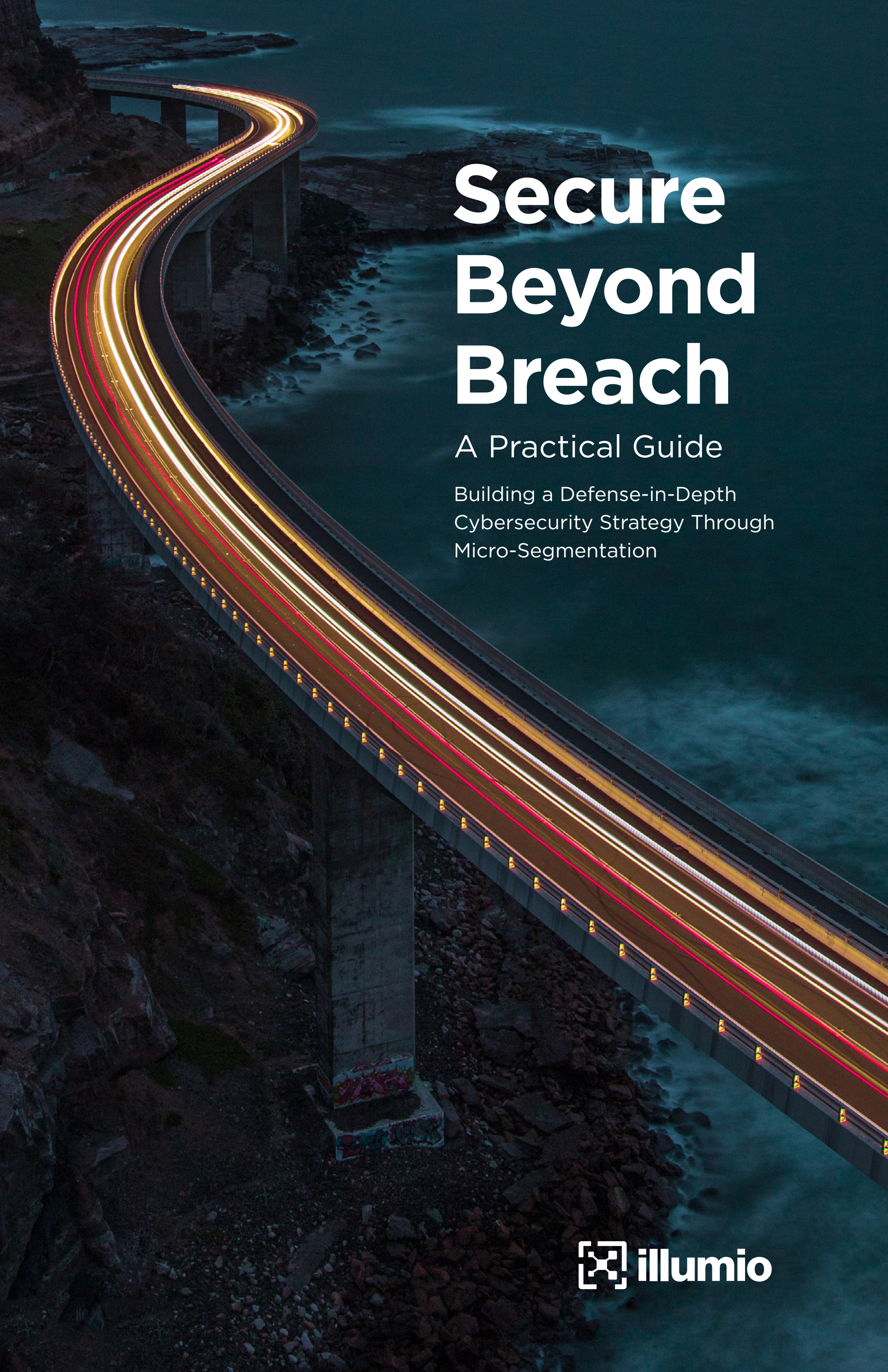


Secure Beyond Breach

A Practical Guide

Building a Defense-in-Depth Cybersecurity Strategy Through Micro-Segmentation

illumio

Table of Contents

Foreword: Become Secure Beyond Breach

JONATHAN REIBER

The cyberthreat has become a top-tier threat to international security and to organizations across the globe. Three trends made it so: the vulnerability of the data of cyberspace, the digital transformation of global society, and a lack of investment by organizations and governments in the people, processes, and technologies required to deter and defend against cyberattacks. It is not a question of if but when an organization will be breached in cyberspace. Governments, corporations, and other organizations have taken steps to improve their cybersecurity posture by building cybersecurity teams, developing response policies and mechanisms, and implementing security technologies – but progress has been insufficient to meet the threat.

Nation-state and non-state attackers steal, destroy, and manipulate data in and through cyberspace. Adversaries flourish in the "gray space" below the level of outright conflict and appear undeterred in pursuing their goals. Consider just a few examples: China's campaign to steal U.S. intellectual property, including data on the F-35 Joint Strike Fighter; North Korea's 2015 theft of $81 million from the Bangladesh Central Bank and U.S. Federal Reserve; China's theft of 21.5 million federal personnel records from the U.S. Office of Personnel Management (OPM); and Russia's destructive attacks on the Ukrainian electric grid in 2015–2016.

Each of these attacks impacted the victim companies and countries significantly. Nation-states in particular have the resources to put hackers on salary and can work diligently over time to penetrate a target. In recent years they've shifted focus from data theft and destruction to data manipulation of political and media targets. The Russian hack of the 2016 U.S. presidential election is the most notable example. On the express direction of Russian President Vladimir Putin, Russian military intelligence hacked into the networks of U.S. political organizations and political leaders, and exploited vulnerabilities in social media business practices to spread propaganda and foment mistrust within the American population. The Russian operation hit three parts of the American "center of gravity" during a period of acute transition: the American population, political leadership, and key technology companies. Other states have since taken similar actions. China reportedly penetrated Cambodia's electoral networks in 2018, affording it the potential opportunity for election manipulation.

Deterring and defending against an advanced attacker requires countries to implement comprehensive cybersecurity strategies – and the public and private sectors each have unique roles to play. While governments need to take the lead on deterrence strategies and strategic response options, companies and organizations can and must invest in cybersecurity capabilities to prevent intruders from gaining access to their data.

For an organization to defend itself against an advanced attacker, perimeter defenses like firewalls and multi-factor authentication are not enough. They cannot help once an adversary has broken into a data center or cloud environment. Strategically, organizations need to "assume breach" and plan for intruders to break into the interior. The average dwell time for an intruder to remain inside a network is six months. Once inside an insecure environment, absent internal network segmentation an intruder can move laterally with ease, just like the Chinese intruders did once they gained access to the networks of the U.S. Office of Personnel Management – a defining case study for data center security that will be discussed in this book.

From a security standpoint, the map of the world has changed. It used to be that hostile actors had to cross oceans or mountains to invade a country and break past the front gate. The internet has shrunk the strategic map of the world and has brought the enemy to every organizations' front door. Once inside, they can access the "crown jewel" applications that power an organization's missions, whether that be the databases that store your personal identity, the cloud services that store pictures of your children and your financial data, or the servers that transmit command and control instructions for a military. All of these applications live and operate within data centers and cloud environments.

Cyberspace connects every part of civilization. It is the new map of the world. A good micro-segmentation (sometimes referred to as security segmentation) strategy helps you see and control your terrain, map your own applications, and set rules for how servers interact. Rather than simply segmenting networks on a macro level, however, a robust segmentation strategy takes a granular approach, identifying and setting rules between key workloads, applications, and servers. This granular approach has been called "micro-segmentation" and it builds fences to ensure appropriate access and data flows within an organization. Once those fences are built, you can stop intruders from moving unencumbered from one server to another. Micro-segmentation provides a foundation of cyber resilience for an organization to withstand an attack, a final layer in a "new security stack" of firewalls, encryption, and multi-factor authentication. It provides a true defense-in-depth cybersecurity strategy.

This book will show you how to implement a micro-segmentation strategy from start to finish. It is designed to help you ensure that your missions continue even if the enemy has scaled your exterior walls. This book will help you become secure beyond breach.

If you would like to learn more, contact Illumio by email at info@illumio.com, by phone at +1-855-426-3983, or on Twitter at @illumio.

01 Introduction: The Evolving Cybersecurity Landscape

JONATHAN REIBER

Assume Breach

Organizations are being taken apart in cyberspace. Across the globe the total number of internet users has increased to four billion with an expected addition of one to two billion new users in Asia and elsewhere by the end of the decade. More data is being created and stored across more devices and data centers around the world than ever before. Yet access to data has increased without a commensurate or popular understanding of cybersecurity risk. The result is that the world is behind in cybersecurity and vulnerable to a range of digitally enabled attacks.

Intruders regularly gain access to sensitive data and impact key missions in public safety, finance, and national security but also manipulate data in political campaigns, alter research institution data, and impact public health security. Every day we learn about another intrusion and mass data theft.

> Why are breaches having such an impact? Part of the reason lies in how organizations secure their data behind the perimeter defenses along the border between an organization's network and the open internet.

