ON2IT
ZERO TRUST INNOVATORS

WHITEPAPER

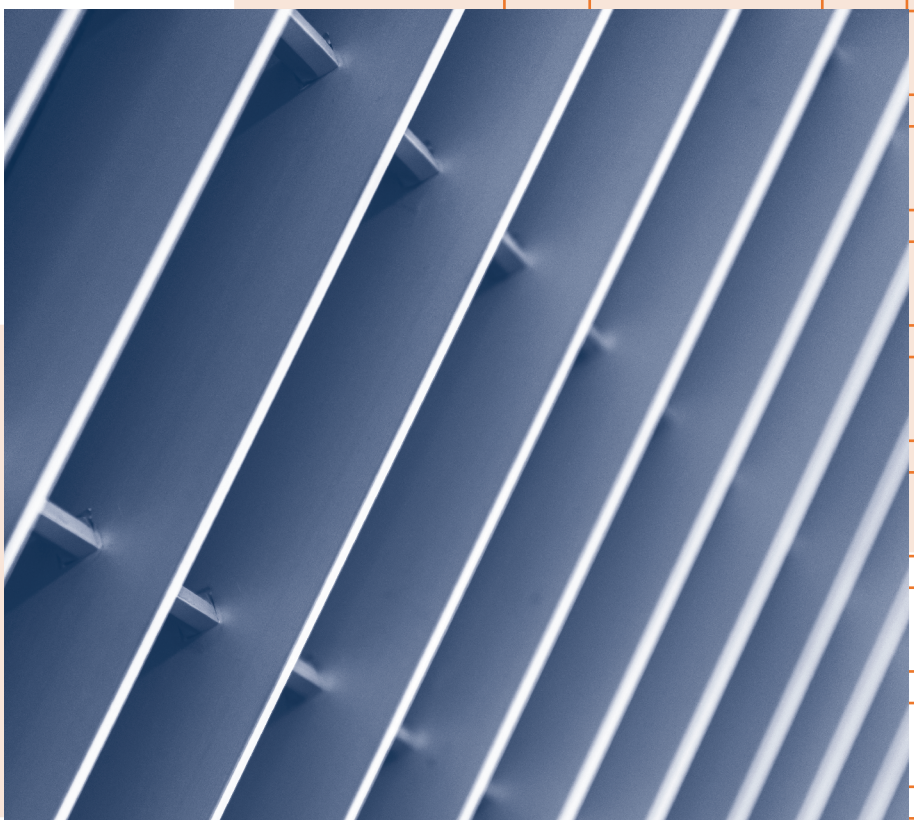
HOW A ZERO TRUST STRATEGY PREVENTS DATA BREACHES USING UNKNOWN THREATS

LIEUWE JAN KONING

CTO ON2IT

ROB MAAS

Lead Architect and Zero Trust Principal ON2IT



ZERO TRUST
CONTINIOUS COMPLIANCE
DAAS
CMDB
IT-SECURITY
MICROSEGMENTATIE
SAAS
SHADOW-IT
GDPR
ZERO TRUST RISK MAP

HOW A ZERO TRUST STRATEGY PREVENTS DATA BREACHES USING UNKNOWN THREATS

The U.S. Department of Commerce's National Institute of Standards and Technology's *National Vulnerability Database* hit a record high of reported vulnerabilities (CVEs) in 2021.

The new record, the fifth straight year the record has been broken, came to 18,378 vulnerabilities reported. More than 3500 of those were high-severity vulnerabilities.

Researchers at Redscan Cyber Security Ltd. analyzed the 2021 data and found that of those reported, 90% can be exploited by attackers with limited technical skill, while 61% of CVEs require no user interaction such as clicking a link, downloading a file or sharing credentials.

In virtually all instances, once a CVE is published, security researchers or the vendor(s) whose product is involved offer guidance for mitigating measures. These include software patches, software settings, detection of suspect traffic behaviors or other preventive measures specific to a vulnerability.

KNOWN OR UNKNOWN: IT'S NOT THAT SIMPLE

Publication of a vulnerability through the CVE process makes it a 'known threat'. You can reasonably expect your cybersecurity technology vendors to use the associated indicators-of-compromise (or IOC's) in their products, to offer a basic protection against the now known vulnerability and all known exploit techniques. Obviously, when it comes to applying patches and many other countermeasures, most of the burden is on your SOC, dev-ops or system/network administrators.

