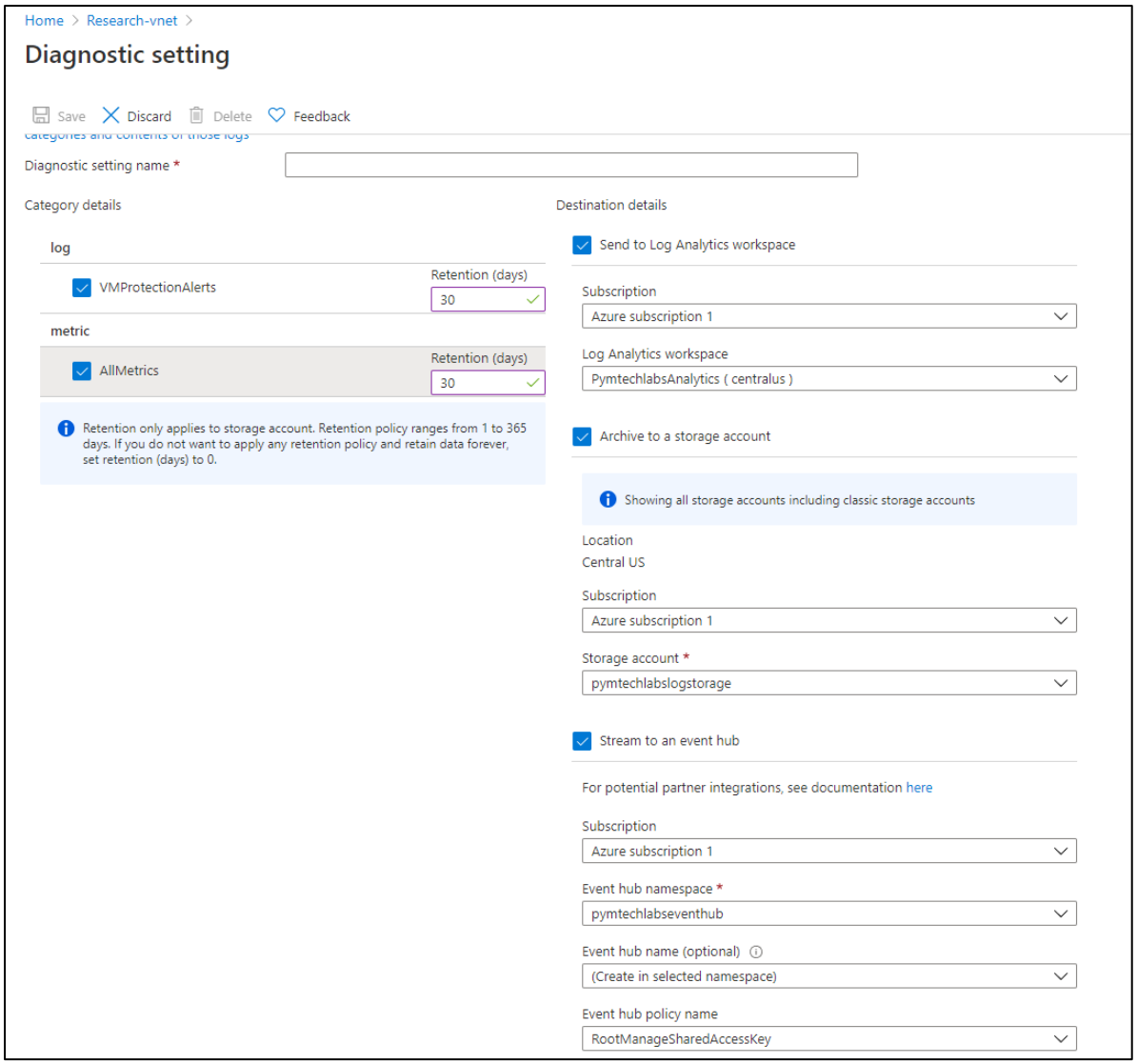
Flow logs are the source of truth for all network activity in your cloud environment and are a "must have" for any investigation.

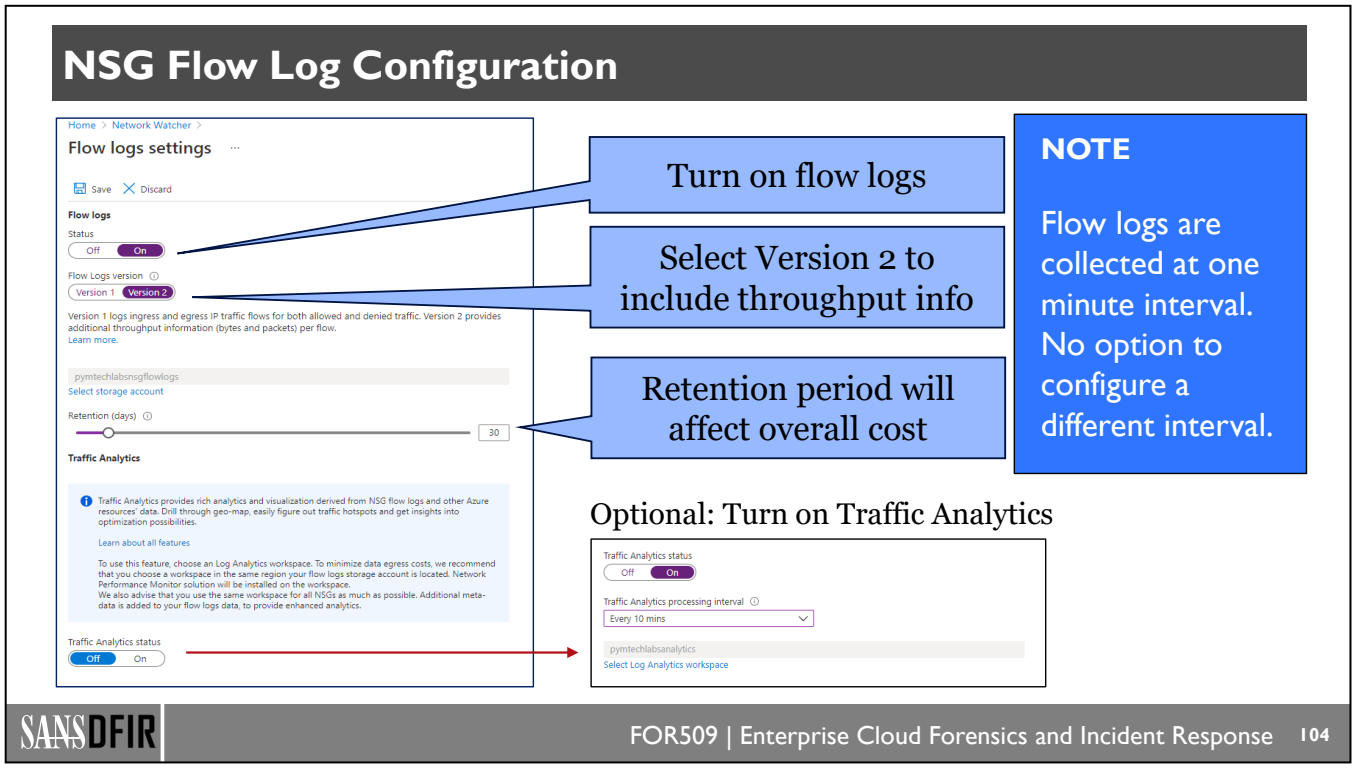
1. https://for509.com/nsgflowlogs

© 2022 Pierre Lidome

Technet24

Logs can also be obtained from the virtual network (VNet). However, they are not as valuable as the NSG flow logs. If you wish to capture these logs, you will configure them from the diagnostic settings for that specific VNet (Research-vnet in the example below):





There are three steps to setting up an NSG flow log:[1]

1. Enable Network Watcher. Network Watcher needs to be enabled for each region. If you have VMs in multiple regions, don't forget to enable it for every region.
2. Register Insights provider. Microsoft.Insights is the provider that enables the login and, as such, needs to be registered, as shown in the first reference. This needs to be done for every subscription.
3. Enable the NSG flow log. Version 2 is strongly recommended, as it captures throughput information.

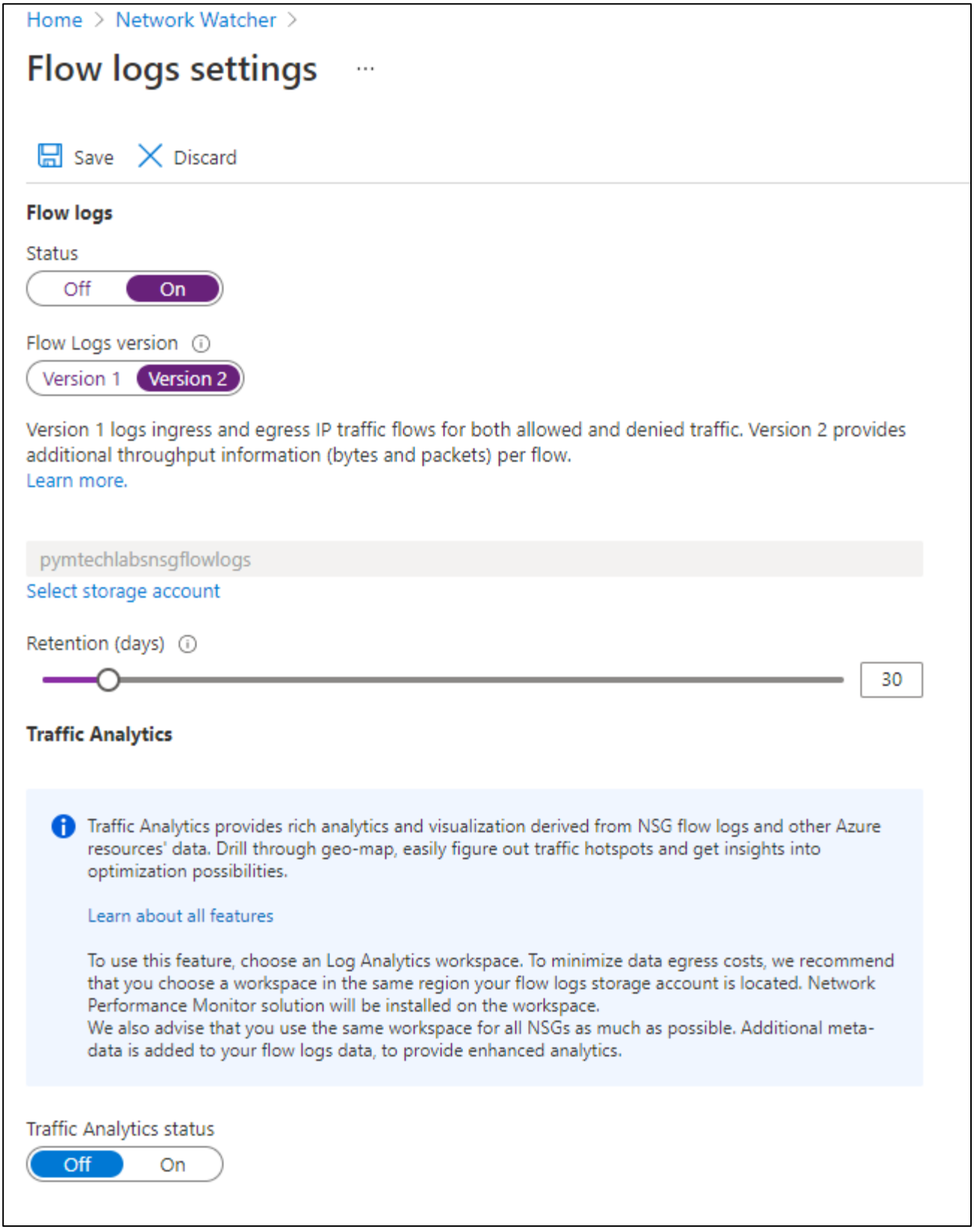
Once the NSG flow log is enabled, you have two choices to consume the information:

1. Export the data from Azure and import it to a SIEM.
2. Enable traffic analytics[2] and visualize the data in your Azure Log Analytics workspace.

1. https://for509.com/nsglogsetup
2. https://for509.com/trafficanalytics

© 2022 Pierre Lidome

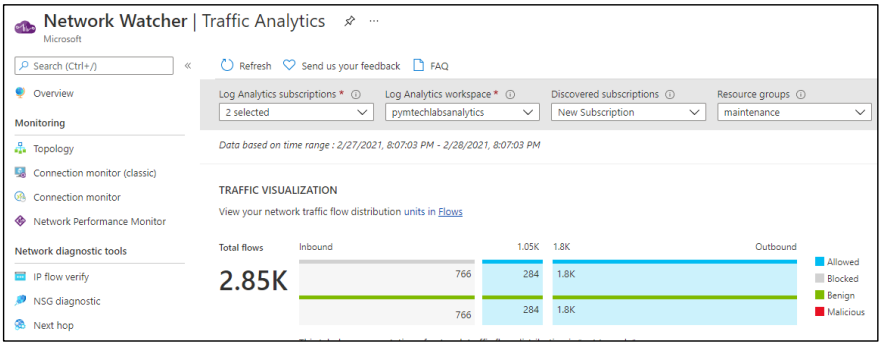
Technet24



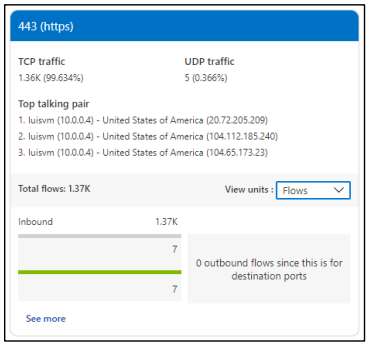
© 2022 Pierre Lidome

Traffic Analytics

Visualization from Traffic Analytics



Protocol data



Look for unusual traffic surges and large amount of blocked traffic

Traffic Analytics leverages the Log Analytics workspace to provide insights into traffic flow. As stated by Microsoft, Traffic Analytics[1] enables you to:

1. Visualize network activity across your Azure subscriptions
2. Identify security threats
3. Understand traffic flow patterns and optimize your network deployment
4. Pinpoint network misconfiguration

From the incident response and forensic perspective, Traffic Analytics is a great way to quickly identify where undesirable traffic may be coming from. While many organizations will prefer to visualize this information in their SIEM, it's good to know that Azure offers an in-cloud solution that is very simple to set up.

