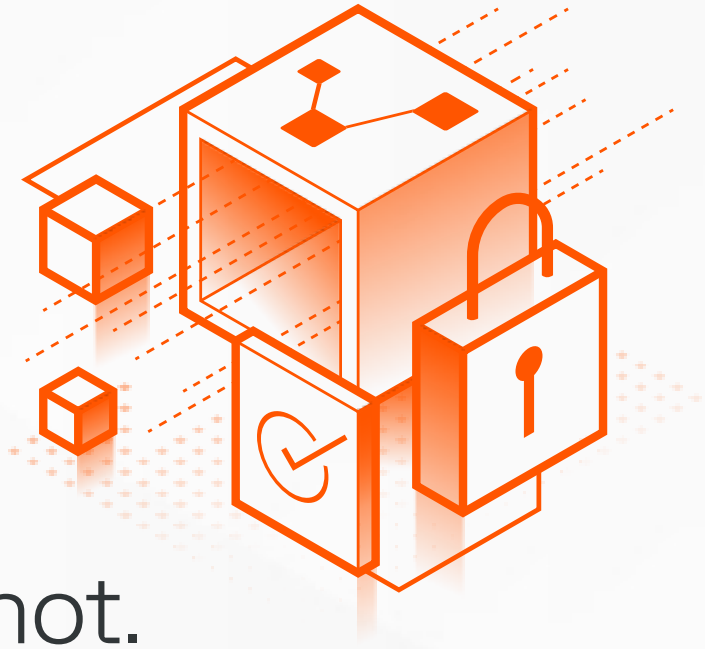
illumio

EBOOK

The Buyer's Guide to Modern Microsegmentation

Four approaches that fail — and what actually stops lateral movement

Contents



INTRODUCTION

Segmentation is everywhere. Effectiveness is not.

Cybersecurity has a spending problem — not too little, but too much of the wrong kind. Year after year, budgets grow. Tool counts climb. And breaches keep getting worse. The IBM Cost of a Data Breach Report puts the average cost of a breach at $4.88 million, a 10% jump from the year before and the largest spike since the pandemic.[1] If the old model worked, those numbers would be going down. They aren't.

The core of the problem is a decades-old bet on prevention. Stop every threat at the perimeter. Detect every intrusion before it spreads. Buy another tool, add another layer, hope the stack holds. But attackers keep getting in. Ransomware still jumps from host to host. Key apps still go offline. Business still halts for days or weeks. The model isn't underperforming. It's fundamentally broken.

Microsegmentation was supposed to be part of the fix. And on paper, it is. Nearly all large firms now report using at least one method to segment their networks. Analysts now treat it as a baseline, not a nice-to-have. But adoption alone doesn't equal protection.

Many firms believe they have solved the segmentation problem when they have really just applied the same broken thinking to a new category. They rely on legacy tools that were built for static, perimeter-centric networks — tools that can't keep up with modern, hybrid environments. It's like putting locks on every door in a building but handing everyone a master key. The look of control is there; actual protection is not.

This is what happens when the industry treats segmentation as a checkbox instead of a strategy. The tool gets bought. The box gets checked. And when the breach comes, lateral movement spreads unchecked because the controls were never built to stop it.

This e-book:

- Explores the four most common ways to segment IT environments
- Shows where each one falls short
- Defines what a modern approach requires

The goal: help security leaders see past the checkbox, judge their current posture honestly, and build a segmentation strategy that actually contains breaches when prevention fails — because it will.

[1]Ponemon Institute. "Cost of a Data Breach Report 2024." July 2024.

Why older segmentation methods no longer work

Modern IT runs on speed and scale. Apps are spread across data centers, clouds, and edge sites. Containers spin up and shut down in seconds. IP addresses shift as workloads constantly grow, shrink, and move.

Yet many older methods still assume a world that no longer exists: fixed network lines, static workloads, and traffic that flows in set patterns. When the setting changed, these models became a liability.

The result is fragile controls that are hard to keep up and easy to bypass. Field work shows a clear pattern: when segmentation depends on the network, security teams must choose between keeping things running and strong enforcement. Over time, keeping things running wins. Policies get looser. Gaps pile up. The original intent erodes.

Network-centric models can't keep pace with modern app designs. When segmentation is tied to the network, every change in the setting demands a matching change in policy. That link adds risk, cost, and delay.

