Troubleshooting Mobile Email Delivery and AirWatch Issues

Introduction

VMware AirWatch is definitely the leading solution in the Mobile Device Management sector. This essential piece of infrastructure that manages all the connected mobile devices also plays an important role in email delivery to end-users.

When a user is reading his emails on his phone, he doesn't really care which infrastructure layer is involved in providing the service. Cloud, hybrid, on-premises, it just has to work!

AirWatch is a critical piece of the infrastructure that concurs to the messaging and collaboration service delivery. And that is why mobility teams need to have a view of their servers, as well as the overall service.

The purpose of this RoboTech is to explain how GSX Solutions can help mobility admins to manage the service quality that they really deliver to their end users.

At GSX, our mission statement is to provide out-of-the-box and agentless monitoring solutions that enable the IT teams to be proactive on incident resolution and focus on the end-user experience.

Let's see first in details what are the capabilities of AirWatch monitoring.

AirWatch servers deep dive

GSX for AirWatch connects at each server remotely and looks at every critical role to test and retrieve critical statistics.

AirWatch specific services monitoring

Each AirWatch role delivers specifics services that are tested from:

A Windows services perspective

- Device tunnel, interrogator, log manager queue services
- Tunnel servers
- Log Manager, Master and MEG queue services
- Messaging services
- EAS Integration
- Mobile Access Gateway
- Alert, bulk import services
- Device scheduler
- Etc.

Each service is monitored in real-time and is provided with some alerts in case of failure. The second important aspect of each AirWatch role is to be able to access to the different consoles and critical UL end-points.

AirWatch Web consoles monitoring

For that, GSX tests the connection to each of them, all the time:

- Web console availability
- Admin console
- Open enrollments
- My device
- Device management
- Self service
- MDM enrollments
- MSMQ queue monitoring
- SEG Console and console management
- Microsoft ActiveSync
- Content / Default (MAG server)
- Etc.

The access to these end-points and consoles are highly critical, as every action an AirWatch admin can do has to be done through them.

But to make sure nothing bad will happen with the web consoles, it is extremely important to constantly monitor the health of the IIS servers they are using. And that's another key point of GSX.

AirWatch IIS monitoring

Each IIS servers have their Application pools monitored:

- ASP.net Application restarts
- ASP.net Application running
- ASP.net Worker process restarts

Each critical IIS statistic is collected to provide a view on the health and usage of these critical parts:

- Connection attempts per second
- Get Request per second
- Post Request per second
- Application restarts
- Worker process restarts

So between the health of the IIS and the real availability of the end-point, the entire environment of the Web Console is constantly monitored.

AirWatch system monitoring

And finally, on top of all these particular tests and counters, every server and every role have at first a basic check of their system counters:

- CPU

- RAM
- Disk IO
- Disk Space
- And Network IO

These system counters can give precious information on the health of the infrastructure when they are combined with the specific tests and statistics for AirWatch service delivery.

To finish with the in-depth server monitoring, it is really important to consider the AirWatch database as a specific object.

AirWatch database monitoring

Based on SQL, the database needs specifics monitoring to check its health and retrieve every usage and health critical statistics.

For that GSX connects remotely with SQL queries to the database and first collects specific AirWatch statistics:

- Enrolled users
- Certificate details
- AirWatch version
- Compromised devices

As well as critical real-time information on its health:

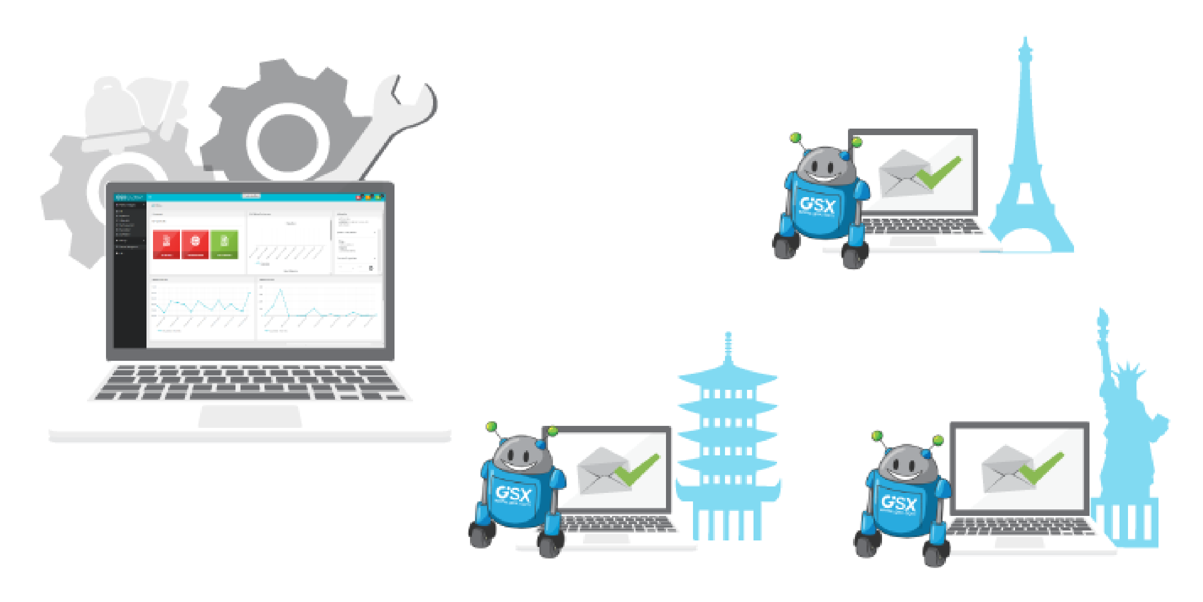
- Transaction log file size, Index Rebuild and Data file size drive
- SQL server Job History, Page Life expectancy and disk queuing
- Buffer Hit Ratio and Average Wait time
- Lazy writes per second and Number of Deadlocks
- Active Transaction and Users Connections
- Page reads per second and Page writes per second

