SECURITY · PERFORMANCE · SEO · UPTIME

When anyone can hack, you can no longer afford to work reactively.

Why watching the front door is no longer enough, and how to bring your clients' entire digital house into one view: from a visitor's first click to the system it connects to.

LAYERS COVERED	CHECK LOCATIONS	SHORTEST INTERVAL	EDITION
4	NL DE US UK	30s	2026.1

SUMMARY

The gap is not in your firewall. It is between your tools.

A serious attack no longer requires specialist knowledge. Ask an AI model the right question and you get a step-by-step plan and working code. The brake is off: the number of parties capable of doing damage is no longer something you can survey.

For a managed service provider that changes the arithmetic, because you are judged on something broader than security alone. A hacked webshop, a server that hangs, a page that takes eight seconds, a search ranking that disappears: to your client it is the same story. Their business is at a standstill, and you should have seen it coming.

Most providers measure something. Almost nobody measures the whole. Uptime sits in one tool, security in a scan that runs quarterly, performance in a report nobody opens, SEO at the marketing agency. Four pictures, four truths, and not one signal that runs across them. This whitepaper describes the alternative: a single model in which those four share one timeline, what belongs in each layer, and how to have it running within ninety days.

THE ARGUMENT IN FIVE LINES

- The barrier to attack is gone. AI supplies the knowledge an attacker used to need themselves.
- Your client draws no distinction between a hack, an outage and slowness. All three cost you trust.
- Separate tools give four pictures and no coherence. An incident rarely starts exactly where you happen to be looking.
- Working proactively is not an extra service. It is the condition for meeting your SLA at all.
- Evidence is part of the job. NIS2, ISO 27001 and your own quarterly report ask for demonstrable control, not a gut feeling.

CONTENTS

CHAPTER 01

Anyone can carry out a serious cyberattack today.

All it takes is asking any AI model what to do, and it hands over everything step by step: the right instructions and ready-made code. Knowledge is no longer a barrier.

Until recently, knowledge was the scarce ingredient in an attack. Someone had to know what an injection is, where an outdated plugin gives itself away, what a payload looks like that slips past a filter. That took years to acquire. It filtered by itself: most attempts came from automated bots knocking on doors they already knew.

That filter is gone. The attacker no longer has to understand what they are doing, they only have to ask. The effect is twofold: more attempts are made, and the average attempt is better than it used to be. Anyone with an internet connection is suddenly dangerous.

> **Knowledge was the brake on the number of attacks. That brake has been removed, and nobody put it back.**
>
> THE SHIFT THAT DEFINES 2025 AND 2026

Meanwhile the surface they can knock on keeps growing. A typical client now runs a CMS with fifteen plugins, a link to an accounting package, mail that passes through three parties, certificates quietly expiring, and DNS at a registrar someone once made an account for. Every one of those parts can be the weak spot, and not one of them announces itself.

01

Knowledge became free

What used to be a speciality is now a prompt. The attacker's skill says nothing about the quality of the attack any more.

02

The surface grows

Plugins, integrations, subdomains, certificates, DNS and mail. Every part is a way in, and forgotten subdomains are the most popular one.

03

The window shrinks

Hours often separate a published vulnerability from its first exploitation. A quarterly patch round stops being a policy and becomes a bet.

WHY THIS HITS YOU BEFORE IT HITS YOUR CLIENTS

A managed service provider is by definition a more attractive target than the client is. You hold one set of credentials to dozens of environments. That means two things: your own hygiene counts for more than your clients', and you have to be able to prove it is in order. Chapter 06 is about that proof.

CHAPTER 02

When it goes wrong, the client looks at you. Not at the attacker.

Your clients do not care whether it is a hack, a slow server, a lost search ranking or a system that hangs. They want one thing: that their webshop, portal and systems keep running, stay fast and stay findable.

That is not an unreasonable expectation, it is exactly what they pay for. The awkward part is that the client cannot tell the four causes below apart, while you have to look them up in four different tools.