Think of it as drawing lines on a map that keeps changing. Every time the terrain shifts, the lines must be redrawn. At some point, teams stop trying. The lines stay where they are, and the map no longer matches the reality on the ground.

The result is a posture that looks good on paper but fails under pressure. Audits pass. Boxes get checked. But when an attacker gets in and starts to move, the controls that should stop them are full of holes.

The four common approaches

Most firms rely on one or more of these methods today. Each has value in certain cases, but none fully handles the realities of hybrid and multi-cloud settings on its own.

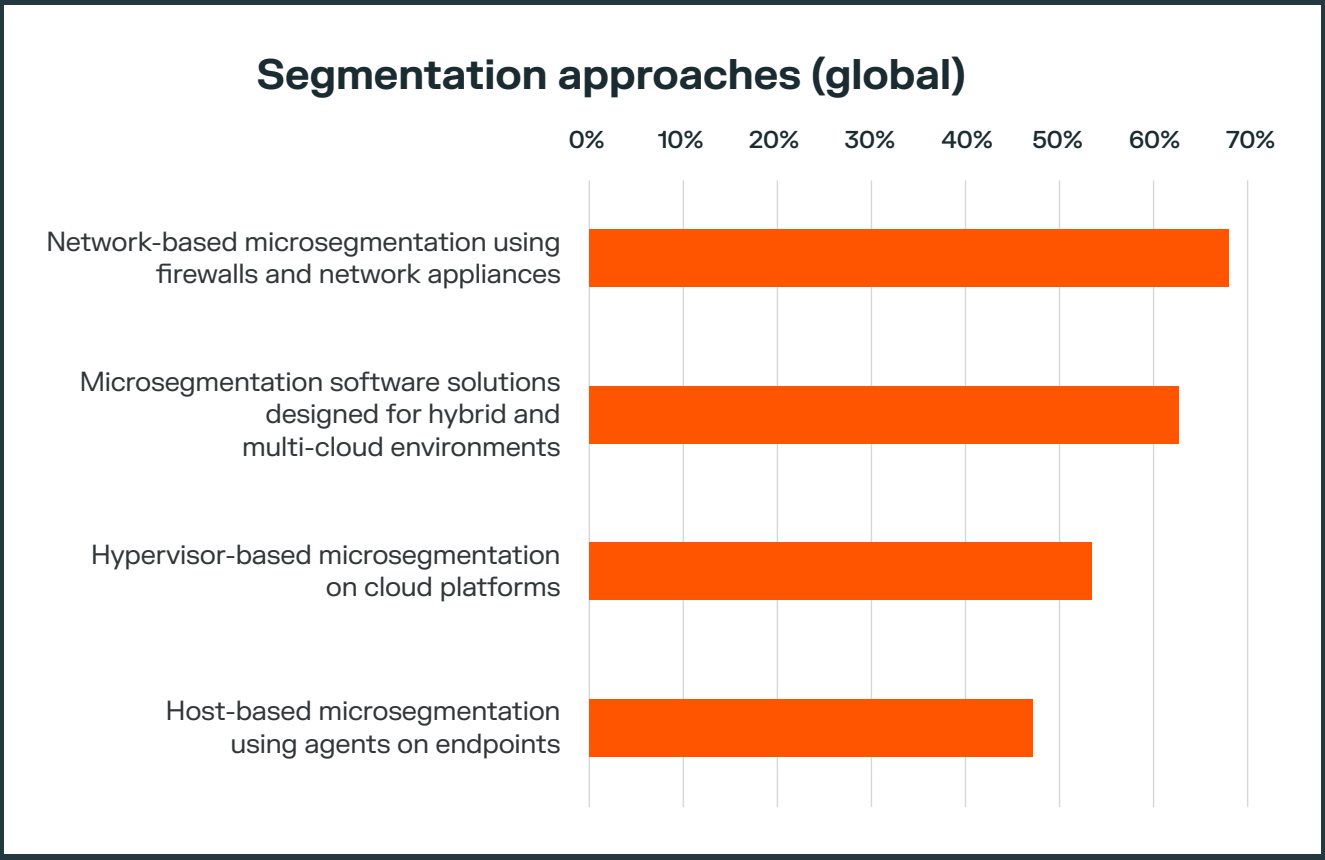
Microsegmentation by the numbers

Most teams use more than one method to segment their networks. But the mix tells a story about gaps, not strength.