It is very important to manage the health of the SQL database because any performance issue can impact any user trying to synchronize its mobile.

We’ve just seen that an in-depth monitoring of every services and critical statistics of each role is critical. But it is not enough to ensure your end-user satisfaction.

That is why GSX for AirWatch also directly monitors the end-user experience from any location you want to test through the GSX Robot Users.

Capture the true end-user experience



For those who are new to GSX and don’t already know what a GSX Robot User is, a quick check of our blog describing how Robot Users works could be helpful.

To summarize, a Robot User is a piece of software that you install on any workstation or window VM and that will use the service you want to test exactly as a user would do.

Doing that, it measures the availability and the performance of this service from an end-user perspective.

The Robot Users are installed in multiple remote locations and emulate a mobile device that:

- Opens mailbox and sends email
- Proceeds to free/busy lookups
- Creates meetings, folders, email and tasks
- Resolve a user and perform a search on its mailbox
- Download an attachment from an email

The Robot Users use your entire mobile infrastructure, measure the time it takes for each action, and alert you in case of any failure or performance problem.

As a result, you can know about any potential issues before real users start to complain.

On top of that, they perform network checks and retrieve critical information that will help you to understand the impact of the network on the end-user experience:

- One-way latency and round-trip time
- DNS resolution time

- TraceRoute with number of hops and latency
- Packet loss

The network statistics are critical to understanding if the network has an impact on the end-user experience.

They also allow the mobile team to understand the impacts of any network change on the mobile end-user experience.

Thanks to the Robot Users, GSX provides you with information about the end-user experience delivered through AirWatch and Collaboration infrastructure.

To understand a bit better how all of that is helping your mobility team and the overall IT management to manage the service quality delivered to end-users, here are some uses cases we faced with our customers.

Solving mobile connectivity issues with email

The situations we will see here really happened to one of our biggest customers. That is why part of the solution was designed with them to ease the detection and the troubleshooting of these problems.

This customer is a large oil & gas company that has worked with GSX for many years and that expressed interest in our AirWatch project. Since they have multiple locations in many countries, it was really important for them to understand in real-time, if their employees can really access the company's IT resources.

This company is using Office 365 and AirWatch servers to manage and secure their 15,000 devices. Regularly, remote users were opening tickets related to email synchronization or availability through their mobile devices.

Web Listener and EAS integration service issue

Situation

In this first example, after a first location started to complain, tickets were opened from multiple places because it seemed that no mobile devices were able to receive any new email.

This situation quickly became a huge issue.

The support team started to verify that the impacted users had no problem with their rights to access the messaging and mobile environment, but everything looked fine.

After a few other checks, the AirWatch support team became involved and started to check the Active Sync end-point availability.

But everything was fine there as well.

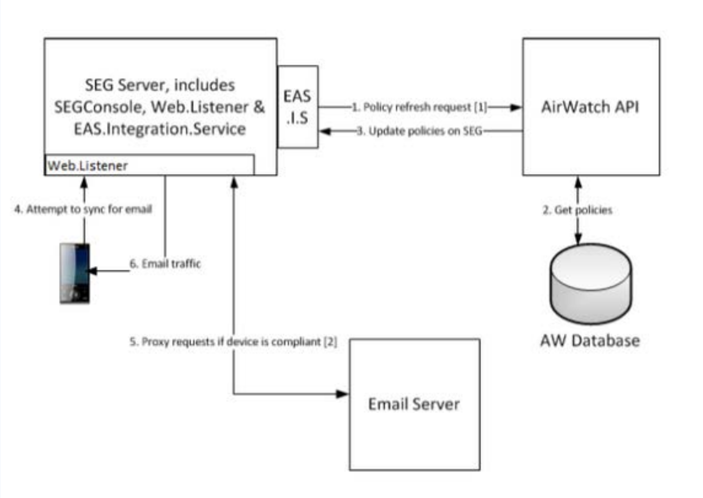
Then they went through every AirWatch server, role, and database to understand their health and what had gone wrong. The issue was identified on the SEG server. Even if the Console looked fine, no mobile device was able to access the SEG server.

