Threater Enforce in Azure - Gateway Load Balancer

November 01, 2023

1. Overview

This document provides end-to-end guidance for a typical deployment of Threater Enforce software leveraging a **Gateway Load Balancer** in Azure, including a working example configuration.

2. Prerequisites

It is possible that you may be reading an old version of this document. We recommend that you check the following publicly available link to make sure you have the most recent copy of this document:

https://threaterproduction.blob.core.windows.net/docs/Threater+Enforce+in+Azure.pdf

This document will be most useful to readers who are IT and/or cyber security professionals with:

- A solid understanding of computer networking.
- At least a cursory understanding of Azure.
- At least a cursory familiarity of Threater Enforce and the Threater portal.

Additionally, although it is not explicitly required, readers who have a deep working knowledge of standard Azure principles, including resource groups, security groups, subnets, routing, Internet gateways, and instance management will gain the most benefit from this document.

This is because cloud deployments can feel overwhelming at first for IT personnel who have never managed cloud services. There is much to learn!

The following Azure deployment prerequisites are also important:

- An existing Azure account with Azure portal access.
- Sufficient permission to perform deployments via Azure ARM.

IMPORTANT: Keep in mind that the tools and instructions presented here only demonstrate one possible use-case of Threater Enforce in Azure; they are not a "turn-key" or "one size fits all" solution for securing your Azure infrastructure. The goal of this document is to provide you with the information and tools necessary so that you can integrate Threater Enforce into the specific needs of your cloud security stack.

In the following document it is important to recognize some of the subtle differences in terminology:

- **Enforce:** Threater provided **software only** component of deployment
- **Enforcer:** Deployed **instance** running Enforce software (including host and operating system); can be virtual or on-premises

Lastly, it is worth mentioning some abbreviations you may encounter in this document:

- **ARM:** Azure Resource Manager
- **ASN:** Autonomous System Number
- **BYOL:** Bring Your Own License
- **DHCP:** Dynamic Host Configuration Protocol
- **DPDK:** Data Plane Development Kit
- **EULA:** End User License Agreement
- **GDPR:** General Data Protection Regulation
- **GWLB:** Gateway Load Balancer
- **HA:** High Availability
- **IOC:** Indicator of Compromise
- **JSON:** JavaScript Object Notation
- **NAT:** Network Address Translation
- **RFC:** Request For Comments (Internet Specification Document)
- **SIEM:** Security Information and Event Management
- **SLA:** Service Level Agreement
- **TLS:** Transport Layer Security
- **UI:** User Interface
- **VNet:** Azure Virtual Network
- **VPN:** Virtual Private Network
- **VXLAN:** Virtual Extensible LAN

3. Note About Customer Data Retention and Privacy

Before going further it is important to mention that customer configuration data is retained and managed by the deployed Threater Enforce software. This includes information such as local administrator usernames and passwords, as well as detailed connection logging information. It also includes non-customer-specific information, such as standard out-of-the-box threat feeds and related threat intelligence.

Any and all "customer" specific attributed information is transmitted nowhere else, ever, until and unless the customer decides to do so. For example, it is common for advanced customers to choose to export our RFC-compliant logs data to any number of third-party SIEM tools of their choosing. Common connectivity that we see customers use include connectors to systems like Splunk, IBM QRadar, and Gravwell. For those who are interested, we talk more about such configurations later in this document when we briefly discuss software configuration, akin to what our customers have come to expect from our on-premise deployments.

And, of course, the protected instances should always be considered by the end customer as points of presence for customer data. After all, you're installing Threater Enforce for a reason, and one primary reason is protecting your data stored/used in any number of instances sitting behind an Enforcer protection point! That's the beauty of Threater Enforce running both in Azure and on-premise: they can protect anything, anywhere, seamlessly, operating effectively as a bump-in-the-wire.

We take great pride in collecting as little information as possible about our customers and even when we do have reason to collect customer-attributable information (for example, detailed access or logging information), it is transmitted nowhere at all without the customer taking action. Our customers decide where their data goes, always, without fail. This has allowed us to do very well in places in the world where privacy is at a premium, such as the European Union (GDPR, etc.), where we have many customers.

4. Threater Enforce in Azure (Gateway Load Balancer)

Threater Enforce is the only active defense cybersecurity platform that fully automates the enforcement, deployment, and analysis of cyber intelligence at a massive scale. As the foundational layer of an active defense strategy, our patented solution blocks known threats from ever reaching your networks. Threater Enforce utilizes immense volumes of cyber intelligence from over 50 renowned security vendors to provide unparalleled visibility over the threat landscape resulting in a more efficient and effective security posture. Security teams at companies of all sizes use Threater Enforce to deploy active security, gain real-time network

visibility into threats and policy violations, ensure their network is protected, and reduce manual work.

Threater Enforce:

- Deploys as a standard Azure image with Threater Enforce software pre-installed and a **BYOL subscription model.**
- Allows Threater patented technology to protect your Azure infrastructure by allowing and/or blocking incoming and/or outgoing packets in real-time, based on policy and list configurations.

Configuration is managed entirely in the cloud via the Threater portal which is hosted at **https://portal.threater.com** and provides centralized management of all Enforcer instances regardless of whether they are deployed on-premise, in the cloud (such as Azure), or both. The platform features always-on control and synchronization of geo-IP data, ASNs, allowed lists, denied lists, threat lists, policies, and more in real-time. Threater Enforce provides best-in-class protection with no measurable impact to network performance, regardless of the number of IOCs that you are protected against. Any changes in configuration of any type, including list contents and policies, are always propagated in real time to all Threater Enforce software installations, whether they are on-premise or in the cloud.

5. BYOL Support and Pricing

Our **BYOL** Threater Enforce customers with active subscriptions are able to receive various levels of support. As our support plans can vary over time and in some cases from subscription type to subscription type, we do not embed tier descriptions or SLAs and the like in this deployment document. Instead, we refer the reader to our online corporate website documentation, with a good starting point being the following link:

https://support.threater.com

Our software subscription pricing is identical for both our on-premise and cloud deployments. For detailed pricing information for standard BYOL subscriptions, please see our support link above. Existing on-premise customers looking to leverage our solutions in Azure can of course contact their existing Threater Sales Representative for more complex configurations. Also, it is trivial from within our portal to move BYOL subscriptions from existing on-premise devices to new cloud instances.

6. Threater On-Premise vs. Threater in Azure

Traditional on-premise Threater Enforce software installations operate as a layer 2 bump-on-the-wire. Anything arriving at the "inside" port will be propagated to the "outside" port and vice versa, unless the packet is blocked due to your configured policy and list configurations.

In the cloud, things are a bit different, since you don't have full control of the underlying networking like you do in on-premise environments. Fortunately, The Azure Gateway Load Balancer together with a virtual appliance running Threater Enforce software can be leveraged to provide the same level of protection as on-premise deployments. Using the Gateway Load Balancer Azure resource, Threater Enforce can efficiently examine ALL inbound and outbound traffic of any public IP exposed from within Azure.

7. Integration with the Threater Portal

Our Threater Enforce software in Azure interacts with the Threater portal in exactly the same way that our on-premise deployments do, with no exceptions.

All of our Threater Enforce deployment paradigms leverage exactly the same codebase.This is very different from offerings from other security vendors in the Marketplace who had to create unique form-feature-function products between their on-premise and cloud offerings. Because of our patented architecture, we didn't have to do that. It's exactly the same.

This was achievable for us since our Threater Enforce on-premise and Azure-based design is based on a Linux software stack leveraging a bump-in-the-wire network architecture which further leverages DPDK, which deploys beautifully on both standard on-premise equipment as well as into Azure.