WHAT THE CLIENT REPORTS	WHAT IS ACTUALLY GOING ON	WHAT IT COSTS
"The site is down"	An expired certificate, a changed DNS record, a container that fell over	Revenue per minute, plus an evening of recovery
"It is so slow"	A heavy query, an uncompressed image, a third-party script	Conversion and ranking, slowly and invisibly
"We have dropped out of Google"	A robots rule that hitched a ride on a release, a canonical that vanished, backlinks that fell away	Months to climb back
"Our mail is not arriving"	SPF, DKIM or DMARC not in order, domain being spoofed	Deliverability and reputation
"We have been hacked"	A plugin three months behind	Recovery, disclosure duties, trust

> **An angry phone call is not an alert. It is an invoice you can no longer prevent.**

The moment your client's business stops and you hear about it through that call, you are late. Not slightly late: at that point you start three things at once. Working out what is wrong, fixing it, and explaining why you did not know sooner. The third costs the most, because you always lose that one.

It is also where the business model creaks. An SLA with a four-hour response time is worth nothing when the starting gun is a client calling. The clock had already been running for an hour. Working proactively does not sell a faster response, it sells a smaller incident, and that is the only difference the client really notices.

CHAPTER 03

The problem is not that you measure nothing. It is that nobody sees the whole.

Nearly every provider has an uptime pinger, a security scan somewhere, a performance report from a browser tool and an SEO overview supplied by the client's agency. Every tool is right about its own slice. That is exactly the problem, because an outage does not respect that division.

Take an expiring certificate. The uptime tool checks whether the server answers, and it answers perfectly, so the dashboard stays green. Browsers meanwhile refuse the connection, conversion stalls, and search engines start logging errors. Weeks later something shows up in the SEO report. At no point was anything red that anyone actually read.

SYMPTOM 01

Blind spots between tools

Each tool watches its own layer. Whatever falls between two layers is watched by nobody. That is precisely where incidents start.

SYMPTOM 02

Alerts without an owner

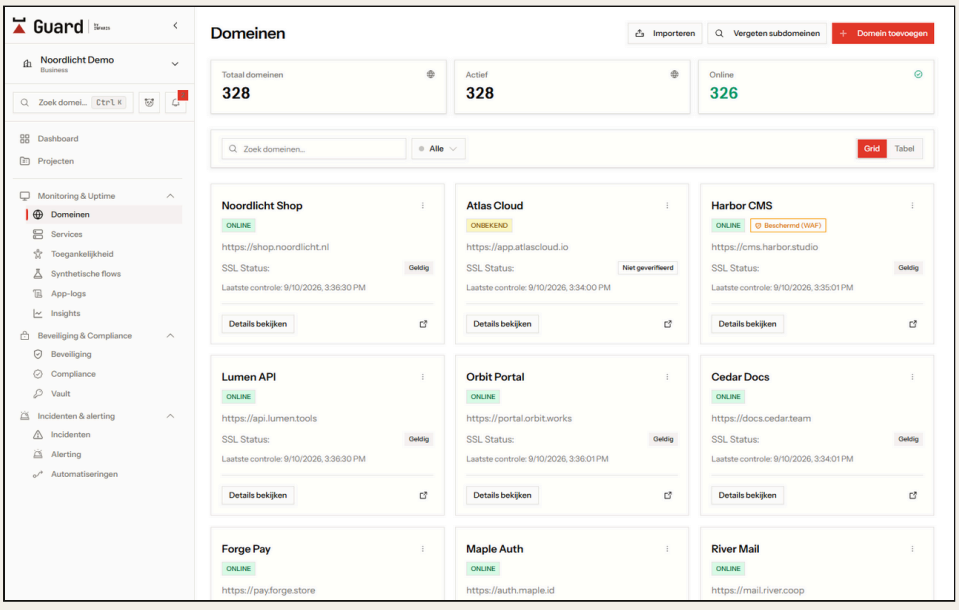
Notifications land in four places, two of which are only read by a mailbox nobody owns. Without a route, an alert is the same as no alert.

SYMPTOM 03

No story afterwards

When the client asks how last quarter went, you glue four exports together. That is work, and it convinces nobody.

There is a fourth effect, and it is the most expensive. When nobody sees the whole, every alert becomes a separate event instead of a pattern. The third time the same client is slow, you treat it like the first.