Firewall-based segmentation remains the most common approach. Nearly 68% of firms use network firewalls or appliance-based methods, often layered together at key choke points.

Hypervisor-based controls cover about 53% of teams, almost always limited to a single platform. Agents are the smallest segmentation category at 47% but are the fastest-growing category.

The pattern is clear: most firms have segmentation in place, but the tools they rely on weren't built for hybrid, multi-cloud settings. High adoption hasn't led to strong outcomes. The gap between having segmentation and having effective segmentation is where breaches thrive.



Source: The Containment Gap

1. Network-based firewall segmentation

Firewalls are the default starting point for most segmentation efforts — and for too many firms, the ending point as well.

What it is

This method uses network firewalls to control traffic between systems based on IP addresses, ports, and protocols. It is the oldest and most common form of segmentation. The tools in this category include stateful packet inspection firewalls, next-generation firewalls (NGFWs) with app-aware filtering, internal segmentation firewalls placed between network zones, and cloud-native constructs like security groups and network access control lists (ACLs). All of these share the same basic model: filter traffic at the network layer using IP and port rules.

Why teams use it

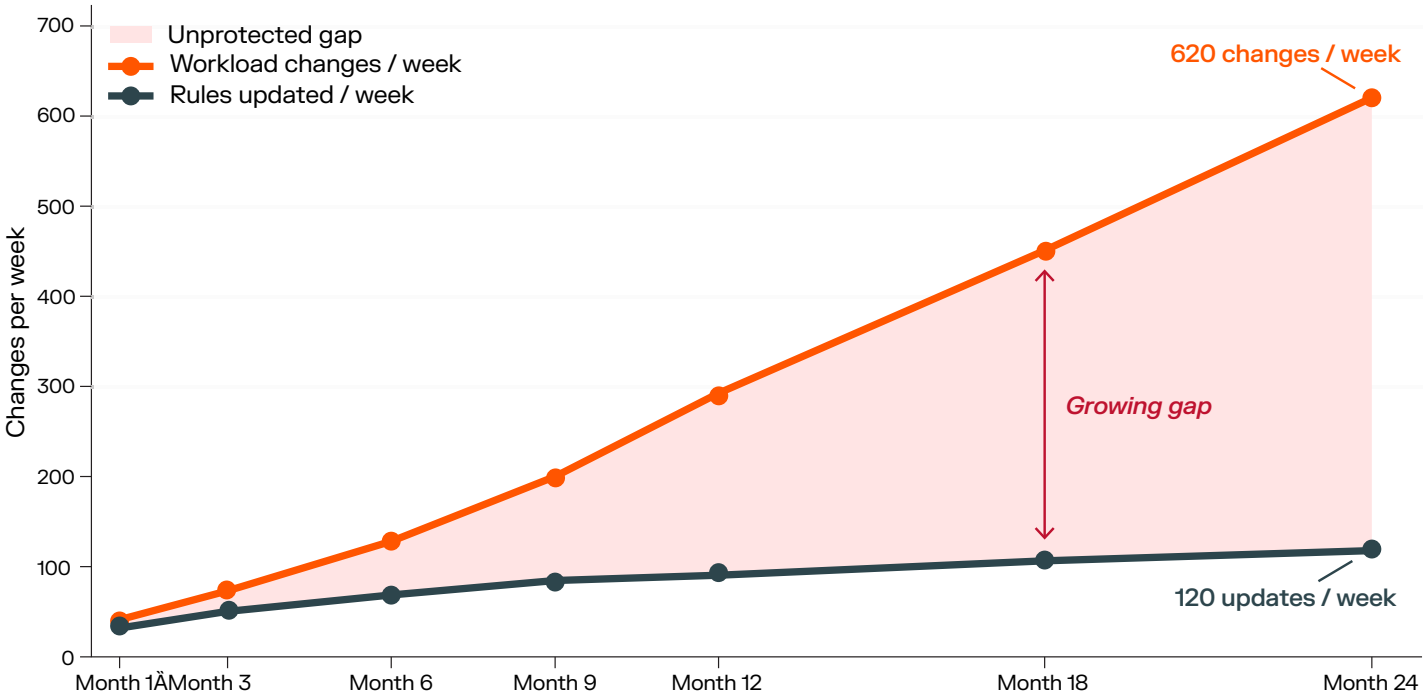
These tools are already in place in most settings, and teams know how to run them. For broad segmentation — splitting production from development, for example — firewall-based methods work and are well understood. Most firms have years of staff training, playbooks, and change processes built around firewalls and ACLs.

