Brought to you by:

solarwinds

Network Monitoring

for dummies®
A Wiley Brand

Understand why you need monitoring

Monitor frameworks and technologies

Discover best practices

Leon Adato

2nd SolarWinds Special Edition

About SolarWinds

SolarWinds (NYSE:SWI) is a leading provider of powerful and affordable IT infrastructure management software. Its products give organizations worldwide, regardless of type, size, or IT infrastructure complexity, the power to monitor and manage the performance of their IT environments, whether on-premises, in the cloud, or in hybrid models. The company continuously engages with all types of technology professionals — IT operations professionals, DevOps professionals, and managed service providers (MSPs) — to understand the challenges they face maintaining high-performing and highly available IT infrastructures. The insights gained from engaging with them, in places like the THWACK® online community, allow SolarWinds to build products that solve well-understood IT management challenges in ways that technology professionals want them solved. This focus on the user and commitment to excellence in end-to-end hybrid IT performance management has established SolarWinds as a worldwide leader in network management software and MSP solutions. Learn more today at **www.solarwinds.com.**

Network Monitoring

2nd SolarWinds Special Edition

by Leon Adato

for dummies®
A Wiley Brand

Network Monitoring For Dummies®, 2nd SolarWinds Special Edition

Published by
John Wiley & Sons, Inc.
111 River St.
Hoboken, NJ 07030-5774
www.wiley.com

ISBN: 978-1-119-60303-0 (pbk); ISBN: 978-1-119-60302-3 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Publisher's Acknowledgments

Some of the people who helped bring this book to market include the following:

Project Editor: Carrie Burchfield-Leighton

Editorial Manager: Rev Mengle

Acquisitions Editor: Ashley Coffey

Business Development Representative: Molly Daugherty

Introduction

Picture this scenario: You get to your desk at 9 a.m. sharp, having had a great morning workout, followed by a shower, a fantastic cup of coffee, and a frustration-free drive to the office. You're fresh, focused, and ready to make a serious dent in that growing to-do list, which includes curious items like users complaining that "the Internet" gets really slow every so often, and the CFO thinks the company's overpaying for WAN bandwidth. How much are you using?

Logging on to your PC, you notice no emails have come in overnight. "That's odd," you're thinking. Seeing you arrive, your buddy now walks over and says, "Looks like something's wrong with email." You log in to the email server and find out that it's . . . well, you don't actually log in to the email server. The remote desktop won't make a connection. You try pinging the box, and there's no response. You wonder if the problem is in the network or somewhere else in the system. With a sinking feeling, you make the long journey to the server room. All hope of working on your to-do list is now gone as you stab a finger at the server's power switch. A few moments later, you're logged on at the console. A pop-up alert on the screen tells you that one of the drives is completely full.

Much . . . (much!) later in the day, a picture forms of what happened. Sometime during the night (2:30 a.m. to be exact) the data drive filled up, causing mail services to stop. Shortly after that, errors on the system drive reached a critical point, and the entire system crashed. Meanwhile, in the heat of fighting this fire, you didn't dig deeper to note the data drive has been hovering at 95 percent capacity for over a week. And the drive containing the operating system has been throwing read/write errors every 15 minutes for the last 17 days.

About this time, your manager, who's been keeping a respectful distance while you worked, lets you know the CEO is back from his contract discussions overseas. During the flight home, the CEO needed to send some follow-up documentation to the customer. When the corporate email wasn't responding, he resorted to creating a professional-sounding Gmail account and sent the files from there. The three of you are scheduled to sit down and debrief

the situation in 30 minutes. You start to pull some notes together for what you predict will be an uncomfortable conversation. Well, it *was* going to be a great day.

About This Book

The situation in this Introduction may be a typical one for you in your Information Technology (IT) monitoring scope. If you can relate, then this book is for you! *Network Monitoring For Dummies,* 2nd SolarWinds Special Edition, provides an introduction to IT monitoring for someone who is familiar with IT in general but not with monitoring as a discipline. As such, (almost) no former knowledge or experience is required before delving into the chapters of this book. If you already have experience with monitoring, this may not be the book for you. But then again, couldn't we all use a refresher? It couldn't hurt.

I have attempted to make this book tool-agnostic. The purpose of this book is to give you a basic understanding of why you need monitoring, what the monitoring tools are, and some best practices of networking monitoring.

Icons Used in This Book

This book uses the following icons to call your attention to information you may find helpful in particular ways.


REMEMBER

The information marked by this icon gives you certain details that are important to remember. This way, you can easily spot noteworthy information when you refer to the book later.


TIP

This icon points out extra-helpful information, including ways to save time, money, and headaches.


WARNING

Paragraphs marked with the Warning icon call attention to common pitfalls you may encounter.

IN THIS CHAPTER

- Discovering that monitoring isn't side work
- Seeing the benefits of network monitoring
- Understanding how an effective monitoring solution is built

Chapter 1
Monitoring as a Discipline

Monitoring as a discipline means devoting your focus as an IT professional to ensuring your network, servers, applications, and so on are all stable, healthy, and running at peak efficiency. It means not only knowing when a system has crashed but also when a system *will* crash, and intervening so the crash is avoided. This chapter gives you insight into monitoring as a discipline, the benefits of monitoring, and building an effective monitoring solution.

It's Not Side Work

Currently, many IT shops run without significant monitoring solutions. Others go about it in a piecemeal fashion, allowing teams or even individuals to deploy solutions with no thought to interoperability, scalability, or standards. But in the not-so-distant future, I hope the idea of having a monitoring team is as natural as the teams of network, server, virtualization, storage — and yes, security — administrators we have today.