DOMAINS EVERY CLIENT, EVERY DOMAIN, ONE STATUS

CHAPTER 04

We do not just check whether the door is locked. We check the whole house.

Security, performance, SEO and uptime, from click to integration. The moment something starts to falter, your team knows first and sees straight away where it sits.

The model below has four layers. They are not sold separately, because that is exactly the mistake chapter 03 describes. They share one timeline, one client structure and one alert route. The next four pages walk through them one by one.

LAYER 01

Uptime and stack

Is it running, and what is it running on? Probes from four countries, SSL and DNS, the machines underneath and what is installed on them.

LAYER 02

From click to integration

Does it actually work, for a real visitor? Synthetic flows, Core Web Vitals from real traffic, accessibility and findability.

LAYER 03

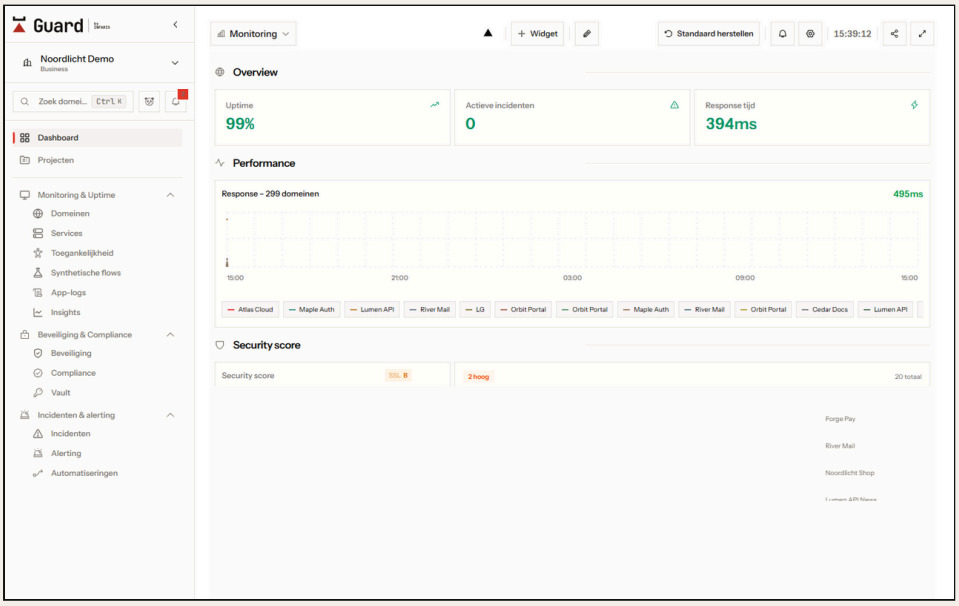
The door and the lock

Is it secure, and can you prove it? Security score, CVEs against your own stack, mail security, secrets and compliance.

LAYER 04

Before the client

Who knows first, and what happens then? Incidents, alerting, automation, status page and reporting.



DASHBOARD

FOUR LAYERS, ONE VIEW

LAYER 01

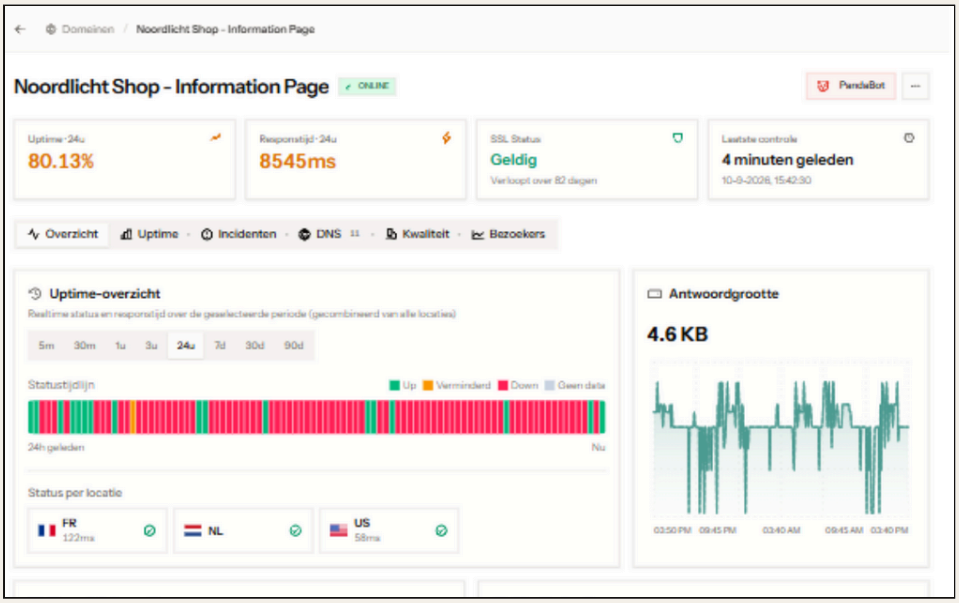
Is it running, and what is it running on?

