WHITEPAPER

How monitoring can help keep your SQL Server estate secure

Compliant
Database DevOps

redgate

How monitoring can help keep your SQL Server estate secure

Contents

“If one aspect of security fails, then it all fails”

Introduction

Any organization is obliged to take a range of precautions to ensure the security of the data it holds. This isn't just to comply with privacy legislation but also so it can protect its assets.

There are a whole range of precautions that concern the individual members of the organization; including the network, the development and usage of data systems, and the operational supervision of these data systems. In this whitepaper, we are concerned with just the last category, the operational supervision of data systems.

It is, of course, difficult to isolate just one aspect of data system security because even the very finest of practice in security operations can come to naught if, for example, **a member of staff removes a hard disk**, or **copies sensitive data onto a laptop and allows it to be stolen**.

If one aspect of security fails, then it all fails. However, if operational supervision or surveillance of data systems is neglected, then the data is vulnerable to unexpected attacks.

Surveillance in IT Operations

The most effective systems that check on security don't stop at the point where they are able to monitor for all known means of attack against your data systems. There are just too many other imaginable possibilities, and many that aren't imaginable. The villains sometimes repeat themselves, but often they don't. To spot novel or unusual attacks require a different approach.

Yes, you need to spot the symptoms of known methods of attack, but also to adopt a broader 'surveillance' approach that will alert you to other, possibly suspicious behaviour. The technique of surveillance in this context has nothing to do with checking on people, but is all about monitoring unusual behaviour, activities, or other changing information of data systems, in order to get a more rapid warning of intrusions or suspicious activities, in whatever form they take.

You will, of course, counter a known threat such as a SQL Injection from an application by **monitoring for the typical SQL syntax errors** that occur as the attacker attempts to 'blind-navigate' the schema, guessing the names of your tables, procedures and so on. However, nothing can prepare you for a less predictable attack, when the attacker has somehow gained more knowledge of, or better access to, your database systems. You can't even easily predict their intentions, so you'll also need to be monitoring more broadly, looking for unusual patterns of activity by authorized users.

In short, it is not enough to put up the metaphorical fortified gate on the castle, you also need the people in chainmail with spears to 'stand sentry' and patrol the perimeter on watch for anything suspicious or unusual. The IT Infrastructure must accurately implement the security architecture, and be able to detect unusual events, and quickly alert the people who are best-placed to investigate, giving them enough detail to enable them to drill into the detail. The management must ensure that this system is tested to make sure that it can react to unexpected attacks.

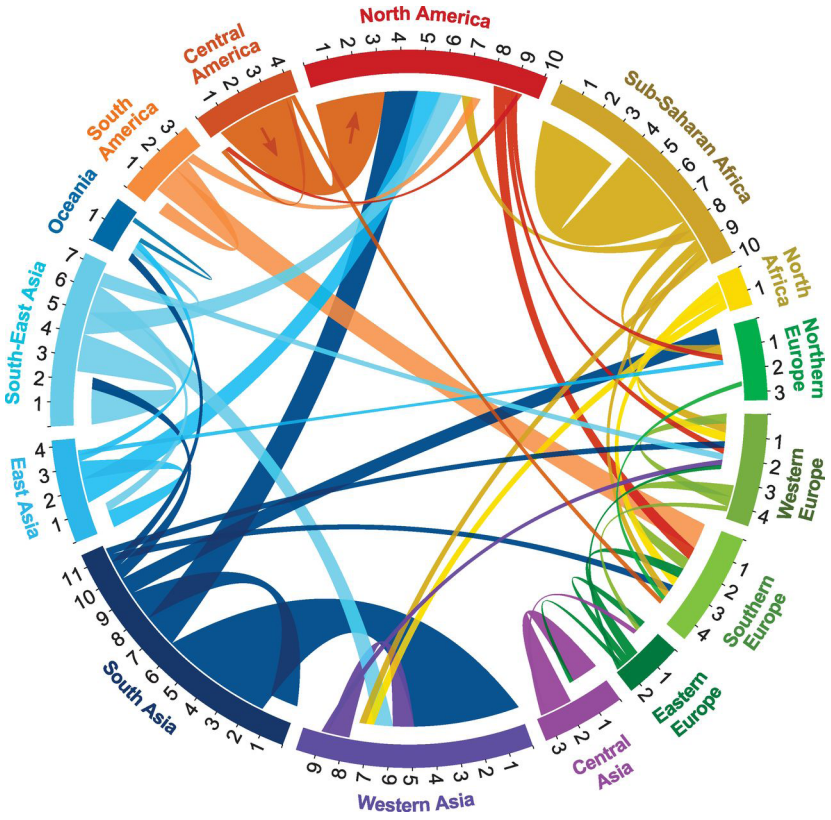
For example, knowing when the server is starting to get massively scanned allows you to prevent subsequent advanced attacks if you react quickly. Everything needs to be visible: the more visible it is, the sooner you can spot when things go wrong.