Where it breaks down

IP-based rules are brittle when things change fast. When workloads scale on their own, move between clouds, or swap IP addresses, the rules that should protect them go stale. A single bad rule can take down live traffic.

In practice, rules grow faster than teams can track them. A mid-size firm might have tens of thousands of rules across many firewalls. Over time, gaps pile up, clarity drops, and no one can tell what is really being enforced. Security teams call this “rule sprawl.”

App-to-app links are also hard to map from firewall data alone. Without clear sight into how apps talk to each other, teams write broad rules. Broad rules defeat the whole point of segmentation.



2. Appliance-based segmentation

When teams want deeper inspection than a firewall can offer, they often turn to dedicated appliances. The tradeoff is more control at the cost of more complexity.

What it is

Traffic is routed through dedicated physical or virtual appliances that inspect and control flows. These devices sit inline and enforce policies at choke points in the network. The tools in this category include deep packet inspection (DPI) devices, inline intrusion prevention systems (IPS), SSL/TLS inspection appliances, and virtual appliances deployed as transit gateways in cloud settings. What they share is the same basic design: all traffic must pass through the device before it can reach its target.

Why teams use it

This method offers a single point of control and strong enforcement at key spots in the network. For deep packet checks or protocol-level review, appliances can do things other methods can't.

Where it breaks down

All traffic must pass through the device, which adds lag and creates a single point of failure. During peak load or an active incident, the device itself becomes the bottleneck. If it goes down, enforcement goes with it.

Scaling means buying more hardware or spinning up more instances. This model assumes traffic flows are set and centered — rarely true in cloud-first designs where workloads talk to each other across regions and providers.

It's like routing all highway traffic through one toll booth. At low volume, it works fine. As traffic grows, the booth becomes the choke point. And if the booth closes, everyone stops.

3. Hypervisor-based segmentation

Hypervisor-based segmentation moves enforcement closer to the workload — but only within the platform it's built on. That distinction matters more than it might seem.

What it is

Controls are enforced at the virtualization layer, within a specific hypervisor platform or cloud stack. Policies apply to virtual machines based on their role and placement within the virtual setup. The tools in this category include distributed virtual firewalls built into hypervisors, virtual switch filters that inspect traffic between VMs on the same host, software-defined networking (SDN) platforms with built-in, and cloud-native VPC firewalls and virtual network perimeter controls. Each of these enforces policy within a single platform or stack.

Why teams use it

Within one virtual setup, this method offers fine-grained control and tight ties to the platform. For firms that run one stack, it can work well within that scope.

Where it breaks down

The limit is scope. This method is confined to the platform it runs on. It doesn't stretch easily to other clouds, bare-metal servers, containers, endpoints, or OT systems. In practice, this forces teams to manage different tools for different parts of their setup.

The result is split enforcement. Rules in one setting don't carry over to another. Gaps form at the edges between platforms. And the work of managing many tools adds up fast.

Cases in manufacturing and critical systems show this clearly. When workloads span IT and OT lines, platform-specific tools leave key systems exposed. A plan that only covers part of the setting gives a false sense of safety.