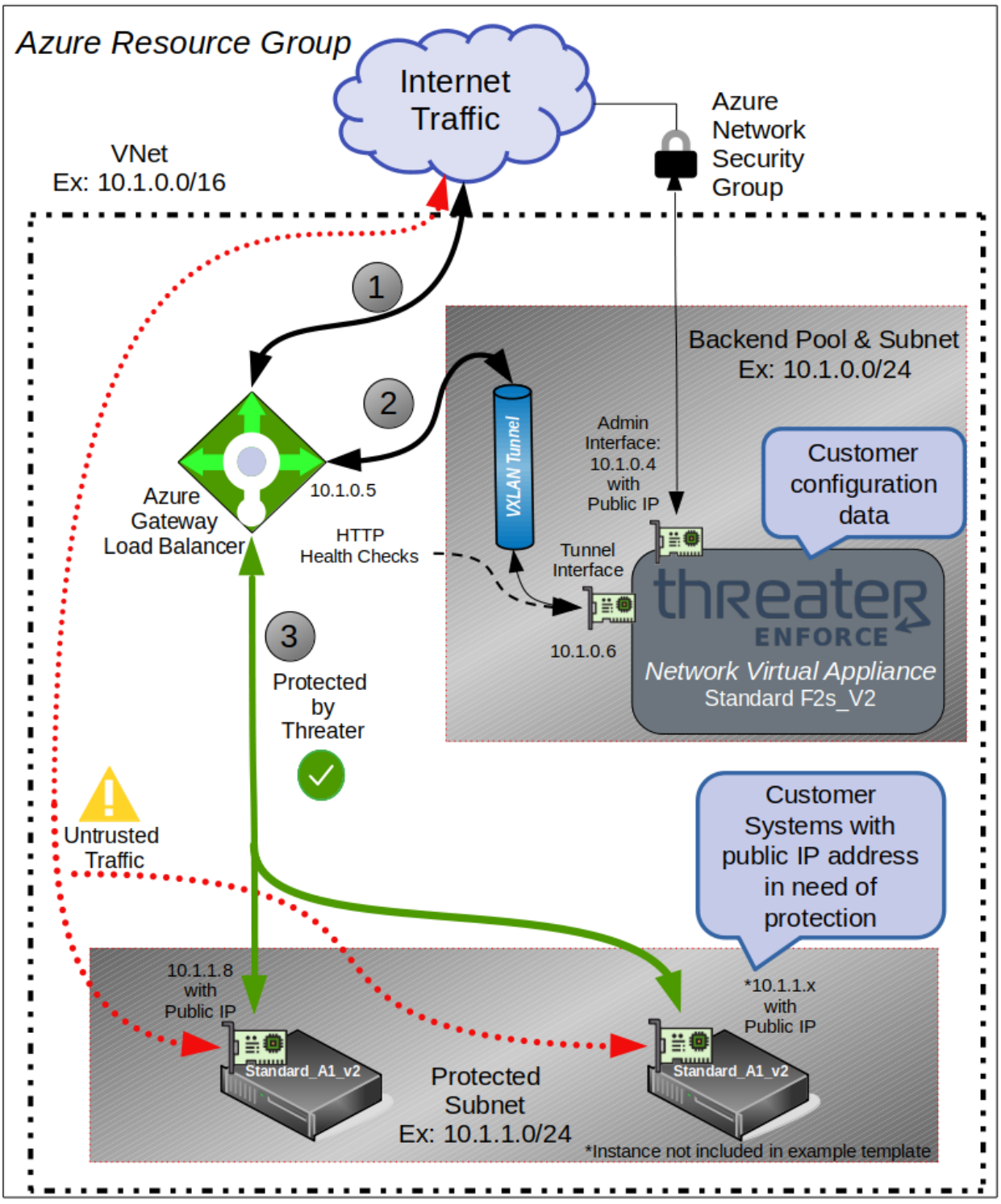
Our customers use the same Threater portal for management whether they are deployed on-premise, or in the cloud, or any combination thereof, without having to make any distinction between on-premise or cloud-based deployments.

8. Example Network Deployment Diagram

The following diagram demonstrates a simple Azure Gateway Load Balancer deployment that features two instances that each expose a public IP address. In this example both instances are protected by a single Enforcer instance. Note that the following diagram is simply an adaptation of the example found in the Azure ***Gateway Load Balancer Overview*** located here:

https://learn.microsoft.com/en-us/azure/load-balancer/gateway-overview

The VNet is assigned a private address space of 10.1.0.0/16. You are free to choose any valid Azure VNet address space you like. If your scenario warrants modifying this, you must of course adjust all other private IP mappings for the various associated subnets so that they fall within the constraints of the network you choose to deploy.



Once the Enforcer is deployed, it will include two interfaces: the standard administration interface and a **VXLAN Tunnel** interface. These are discussed in more detail later in the document.

The diagram above shows how the Gateway Load Balancer coupled with an Enforcer instance operates:

1.	The Gateway Load Balancer Intercepts incoming (untrusted) traffic before it arrives at any instance interface. This is shown in the path labeled with circle (1).
2.	The Gateway Load Balancer then VXLAN encapsulates the traffic and forwards it to the Enforcer where it is de-encapsulated for examination and either dropped or returned to the Gateway Load Balancer; this step is labeled with (2). All packets returned to Gateway Load Balancer are VXLAN re-encapsulated before transmission back.
3.	The Gateway Load balancer then de-encapsulates traffic received from the Enforcer and forwards it to the original destination. This is shown in the path labeled with circle (3).

The same rules apply for traffic originating from protected VM instances but in reverse.

8.1. Threater Enforce Interfaces

8.1.1. Tunnel Interface

VXLAN encapsulated traffic from the Gateway Load Balancer arrives for inspection at the Enforcer "tunnel" interface. Note that the Gateway Load Balancer ensures flow stickiness by routing both forward and reverse flow packets to the same Enforcer instance in cases where two or more Enforcer instances are in use.

From the perspective of an end-user, the VXLAN encapsulation is mostly irrelevant as it only applies to traffic that flows between the Gateway Load Balancer and the Enforcer instance. The Threater Enforce is equipped to handle the encapsulation "out-of-the-box" so there is nothing that needs to be configured on the instance itself. When the Enforcer boots, it will recognize that it is paired with an Azure Gateway Load Balancer and take care of managing the encapsulation for you.

One minor exception here is the configuration of VXLAN Port and Id fields. These will be configured as part of the Threater Enforce License and Configuration step below.

Note that the Enforcer tunnel interface handles BOTH inbound and outbound traffic. With an Azure Gateway Load Balancer, there is no concept of physical inside/outside bridging pairs as exist with on-prem deployments. Fortunately, Azure's Gateway Load Balancer, identifies the direction for us via a unique VXLAN identifier in the encapsulation header. Threater Enforce uses this VXLAN identifier to determine packet direction.

In our example we apply a security group to the tunnel interface so that only the following services are permitted. We recommend keeping these Azure NSG rules in place in production environments as well.

- UDP on port 10800 outbound (VXLAN tunnel traffic)
- UDP on port 10801 inbound (VXLAN tunnel traffic)
- TCP on port 80 (Health Probes from Gateway Load Balancer to the Enforcer Tunnel interface

8.1.2. Admin Interface

The Threater Enforce standard administration interface is the default interface that the instance receives when it is deployed. In our example, we've specified our appliance admin target subnet as 10.0.0.0/24 so the Enforcer will be assigned a private IP in this address range.

You can choose to assign your Threater Enforce admin connection a public-facing IP on Azure. We do that in our example configuration, and as such we will make absolutely sure to lock down access via a proper Azure security group definition so that administration access is available only to individuals and systems who should have access. However, at a minimum you will need to open HTTP (port 443) for inbound connections so that the Enforcer can be managed via its user interface.