The base layer answers the question your client asks when they call, but before they call. Checks run every 30 or 60 seconds from the Netherlands, Germany, the United States and the United Kingdom, so an outage at one provider does not present itself as an outage at the client.

- **Uptime and response.** Multiple locations, multiple protocols, with a response time you can follow over time rather than a green or red dot.
- **SSL and DNS.** The full certificate chain, the expiry date well in advance, and an alert the moment a DNS record changes. That last one is more often an attack than a coincidence.
- **Services and topology.** Which machines exist, what runs on them, how they depend on each other, and which packages are behind. Patch management as a list, not an intention.
- **The agent on the machine itself.** A lightweight agent reports disk, memory, processes and updates from the inside. Looking from outside does not show you the disk will be full in two days.
- **Projects per client.** Everything grouped by client or environment, so whoever is on call does not have to guess which domain belongs to whom.

WHERE IT USUALLY GOES WRONG

Forgotten subdomains. An old staging environment, a campaign site from 2021, a test domain nobody cleaned up. They are exposed, they run outdated software, and they appear on no list at all. Start your inventory there, not at the production site you already watch.



DOMAIN DETAIL UPTIME, RESPONSE AND STATUS PER LOCATION

LAYER 02

Does it actually work, for a real visitor?

A server that answers is not yet a working webshop. This layer tests what the visitor actually does and measures what they actually experience, including the part that happens on their own device.

REPLAYED

Synthetic flows run in a real Chromium and click through the path that earns money: sign in, search, add to cart, check out. If step four breaks, you know at three in the morning and not at nine.

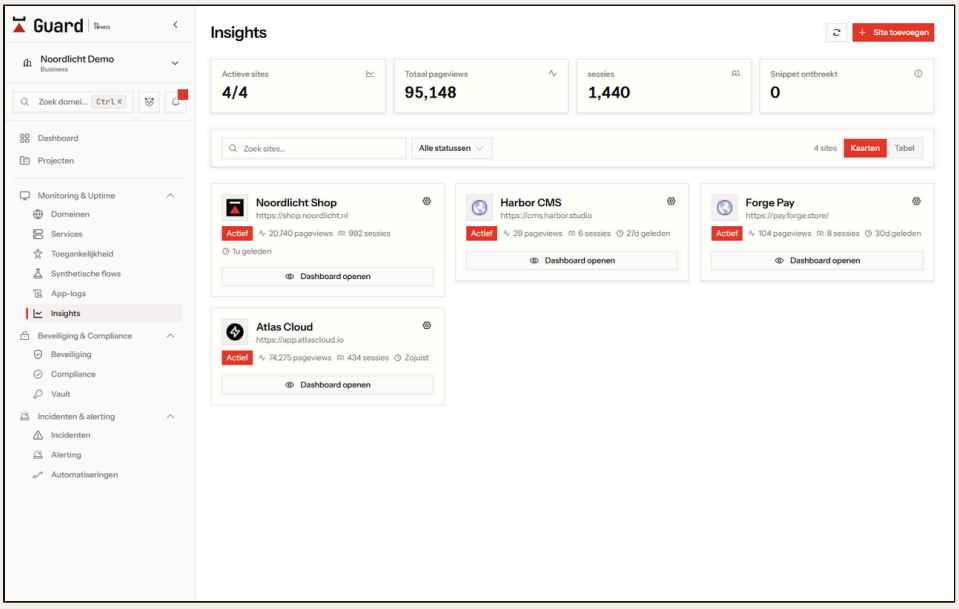
Change detection compares pages with how they looked yesterday. A price that shifts, a form that disappears or a script that appears stands out immediately.