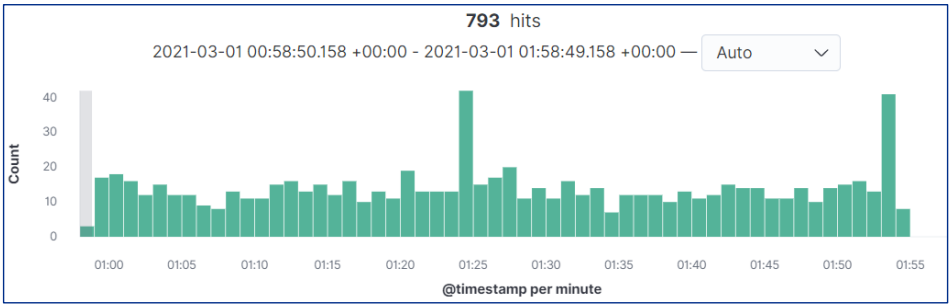
1. https://for509.com/trafficanalytics

© 2022 Pierre Lidome

Technet24

NSG Flow Log: Import into SOF-ELK

- `[elk_user@sof-elk]$` **`python3 ./download_blobs_multithreaded.py`** `(script provided in the notes & class github)`
- `[elk_user@sof-elk]$` **`/usr/local/sof-elk/supporting-scripts/azure-vpcflow2sof-elk.py -r /directory_to_PT1H.json_files -w /logstash/nfarch/outputfile.csv`**



NOTE

Traffic surges are easily identifiable once the NSG log is imported into SOF-ELK

Importing the NSG flow log to SOF-ELK is a three-step process:

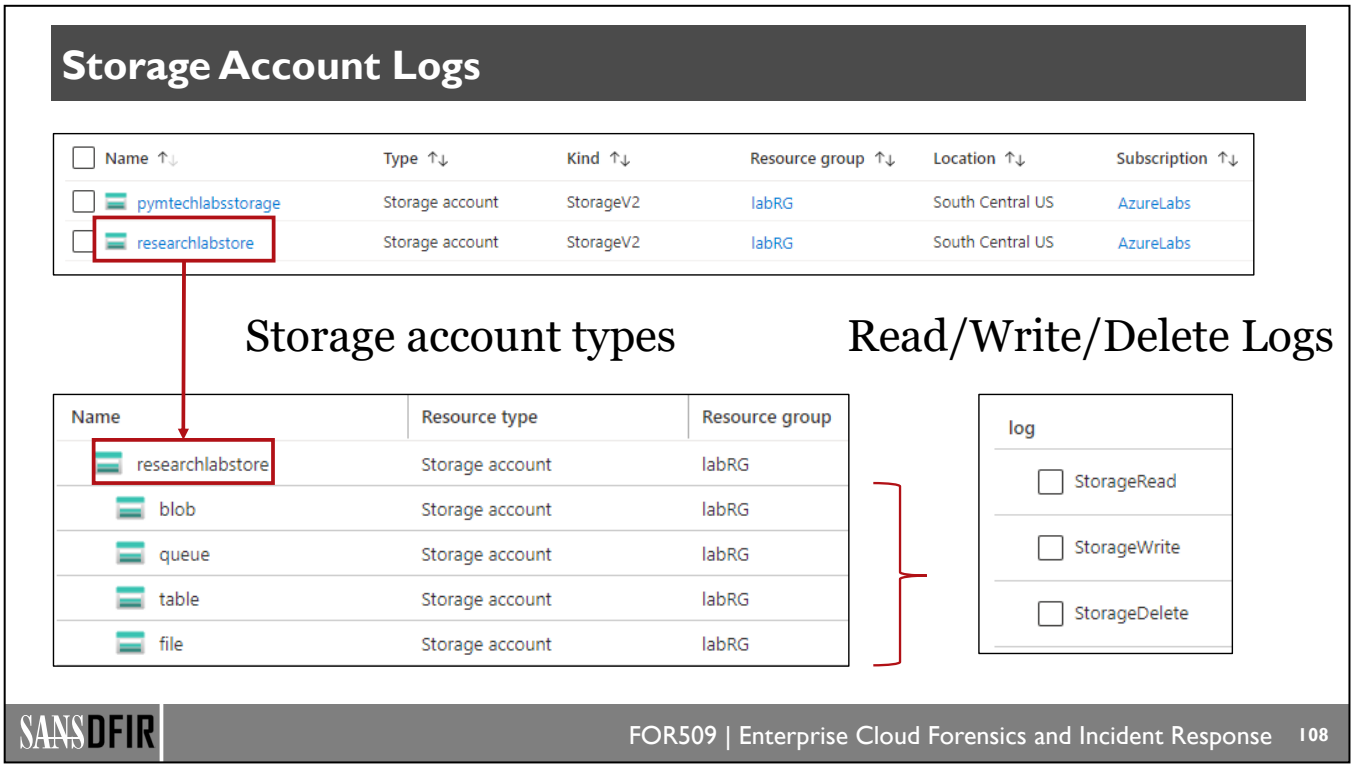
1. Transfer the NSF flow log to your SOF-ELK VM using either Azure Storage Explorer or a Python script.
2. Convert the NSF flow log using azure-vpcflow2sof-elk.py.
3. Copy the file to the `/logstash/nfarch` directory (combined with step 2 in slide example).

Once all the steps are successfully completed, the NSG flow log will be visible in SOF-ELK. In this example, we imported 793 events.

The Python script to download the Azure blobs is available in the reference.

Reference:
https://github.com/dlcowen/sansfor509/blob/main/Azure/download_blobs_multithreaded.py

© 2022 Pierre Lidome



Storage accounts are key resources in Azure. However, logs are not automatically enabled. To enable logs, Azure offers two options:

1. Diagnostic settings – preview
2. Diagnostic settings – classic

The "Diagnostic settings – preview" offers the best logging options. In that menu, logs can be individually configured for each type of data storage: blob, queue, table, and file. In addition, the logs can be sent to a Log Analytics workspace, a different storage account, an event hub, or a partner solution. These are the same options we previously saw for the tenant and subscription logs.

Storage account logs can be individually set to record `Read`, `Write`, or `Delete` operations. Depending on your goals, you may want to be very selective, as these logs can be extremely noisy.

If the StorageRead log isn't enabled, there is no evidence of data being downloaded from the storage accounts, making it an ideal data exfiltration method.

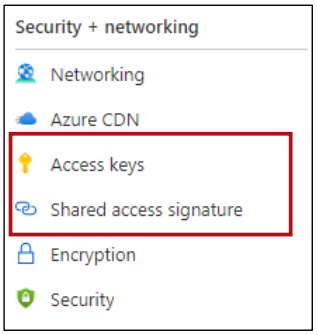
The Microsoft Threat matrix for storage services[1] maps the risks associated with storage according to the MITRE ATT&ACK® framework. One of these risks is data exfiltration. To track data exfiltration, you would want to record `Read` operations and filter the logs for the `GetBlob` operation, as we will see in the next lab.

1. https://for509.com/storage-threat-matrix

© 2022 Pierre Lidome
Technet24

Storage Account Access

- Access credentials options:
  - Shared access signature (SAS)
  - Access key
  - Configured for public access
- Access can be configured
  - At the storage account level
  - At the data storage level
    - Containers (blob)
    - File shares
    - Queues
    - Tables
- SAS can be generated for a specific blob



Access to storage accounts can be configured at different levels:
1. Storage account
2. Data storage: container, file shares, queues, or tables
3. Blob level

Storage accounts have two access keys associated with them. Azure provides two access keys so that you can maintain connections using one key while regenerating the other. Regularly rotating access keys is highly recommended.

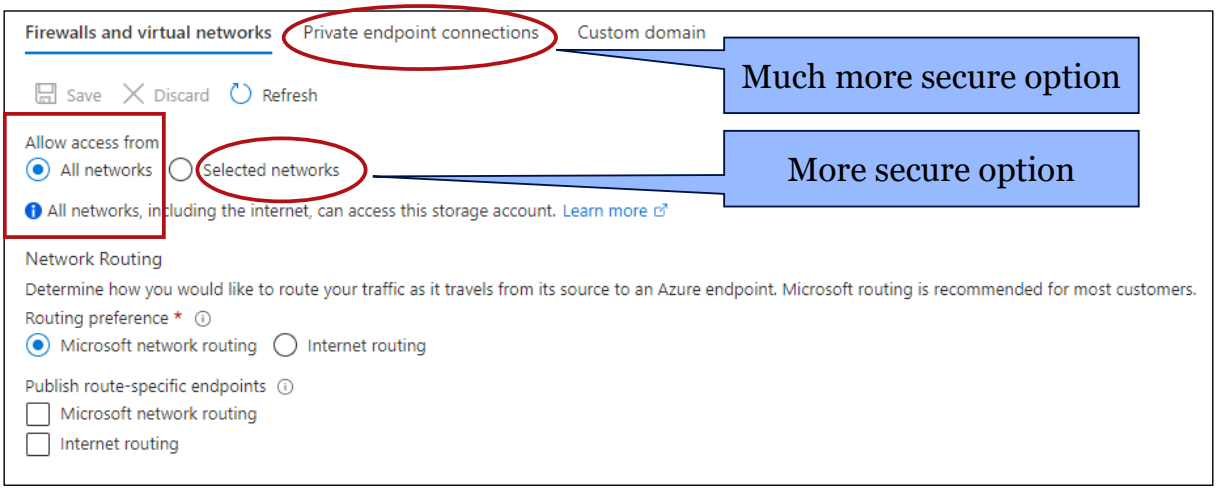
A better way to access storage resources is to use a shared access signature (SAS). An SAS grants restricted access rights to a storage resource for a specified period of time. An SAS can be generated at each level, the most restrictive being an SAS that only grants access to a single blob.

The worst possible option (unless done intentionally) is to configure a storage resource for public access.

© 2022 Pierre Lidome

Internet Access to Storage Accounts

By default, storage accounts are **accessible over the internet** irrespective of any NSG you may have configured!



As mentioned at the beginning of the class, storage accounts exist in a global namespace and must have a unique name across the entire Azure cloud. This feature enables access to your data from anywhere in the world using the URL https://mystorageaccount.blob.core.windows.net and the corresponding access key or SAS.

Global access from the internet is the default setting when an account is created. From an incident response and forensics point of view, this is less than ideal. A better configuration would limit access to specific networks rather than allow connections from anywhere on the internet.

An even more secure configuration is to use private endpoint connections. A private endpoint is a network interface that connects you privately and securely via Azure Private Link. Private endpoints require a lot of advanced planning and may not be for every organization.

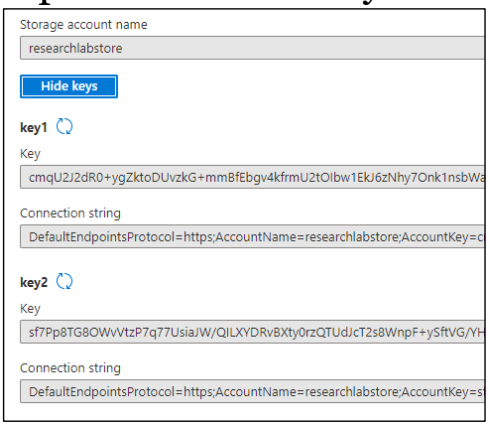
Unfortunately, it would appear that most companies choose the default settings, potentially exposing their storage resources to anyone who obtains credentials or, even worse, finds a storage resource that has been accidentally configured for public access.

© 2022 Pierre Lidome

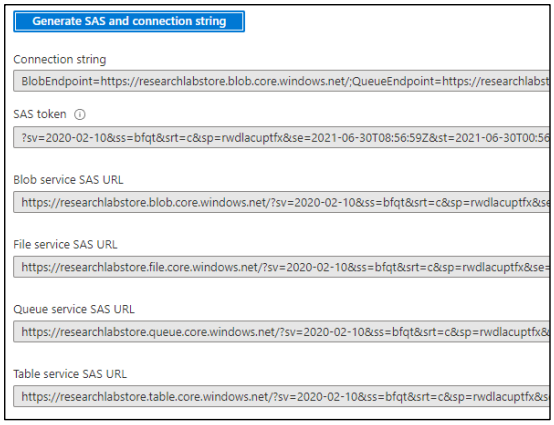
Technet24

Data Exfiltration: Generate Keys

Option 1: Access Keys



Option 2: Shared Access Signature



Activity Log Entry (Key Access):

```
"operationName": "MICROSOFT.STORAGE/STORAGEACCOUNTS/LISTKEYS/ACTION"
```


Access keys or SAS can easily be created from the Azure console, PowerShell, CLI, or Graph API.

Option 1 shows the two access keys associated with the `researchlabstore` storage account (intentionally truncated).

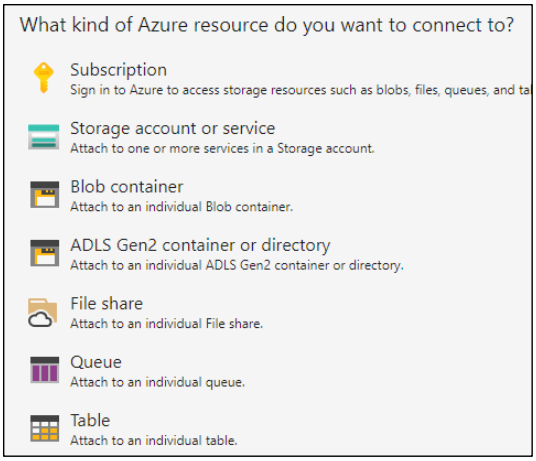
Option 2 shows the SAS created for each data storage level (blob, file service, queue, and table).

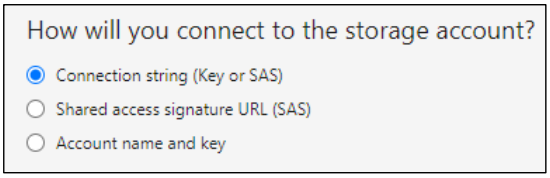
When investigating an incident, you should look for "LISTKEYS/ACTION" operations to find if any principal has enumerated the storage credentials.

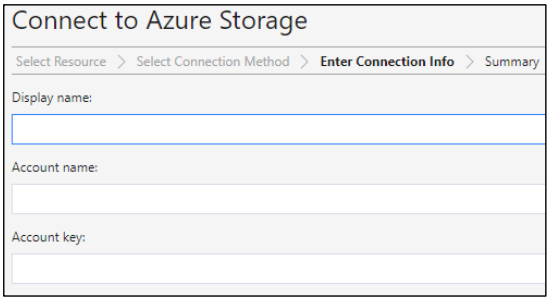
© 2022 Pierre Lidome

Data Exfiltration: Configure App

Example: Azure Storage Explorer





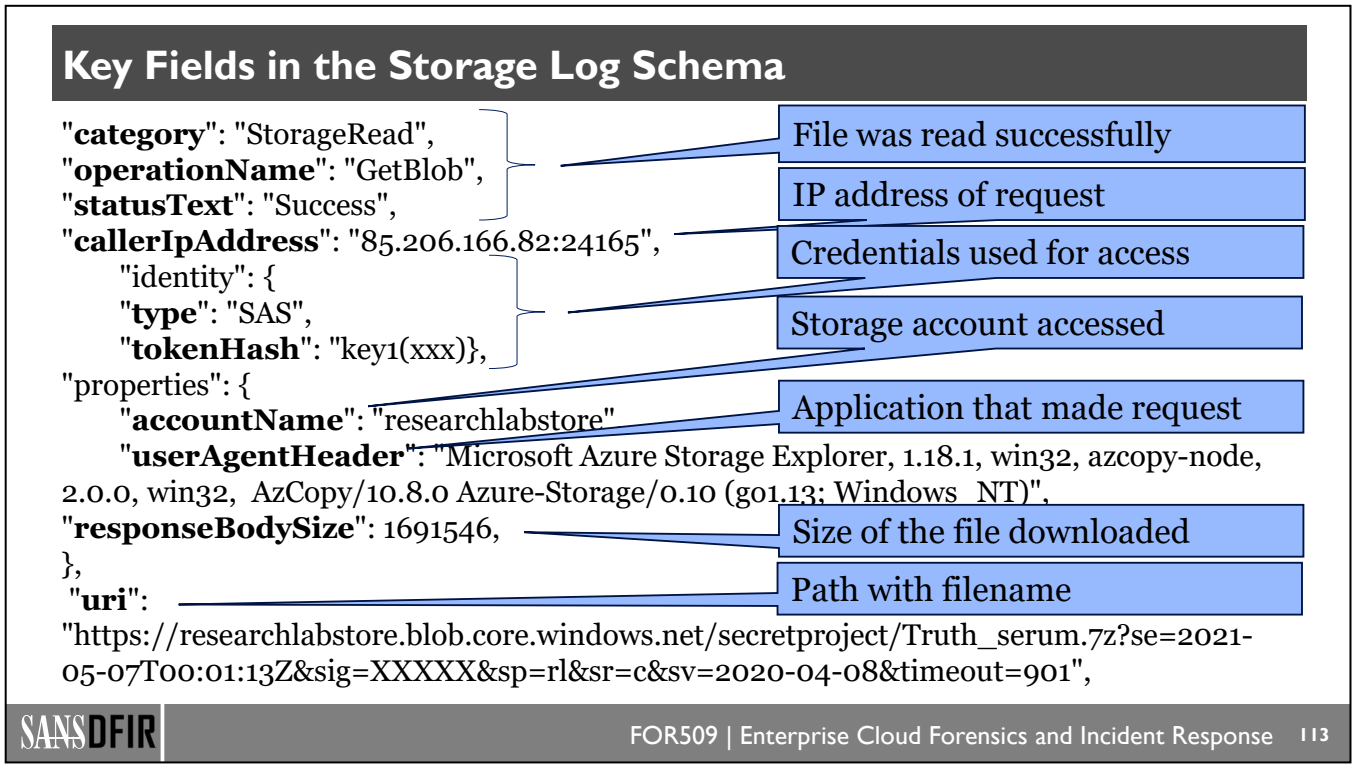


Once a threat actor has obtained storage credentials, it's trivial to exfiltrate the data associated with that storage account (or specific data levels depending on the credential). This slide shows the various configuration options for Azure Storage Explorer, depending on which types of credentials (access keys or SAS) were obtained.

The only log that will show data being transferred from a storage account is the StorageRead log. That log isn't enabled by default, and without it there will be no evidence of data exfiltration. Note that the log needs to be enabled on every single storage account. This is best accomplished via a policy at the management group level.

© 2022 Pierre Lidome

Technet24



StorageRead, StorageWrite, and StorageDelete logs are similar. We will discuss the StorageRead log since we are focusing on potential data exfiltration.

There is a large number actions tracked in the **operationName** field in the StorageRead log,[1] but for the purpose of tracking data exfiltration, we will want to focus on the GetBlob operation.