Alternatively, if your IT staff is knowledgeable about advanced Azure configurations, it is certainly wise to disavow a public IP altogether, and use a properly configured VPN to access your VPC's administration subnet. Those and other more advanced configurations are beyond the scope of this document, and if they are of interest to you, we recommend that you contact Azure directly for assistance and training as needed.

8.1.3. Health Probes

The Gateway Load Balancer will also issue heath probe checks on configurable intervals to ensure that associated Enforcer instances are operating correctly and can accept new flows. Note that these probe messages are sent to the "tunnel" interface -- not the admin interface. When the Threater Enforce software is operating normally (ready to accept traffic), it will respond as healthy and the Gateway Load Balancer will be free to send it new flows for inspection. Important: The Threater Enforce software only responds to HTTP (port 80) health probe requests at the path:

/api/v1/healthcheck/gwlb

Threater Enforce will not respond to any other health probe configuration so it is absolutely imperative that the health check is configured correctly in Azure. If health probes are not properly configured then connections to protected instances will likely be unreachable as the Gateway Load Balancer will determine that there is no appliance available to handle new connections.

9. Deployment Time Guidance

The deployment time will likely vary from customer to customer, depending on your level of expertise and familiarity with the cloud, and depending on whether you are building out everything from scratch.

Generally, by using the Azure ARM template we provide in a subsequent section, deployment will take anywhere from 15 to 30 minutes, total.

10. Obtaining the Enforcer Image

10.1. Azure Enforcer Image

We currently do not support Enforcer deployments via the Azure commercial marketplace. There are several reasons for not doing so. First, Azure Marketplace is complex to configure and it includes marketing capabilities and third-party integrations that are not necessary for us to deliver Threater Enforce to customers. In addition, there would be significant effort and cost required for us to replicate and maintain copies of the latest released image in all Azure regions where customers want to deploy Threater Enforce.

Currently, the most efficient means of delivery is via an Azure generalized VHD image which customers can import into their own Azure subscription and deploy or replicate the image as needed. The latest generalized Enforcer VHD can be obtained here:

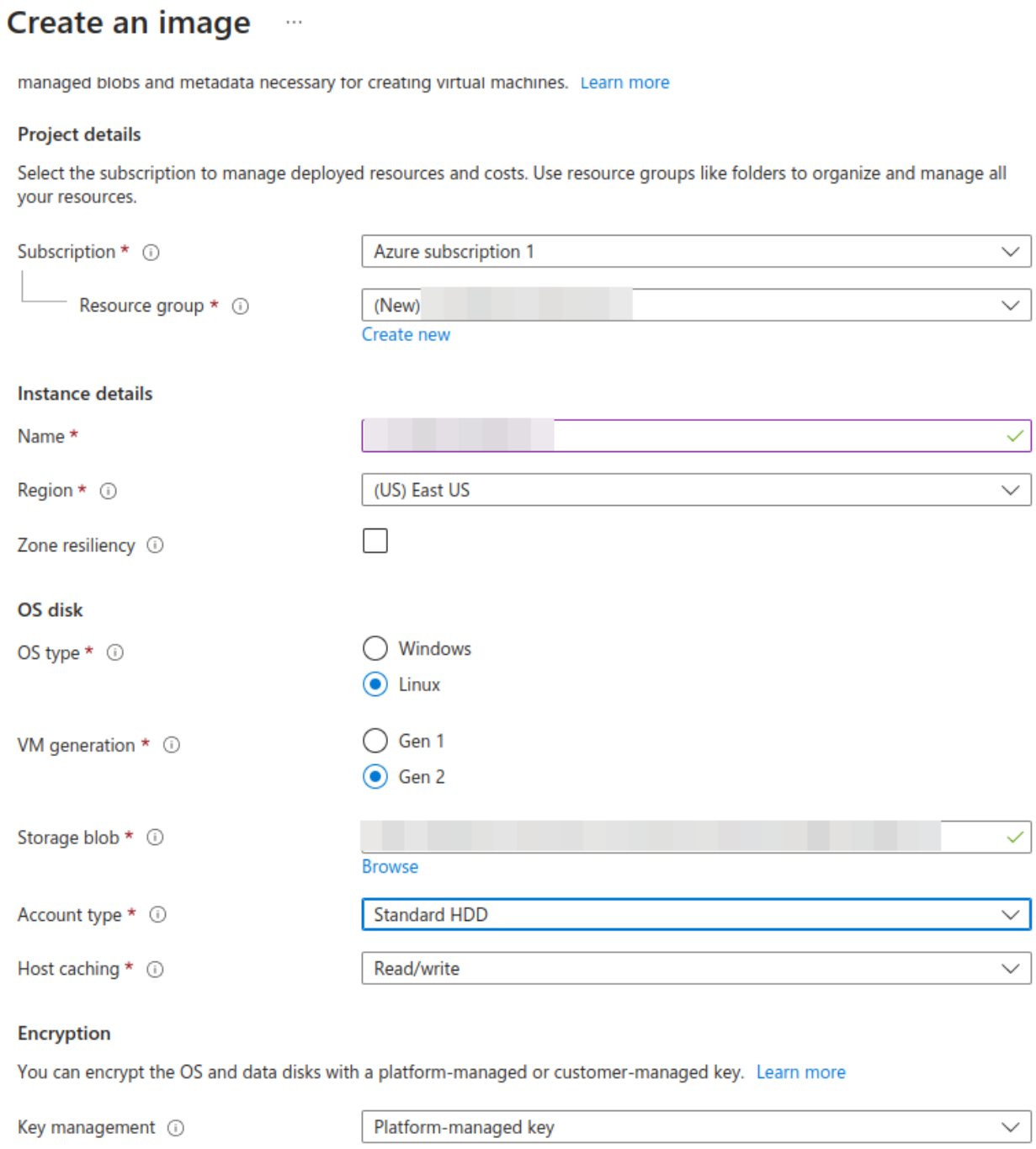
https://threaterproduction.blob.core.windows.net/vhd/ThreaterEnforce.vhd

Note: This link always provides a VHD of the latest released Enforcer image.

While the details of importing a VHD into Azure are beyond the scope of this document, the basic steps would be as follows:

1.	Create an Azure Storage Account with a container that can store your copy of the VHD file.

2.	Ensure that the container in the previous step has the role: "Storage Blob Data Owner."
3.	Install (if not already installed) the Azure command line tool 'azcopy' and perform: 'azcopy login' and follow the authentication steps.
4.	Once authenticated, use 'azcopy' to copy from the link above to your container as follows: azcopy copy \ 'https://threaterproduction.blob.core.windows.net/vhd/ThreaterEnforce.vhd' \ 'https://<storage_account>.blob.core.windows.net/<container>/<dest_name>.vhd' NOTE: this will copy directly across Azure infrastructure and is fast.
5.	Verify that you now have a copy of the VHD file in your container.
6.	The last step is to create a deployable Azure image from the VHD file. To do this, enter “images” in the Azure portal search bar and open “Images->Create Image.”
7.	You should see the dialog presented below where instance details are specified as follows: a. Select an existing resource group or create a new one for the image. b. Create a name for the image and select a region. c. Ensure that the OS type is Linux and VM Generation is Gen 2. d. Select the storage blob VHD file that we copied in previously from the drop-box. e. The remaining settings can remain as demonstrated below.



<u>IMPORTANT</u>: Please note that the image we share with you will have all of the Threater Enforce software pre-installed and will be ready for final configuration. However, it does not include the required BYOL software subscription component:

- It includes only the Azure infrastructure/hardware component of the instance deployment. For example, at the time this document was last updated, the

recommended deployment for the Threater Enforce is a **F2s_v2 (2 core / 4GB RAM)** instance type. When spun up in region **East US (Virginia)**, it costs **$0.085/hr** (on-demand pricing as of October 2023), which is a 24x7 annualized cost of **$744.60/yr** Threater, does not see any of that money - that goes entirely to Azure.