ACTUALLY MEASURED

Insights collects Core Web Vitals from real visitors: how fast the page loaded on their device, on their connection. That is the figure search engines act on, not the score of a test machine on fibre.

App logs catch the errors the visitor never reports but does run into, and tie them to the moment they started.

- **Accessibility.** WCAG checks per page, with a score that measures how accessible the page is rather than how complex it is. Relevant now that the European accessibility directive reaches commercial sites too.
- **SEO and backlinks.** Findability in the same loop as the rest: indexability, structure, and the links pointing at the client. A lost ranking is an outage, just a slow one.
- **The browser extension.** Scans whichever tab is open for cookies, headers, mixed content, accessibility, SEO, privacy and performance. Useful during an intake or a sales call, with nothing to set up first.



INSIGHTS REAL VISITORS, REAL LOAD TIMES

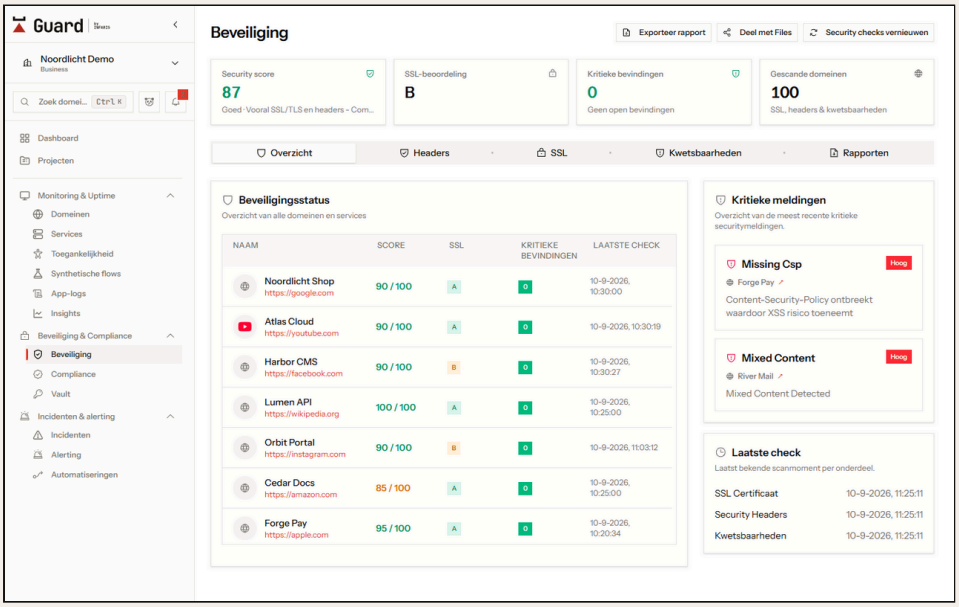
LAYER 03

Is it secure, and can you prove it?

In this setup security is not a separate product but a layer across the same domains, machines and integrations you already watch. That is the difference between a scan that produces a PDF every quarter and a score that moves with whatever you deployed today.

- **Security score per domain.** Headers, cookies, encryption, mixed content and exposed paths, summarised in one figure with the findings underneath. Usable towards the client, detailed enough for your engineer.
- **CVEs against your own stack.** Vulnerabilities are matched against what is actually running, CMS and plugins included, so you do not get a thousand advisories of which three apply.
- **Mail security.** SPF, DKIM and DMARC, including the quiet cases where the record exists but enforces nothing.
- **Vault.** Passwords, keys and tokens stored with the product they belong to, rather than in a spreadsheet or a chat thread from 2023.
- **Patch management.** One list of what is behind, for how long, and how serious that is. That is the list an audit asks for.

> A security report from last quarter describes an environment you have deployed twelve times since.