To get to that future, people who have an interest and a passion for monitoring need the information to get up to speed on common terms, concepts, and techniques, and then they need the tools to turn that knowledge into results.

Looking at the Benefits

If you've worked in IT for more than 15 minutes, you know that systems crash unexpectedly, users make bizarre claims about slow Internet, and managers request statistics that leave you scratching your head. The answer to all these challenges (and many, many more) lies in effectively monitoring your environment, collecting statistics, and/or checking for error conditions so you can act or report effectively when needed. This goes well beyond a passive "make sure everything is green" approach to one that includes resource optimization, performance optimization, and proactive prevention and remediation.

TIP

Industry studies peg the cost of downtime in the hundreds of thousands of dollars per hour, so the benefits of monitoring are indisputable:

- Improved operational efficiency and reduced costs
- Improved time-to-resolution and reduced downtime
- More efficient use of resources

Building an Effective Monitoring Solution

Attaining the benefits of monitoring is easier said than done. Saying "let's monitor our IT environment" presumes you know what you should be looking for, how to find it, and how to get it without impacting the system you're monitoring. You're also expected to know where to store the values, what thresholds indicate a problem situation, and how to let people know about a problem in a timely fashion.

Yes, having the right tool for the job is more than half the battle. But, it's not the whole battle, and it's not even where the skirmish started. To build an effective monitoring solution, the true starting point is learning the underlying concepts. You have to know what monitoring is before you can set up what monitoring does.

REMEMBER

Network monitoring is the practice of continuously monitoring the network and providing notifications to an administrator when a network element fails. Monitoring is usually performed by software or hardware tools and doesn't have an effect on the operation or condition of the network. Monitoring can be performed passively or actively.

IN THIS CHAPTER

- Looking into monitoring basics
- Knowing monitoring technologies
- Diving deeper into monitoring your network

Chapter 2
Monitoring 101 and Beyond

Every monitoring system, regardless of vendor or packaging, utilizes basic principles and technologies. This chapter lays out those core techniques of monitoring your network.

Defining Monitoring Basics


REMEMBER

A few fundamental aspects of a monitoring system exist across the board, no matter what software you use, or the protocol, or the technique. These basic technologies used for monitoring include the following:

- **Element:** An *element* is a single aspect of the device you're monitoring, which returns one or more pieces of information.
- **Acquisition:** How you get information is another key concept. This process is called *acquisition.* Does your monitoring routine wait for the device to send you a status update (push), or does it proactively go out and poll the device (pull)?
- **Frequency:** How often information comes back is called *frequency.* Does the device send a "heartbeat" every few minutes? Does it only send data when there's a problem?

- **Data retention:** Monitoring, by its very nature, is data-intensive. Whether the acquisition method is push or pull, those statistics typically have to go somewhere, and they pile up pretty quickly. At its simplest level, *data retention* is a Yes or No option. Either the statistic is 1) collected, evaluated, acted on, and then forgotten, or 2) data is kept in a data store.
- **Threshold:** One of the core principles of monitoring is that you collect a statistic and see if it has crossed a line of some kind. It can be a simple line (is the server on or off?), or it can be more complex. Regardless, that line, which is crossed, is called a *threshold.*
- **Reset:** *Reset* is the logical opposite of threshold. It marks the point where a device is considered "back to normal."
- **Response:** What happens when a threshold is breached? Response defines that aspect. A *response* could be to send an email, play a sound file, or run a predefined script.
- **Requester:** From what point in the environment are the monitoring statistics being requested? In its simplest terms, you have two choices: either a piece of software running on the monitored device itself (for example, an agent), or some location outside of the monitored device (agentless).

Monitoring Technologies

Regardless of what monitoring vendors will have you believe, a finite and limited number of technologies can be used to monitor. Where the sophistication comes in is with the frequency, aggregation, relevance of displays, ease of implementation, and other aspects of packaging. These technologies include the following:

- **Ping:** Ping sends out a packet to the target device, which (if it's up and running) sends an "I'm here" type response. The result of a ping tells you whether the device is responding at all (up) and how fast it responded.
- **SNMP:** Simple Network Management Protocol (SNMP) has a few pieces that combine to provide a powerful monitoring solution. SNMP is comprised of a list of elements that return data on a particular device. It could be CPU or the average bits per second transmitted in the last 5 minutes. SNMP

provides data based on either a Trap trigger (when one of the internal data points crosses a threshold) or an SNMP poll request.

- **ICMP:** The Internet Control Message Protocol (ICMP) is used by network devices, such as routers and switches, to send error messages indicating that a host isn't reachable along with some diagnostics.
- **Syslog:** Syslog messages are similar to SNMP traps. A syslog service or agent takes events that occur on the device and sends them to a remote listening system (Syslog destination server).
- **Log file:** An application or process writes messages to a plain text file on the device. The monitoring piece of that comes in the form of something that reads the file and looks for trigger phrases or words.
- **Event log:** Event log monitoring is specific to Windows. By default, most messages about system, security, and (standard Windows) applications events are written here. Event log monitors watch the Windows event log for some combination of EventID, category, and so on, and perform an action when a match is found.
- **Performance monitor counters:** Performance monitor (or PerfMon) counters are another Windows-specific monitoring option that can reveal a great deal of information, both about errors on a system and ongoing performance statistics.
- **WMI:** Windows Management Instrumentation (WMI) is a scripting language built into the Windows operating system that focuses on collecting and reporting information about the target system.
- **Script:** Running a script to collect information can be as simple or complicated as the author chooses to make it. In addition, the script might be run locally by an agent on the same device and report the result to an external system. Or, it might run remotely with elevated privileges.
- **IP SLA:** Internet Protocol Service Level Agreements (IP SLAs) are a pretty comprehensive set of capabilities built into Cisco equipment (and others nowadays, as they jump on the bandwagon). These capabilities are all focused on ensuring the WAN, and more specifically VoIP, environment is healthy by using the devices that are part of the network infrastructure, instead of requiring you to set up separate devices to run tests.