- Bandwidth charges are also billed separately to your account by Azure, and that money too goes entirely to Azure.
- The Threater Enforce subscription must be purchased separately and directly from Threater as a separate software subscription, which leverages an identical software subscription pricing model that we use for on-premise deployments.
- Without the BYOL subscription attached via the Threater portal, the Enforcer instance that you deploy will stay in a special **allow all** mode and will just blindly forward packets without performing any packet protection or logging. This means that even without a BYOL subscription, you will still be able to complete the example configuration, but the result will be that all traffic is allowed to pass in both directions. None of the traffic will be logged, and none of the traffic that we would have detected as malicious will be blocked until you install a valid subscription for each Enforcer instance. The subscriptions must be obtained directly from Threater and attached to each deployed Enforcer using the Threater portal.

11. Deploy via ARM Template

Now that we have described the deployment resource group in detail and have access to the proper VM Image we are ready to deploy a working example.

11.1. Why ARM Templates?

Our deployment method opts for the use of JSON formatted Azure ARM templates to perform the deployment. While it is certainly possible to deploy entirely via the Azure CLI or Azure portal, deployment via ARM templates is much more seamless and is by far the easiest way to build the example(s). This is because the templates manage all resource dependencies and you don't need to be an expert with the CLI or portal to use them.

Any ARM templates provided here are under the MIT license and you are fully within your rights to modify it in any way you may require. Azure ARM templates are simple JSON documents and can be modified with any text editor.

You can learn more about Azure ARM templates here:

https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/overview

Note that the template only deploys one Enforcer instance in the backend pool and one protected instance in the protected subnet. This is to keep things simpler and also to keep Azure infrastructure costs down. Adding either more protected instances or more Enforcer instances to the templates is perfectly fine though.

11.2. The ARM Template

The ARM Template can be downloaded from this link:

https://threaterproduction.blob.core.windows.net/templates/enforcer-deploy.json

11.2.1. Update ARM Template Parameters

After the ARM Template is downloaded, we will need to edit it and fill in some parameters that are unique to your Azure account. The parameters that require update are labeled with "CHANGEME" and the "description" field of each parameter thoroughly explains what needs to be filled in. Therefore, the purpose of each parameter is not discussed here. Note that all of the template parameters that need to be configured are at the top of the file and can be easily identified. All parameters pre-filled with "CHANGEME" must be populated correctly; they cannot be left as-is or empty.

11.2.2. Create an Azure Resource Group

The ARM template we will deploy in the following step will require an Azure Resource Group for the template resources. Simply use the Azure portal or CLI to create a new resource group for this purpose before deploying the template.

11.2.3. Deploy

Once the ARM template parameters are updated, we are ready to deploy.

In the Azure portal click on "Create a Resource" then in the search text box enter "Template" and choose the selection: "template deployment (deploy using custom templates)" that appears. Next, click "create" at the bottom of the Template Deployment tile and then "Build your own template in the editor." Finally, click "Load file", then select the updated file from the local file system and click "Save".

You will be presented with a "Custom Deployment" screen where you will need to specify the resource group you created in the previous step. Now click "Review and Create" to build out the template. The entire VNet will be built for you in just a few minutes. It's really that easy!

A more detailed detailed guide to deploying custom templates can be found here:

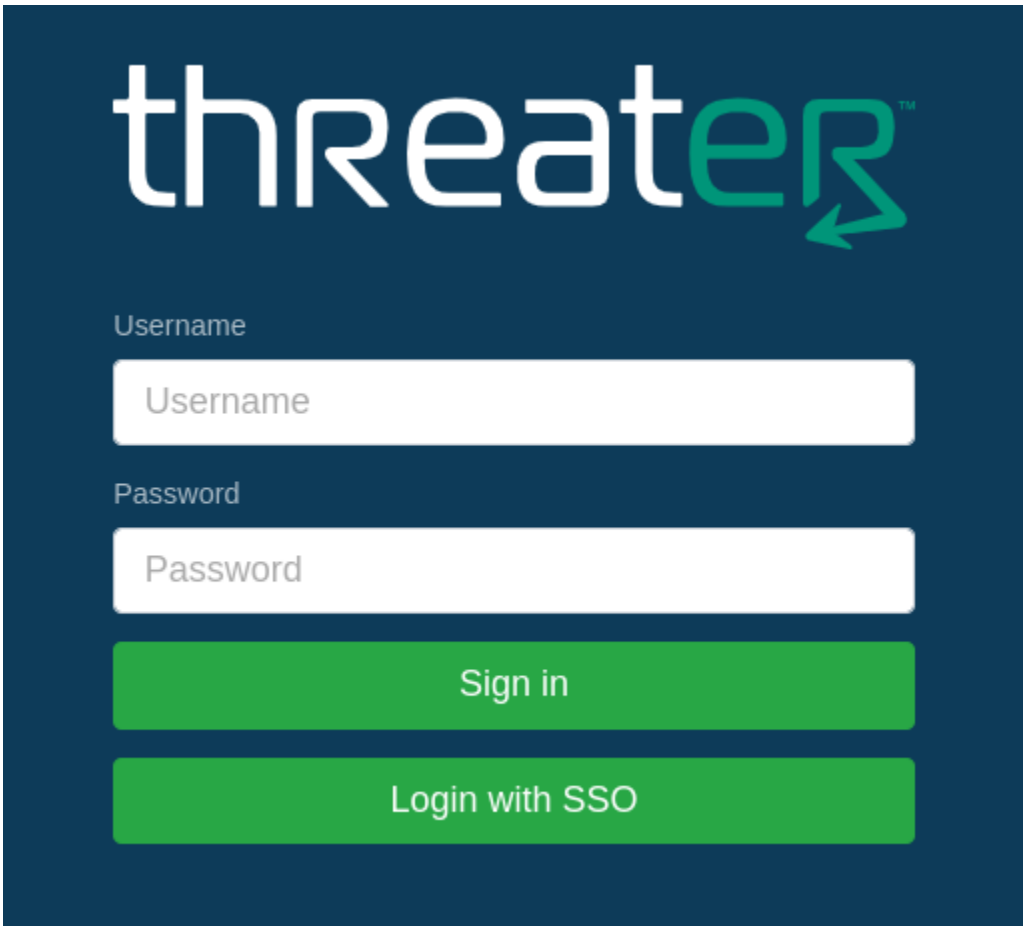
https://learn.microsoft.com/en-us/azure/azure-resource-manager/templates/deploy-portal#deploy-resources-from-custom-template

11.2.4. Verify Threater Enforce Login

After Azure finishes building out all of the resources, we can verify that the Enforcer instance is running by pointing a web browser to the public IP assigned to the Enforcer's admin interface.

Note that we have utilized security group configurations in the deployed template to safely lock-down initial administration access solely to the public-facing IP address of the system you're currently using. That means that generally you must access the Threater Enforce UI for initial configuration from that same system. If you are behind a NAT point at your current location, then be aware that any other system that NATs to the same public-facing IP will be able to connect as well (if they have the proper credentials to connect, of course).

The following login page should render in your browser after navigating the security warnings:



If the login page renders correctly, the Enforcer deployment was successful!

The Enforcer is now ready for configuration.

12. License and Configuration

To finish the deployment, you **MUST NOW** configure your instance before it will function properly.