The term 'known vulnerability' suggests that things are under control, but for all practical purposes, for you it still is an unknown threat in your

> **"YOU CAN REASONABLY EXPECT YOUR CYBERSECURITY TECHNOLOGY VENDORS TO USE THE ASSOCIATED INDICATORS-OF-COMPROMISE (OR IOC'S) IN THEIR PRODUCTS, TO OFFER A BASIC PROTECTION AGAINST THE NOW KNOWN VULNERABILITY AND ALL KNOWN EXPLOIT TECHNIQUES."**

infrastructure until you have applied patches and/or implemented all advised countermeasures and mitigated the threat.

Even when you act immediately after the publication of a new CVE, you cannot always be sure that the vulnerability has not been exploited in your environment. In many high-profile cases, researchers noticed active exploitation of a vulnerability in the weeks or even months before the CVE publication.

For known vulnerabilities the dwell time, the time between the initial penetration/compromise of an organization's environment and the point in time that it's discovered, averages more than 24 days, Mandiant researchers claim in their 2021 M-Trends threat report.

We can only guess how much longer the dwell time is when you have to detect and chase vulnerabilities that are not yet seen on the radar.

Another point to consider: even when vulnerabilities become CVEs, we frequently see a rapid development of new exploit techniques not addressed in the initial remediation measures.

THE UNKNOWN UNKNOWNS

And then there are the really unknown unknowns. We are literally in the dark when it comes to the number and the impact of the unknown threats being exploited every day to gain access to your most valuable data and assets.

For nation states and highly organized international criminal organizations, unknown vulnerabilities and effective exploits are most valuable assets. They can use them for their own purposes, or sell the techniques on the dark web. Either way, they have zero motive to publish their treasures through the CVE process.

Given the sheer number of CVE's published, and the potentially devastating consequences of targeted attacks using new and unknown threats, it only seems obvious that you need an updated strategy for cybersecurity than individually chasing a never-ending stream of bugs, vulnerabilities and exploits.

> **"EVEN WHEN YOU ACT IMMEDIATELY AFTER THE PUBLICATION OF A NEW CVE, YOU CANNOT ALWAYS BE SURE THAT THE VULNERABILITY HAS NOT BEEN EXPLOITED IN YOUR ENVIRONMENT."**

ZERO TRUST TAKES A DIFFERENT APPROACH

The Zero Trust strategy is based on architecting so-called protect surfaces around your most valuable data, assets, applications and services.

Older cybersecurity approaches try to protect the gigantic attack surface of the IT-infrastructure as a whole. Zero Trust's focus on many smaller protect surfaces reduces the overall attack surface in orders of magnitude to many tiny and easily known logical objects.

For these protect surfaces, we can explicitly specify what kind access is permitted (known), thereby eliminating many of the unknown threats targeted at the infrastructure as a whole.

So, rather than implementing specific mitigations for vulnerabilities, Zero Trust is fundamentally based on achieving the optimal and most general prevention of data breaches, designing protection from the inside out.

Zero Trust achieves its mission by building on three concepts that have remained unchanged since John Kindervag coined the term Zero Trust in his seminal 2010 paper No More Chewy Centers: Introducing The Zero Trust Model Of Information Security.

Kindervag flipped the mantra "trust but verify" into "Never trust, always verify." His basic concepts have remained the cornerstone of all authentic Zero Trust architectures developed in the last decade.

It is noteworthy that Forrester, trying to take back some of the intellectual ownership of Zero Trust (John Kindervag worked at Forrester while developing his ideas) recently published a new definition of modern Zero Trust that is still closely aligned with the three principles developed more than a decade ago.

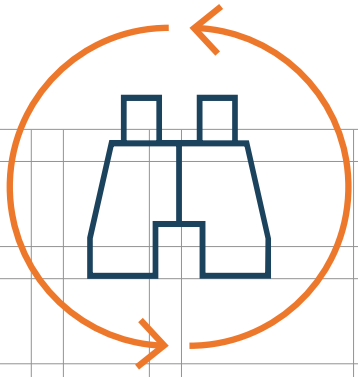
THREE SIMPLE CONCEPTS

When you eliminate the concept of trust (hence Zero Trust) from the network, Kindervag says, it becomes natural to ensure that all data, applications, assets and services (DAAS) are securely accessed — no matter who creates the traffic or where it originates from.

In the Zero Trust Model, security professionals must assume that all traffic is threat traffic until it is verified that the traffic is authorized, inspected, and secured. In Kindervags current terminology: start with the protect surfaces that need protection, and design outward from there.