Consider the case of the Chinese hack of the U.S. Office of Personnel Management in 2015. One of the smallest agencies of the U.S. government, OPM serves as the chief human resources agency for governmental personnel. Among other personnel duties, OPM handles the sensitive personal information of anyone who holds a position involving national security or law enforcement, from the federal courts to the Defense Department.

In 2015, OPM repelled over 10 million attempts per month to hack its networks. An advanced adversary broke past OPM's perimeter defenses, moved laterally throughout the internal network, and found the servers that held the nation's most sensitive data regarding U.S. government personnel. How? The intruders gained a foothold on a low-value server. Once inside the network, they began to steal credentials, eventually stealing those of a system administrator. From there they used trial and error to find the credentials required to implant malware on the "jumpbox," a key server within the OPM network that connected to many other servers across the data center. By controlling the jumpbox, the intruders gained access to every part of OPM's digital terrain.

The intruders were inside OPM's networks for months and the jumpbox held the keys to the kingdom. From there the Chinese gained access to some of the United States' crown jewels: all of the personally identifiable information for 21.5 million employees across the U.S. federal government.

The OPM hack is one of the most well-known cases of an intruder gaining open access to an organization's crown jewels by moving laterally throughout a network. But it is a common story. In 2013, a hostile actor stole over 11 gigabytes of private data for 70 million Target customers. The intruder began by conducting reconnaissance through open source reporting of Target's point-of-sale system, ran a phishing campaign against a refrigeration company contracted by Target (from which the intruder stole credentials and gained access to Target's network), and broke into a low-value server of the refrigeration company. Once inside Target's network, the intruder moved laterally throughout the data center until they made their way to a server holding mass quantities of customer data.

Like OPM, the Target intrusion could have been limited had Target implemented micro-segmentation across its data centers and cloud environments. Similar stories play out in every instance in which an advanced intruder breaks into an insecure cloud or data center environment. The 2018 hack of the Singaporean healthcare provider SingHealth involved a nearly identical problem, and the attacker gained access to a treasure trove of data.

The Call for Micro-Segmentation

At its most basic level, the goal of micro-segmentation is to put walls around vital applications to segment them away from the rest of the cloud environment or data center (and therefore to put some distance between an organization's vital applications, its "crown jewels," and the open internet). Cybersecurity is partly a statistical problem for the defender. A government organization like OPM has to have its perimeter defenses set to defend itself correctly millions of times per month and hundreds of millions of times per year. Yet an intruder only has to get it right once to break in and gain access to an organization's crown jewels. Micro-segmentation assumes that at some point you are going to be breached. It establishes an internal defense to prevent breaches from spreading.

Micro-segmentation provides a deep foundation for cyber resilience within a suite of cybersecurity investments that an organization can make, from multi-factor authentication to malware detection to encryption. Installing micro-segmentation software on key enterprise applications improves their security posture, but for critical infrastructure, it also improves the overall cybersecurity and health of the nations thatit serves.

> Securing the perimeter is not enough. Today organizations need to be secure beyond breach. That's what micro-segmentation is about.

This first chapter explains the benefits of micro-segmentation for companies and countries, describes how it helps keep intruders from gaining access to critical data, and recommends that companies take the next step in their cybersecurity journey by securing their interior. From this point forward, the book explains how organizations can implement an effective micro-segmentation strategy across their network enterprise.

Protecting the crown jewels

History shows that it is not a question of if but when an intruder will break through an organization's network defenses. This is what people mean when they say "assume breach." Security capabilities like multi-factor authentication and firewalls help keep intruders out by securing the perimeter and closing off points of entry wherever possible. Perimeter defenses and internal analytic tools won't help secure an organization if an intruder breaks in, however, and absent an internal defense the intruder will move laterally throughout a cloud environment.

Prioritization matters for any effective security strategy, but especially when it comes to protecting an organization's most important data. Consider the analogy of a country. Within any nation-state, some organizations matter more for national security than others; public health, safety, finance, energy, and military organizations often fall under "critical infrastructure" that deserve extra cybersecurity protections. Since 2012, the United States government has regularly conducted an annual survey to identify the most cyber-vulnerable organizations in the country, and those organizations fall onto a designation known as the "Section 9" list.

By analogy, every organization has its "crown jewels" within the information technology and data infrastructure that are vital to the organization's overall mission. For OPM, the crown jewels were the database and data for the national security community of the United States. In the United States' nuclear enterprise, they could be the data that underpins national communications and command and control to maintain deterrence and ensure stability. For Target, the crown jewels were the database that held credit card information for 70 million customers. The security of all this vital data can impact the well-being of organizations and countries, so it needs extra protection in case perimeter defenses fail.

How does it work?

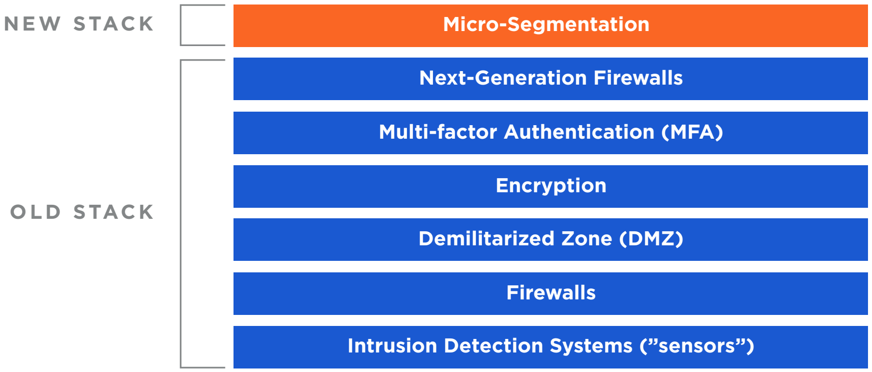
An organization begins the micro-segmentation process first by identifying its "crown jewel" applications – the applications most important to the organization's overall missions and security – and then mapping how all of its applications and workloads interact within a data center or cloud environment. The process of identifying the crown jewels focuses an organization on its priorities; an application dependency map shows all of the interconnections between applications. A strong micro-segmentation strategy then sets policies to govern interactions between applications.

The effects are threefold. Security teams know what matters most and can visualize how applications interact through an application dependency map that can be augmented with vulnerability data. And, most importantly, if an intruder exploits one server by phishing someone on the marketing team and tries to move laterally toward a server that holds customer health information, a successfully segmented network will stop the intruder in their tracks and prevent them from gaining access to the crown jewels.

Micro-segmentation is not a be-all and end-all cybersecurity solution. There is no such thing.

> Firewalls, intrusion detection systems, intrusion prevention systems, multi-factor authentication, and encryption all comprise elements of the "old" security stack. The addition of micro-segmentation forms a "new" security stack to minimize the impact of a breach.

Micro-segmentation provides a baseline, a foil against vulnerabilities, and a final defense in the event that an attacker gets through. It can be operationalized within preexisting network infrastructure. Good micro-segmentation works for on-premise servers, clouds, containers, and data centers.


NEW STACK
Micro-Segmentation
OLD STACK
Next-Generation Firewalls
Multi-factor Authentication (MFA)
Encryption
Demilitarized Zone (DMZ)
Firewalls
Intrusion Detection Systems ("sensors")

The Case of OPM

So how would micro-segmentation or micro-segmentation help an organization withstand a breach, regardless of the adversary's strategic intent (data destruction, manipulation, or theft)? Let's take the case of OPM. The above narrative reveals a number of sequential problems: an open vulnerability, insecure credentials, lack of internal segmentation, extended dwell time for the attacker to operate undetected, and, after discovery, challenges in determining whether and how the intruder and their tools have been removed.

A good micro-segmentation strategy can help address each of these sequential problems.

- **Manage open application vulnerabilities:** A strong micro-segmentation product uses host-level vulnerabilities to create a vulnerability map, add network connectivity, and display a quantitative risk measurement. Segmentation can be a compensating control for any inability to patch.

- **Mitigate risk of insecure credentials:** Although micro-segmentation doesn't take the place of identity and access management, when systems are segmented, tools can identify and block network pathways between systems that are not expected to communicate – including authentication traffic. Because the whole data center has sensors on each server, any unexpected traffic automatically triggers alerts and alarms.

- **Prevent lateral movement:** A lack of internal segmentation enables lateral movement. A robust platform should segment applications and prevent lateral movements from occurring.

- **Detect unauthorized access:** Data center alarms should make it difficult for an intruder to send port scans, conduct reconnaissance operations, or violate segmentation policy without triggering alerts. This decreases undetected dwell time.

- **Contain breaches:** After a breach it can be hard to determine whether an intrusion has been contained. An application dependency map shows traffic flows and maps traffic against internal security policies. In the event of a breach, segmentation can be tightened on a per-system, application, or environment basis as needed to contain the breach. If a service has been compromised, segmentation can be used to turn off a service or services instantly, in either a limited or bulk fashion. A segmentation policy violation should also elevate breach management notifications quickly to the security operations teams.