So, what was the root cause? Two separate issues were detected.

The first one impacted the Web Listener of the SEG server.

After deeper troubleshooting, they understood that the IIS application pool went down, preventing the Web Listener from functioning.

The second issue was related to the EAS integration service that also went down, preventing the connection between the SEG server and the Database through the AirWatch API.

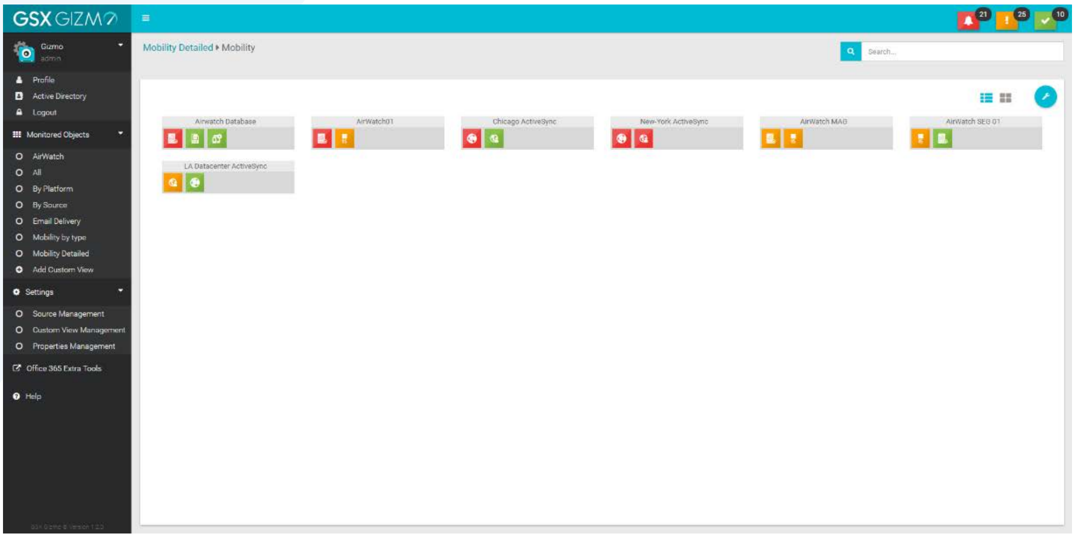

SEG Server, includes SEGConsole, Web.Listener & EAS.Integration.Service
Web.Listener
EAS .I.S
1. Policy refresh request [1]
3. Update policies on SEG
AirWatch API
2. Get policies
AW Database
4. Attempt to sync for email
6. Email traffic
5. Proxy requests if device is compliant [2]
Email Server

Because of that, the policies could not be refreshed, also preventing mobiles from getting their new emails.

This case involved a lot of troubleshooting and problematic downtime for the end-users.

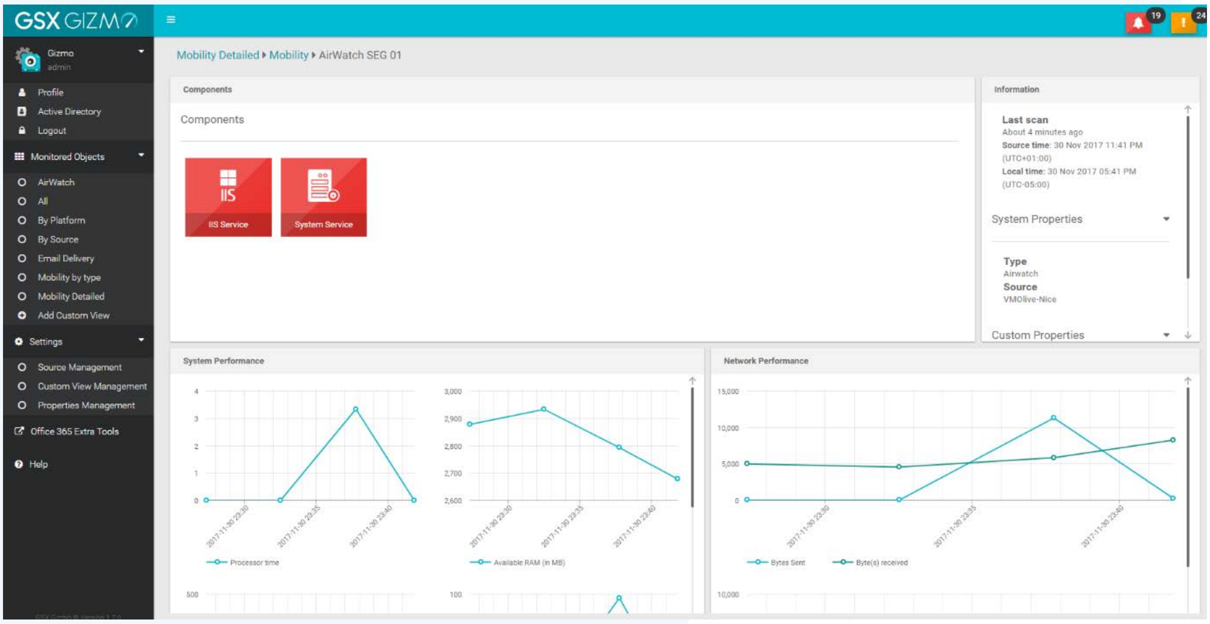
So let's see how you can easily detect that with GSX.