The second key concept in Zero Trust is access control. Determine who needs to have access to a resource to get their job done. It is common to give too many users too much access to sensitive data for no business reason, but the Least Privilege principle should be your aim.

In Zero Trust someone will assert their identity and then we will allow them access to a particular resource based upon that assertion. We will restrict users only to the resources they need to



"INSTEAD OF TRUSTING USERS TO DO THE RIGHT THING, WE VERIFY THAT THEY ARE DOING THE RIGHT THING. "

perform their job. To achieve that goal, Kindervag promotes the use of access policies using the so-called Kipling method.

Using natural language rules based on Who, What, When, Where, Why, the method allows easily created, easily understood, and easily auditable Zero Trust policy statements for various technologies.

A number of technology vendors seem to equate Zero Trust with identity. While identity is a key element in Zero Trust, the use of the different identity and authorization technologies is subject to the requirements in the definition phases of the protect surfaces, and not the other way around.

But Zero Trust does not stop there. Instead of trusting users to do the right thing, we verify that they are doing the right thing. All traffic going to and from a protect surface must be logged and inspected for malicious content and unauthorized activity up through Layer 7.

Many security professionals do log internal network traffic, but that approach is passive and motivated by compliance requirements for retention of data.

In Zero Trust, logging and inspection are the foundation for proactive and real-time protection capabilities and safeguarding the correct deployment of protect surface policies.

HOW DOES THAT STOP UNKNOWN THREATS?

Eliminate trust and start with protect surfaces, determine who or what needs access and log and inspect all traffic.

These three concepts have evolved into workable reference architectures and a great many security products supporting specific functions required for Zero Trust implementations, such as identity and device management, encryption, monitoring and inspection.

Following the adoption of the next-generation firewall as the initial de facto device to implement the Zero Trust concept of microsegments and their associated microperimeter, many vendors followed with products enabling efficient segmentation of cloud, network and datacenter infrastructures. The concepts of protect surface and microsegment also the concept of a tenant in a SAAS environment

So, in 2022, for many security professionals it seems obvious and intuitive that the application of the principles of Zero Trust leads to a better prevention against known and the still unknown threats.

The unconditional inclusion of Zero Trust in president Biden's 2021 Executive Order on Improving the Nation's Cybersecurity cemented the status of Kindervag's strategy as a preventative approach to cybersecurity. Still, for many practitioners, it is not obvious what the actual ZT mechanisms are that prevent a data breach.

LET'S TAKE LOG4J

It is instructive to refer to the recent and well-publicized Log4J vulnerability to get a better grip on how a Zero Trust approach deals with prevention if Log4J was an unknown vulnerability without a CVE or mitigation.

The interesting thing about the Log4J exploit is that the initial stage of the attack chain could not have been prevented by a strict identity or access policy. Most of the applications using the Log4J Java component for logging server activity were by their very nature publicly accessible without the need for strict access control.

A good example is the Apple server that is used to change the name of your iPhone. Also 2FA or even stronger forms of authentication would not block the initial attack. In MITRE ATT&CK terms: the exploit starts with a public facing application, before the next part of the adversary behavior is enabled.

The fundamental weakness in Log4J CVE is that the inclusion of a specific string in the logfile (generated by a seemingly harmless request) is the entry-point to the exploit. We are talking about the "message substitution" feature—which allowed for programmatic modification of event logs by inserting strings that call for external content.

The code that supported this feature allowed for "lookups" using the Java Naming and Directory Interface (JNDI) URLs. This feature inadvertently made it possible for an attacker to insert text with embedded malicious JNDI URLs into requests to software using Log4j—URLs that resulted in remote code being loaded and executed by the logger.

From the perspective of the public facing application, the insertion of a text-string in the logfile by itself is not a behavior that warrants inspection, but is behavior by design.

It's interesting to note that the exploitable features of Log4J are not even required in most environments. The Zero Trust paradigm of least privilege theoretically would have forced this feature to be disabled if not required (which is true in the majority of the use cases).

"IT'S INTERESTING TO NOTE THAT THE EXPLOITABLE FEATURES OF LOG4J ARE NOT EVEN REQUIRED IN MOST ENVIRONMENTS."

The problem here is that many users are not even aware that Log4J is a component under the hood of an application they are using, let alone how Log4J was actually configured within that application.

So, although Zero Trust theoretically would have been able to offer serious prevention against Log4J exploits by disabling features that are not needed, the practical implementation of this concept is difficult in the case of Log4J. You can't disable what you don't see.