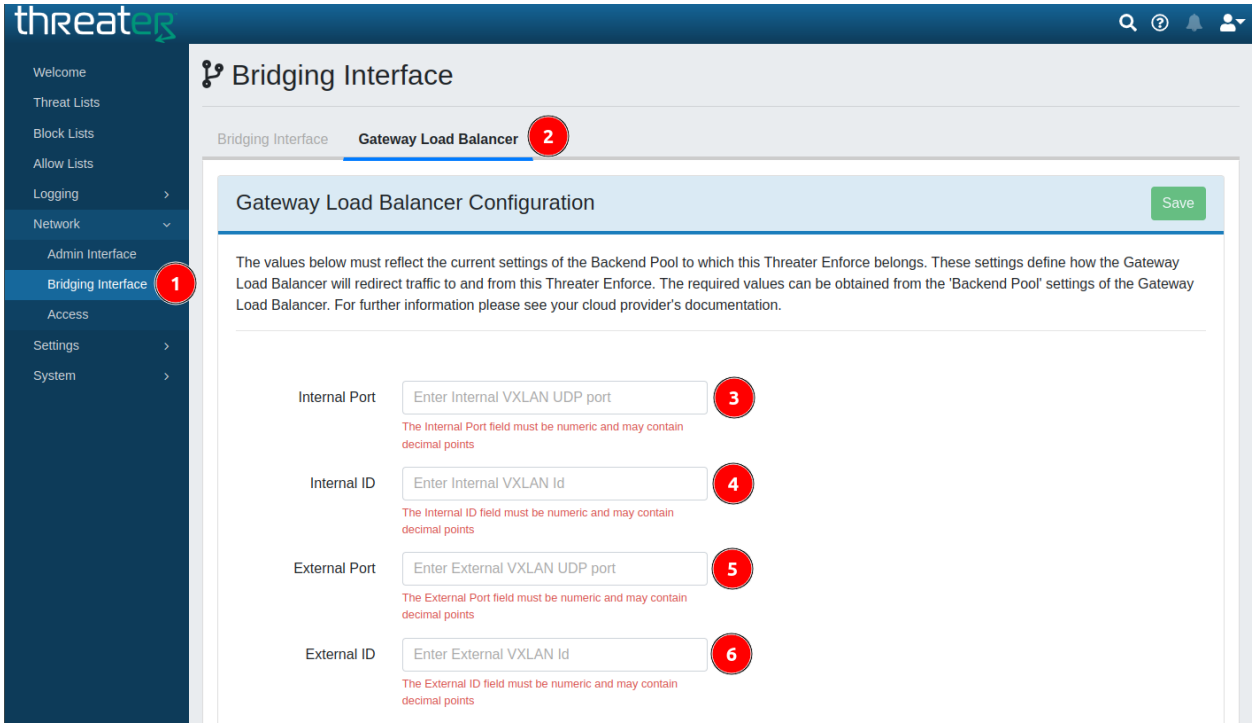
12.1. Initial Threater Enforce Configuration

A typical initial Threater Enforce configuration flow is as follows:

1.	Login with the default administration user: admin and default password: admin
2.	Accept the displayed terms of service / end user license agreements (a one-time operation).
3.	On the next screen (another one-time operation), provide your login credentials for your pre-existing Threater portal account. This is a key step that will register this Enforcer instance with your Threater account. Once authenticated, the screen will close and you'll be presented with the standard Threater Enforce "Welcome" page.
4.	Select System > Users to set up any required local users. At minimum, you should change the default admin password immediately. As with any production system and especially security controls, it is never wise to leave the default login credentials in place. Make sure you choose a strong password that you can remember or use a secure commercial password storage solution.
5.	Check your Network > Admin Interface and make any changes needed. In most scenarios no changes will be required here.
6.	Set DNS via Network > Admin Interface > DNS. By default the system will utilize Google's DNS for the primary and Cloudflare's for the secondary, but you can change these if you prefer others.
7.	Generally you'll stick with your Azure security group configurations for access, but you can decide on any extra admin access subnets if needed via Network > Access.
8.	Enter a unique hostname via Settings > General. Uniqueness is important so that if you later decide to leverage our powerful syslog export feature, individual installations will be able to be uniquely identified by their hostname.
9.	Set your desired time zone via Settings > Date & Time.

10.	If desired, set an NTP server of your choosing via Settings > Date & Time > NTP Servers. By default we configure Google's time services via time.google.com, but you can change this to whatever you'd like.

Azure deployments also require that the VXLAN settings on the Enforcer tunnel interface be configured. These must match exactly those of the Gateway Load Balancer backend pool to which the Enforcer belongs. These settings are found under Network > Bridging Interfaces > Gateway Load Balancer from the UI navigation. Azure defaults are: 10800, 900, 10801, and 901 so we have used those values in the example template. Simply enter these values in bubbles 3,4,5, and 6 respectively and click Save.



12.2. Register with Threater portal

Now that the basic networking configuration for your Enforcer instance is complete, open up a separate browser tab and connect to the Threater portal platform at:

https://portal.threater.com

After logging in, select Subscriptions in the left-hand navigation, and assign an unassigned subscription to the new Enforcer instance we just created above.

Within a few minutes of assigning your BYOL license, your new Enforcer instance will start communicating with Threater's central systems to synchronize policies and lists. If you're an existing Threater customer, then this process is seamless. If you're a new Threater customer, we recommend you consult our available documentation on configuring and using our portal.

In no time, you'll be up and running, with active protection in place for all network traffic flowing between your protected instances and the Internet, just like you're used to when running our solution entirely on-premise.

From that point forward, your lists and policies associated with your Enforcer instance can largely be managed from the portal, in exactly the same way you manage your on-premise installations.

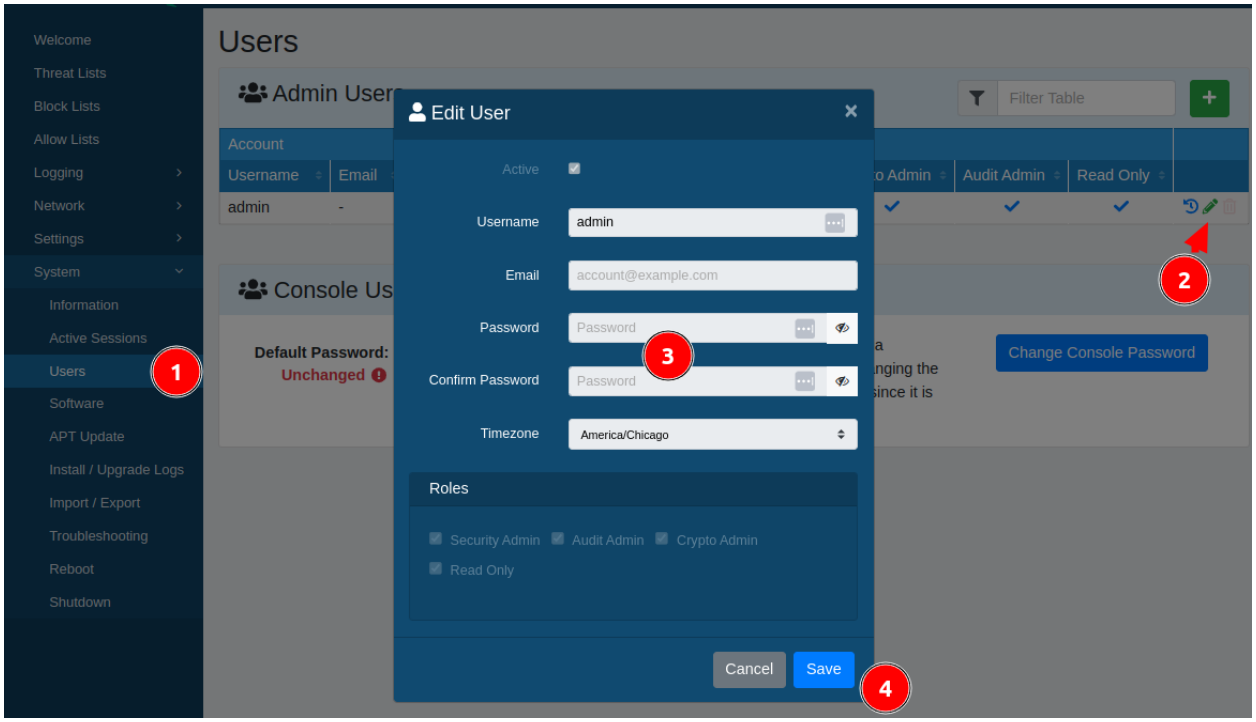
Note that if you do not fully assign a BYOL license and configure your system as described in the contents above, then your Enforcer instance will afford you no protection whatsoever, nor will it log any information. It will simply pass ALL traffic in both directions and log nothing. Only after applying your BYOL license and configuring the system properly will bidirectional traffic be fully protected alongside comprehensive logging.