Good segmentation assumes vulnerabilities and sets alarms and controls to manage breach. A segmented network can prevent hackers from moving laterally. Micro-segmentation helps you meet your security goals. It can also help you stay ahead of the regulatory environment and meet your compliance obligations.

The evolving regulatory environment

The regulatory environment is changing to impose strict breach management obligations on companies. In 2018, cybersecurity and privacy regulations increased in both the United States and Europe with the passage of Europe's General Data Protection Regulation (GDPR) as well as Colorado's and California's strict state cybersecurity laws. New York's Department of Financial Services (DFS) enacted strict cybersecurity regulations on the banking sector, insurance companies, and other institutions that fall under its jurisdiction. Increased sector regulations have a trickle-down effect on contracting organizations, and audits have increased in depth and frequency for breach management. European regulators have required segmentation in directive legislation and European Union states are now adopting that legislation within their own standards, including France. U.S. federal directives are also increasing their focus on segmentation in a recognition of the unique security role that the technology can play if implemented correctly.

Some ask: but will this work to alter the cybersecurity landscape? We know we can change the game through smarter internal security. A historical example is the early regulation of the payment card industry. In the '90s, the world suffered increasing network breaches for payment cards. As a result, the industry came together to create the Payment Card Industry Data Security Standard (PCI DSS), an information security standard for organizations that handle branded credit cards from the major card schemes. The standard was created to increase controls around cardholder data to reduce credit card fraud – and it worked. With PCI compliance, breaches dropped. Over time, smart regulations regarding internal data center security should facilitate a decrease in breach impact.

Conclusion

The purpose of cybersecurity technology is to help humans manage their cybersecurity risks with greater ease and effectiveness. Attackers always find vulnerabilities in code and exploit human weaknesses. After a breach, while forensics and analysis need to identify what went wrong, to over-emphasize specific vulnerabilities or instances of human error misses the larger strategic opportunity.

> No perimeter is perfect, and even the best-trained teams cannot keep an intruder from moving throughout a cloud environment if the house has no alarms and all the doors have been left open. Organizations need to secure their data centers from the inside.

Micro-segmentation adopts an adversary-focused mindset and plans for breach. All it takes is one foothold to gain immediate access to an insecure data center. When perimeter and user-focused defenses inevitably fail, critical infrastructure companies and organizations across the globe require a robust internal defense system to stop intruders and withstand attacks. A resilience-focused strategy will make the difference.

So what are the steps required to ensure success? That is what this book is about. The first step is to have strong security leadership within your organization. Good leaders do not lead from a management standpoint alone, but from a standpoint of storytelling and cultural change.

Leadership hinges on storytelling. On a leader's ability to tell a compelling narrative about how to move the organization forward toward a meaningful goal, a story about the need for security within an overarching corporate culture. A story that places each employee within a broader strategy of positive change. Good leadership builds a strong workforce culture. Culture can then set a parameter of behavior – for security, for innovation, for creativity.

Micro-segmentation can be a challenging undertaking and requires proper planning but once operationalized it provides an underpinning of security. If the first step in the project is implementing strong leadership and cultural change, the next is having an organization that is open and willing to transform, which is the subject of the next chapter.

02 Preparing Your Organization for Success

MATTHEW GLENN

When you assume breach and consider micro-segmentation as a control that you want to adopt, it is important to note that this cannot be done in absentia of other individuals and groups within your organization. This chapter describes the different people that may need to be involved in two different phases of implementation.

In phase I: Identifying Initial Target Applications, the organization identifies the applications and infrastructure that they want to protect. Phase II: Engineering the Solution is the implementation stage, with a focus on the segmentation process. The same people do not need to be involved throughout the process – in fact, some team members may come and go on the project. You (and your organization) need to be comfortable with this fact and set the proper expectations for the people involved.

For instance, the accounting team may need to be consulted during the period in which the organization identifies critical applications. Accounting may know which applications are used to generate and recognize revenue but may not be involved with the actual job of implementing segmentation – nor in the ongoing effort to make segmentation part of business as usual. Meanwhile, the Linux platform engineering team might take over after the applications that need to be protected are identified and might be involved throughout the micro-segmentation program.

Chapter 4 discusses how to avoid "boiling the ocean," or, how to make sure you get the micro-segmentation program up and running. The key to not boiling the ocean is identifying first desired outcomes and creating a plan that allows those outcomes to be realized.

The question is how to avoid the temptation to do everything on the segmentation journey. Applications naturally sprawl and may live in existing data centers and public cloud environments. The answer is to get the right people involved in determining your first desired outcome – that is, what you want to secure beyond the breach. That's why we begin with phase I.

Phase I: Identifying Initial Target Applications

To determine the first cybersecurity goals of your organization, it is critical to get the right people in the room. Since no single person can contextualize the entire application ecosystem and understand which applications are the most business critical, it is good to get team members from each department to a meeting (or series of meetings) to find out which applications their respective groups truly "can't live without."

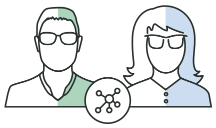
In some organizations there have been past efforts to identify critical applications. That information should be brought to the working group; starting from scratch may be unnecessary.

First, build your tiger team. Appoint a tiger team leader who has core people skills: they can influence and win over others, build alliances, dig into research, and drive change with authority. If this person doesn't already have position authority to drive change, a senior leader should empower them with this authority.

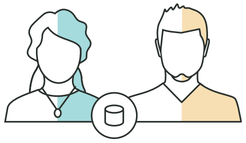
Security Team

Application Team

Engineering Team



Network Team



CMDB Team

Operations Team

The tiger team in phase I should include the person who will eventually own the micro-segmentation service for the organization (see description in phase II) as well as representatives from these departments:

- Sales
- Finance
- Engineering
- Operations
- Marketing
- IT / Security

It is critical that key stakeholders appoint accountable people to the tiger team.

Tiger team members should work with others to rank the organization's applications on a scale from most to least critical or non-critical. As representatives respond, you will quickly see that many applications are used by multiple groups within the organization – and while one group may view an application as critical to their function, others may see it as low priority. At the very least, the organization will learn how different applications impact different groups.

Once the applications are listed and identified with regard to criticality, you need to assess the ranking – the challenge with identifying and protecting the initial applications will be people and process, not technology.

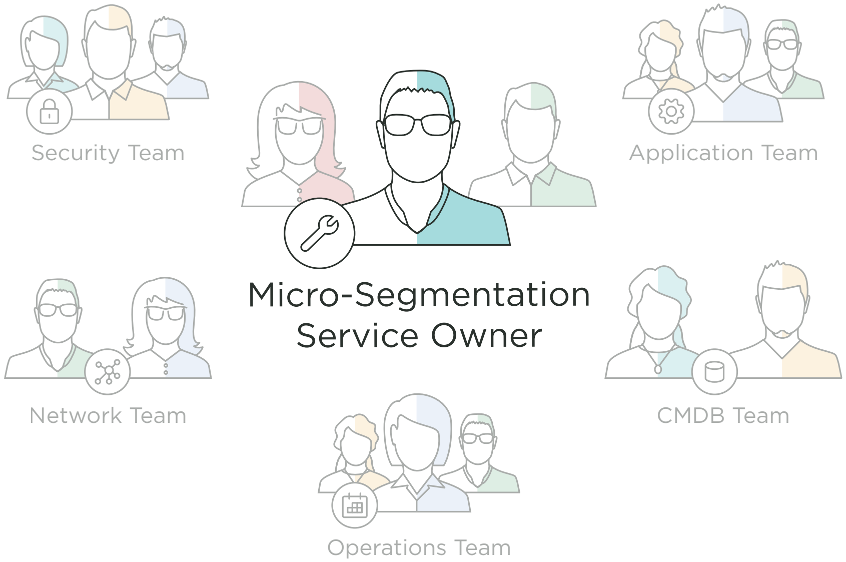
Repeatable processes will be vital to the program's success. Ensure that your organization can make the internal process changes required to protect the applications. For instance, make the tiger team choose the top thirty applications to prioritize for segmentation; any additional applications can be protected after your organization builds out the micro-segmentation process for the first tranche of applications.

Remember, the journey to micro-segmentation is not principally a challenge of technology; people and process determine the organization's success.

Phase II: Engineering the Solution

Choose a project lead

Once the initial target applications have been identified, a senior executive or executive team will need to identify the person who will serve as the long-term micro-segmentation service owner. The owner may be the same person who led phase I and was on the tiger team. Multiple people can succeed in this effort. So what are the critical skills and components?



Given the different groups involved in the segmentation effort, the leader must:

- work across different functions;
- have a proven track record of driving initiatives within the organization;
- understand how the organization works;
- prioritize and make strategic decisions.

This person is likely to have the most critical role in the segmentation effort and should be someone in IT infrastructure, network engineering, or security. Most important are leadership skills to build partnerships and alliances across the organization as well as technological acumen to make decisions about tradeoffs.

Let's get into the range of teams who will be involved in the micro-segmentation effort throughout the process and into sustainment.

- **Application owners and teams:** These teams know how their application(s) communicates and which workloads are part of their application(s). These teams also know about applications that are being redesigned or re-platformed, which may be a way of getting ahead of segmentation so that policies can follow the application development lifecycle.