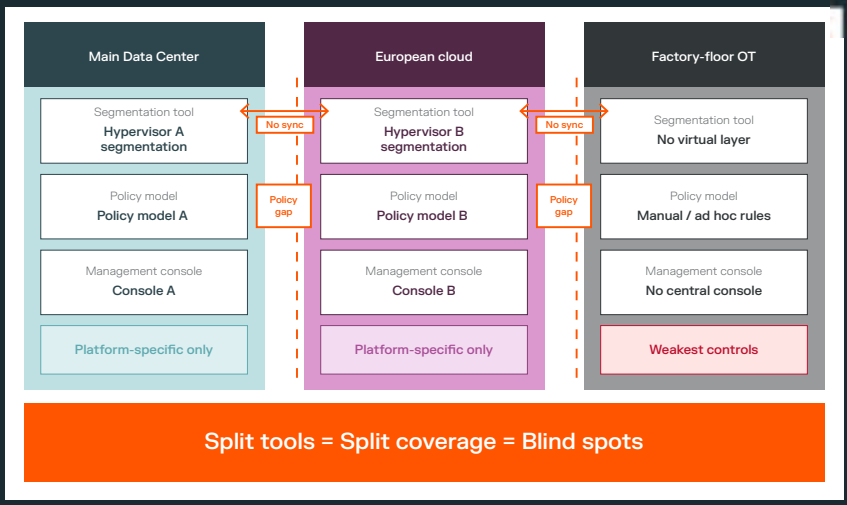
The hidden cost of split enforcement

A global manufacturer runs VMware in its main data center, a different hypervisor in its European cloud, and has no virtual layer at all in its factory-floor OT network. Each setting has its own segmentation tool, its own policy model, and its own management console.

When the security team tries to enforce a single policy across all three, it must write three sets of rules, manage three dashboards, and reconcile three views of what is allowed. Drift between them is constant. A rule change in one console doesn't carry over to the others.

The result: more staff hours spent on upkeep than on strategy. Gaps at the seams between platforms. And a growing risk that the OT network — often the most critical and least visible — is the one with the weakest controls.

This isn't a rare edge case. Any firm that runs more than one platform faces the same problem. Split tools mean split coverage. And split coverage means blind spots.



4. Host-based agent segmentation

The first three approaches all share a flaw: enforcement is tied to something other than the workload itself. Host-based agents fix that — but not every tool solves the whole problem.

What it is

Policies are enforced right on each workload using a lightweight software agent installed on the host. This puts control as close to the workload as possible, no matter where it runs. The tools in this category include host-based firewall management platforms that program native OS firewalls like iptables and Windows Filtering Platform (WFP), eBPF-based enforcement agents that filter traffic at the kernel level, and endpoint agents that combine workload telemetry with policy enforcement. Some endpoint detection and response (EDR) platforms also offer basic host-level firewall controls, though their main focus is threat detection rather than segmentation.

Why teams use it

Agent-based enforcement breaks the link between segmentation and the network. It works across data centers, clouds, containers, and endpoints. Policies follow the workload wherever it goes.

Where it breaks down

An agent alone doesn't mean good segmentation. Without clear sight into how apps talk to each other, policy design is guesswork. Teams often start with loose rules and tighten them slowly, leaving gaps that last for months.

Customer stories point to a key lesson: you must see before you enforce. Without a real-time map of app links, teams can't write policies that are both strong and safe. Enforcement without sight leads to rules that are too loose (and fail to contain breaches) or too strict (and break apps).

The agent is only one piece. What matters just as much is the policy model, the insight into traffic flows, and the ability to keep enforcement steady across all settings at scale.

Comparing the four approaches

	Network Firewalls	Inline Appliances	Hypervisor-Based	Host-Based Agents
Enforcement point	Network perimeter and internal zones	Dedicated physical or virtual devices	Virtualization layer within a single platform	Directly on each workload via software agent
Policy basis	IP addresses, ports, protocols	Deep packet inspection at choke points	VM role and placement within a virtual stack	Workload identity, independent of network
Core limitation	IP-based rules go stale as workloads scale and move	Adds latency; creates a single point of failure	Confined to one platform; doesn't span hybrid environments	Requires real-time visibility to avoid guesswork policies
Scaling challenge	Rule sprawl outpaces manual updates	More traffic means more hardware or instances	Each new platform needs its own toolset and rules	Agent deployment and policy consistency across all environments

What modern microsegmentation requires

The gaps in these methods all point to a broader truth: many tools in use today weren't built for modern settings. They didn't start out broken. They were built for a world of fixed, central systems — and that world no longer exists.

These gaps also explain why Zero Trust architecture has moved from concept to mandate. The principles behind Zero Trust — never trust, always verify, assume breach — demand segmentation that's fundamentally different from what most organizations have deployed. Modern microsegmentation must meet five criteria to deliver on that promise:

Software-defined and decoupled from the network

Segmentation can't depend on hardware, fixed topology, or network constructs. Policies must be defined in software and remain stable even as the infrastructure beneath them changes. When segmentation is tied to the network, every change becomes a security event. Cutting that link lets security keep pace with environments that shift constantly.

Identity-based

Policies must be built around what a workload is and what it does, not its IP address. When a workload moves, scales, or changes its IP, the policy should follow. This removes the brittleness of IP-based rules.