12.3. UI Access: Password Rotation

Best practice (as described earlier in this document) ensures that administration access should be locked down to specific known IPs through the security group configurations described, but it is still a good idea to make sure you change your system passwords regularly.

And on that note, we strongly urge customers to rotate your local UI passwords.

Like most modern systems, changing your password is as simple as logging into the UI, navigating to the user configuration, and then modifying the password. The flow graphic is:



The steps are straightforward:

1. Select System > Users
2. Click the green pencil edit icon
3. Enter and confirm the desired password when prompted
4. Click Save

13. Test out the Deployment

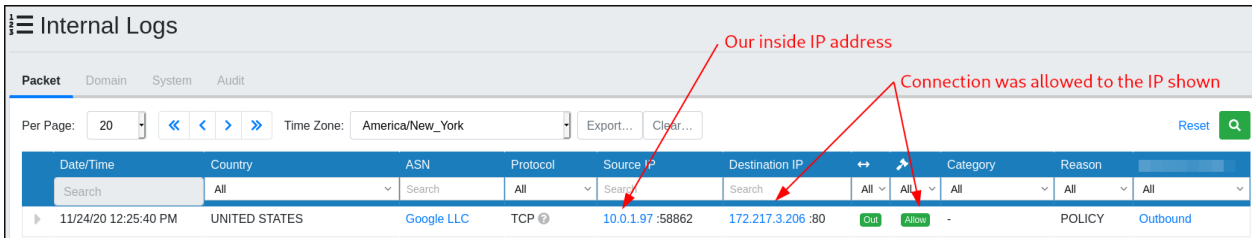
Now that everything is configured we can log into the Enforcer's UI and investigate the Internal Logs to see the connection activity in more detail. To generate a logged connection let's first login via ssh to the protected Standard_A1_v2 instance and issue an outbound HTTP request to google.com:

$ curl google.com

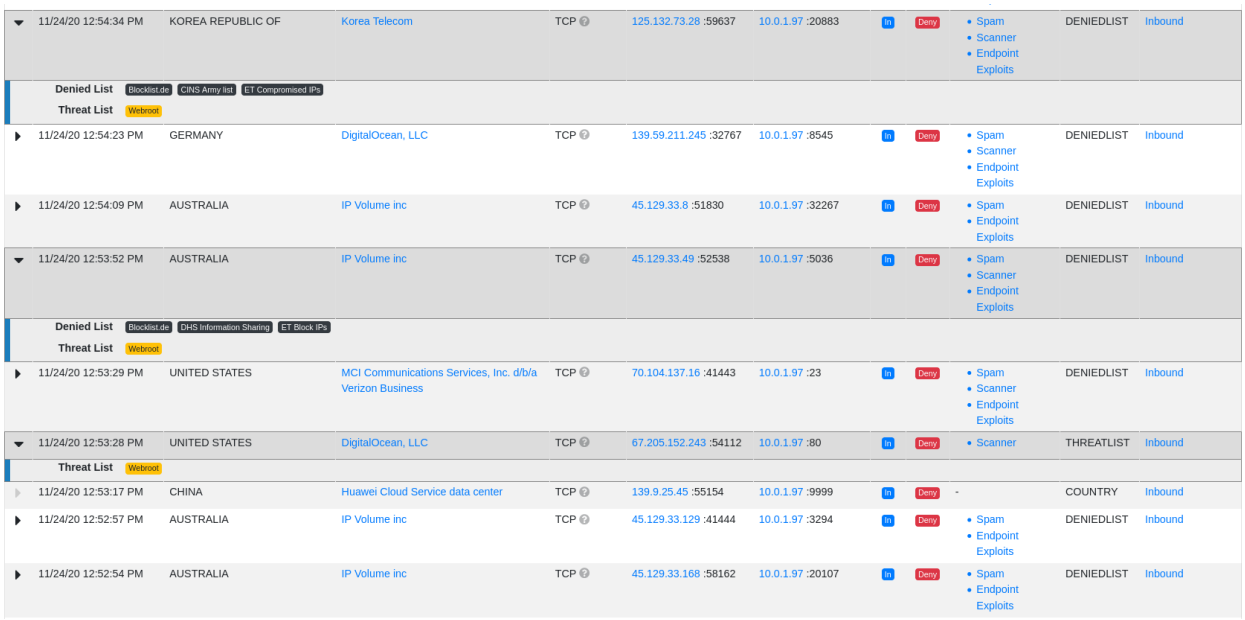
The request should return 301: The document has moved response (which is what we expect in this case).

In the Enforcer's UI we can now investigate the Internal Logs to see the connection activity in more detail. As we might expect, we can see that the destination IP was indeed registered to a Google ASN, and the IP maps to a known location in the US. We see that our protected IP address that attempted to reach it was 10.0.1.97 (you will likely see a different address when

you try it, as the addresses assigned to your protected instances will likely differ). We see that the TCP connection was allowed from source port 58862 (this port number will also likely differ in your case as it is generally randomly assigned by the test instance's operating system) to destination port 80 on the Google server, as it passed all relevant outbound policy criteria.



An obvious question is what might it have looked like if access to a known malicious site had attempted activity to or from that protected IP address? The short answer is that if it's known-malicious and on any threat or denied list attached to your policies of record, then it's going to be blocked. Here's a live example of an extraordinary amount of blocked, known-malicious traffic that we witnessed after manually opening up the outside subnet to all IPs and port access:



As you can see, the Threater Enforce software seamlessly blocks a ton of very bad stuff, leveraging our best-in-class third party threat list and denied list integrations. We've got several that come out-of-the-box, and a plethora of integrations for other proprietary threat intelligence feeds. For a full list of all of our supported integrations please visit our website.

You can see in the partial screenshot above that the malicious behavior we encountered was detected stemming from multiple countries, with multiple attributed threat and deny lists - attributable to a variety of proprietary and open source third party feed information. One of the great benefits of our patented platform is regardless of how much threat intelligence is used or how many blocks are being enforced, there will be no decrease in your network performance. That is, whether you are protecting against one or tens of millions of threats, both our on-premise and cloud-based Threater Enforce software will continue to operate at line rates with no additional latency.

Granted, this extraordinary amount of nefarious activity shown above can be at least somewhat mitigated through the use of proper Azure security group policies. But you can never lock things completely down via Azure security groups only. And as you can see by the nefarious activity above and specifically the targeted port numbers, very little of it is attributable to "web sources" via ports 80 and 443 - that is, web application firewalls and the like are not sufficient protection by themselves.

For example, if you're a typical business, you will have some number of public facing access points and/or open ports, and that's where you have risk. That's where you are vulnerable, especially when services (such as VPN services) must be kept open to large swaths of the world for large multi-national organizations or companies who do business with them. Those holes are what the bad guys will attempt to exploit. Just like they're trying to do here against our simple little test instance. And that's where Threater shines.

14. Data Encryption and Secure Communications

As was likely evident when you went through the deployment steps, there are no specific data encryption considerations for the deployment. Everything is automated within the Threater provided VHD image and centrally managed by HTTPS connections which ensures on-the-fly standards-based public/private key sessions, as negotiated by an end user's browser.