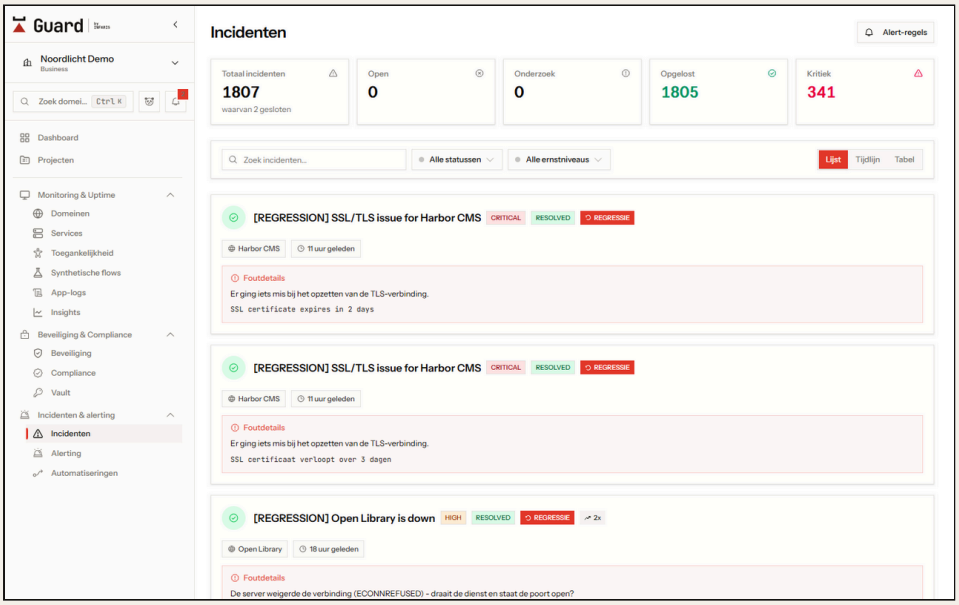
SECURITY SCORE, FINDINGS AND HISTORY PER DOMAIN

LAYER 04

Who knows first, and what happens then?

Measuring only matters if someone acts on it. This layer turns a measurement into an intervention, and an intervention into a story you can retell.

- **Incidents on one timeline.** Signals from every layer come together in a single incident, with the events around it. A deploy, a DNS change and a spike in errors then sit under each other instead of in three systems.
- **Alerting where your team already is.** Slack, Teams, Discord, Telegram, email, push or your own webhook, with rules per client, per severity and per time window.
- **Automation.** Fixed responses to recurring signals: a restart, a ticket, a message to the client. Whatever you have done by hand three times should be a rule.
- **Public status page and Status Hub.** If something does happen, a clear status page shows you are in control. That saves calls, and it saves explaining.
- **Reporting and SLA.** Per client, per period, automatically. No more gluing exports together the week before the quarterly review.
- **MCP, CLI and API.** Everything the dashboard shows is also available to your own scripts, your own portal or an AI assistant your team uses.



INCIDENTS EVERY SIGNAL, WITH ITS CONTEXT

CHAPTER 05

Five levels. Most providers sit at two.

Introducing monitoring is not one step but five, and the gain is not in the first. Use the model below to work out where you stand, per client rather than overall. The spread inside your own portfolio is usually wider than you think.

LEVEL	HOW TO RECOGNISE IT	WHAT THE CLIENT NOTICES
00 BLIND	No structural checks. Problems arrive through the service desk.	They are always the first to know.
01 PINGING	Uptime is watched, nothing else. Alerts land in a mailbox.	Hard outages get noticed, the rest does not.
02 COLLECTED	All four layers are measured, but in separate tools and without a shared timeline.	You know more, but still react after the fact.
03 CONNECTED	One timeline, one alert route, clear owners and agreed response times per severity.	Outages become shorter and rarer.
04 PREDICTIVE	Trends and thresholds lead to action before something breaks. Recurring responses are automated.	Mostly they notice they rarely have to call you.

The three agreements that make the difference

The jump from level 02 to 03 is not a purchase but an agreement. Three of them, and they fit on one page.

AGREEMENT 01

Every alert has an owner

A signal with no name behind it is noise. Record per severity who looks, within how long, and who takes over outside office hours.

AGREEMENT 02

Thresholds per client, not per tool

What is critical for a webshop is noise for a brochure site. Set thresholds per client, or your team will learn to dismiss alerts.