- **Flow:** Standard monitoring can tell you that the WAN interface on your router is passing 1.4 Mbps of traffic. But who is using that traffic? What kind of data is being passed? Is it all HTTP, FTP, or something else? *Flow* (most commonly referred to as *NetFlow*) monitoring answers those questions. It sets up the information in terms of conversations and monitors who, what, and how network traffic is being used.

Going beyond Monitoring Basics

Monitoring your network allows you to be alerted to possible potholes before your users hit them at top speed. In this section, I provide deeper insight for monitoring your network.

Device availability, fault, and performance

In most modern network monitoring systems, devices are monitored for the following:

- Availability (is the device reachable?)
- Faults (detection, isolation, correction, and logging of network events)
- Performance (efficiency of the network, including throughput, utilization, error rates, and response time)

REMEMBER

Monitoring here relies primarily on SNMP and ICMP, with more advanced monitoring taking advantage of packet inspection. Some of the key metrics you should look at include response time and packet loss, CPU load and memory utilization, and hardware health details.

Traffic and bandwidth

Understanding how network bandwidth is being used is critical in ensuring the availability and performance of business services. Bandwidth and traffic usage are most often monitored using the Flow (most commonly referred to as *NetFlow*) technology that is built into most routers by looking at "conversations" between devices.

When monitoring traffic and bandwidth, pay attention to

- Interface utilization
- Applications, users, and protocols generating traffic (who and what are generating traffic)
- Endpoints (where traffic is coming from and going to)
- Conversations (who is talking to whom)

WAN

You may not own the WAN between your sites and remote locations and can't directly monitor the fault, availability, and performance of the devices within the WAN. If that's the case, you can use a technology such as IP SLA to generate synthetic traffic or operations to measure the performance between two locations or devices, determining the performance of the WAN.


REMEMBER

IP SLA is especially beneficial when monitoring certain applications that are particularly sensitive to delay, jitter, or packet loss such as VoIP or video streaming.

IP address monitoring

A network can have thousands of IP addresses in use at any given time. A duplicate IP assignment, exhausted subnet or DHCP scope, or misconfigured DHCP or DNS service will cause a network fault.


TIP

Look for a solution that monitors these IP resources and that can proactively alert you of problems to help you plan for orderly expansion.

Discovering the Different Monitoring Tools

After all is said and done, you still need to buy or build a tool (or set of tools) that help you monitor all the elements of the IT stack. This can be done with discrete specialized tools that monitor a specific element (for example, network monitoring, storage monitoring, virtualization monitoring, and so on) or with a fully integrated suite of products that provides a common platform

across the entire stack. Each approach has its advantages and disadvantages.

Regardless of which approach you choose, all software vendors are selling solutions that work from the same basic playbook. What should you look for as a differentiating factor? What is it, exactly, that makes brand X so much better than brand Y? The answer has as much to do with you and your organization as it does with how monitoring is performed.

Will your monitoring team be one person who is also your server team, network team, help-desk team, and database team? If so, you probably need a tool that sacrifices comprehensive options for simplicity and manageability. Does your organization need absolute flexibility so that the monitoring solution is the one-stop-shop for all your needs? You will pay more, and require more staff, but at the end of the day (or month, or more likely year) you will have a software suite that fits you like a glove.

With all of that said, the nontechnical items you should consider include the following:

- **Cost to purchase and install:** This includes hardware requirements and the specific needs for your environment. Do you need a separate system to monitor devices in your firewall and/or remote sites? How many monitoring systems do you need for all the devices in your company? And so on.
- **Ongoing maintenance cost:** These include license costs in year two and beyond.
- **Support requirements:** How many people are needed to maintain the system? This is one of those questions that you should *never* trust the vendor to answer. Talk to some other companies that are using the software.
- **How much customization is needed?** Again, talking to other companies is extremely useful here.

To learn more about SolarWinds network monitoring solutions, visit `www.solarwinds.com`.

IN THIS CHAPTER

- Digging into what hybrid IT means
- Seeing the benefits of network monitoring
- Discovering what's almost as good as authority (hint: it's not chocolate)

Chapter 3
Monitoring Systems in the Cloud

According to the Cisco Global Cloud Index, by the year 2020, about 98 percent of all compute workloads will be processed by a cloud-based architecture. Out of that jaw-dropping percentage, 66.5 percent (or 68 percent of all cloud-based compute workloads) will be in public-cloud spaces, such as Amazon Web Services and Microsoft Azure, leaving 31.5 percent in private-cloud environments. Just these few numbers should explain the current gold-rush atmosphere surrounding investment in and development of cloud-centric tools and solutions.

This has created a techno-economic feedback loop where companies move their data and workloads to the cloud, creating interest and investment, which builds new and better solutions, generating excitement that causes even more companies to move their data and workloads to the cloud.