The **callerIpAddress** will provide the IP address of the threat actor making the request. If the storage account is configured with the default network security, this can be any IP address on the internet.

The **identity** block is important so that we can track which access key may have been compromised.

The **accountName** will provide the name of the storage account.

The **userAgentHeader** may indicate which application accessed the storage account. This may be of limited use, as many threat actors are likely to write their own scripts and not use an application such as Azure Storage Explorer. However, it's something to keep in mind as an investigative data point.

The **responseBodySize** will show the size of the file being downloaded. Combining this field with the **callerIpAddress** could create an interesting report to show abnormally large file transfers.

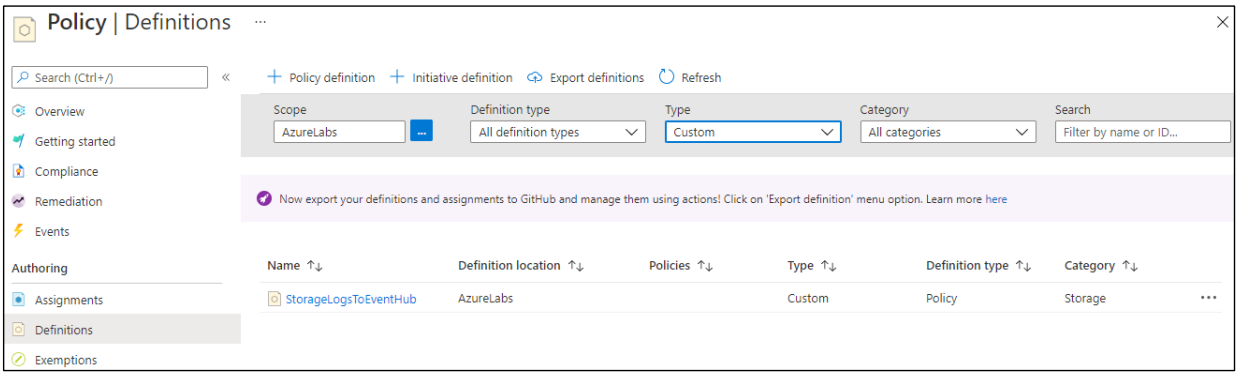
Finally, the **uri** will provide the full path, which includes the filename being downloaded.

1. https://for509.com/storage-ops

© 2022 Pierre Lidome

Enable Storage Logs via Policies

- Create a custom policy to enable storage logs
- Assign to each subscription (or to a management group)
- Class GitHub contains an example policy to send storage logs to an event hub



Enabling storage logs on each storage account can be a near impossible task for a corporation with a large number of storage accounts. Fortunately, you can create a policy that requires storage logs to be enabled on all storage accounts: existing ones and future ones.

Policies can be assigned at the management group level or subscription level. As a matter of fact, policies can be written for pretty much anything in Azure, not just storage logs. Azure already includes many predefined policies. If one isn't available to accomplish your goals, you can create a custom policy. However, creation of policies is beyond the scope of this class.

Unfortunately, there is no predefined policy to force storage account logging. As an example, you can find a policy in the class GitHub to enable storage logs and send them to an event hub. Note that there is a limitation whereby the event hub and the storage account must be in the same region. As a consequence, you will have to create an event hub in every region in which you want to have storage account logs. Again, this is just an example, and you could equally create a different policy that sends storage logs to a Log Analytics workspace or another destination.

A big thank you to Microsoft support for their help in creating this policy.

© 2022 Pierre Lidome

Technet24

Lab 2.4 Preview

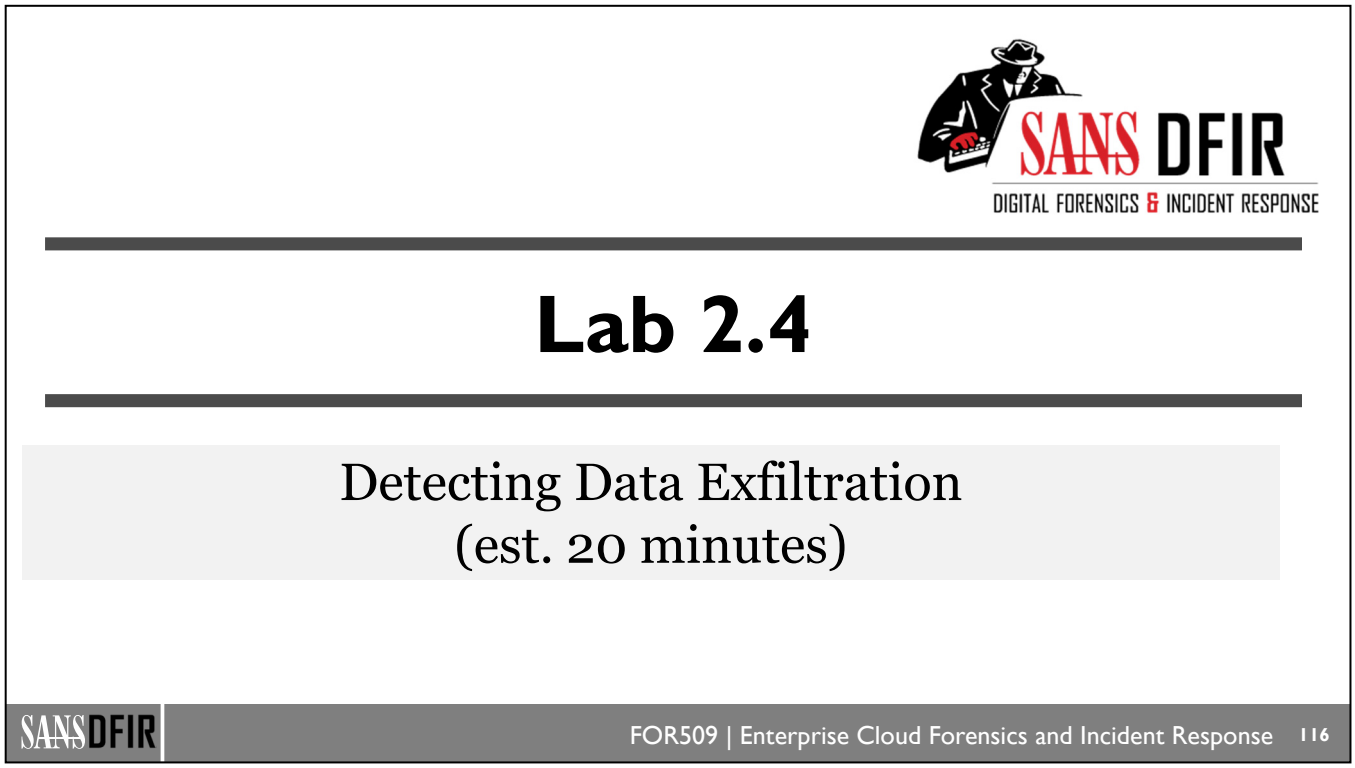
Lab 2.4 will investigate a data exfiltration scenario. We will compare Netflow logs and storage account logs.

The following fields will be important in this lab:

- `ips`
- `operation_name`
- `result_type`
- `uri`
- `user_principal_name`

This page intentionally left blank.

© 2022 Pierre Lidome



This page intentionally left blank.

© 2022 Pierre Lidome

Technet24

FOR509.2: Microsoft Azure

Section 2.1: Understanding Azure

Section 2.2: VMs, Networking, and Storage

Section 2.3: Log Sources for IR

Section 2.4: Virtual Machine Logs

Section 2.5: In-Cloud IR

This page intentionally left blank.

© 2022 Pierre Lidome

Microsoft Azure Roadmap

- 2.1: Understanding Azure
- 2.2: VMs, Network, and Storage
- 2.3: Log Sources for IR
- 2.4: Virtual Machine Logs
- 2.5: In-Cloud IR

- Windows Agents
- Windows Azure Setup and Log
- Import the Windows Event Log to SOF-ELK
- Case Study: Search for User Logins
- SOF-ELK Visualization Example
- Linux Logs
- IIS Logs
- VM Insights

By using agents, it's possible to obtain operating system logs without ever logging in to the VM. This is a great benefit to an investigation, as it avoids trampling over potential evidence.

Many companies will have EDR (Endpoint Detection and Response) systems, but for those that don't, this is a low-cost alternative. There is no charge for the agents. The cost is only in the storage used, and the configuration offers the possibility to set a quota to minimize cost.

These agents have many features to monitor metrics and the health of the VMs. These aren't as interesting to us as incident responders. We will focus on the ability to obtain log information.

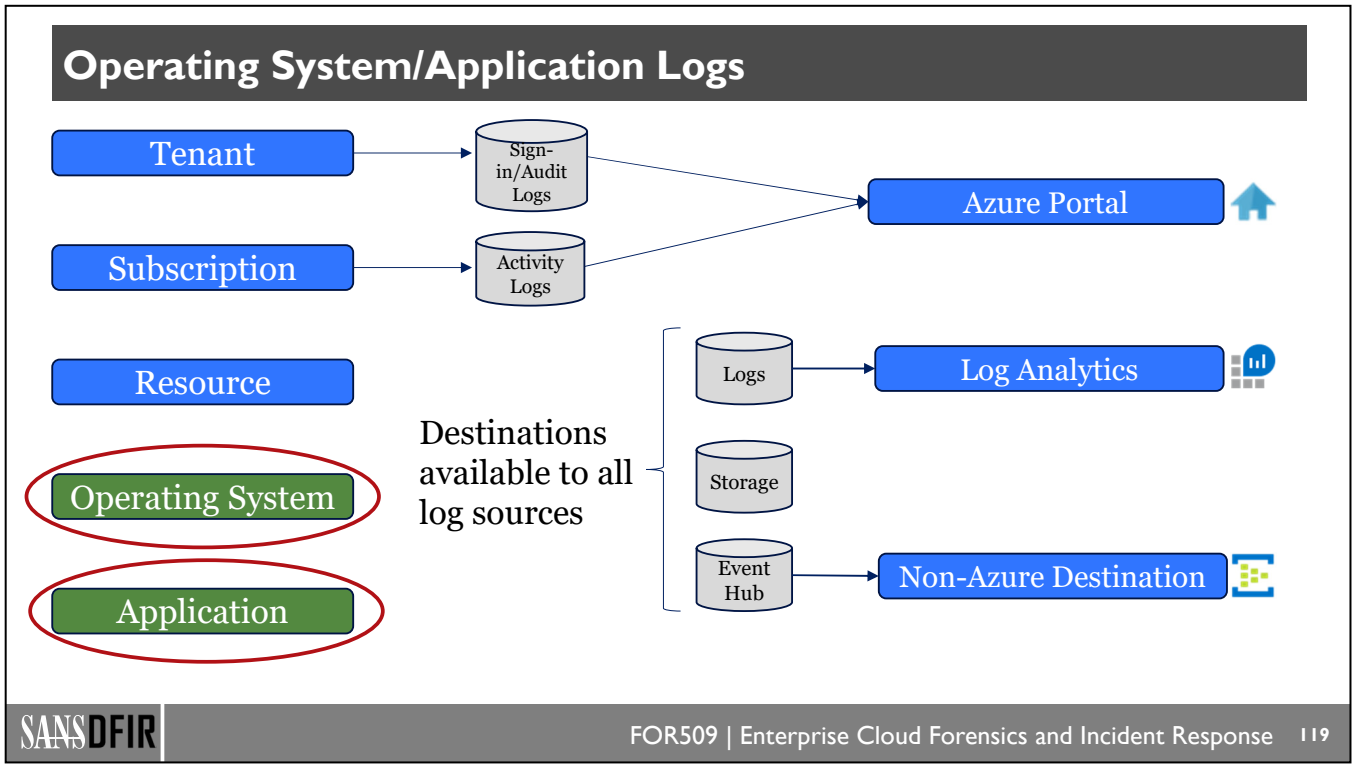
The Azure feature that organizes these logs and performance data is called Azure Monitor Log.[1]

We will review the Windows agent first and then the Linux one.

1. https://for509.com/azuremonitorlog

© 2022 Pierre Lidome

Technet24



This page intentionally left blank.

© 2022 Pierre Lidome

Windows Agents

	Azure Monitor Agent	Diagnostics Extension (WAD)	Log Analytics Agent	Dependency Agent
Data Collected	• Event logs • Performance	• Event logs • Performance • ETW events • File-based logs • IIS & .NET logs • Crash dumps	• Event logs • Performance • File-based logs • IIS logs • Insights	• Process dependencies • Network connection metrics
Data Sent To	• Azure Monitor Logs • Azure Monitor Metrics	• Azure Storage • Azure Monitor Metrics • Event Hub	• Azure Monitor Logs	• Azure Monitor Logs

- Agents collect both performance metrics and logs
- Four different agent options, but most are for in-cloud consumption
- Only the Diagnostics Extension will send data to Azure storage or an event hub
- Multiple agents can be installed on a VM

Azure offers four possible agents to collect a wide variety of metrics and logs.[1] In selecting the correct combination of agents, it's important to first consider how the data will be consumed. Only the diagnostics extension is able to write the data to a storage account or an event hub.

The Azure Monitor agent is now generally available as of November 2021. It's intended to replace legacy agents such as the Log Analytics agent and the Diagnostics Extension. The main benefit of the new Azure Monitor agent is the implementation of data collection rules which facilitate a higher degree of granularity to define what information to collect.

However, there are still gaps (such as the inability to write the logs to a storage account) and for now we will focus on the diagnostic extension. Microsoft frequently refers to that feature as Windows Azure Diagnostics (WAD), and the logs start with that prefix.

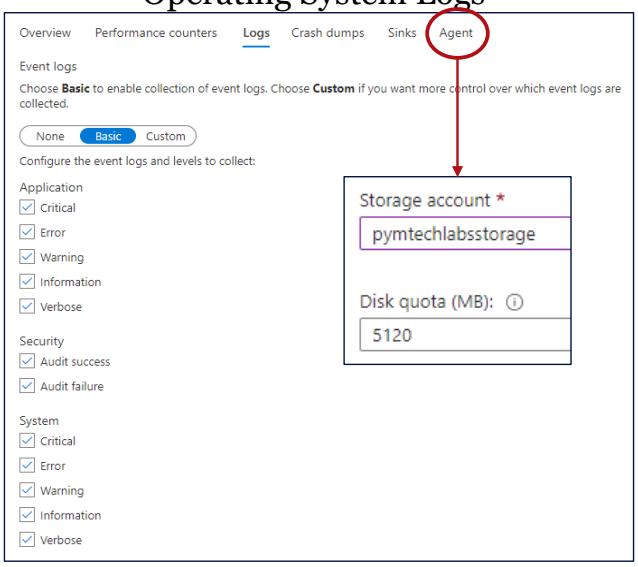
1. https://for509.com/monitoragents

© 2022 Pierre Lidome

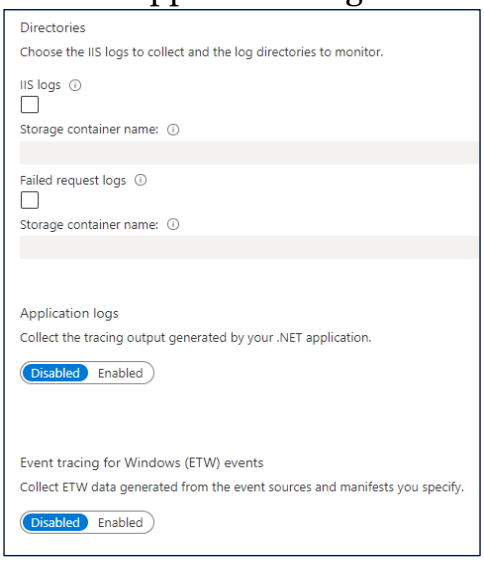
Technet24

Configuring WAD

Operating System Logs



Application Logs



Before we can extract any logs, we need to configure WAD. First, you will select the diagnostic settings for your virtual machine. Then, under the "Logs" tab, you can select the event logs and levels that you want to collect. Finally, under "Agent," you will select the storage account to send the logs to. You can also choose a disk quota to make sure the logs don't grow indefinitely.

As with other logs, selecting everything is not recommended, as you will get a lot of noise and use a lot of storage.