There is a case to be made here that this is a bad cybersecurity practice, you should have an up-to-date SBOM (software bill of materials) and know which components are used in the software you use. Depending on your vendors, this is an ideal situation which may not be practically feasible.

WHY IS THAT TYPE OF TRAFFIC ALLOWED IN THE FIRST PLACE?

So far, we have seen that in the initial stages of an attack chain, especially when we deal with public facing services, the fundamental principles of Zero Trust do not always automatically offer prevention against attacks.

But when we follow the exploit trail of a Log4J (and a large number of similar exploits), the rigid application of Zero Trust principles offer a firm line of defense against further infiltration and resulting damage. This is most obvious when a compromised server makes a request to an outside server to download and execute malicious software.

In the case of Log4J, the exploit depends on protocols that should be disallowed unless there is a strict policy in place. In such scenario's, Zero Trust - when properly implemented - really shows its preventive muscles against unknown exploits.

One of the most overheard remarks in the mitigation of Log4J probably was: "why is this type of malicious traffic to outside servers (regardless of if we already know they are malicious) allowed in the first place?

By default we should block this traffic, unless we have a clear policy allowing these outgoing connections. In Kipling-terms, you should have been able to answer the questions:

- What is the destination of the traffic?
- Why is access required
- How may access be obtained, and with which applications?

By doing so, we could have stopped Log4J and similar attacks from fetching content that serves as a stepping stone for further malicious activity once executed.

This approach to prevention is obviously more effective and general than trying to tailor prevention to a specific case. Initial mitigations for Log4J, for instance, checked for suspect strings that were injected, but hackers rapidly became very creative in finding new circumventing techniques for inserting exploitable strings in the log files.

This illustrates the difficulty of trying to fight every new technique and variation hackers develop.

PROTECT SURFACES AND MICROPERIMETERS

The Zero Trust concept of the protect surface is another extremely effective general prevention technique. Dividing a monolithic infrastructure of data, applications, assets and devices into smaller protect surfaces guarantees that the most valuable

> **"THE CONCEPT OF PROTECT SURFACE OFFERS SEVERAL LAYERS OF GENERAL PRE-VENTION AGAINST LOG4J TYPE OF ATTACKS. "**

crown jewels are protected by technology and policies that are aligned with their importance. When a Kipling method policy is properly deployed, a so-called microperimeter is placed around the protect surface.

The microperimeter ensures that only known, approved and validated traffic has access to the protect surface, based on policy. Another Zero Trust architectural principle is to move your defensive measures as close as possible to the protect surface to enforce the most effective preventative controls. This is a stark contrast with the traditional "outside perimeter" / DMZ based defensive measures in older architectures, in which the infrastructure inside the single perimeter is considered trusted.

The concept of protect surface offers several layers of general prevention against Log4J type of attacks. Because it puts strict policy limitations on the traffic between the various parts of your infrastructure, it is much harder for attackers to proceed after the initial access, through discovery, lateral movement, and the establishment of a command-and-control instance.

Even when the attackers succeed in establishing a malicious foothold, segmented protect surfaces offer another level of preventive protection against data collection and exfiltration of data. Strictly enforced protect surface policies limiting the usage of protocols such as JNDI, DNS, SMB or LDAP would have offered an extremely effective generic prevention against many notorious vulnerabilities. And even if an attack on one part of the infrastructure is (partly) successful, the blast radius is reduced greatly.

LOGGING AND INSPECTION STILL RELEVANT

The Zero Trust principle of logging AND inspecting all traffic, by itself does not seem to constitute a preventive measure against unknown threats.

Strict granular access controls must do the heavy lifting, but the inspection of all traffic helps general prevention by making sure that the policies specified are in place and effective (never trust, always verify).

Logging of all event data also makes the SOC analyst lives a lot easier when they want to look for suspect traffic or malicious requests in order to constantly optimize measures and access policies

Many vendors claim to offer prevention. But when you dig deeper in the technology, much of that prevention is based on chasing known indicators-of-compromise (such as hashes, domain names or IP-addresses) specific to a published CVE or other vulnerability. AI-based tools looking for so called behavioral indicators of compromise take a more general approach, but follow the same conceptual model. They are chasing the burglars that are already in your house.

Obviously, proactive threat hunting to scan for vulnerable impacted DAAS-elements makes sense, especially after a disclosure or publication of a major new threat.

But you should spend most of your time and budget on implementing a strategy that tries to keeps them out in the first place.

Zero Trust is that strategy.

MORE INFORMATION?
www.on2it.net/zero-trust