AGREEMENT 03

What happened three times becomes a rule

Any action your team performs manually for the third time is a candidate for automation. Keep that list explicitly.

CHAPTER 06

Control you cannot demonstrate does not count.

At some point somebody asks for evidence. A client passing their own NIS2 obligation down to their suppliers. An auditor preparing an ISO 27001 certification. An insurer. Or simply a prospect who wants to know how you work.

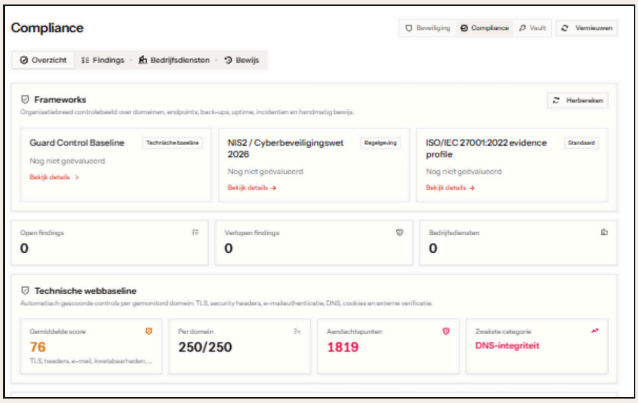
At that moment a tool that only raises alarms is not enough. You need history, and that history has to come from somewhere nobody can adjust after the fact.

WHAT GETS ASKED

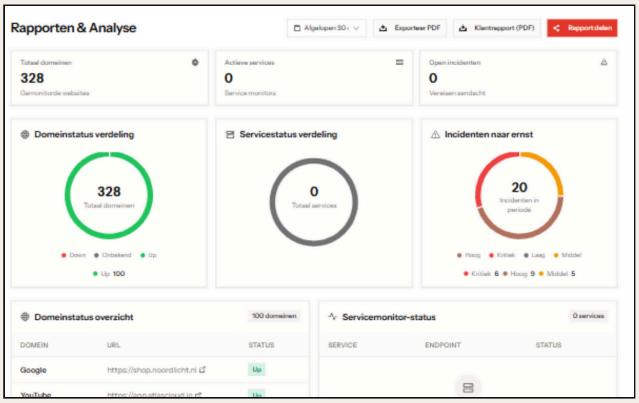
- Which systems you manage, and since when.
- Which vulnerabilities were known, and when they were closed.
- How many incidents there were, how long they lasted and what caused them.
- Whether the agreed availability was actually met.

WHAT IT GIVES YOU WHEN YOU HAVE IT

- A compliance overview that places NIS2, ISO 27001 and your own baseline next to your real measurements.
- Reports per client and per period, automatically, in your own house style.
- An SLA overview showing whether the target was met, without calculating first.
- A public status page that has the conversation you would otherwise have fifteen times.



COMPLIANCE STANDARD NEXT TO MEASUREMENT



REPORTING PER CLIENT, PER PERIOD

IN PRACTICE

Start with the quarterly report, not the certification. Once your clients get an overview every quarter that holds up, you already have the history an auditor will ask for later. The other way round rarely works.

CHAPTER 07

Not everything at once. But in this order.

The most common mistake is switching everything on for every client at once. Week one then brings a stream of alerts. Your team learns to dismiss many of them correctly, but soon starts dismissing alerts that deserve attention. Build it up in three blocks of thirty days.

DAY 1 TO 30 · INVENTORY AND VISIBILITY

- Map every domain and subdomain, forgotten ones included. Group them per client.
- Switch on uptime, SSL and DNS for everything. No outbound alerting yet, only measurement.
- Run one full security scan across the portfolio and use the result as your baseline.
- Pick the three clients where the risk is highest. They go to the next phase first.

DAY 31 TO 60 · DEPTH AND ROUTING

- Build synthetic flows for the path that earns money at each client. Start with checkout and sign-in.
- Switch on Insights for the most important sites, so you build real visitor data instead of test scores.
- Set up the alert route: a channel, an owner and a response time per severity. Write it down.
- Tune thresholds per client. Expect two rounds of adjustment, that is normal.