The 'sudden flicker'

The monitoring tool should be central to the way an Operations team works. It should be the first thing they open in the morning (yes, even before Outlook or Slack), and it should help them prioritize their daily tasks.

With the right solution, the standard daily server checks (which can take hours) are already done, so the DBA can move on to any problems that have arisen, target the cause, and fix them.

During the day, alerts will keep the team informed on any new problems that arise, and the monitoring tool should be flexible enough to provide alerts in the way that best suits how the team works – within the tool, via email, or through another tool like Slack. Ideally, you should also be able to determine alert severity to help prioritize them when they come in.



From Quantifying Global International Migration Flows - Abel & Sander 2014

A diagram like the one above, if played over a time period and compared with a baseline, will allow you to spot the occasional unusual flicker.

We are good at doing this. Our predator inheritance means that the human brain has an uncanny knack of spotting unusual patterns of movement in a complex landscape. Our hunter-gatherer ancestors never jumped out of bed, thinking how great it would be to catch and eat a deer today, and then stood in wait at the spot they last saw one. Instead, they waited in the cover, staring at the landscape for any sign of movement. In effect, the prey set the agenda.

Today, the attacker of any computer system sets the agenda. If they penetrate the standard and essential defences, we must detect this and react.

There is a limit to what is possible in terms of defence, merely by predicting all probable means of attack. As well, we must spot, or be alerted to, anything odd in a mass of data over time, compared with a baseline; any sudden flicker that shouldn't normally have happened. Only then, can we dive into the detail.

In a criminal investigation, we might pull over a suspicious car and search it. In IT, we drill into the detail. In both cases we don't do it on impulse, we are wanting to understand what caused that 'sudden flicker'.

Consequences of a lack of surveillance

The recent (Friday 11th 2018) attack on **VFEmail** offers a vivid example of the need for broad-based surveillance. The attack wiped all the disks on every server, erasing almost the company's entire infrastructure, including mail hosts, virtual machine hosts, and a SQL server cluster.

> **Within just a few hours, all data back to 2016 was permanently lost.**
> **"At this time, the attacker has formatted all the disks on every server. Every VM is lost. Every file server is lost, every backup server is lost…Yes, @VFEmail is effectively gone. It will likely not return. I never thought anyone would care about my labor of love so much that they'd want to completely and thoroughly destroy it."**

This devastating attack was detected only when all the servers for the service went offline, without any notice. This moment of death finally fired an alert. The intruders were stopped only when they were half way through destroying the backup servers.

The company was good at privacy security, but only guarded against the expected threats, and critically they didn't have offsite/offline backups. If the effects of the intrusion had been measured and alerted, then the intruder could possibly have been

stopped hours earlier before the damage had become irrecoverable.

Unfortunately, the style of attack was so unusual that it was never predicted and so the broader signs of such an attack were not monitored.

What would they have needed to measure? The attack took several hours, and the path of destruction was bound to have been represented by very unusual activity within the servers and disks. If you'd been in the server room at the time, you'd maybe even have heard the rustle of the disks being formatted.

A similar attack to a cloud-hosted system some years ago was even more devastating because the login, gained through an undetected brute-force attack, had enough privileges to terminate the entire service without trace.

When you don't know what to expect, you are compelled to create a baseline and monitor all the activity that you can, even the type and level of server disk activity. Then you can fire an alert when, for example, there is sustained and regular disk activity at a level that is unprecedented, or if it suddenly drops beyond the normal range.

Mutual monitoring rather than self-monitoring

Just as the deer with a spear in its hind quarters, thrown by our hunter-gatherer, would not always be best-placed to raise the alarm, so it is with a computer system. A system that is wounded or destroyed cannot fire an alert, and an attacker often disables the built-in alerting system as a first step, so that, to extend the deer analogy, the herd must monitor each other. Similarly, the wounded database server relies on another server in the system, the monitoring server, to raise the alarm.