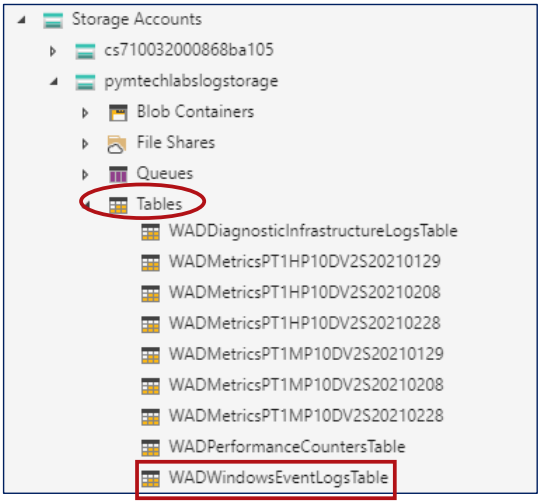
You may also collect logs for IIS, tracing output from .NET applications, and Event Tracing for Windows (ETW).

For the purposes of incident response, we will focus on operating system logs.

Now that these logs are collected, we need to look at the structure of WADWindowsEventLogsTable.

© 2022 Pierre Lidome

Windows Azure Diagnostics (WAD)



- Logs we covered earlier in the class were stored in blob containers and formatted in JSON
- WAD logs are stored in tables, and exports are limited to .csv files
- SOF-ELK ingestion script is included in the latest release of SOF-ELK
- WADWindowsEventLogsTable is the most interesting table for incident response and forensics, as it contains the Windows event logs

The WADWindowsEventLogsTable is of particular interest because it contains Windows event logs. This is a great opportunity to obtain operating system logs without the need to log in to the VM itself.

So far, all our logs have been stored in blob containers and formatted in JSON, which is easy to download via the Azure Storage Explorer. As previously shown, copying these logs to the `/logstash/azure` directory of your SOF-ELK VM will make them easily accessible and searchable in Kibana.

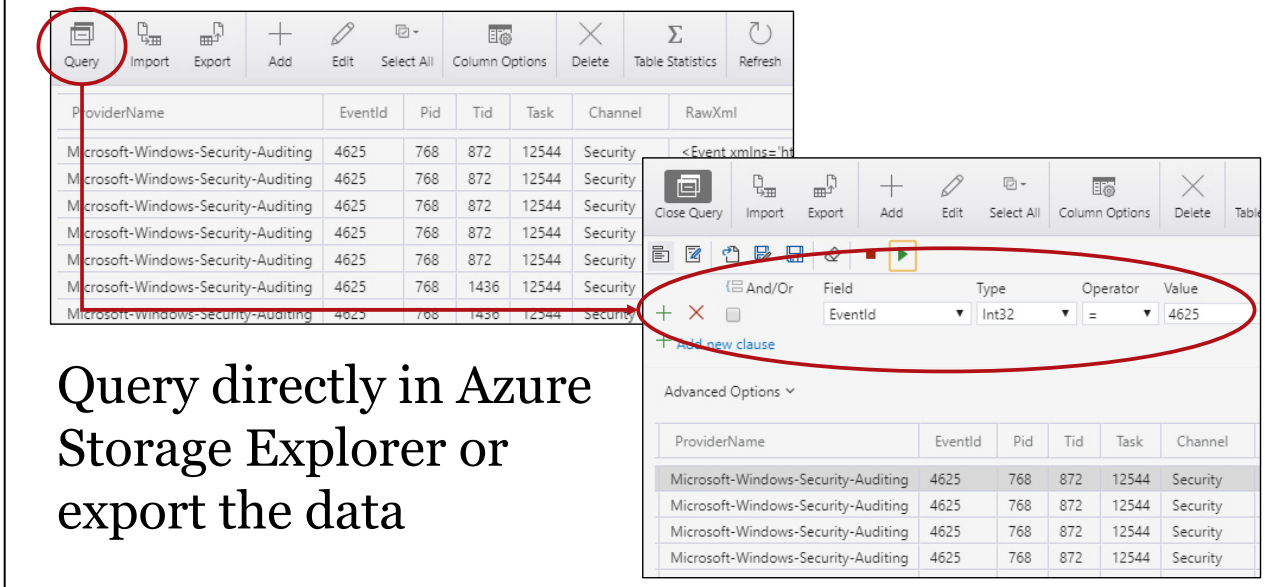
Unfortunately, the Windows Azure diagnostics logs aren't so easily accessible. They are stored in a NoSQL table in your storage account. Azure Storage Explorer can access this table and export it to a .csv file.

In the next slides, we will show you how to extract the information and import it to SOF-ELK.

© 2022 Pierre Lidome

Technet24

WADWindowsEventLogsTable



Query directly in Azure Storage Explorer or export the data

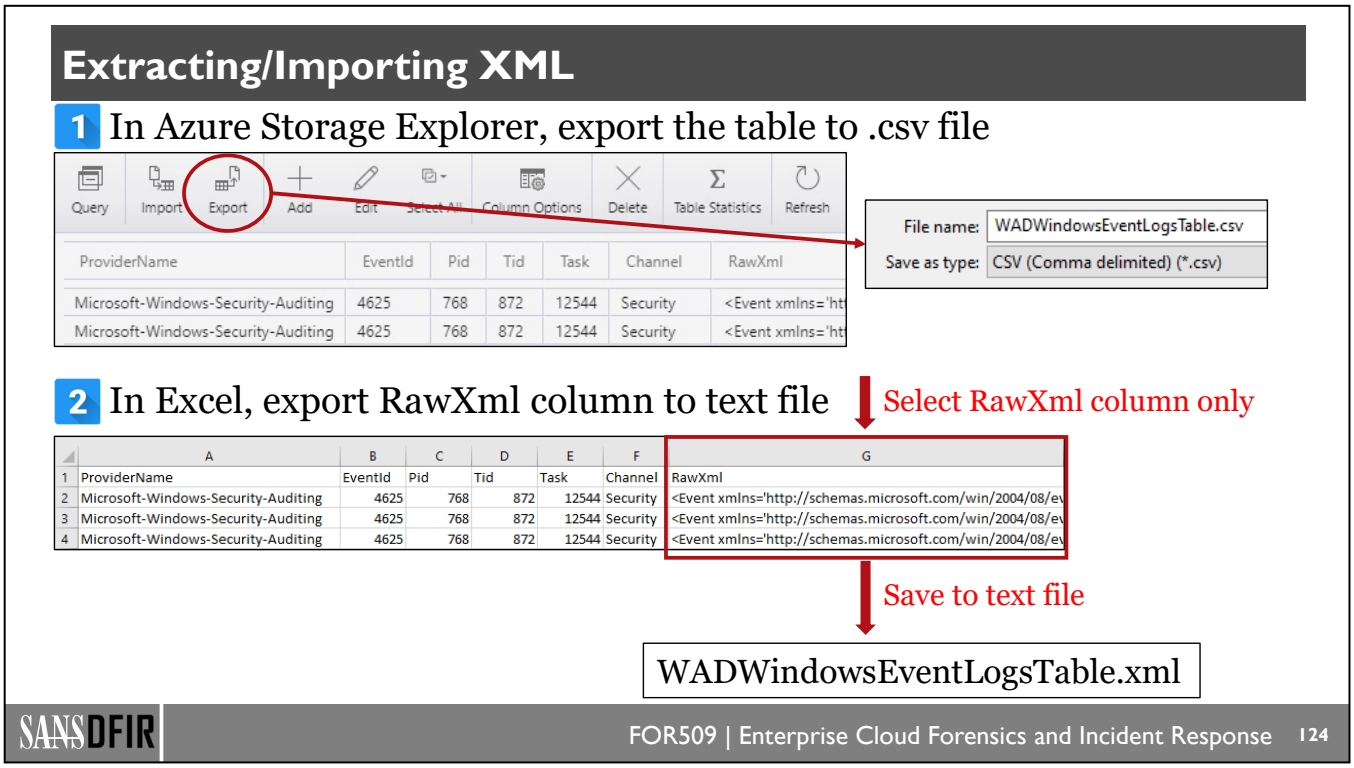
Using Azure Storage Explorer, we can examine the contents of WADWindowsEventLogsTable. You will notice a large number of columns. The column called RawXml contains all the data concerning that particular event. Every other column is derived from that column.

The benefit of breaking down the RawXml column to individual columns is that you can perform queries directly in Azure Storage Explorer. In the above examples, we are looking for EventId 4625, which is "An account failed to log on."

It's quite advantageous to be able to query operating system logs without needing to log in to the virtual machine or even Azure. As long as Azure Storage Explorer has been configured with the storage account access key, you can query to logs directly from your computer.

While this is convenient for a quick search, for a larger investigation it's preferable to import these logs into a SIEM platform like SOF-ELK and have the ability to correlate these events with other logs.

© 2022 Pierre Lidome



SOF-ELK has a Logstash import script to process the RawXml column. We need to extract that column from the WADWindowsEventLogsTable. Many solutions are possible, and in this slide, we propose a simple one that consists of two steps:

1. In Azure Storage Explorer, export the table to a .csv file.
2. In Excel (or any other tool of your choice), export the RawXml column to a text file.
   a) Be sure to name that file with an .xml extension, as the Logstash script will expect it.
   b) Don't forget to remove the header row (first row).

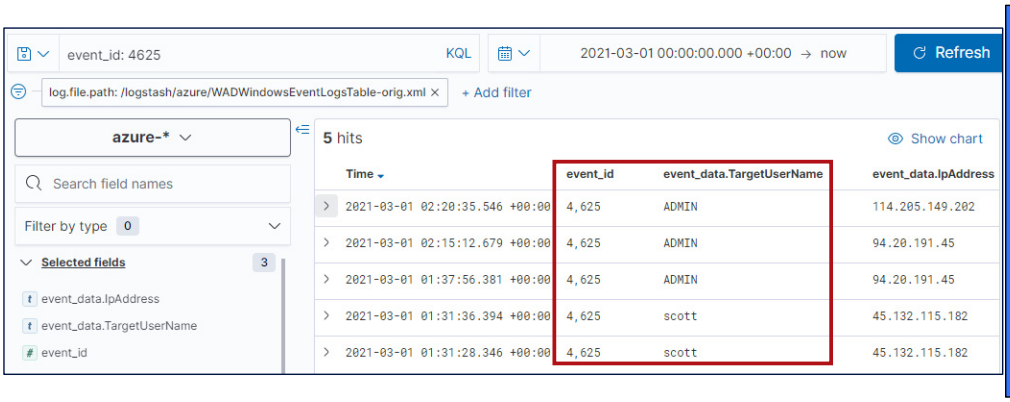
© 2022 Pierre Lidome

Technet24

Import into SOF-ELK

```
C:\> scp WADWindowsEventLogsTable.xml elk_user@192.168.223.130:.

[elk_user@sof-elk]$ cp WADWindowsEventLogsTable.xml /logstash/azure
```



NOTE

Notice the failed logins (Windows Event ID 4625). Leaving RDP open to the internet is a bad idea!

To ingest the .xml file into SOF-ELK, follow these two steps:

1. Copy the .xml file from your computer to the SOF-ELK VM. `scp` is the easiest way to perform this step. You will need to enter the correct IP address for your SOF-ELK VM. The password for the elk_user account is `forensics`.
2. Copy the .xml file to the `/logstash/azure` directory. ELK will start ingesting the data right away. Depending on the size of the file, you may have to wait a few minutes before the data shows up in Kibana.

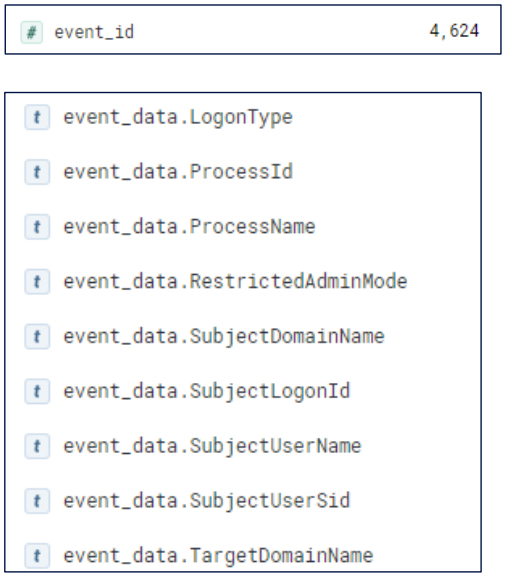
Now that the data is available to us, we can easily query for failed logons like we did in Storage Explorer (event ID 4625). You will notice three failed logons with a username of ADMIN and two with a username of scott. The ADMIN failed logons demonstrate the danger of leaving a VM with port 3389 open on the internet.

Events are logged to the Windows event log by various providers. Each provider formats their log entry differently, making parsing these events quite challenging. In order to speed up the data ingestion, the Logstash script is only processing events from the "Microsoft-Windows-Security-Auditing" provider.

© 2022 Pierre Lidome

Windows Event Field Mapping in SOF-ELK

- The event ID is shown in field: event_id
- The other Windows event fields are mapped to event_data.<field name>
- Fields will vary depending on the event
- Windows VMs tend to have very noisy logs, so it's best to filter by event_id first



Windows event logs are mapped in SOF-ELK under the event_data.<field name>.

Different fields will be available depending on the event_id.

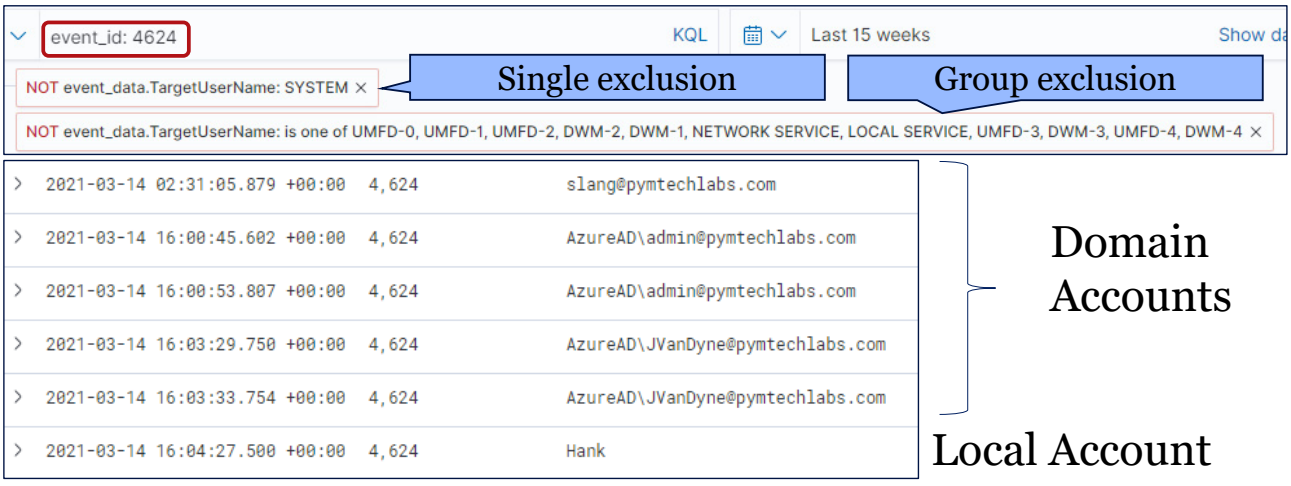
Azure Windows VMs tend to log a lot of information if the audit success and failure options are selected. It's best to first filter for a specific event_id and then start removing noisy events (for example, the SYSTEM user).

© 2022 Pierre Lidome

Technet24

Search for User Login

Example: Only want to see successful logins for "real" users (not system accounts) by filtering on event_id: 4624



If we want to see "real" user logins, it's not enough to filter for event_id 4624. We also need to filter out all the system accounts. ELK supports filtering a single item at a time:

```
NOT event_data.TargetUserName: SYSTEM
```

Or you can filter a group of items with the operator "is one of":

```
NOT event_data.TargetUserName: is one of UMFD-0, UMFD-1, UMFD-2, etc.
```

While a bit tedious, you only need to do it once, as you can save your search.

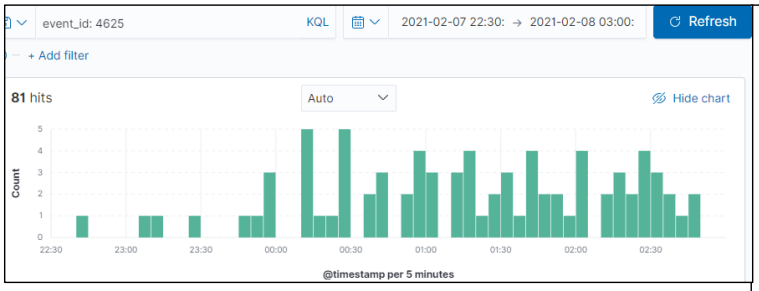
For future reference:

- UMFD-* system accounts are generated by the User Mode Driver Framework and are used by the Usermode Font Driver Host process (fontdrvhost.exe).

