VMWARE NSX & OTRS

Automating Security with Help Desk Systems

Sander Martijn (sander@vmguru.com)
Anne Jan Elsinga (aelsinga@vmware.com)
Martijn Smit (msmit@vmware.com)

vmware®

VM GURU

Table of Contents

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com

Executive Summary

In an ever-changing world we rely on data, lots of it. The data itself has transformed from a by-product to the product itself. Keeping that data secure and out of the hands of anyone that isn't allowed is getting harder, since there are more systems to manage. To keep the data secure and safe companies have to conform to various laws and standards, for example "Privacy Shield" and "General Data Protection Regulation (GDPR)". With the rise of more laws and market standards it can be hard to be compliant.

This whitepaper describes how VMware NSX can be used with service management tooling to create a dynamic security infrastructure in which we create "Just in Time" / "Just Enough". This means for example that administrators get remote access to systems only when they actually need it. No unnecessary connections are possible, keeping the systems more secure.

Problem

In this day and age data protection is a top priority for every company. Specifically, privacy sensitive data, also called personally identifiable information (PII) is a much-discussed topic.

This has even lead the way to certain laws like "Privacy Shield (US)" and "General Data Protection Regulation (GDPR EU)". These laws dictate measurements on how data should be handled and force companies to chance the way they look at security.

One such measurement is that access to data should only be possible when you actually need it.

Now in most IT infrastructures you will find that administrators are able to access servers through RDP or SSH at any time, as they need to maintain the systems. Outside these maintenance windows they have no need to have access, but the connections are still possible.

When these maintenance ports (RDP, SSH) are constantly reachable in the network, they are open for probing and attack. Although these maintenance ports are usually only open for the internal network, these days the internal network cannot be trusted anymore due to changing dynamics (more users in the network due to VDI & IoT). This is where VMware NSX can be used to block all unnecessary connections by default (micro-segment) and only open ports when they are needed.

In this construct, granting and revoking access to servers is a tedious and error prone task. Plus, not many companies will revoke the access when the maintenance is done.

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com

VMware NSX

VMware NSX is the market leading implementation of network virtualization from VMware. By delivering a completely new operational model for networking that breaks through current physical network barriers, NSX enables data center operators to achieve orders of magnitude better agility, economics, and choice.

With NSX, virtual networks are programmatically created, provisioned and managed, utilizing the underlying physical network as a simple packet forwarding backplane. Network and security services in software are distributed to hypervisors and “attached” to individual VMs in accordance with networking and security policies defined for each connected application. When a VM is moved to another host, its networking and security services move with it. And when new VMs are created to scale an application, the necessary policies are dynamically applied to those VMs as well.

NSX is a completely non-disruptive solution:

- Deploys on hypervisors connected to any existing physical network infrastructure and supports next generation fabrics and topologies from any vendor;
- Requires no changes to existing applications and workloads
- Allows IT departments to incrementally implement virtual networks at whatever pace they choose (without any impact to existing applications and network configurations)
- Extends visibility to existing networking monitoring and management tools to deliver increased visibility into virtualized networks

The net result is a transformative approach to data center networking that – among its many other benefits – matches the velocity and agility demands of today’s businesses by reducing service delivery times from weeks to seconds.

Being able to automate network operations is a big benefit of NSX and it is the use-case on which the rest of this whitepaper is based on; automating security using third-party platforms such as OTRS.

Learn more here: http://vmware.com/go/nsx

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com

OTRS

OTRS is one of the most flexible web-based ticketing systems used for Customer Service, Help Desk, IT Service Management. With a fast implementation and easy customization to your needs it helps you reducing costs and increasing the efficiency and transparency of your business communication.

OTRS is one of the most successful and long-lasting open source projects in the area of help desk and IT service management worldwide. More than 5,000 active community members improve the service management software with every release by reporting bugs, adding self-developed improvements or new functionalities, and maintaining and extending the 38 supported languages.

Because it is open source, OTRS is easy to extend and expand with plugins and an easy to use API.