DAY 61 TO 90 · EVIDENCE AND AUTOMATION

- Prepare the first quarterly report and discuss it with three clients. Their questions tell you what is still missing.
- Publish a status page for the clients with the most end users.
- Automate the three actions your team performed by hand most often in this period.
- Put the day 1 baseline next to the day 90 position. That is your sales story, ready made.

ONE PITFALL TO AVOID

Do not switch on alerting until the thresholds are right. A team that learns in week one that alerts usually mean nothing will still be dismissing them a year later. Trust in the alert is the product, not the alert itself.

AFTER DAY 30

A complete inventory and a baseline. At most clients that conversation alone is a sales conversation.

AFTER DAY 60

Alerts that hold up, reaching the right person within an agreed time. This is the jump to level 03.

AFTER DAY 90

A report you are willing to send, and a demonstrable difference against the day 1 baseline.

CHAPTER 08

Sixteen questions about your own estate.

Run through them for one client, not for your whole portfolio. If more than four boxes stay empty, that client sits at level 02 or lower from chapter 05.

- [ ] I have a current list of every domain and subdomain for this client, old campaign and test environments included.
- [ ] I know when a certificate expires in thirty days, not when it has expired.
- [ ] I get an alert when a DNS record changes, including when I made the change myself.
- [ ] I measure from more than one location, so a network problem does not look like an outage.
- [ ] I know which software runs on the machines and what of it is behind.
- [ ] The path that earns money at this client is walked automatically, not just the homepage requested.
- [ ] I measure load times from real visitors, not only from a test machine.
- [ ] I see it when the site drops out of the index or when important backlinks fall away.
- [ ] SPF, DKIM and DMARC are correct and actually enforce something.
- [ ] Vulnerabilities are matched against what runs at this specific client.
- [ ] Credentials live in a vault, not in a spreadsheet or a chat thread.
- [ ] Every alert has an owner and an agreed response time, outside office hours too.
- [ ] Alerts arrive where my team already works, not in an unread mailbox.
- [ ] During an outage the client can see what is going on without calling.
- [ ] I can show within five minutes whether the agreed availability was met last quarter.
- [ ] I can prove when a known vulnerability was closed, with a date and evidence.

0 TO 2 EMPTY

This client sits at level 03 or higher. Aim for the step to predictive work: trends, thresholds and automation.

3 TO 6 EMPTY

You measure plenty, but the picture falls apart. The gain is in one timeline and an alert route with owners, not another tool.

7 OR MORE EMPTY

Start with the inventory from chapter 07. There are almost certainly exposed domains that appear on no list at all.

IN CLOSING

Waiting for the phone to ring is playing with fire.

Guard was built to run this model rather than only describe it. Security, performance, SEO and uptime run across the same domains, the same client structure and the same timeline, with one alert route underneath and reporting on top. What you read in this whitepaper is how the product works.

LAYERS 01 AND 02

Measure

Uptime, SSL, DNS, services and agent. Synthetic flows, Core Web Vitals, app logs, WCAG, change detection, SEO and backlinks.

LAYER 03

Secure

Security score, CVE matching against your stack, SPF, DKIM and DMARC, Vault, patch management and compliance against NIS2, ISO 27001 or your own baseline.

LAYER 04

Respond

Incidents on one timeline, alerting to Slack, Teams, Discord, Telegram, email, push or webhook, automation, status page, reporting, SLA and MCP, CLI and API.

SEE FOR YOURSELF

- Try the browser extension on a client's site. It needs no setup and gives you a picture within a minute.
- Onboard one client fully, forgotten subdomains included. The first inventory nearly always turns up a surprise.
- Ask for a demo in which we measure your own domains instead of our demo environment.

Guard by INFARIS

guard.infaris.com

REQUEST A DEMO: INFARIS.COM/EN/DEMO
QUESTIONS: INFARIS.COM/EN/CONTACT

GUARD BY INFARIS · WHITEPAPER, EDITION 2026.1 · PUBLISHED BY INFARIS, THE NETHERLANDS
MONITOR EVERYTHING. MISS NOTHING. · SCREENSHOTS COME FROM A DEMO ENVIRONMENT WITH FICTIONAL CLIENT NAMES.