Consistent across all environments

From data center to cloud to endpoint, segmentation must use one policy model. Split enforcement creates gaps. One model ensures that an attacker who gets into one system can't exploit a gap to reach another.

Continuously adaptive

Environments don't stand still; segmentation policies can't either. Modern microsegmentation must detect changes in the environment — new workloads, shifted dependencies, decommissioned assets — and adjust policy in response. Static rules written once and left alone are the reason firewalls end up with tens of thousands of stale entries. Effective segmentation adapts continuously or it drifts into irrelevance.

Built for breach containment

Segmentation exists to limit the spread of attacks when prevention fails. Policies must be designed to stop lateral movement during a live incident, not just pass an audit or lower a risk score on paper.

Segmentation that meets these five tests adapts as settings evolve. It doesn't force teams to rewrite rules or rework networks. It gives a durable, scalable control that works in the real world.

Breach containment in action: the Bishop Fox ransomware test

To test whether microsegmentation can stop ransomware in practice, Bishop Fox ran a controlled attack simulation.[2] The setup modeled a real-world breach: an attacker gains a foothold on one system and tries to spread.

Without segmentation, the attacker moved freely. Within minutes, ransomware reached critical systems across the network. The entire setting was compromised.

With identity-based microsegmentation in place, the result was different. The attacker gained the same initial access. But every attempt to move laterally was blocked. The ransomware couldn't spread beyond the first host. The rest of the setting stayed up and running.

The test showed two things. First, containment works. Even when prevention fails, strong segmentation limits the blast radius to a single system. Second, speed matters. The policies were already in place and enforced in real time. There was no need to scramble, write new rules, or shut down the network.

This kind of third-party proof is critical. It shows that breach containment isn't just a concept on a slide. It works in conditions that mimic a real attack.

Why a modern approach delivers better results

When segmentation is built around identity rather than the network, firms gain clear, real advantages. Here's what to consider as you build or refine your detection and response approach.

Clear sight into app traffic

Before a single policy is written, a modern platform maps how apps talk to each other. This reveals links that teams didn't know existed. It removes guesswork from policy design. And it cuts the risk of breaking live systems during rollout.

Policies that follow workloads

Identity-based policies move with the workload. Whether an app runs in a data center today and moves to the cloud tomorrow, the policy stays attached. No need to rewrite rules each time something changes.

Faster response during live attacks

When an attacker gets in, speed matters. Modern microsegmentation lets teams isolate hit systems in minutes, not hours or days. Large-scale rollouts across global firms have shown that segmentation can be turned on fast during an incident, shrinking the blast radius and keeping the business running.

Lower ongoing cost

When segmentation isn't tied to the network, security teams spend less time on rules and more on strategy. Policy changes don't need sign-off from network ops. Enforcement doesn't wait for hardware swaps. The result is a leaner, more lasting operation.

The total effect is large. Firms that adopt a modern, identity-based method report faster time to enforcement, fewer outages from bad policy, and smaller blast radius during incidents.

[2]Bishop Fox. "Ransomware Scenario Emulation." August 2022.