How GSX for AirWatch is helping



As you can see in that first screenshot, the GSX real-time interface provides you with a direct view of the health of any AirWatch server you have, as well as any other part of your mobility service.

You can drill down in each server to understand the problems.



Mobility Detailed ▸ Mobility ▸ AirWatch SEG 01

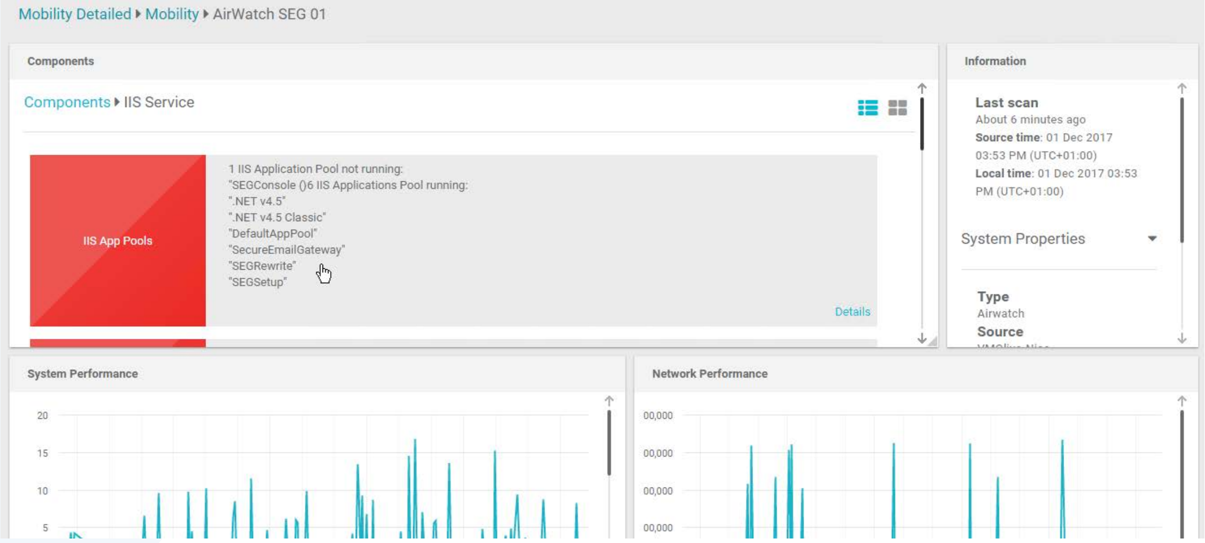
Components

Information

Last scan
About 4 minutes ago
Source time: 30 Nov 2017 11:41 PM (UTC+01:00)
Local time: 30 Nov 2017 05:41 PM (UTC-05:00)

Type
Airwatch
Source
VMOlive-Nice

Here you can immediately see that there are issues on the AirWatch Secure Gateway Server, and you can immediately identify what is causing the issue.



Mobility Detailed ▸ Mobility ▸ AirWatch SEG 01

Components

Components ▸ IIS Service

1 IIS Application Pool not running:
"SEGConsole ()6 IIS Applications Pool running:
".NET v4.5"
".NET v4.5 Classic"
"DefaultAppPool"
"SecureEmailGateway"
"SEGRewrite"
"SEGSetup"

Information

Last scan
About 6 minutes ago
Source time: 01 Dec 2017 03:53 PM (UTC+01:00)
Local time: 01 Dec 2017 03:53 PM (UTC+01:00)

Type
Airwatch
Source

As we've seen earlier, we run multiple tests on the IIS services, providing you with the status on the Applications pools and looking at each website and end-point.

You can see directly that the Web Listener is having problems, preventing any mobile to reach the AirWatch servers and then to synchronize their emails.

You can directly see as well that the EAS service is not running, preventing mobile to synchronize with the servers.

Thanks to GSX, you won't have to spend time troubleshooting your issues because you already have all the information you need in front of you to directly address the root causes.

Let's now check out a second use case that happened to the same company.

Troubleshooting local issues with ActiveSync

Situation

This time, all the opened support tickets came from one location.

The 1st level support team was involved, checking what they could in term of user's rights and sending the issue to the mobility team. They checked every server, role and database of the AirWatch environment and everything was fine.

They checked the availability of the ActiveSync end-point and everything looked ok. They then involved the messaging team, who checked the Office 365 portal for potential messaging issues. They found nothing.

Finally, after extensive research with the local network team, they found out that the ActiveSync end-point was not reachable from that particular location because of new local firewall rules.