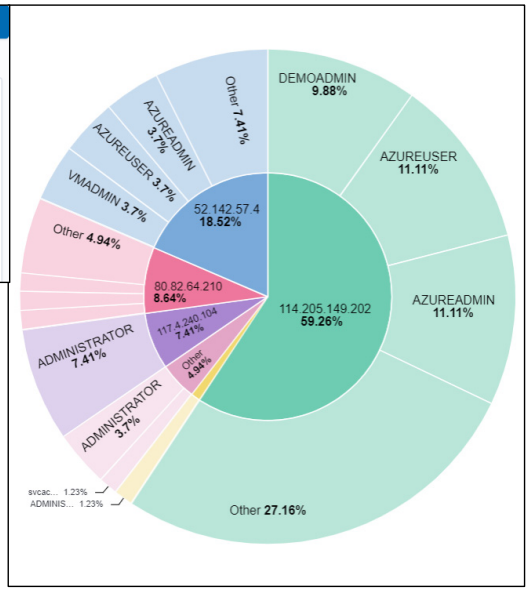
- DWM-* accounts are associated with the Desktop Window Manager process (dwm.exe).

© 2022 Pierre Lidome

SOF-ELK Visualization Example



- Windows VM with RDP open to the internet
- Four-hour time window: 81 failed login attempts
- Notice the accounts being "tested"

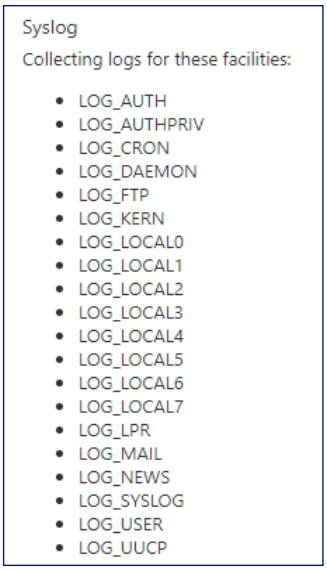


We have looked at numerous logs, and now that they are imported in SOF-ELK, we can create interesting visualizations. This example illustrates the danger of leaving port 3389 (RDP) opened to the internet. In a four-hour window, multiple unauthorized actors attempted to guess passwords to accounts that don't even exist. The choice of account names is noteworthy, if anything to avoid using these names for your real system accounts.

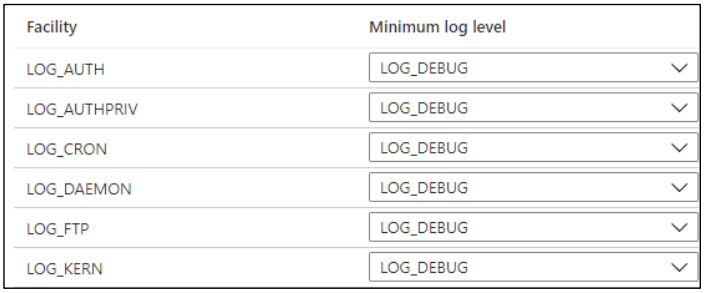
© 2022 Pierre Lidome

Technet24

Linux Logs



- Various Linux logs are collected by the agent
- Each event has a facility and severity
  - Facility: Source of event. Example: Kernel, ftp, cron, etc.
  - Severity: Importance. Example: Emergency, alert, etc.
- You can specify which facility & severity to collect



To set up logging for a Linux machine, you will select the diagnostic settings for your virtual machine. Similar to other diagnostic settings, you will need to select the storage account you wish to use to store the logs.

For Linux machines, you have two tabs: metrics and syslog. The metrics tab will allow you to select the sample rate for Processor, Memory, Network, File System, and Disk. Metrics can sometimes be useful in incident response. For example, an investigation of crypto mining may show a high-processor utilization, or an investigation of ransomware may show high disk and filesystem utilization.

However, we are mostly interested in the syslog configuration. On that tab, you will see a number of "Facility" options. A facility represents the machine process that created the syslog event. For example, that could be the kernel, ssh daemon, mail system, etc. On a Linux machine, these logs may be stored in separate files: auth.log, kern.log, syslog, etc. Azure combines everything in a single table. If you want to learn more about the syslog protocol, see RFC 5424.[1]

The last selection you need to make is the log level. There are seven log levels:

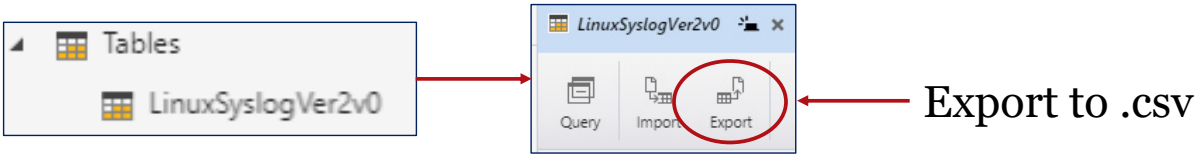
1. **Debug**: Very verbose logs, mostly used to debug problems.
2. **Info**: Informational messages; no action required.
3. **Notice**: Normal but significant condition.
4. **Err**: Error condition, but non-urgent failure.
5. **Crit**: Critical condition; should be corrected immediately because there is a failure.
6. **Alert**: Action must be taken immediately.
7. **Emerg**: Emergency; the system in unusable.

1. https://for509.com/rfc5424

© 2022 Pierre Lidome

LinuxSyslogVer2v0 Table

Linux logs can be accessed in Azure Storage Explorer under Tables -> LinuxSyslogVer2v0



Example of user scott login to host UbuntuMachine using SSH

EventTime	Facility	Host	Msg	Severity	ident	pid
2021-03-16T01:21:45+0000	auth	UbuntuMachine	Accepted password for scott from 45.56.183.51 port 53501 ssh2	info	sshd	7085
2021-03-16T01:21:45+0000	authpriv	UbuntuMachine	pam_unix(sshd:session): session opened for user scott by (uid=0)	info	sshd	7085
2021-03-16T01:21:45+0000	auth	UbuntuMachine	New session 27 of user scott.	info	systemd-logind	1083
2021-03-16T01:21:45+0000	daemon	UbuntuMachine	Started Session 27 of user scott.	info	systemd	1

The logs are found in a table called LinuxSyslogVer2v0. While the logs can be reviewed and searched inside Azure Storage Explorer, exporting them to a .csv file provides more options.

The .csv file will need a bit of cleaning up, and you should focus on the following fields:

- **EventTime**: The time at which the event occurred. The log contains numerous timestamps, but this is the most important one.
- **Facility**: As described in the previous slide.
- **Host**: The name of the machine (there is a redundant field called hostname).
- **Msg**: The most important field that contains the actual event.
- **Severity**: As described in the previous slide.
- **Ident**: The process that generated the event.
- **Pid**: The process ID.

The example above shows user scott logged into a machine called UbuntuMachine using SSH with password authentication (a bad practice; should be using a certificate!).

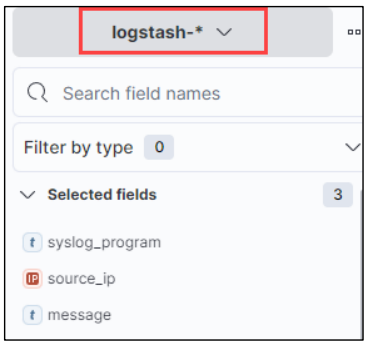
Excel is a great tool for analyzing a small amount of data. However, importing this log in SOF-ELK is preferable when dealing with multiple hosts and to correlate the activity with other logs.

© 2022 Pierre Lidome
Technet24

LinuxSyslogVer2v0 Table to SOF-ELK

To search syslog in SOF-ELK:

- Copy csv file to `/logstash/azure`
- Syslog events are in the `logstash-*` index
- Key fields are:
  - `syslog_program`
  - `source_ip`
  - `message`



Time	syslog_program	source_ip	message
2021-03-16 01:21:45.000Z	systemd	-	Started Session 27 of user scott.
2021-03-16 01:21:45.000Z	systemd-logind	-	New session 27 of user scott.
2021-03-16 01:21:45.000Z	sshd	45.56.183.51	Accepted password for scott from 45.56.183.51 port 53501 ssh2
2021-03-16 01:21:45.000Z	sshd	-	pam_unix(sshd:session): session opened for user scott by (uid=0)

SOF-ELK will easily give us the same results as long as we select the correct index. Syslog events are standardized across most UNIX platforms, so SOF-ELK will store these events in the `logstash-*` index (not `azure-*` like most of our other logs).

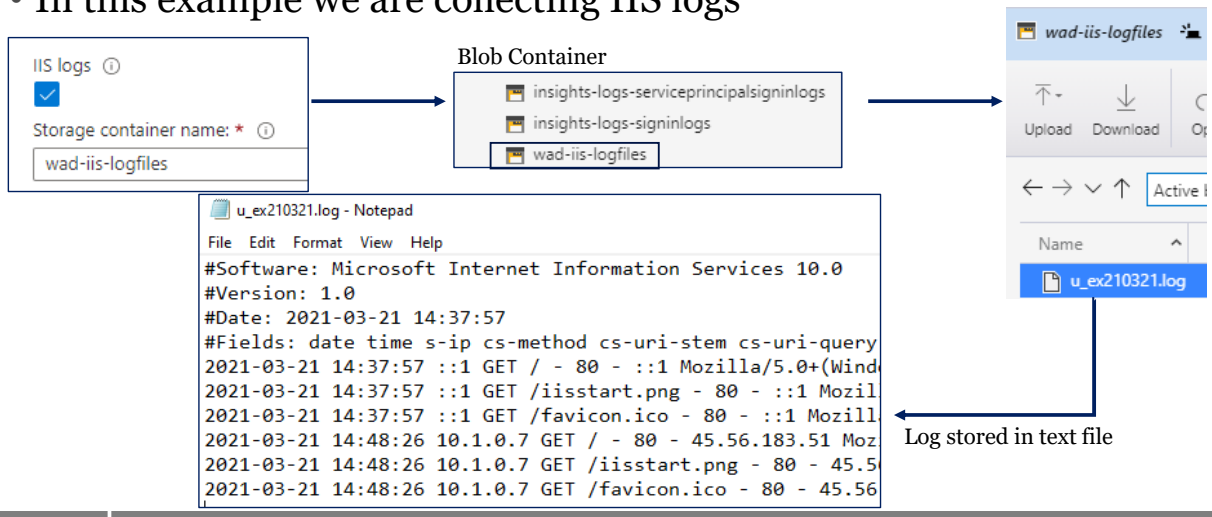
`cron` (UNIX equivalent of the task scheduler) is very noisy, so we need to eliminate these events with the filter `NOT syslog_program: CRON`. You can add additional filters if you have other noisy processes.

By selecting the fields `syslog_program`, `source_ip` and `message`, you will get a nice table that will help you easily see all logins to the machine.

© 2022 Pierre Lidome

Application Logs: wad-iis-logfiles Example

- There are many application logs that can be collected
- In this example we are collecting IIS logs



There are many application logs that can be collected by the diagnostic agent. This includes tracing output generated by your .NET application and Event Tracing for Windows (ETW) events.

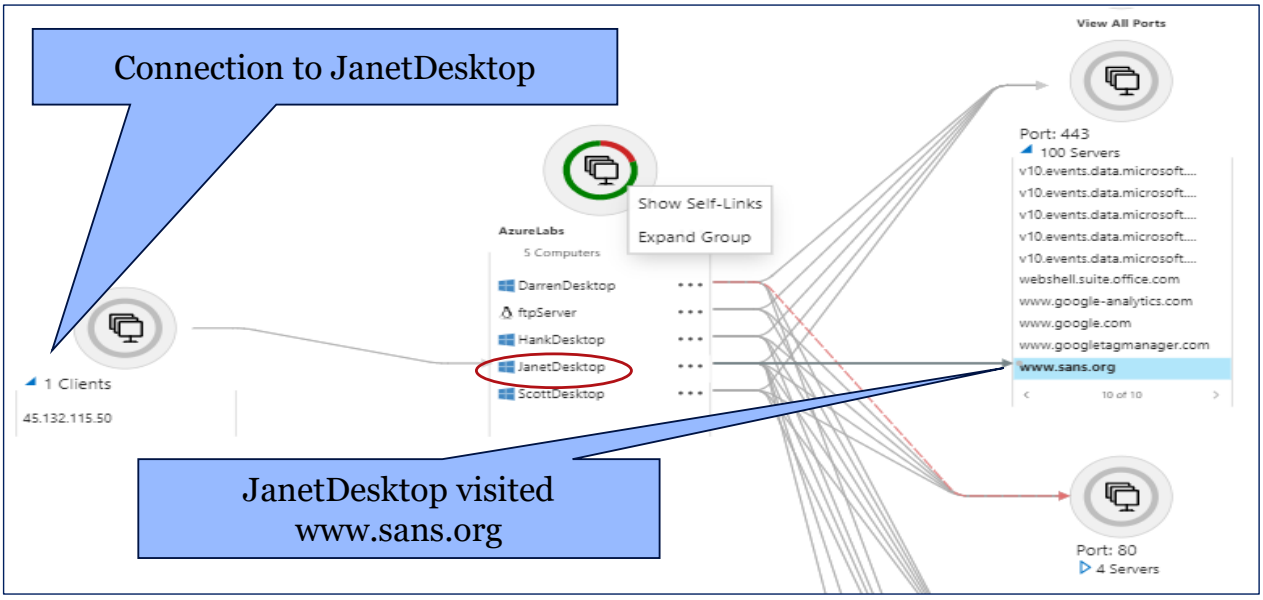
In case you are not familiar with ETW, it's the ability to capture kernel and application events in order to diagnose system and application performance issues.

In this example we are collecting IIS logs, which are stored in a blob called `wad-iis-logfiles`. The log is stored in a plaintext file, which is different from the other logs we saw earlier in this class that were all in JSON format.

© 2022 Pierre Lidome

Technet24

VM Insights



Azure has an interesting feature called VM Insights.[1] When enabled, this feature allows you to visualize various components for Windows and Linux virtual machines. The primary purpose of VM Insights is to monitor the performance and health of virtual machines.

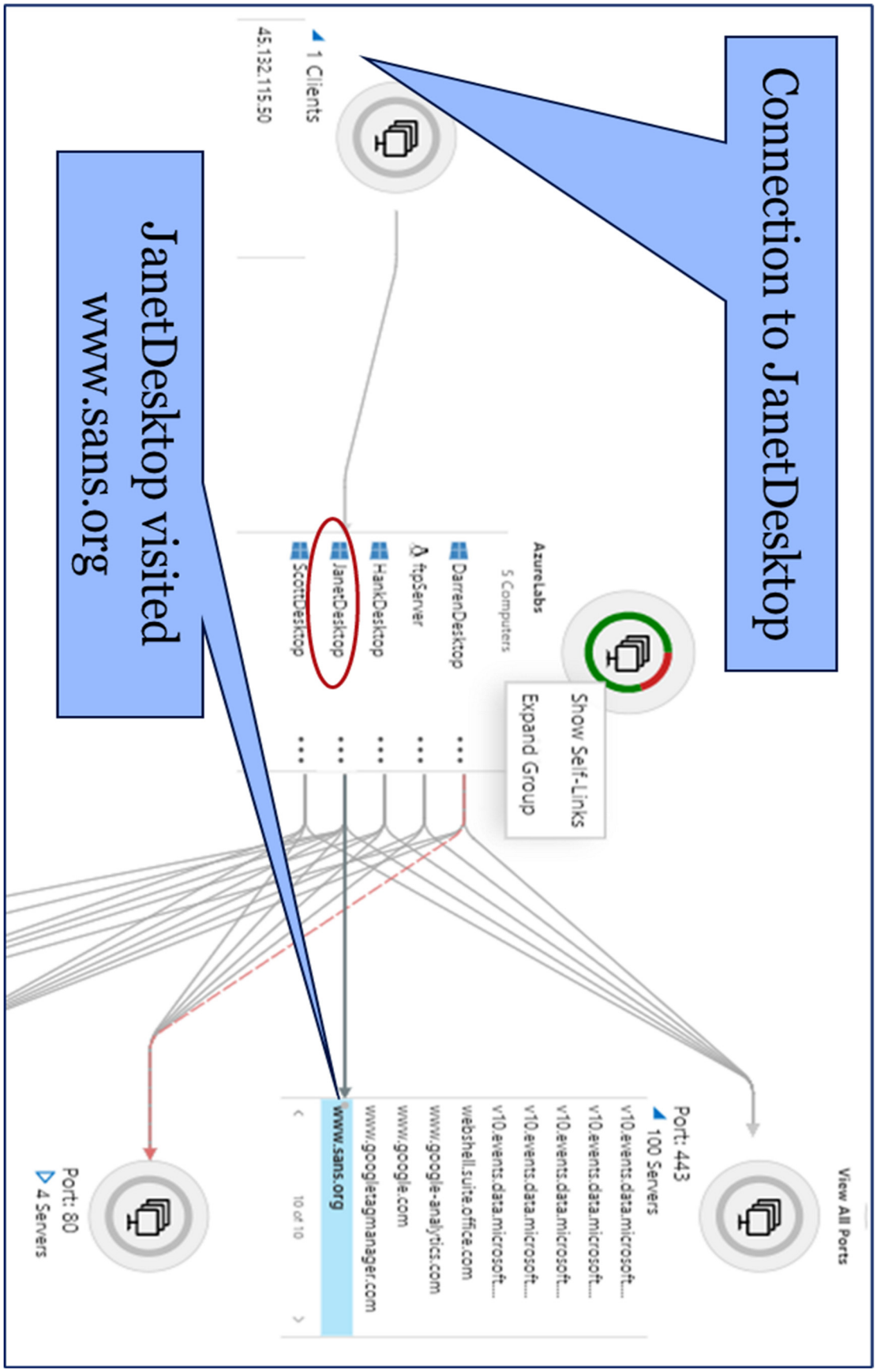
However, one of the features of VM Insights is to display a map of the environment that shows the connections to and from each virtual machine. This can be very valuable if you need to investigate an incident in an environment that's new to you.