- **Core service engineers:** This team runs "core services" like Active Directory, NTP, Syslog, DNS, and other critical systems within the organization. Because most applications use these core services, core service engineers need to be involved early to identify the workloads in the services they provide. They attest to the workloads that are part of their applications. They are involved again during the policy development phase, when organizations opt to segment core services.

- **Network team:** Traditionally involved with network segmentation, this team needs to understand how the micro-segmentation effort will impact them.

- **Configuration management database (CMDB) team:** All micro-segmentation solutions on the market use tagging or labeling for writing policy. Ultimately those tags and labels should come from a canonical source of truth. One recommendation is to use the CMDB as this source, which means that the solution should synchronize with the CMDB around workloads and their tags and labels.

- **Security team:** This team receives alerts from blocked traffic and must be involved with onboarding the segmentation solution, so it is critical to get the security team involved early.

- **Security operations center:** In the end, the micro-segmentation solution will be passed on to operations. Therefore, a set of workflows need to be developed in the security operations center (SOC).

Identify applications to be segmented

Once the leader has been identified, it is a good idea to get the application engineering team into a room to conduct analysis of the applications that will be segmented. Application engineering and the owner of the micro-segmentation project should put them into three categories.

Category 1 applications will not be re-platformed or updated any time in the near future. These are the truly brownfield applications that will require deep application dependency mapping (as discussed in chapter 5).

Category 2 are those applications that are set to be re-engineered, or new versions that are set to be delivered. These applications can be handled by including segmentation in the application development lifecycle.

Category 3 applications will be re-platformed; that is, moved from IaaS to containers or moved to the cloud.

Define roles and responsibilities

Once the categories of applications have been listed, the next step is to gather the different teams and organize the effort.

For category 1 applications, the critical teams to involve are:

- **Application service owners, application developers, and application security** – responsible for ensuring that applications are running and available at all times;

- **CMDB team** – responsible for maintaining the metadata (labels and tags) that will be used to write segmentation policies;

- **Core service engineering** – run Nagios, Active Directory, NTP, and other services that most (if not all) of the workloads within an organization use;

- **Platform engineering** – primarily responsible for Linux and Windows engineering.

Hold a kick-off meeting to introduce the team to the micro-segmentation project and inform the key stakeholders what roles they will have in implementation. Here are examples of roles and responsibilities, which vary by organization:

Core service engineering

- Be the first team involved in the program.

- Early on, ensure the systems that support all workloads (Active Directory, etc.) are classified correctly within the CMDB.

Application service owners, application developers, and application security

- Attest to the workloads that are part of their application.

- Observe and approve flows within or between application instances.

- Possible involvement in policy development depending on the granularity of the policy.

CMDB team

- Approve and reclassify the workload classifications within the CMDB.

- Secure the CMDB (because it will be the canonical source of metadata truth).

OS platform engineering

- If a solution is chosen that uses the native security controls found in the operating system, buy in early because the host-based control will be used widely in the data center.

For category 2 and 3 applications, the critical teams to involve are:

- **Application developers and application security –** early collaboration needed to get new versions of applications launched with policy

- **DevOps team –** so that new applications are launched with policy in place rather than using application dependency mapping (DevOps will integrate the segmentation solution into their workflows and ensure that new workloads and applications launch with the required levelof security.)

- **Any team** that is delivering a new platform that will support the new application (cloud engineering, container engineering, etc.)

Category 2 applications are easier because they do not reside in production and are actively being built. These "new" (or new versions of existing) applications make it easier to derive policy – and will significantly help the organization get policy into the development lifecycle.

Organizations that want applications ringfenced when they launch build policy through the application development lifecycle. For instance, a developer requests containers or virtual machines for version 2.0 of an application. When that application launches, it inherits a default policy wherein all of the workloads can communicate with one another, but the containers and VMs cannot communicate with the outside world. All of the workloads in the application receive a default set of policies such that they can use NTP, Active Directory, and core services, but all workloads within the application are "ringfenced" when they launch.

Then the front end of the application is opened to proxies or subnets where users reside.

Any inter-application traffic is approved by security governance. Usually in these cases organizations only want to be involved in inter-application traffic. Many organizations allow inter-application traffic within the development environment to be approved by default, but inter-application traffic within production must be approved manually. By categorizing applications and adopting a strategy that incorporates a strategic view into critical applications and those applications that are being re-platformed, an organization can segment their most critical applications.

Conclusion

The top five errors that organizations commit while on their segmentation journey:

1. **Not getting teams involved early.** Segmentation requires participation from different groups in a company. By not involving them early, the program will be met with resistance.
2. **Boiling the ocean.** Trying to segment all applications at once often results in no outcomes. By focusing on people and process early, the organization has a higher probability of long-term success.
3. **Not tackling core services first.** Core services are those applications that all applications talk to or use. Failing to identify core services in advance makes the process of segmentation difficult to achieve.
4. **Not getting CMDB teams involved.** Ultimately, policy will be written using tags – those tags should come from a CMDB.
5. **Not looking at upcoming applications.** By positioning segmentation early in the application development lifecycle, an organization can segment their applications without having to build large-scale application dependency maps.

This chapter has identified the key steps that organizations can take to prepare themselves for success in the micro-segmentation project. Leadership matters most, as people and process are the greatest challenge along this journey. Once the organization is prepared, technology solutions flow far more easily.

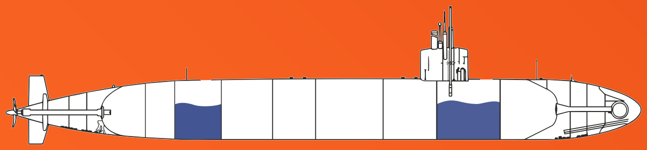
The next chapter explains the key role of metadata in your overall micro-segmentation strategy. The control of metadata underpins every part of the project. Once you have organized your teams, you attack metadata – and from there you can map your environment, begin to implement security and make policy decisions, and sustain your project over time. Absent control of metadata, none of these important security steps are possible.

03 The Green Pill of Metadata

RON ISAACSON

Micro-segmentation is all about preventing lateral movement throughout your data center and cloud environment. Ultimately, that's how you protect your endpoints – by controlling the environment so that breaches do not spread to other users.

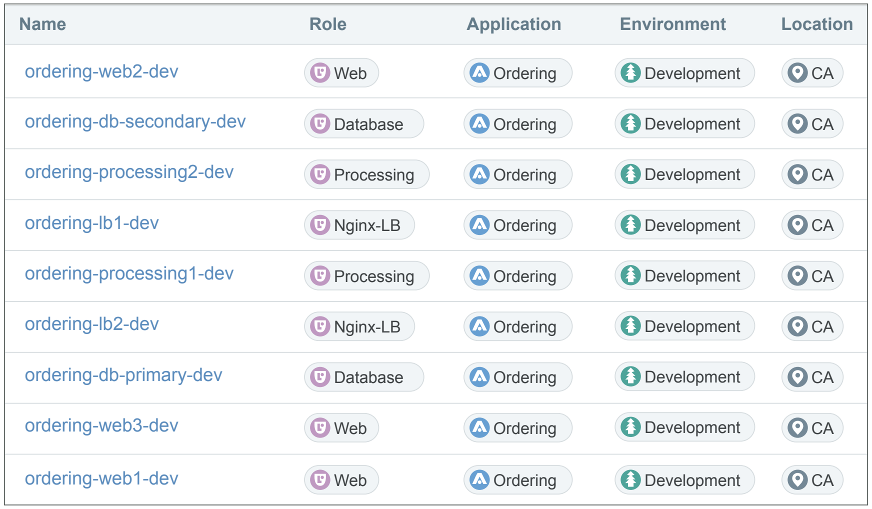
Consider the metaphor of a submarine. If your perimeter firewall is the pressure hull and your internal network firewalls are the bulkheads, micro-segmentation lets you put a watertight seal around every single person, compartment, and object on your vessel.



Micro-segmentation gives you the power to apply tailored security policies to every server in your data center: your ordering servers can connect to your processing servers, but your payroll servers shouldn't talk to either of them. To craft and enforce a policy like this, you need to know which servers belong to which of those applications. This brings us into the world of *metadata*.

What Is Metadata?

Metadata is the information about your servers that you use to make security (and other important) decisions. In a typical enterprise, metadata might include things like: what application is running on each server, what role or function the application performs, where the application is located, and whether the application is used for development or production.



Name	Role	Application	Environment	Location
ordering-web2-dev	Web	Ordering	Development	CA
ordering-db-secondary-dev	Database	Ordering	Development	CA
ordering-processing2-dev	Processing	Ordering	Development	CA
ordering-lb1-dev	Nginx-LB	Ordering	Development	CA
ordering-processing1-dev	Processing	Ordering	Development	CA
ordering-lb2-dev	Nginx-LB	Ordering	Development	CA
ordering-db-primary-dev	Database	Ordering	Development	CA
ordering-web3-dev	Web	Ordering	Development	CA
ordering-web1-dev	Web	Ordering	Development	CA

The metadata about your workloads might be stored in a configuration management database (CMDB), a repository built for this purpose. Or it might be in a spreadsheet. Maybe the metadata isn't written down anywhere, but your servers follow a naming convention that helps identify it. In a small organization, you might even know all the metadata by heart.

