illumio

6 Steps to Implementing a Zero Trust Model

Zero Trust isn't hard — if you're practical

By Raghu Nandakumara
Senior Director of Industry Solutions Marketing
Illumio

In collaboration with

HELP AG
an e& enterprise company

Table of Contents

Introduction

Despite acknowledging its security benefits, many organizations hesitate to implement a Zero Trust model.

The primary concern is that "brownfield" environments have too much technical debt to overcome, so it can be applied only to net new environments (greenfield).

Further, organizations assume benefits can be realized only when everything has been Zero "Trustified" all at once — that there's no in-between state on the road to Zero Trust that's beneficial and achievable.

According to Forrester's Zero Trust framework, to achieve a complete Zero Trust posture, an organization must:

1. **Implement least-privilege access across all workloads, networks, people, devices, and data.**
2. **Ensure these controls are fully driven and maintained through automation.**
3. **Leverage visibility as a facilitator for #1 and #2.**
4. **Monitor continuously to maintain the integrity of the Zero Trust state.**

That's quite a task. No wonder some companies choose to defer putting it into practice. But what if, instead of taking the "all or nothing" waterfall approach to delivering Zero Trust, we took a more incremental, agile approach that allows an organization to make small, realistic steps toward achieving Zero Trust.

> An incremental, agile approach allows an organization to take small, realistic steps toward achieving Zero Trust.

6 steps to Zero Trust

Each step builds on what comes before, continuously improving the overall security state, even empowering brownfield environments to adopt and benefit from Zero Trust.

1. **Identify what to protect:** Identify the data, application, or business process you want to protect.
2. **Determine which Zero Trust pillar to focus on:** Which of the Zero Trust pillars are you going to build controls for? Choose a pillar that will secure your most critical assets and systems first. Segmentation is a good place to start.
3. **Specify the exact control:** Let's assume you want to segment the workloads that run your critical business process from the rest of the network. So, the Zero Trust outcome you are trying to achieve is least-privilege access over the network to workloads running this critical process.
4. **Prescribe what data is needed:** You now need the data and visibility (in Illumio's case, we provide a map) to build the specific policy that will achieve the outcome. This consists of relevant metadata to identify workloads and their associated dependencies, plus traffic data that determines the nature of those dependencies.
5. **Design the policy:** Armed with these data points, you can build a Zero Trust segmentation policy for this particular business process and validate it. Test before implementing. Zero Trust will get zero thanks otherwise.
6. **Validate, implement and monitor:** Once the policy is in place, traffic and tamper monitoring allows us to continually monitor the posture of your environment and react to any changes, either manually or through automation.

The remainder of this ebook will explain each step to achieving Zero Trust with a practical, iterative approach.

An incremental six-step approach to Zero Trust

1 Identify what to protect

In the infancy of workplace technology, organizations took a very "horses for courses" approach to adopting new capabilities — whether new hardware, operating systems or software. It was always about what most suited the job at hand, rather than trying to solve for a generic use case. On a small scale, these ad hoc solutions were manageable, and more productive than trying to pursue strategic solutions that could be relevant across the board.

> With an "all or nothing" approach, showing ROI can be difficult.

But as organizations grow, there's a tipping point where the cost and effectiveness of running ad hoc solutions are overruled by the ability to leverage generic components that can be reused to build effective solutions everywhere. This context illustrates how you can make quick, early progress with a strategic long-term and complex initiative like adopting Zero Trust by starting with tactical, highly specific, short-term, and relatively simple steps.

Ultimately, the objective should be to put a Zero Trust framework in place across your organization. But achieving complete Zero Trust requires many steps, and showing ROI can be difficult with an "all or nothing" approach. If you can't show measurable progress in the short term, interest around the initiative often fades.