This growth has continued for several years now, but one area has lagged conspicuously behind: monitoring. Part of the reason for this is the rate of change itself. Every time you think you've nailed down what "cloud-centric monitoring" is, the technology shifts, and you have to reevaluate.

But that's no longer the case, and any IT pro with an eye to monitoring must consider the cloud — both in terms of monitoring

that runs from the cloud as well as solutions that can monitor networks and systems in the cloud.

Hybrid IT: The Technology Nobody Ever Asked For

Despite its value in describing the state of corporate cloud usage, the Cisco Global Cloud Index report doesn't capture the reality for today's boots-on-the-ground IT professionals. For that perspective, you must look at a different set of numbers.

SolarWinds publishes the IT Trends Report (`it-trends.solarwinds.com`). The 2017 study tracked cloud adoption. It revealed that although 95 percent of those surveyed have moved something to the cloud in the past 12 months, just 1 percent of the respondents were 100-percent cloud-based. The majority had up to 25 percent of its architecture in the cloud; the rest remained on-premises. This has come to be known as hybrid IT.

You know what no IT professional has ever said?

"Wouldn't it be awesome if we split all our critical infrastructure so that some of it was here in our office, and some was operated by complete strangers. And I don't just mean remote data centers — I mean those nameless, faceless people would write the software, program the network, do the updates, everything. And our business would totally rely on that software. And not only that, but also all the connectivity between our office and that business-critical application would be owned and operated by any number of other organizations, and we'd have no idea who they were. But best part is that we — you and me — would still be responsible for all of it. If something happened on interface 3 of router 17 of our ISP's ISP's ISP, it would still be our responsibility to figure it out and get it fixed. Doesn't that sound wonderful?"

And that, my friends, is hybrid IT.

Hybrid IT is the condition in which part of your infrastructure is in the cloud and part of it remains on-premises, with your team responsible for all of it — including the Internet, which connects the two. It's not only a reality, but it's also the norm among companies, according to the SolarWinds IT Trends Report. Although very few companies have resisted the urge to move something to the cloud, very few companies have moved everything "up there."

REMEMBER

The reason I bring this up isn't to depress you; I just want to point out that a "pure cloud" monitoring solution isn't going to cut it any more than a purely on-premises solution will. You need tools and techniques that do both and also bridge the gap between the two.

The Secret to Success

It can be argued that, to be successful as an IT professional, you need three things:

- Responsibility
- Accountability
- Authority

Most of you probably get *responsibility* right after you're hired and before you even figure out where the restrooms are. Heck, just passing within five feet of a server can make you responsible for it in some companies.

The same goes for *accountability*, which is why you've probably had phone calls like this one:

> "Hey, Leon, the database on Server 19 is acting funny. Yeah, I know you're not the DBA. Yes, I know you're not the server admin either. But I clearly heard you from two cubes away mention Server 19 last month, so I figured you were doing something on it, and now the database is acting all weird, and . . . look, I know it's 2 a.m., and I'm not happy about having to call you either, but could you just take a look at it?"

Authority, on the other hand, is what you usually must fight tooth and nail to get. And, of course, it's the one thing that's crucial to be effective at your job. You need to have the authority to make decisions, approve purchases, and issue guidelines that are respected and followed.

While all of this has been true since IT went GOTO 0 the very first time, it's even more tenuous in the modern hybrid IT world. In fact, the SolarWinds IT Trends Index showed that over half of IT professionals consider lack of control over the performance of cloud-based workloads a top challenge and still a considerable barrier to migration.

But this is IT, so of course there's a hack — a workaround that gives you something almost as good as authority, and that's *visibility.* If you can see the environment and its current state, you have a better chance of getting ahead of any issues that crop up. And if you can see the specifics of those issues — the more details the better — and communicate those details to the true owner, you have a much faster route to resolution, which in turn relieves some of the stress associated with moving a workload to the cloud. This sounds a lot like good, plain old "monitoring." And the fact is, much of it is good, plain old monitoring. But hybrid IT and the cloud have forced IT pros to reimagine some old techniques with a new spin and invent other techniques out of whole cloth.

Unraveling the Internet

In a hybrid IT environment, the biggest challenge by far, in terms of the authority/visibility conundrum, is the Internet. With every other piece of the puzzle, you have some fundamental insight. On-premises systems are onsite, so they're a piece of cake. With the cloud-based systems, you can still lean on the vendor-customer relationship.

WARNING

With the Internet itself, you need to overcome at least two hurdles:

- **It's massively multipath in nature.** Any given packet can take any one of a number of different routes to the destination, and unlike your network, the routes (or paths) through the Internet can (and do) change at any moment.
- **Your connection to your cloud-based environment may start with your ISP, but it doesn't end there.** Your ISP has an ISP that may have its own ISP. And working from the other end, your cloud provider has an ISP that likely has an ISP as well.

Like many IT challenges, after you can identify it, you're halfway to solving it. The answer lies in finding a system that maps out all those nodes across the Internet. You need something that does it in a way that gives you information about each of the devices (hint: not ping) and isn't blocked by intermediate firewalls (spoiler: not traceroute).

IN THIS CHAPTER

- Exploring the philosophy of alerting
- Effectively triggering alerts
- Informing the alert recipient with meaningful alerts

Chapter 4
All about Alerts

For many companies, teams, and IT professionals, alerting is seen as the reason for monitoring. If you can't get an alert when something is going wrong (so the thinking goes), why bother monitoring at all? At the same time, alerting is also seen as the curse monitoring brings because it's a source of constant interruptions, false alarms, and "noise." Or at least, that's how it is sometimes perceived.