The choice of storage depends entirely on the organization's size, budget, and capabilities. Large organizations need a CMDB product of some kind but it is a significant effort on which they may spend millions of dollars. The CMDB is not a prerequisite for every organization. If you are a smaller organization with just a few hundred workloads, keeping your catalog in an Excel spreadsheet can work as long as the catalog is well maintained. No matter where you keep it, storing and maintaining up-to-date metadata is key to understanding and protecting your environment.

Uh oh. That could be a problem.

"Well, I guess that's the end of that! If detailed metadata is needed for micro-segmentation, then I should probably quit now."

> If that was your first reaction, you're not alone. If you took a poll of IT managers and asked how many could tell you exactly what every single workload does, you'd get a lot of blank stares.

Even among enterprises with actively managed CMDBs, the metadata is rarely complete or correct; somewhere between 50 and 80 percent is typical. Maybe you've promoted a server from development to production and have forgotten to update the catalog. Or an application owner decided to change what runs on a workload and didn't tell anyone. Chasing down incorrect metadata is the bane of every IT operations team.

Why Is It So Hard to Get the Metadata Right?

A better question might be: why would you expect it to be right?

Change happens. The MAC process (Move, Add, Change) is fundamental to every IT organization. With a lot of stakeholders and many moving pieces, steps are often missed. But the biggest reason metadata is so often wrong is a simple one: most organizations have no reason for metadata to be correct.

What happens if a server is misclassified? Under normal circumstances, maybe nothing happens. In the event of an outage, you might spend some time on the wrong path because your understanding of the impact is incorrect; this type of detour is generally written off as overhead cost. Nobody ever got fired for forgetting to update the CMDB.

To get high-quality metadata, you need to meet three essential criteria:

- **Incentive:** There needs to be strong motivation to keep your metadata up to date.
- **Consequence:** Something bad needs to happen if your metadata is incorrect.
- **Process:** The steps for populating and maintaining your metadata need to be ingrained into every one of your MAC workflows.

Let's talk about how micro-segmentation can help with all three of these criteria.

CRITERIA 1: Incentive

Consider the question posed earlier: why would you expect your metadata to be correct? Every piece of metadata starts with a person. It can be an application owner, a service manager, or someone who unboxes servers and puts them in racks. The information about your workloads needs to get from that person's head into your catalog.

What incentive do the people in your organization have for getting that information where it needs to go? What would make an application owner want to update the CMDB?

> The first step toward micro-segmentation is understanding your environment. You can't begin to talk about security policies until you know what your workloads are doing.

An entire chapter of this book (chapter 5) is dedicated to the process called application dependency mapping, which helps you learn enough about your workloads to participate in the micro-segmentation process.

Having good metadata will give you helpful insights into how your application works, and you will probably identify connections that you didn't even know existed. Do you have an old process that you thought was decommissioned but is still running somewhere? Are you making accidental cross-connections between your development and production environments? How about forgotten legacy applications? These are all common sources of risk, but they cannot hide from your metadata.

There are many other benefits to be gained from having high-quality metadata, extending far beyond micro-segmentation . We'll come back to that later.

CRITERIA 2: Consequence

The benefits of segmentation serve as a carrot for organizations to get their metadata in order; now it's time for the stick. To really get your metadata in shape, there needs to be a penalty for getting it wrong. Sticks are important for driving organizations to invest appropriately and get the process right.

In most organizations, a penalty is already in place, but it's levied on the wrong party. Operations teams may struggle to respond to outages or compliance teams may have a hard time meeting their reporting obligations, all because nobody is quite sure what each workload is doing.

> There needs to be a clear correlation between actions and penalties, and they need to be aligned to the appropriate teams.

For example, penalties are rarely felt by the application or server owners, who are the only people empowered to clean up the metadata. Therefore, imposing penalties on application and server owners may be a solution to consider.

Remember that micro-segmentation is for security above all else. The goal of your segmentation project is to reduce risk by preventing unauthorized connections. Before you can claim victory, you need to enforce restrictions that stop those connections from happening in the first place.

Micro-segmentation is data-driven at its core. In a successful segmentation project, your security policy is based on your metadata. But to be successful in the long term, your micro-segmentation program must also be adaptive (i.e., able to respond to changes in your environment). Writing a bunch of static rules isn't going to cut it.

Have you spotted the consequence yet? If your allowed connectivity is based on your metadata, and your metadata is wrong, then your application won't be able to make the connections it needs to function. Incorrect metadata leads to a non-working application. To make sure your metadata is always correct, use it in your security policy to ensure that your systems can't function if something is wrong with it.

CRITERIA 3:
Process

Finally, maintaining your metadata needs to be like brushing your teeth in the morning.

> Introducing this routine into your culture can be surprisingly easy, as long as your stakeholders know what's expected of them and how to accomplish it.

Stakeholders should be able to complete these tasks with limited friction, and organizations can make the process easier for them to complete.

Sophisticated CMDBs often have delegated administration, self-service, and data flows to and from other systems. Application owners can make direct updates to the metadata for their workloads; the procurement system feeds directly into the CMDB so each new asset is cataloged before it hits the loading dock. However, few organizations are this streamlined.

A more common approach is to have a central administrator or team who maintains metadata responding to change tickets or requests from stakeholders. The metadata itself might be stored in a spreadsheet or simple database. In some cases, your segmentation software might even double as your catalog of record.

The exact mechanics aren't important. The key is that you have a single repository where all of your metadata is stored, an easy-to-access process for keeping metadata up to date, and an understanding throughout your organization of how to invoke that process when needed.

If You Build It, They Won't Come

> One key mistake many organizations make is treating the metadata problem as a technology problem. You can build an excellent CMDB that's user-friendly and loaded with features, but that won't get anybody to care about the content that the database holds.

Successful organizations treat the metadata problem as a *data* problem. Do you have someone who has been in your organization for a long time and knows about the systems and the people? Someone who's motivated, and maybe a bit of a stickler for things being complete and correct? That person might be a good choice to lead your metadata charge.

Investing in the guardianship of the data and being zealous in the pursuit of an accurate catalog is the best way to get a successful result from your metadata program.

Additional Benefits

Let's say you're convinced. You kick off your micro-segmentation program in earnest. Recognizing the importance of keeping high-quality metadata, you put processes in place to make it easy for stakeholders to update their metadata, and you use a metadata-driven security policy to force their hand. You're done, right?

Actually, you're just getting started. As a side effect of running a successful micro-segmentation program, you now have a complete and accurate catalog of your workloads. This by itself is highly valuable and can be used for many other purposes.

What could you do with guaranteed accurate metadata? Here are some examples:

- Highly accurate reporting on the state of your environment
- Automated monitoring or alerting
- Improved response to risk and security issues
- Quick identification of anomalies or trouble spots in your plant
- Better accountability by application owners for what happens on their workloads

Chances are you'll find your own way to benefit from this trove of insights, and it wouldn't have been possible without micro-segmentation.

Conclusion

Metadata is at the heart of every successful micro-segmentation program. Few start out with correct and complete metadata about their workloads, but that should not be a deterrent. A high-quality catalog is within reach, and with it, a data-driven approach to security that will also benefit your organization in countless ways.

Once you have control of your metadata, you can move to the most important part of the project: beginning the process of implementation. The trick to initiating a micro-segmentation project is to start where you can (which is often where you must, from an audit and compliance standpoint) and to try not to boil the ocean. Start small and achieve results that matter fast. Be methodical in your approach.

04 Don’t Boil the Ocean

P.J. KIRNER

First Principles

One single concept underpins much of cybersecurity practice, including micro-segmentation (sometimes referred to as security segmentation): the principle of least privilege. First used in computer science in 1974, least privilege is the practice of limiting a user’s access to the information required to complete their job. Another term for least privilege is “Zero Trust.” Zero Trust (sometimes also referred to as default-deny or whitelist) is an approach to security where the default stance is to block access unless explicitly authorized. Micro-segmentation helps organizations implement a Zero Trust, least privilege strategy.

The principle of least privilege applies not only to users but also to workloads and applications in the data center and cloud and to IoT and other devices that are part of the network. Least privilege also applies to services provided by the environment. An IoT-enabled smoke detector should not be able to access human resources systems to control personnel information, for example. Personnel management is not part of the job responsibility of a smoke detector – and such unnecessary connections present unacceptable risk within the enterprise.

> Micro-segmentation is simply the application of the principle of least privilege to the machine-to-machine and application-to-application traffic inside a data center.

Applications and machines should have the same need-to-know limits imposed on them as humans. Once broadly applied across the data center, this technique limits the movement of bad actors inside an enterprise infrastructure.

So once you have your teams aligned and your metadata managed, where do you start?

Where to Start

Given that the concepts of least privilege can and should be applied pervasively, the most common impediment to segmentation success is to assume we can “boil the ocean.”