Instead, look to achieve enterprise-wide adoption of Zero Trust by targeting an application or collection of applications that:

- **Have a strong driver to adopt Zero Trust security principles.** This could be a compliance or regulatory mandate, or an audit finding that must be remediated. Another strong driver for adopting Zero Trust comes from an "incident," and as the old saying goes, "never waste a crisis." This ensures there is a willingness (and need) to accept change.
- **Are marked as critical applications.** Focusing on critical applications applications early in the process provides the best opportunities for learning and can provide confidence in the technology's benefit to other, less critical parts of the enterprise. These are also applications that key decision-makers are often most aware of, so progress will give them a direct line of sight to ROI from the Zero Trust initiative.
- **Have a willingness to be guinea pigs.** This is an experiment, and there's no doubt it comes with risks. By adopting Zero Trust, you're flipping the usual access models on their head, which could result in growing pains. Working with application teams that are comfortable with adoption risks is hugely valuable. They'll be your future champions as you look to widen adoption.

SWIFT (Society for Worldwide Interbank Financial Telecommunication) or PCI-DSS systems, development workloads on a production network, critical security services, applications running unpatched or end-of-life components are all great candidates to be initial adopters of Zero Trust segmentation. They tick at least two of the requirements above. Do you have applications that fit in those categories?

> Focusing on critical applications early in the process gives decision-makers a direct line of sight to ROI.

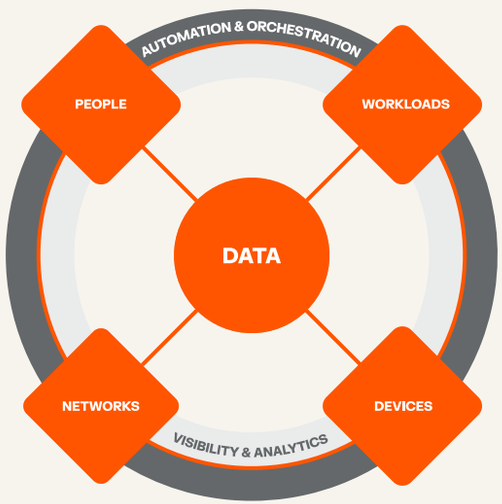
2 Decide which Zero Trust pillar to work on

Imagine you've found an application that handles PCI data so it needs to be compliant with PCI-DSS requirements. It also has an audit finding around "excessive network access," marked as a critical component in your customer payments system, and the application team has agreed to work with you in adopting a Zero Trust framework to secure the application. Your application meets all three criteria.

Now you can move to Step 2 of implementing Zero Trust.

In the Forrester framework, there are seven Zero Trust pillars. Trying to address every pillar is unrealistic. Plus, it complicates the task of making relatively quick and measurable progress. Besides the framework, Forrester provides a Zero Trust Maturity Model Assessment Tool to help practitioners identify gaps in their organization's adoption of Zero Trust. The results of the assessment can identify the appropriate Zero Trust pillar for your initiative.

Forrester's Zero Trust Framework Pillars



3 Specify the exact control

Workload protection encompasses many security capabilities, including effective securing and patching of the OS and installed applications, host-based threat protection controls such as antivirus or endpoint detection and response (EDR), file integrity monitoring, and host-based firewalling.

Implement microsegmentation to address excessive network access to application workloads and move your organization forward in its Zero Trust journey.

Since your Zero Trust Maturity Assessment identified excessive network access to application workloads, microsegmentation is a critical security control for your organization's Zero Trust journey.

To effectively implement microsegmentation, you need excellent visibility into your application's upstream and downstream dependencies. You need to have the right information with which to build and validate the least-privilege, allow-list Zero Trust policies that will protect your workloads. And that takes us to the next step.

4 Gather the information you need

Once you've zeroed in on what you want to improve protection for — workloads — and the controls you want to employ — microsegmentation — you can begin gathering the information needed to implement those controls. How do you accomplish this? There are two possibilities.

You have pre-existing knowledge of necessary flows: specify a segmentation rule based on Source IP, Destination IP, Port, and Protocol

You're starting from scratch in a brownfield environment: Get traffic logs to help you identify flows that may be relevant

Have you ever spent many hours looking at firewall traffic logs to figure out what a particular connection does? Have you been forced to hunt for information or people who can provide valuable context to a flow so that its purpose can be understood? Instead, what if you could understand traffic event context just by looking at it? Let's use an example.