How Illumio delivers modern microsegmentation

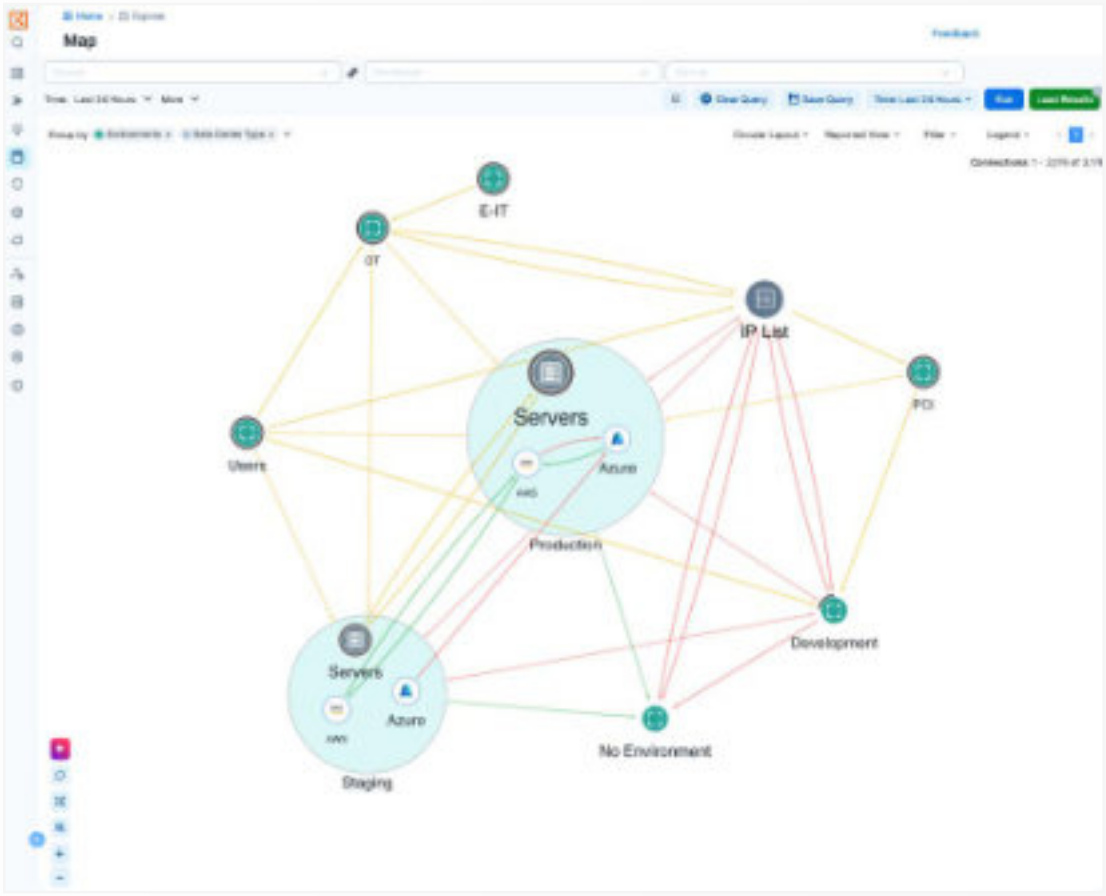
The five tests outlined earlier — software-defined, identity-based, decoupled from the network, consistent across settings, and built for breach containment — describe what modern microsegmentation must do. Illumio Segmentation meets all five.

This section walks through how, with enough detail for security leaders and architects to see the difference between a modern approach and the legacy models covered earlier.

See before you enforce

Illumio starts with visibility, not policy. The Illumio platform builds a real-time map of how apps talk to each other across every setting — data center, cloud, container, and endpoint. This map shows traffic flows, dependencies, and patterns that most teams have never seen in one place.

This isn't a one-time scan. The map updates as the setting changes. New workloads appear. Old ones shut down. The map stays current. Security teams use it to understand what is happening before they write a single rule. Illumio Insights takes this further. It layers risk context on top of the traffic map, highlighting where the setting is most exposed and where enforcement will have the greatest impact. This lets teams focus on the highest-value policies first rather than trying to segment everything at once.



Identity-based policy, not IP-based rules

Illumio uses a label-based policy model. Each workload gets a set of plain-language labels — role, app, setting, and location. Policies are written against these labels, not against IP addresses.

When a workload moves, scales, or changes its IP, the labels stay attached. The policy follows. There is no need to rewrite rules, update firewall tables, or file change requests. The link between policy and workload is durable by design.

This model also makes policies easy to read. A rule that says "web servers can talk to app servers in production" is clearer than one that says "10.0.3.x can reach 10.0.7.x on port 443." Clarity matters when teams need to audit, update, or debug policies under pressure.

Enforcement at the host, not the network

Illumio enforces segmentation using a lightweight agent on each workload. The agent programs the native OS firewall — iptables on Linux, Windows Filtering Platform (WFP) on Windows — to enforce policies right where the workload runs.

This means enforcement isn't tied to the network. There is no inline appliance. No traffic hairpinning. No single point of failure. If the network changes, the policy stays in place. If one host goes down, every other host keeps enforcing on its own.

The agent is lightweight by design. It doesn't inspect packet payloads or add latency to traffic. It enforces allow and deny decisions at the OS level, which means it works at line speed with minimal overhead.

One policy model across every setting

The same label-based policies apply whether a workload runs on bare metal in a data center, as a VM in a private cloud, as a container in a public cloud, or on an endpoint. There's no need to manage separate tools or write platform-specific rules.

This solves the split-enforcement problem that plagues hypervisor-based and appliance-based methods. One console. One policy model. One view of what's allowed and what's blocked. When the team writes a rule, it applies everywhere that label exists.