The reality is that monitoring alerts, as a blessing or a curse, depend largely on the design and implementation of those alerts, more so than any specific monitoring tool or technique.

In this chapter, I lay out alerting concepts and techniques, so you can avoid the noise and create alerts that are efficient, effective, and actionable.

The Philosophy of Alerting

Before digging into the specifics of how to set up alerts, it's important to understand a bit more of the why and what. Why are alerts viewed with everything from annoyance to downright loathing in many organizations? What are some things IT professionals can do to create effective monitoring alerts? This section helps you answer those questions.

Alerts should not be noise

Often, you hear the accusation that network monitoring alerts are just noise. One reason for this is the common mistake of turning on all alerts that come out-of-the-box in the monitoring solution. The logic goes, if vendors thought they were good enough to include by default, they must be good for most (or all) companies. Nothing could be further from the truth.

The alerts you find built in to monitoring software are often included as examples of specific techniques or types of alerts. For example, if the monitoring solution can send a message when Linux CPU is over a certain threshold and include a list of the top ten processes running at that time, you'll likely find a sample alert that does just that — not because CPU over 90 percent is a particular best practice, but because it's easily understandable for someone who wanted to see that technique in action.

WARNING

Treat the out-of-the-box alerts as polite suggestions because often they're included by the vendor simply to show you what could be done with alerting, and how to set it up.

Alerts are considered noise in organizations for many reasons:

- **Too many:** Alerts trigger too often, so you may feel like you've barely responded to one when another shows up.
- **Not enough detail:** The alert message says something like "system down" without any other details — not helpful.
- **False alarms:** Of course, when the alert turns out to be just plain wrong, nobody is happy.
- **Too sensitive:** The alert indicated something happened, but it wasn't bad enough to require someone to respond.

The good news is that these issues are all extremely solvable with a bit of planning and forethought.

Alerts should be actionable

In a relay race, only one person carries the baton at a time. But that doesn't mean the only person who cares about the baton is the one carrying it. In fact, many people care about both the baton and the race itself — everyone from the coach, the other

teammates, the spectators in the stands, and more. But even if thousands of people care about the baton's progress around the track, the fact is that there is still only one person carrying it at any given moment.

In the race to mean time to repair (MTTR) that IT professionals run every time there's an issue in their environment, monitoring alerts are the baton. Only one recipient gets the alert.

WARNING

You can take this "relay race" metaphor too far. When I say that the alert goes to just one recipient, I don't mean that it has to go to a single individual. The recipient can be a team. My point is, the alert goes to the person or people who have a responsibility to fix the issue, and not to people who are interested or supportive but not involved in fixing the situation. That means there is no such thing as an "FYI" alert. If you get the alert, you're responsible for fixing the issue. If you aren't responsible for fixing the alert, then you shouldn't be getting the alert.

Alerts are not just emails

Many IT professionals and monitoring teams are under the mistaken impression that the only output of an alert is an email. There are many actions that an alert can perform once triggered — most of which I talk about in the later section "Automated Alert Actions."

Here, I want to focus on the messaging that comes from alerts. Of course, it can be an email. But it's just as valid for an alert notification to take the form of a ticket within the corporate help-desk system. It's also possible for an alert notification to go to a messaging system such as Slack, Skype for Business, or a self-hosted messaging platform. It's also possible that the alert goes to a logfile, which can be sent in real time to a variety of displays.

TIP

Think about the output of your alerts in ways that go beyond the simplistic or out-of-the-box behavior. Match your messaging to the need you have of how that information will be consumed.

Alerts should not be noise. Alerts should be actionable. Alerts are not just email. Those ideas are important, but they certainly don't tell you *how* to create good monitoring. Hop to the next section to dig into the details.

Getting Trigger Happy

The first stop in your quest to create alerts that are meaningful, effective, and actionable is the alert trigger itself. If many alerts are guilty of triggering too often or too soon (and they are), then you should look at why that is and what you can do about it.

Setting a trigger delay

Let's take one of the simplest alerts — the old "Tell me when the system is down" (when it stops responding to ping). Should you send an alert when a single ping is missed? When 5 are missed? How about 20?

The reality is that in any environment pings momentarily fail all the time. Triggering after a single failed test would create an uncomfortable amount of alerts that, ultimately, are proven to be false alarms. Inserting a delay allows the network monitoring solution to determine whether the device is truly down or just momentarily busy and unable to respond to that one ping.

REMEMBER

This concept of a delay becomes even more important when you expand beyond simplistic "is it down" alert types and into examples where you want to know when an application has high CPU at the same time there's a high user-connection count and long-running queries.

But there's a gotcha you need to keep in mind. All monitoring systems have three aspects that you must fully understand before you start injecting alert trigger delays into the mix. They are

- **The polling cycle:** The polling cycle is the frequency with which the network monitoring solution goes out and gathers data from the target devices. It could be a blanket 5 minutes; or it could be 2 minutes for up-down data, 5 minutes for hardware metrics, 7 minutes for network statistics, and 15 minutes for disk information.
- **The alert trigger query cycle:** The alert trigger query cycle refers to the concept that an alert trigger is simply a query that is run against the monitoring database, checking (querying) for various conditions. When the trigger returns data, that means there's an alert condition. This alert trigger query is run at some interval, whether that's every 60 seconds or every 5 minutes.

» **The alert trigger delay you're introducing:** The alert trigger delay is the delay this section is asking you to consider including.