Just like on-premise deployments, our cloud deployments encrypt all data transfer between the Threater Enforce software and the Threater portal, utilizing HTTPS transactions via port 443. As such, standard techniques leveraging internally managed public/private keypairs are used, with standards-based negotiation for any and all access.

An example would be when the Enforcer instance communicates to the portal to determine if there is any new real-time threat intelligence information ready to be retrieved.

Note that the architecture is a 100% pull architecture (vs. a push architecture) with respect to the Threater Enforce software. That is, our Threater portal has no way to directly reach Enforcer deployments on its own. Instead, just like our on-premise deployments, the Threater Enforce software (even when running in Azure) always initiates secure communications to the Threater

portal at which point the portal can provide new information, and never the other way around. Specifically, these communications are:

- Feed data, statistics and related metadata, and health checks are sent over TLS.
- All threat intelligence and related feed data is sent over TLS, encrypted and signed.
- All software updates scheduled by the end customer from the Threater portal and subsequently delivered and transferred over TLS.
- All TLS connections to the Threater portal are verified by certificates.
- And last but not least, each and every such communication uses TLS by way of HTTPS on port 443, always, without exception. Although often not applicable in the cloud, this fact is often quite useful for our on-premise customers when they choose to deploy us on the near-side of an existing on-premise next-generation firewall, as generally port 443 is already open, so no further external network configuration is typically required.

15. Maximizing Uptime

Our on-premise solutions are sometimes deployed by our customers as HA pairs. In addition, most of our on-premise hardware includes NICs that can be manually or automatically placed into a physical bypass mode, where even in the event of power failure or some other catastrophic hardware failure, traffic will pass through the system housing of the Enforcer unabated. Unfortunately, in Azure, neither of these can happen as there is no direct hardware bypass capability or access in Azure network infrastructure.

To solve this problem Azure has designed its Gateway Load Balancers with HA capabilities in mind as they can: distribute incoming traffic across your Azure instances in a single Availability Zone or multiple Availability Zones. Thus, by intelligently distributing multiple Enforcer resources across availability zones it is possible to design a service with high availability that is perhaps superior to that of on-prem hardware bypass.

Thus, it is indeed possible to protect against hardware faults in the cloud. However, creating HA services in Azure is beyond the scope of this document so please reference Azure documentation for further details on this topic.

16. Monitoring Health

Monitoring health is simple. There are two recommended ways to check health:

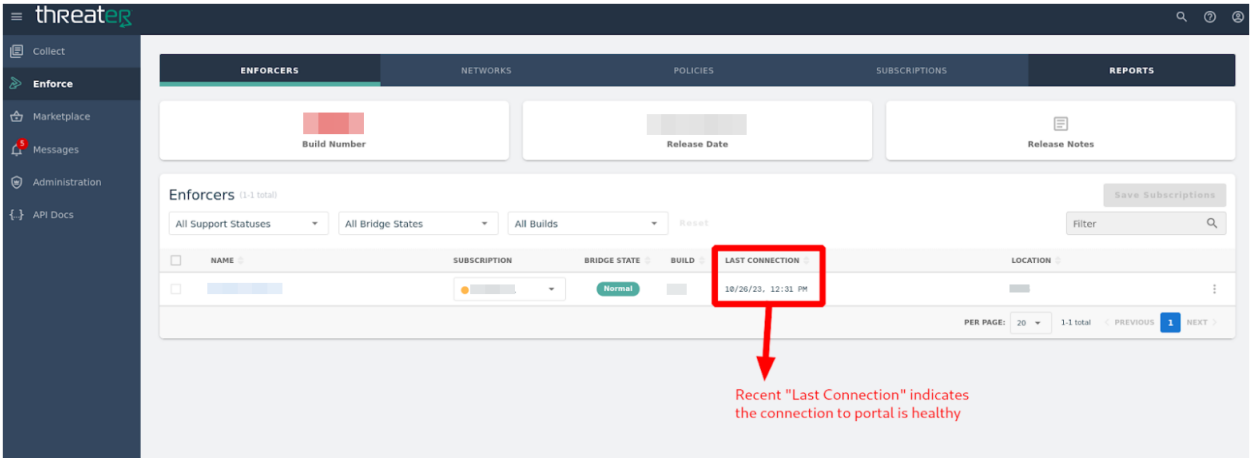
- via the Threater portal
- via Enforcer’s exportable RFC-compliant system logs

Both are valuable, and the use of one does not preclude the use of the other. In fact, it is our strong recommendation that customers leverage both paradigms, which we discuss in more detail below.

16.1. Checking Health via the Portal

This is the recommended way to verify health, as you can log in exclusively to the Threater portal to view information about your Threater Enforcer instances - whether on-premise, in the cloud, or both. This way you don't have to concern yourself with individual asset logins.

Threater Enforce software automatically and routinely connects to the Threater portal. This connection information can be found in your portal views:



16.2. Checking Health via the Syslog Export Capability

Although leveraging the portal's health monitoring tools are useful, we also highly recommend that you connect the Threater Enforce software's powerful RFC-compliant syslog export capability to one or more syslog sinks of your choosing. Not only does this allow you to monitor health, but it also allows you to monitor all detailed low level activity for all connections allowed and denied, which can be very important when analyzing network and security behavior. Additionally, it provides a means for you to backup critical historical security/log information routinely, in real-time.

Targeted systems for the logs exports could be a SIEM such as Splunk or IBM QRadar, or analysis tools like Gravwell, or even open-source syslog-ng. Configuring one or more of these types of exports will allow you to see real-time low level detail of the system activity, to include monitoring of the underlying software components. Detailed documentation on our RFC-compliant syslog export capability is available here:

https://threater-marketplace.s3.amazonaws.com/Threater+-+Syslog+Export.pdf

The nice thing about using a third party tool such as a SIEM is that you can leverage that tool to alert you and potentially take action on any behavior that you'd like. For example, many of our customers prefer to leverage SIEM and SIEM-like tools to initiate HA failover scenarios, as part of standard security stack best-practices.

17. Backup And Recovery

Threater Enforce deployments in Azure, just like our on-premise deployments, have a comprehensive built-in ability to backup the entirety of the configuration. The backup can be downloaded as a JSON file. It can then be re-imported later for recovery purposes.

Once logged into the Enforcer's UI as previously described in this document, you can use the following flow to backup and restore the configuration:

The referenced numerical steps become:

1.	Select System > Import / Export.
2.	Select Export.
3.	Select either the Copy button to copy the configuration contents to the clipboard, or the Export button to export it to a persistent JSON file.

4.	When you are ready to restore a backup configuration to a particular system, choose the Import tab and point it at your previously exported configuration file. That's it!

Note that when importing a backup configuration for restore, it is instantaneously applied. There is no wait time or need to reboot.

This technique is also useful if you find that you had a need to fully terminate the current instance (or if it suffered a hardware failure - failures occur sometimes in the cloud too!), as opposed to stopping it for an eventual restart. When you relaunch a previously terminated instance from Azure Marketplace, it will come up as a brand new instance with no configuration. You could of course reconfigure everything manually as described earlier in this document, but if you had a backup JSON configuration, it is trivial to reload it. After importing the configuration, the Threater Enforce software will then be back in the exact same configuration state as when the backup was made, and will immediately start functioning just as you left it.

Note that one of the nice benefits of our architecture is that the only thing to be concerned about for backup and recovery with our architecture is this configuration itself! Furthermore, since it is a simple JSON file, it is trivial to maintain in any system of your choosing, following whatever best practices that your corporation employs. There is very little post-deployment Threater Enforce configuration since most of the functionality is managed exclusively by the Threater portal which means that it isn't necessary to worry about routine, automated configuration backups.

18. Software Patches and Upgrades

Our on-premise and cloud-based software uses the exact same codebase. As such, whenever we release software, it is always, without fail, released for both.