In this example, you can see on the left side that a machine located at IP 45.132.115.50 connected to JanetDesktop. In the middle you see every VM that has been configured with VM Insights. On the right side, you see every outbound connection. By selecting a specific outbound connection, you will see which VM initiated that connection.

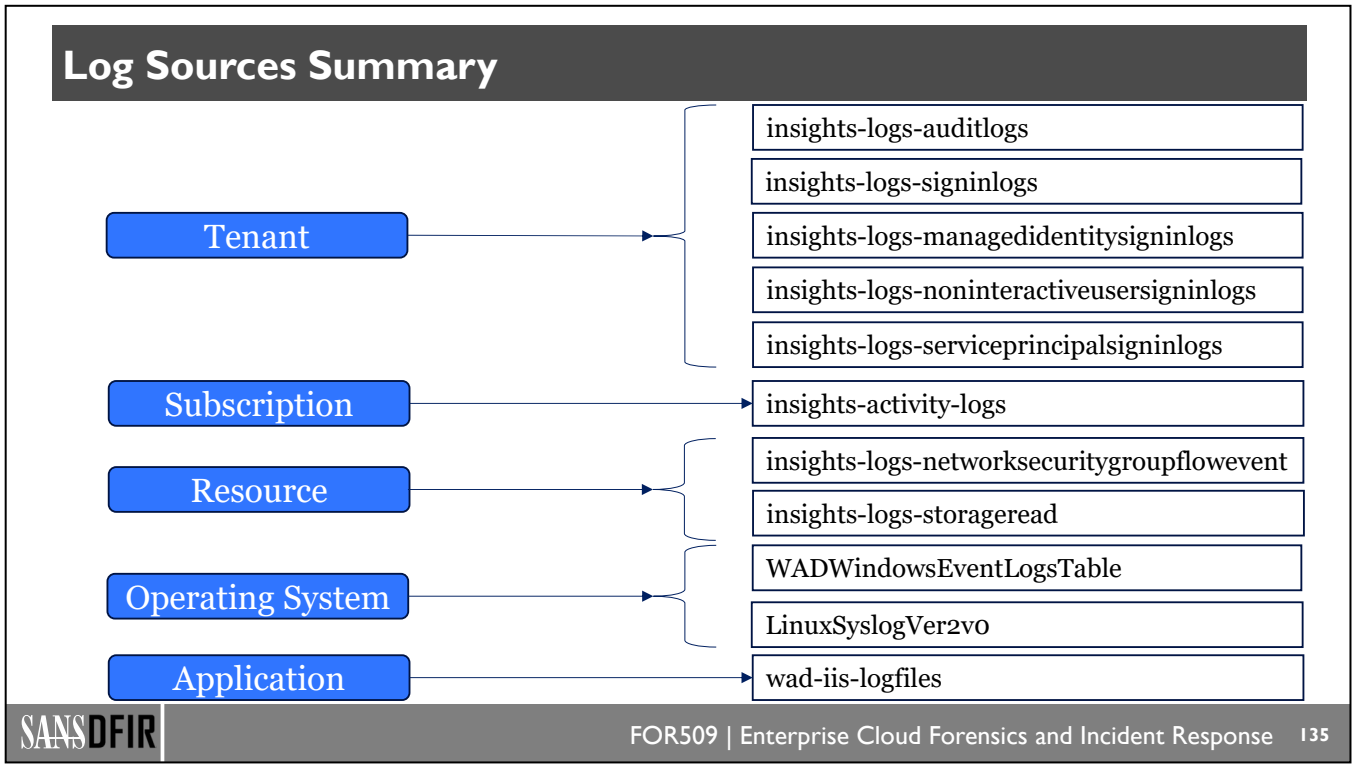
To enable VM Insights, you will need a Log Analytics workspace and to configure each virtual machine to send data to that workspace.

1. https://for509.com/vminsightsmap

© 2022 Pierre Lidome



© 2022 Pierre Lidome
Technet24



We have looked at numerous log sources. In this slide, we summarize these log sources with their names. These are just the ones we have discussed in this class and are most relevant for incident response and forensics. There are many more log sources available in Azure, but they usually focus on the performance and health of resources.

The tenant, subscription, resource, and application logs are found in container blobs. The operating system logs are found in tables, which adds some complexity when trying to export them.

We have created Logstash ingestion scripts for all these log sources (except the IIS application log since it's a plaintext file) so that you may import and analyze them in SOF-ELK.

© 2022 Pierre Lidome

FOR509.2: Microsoft Azure

Section 2.1: Understanding Azure

Section 2.2: VMs, Networking, and Storage

Section 2.3: Log Sources for IR

Section 2.4: Virtual Machine Logs

Section 2.5: In-Cloud IR

This page intentionally left blank.

© 2022 Pierre Lidome

Technet24

Microsoft Azure Roadmap

- 2.1: Understanding Azure
- 2.2: VMs, Network, and Storage
- 2.3: Log Sources for IR
- 2.4: Virtual Machine Logs
- 2.5: In-Cloud IR

- Imaging a Drive in the Cloud
- In-Cloud Investigations
  - Snapshots
  - Create a Forensic VM
  - Run Forensic Tools
- Forensic VM Portability
- Other Azure Resources
  - Azure Sentinel
  - Azure SimuLand
  - Microsoft Incident Response Playbooks
  - Recommended Projects

This page intentionally left blank.

© 2022 Pierre Lidome

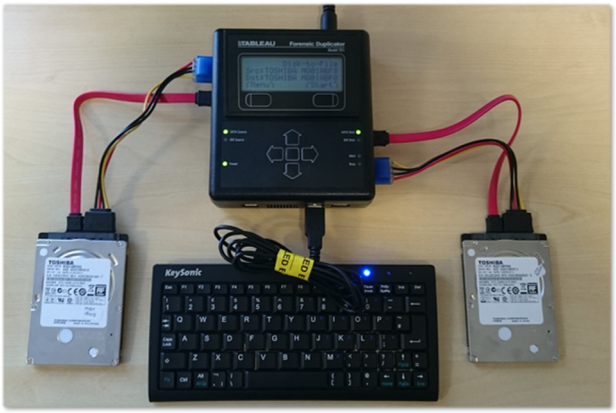
Imaging a Drive in the Cloud

Once we have reviewed all the logs and still need more data, how do we image a drive in the cloud?

The "old" method with a disk duplicator is clearly not possible

For large investigations, it's possible to use Azure's Import/Export service

Otherwise, an in-cloud investigation is the most efficient option



We have spent a lot of time reviewing all the log sources available to us for our investigations. There comes a point where we need the actual data from the machine. This section will take you through the steps necessary to acquire an image of a VM.

Once an image has been acquired, what do we do with it? While downloading it to our traditional forensic workstation may sounds like the easiest option, there is a cost for data egress. Given disk sizes in the hundreds of gigabytes, that cost can be significant. Further, downloading a large amount of data may take a lot of time, which would hinder our investigation.

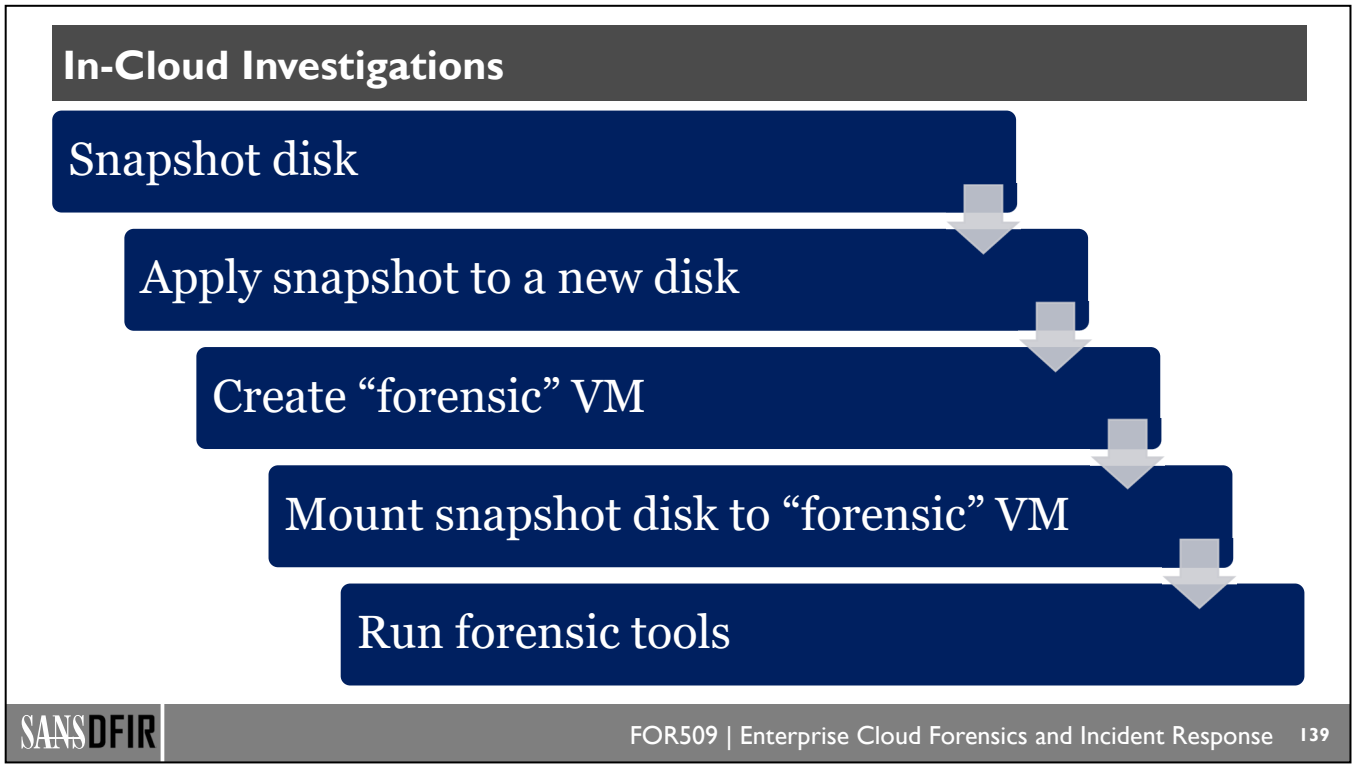
The solution is to perform the forensic analysis in-cloud. For that purpose, we will create a new VM, called "forensic VM", to access the imaged disk, thus maintaining the integrity of the original VM, which we will designate as "victim VM".

If you are facing an investigation requiring a large amount of data from Azure, you can use the Azure Import/Export service to request that data.[1]

1. https://for509.com/exportservice

© 2022 Pierre Lidome

Technet24



Our in-cloud investigation will follow these five steps:

1. Snapshot the OS disk from the “victim VM”.
2. Create a new disk based on the contents of the snapshot.
3. Create a new VM with our forensic tools: “forensic VM”.
4. Mount the disk from step 2 to “forensic VM”.
5. Run your favorite forensic tool.

© 2022 Pierre Lidome

Step 1a: Snapshot VM's Disk

The key to imaging a drive in the cloud is a feature called **snapshot**.

- A snapshot is a full, read-only copy of a virtual hard drive
- You can take a snapshot of an OS or data disk
- Snapshots can be taken while the VM is running or shut down

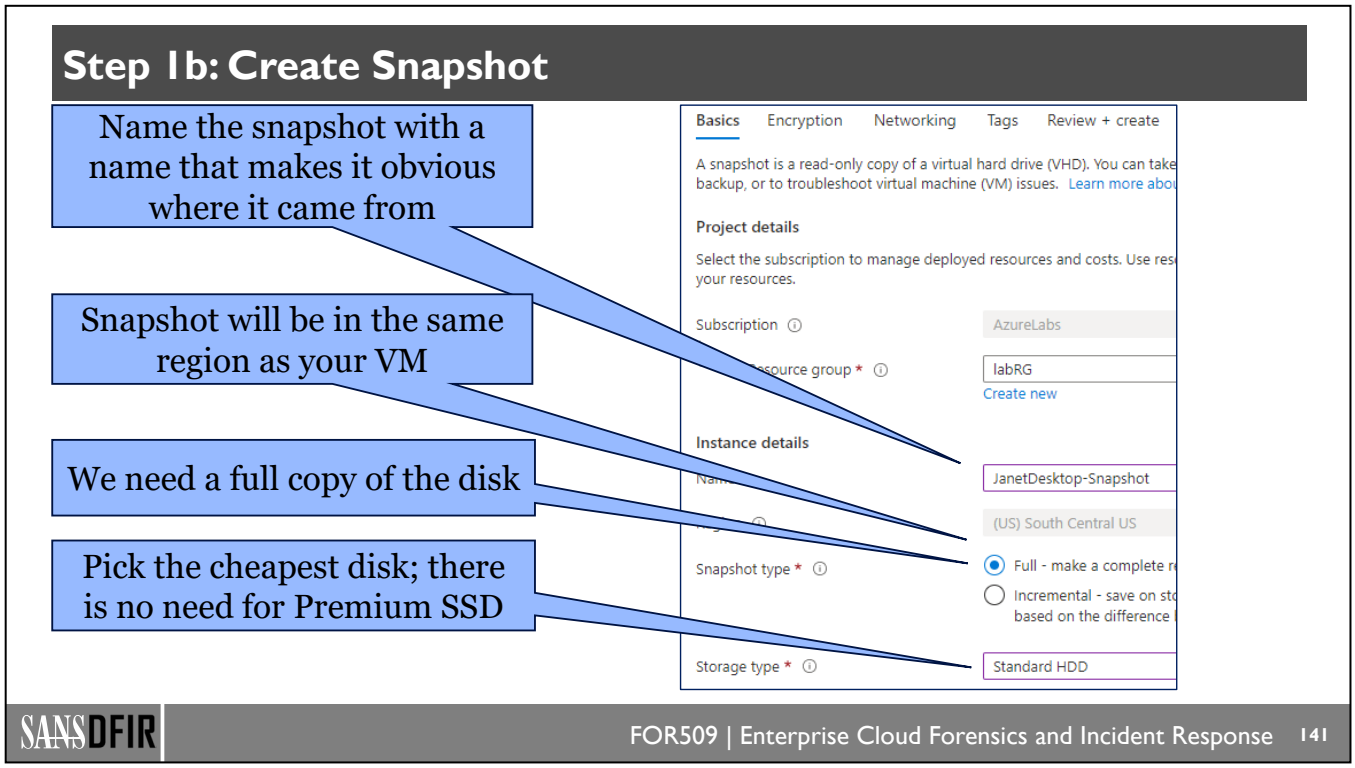
A snapshot is a full, read-only copy of a virtual disk. It's an amazing technology that allows you to make a copy of a disk in just seconds. You can snapshot a disk even when the VM is running.

To create a snapshot, you will select the VM and then the disk. From there, you will have an option to create a snapshot.