REMEMBER

When you introduce an alert trigger delay, make sure you understand the polling cycle and alert trigger delay and account for it so that your delay is really a measurement of how many polling cycles you want to delay, not just how many minutes you want to wait.

Single element triggers

IT environments are a complex web of interdependent connections between networks, servers, services, and data. Network monitoring alert triggers are, of necessity, going to reflect that complexity. Take a "simple" alert as an example: disk full. At first blush, you'd think it's just a matter of setting a threshold to trigger when "percent disk utilization" is over some number (for the sake of argument, let's say 90 percent). But what about those spiffy 3 terabyte (TB) drives you can pick up for under $100? Your simple 90 percent threshold will trigger when the drive is down to a measly 300 gigabytes (GB). Three. Hundred. Giga. Bytes. I don't know about you, but if someone woke me up at 3 a.m. because there was "only" enough space to store 7,500 CD-quality song files, I'd tie that person down and stick a hard drive where the SCSI port don't shine.

Instead, it's perhaps better to add a second element that looks at the actual space. So, the trigger becomes this:

WHEN "percent disk space utilization" > 90 percent

AND "Disk space remaining" < 20 GB

Another situation I encounter is when I ask, "When you get that alert, what do you do?" and the answer is "Well, nothing. At least not after the first one. But if I get three of them in 15 minutes, I know it's a problem, and I jump on it." Let's be clear: I don't get paid by the bushel for sending out alerts.

Trigger delays are all about letting you know when a condition has persisted for a certain amount of time. But there's also the ability to alert when a problem has occurred a certain number of distinct times.

In addition to "when it happens X times in a row," some monitoring solutions trigger when a condition occurs X times out of Y polling cycles. So maybe you want to alert when the number of customer connections spikes over 100, but only if that happens three out of five polling cycles. You'd get an alert if customer connections were 100+ on polling cycle 1, 2, and 3. But you would also get an alert if the value was high on polling cycles 1, 4, and 5.


TIP

Learn to look past the simple and simplistic alert triggers (often these are the ones that come out-of-the-box from the vendor) and think about the conditions that truly indicate an actionable problem.

Hitting reset

Another aspect of alert triggers that frequently gets overlooked is actually the anti-trigger: the reset threshold. The idea of a reset is that it's a test, which, when met, tells the network monitoring solution to consider the problem to have reversed.

In most cases, IT professionals set this to "when the trigger is no longer true" and call it a day. Trigger when CPU is over 90 percent? Well, if it's 89 percent we're obviously good. Close that ticket and grab a beer! While this logic is true in some cases, I have found that more often it's a wasted opportunity you could use to make alerts less noisy.

Go back to the disk alert I introduced in "Single element triggers" for an example of how this might work:

If you recall, I said I would alert when "percent disk utilization" was over 90 percent AND "disk space remaining" was under 20GB.

Using a simple reset, if the disk utilization goes to 89 percent OR the disk space jumps to 21GB, the alert will clear. That's not particularly reassuring.

Instead, I can set the logic to reset when "percent disk utilization" is under 80 percent *or* "disk space remaining" is over 50GB.

On top of that, the reset should have a time delay so that I don't end up resetting the alert until it is truly all clear.

Making Meaningful Messages

Even if you have insightful trigger logic that only creates alert notifications when there's a verifiable and actionable issue, all your hard work can be foiled by a confusing or terse message that causes the recipient to waste valuable time trying to figure out what actually happened, or worse, simply ignore the alert all together.

TIP

The good news is that solving this is very simple: Add more information to your messages!

Where many IT pros get stuck is understanding exactly what information is useful and what creates confusion or obscures the issue. And that's where I can help. Here is your official checklist of the data elements that, I believe, must be included in every message:

- **The "identification" of the device:** Includes the local name, the DNS name, the IP address, and more
- **Information about the operating system:** OS name, version, and so on
- **Other information to identify the device:** Includes location, group, owner, and so on
- **The time the alert occurred:** Which may be different than the time the notification is sent out
- **The current value of the problem elements:** Including the values at the time of the problem AND at the time the alert is sent out
- **The threshold values which were breached:** So that the responder understands how far over or under the current state is
- **The duration of the problem:** The time of first detection versus the time the alert notification went out
- **A live link:** Allows the recipient to click to see the current state of the devices, elements, and/or items that have a problem
- **Information about the alert itself:** The name of the alert that was triggered, the polling engine or machine that collected the data, and so on

While it means more work during alert setup, having an alert with this kind of messaging means the recipients have several answers to "Why did I get this alert?" at their fingertips.

Automated Alert Actions

I have two questions I love asking when it comes to helping someone set up a new network monitor or alert. The first is "How do YOU know when something has gone wrong?" The second (and more important of the two when it comes to automation) is "Okay, then what?"

Maybe after the IT person gets an alert, she clears a queue, or restarts a service, or deletes all the files in a temp directory. Whatever the action is, it's very likely going to be something that could've been performed without human intervention. That's a quick win for automation — as long as it solves the problem. Not just once. Not just some of the time. Always. Automated alert responses must always solve the problem, or else you have another problem to solve.

As anyone who has been working in IT for more than 15 minutes should know, while many problems are tediously repetitive, sometimes you get the one weird case that resists all your usual tricks. This is why sophisticated monitoring solutions allow you to build an alert that triggers an initial action, then waits a specified amount of time. If the condition persists, a second (or third, or fourth, or whatever) action will be triggered.

Therefore, a disk-full alert could first clear the temp directory, wait 10 minutes, then remove specific application log files more than one month old, wait another 10 minutes, create a ticket for the server team, wait one hour, and, if the problem continues to persist, page the team lead as an escalation point.