By using OTRS it is possible to create a work process which only allows access to servers when needed. When an administrator receives a ticket regarding a specific server, only then does the administrator need to be able to reach a certain server to do their maintenance and/or changes. Once the maintenance has been completed, the access should be disabled again.

Learn more here: https://www.otrs.com/

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com

Solution

By using both the security and automation options provided by NSX it is possible to create a dynamic security framework. Allowing only access to servers when it is actually needed and keeping access closed while it is not.

Using NSX makes it possible to dynamically allow or deny access to a certain server. This can be achieved by creating a security policy that will allow access from one security group to another.

For example, a security group containing the administrator virtual desktops or management server, could be used in a security policy that would allow RDP to another security group.

By using the NSX API it is possible to set a security tag on a server. Receiving this tag will automatically add the server to the security group. And thus, allow RDP access from the administrator security group to that server.

Denying access to the server would only mean removing the security tag through the use of the NSX API.

This example solution also makes use of OTRS. This ticket system has to ability to lock a ticket when an administrator starts working on it. It is also possible to execute a custom action on an event like locking a ticket.

Also, OTRS allows us to modify the form that is used to create a ticket. By adding a mandatory field containing the server name it is possible to extract this name at a later stage in the process.

Tying this solution together is a Python based web service acting as middleware. This middleware service receives incoming web hooks from OTRS and reaches out to VMware vCenter and NSX to configure the security tag.

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com

OTRS Integration

With OTRS it is possible to generate an event when certain actions are being taken. An example of one of these actions is the locking or unlocking of a ticket. This lock/unlock action is performed on a ticket when an administrator assigns the ticket to themselves to start working on it.

OTRS generates an event with two parameters; the ticket ID and ticket number.

In this case we use this generated event and use it to send an API call to a middleware service. In this API call we include the ticket ID so the middleware service can use that to get the ticket details.

This middleware will then query the OTRS API for the ticket details. The result of the OTRS API call will contain all information about the ticket; is it unlocked or locked, which server name (VM) is included in the ticket, etc.

Middleware

For this solution to work it needs a middleware service to glue it all together. In this case Python is used to create a web service that will handle all the API requests, which we've named OTNSX

As mentioned in the previous chapter when a ticket is locked or unlocked in OTRS a HTTP request is made to the middleware service. This request includes the ticket ID and will be used to retrieve additional information from the ticket.

By using the OTRS API the middleware service will do a lookup for the ticket information with the ticket ID that was passed from OTRS. Among the data that is returned there are two variables that are interesting.

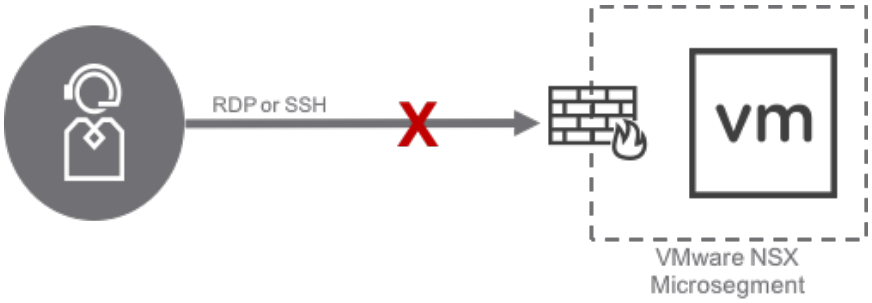
These variables are the virtual machine name in the mandatory field we've created earlier and the status of the lock on the ticket (locked or unlocked). With the virtual machine name, the next API call can be made to vCenter and get the virtual machine ID.

With the virtual machine ID retrieved, an API call can now be made to the NSX API to set or unset the security tag. Depending on the ticket now being locked or unlocked the API call will either add or remove the virtual machine to a NSX security group. When the virtual machine is added then a connection can be instantiated by the administrator. On removal from the security group connections will not be allowed anymore.