Sarbanes-Oxley was the first regulatory framework that pointed out that a monitoring system had to be external from the system being monitored. They were focused on a special type of intrusion, that of financial fraud, where 'self-monitoring' has proved to be inadequate. The same is true of any sort of attempts at intrusion or surreptitious interference. If the first task of the intruder is to compromise or disable the alerting mechanism, then it must be as difficult as possible to do so.

Extending the monitoring of SQL Server for surveillance

All organizations that handle data do so in the face of the wide-ranging requirements, and growing legislation, as embodied by the **GDPR**, the **Payment Card Industry Data Security Standard** (PCI DSS), the **Health Insurance Portability and Accountability Act** (HIPAA), the **Sarbanes-Oxley Act** (SOX), U.S. government regulations such as NIST 800-53, and many others.

Therefore, any monitoring tool that aspires to compliance must allow for a broad range of monitoring, including both the processes of a database application, as well as the database system. This process of **Database Activity Monitoring** (DAM) collects data and aggregate it in a central location for analysis. It detects suspicious behavior in terms of activities that appear to violate the security policies or are anomalous and aims to detect malicious or suspicious activity by any of the database logins, including sysadmin accounts.

Any SQL Server monitoring tool must monitor errors, session activity, and individual SQL statements that access database objects that contain sensitive information, as well as any changes to database objects. It must also be able to monitor changes in database server and database settings.

Redgate's **SQL Monitor** already monitors the standard metrics of the SQL Service, such as **disk IO** activity. It establishes a baseline for each and allows you to set alerts when there is a substantial variance from this baseline.

From this, some additional security-focused custom metrics can be put in place, for more robust monitoring:

- Unless **authentication failure in SQL Server**, even on non-production servers, is kept under surveillance, then brute-force attacks will go undetected.
- If you do not monitor **changes in database or server configuration**, you may miss the signs of an intruder preparing to export the 'payload', a euphemism for your data.
- If you fail to monitor **certain types of SQL error**, then you will miss the signs of an attempt at SQL Injection.
- If you fail to monitor for **changes in permissions and role membership**, as well as **unauthorized changes to database objects** (database drift), you may miss attempts at fraud or data theft.

Keeping the host server under surveillance

Once you have the essentials of the SQL Server itself under surveillance, you still have work to do. Monitoring the service without checking the server itself is like buckling on your child's safety belt, without bothering to check the driver, or the overall state of the car (or, if this is a managed **Azure SQL Database** service, taxi).

Windows already logs a huge number of Windows Events from a large variety of providers within the server, and until you're monitoring all the ones judged High/Error or Medium/Warning severity, then there is no point in relaxing. After all, a brute-force password attack to gain a server login is very noisy in terms of events and many intrusions are based on the attacker gaining a Windows login that has access to the database. Any attack on the server is likely to leave its tracks in the logs and so Windows events must be checked.

Summary

> **"To expect the unexpected shows a thoroughly modern intellect"**
> **Oscar Wilde**

However effective a security system is in detecting all known methods of breaching databases, it can be completely helpless in the face of novel or unexpected methods of attack, or attacks from an unusual direction. It is, in fact, impossible to expect the unexpected; instead a very different approach of surveillance is needed.

To perform database surveillance or activity monitoring the monitoring tool must provide and track a broad range of security and general server activity metrics, and on top of this provide:

- A way of storing metrics that allows you to compare the level of a metric with other time periods, or to plot its variance over time in comparison to other metrics.
- An effective alerting system that can be configured to trigger in a variety of circumstances.
- A method of easy customization that allows a variety of sources to be monitored.
- A system for representing this monitoring data that makes it easy to spot suspicious activity, or an easy way of exporting the data to a system that can.

And finally...

Regardless of how you monitor and manage your SQL Server estate, this whitepaper has helped you understand some of the principles that will ensure you are doing so efficiently, purposefully, and with security in mind.

At Redgate, the leading provider of software for professionals working on the Microsoft Data Platform, we offer an ingeniously simple SQL Server monitoring solution that helps you proactively monitor your estate, so you and your team can find and fix issues before they become problems.

With SQL Monitor from Redgate you can continuously optimize your processes by diagnosing and resolving causes of operational and performance issues, including deployments.

From a single pane of glass, you can stay on top of your SQL Server estate, regardless of how big it is or where your servers are located.