One very nice benefit of our architecture is that the software patch and upgrade process is entirely managed by the Threater portal.

This means that once a subscription is attached and Threater Enforce is securely communicating to our portal, you are able to schedule software patches and upgrades directly from the portal itself. You can even elect to do an immediate unscheduled software upgrade. In each case, the portal will instruct the Threater Enforce software to download and install updates at a configurable time.

This is particularly beneficial for customers with both on-premise and cloud deployments - they can upgrade any or all of them centrally with no procedural differences whatsoever, all from the Threater portal. In such cases, Threater Enforce software does not need to be directly accessed by the end user at all during the upgrade process. It's all handled for you, automatically. You just

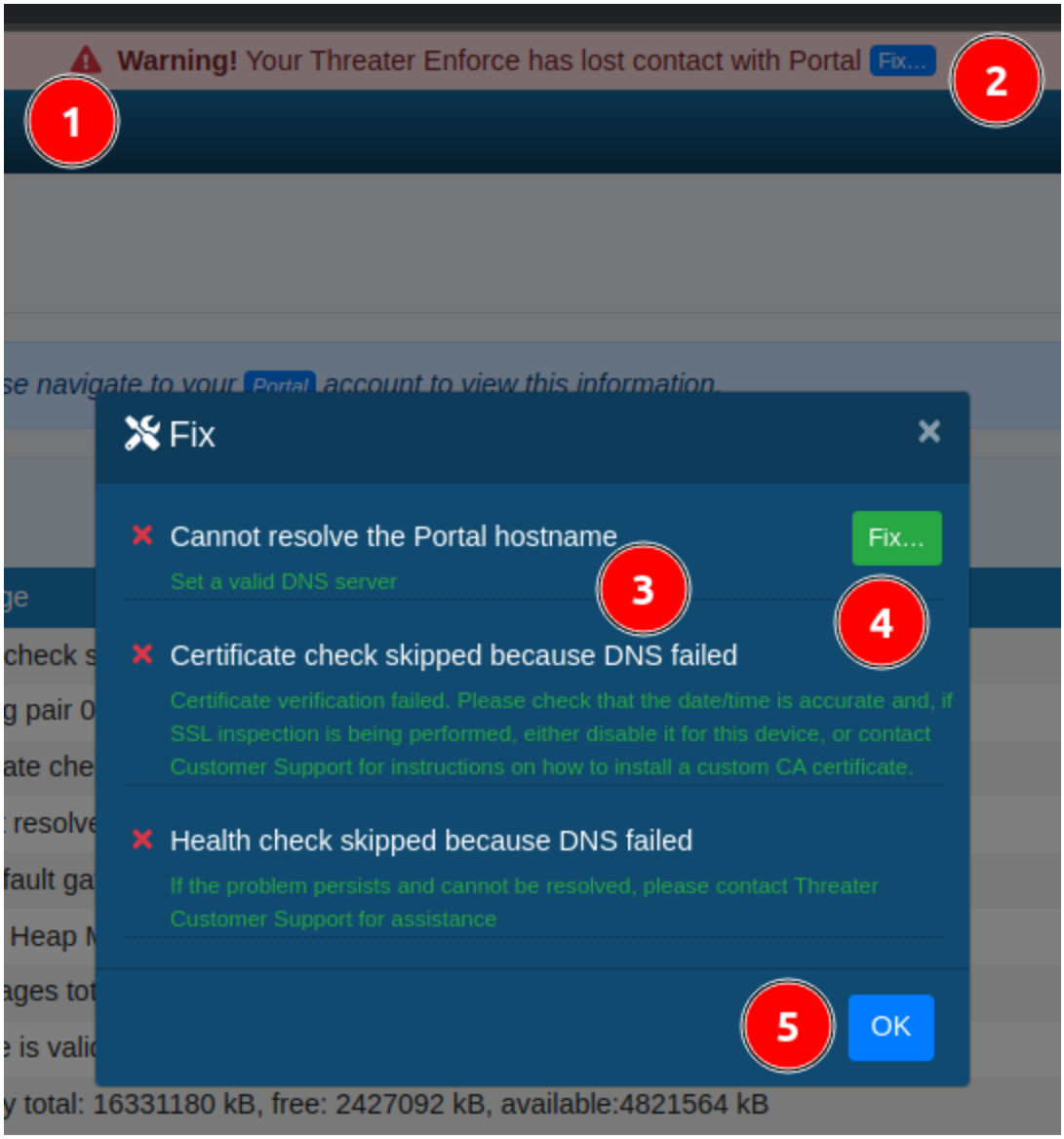
decide when you want the upgrade to occur. We generally recommend doing it in a short maintenance window. In general, it takes no more than about 5 to 10 minutes start-to-finish for a software upgrade to complete.

Additionally, when we release a new software version, we also update the Azure public VHD offering to match. That ensures that any new Threater customers deploying on Azure for the first time will always have the "latest-and-greatest" software available to them, just like they would get if they ordered a new on-premise system.

19. Handling Faults

We've gone the extra mile to make critical fault identification and recovery trivial, with straightforward instructions and "Fix It" style guides directly in the UI. We accomplish this with an in-your-face banner that will dock to the top of the UI whenever such an anomaly arises, enabling rapid remediation. This applies for absolutely everything that can go wrong relating to the deployment.

Here's an example where a user misconfigured their DNS causing a failure to connect to our portal. That's bad, since it means that the Threater Enforce software wouldn't be able to retrieve updated threat intelligence. As you can see below, the system intelligently detects this and other catastrophic faults, and actually tells you about them so you don't have to guess. And then it helps you fix it, with a dialog similar to what is shown below:



The flow here becomes:

1.	The warning bar clearly tells you something is wrong.
2.	The blue "Fix" button takes you to the fix modal.
3.	The modal specifically tells you what it figured out - that it can't connect to the Threater portal, and it's likely because of a DNS configuration problem.
4.	You click the green Fix button and it will take you to the appropriate configuration screen for fixing.

5.	And finally, you'll click OK.

Assuming you've fixed it properly with the guided fault remediation flows, the warning will disappear within seconds.

That same flow is the handling and remediation flow for all fault conditions with remediation pathways. The goal of these remediations is to quickly get a user back up and running whenever anything critical is detected.

Additionally, as mentioned in multiple places elsewhere in this deployment document, it is highly beneficial to export our RFC-compliant syslog data to one or more target systems, such as a SIEM, so that you have the ability to do detailed low-level fault analysis at your discretion, or perform historical analysis as needed. This is also highly desirable since SIEMs and related tools can be easily configured to alert on any number of criteria by way of things like email, SMS, and so on.

20. Next Steps and Cleaning Up

If you used this guide to construct a live, production deployment that you wish to use moving forward to deploy protected instances, then you can leave everything configured as-is. If not, any resources that we have just created can be completely deleted by deleting the associated Resource Group from the Azure portal. There is no need to remove each resource individually as Azure takes care of that for us.

21. Summary

We have now finished a complete example configuration within the confines of Azure entirely from scratch. We have:

- Presented a diagram that demonstrates how Threater Enforce is deployed in Azure Gateway Load Balancer mode.
- Deployed a functional resource group to demonstrate the capabilities of Threater Enforce
- Applied a Threater Enforce BYOL subscription via the Threater portal
- Evaluated Threater Enforce allowing traffic considered "trustworthy"
- Investigated log examples of Threater Enforce allowing trustworthy content while blocking malicious content

Our existing customers already know how simple, smart, and scalable our patented technology is for their on-premise Threater Enforce deployments. And now, with Threater Enforce protecting

native Azure infrastructure, we are pleased to offer the same simple, smart, and scalable capabilities providing a robust layered security architecture. Everywhere.

Learn more about us by visiting our website at:

https://threater.com