As with any good security or IT project, the chances of success increase when a high-priority business need aligns with security goals. Finding this alignment can sometimes be a challenge. An example of an early opportunity for success could be identifying a single critical application, ideally one that will likely be audited and that comes with a financial or reputational impact.

> A great first win is to segment the application and block unnecessary attack paths into the application. This helps a business owner solve an audit item problem and encourages everyone to take security responsibilities seriously. It also shows how a team of people can make progress happen not only effectively but quickly.

These three benefits – support for the business, security effectiveness, and operational efficiency – are the key ingredients for a first win in any enterprise.

These significant early accomplishments build confidence across the organization and provide a foundation for the project's continued success. You can share the results of the early win and attract other business leaders within your organization to approach the project with interest. They will perhaps even self-select into the next round of micro-segmentation projects for the enterprise.

Be strategic: start with the crown jewels

You need a strategy to identify where to start. First, you need the right tools to understand who all your users are, which was discussed in previous chapters. Second, you need to have a catalog of all your information – the green pill of metadata. Third, you need a whitelist mapping of users to information that is driven by a well-defined need-to-know rationale. Finally, you need a security control that can enforce your map, the policy decision process.

Start with a survey of your digital "crown jewels," as explained in chapter 2. Many organizations have already done this work and categorized applications that are of critical value to the business. Working from an existing application helps ensure the micro-segmentation project has significant value to the business. Unless you're The Coca Cola Company, improving the security of your employee beverage tracking application probably isn't enough. So if you haven't made a complete application list, you should identify and target one of your digital crown jewels first.

If there is a critical application with security and audit findings against it, that is a great place to start. The immediate, pressing need of an internal or external audit – or a mandate for business compliance – provides ample opportunity to show immediate and quantifiable impact across the enterprise. The scope of the first deployment should not be the largest or smallest within the organization, or the most complex. Too small a scope may provide too little value and visibility, and too large a scope may introduce unachievable complexity, resource requirements, and risk. Starting with an application dependency map is a valid analytical approach for gaining insights on where to start.

Finally, there needs to be a willingness for operational change on behalf of the application owners. In enterprises, some applications in operation have been functioning for years, barely touched or assessed. The application "just works" and is reliable; owners are resistant to any sort of change or perceived tinkering. The developers of the application have long since departed, leaving scant documentation, and the operational team responsible for its health fear the instability of change on a system they don't fully understand. Such an application isn't the best first candidate.

Instead, choose a healthy application with a business need for ongoing change, such as a new version or feature deployment, or an application whose function is well understood.

In summary, the key criteria for deciding where to start the micro-segmentation project are:

- high value (a.k.a. digital crown jewels);
- mandate (audit findings);
- alignment with business needs and programs for change (e.g., new version or feature deployment).

Transformational Change

Starting strong requires the involvement of the right people. These are the key stakeholders around which the working group is built, such as the security team, the application owner and/or development team, and the system administrators who often own the operational aspects of the system. Without the support of these individuals, decisions cannot be made and key actions will languish.

Micro-segmentation is a disruptive change to the status quo, and deploying it within an organization will require a new process. The first deployment is an opportunity to build that process by understanding what works for the organization and how the various teams interact with each other. The goal is to systematize the process within this first deployment and to document early learnings to promote ease, pace, and stability for wider adoption. Process-oriented members of the team are critical at this stage.

To guide the decision of where to start, let's use the analogy of building a city. Prior to breaking ground, there must be a full survey and map of the land and the surrounding area. The impact needs to be understood; the builders need to know what they are "working with." Having a full application dependency map of the data center and various clouds enables intelligent and impactful decisions to be made easily, ensuring that each edifice is correctly built from the foundation up and remains stable no matter how many floors are added.

Simply showing application owners a full-fidelity map of the environment with all assets and traffic flows often yields one of these "aha" moments:

- "I didn't know that application was still running."
- "I didn't know those two things talked to each other."
- "That's not supposed to be happening."

That last revelation is always the most exciting. But the actual output of this map process will be that application owners can determine what their workloads are doing all the time. Application owners know their applications best – here is where we ensure all the applications are present and accounted for.

Once the initial survey of the land (building the map) is complete, the next step is analogous to building the roads, the mass transport systems, and electrical grid – foundational infrastructure to support the city. In micro-segmentation, this equates to authoring policy related to core services.

Dial-tone services are the vital core services for getting your infrastructure up and running. They allow other systems to run, such as DNS, DHCP, Active Directory, Syslog, and Chef/Puppet, which are non-negotiable components of a data center; every workload requires access to them. Starting here sets the foundation for every other application to gather an immediate view of the flows to these core business services. It's worth spending the time to ensure as much information as possible is captured at this stage, as it will provide the foundation and ease to application owners for flow attestation, removing burden, and eliminating potential confusion at later stages.

Identify core services

Secondly, identifying core service applications like Active Directory and Network Time Protocol (NTP) is similar to the planning and initiation phase of any large project.

> Early planning may take time, but it increases the pace and the likelihood of success in later stages of a project.

Why focus on core services? It is critical to invest in a foundational program like identifying core services early. If the outcome of a micro-segmentation project is geared towards financially high-value business applications rather than core business services, then segmentation can seem like a thankless task with slow progress. The project team puts significant time and energy into building an environment that is essentially seen as a utility to the application owners. Since management won't see buildings rising on the skyline, it's important to articulate the value proposition: core utilities and transport are what enable a city to thrive, even if we don't tend to think about them that often. If the electricity goes down in your city, the strong security of your financial institution matters not: nothing will work.

The gains made in the early stages for core services help each subsequent adoption area and application, increasing speed and accuracy of later deployments. By mapping the core services that all applications within an environment use, the number of flows requiring investigation and attestation by application owners is dramatically decreased, and the environment seems significantly less confusing. Fortunately, core service applications do not change frequently, so the work has a long-standing impact.

Once core services are secured, the team can turn its attention to individual business applications that require protection. Application owners and developers are typically heavily involved at this stage – after all, it's their workloads that will block invalid traffic. But application owners can be famously reluctant to allow anything onto their production systems that may impact operations in any way, even in the case of a security product that will protect those applications and the interests of the broader organization. Modern organizations know that their data centers and software are often heavily customized, so the question of compatibility and impact always looms large in the minds of IT managers. It isn't enough to prove how segmentation works within a custom-built test environment; the workloads are unfamiliar to the application owners and not subject to the stresses seen "in the wild." There has to be a way to test policy within a test-bed environment that reflects reality.

Fortunately, there is a way forward.

The "lower region" proving ground

Organizations that build and maintain critical production environments frequently have a test, development, or user acceptance testing environment. Often referred to as "lower regions," these environments resemble the interaction of production workloads, albeit in a smaller deployment, and are the areas in which software is built, modified, and tested before deployment in earnest. These regions are, by definition, more tolerant to change and less critical to the organization in the event of impact.

Here's where a production deployment of the segmentation policy begins. Environments where production applications have test versions of the same infrastructure (hardware and software) deployment give us a familiar and tolerant proving ground from which to begin protecting critical business applications. Once this stage is complete, the next step is simply to deploy segmentation software to the production environment and promote the policy using the same attestation methods used up to this point.

There is an additional benefit of mapping and protecting test, development, and user acceptance testing environments through segmentation: environmental separation.

We don't have to look far to find the stories of environments that were breached by accessing lower regions and finding a way to production because environments were either less heavily protected or were subject to more frequent change, or the avenues to production from the lower regions were not well regimented. Some of the most vivid examples of this type of breach involve developers who made changes or executed actions within a production environment mistakenly believing they were couched in the safety of a test environment. At the lower end of the impact scale, this insecurity can cause resource drain and business impact as an organization remedies and recovers the error; at worst, it can cause irreparable reputational impact and huge fines levied by regulatory authorities.

To author policy in these less impactful regions is hugely beneficial to the micro-segmentation program. It can remove the concerns around impact within a production environment and become half of the equation of environmental separation – a use case and an end in itself.

Winning Over Others for the Long-Term

At this point we are far along in the segmentation journey. An adaptive map has been built, with multiple uses other than being the "secure what you see" starting point. (See a deep dive into the mapping process in chapter 5). You have authored an infrastructure core business services policy to protect critical infrastructure for protection and attestation for all future application adoption. (See a deep dive into the policy decision process in chapter 6). The ability to build a policy against a business application has been proven, solving a security problem relevant to the business, and pushed into production.

The next step might be to repeat this success on a second application. To do so, you must have the support and interest of business and application owners, and this frequently has one solution: evangelize.

The micro-segmentation implementation is a service within an organization and can be offered as such to different lines of business. Business owners know their needs best. Some business owners may find the greatest value in visibility of their mapped connections, others wish to monitor policy violations, and others still find that their applications would benefit from complete segmentation from the rest of the environment. Selling the strategy becomes most impactful when

coupled with an active, in-house deployment and success story. This method targets the security team's customers, but that isn't the only place to spread the good word.