For most investigations, it should be sufficient to snapshot the operating system (OS) disk. However, if required, you may also snapshot any data disk associated with that VM. Be aware of the on-going costs associated with snapshots ($0.05/GB/month for standard storage and $0.132/GB/month for premium storage).

© 2022 Pierre Lidome

Technet24

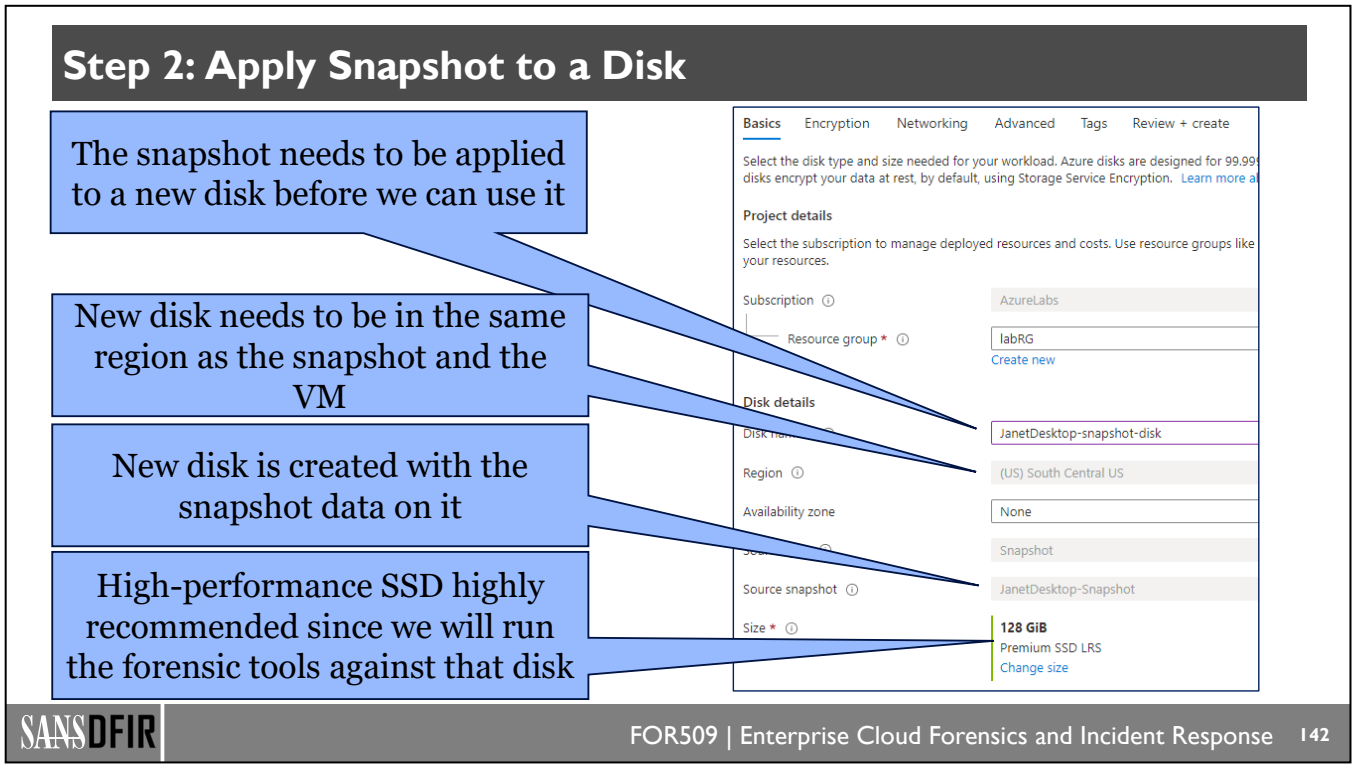


When you're creating a snapshot, it's important to give it a name that quickly identifies it as a snapshot. The reason is that snapshots can't be used as-is. They need to be applied to a new disk (which we will see in the next step). When performing that step, you want to be sure to select the correct snapshot, hence the importance of a descriptive name.

For the purposes of our investigation, we want a point-in-time copy of the entire disk. Hence, we will select "Full" in the snapshot type. The "Incremental" choice is used when using snapshots for ongoing backups.

Since the snapshot will be applied to a new disk, there is no need to spend money on premium storage at this point. We recommend using "Standard HDD".

© 2022 Pierre Lidome



We now need to create a disk that contains the snapshot data. A good practice would be to name this disk the same as the snapshot and add "-disk" at the end. This will prevent getting confused with all the other disks you may have in your subscription.

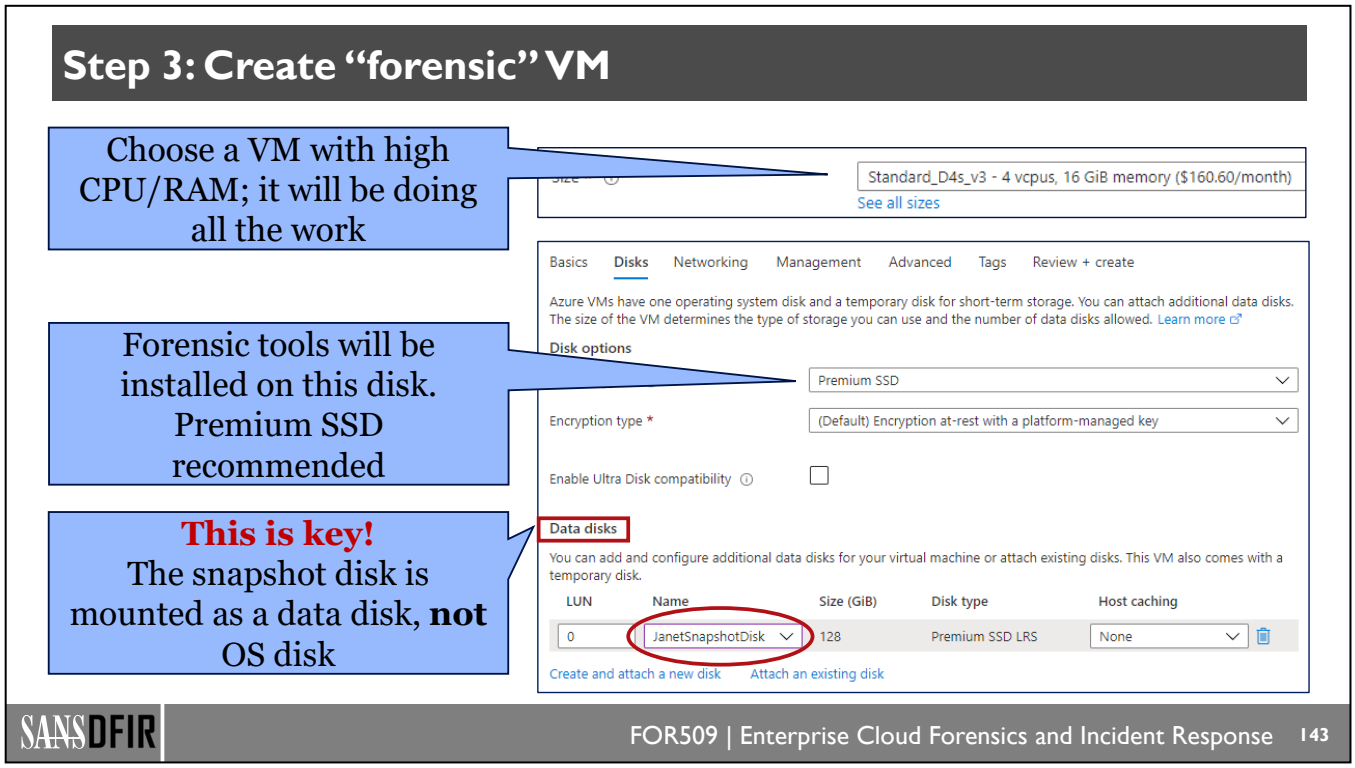
Instead of creating a blank disk, we are specifying a source type of "Snapshot" as well as the name of the snapshot. This way, the disk will be created with all the snapshot data on it.

For this disk, it's worth getting Premium SSD, as we will be running our forensic tools against it. A fast disk will help you process the data faster.

If you need to save on cost, at this point you could delete the snapshot. We strongly recommend against it, as you may need to repeat this procedure if the data gets corrupted on the disk. Remember that while the snapshot is read-only, this disk isn't.

© 2022 Pierre Lidome

Technet24



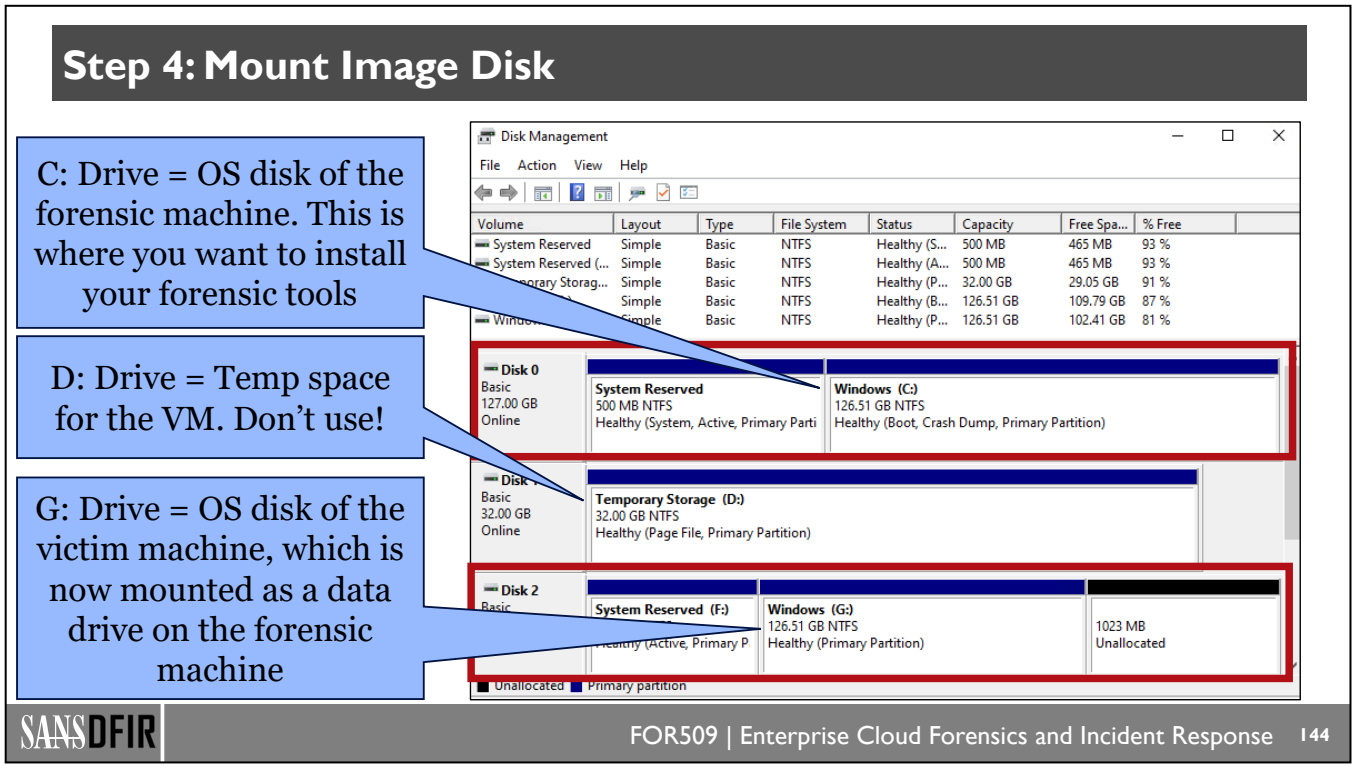
The "forensic VM" is the VM that will do all the work, so it should be created with robust CPU and memory. The exact specifications will depend on the forensic software you choose, but in general 4 vCPUs and 16GB of memory should provide plenty of horsepower. Be sure to create this VM in the same region as the snapshot disk from the previous step.

This VM will be created with its own OS disk. This is the disk where you will install your forensic software and store the results. Premium SSD is recommended to optimize the performance of the VM. Azure will provision a 128GB OS disk with over 100GB of free space, which should be plenty for your needs.

Here is the critical part: under "Data disks," you will select **"attach an existing disk"** in order to mount the disk that we created in the previous step. If you forget to specify the data disk during the VM creation, you can add it afterward. We strongly recommend that you shut down the VM before attaching a data disk. VMs have been known to get corrupted if you add and remove a data disk while the VM is running.

During the disk creation step, you will have an option to create the VM. Don't do it! If you do, the VM will be created with the snapshot disk as its OS disk. In effect you will just be cloning the "victim VM."

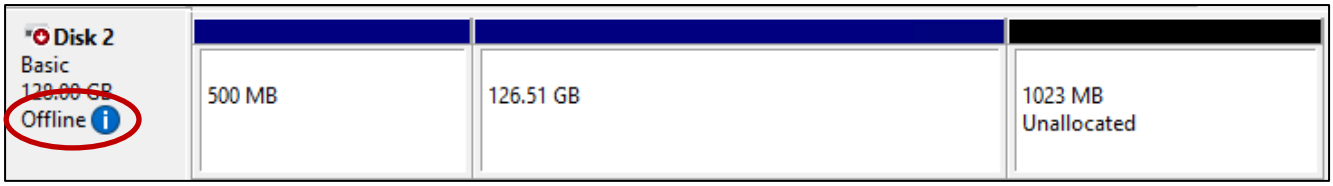
© 2022 Pierre Lidome



Now that the VM is running, you will see three disks under Disk Management:

1. Disk 0: The VM's OS disk (C: drive) plus the typical "system reserved" partition for Windows.
2. Disk 1: Temporary storage for the VM (D: drive). You may use it, but it's not a persistent disk, so don't put anything important on it.
3. Disk 2: A perfect copy of the "victim VM" OS drive. As you would expect, it contains the "system reserved" partition plus the OS partition. We are interested in the OS partition, which is mounted as the G: drive in this example.

When you first start Disk Management, you will see that Disk 2 is "Offline." Right-click on it and select **"Online."** Windows will then assign a letter to each partition.

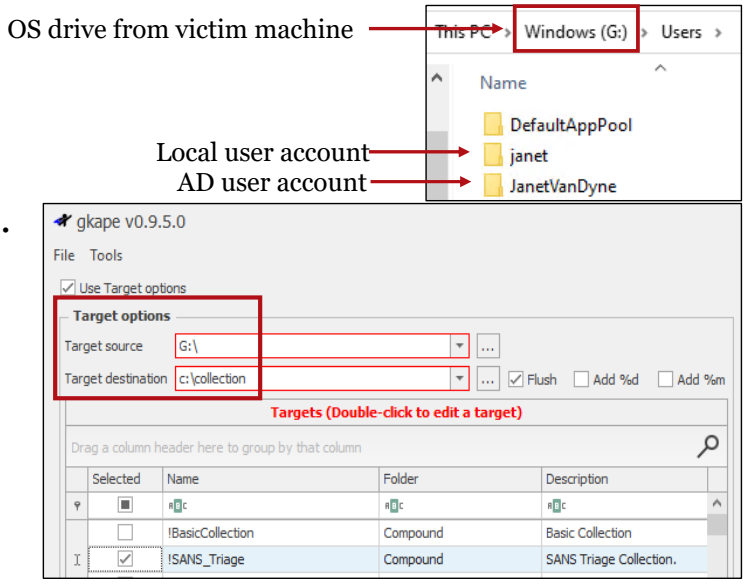


Remember that all drives are writable. If you corrupt the G: drive by mistake, you can repeat the process from step 2 since the original snapshot is forensically sound.

© 2022 Pierre Lidome
Technet24

Step 5: Run Forensic Tools

- G: drive contains the user directories from JanetDesktop, as expected.
- Install and run your favorite forensic tools, such as KAPE.
- Point your forensic tool to the G: drive to collect data.
- Store your results on the C: drive so you don't overwrite your evidence (restore from snapshot if you accidently do).



You are now ready to run your favorite forensic tool, as taught in FOR500 and FOR508.