Usual log traffic

- Source: 10.0.0.1
- Destination: 192.168.0.1
- Port: 53
- Protocol: UDP (User Datagram Protocol)
- Action: Allow

Traffic log with context

- Source: 10.0.0.1
- Source Context: Web Server, Payments Application, Production, UK
- Destination: 192.168.0.1
- Destination Context: DNS Responder, DNS Infrastructure, Production, UK
- Destination Process: named
- Port: 53
- Protocol: UDP
- Action: Allow

The version with context is clearly superior. It provides a complete picture of the flow: You can see that the Production Payments Web Server has a dependency on the Production DNS Responder, which has the named process receiving connections on 53/udp.

What if you could understand traffic event context just by looking at it?

You can quickly decide if it's a flow you're interested in, if it's normal traffic or whether it warrants further investigation. You can classify it easily because of the additional context you have.

When talking about microsegmentation for protecting workloads, the minimum metadata you need outside a standard traffic report describes workloads in the context of your data center applications and environments.

Illumio uses this metadata harvested from an organization's CMDB or other authoritative source to populate the labels associated with a workload. These labels associate a role, application, environment and location with each workload and help build a rich application dependency map that clearly identifies upstream and downstream dependencies for each application. This puts us in a great position to review flows and design policy.

5 Design the policy

You'll see below examples of a few traffic log entries that can be used to build a policy:

Traffic log connection 1:
- Source: 10.0.0.1, 10.0.0.2
- Source Context: Web Server, Payments Application, Production, UK
- Destination: 192.168.0.1
- Destination: Context: DNS Responder, DNS Infrastructure, Production, UK
- Destination Process: named
- Port: 53
- Protocol: UDP
- Action: Allow

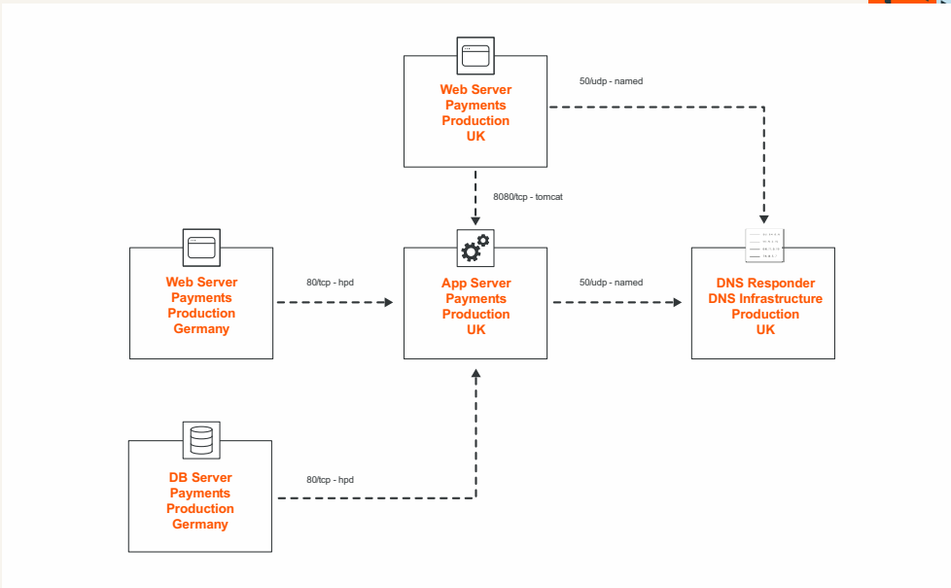
Traffic log connection 2:
- Source: 10.0.0.1,10.0.0.2
- Source Context: Web Server, Payments Application, Production, UK
- Destination: 10.0.1.5,10.0.1.6,10.0.1.7
- Destination Context: App Server, Payments Application, Production, UK
- Destination Process: tomcat
- Port: 8080
- Protocol: TCP
- Action: Allow

Traffic log connection 3:
- Source: 10.0.1.5, 10.0.1.6,10.0.1.7
- Source Context: App Server, Payments Application, Production, UK
- Destination: 192.168.0.1
- Destination: Context: DNS Responder, DNS Infrastructure, Production, UK
- Destination Process: named
- Port: 53
- Protocol: UDP
- Action: Allow