The opposite approach to a targeted offering is much broader and involves adopting micro-segmentation strategies as part of the application lifecycle itself. In this instance, security is built into application deployment. As new versions of applications get pushed through the lifecycle, segmentation software and policy comes along with the application through the test phase and into production. Making it part of the normal process allows the policy to be maintained and updated in lockstep with the application, keeping the application owners involved and firmly in the driver's seat. Keeping application owners accountable for the development and maintenance of the policy keeps the policy tightly aligned with the business process, helping it become part of the culture and fabric of the organization.

A final option is to mandate micro-segmentation for all newly deployed application infrastructure, ensuring policy deployment from inception and building it into the lifecycle once again. This strategy allows for progress in a "brownfield" of existing environment applications and ensures all new "greenfield" applications adopt micro-segmentation by default.

The best solution is to use both approaches simultaneously to ensure you don't face a long tail project that constantly increases in scope with every passing application update and introduction. Adapting this strategy aligns an organization with the general best practice guideline and industry movement of developing applications that are secure by design. Security is considered throughout the development of an application, not overlaid when development is complete.

Conclusion

Adopting a Zero Trust strategy and applying least privilege means committing to a journey of continuous improvements. This journey makes you secure beyond breach. It begins by setting reasonable goals, iterating, proving value, and evangelizing the success of the business. True cybersecurity is not a silver bullet, one-size-fits-all solution. It is a process of maintaining control over the environment and updating the structures – just like in a real-life city.

In order to build a city, you need to start by building your application dependency map. A map enables your security by helping you visualize your applications, see the connections between them, and then set rules and control the interior of your terrain. Without a map, you are lost. With a map, you can control your terrain and prevent the spread of breaches. Building a map isn't easy – and requires its own chapter.

05 Mapped Out: Application Dependency Maps and the Path to Security

NATHANAEL IVERSEN

Everybody wants a more secure enterprise, and enterprise customers want their suppliers to be secure too.

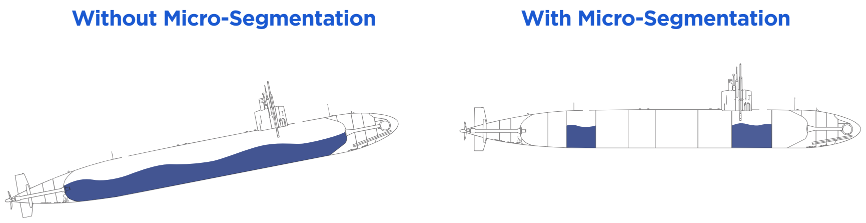
The previous chapter walked us through all the steps of implementation and how to achieve early wins. The next two chapters will drill down on two fundamental parts of a successful micro-segmentation strategy: first, the benefits of developing a mature, visual map of your applications; second, the policy decision process.

Why is it so important to have a visual representation of your application map, and to be able to see it live and in real time? Imagine that you are in charge of securing a city during a prime minister's visit from a foreign state. The first step in that process is to understand where the prime minister will stay and where he or she will visit. For this purpose you would need a map of the city, right? You cannot secure the city if you do not understand its layout.

Your applications are no different than this fictional city. You need to understand how applications and data interact across the enterprise. For most organizations, it is impossible to develop a consensus view on how applications and systems interact because they lack a comprehensive picture of the environment.

Without a map, teams see only their own neighborhood. The security team has its own understanding, the application team has a different view, and the network team works from a completely different data set. When everyone sees part of the terrain, no one has a comprehensive view of the landscape and the organization cannot make informed, timely decisions on how to secure assets. Organizations need an application dependency map to understand their environments and then must use that map to invest in security solutions to protect the crown jewels within the system and maintain command and control of the network at a segmented level.

Imagine a submarine without compartments and another one with compartments that prevent a hull breach from sinking the ship.


Without Micro-Segmentation

With Micro-Segmentation

The image of the compartmented submarine is a map itself: without a map you cannot see the ship in any detail, and without granular insight or control you cannot close off parts of the hull in the event of breach. Without a map or detailed control, your only choice is to take large-scale actions to protect significant portions of the data center; targeted assessments or tailored security changes are nearly impossible.

In short, the map will set you free.

From the Darkness to the Light

What does a world look like without an application dependency map? It's like you're wandering around in a dark and unknown land with a flickering light. This is what everyday life is like for most organizations as they face off against adversaries inside their networks.

There is a clear tension between what organizations want to achieve from a security standpoint and what they are able achieve given their current information technology stack. To get out of the dark, organizations can make a modicum of investment to map their way into clarity and security.

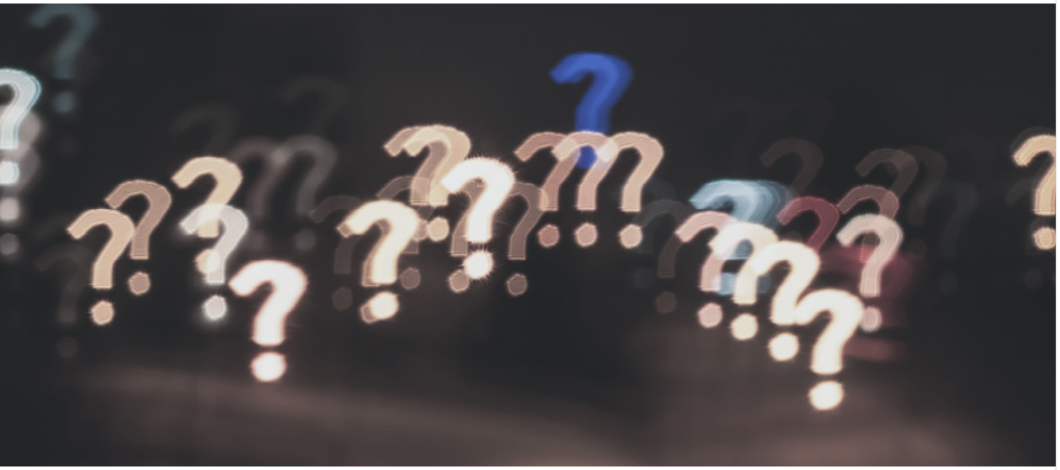
What are some of the tangible problems organizations face?

Most organizations struggle to quickly identify traffic that crosses environments (such as a development workload talking to a production workload), tie it to a specific application, and present the information in a coherent fashion without weeks of manual work. Yet that information is critical to informed cybersecurity decision-making.

In the modern enterprise, systems are deployed in physical or cloud locations and the traffic between locations is almost always handled by the network team and hardware firewalls, with coarse-grained rules that apply to hundreds or thousands of systems. Almost all locations are further divided into environments like production, staging, and development. These divisions create silos that degrade clear, scalable security operations, and in many cases those are only product developmental concepts that exist without clear network delineation between them.

The situation grows more complex when we look within an environment to identify traffic to and from a single application and show inter-application traffic. This is like being inside a cave without a map or a spelunking helmet. A security team would want to know about traffic that might remain within a server and pass within it. Teams across the organization would like to know about the scope and reach of core services.

But even with core services, there are a range of unanswered questions. Does anyone know that some development systems connect to production systems? Do connections exist to Active Directory domain controllers or critical applications? Often dial-tone services in the data center, including DNS, backup, and domain services, extend far beyond their believed borders and no one can see it all.



Life without a map is dark indeed.

Without a map, the risk of breach increases because teams are unaware of the many paths to move between applications and often don't realize where protection is needed. Consider that many organizations keep their primary systems in a data center that houses transaction systems such as IP security cameras, point-of-sale terminals, and other customer-facing technology exposed to the open internet by virtue of its function. Some of the most infamous hacks of the last decade followed this exact pattern.

No one wants an IP camera to connect to a core database or another critical system, but often they are connected and no one knows because they live without a map.

Most (but not all) organizations have a sense of their crown jewels – those applications and workloads that define the business and without which they cannot operate. Generally, these applications and systems are inspected by audit and compliance teams, and those teams want to know that critical systems are segmented from the rest of the data center population (as discussed in chapter 4). As with a treasure map, the security teams know the crown jewels are there but often cannot see the path or understand the environment within or around them. Critical applications connect and send data to many other systems scattered throughout the enterprise – and a hundred workloads can have thousands of interactions between them.

It is impossible to secure the crown jewels without an effective micro-segmentation strategy, and the first step is to build an application dependency map. If no map exists for the most important data center services and applications, it will be impossible to tighten security across the data center in the event of a breach. But with a map and the controls that a map affords, security teams can understand their terrain and have a better chance of controlling and preventing adversaries from gaining access to an organization's crown jewels once they have breached the perimeter.

Typical Outcomes of Successful Application Dependency Mapping

More good news: the benefits of enterprise-wide application dependency maps go well beyond segmentation. At first glance, the link between segmentation and application visibility is clear and obvious, but during the delivery of a micro-segmentation project a surprising variety of uses comes up.

Shared perspective

Consider the reaction of a senior vice president (SVP) of security and networks of a large enterprise who first sees a successfully deployed dynamic application dependency map. Intended as a starting point for building segmentation policy, the SVP quickly realizes the benefits of a common framework of understanding – and that common framework became an end in itself.