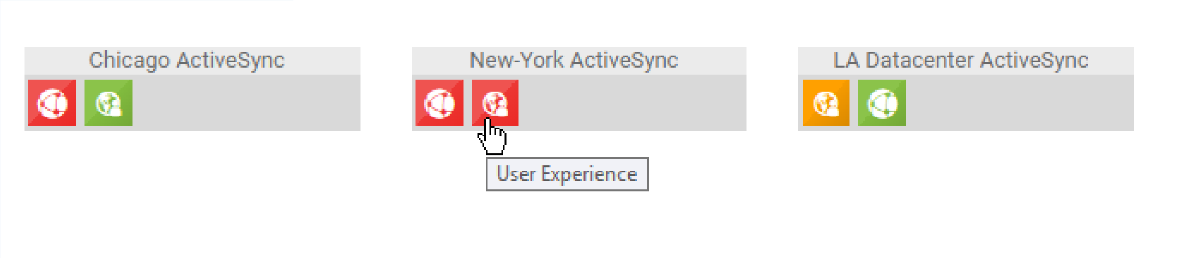
This example perfectly illustrates why the GSX Robot Users are so important.

So let's see how you can easily detect that with GSX.

How GSX helps to solve the issue

The Robot Users provide you with visibility from the user perspective. And nowadays, most companies have multiple locations and users sometime spread all over the world. The difficulty for the mobility admin team is to be able to run tests from where the users are.

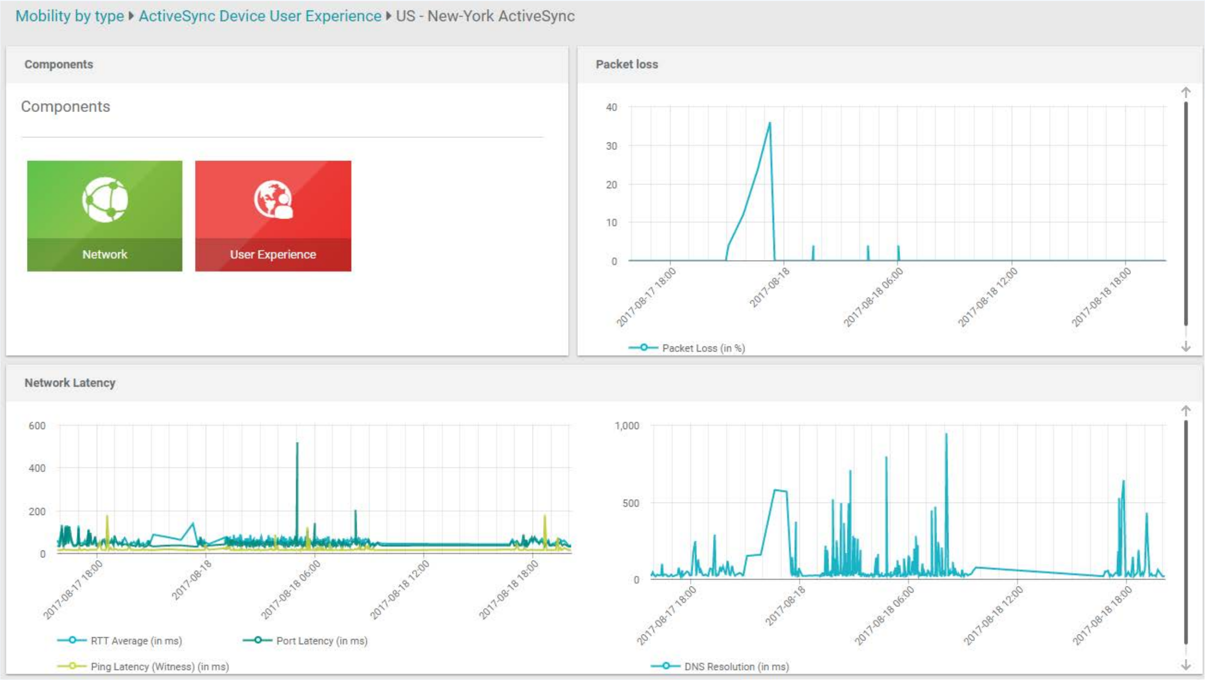
The Robot Users can be deployed easily in any of your locations and use the services as a user.



Here, using the Active Sync protocol, the Robot Users emulate and mobile devices and act exactly like it.

We can immediately see that everything works fine in Chicago, and almost fine in Los Angeles, but for sure there is a problem in the New York office.

Like we drill down to servers, you can drill down to Robot Users to look deeply into what is going on in that location.



The Robot User is performing every action as a user, so it is not just connecting to the ActiveSync end-point. It is also opening the mailbox, creating an email, a meeting from your phone, looking for the free busy statuses of the participant, etc.

Each of those actions are reproduced from your different locations.

On top of that, the network checks are done to make sure it is not the network that is the root causes of your latency.

Again, you have all the information right under your eyes that drastically cuts the time it takes for you to troubleshoot and to repair a mobility issue.

The final use case explains why it is important for mobility teams to have an overview on the overall collaboration environment to reduce the mean time to repair incidents.

Troubleshooting large email delivery issues

Situation

Once again, it's the same company with the same issues. But this time, multiple tickets were opened from multiple locations around the world and the support desk was on fire.