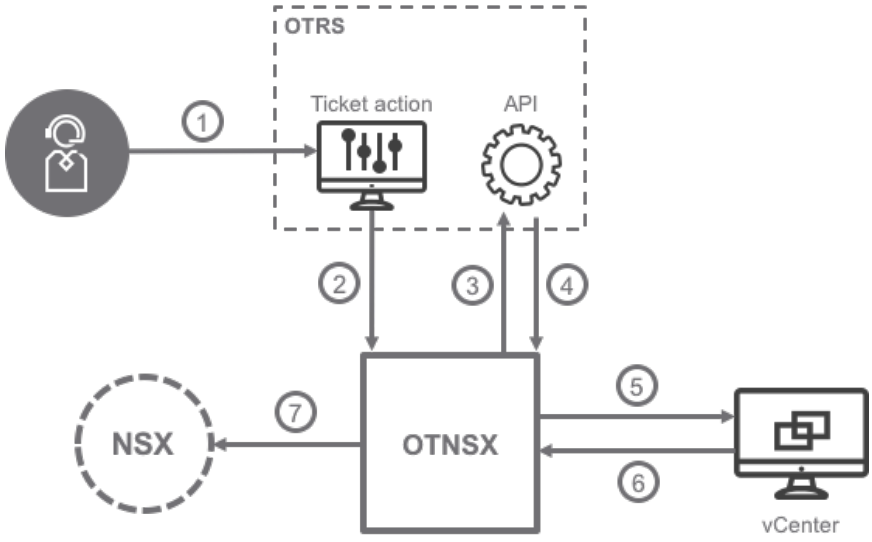
VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com

Workflow

Below is an example of an administrator trying to connect over RDP or SSH access to a virtual machine. Since the administrator isn't working on a ticket logged in OTRS, he won't be able to connect to the virtual machine.



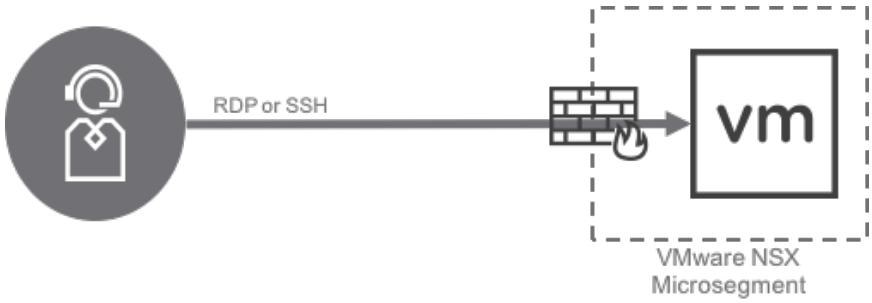
The administrator will first have to go to OTRS and lock the ticket that is linked to the virtual machine needing maintenance. This lock action will start the process shown in the image below.



1. Administrator locks the ticket inside OTRS.
2. Through the custom action on the 'Lock' event in the ticket, a bash script is started that calls the middleware service via an URL.
3. The middleware service queries OTRS through the API with the ticket number to retrieve the ticket details.

4. The OTRS API response contains the lock status and virtual machine name.
5. The middleware service queries vCenter for the ID of the virtual machine (MoRef ID).
6. vCenter responds with the MoRef ID.
7. The Middleware service connects with the NSX manager containing the MoRef ID, requesting to provide the VM with a security tag. This security tag is used by a security group assuring the VM will become member of the security group.

Only after the administrator has locked the ticket he will be allowed access to the virtual machine.



When the administrator is done with the maintenance and unlocks the ticket, the same process is executed. Only now the security tag will be removed from the virtual machine, which will automatically remove it from the security group and close the maintenance ports.

Installation

With the theoretical workflow explained in the previous chapters, the upcoming chapters will go into how to install the middleware agent (OTNSX) and the configuration needed inside OTRS.

Configuration – OTRS

OTRS has a job engine, which can execute tasks based on certain events. In our workflow as described in the previous chapter, we are using a “Lock” event on a ticket to capture when a system administrator clicks the “Lock” button. This fires off a bash script when this event occurs. This bash script will then activate the middleware.