Traffic log connection 4:
- Source: 10.1.0.1,10.1.0.2
- Source Context: Web Server, Payments Application, Production, Germany
- Destination: 10.0.1.5,10.0.1.6,10.0.1.7
- Destination Context: App Server, Payments Application, Production, UK
- Destination Process: httpd
- Port: 80
- Protocol: TCP
- Action: Allow

Traffic log connection 5:
- Source: 10.1.2.1,10.1.2.2
- Source Context: Database Server, Payments Application, Production, Germany
- Destination: 10.0.1.5,10.0.1.6,10.0.1.7
- Destination Context: App Server, Payments Application, Production, UK
- Destination Process: httpd
- Port: 80
- Protocol: TCP
- Action: Allow

APPLICATIONDEPENDENCYMAP



Using this, you can quickly derive the application dependency map.

Look at your application dependency map to determine which flows you want to permit. Based on knowledge of your application, you know the following are required flows:

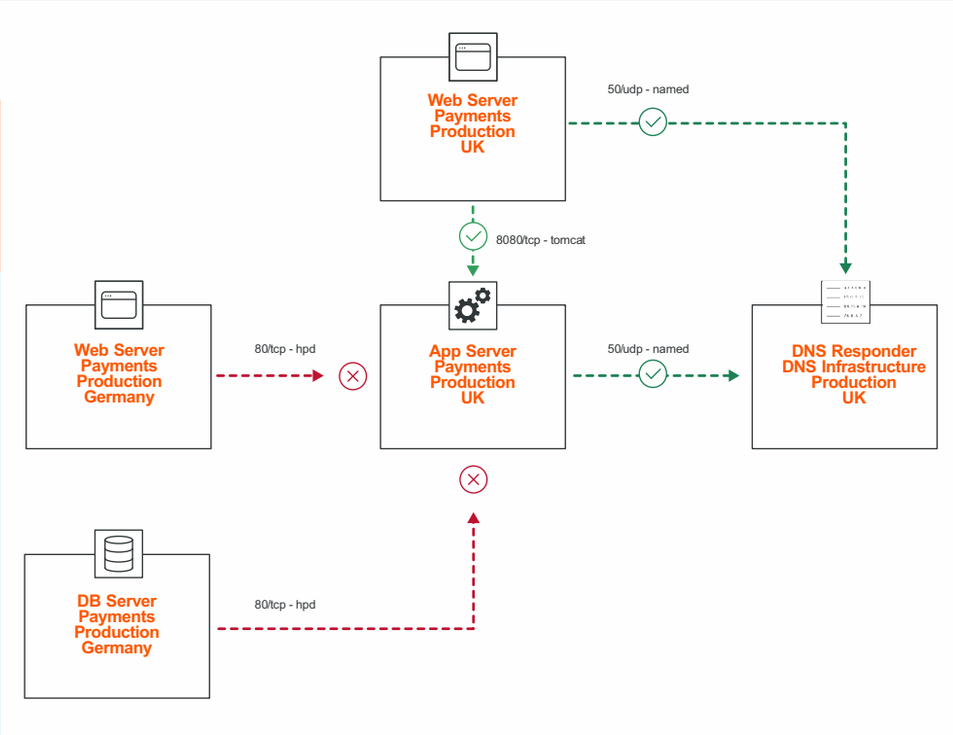
1. Web Server, Payments, Production, UK → DNS Responder, DNS Infrastructure, Production, UK on 53/udp
2. App Server, Payments, Production, UK → DNS Responder, DNS Infrastructure, Production, UK on 53/udp
3. Web Server, Payments, Production, UK → App Server, Payments, Production, UK on 8080/tcp

You also know the following two flows don't look right and shouldn't be included in your initial rules:

1. Web Server, Payments, Production Germany → App Server, Payments, Production, UK on 80/tcp
2. DB Server, Payments, Production, Germany → App Server, Payments, Production, UK on 80/tcp

The application dependency map that you'll use to build rules will end up looking like this:

APPLICATIONDEPENDENCYMAPWITHBLOCKEDFLOWS



How can you express these rules?