Pretty soon, the entire support level 3 was involved trying to understand what was going on. ActiveSync end-point was checked, and every AirWatch server was checked, but everything looked fine.

The messaging team checked Office 365 and it looked fine too. Nothing to worry about apparently. But still, users all over the world were not able to receive email anymore.

Manual mail routing was done to confirm the issue. It was possible to send mail out, but no inbound mail was received.

Once again, the mobility team was involved but had no visibility on the layers of infrastructure that are necessary to provide the mobility service to end-users.

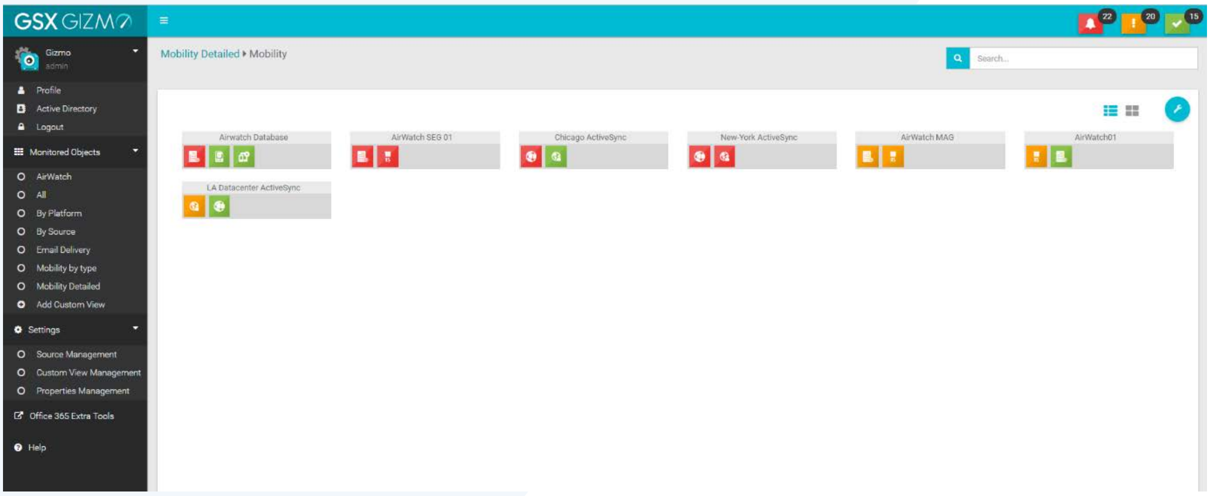
So, what happened? Thousands of emails were processed by the IronPort appliances that were then overloaded, blocking any new inbound mail.

This is another example where GSX would have warned the mobile team as soon as the root cause started.

Let’s see how you can easily detect that with GSX.

How GSX spots email delivery issue impacting the mobile devices

As you can see again here in the interface:



You can monitor your mobility service as a whole with GSX. AirWatch servers are important but as we’ve just seen, it is also important to have Robot Users reproducing Active Sync user’s actions.

The Mail Routing is another very important part to check.

Typically, you can configure every route and every mail flow that you have in your environment.

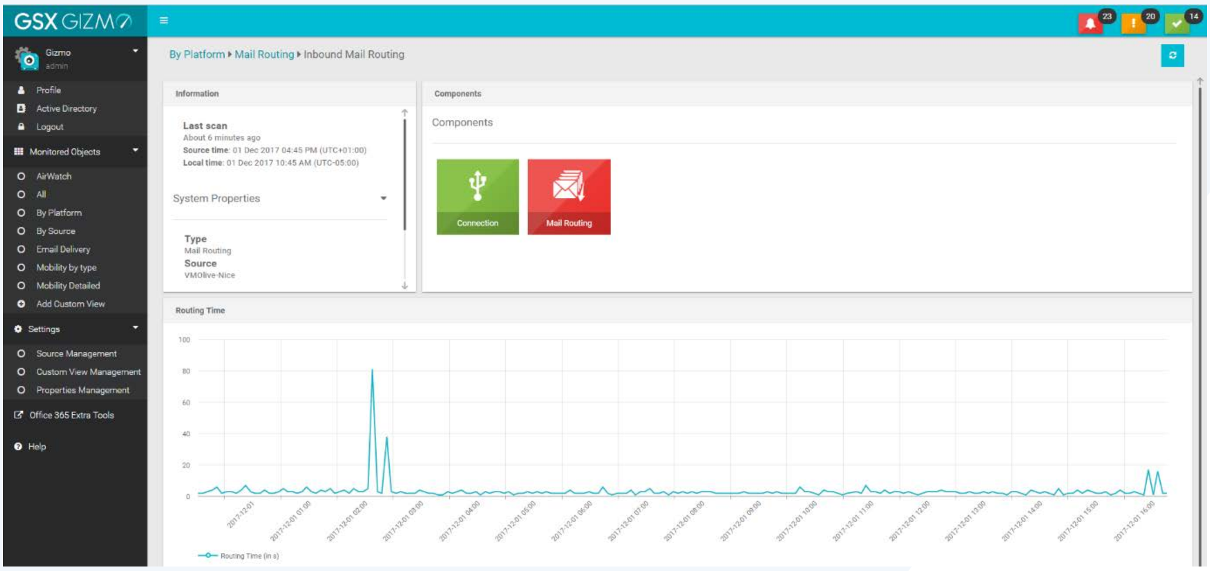
There are a lot of different way an email can go through your environment. It can go through multiple applications, or appliances for example.