But let's say there isn't a definitive action that can be taken. Maybe the answer is to check the last 15 lines of this log file, look at this other counter, and run a test query from the application server to the database. Based on the results of that information, the technician will know what to do next.

In that case, your automated action is to do all those steps and, after, insert those results into the alert message.

Then, instead of a message that says "Service XYZ is down," the technician receives a ticket that already contains greater insight into the conditions at the time of the failure, as opposed to 15 minutes later, when she's dragged her butt out of bed, fired up the laptop, and started to dig into the situation. By doing so, the monitoring system has effectively given staff 20 minutes of their life back. It has simplified the troubleshooting process.

And the best part of all of this is that even if the information you pull isn't 100 percent necessary, the reality is that the information is useful *most* of the time. Gathering that information proves the monitoring system can be leveraged as an always-on, Level One diagnostician.

Doing this kind of thing will make heroes out of you and the monitoring system.

The Elephant in the Room

As exciting as they are, I would suggest that, before you dig automated alert responses, you take a moment to understand a critical concept. In some circles, it's called the detection-prevention-analysis-response (or DPAR) cycle. While any conversation about monitoring will focus on detection (or monitoring) and response (automated alert actions), it would be negligent for you to ignore prevention and analysis.

Alerts are ways of catching errors when they occur (or at least, before they do too much damage). But then it's up to you as IT professionals to determine why they occurred and find a way to keep them from occurring again. When you make the effort to create an alert, you should also commit to doing the hard work of analyzing the situation going forward, looking for patterns and root causes to help ensure the issue is prevented in the future.

Automatic responses to alerts keep your business running and help ensure you get the beauty sleep you need. But in the morning, you and your team need to be able to see that something happened and do the hard work of figuring out why it happened, so you can prevent it from occurring in the future.

IN THIS CHAPTER

- Preparing for the five questions of monitoring
- Setting logical trigger and reset conditions
- Creating meaningful messages
- Adding Actionable Automation
- Ensuring the new monitor works in the real world

Chapter 5
Ten Tasks to Consider in Network Monitoring

I'm often asked (with varying degrees of exasperation, urgency, or enthusiasm), "What's it going to take to get monitoring up and running?" This might refer to a single monitor or to an entire solution. While everyone's environment has its own peculiarities, I can describe a set of tasks that you'll more likely than not have to consider. There are, in fact, ten of them — hence their appropriateness to the traditional Part of Tens, which appears in all *For Dummies* books.

Be Ready for the Five Questions of Monitoring

Before you accept your first monitoring request, you need to understand a universal truth about being "the monitoring expert." Inevitably, you'll find yourself answering the five questions of monitoring. These questions are ones that people never really had to ask when they were monitoring in silos, often under

the radar and on the side. But now that you're in charge of all monitoring, people are going to ask them constantly. They are

- **Why did I get an alert?** The person isn't asking, "Why did this alert trigger at this time?" He's asking why he got the alert at all. This tells you your alert messages don't have enough information. Go back and add more.
- **Why didn't I get an alert?** Something happened that the owner of the system felt should have triggered an alert, but he didn't receive one. This is your clue that you didn't ask enough questions when you set up the monitor and alert in the first place, and now it's triggering at an unexpected time.
- **What's being monitored on my system?** This person wants reports and data she can pull for her system, so she can look at trending, performance, and forensic information after a failure. This tells you that you need to beef up your reporting to show this information.
- **What will alert on my system?** The person asking this question wants to be able to predict the conditions when he'll get an alert for this system. How you address this depends on the capabilities of the systems monitoring solution you have. Some can output monitors, alerts, thresholds, and assignments with ease; others not so much. If you aren't lucky enough to have a tool that does, you'll need to spend a little extra time documenting as you put monitors and alerts in place, so you have this information at your fingertips when needed.
- **What do you monitor "standard"?** Asking what metrics and data are typically collected for systems is the inevitable (and logical) response when you say, "We put standard monitoring in place." You can't make a statement like that without being able to back it up with a handy list.

Fully Describe the Trigger Condition

When someone requests a monitor, make sure she's able to give you all the details. She should not only be able to identify the system(s) to be monitored, the sub-elements, and so on, but also

what "normal" operation for the monitored item should be, what the threshold for a warning and critical state might be, and more.

However, the truth is many folks don't have a clear sense of what the operating boundaries are, which is why I also recommend setting up a monitor in a test environment for a period of time to collect that data. Then you and the requestor review that data as a baseline and fill in the blanks.

REMEMBER

The same level of granularity is needed for alerts. Make sure you discuss exactly which factors contribute to a problem condition — which elements, the duration, the count of events, and so on.

Fully Describe the Reset Conditions

As with the trigger conditions (see the preceding section), take the time to understand what "all better" looks like. Often, it's more than simply when the problem goes away. Reducing monitoring and alerting noise means resetting an alert only when the problem has truly cleared up. Otherwise, you'll end up with *sawtoothing* — alerts that repeatedly trigger and reset.

To that end, your reset logic may have a longer delay than the trigger as well as a different threshold.

Alert Messages Should Be a Tell-All Story

After your network monitoring solution has gathered the data and determined there's a problem, it's no time to play coy. Giving all the information available could save valuable time for the person responding to the alert. That also means thinking creatively about not only what information you may regularly collect, but also what the monitoring tools could obtain at the time the problem is detected, and then include that in the alert message. An example would be pulling a list of the top ten processes (sorted by percent CPU utilization). You don't need that information all the time; you just need it when the CPU alert triggers.