With traditional firewalls, you would be forced to define the rules using source and destination IP addresses. But this completely removes the rich context information you've gained when discovering these flows. It also means this context must be reinserted when the rule comes under review. So, what happens when you add an additional DNS responder, a new app server, or a web server for the payments app?

Just as your policy expands to incorporate a new server with existing context, you'll want the policy to shrink when you decommission a server. If you retire one of your DNS Responders, for example, you'll want all rules that previously allowed access to/from it to be updated so this access is no longer possible.

This is exactly what Illumio's Policy Compute Engine (PCE) does. Microsegmentation policy is defined using metadata, and the PCE determines which workloads match the metadata to compute the rules that must be enforced for each workload to maintain its Zero Trust security posture. Every time there's a change in context, the PCE adapts the policy and notifies workloads of updates.

With this in mind, your Zero Trust policy boils down to the following rules:

Rule 1:

- Source: Web Server, Payments, Production, UK
- Destination: DNS Responder, DNS Infrastructure, Production, UK
- Destination Service: 53/udp
- Destination Process: named

Rule 2:

- Source: App Server, Payments, Production, UK
- Destination: DNS Responder, DNS Infrastructure, Production, UK
- Destination Service: 53/udp
- Destination Process: named

Rule 3:

- Source: Web Server, Payments, Production, UK
- Destination: App Server, Payments, Production, UK
- Destination Service: 8080/tcp
- Destination Process: tomcat

For protecting workloads, the minimum metadata you need to design an effective policy using microsegmentation includes:

1. Real-time traffic events for the workloads you want to protect
2. Contextual data for each workload and connection, including metadata from a system of record such as a CMDB, and details of the communication process sourced directly from the workload
3. An application dependency map (derived from items 1 and 2) that allows an application owner or segmentation practitioner to quickly visualize a specific application's upstream and downstream dependencies

6 Validate, implement, and monitor

Now that you have the microsegmentation rules defined, you're ready to enforce them and protect your workloads, but one key challenge remains. Your Payments application is in production and you don't want to disrupt its functionality while you "ringfence" it. How can you mitigate this risk?

With any segmentation effort, the stage that carries the greatest risk is enforcing policies that specify no other traffic is permitted into or out of the workloads. If the policies are wrong, there's a chance of causing a production outage. The move to enforcement must be controlled, with sufficient opportunities for monitoring so problems can be quickly detected and fixed.

> Anything not explicitly permitted is implicitly not permitted, which maintains the principle of least privilege.

Policy testing

This is where policy testing comes in. Illumio provides a powerful but simple way to do this. One of its most useful features is the ability to move workloads (or groups of workloads) into test mode. Test mode is a non-blocking mode that reports policy violations. For a workload in test mode, the PCE overlays the connectivity graph built using workload flow data with the policy graph.

The policy graph can be thought of as putting bubbles around workloads allowed to communicate over a specific set of ports and protocols. The connectivity graph shows the attempted communications between workloads.

Where the connectivity graph is within a bubble on the policy graph, you get green lines. These are flows that match policy we've authored. Where the connectivity graph crosses a bubble on the policy graph, you get red lines These are flows that don't match an authored policy.

In test mode, these red lines, while not blocking any flows, indicate where we have connectivity attempts that are policy violations. As an application owner, these are the flows you review. Your choices are:

- Flow is required → write policy to turn line green
- Flow is not required → no need to take any action

So, the policy validation process requires iterating through all these "red" lines to determine whether they need to be turned green. Once you've reviewed these violations and made your choice, you're ready to start protecting the workloads. The validation process is complete, time to enforce.

Keep in mind the purpose of the validate, implement and monitor phase is minimizing risk. You don't want to take a "big bang" approach to enforcing policy. Despite detailed validation, you'll still want to take incremental steps in this final phase, which is the granular control that Illumio provides.

Each workload in an application can be moved individually to enforced mode. Once you've fully validated a policy, you can select which workloads you want to apply full protection to first. Let those run with the policy enforced and move the other workloads to an enforced state when you're ready.

If there are any issues with a policy, it will affect only a small set of workloads rather than the entire fleet. It provides another opportunity to fine-tune before enabling policies for the entire application.