Thanks to GSX, you can test all these routes in real time.

You need to know as early as possible when something breaks in order to troubleshoot, understand where the issue is, and fix the root cause.

The mobility team needs to be aware, because when users cannot send or receive email on their mobile they will contact the mobility team. You want to avoid a finger pointing situation that just makes the time to repair longer.

Once again, you can drill down in the mail routing scenario that causes an issue.



By Platform ▸ Mail Routing ▸ Inbound Mail Routing

Information

Last scan
About 6 minutes ago
Source time: 01 Dec 2017 04:45 PM (UTC+01:00)
Local time: 01 Dec 2017 10:45 AM (UTC-05:00)

System Properties

Type
Mail Routing
Source
VMOlive-Nice

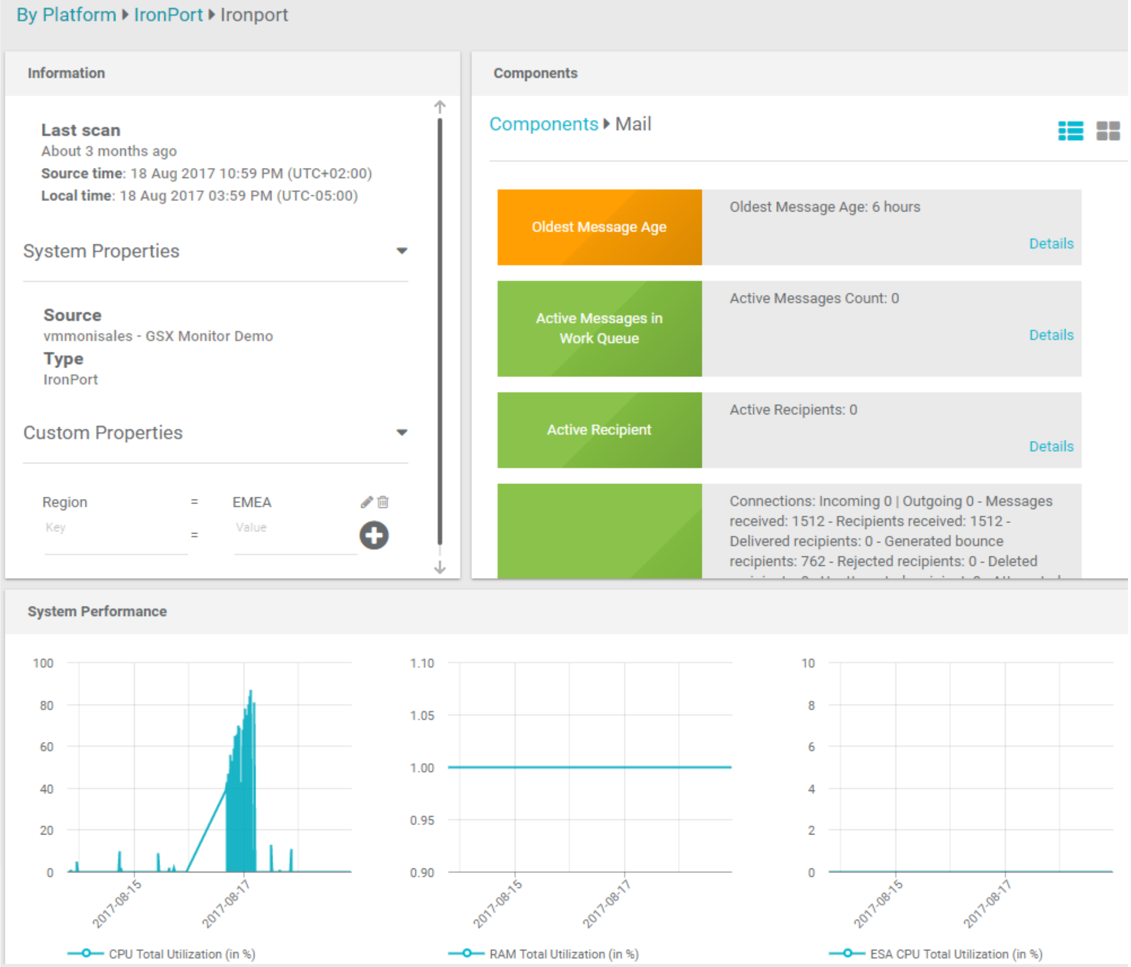
Components

Components

And you can see immediately that the inbound routing is failing and when it started.

The other important thing you have to keep in mind is that email delivery often uses proxy servers or security appliances. In this use case, the IronPort server was just overwhelmed by transactions.

You can directly see that with GSX as well.