Built for breach containment

Everything in the Illumio approach points toward one goal: stopping lateral movement when an attacker gets in.

The visibility map shows where an attacker could move. The label-based policies define where movement is allowed.

[2]Bishop Fox. "Ransomware Scenario Emulation." August 2022.

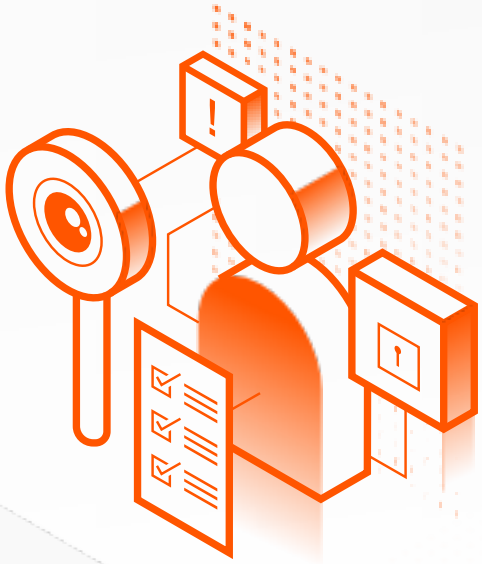
The host-level enforcement blocks everything else. And when an incident starts, teams can use Illumio to isolate hit systems in minutes — without touching the network, without filing a change ticket, and without waiting for a firewall team to respond.

The Bishop Fox ransomware simulation confirmed this in a controlled test.[3] With Illumio in place, the attacker gained initial access but couldn't spread. The rest of the setting stayed up and running. Containment worked in real time, with policies that were already enforced before the attack began.

Scale without complexity

Large global firms run Illumio across hundreds of thousands of workloads. The platform scales without forcing teams to add more hardware, manage more appliances, or split their policy model across multiple tools.

In most environments, scaling means deploying more lightweight agents — not buying more boxes. For settings where agents aren't practical, such as legacy systems, IoT devices, or certain cloud-native workloads, Illumio also supports agentless enforcement using network-level controls. Either way, the same label-based policy model applies. Adding new workloads means applying the right labels, not writing new IP rules. The work stays flat even as the environment grows.



Choosing the right strategy

Before picking any method, security teams should ask five questions. The answers will show whether a given tool can deliver real protection or just add more clutter.

Does this method depend on static IPs or network layouts?

If yes, it will struggle in any setting where workloads move, scale, or change addresses. That covers most modern firms.

Can it adapt as the setting changes?

Segmentation must keep pace with the systems it protects. Tools that need hands-on work each time a workload is added, moved, or retired will fall behind.

Does it give visibility before enforcement?

This is a key test. The ability to see how apps talk before writing a single rule is the gap between precise, useful policies and policies built on guesses.

Can it enforce the same way across all settings?

A tool that only covers one part of the setup leaves the rest open. The same rules must apply in the cloud, in the data center, and on endpoints.

Is it built to contain breaches, not just detect them?

Spotting a breach tells you something happened. Containing it limits the damage. The best plans assume breaches will happen and focus on shrinking the blast.

CONCLUSION

Segmentation is a strategy, not a tool

Microsegmentation is no longer optional. Rules require it. Insurers ask about it. Boards expect it. But the method matters far more than the label.

Legacy, network-bound models struggle in modern settings. They were built for a time when networks were stable, workloads sat still, and traffic moved in set patterns. That time has passed.

Identity-based, software-defined segmentation lets firms contain breaches even when prevention fails. It gives the visibility to build precise policies, the flexibility to enforce them anywhere, and the strength to limit damage during live attacks.

The goal isn't perfect security. No method can stop every breach. The goal is to limit damage when the worst happens — and to do so in a way that scales, adapts, and lasts.

Firms that treat segmentation as a strategy — not just a purchase — will be better placed to protect what matters most in the years ahead.

Try Illumio Segmentation

Legacy segmentation was built for static networks. Illumio Segmentation was built to contain breaches in dynamic, distributed environments — using identity-based policy enforced everywhere.

- Stop lateral movement across clouds, data centers, and endpoints.
- Replace brittle network rules with software-defined enforcement.
- Contain attacker activity without slowing the business.
- Scale segmentation as environments grow and change.

Get started at illumio.com/illumio-segmentation