Once all workloads are enforced and the application is protected, the task is to continuously monitor traffic events for anything unexpected — drops and accepts — and investigate anything that's outside the normal behavior.

> You don't want to take a "big bang" approach to enforcing policy.

A practical approach to Zero Trust

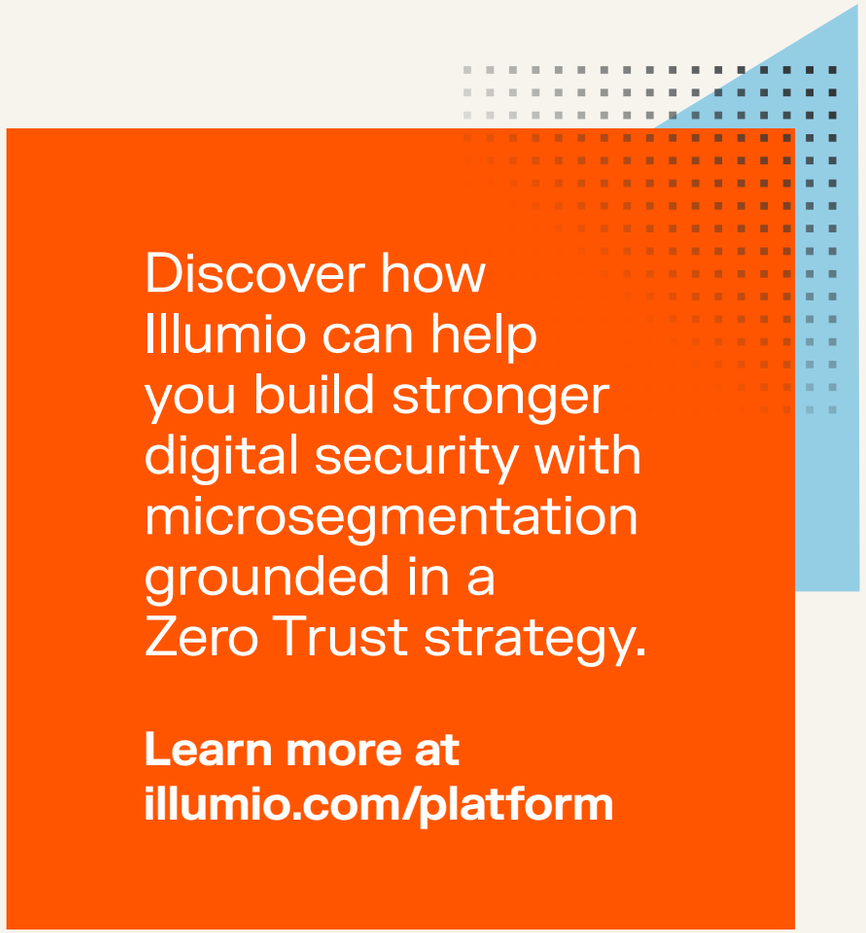
So, there you have it, a walkthrough of the six steps to pragmatically build your Zero Trust security program.

Zero Trust is not an outcome by itself but a security strategy. Each organization needs to understand its maturity across the Zero Trust pillars, identify which pillars need the most focus, and take incremental steps to improve that maturity.

Illumio: lighting a path to Zero Trust

Illumio is the world leader in breach containment, helping organizations around the globe protect their critical applications, workloads, and assets. Enhance security. Simplify compliance. Stay resilient and agile in the face of today's cyber threats.

Illumio's intelligent visibility and automated security enforcement stops lateral movement, preventing malware and cybercriminals from infiltrating your network, data centers, and devices.



About Illumio

Illumio is the leader in breach and ransomware containment, redefining how organizations contain cyberattacks and enable operational resilience. Powered by the Illumio AI Security Graph, our breach containment platform identifies and contains threats across hybrid multi-cloud environments – stopping the spread of attacks before they become disasters. Recognized as a Leader in the Forrester Wave™ for Microsegmentation, Illumio enables Zero Trust, strengthening cyber resilience for the infrastructure, systems, and organizations that keep the world running.

illumio