By Platform ▸ IronPort ▸ Ironport

Information

Last scan
About 3 months ago
Source time: 18 Aug 2017 10:59 PM (UTC+02:00)
Local time: 18 Aug 2017 03:59 PM (UTC-05:00)

System Properties

Source
vmmonisales - GSX Monitor Demo
Type
IronPort

Custom Properties

Region	=	EMEA
Key	=	Value

Components

Components ▸ Mail

Oldest Message Age: 6 hours
Details

Active Messages Count: 0
Details

Active Recipients: 0
Details

Connections: Incoming 0 | Outgoing 0 - Messages received: 1512 - Recipients received: 1512 - Delivered recipients: 0 - Generated bounce recipients: 762 - Rejected recipients: 0 - Deleted

Our customers generally share an overall view of the environment and then have specific views for specific teams like the IronPort team.

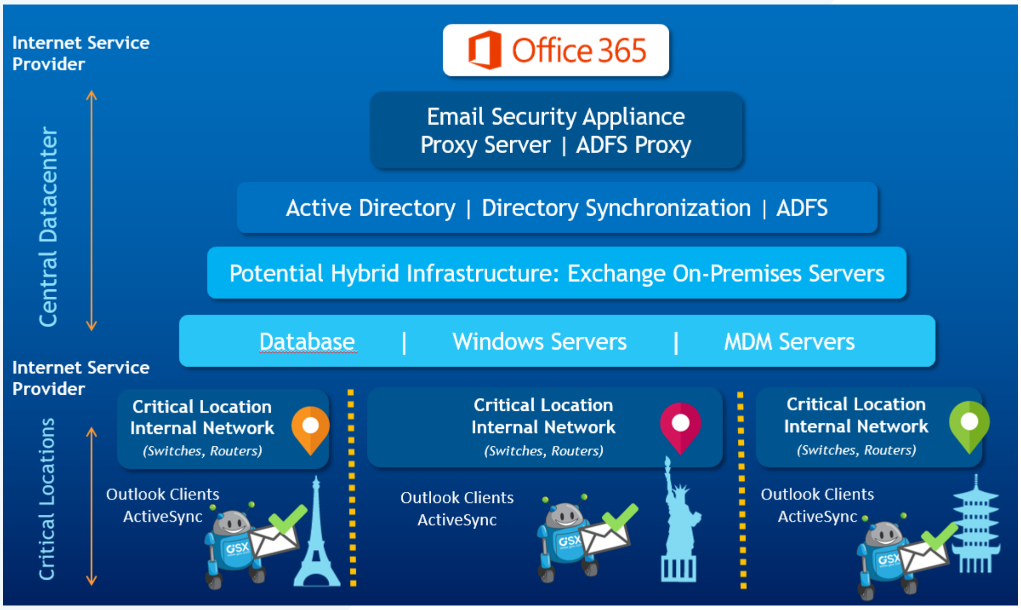
In this example, the mobility team is now immediately able to spot what is causing the issue to their mobile users, even, if the problem is coming for an infrastructure layer they are not responsible for.

Communication between admin teams becomes easier as they can share important information for the service delivery.

Once again, the mean time to troubleshoot and repair is drastically cut thanks to GSX.

The only end-to-end real-time monitoring for mobile users

It is clear that mobility services rely on the AirWatch server's health but also on other layers of infrastructure that impact the mobile end-user experience.


Internet Service Provider
Office 365
Central Datacenter
Email Security Appliance
Proxy Server | ADFS Proxy
Active Directory | Directory Synchronization | ADFS
Potential Hybrid Infrastructure: Exchange On-Premises Servers
Database | Windows Servers | MDM Servers
Internet Service Provider
Critical Locations
Critical Location Internal Network (Switches, Routers)
Outlook Clients ActiveSync
Critical Location Internal Network (Switches, Routers)
Outlook Clients ActiveSync
Critical Location Internal Network (Switches, Routers)
Outlook Clients ActiveSync
GSX

The local IT, the local network, your security appliance, your proxy, your AirWatch servers, services and databases, your ActiveSync end-point and of course Office 365 or Exchange servers all contribute to deliver mobile services to your end-users.

Monitoring the availability, health and performance of these services of course starts by monitoring the core of AirWatch services first.

And GSX is the only solution that can monitor your AirWatch servers in depth.

We check all the Web consoles availability, IIS pools, web listener, SQL database and services that allow the connection with the AirWatch API.

We collect also every performance counter you need from both a system and an application perspective to stay ahead of any potential infrastructure issues.

GSX is also the only solution that provides a measure of the end-user experience directly from where your users are. The GSX Robot Users can precisely tell you if a specific location has an issue, or if all of them are impacted.

They can tell you if the network or ActiveSync is causing problem at the location level, and in the end, if you're users are happy or not.

GSX allows you to know before your users if issues are arising to give you time to work on them before tickets are opened.

But even more importantly, GSX is the only solution that provides total visibility on everything that can impact your mobility services from your AirWatch servers, to your security appliance, to the local network of your locations and finally to Office 365 services and your Exchange servers.