First put the bash script in place on the server that is hosting OTRS. Put it in directory **/opt/otnsx/bin/**. To get you started, here’s the procedure:

```
# mkdir -p /opt/otnsx/bin
# curl -o /opt/otnsx/bin/otnsx-activate-middleware.sh
https://raw.githubusercontent.com/vmguru/OTNSX/master/otnsx-activate-middleware.sh
```

Then edit the script and point the variable *OTNSX_URL* to the middleware web interface, which we will install in the next chapter.

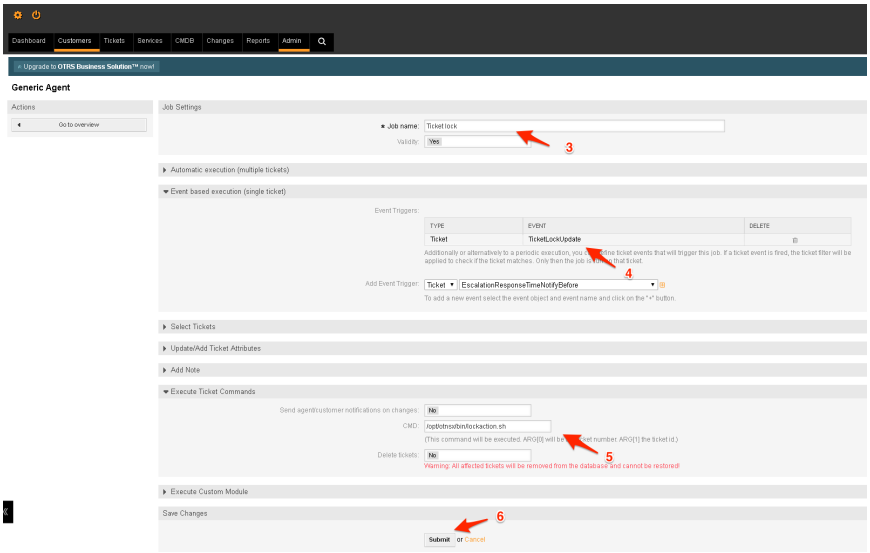
```
# Change this:
OTNSX_URL=”http://192.168.178.19:5000”
```

After editing the bash script, we need to configure a custom action so that OTRS will call this bash script and activate the middleware service.

To configure a custom action to trigger the script:

1. Log in to OTRS as an administrator user.
2. Browse to: Admin -> Generic Agent -> Add job.
3. Enter the Job name.
4. Under **Event based execution (single ticket)** add a **Ticket** event trigger for **TicketLockUpdate.**
5. Under **Execute Ticket Commands** enter the path to the bash script mentioned earlier (/opt/otnsx/bin/otnsx-activate-middleware.sh).
6. Click on **Submit** to save the job.

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com



Screenshot of an OTRS custom action.

Installation – OTNSX

The middleware service that we call OTNSX is a Python based web service and can run on any Linux or Windows server. To run OTNSX the following software needs to be present on the server hosting the service:

- Python v2.7 or v3
- Git

Once these requirements are met run a git clone command to download the OTNSX files:

```
# git clone https://github.com/vmguru/OTNSX.git
```

Using the requirements file you can retrieve the required modules for Python:

```
# cd OTNSX
# pip install –r requirements.txt
```

One of the files that was cloned from GitHub is **config.py**. This file contains the environment variables that are specific to your infrastructure, such as the credentials for vCenter, NSX & OTRS. Edit this file to match the parameters with settings needed for your environment.

After that the python script can be started and the web service will initiate:

```
# python otnsx.py
 * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
```

Conclusion

With the automated security possibilities inside VMware NSX, it is possible to limit maintenance access to servers by default – therefore protecting sensitive access – and only open access when needed. You can use a help desk system to log and control opening network access to servers before performing maintenance on it.

This document has displayed a way to integrate OTRS in this workflow, but this can work with any help desk system which has to ability to execute custom actions.

We've also introduced OTNSX, the middleware service that can receive actions from OTRS, get ticket details from the OTRS API and use that information to determine on which virtual machine to set a NSX security tag in order to open up network access to maintain the virtual machine.

More information about OTNSX can be found on GitHub:
https://github.com/vmguru/OTNSX

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com