Add Automation

If the response to an issue is repeatable (and most are), it can be automated. The value of automated responses can't be understated. It allows for immediate reaction to a problem condition, often clearing up an issue before a human has time to react, and at every moment of the day.


REMEMBER

The best automation is where the problem is completely resolved. But don't discount having an automation script that gets the technician halfway "home"— it's still time back in your pocket, and builds faith and reliance in the monitoring tool.

Have a Knowledge Base

Whether you're setting up a new monitor or an alert (or both), one of the things I insist on is a knowledge article of some kind to go with it. Why, you ask? Because while the monitoring solution is turned on 24/7, 365 days a year, the humans need a break. Just because Claude on first shift knows what to do when this alert comes in doesn't mean Gertrude on third shift will. Asking (nay, demanding!) Claude to document his process ensures that whoever is on call that day will know what to do.

Make It Happen on Purpose

Imagine you do everything I've discussed so far, and the alert never triggers. Then, a few weeks later, you get an angry call from the team manager, asking why your crappy monitoring isn't doing its job. You investigate, only to find that the error message in the original request has never been seen because the error message you were given doesn't exist.

IT professionals with the best of intentions can still get it wrong. Maybe the message in the documentation has changed after a recent patch. Maybe the request contained a spelling mistake, or the words were in a different order. Whatever the cause, this is why I also ask the requestor to make the problem happen on purpose. If he can't, then the monitoring is highly suspect in the first place.

REMEMBER

Some alerts can't be replicated. If you're checking to see if the data center is on fire, you certainly don't want to test your new monitor with a can of gasoline and a match. However, you *can* test the alert by temporarily changing the parameters to "*not* on fire."

Find One in the Wild

Even after testing the new monitor or alert by causing it to trigger on purpose, I still consider that new addition to be in the "testing" phase until I've seen an occurrence "in the wild." Very often, IT professionals develop a case of techno-paranoia, and ask to monitor errors that simply never come to pass.

Marking a monitor as "testing" until seeing one actually occur in production allows you to evaluate monitors regularly and decommission those that have never happened. This process frees resources for the events that are provably occurring (and therefore have a higher return on investment).

Describe the Cost of (Not) Monitoring

Knowing (and being able to articulate) the value that network monitoring brings to the table can make the difference in whether you have respect from the business or not. If you can't explain, with data, the business benefit that monitoring provides, then the business doesn't have a reason to continue putting time and resources into network monitoring.

Start with a baseline of the environment pre-monitoring. How often does a particular problem occur? How long does it take staff to discover the problem? How long does it take to resolve? What is the business impact to this problem?

TIP

This information can often be provided by the person/people requesting the new monitor in the first place.

What you ultimately want is a dollar value for the event. Something that says, "Every time X happened, we spent $123 in staff time to fix it and lost $456 in sales." Then observe the differences after the new monitor or alert (or both) is rolled out. Find the differential, expressed as dollars saved.

Now watch the number of times the alert triggers. You can now confidently report that monitoring represents a value to the business of $XX (the savings for that monitoring or alert times the number of occurrences) for that one event alone. Do that for all (or even most) of your monitors and alerts, and you'll soon have the executive team begging you to find new monitors to roll into production.

GOTO 1

At this point in the process, you may think the final task is to crack open a frosty beverage and bask in the fame and glory that network monitoring is raining down on you. While that's true, there's one more thing you should do: Set a calendar reminder to check back with the requestor in three to six months. The happy and sad truth is that a monitoring engineer's work is never done. Not only will you get more requests based on early successes, but also you need to go back and touch base with the folks you've created a monitor for in the past to make sure it's still performing as expected and desired.

Applications change; patches introduce new features, fix bugs, and even introduce new peculiarities. The person who requested the alert may not realize, until you remind her about it, that the monitor you created at the beginning of the year has been strangely silent since the last patch and should be revisited. Whatever the reason, make it part of your process to go back, check in, and offer to make improvements if possible.

solarwinds

Monitoring software that technology professionals trust.

Over 300,000 customers in 190 countries. From SMB to Fortune 500®

499 of the Fortune 500 are customers

#1 in Network Management*

55+ IT management products in the SolarWinds portfolio

Source: IDC defined Network Management Software functional market as measured by revenue, IDC's Worldwide Semiannual Software Tracker, October 2018.

***Highly commended Best SIEM Solution*

Monitoring is a discipline

See how you can become a disciple. IT's not glamorous, and few of you are getting rich off IT, but without IT, business would come to a screeching halt. Executive management visibility usually happens when things go wrong, so this book helps you steer clear of trouble and provides a primer on network monitoring to help you understand just why proactive monitoring is critical to the success of your business.

Inside...

- The benefits of monitoring
- The difference between monitoring and managing
- Monitoring tools that work for you
- Information on the IT stack
- Best practices of network monitoring

solarwinds

Leon Adato is a Head Geek™ and technical evangelist at SolarWinds and is a Cisco® Certified Network Associate (CCNA), MCSE, and SolarWinds Certified Professional. His 25 years of network management experience spans financial, healthcare, food and beverage, and other industries.

Go to Dummies.com®
for videos, step-by-step photos, how-to articles, or to shop!

ISBN: 978-1-119-60303-0
Not for resale

for dummies®
A Wiley Brand

WILEY END USER LICENSE AGREEMENT

Go to www.wiley.com/go/eula to access Wiley’s ebook EULA.