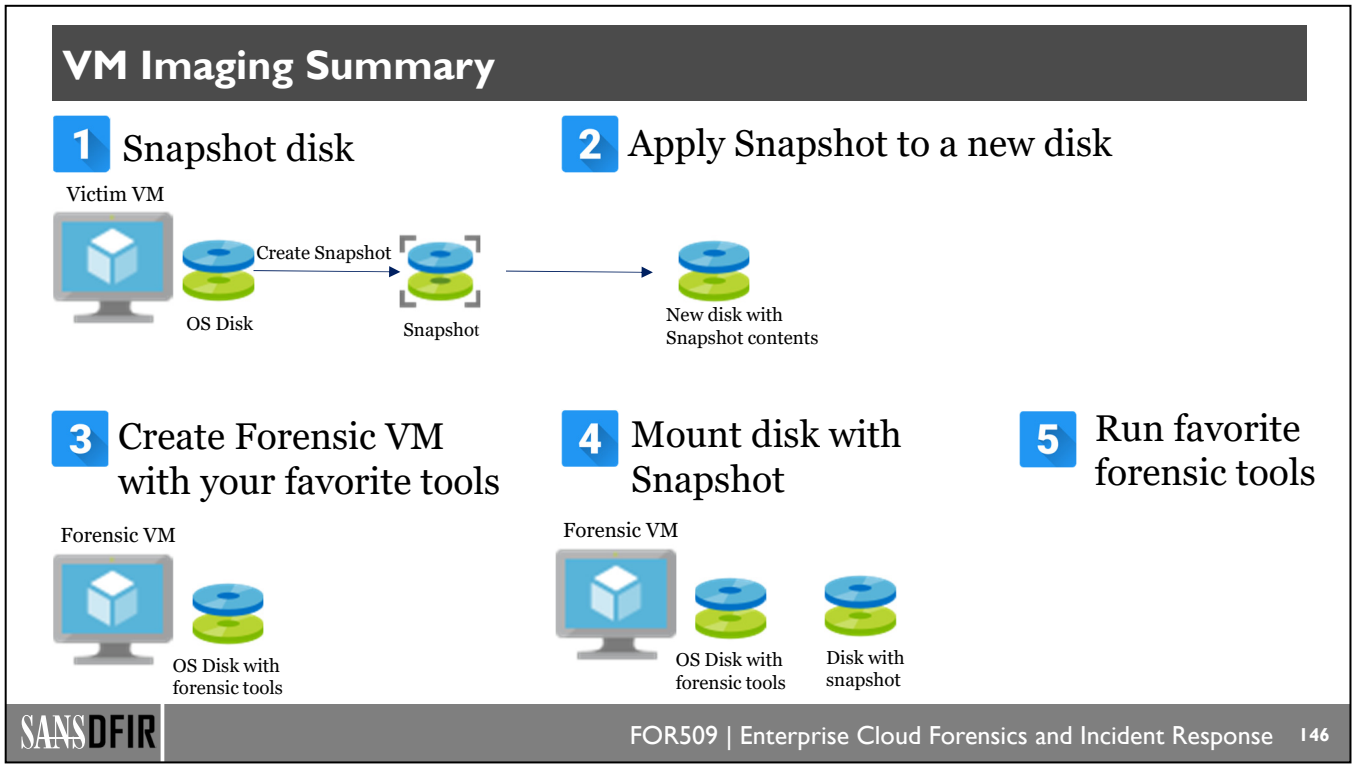
From a process standpoint, we would recommend running KAPE[1] with the SANS Triage Collection option to extract key files from the G: drive. You can store these files in the C: drive or the D: drive and process them with Eric Zimmerman's excellent suite of forensic tools.[2]

In case you haven't heard of KAPE, it stands for Kroll Artifact Parser and Extractor. KAPE is a triage program that collects the most forensically relevant artifacts from a target. Optionally, it can also parse this data and run analysis programs against it.

Using this process avoids the egress charges of downloading an entire VM to your local computer. You have the option of simply downloading the output for your forensic tools or the data collected by KAPE. This is a much smaller and therefore less costly amount of data to transfer compared to an entire VM.

1. https://for509.com/kape
2. https://for509.com/ztools

© 2022 Pierre Lidome



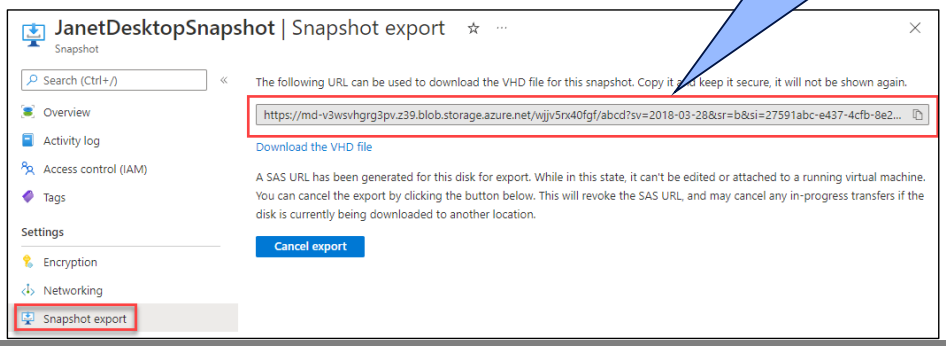
To summarize the process, you will:

1. Create a snapshot of the victim's OS disk. Some investigations may also require you to snapshot the data disk(s).
2. Create a new disk based on the snapshot so that all the information is written to that disk.
3. Create a new VM, which we call the Forensic VM, with its own OS disk. Be careful not to create this VM based on the snapshot you just created. If you do that, you will simply create a clone of the victim VM. During this step, you should also install your favorite forensic tools on the VM.
4. Mount the disk that contains the snapshot as a data disk on the Forensic VM. This step may also be performed during the Forensic VM creation.
5. Run your favorite forensic tools as you have learned in FOR500 and FOR508.

© 2022 Pierre Lidome
Technet24

Alternative Snapshot Access

- Option to download the snapshot VHD
- Can be downloaded to any destination
- May be beneficial for some analysis tools
- Can be time consuming depending on size
- May result in egress network cost



An alternative to the process described in the prior slides is to download the snapshot virtual hard disk (VHD). Certain forensic tools may benefit from having direct access to the VHD.

Another advantage of having a copy of the VHD is the ability to implement automation processes. We will show an example of an automated DFIR process in the AWS section.

On the other hand, since the VHD will be the same size as the allocated disk of your VM, it's likely to be quite large. Unfortunately, the VHD will include all blank space and isn't compressed during the download. As a result, the time to download the VHD can be substantial. Finally, if you download the VHD to a destination outside of Azure or even in a different Azure region, you will incur network egress charges.

If you choose to download the VHD, simply select your snapshot by name and choose the "Snapshot export" menu. You will be prompted to choose an amount of time during which the download URL will be valid. Once you select "Generate URL", you will be shown a one-time URL that allows you to download the VHD. You can access that URL from anywhere on the internet provided that the snapshot's connectivity method is set to public endpoint (you can change that setting in the "Networking" menu).

© 2022 Pierre Lidome

Azcopy Snapshot

- Web transfers can be slow, azcopy is a much better option
- Azcopy is multi-threaded and significantly faster
- Azcopy is a free application from Microsoft (link in the notes)
- <snapshot URL> = URL from snapshot export menu
- Disable MD5 check or transfer will fail final check (per documentation)

```
azcopy cp "<snapshot URL>" "c:\temp\snapshot.vhd" --check-md5 nocheck
```

```
E:\>azcopy cp "https://md-lxdrq5g50drf.z37.blob.storage.azure.net/fslvdc2w41pz/abcd?sv=2018-03-28
&sr=b&si=fe3445e2-20ef-413e-b408-1a4ea85c3af1&sig=bxbxYwh8Sl2xgOav8ry5rzNHa6ukYZ6d%2BiYrbLD%2FRlA
%3D" "e:\elk-snapshot.vhd" --check-md5 nocheck
INFO: Scanning...
INFO: Any empty folders will not be processed, because source and/or destination doesn't have ful
l folder support

Job 404785c0-77d1-dd4f-4599-b445733b2709 has started
Log file is located at: C:\Users\elk_user\.azcopy\404785c0-77d1-dd4f-4599-b445733b2709.log

3.7 %, 0 Done, 0 Failed, 1 Pending, 0 Skipped, 1 Total, 2-sec Throughput (Mb/s): 499.6588
```


The snapshot export menu will provide a URL which can be used to download the snapshot. While the snapshot can be downloaded via a web browser, it will be very slow.

As discussed in the blob slides earlier in this class, Microsoft provides a utility called azcopy[1] which is available at no cost from Microsoft's website. This utility is significantly faster as it's multi-threaded.

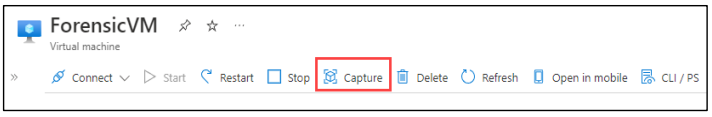
As an example, a 64GB transfer with azcopy (within the same Azure region) will take less than 20 minutes. The same download with a web browser can take close to 1 hour.

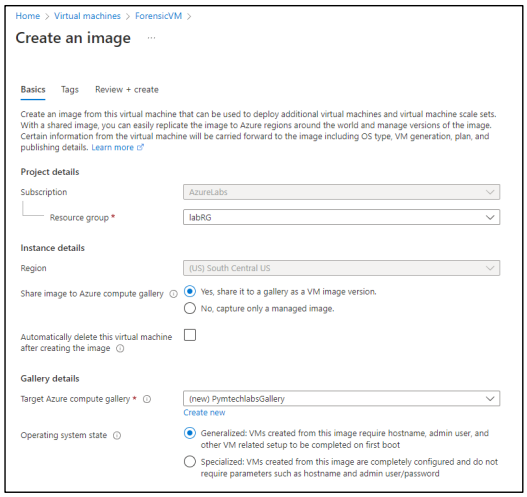
1. https://for509.com/azcopy

© 2022 Pierre Lidome

Technet24

Forensic VM Image Creation





- You don't need to create a new Forensic VM for each investigation
- Azure has the option to capture an image of an existing VM and import it to the Azure gallery to create future VMs
- Another option is to move a VM to a different resource group, subscription, or region, but that feature is unreliable

Azure has an option to create an image from a current VM and share the image in the Azure compute gallery. This will provide the ability to create future VMs based on that image. This feature can be leveraged by creating a VM, installing all the required forensic tools, and then storing an image of that VM in the gallery.

An alternative option is to start from scratch and use the Azure Image Builder to create your own image. This is a more complex procedure.

Finally, Azure has an option to move a VM to a different resource group, another subscription, or another region. Therefore, once you create your forensic VM and install your favorite tools, you can simply move that VM around as needed. Unfortunately, this method has been found to be unreliable.

© 2022 Pierre Lidome

Other Azure Resources

There are many other Azure resources available to help you. Some have additional costs, some are free.

- Azure Sentinel ($)
- Azure SimuLand
- Microsoft Incident Response Playbooks
- Recommended Projects

Other services offered by Azure might be helpful to you if they have been purchased and implemented by your organization.

There are a two interesting articles written by Sonia Cuff that discusses the relationship between Azure Security Center, Azure Defender, and Azure Sentinel:

- https://for509.com/techarticle1
- https://for509.com/techarticle2

This class is about the cloud resources and logs you can leverage to perform incident response and forensics. We focused on the features that are included with every tenant. The next few slides will describe optional features so that you are aware of them as well.

There are also a number of open-source projects worth exploring.

© 2022 Pierre Lidome

Technet24

Azure Sentinel: SIEM/SOAR

Cloud-native SIEM/SOAR

- Can ingest data from multiple clouds & on-prem infrastructure
- Detects and correlates threats using artificial intelligence
- Built-in orchestration to automate tasks

Sentinel is priced per GB of data ingested, making it a significant investment.

Microsoft Azure Sentinel is a scalable, cloud-native, security information and event management (SIEM) and security orchestration automated response (SOAR) solution.[1]

The key features are:

- Cloud-scale SIEM/SOAR. Sentinel is not limited to Azure. It can ingest data from multiple clouds as well as on-prem infrastructure.
- Sentinel leverages Microsoft threat intelligence to detect threats.
- Sentinel uses artificial intelligence to analyze and correlate threats.
- Sentinel includes built-in orchestration to automate common tasks.
- Sentinel leverages the MITRE framework to enable you to proactively hunt for threats.

Sentinel uses a Log Analytics workspace and implements numerous queries to hunt for potential threats. Here are some examples of pre-built hunting queries:

★	Abnormally long DNS URI queries	Microsoft	DnsEvents
★	DNS Domains linked to WannaCry ransomware campai...	Microsoft	DnsEvents
★	Cobalt Strike DNS Beaconing	Microsoft	DnsEvents +1
★	Failed service logon attempt by user account with avail...	Microsoft	AuditLogs +1
★	Failed Login Attempt by Expired account	Microsoft	SecurityEvent +1
★	Multiple Password Reset by user	Microsoft	AuditLogs +4

© 2022 Pierre Lidome

Sentinel is priced per gigabyte of data ingested. The pricing depends on the region. A large infrastructure will likely generate terabytes of data per day, making a tool like Sentinel a significant investment.

1. https://for509.com/sentinel

© 2022 Pierre Lidome

Technet24

Azure SimuLand

- Open source initiative by Microsoft
- Facilitates deploying lab environments to reproduce well-known attack techniques
- Verifies the effectiveness of Microsoft 365 Defender, Azure Defender, and Azure Sentinel detections
- Announcement on Microsoft security blog and project code on GitHub (see links in the notes)
- Picture shows the process integration of SimuLand with threat research

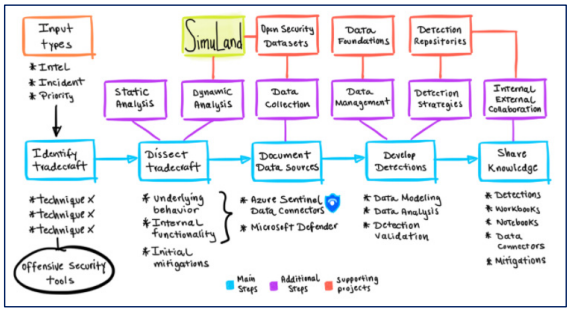


Image Credit: Microsoft (see reference 1 in the notes)

From the Microsoft security blog:

"SimuLand is an open-source initiative by Microsoft to help security researchers around the world deploy lab environments that reproduce well-known techniques used in real attack scenarios, actively test and verify the effectiveness of related Microsoft 365 Defender, Azure Defender, and Azure Sentinel detections, and extend threat research using telemetry and forensic artifacts generated after each simulation exercise."[1]

The code is available on the Azure/SimuLand GitHub.[2]

1. https://for509.com/simuland-blog
2. https://for509.com/simuland-github

© 2022 Pierre Lidome

Microsoft Incident Response Playbooks

Microsoft has published four incident response playbooks:

- Phishing, which provides guidance on identifying and investigating phishing attacks in Microsoft 365
- Password spray investigation
- App consent grant investigation, which focuses on Graph API
- Compromised and malicious applications

Playbooks are very detailed and highly recommended.

Microsoft has published four incident response playbooks:[1] phishing, password spray investigation, app consent grant investigation, and compromised and malicious applications.

For each paybook, Microsoft provides a detailed explanation of the risk and methods to identify the potential attacks.

These playbooks require a good understanding of Azure and are highly recommended.

1. https://for509.com/incident-playbooks

© 2022 Pierre Lidome

Technet24

Recommended Projects

PowerZure from Ryan Hausknecht (@Haus3c)

- PowerShell project to assess and exploit resources in Azure.
- `https://github.com/hausec/PowerZure`

MicroBurst from Karl Fosaaen (@kfosaaen), NetSPI

- Scripts to discover Azure services and audit configuration for weaknesses.
- Post-exploitation actions such as credential dumping.
- `https://github.com/NetSPI/MicroBurst`

Azure AD Connect Password Extraction from Dirk-jan Mollema (@_dirkjan), FOX IT

- Script to extract and decrypt stored credentials from Azure AD Connect servers.
- `https://github.com/fox-it/adconnectdump`

Karl Fosaaen with NetSPI has a great presentation regarding his project:
https://notpayloads.blob.core.windows.net/slides/ExtractingalltheAzurePasswords.pdf

© 2022 Pierre Lidome

Course Resources and Contact Information

Here is my lens. You know my methods. —Sherlock Holmes

AUTHOR CONTACT
Pierre Lidome
plidome@gmail.com
Twitter: @texaquila

SANS INSTITUTE
11200 Rockville Pike, Suite 200
North Bethesda, MD 20852
301.654.SANS(7267)

DFIR RESOURCES
digital-forensics.sans.org
Twitter: @sansforensics

SANS EMAIL
GENERAL INQUIRIES: info@sans.org
REGISTRATION: registration@sans.org
TUITION: tuition@sans.org
PRESS/PR: press@sans.org

This page intentionally left blank.

© 2022 Pierre Lidome

Technet24