This SVP understands that attackers can easily exploit an organization that lacks a consensus view of how applications and systems interact.

> Even before a single line of policy is written, the map creation process forces the team to come to consensus on how the enterprise works.

As micro-segmentation operates on a Zero Trust model, the map drives agreement on all necessary communication – which helps prevents any breakage within the network as everyone sees the whole picture. For this SVP, segmentation is required by policy and regulation, but application dependency mapping drives a more fundamental security need; the entire organization aligns around what needs to be done for security and management.

Happy auditors

Often an organization begins segmentation to satisfy an audit team or a compliance monitoring requirement. This is a classic case of "inspect what you expect." Before an audit or compliance team can sign off, documentation must be provided that proves:

- the risk is understood;
- the risk is clearly mitigated;
- no one has changed anything since the risk was mitigated.

It turns out that each of these requirements can be better understood using a picture than reams of tabular data in a spreadsheet. Auditors are pleasantly surprised when they get a clear application map that shows enterprise connectivity, the operative security policy, and the knowledge that the policy has been continually enforced since it was applied. Simplifying audits makes everyone happy, from the security team to the auditors themselves.

The benefits of alarms and queries – and security

A well-mapped environment offers other operational advantages. In any host-based segmentation approach, every protected system becomes a sensor, and any attempted policy violations on the host are logged. If the map shows policy violations clearly, this feature significantly decreases the operational burden for research and investigation.

The best segmentation solutions also offer exploration tools to ask common questions about the collected traffic data. For example, operations, networking, and security teams often need to know how many systems may be using a particular service, port, or communication pathway at any given time. The whole organization runs more smoothly when these questions can be addressed quickly and accurately – and application dependency mapping helps you to do so.

Then there's the primary security function. When a breach occurs, visibility becomes vital in responding to the incident. How far has the attack progressed? What is compromised? Where are the boundaries, and how much tighter can we make our controls? A chief information security officer (CISO) will have a range of questions during an incident, such as "Can you tell me exactly where certain vulnerable ports are in use?" and "What will happen if I just turn off a particular data center service?" A team's ability to respond quickly will depend entirely on whether they have an accurate, current map of application and core data center service connections.

The Stages of Application Dependency Mapping

Application dependency maps typically develop along a three-stage journey within a micro-segmentation strategy. Each stage brings its own value to the organization. The three stages are: basic application dependency mapping, targeted monitoring, and compliance monitoring.

Up to this point, we have been talking about basic application dependency mapping – the base camp from which to ascend to other security capabilities. For most organizations, basic application dependency mapping is so significant an achievement that it can become a primary goal for the entire micro-segmentation project. It is impossible to develop a fine-grained segmentation policy without a complete basic map.

Basic application dependency mapping

Let's consider some of the specific steps required to achieve basic mapping. We will first consider the technological outcome, and then look at some of the specific benefits.

- **Basic application dependency mapping requires a clear visualization of each application, along with its internal and external dependencies.** An application dependency map shows traffic directionality, as many core services have bidirectional components that are important to separate in order to plan an optimal micro-segmentation strategy, and known internal networks are distinguished from internet and managed systems. For workloads that cannot have an agent installed, you can use NetFlow or similar network device feeds to augment visibility. Scale is important because the system will be accounting for all IP addresses and flows, not just those for which we intend to write a segmentation policy. A global picture may include demilitarized zones (DMZs), cloud environments, and multiple data centers – all of which must be captured in the map. One note: as load balancers break TCP connections, it is important to ensure that they are accommodated, and that the network accounts for underlying traffic.

- **Basic application dependency mapping incorporates metadata labels that correspond to organizational CMDB nomenclature.** For example, the map should include plain language labels like "web servers in the ordering application" as opposed to listing IP addresses or hostnames. Metadata underpins the map, so much so that we have devoted an entire chapter (3) to the role of metadata in effective security.

- **Effective micro-segmentation requires an operating system-based agent to gain process-level information and associate it to network and service ports.** Micro-segmentation only requires knowledge of a flow – its sources, destinations, and perhaps dimensions – in terms of traffic volume. Given the goal of visualizing the entire enterprise, completeness is much more important than volume, so capturing and logging each packet, while it may seem attractive, generally is not wise and certainly is not required for a full-fidelity segmentation policy.

The benefits of basic application dependency mapping are clear. Maps help teams to see the security environment with completeness and detail. Effective mapping allows executive, security, and application teams to easily identify active segmentation policies, the nature of the policies, and the overall risk exposure (or lack thereof) between workloads and applications. Such visibility and control helps the whole team set clear, unified goals. Filtering is imperative, whether by metadata labels, policy state, or number of connections; filtering fosters visibility and allows users to hide certain flows to make sense of the map.

It's like looking at a roadmap without the broader context of what the roads connect. Imagine a map with roads but without cities, gas stations, or villages between them. Now imagine a map with those roads connecting houses, offices, and municipal buildings. With the roads, you can see how the geography connects together and how traffic can flow.

Maps also help security teams to understand and visualize OS-level vulnerabilities and prioritize segmentation as compensating controls. Most organizations scan their systems for vulnerabilities, but without understanding overall connectivity it is impossible to understand how much risk each vulnerability generates. It is invaluable to see threat feed data on a connectivity map, particularly when exposure can be measured and segmentation policies can be seen in relation to a vulnerability. Ultimately, maps help policy and host-focused teams to communicate regularly around a shared understanding of risk.

Targeted monitoring

Once basic visualization has been achieved, organizations tend to focus heavily on policy creation to achieve their micro-segmentation goals. But before policy can be enforced, it needs to be validated. Targeting monitoring gets us there.

Targeted monitoring is simple to describe: after the application dependency map is complete and policy has been created, the system allows all traffic to pass, but compares it against the defined policy. Any traffic that falls outside the policy is immediately alerted to the security information and event management (SIEM) tool for action. The SIEM is the most important correlation engine in the security stack, and your micro-segmentation capabilities will be linked into it. Organizations would typically expect zero traffic to fall outside of defined policy, so any policy violations are normally taken seriously in targeted monitoring.

Additionally, for some systems, the primary value comes from understanding behavior versus policy, not necessarily from blocking traffic. Some high-security or high-volume networks are so tightly controlled that any spurious traffic represents a serious infraction for whomever is generating this traffic. Being able to monitor and detect this traffic can be more important than blocking it because such traffic is unusual. In these cases, violations may result in human resources action against the offender rather than a situation that requires segmentation. When workload roles and permissions are explicitly defined, any variation from policy raises immediate concern and a need for action. Some organizations choose this as a permanent destination for workloads, while others consider it a temporary resting point on the journey toward policy enforcement.

Targeted monitoring as test mode

> Targeted monitoring allows an organization to observe and validate policy without breaking the application.

While traffic permitted by the segmentation policy flows freely, any new traffic observed in targeted monitoring generates an alert. These alerts can be passed to a SIEM or other analysis tools. In this way, targeted monitoring serves as a "test mode."

The proposed policy is active, but traffic still flows if there is a mistake. This allows security providers to “do no harm.”

Some applications have quarterly or seasonal traffic, and it is important know about that traffic before activating a policy that would block it. Throughout the policy testing period, potential violations can be identified and remedied even before enforcement is complete. Teams require effective logging, alerting, and event-handling operations to prevent inadvertent traffic blocks.

When first entering targeted monitoring mode, organizations can take two tracks:

- Directing all alerts to the security operations center (SOC) for analysis and remediation.
- Directing alerts to the policy development team.

Depending on the value of the application, an organization may handle applications differently, and that’s okay. After a period, alerts that were being directed to application development can be sent to the SOC. Each organization will have a policy maturity model that fits its particular operating model.

Post-enforcement monitoring and compliance reporting

New workflow and reporting requirements change the map over time after policies are set. Once the map is complete, the focus shifts from building policies to validating and reporting on the policies that are already in place; most segmentation policies are stable unless they are part of automated workflows. A typical legacy client-server application likely uses the same ports or range of ports to the exact same servers every day. On a more modern automated application, application components may come and go under the direction of orchestration software. In either case, knowing exactly what is permitted or denied is critical. Periodically you will need to prove to compliance and auditing teams that your selected micro-segmentation solution has the necessary visualization, data feeds, and query support necessary, so keeping your map up to date is key.

In many ways, this final stage is about maturing the application dependency mapping function within the organization. Normally, the map is first used primarily by security and infrastructure teams for writing an initial segmentation policy. But over time, the map’s usefulness can expand to encompass several other capabilities.

A significant goal of many deployments is to have visibility during every stage of an application’s lifecycle. If the visibility agent is installed as new systems are instantiated in the development environment, visibility begins even before a new application is configured. As the application migrates into test or staging environments, the full communication profile is already known and draft segmentation policies can be constructed with accuracy. As the application passes into production, the initial deployment will occur with the benefit of knowing exactly how the application function interacts with production core services and other applications. When it is time to move or decommission the application, it will be easy to see exactly what will be impacted.

In this way, visibility becomes a tool used throughout the organization to build, test, deploy, manage, and maintain application services.

As the mapping function is accepted and employed by multiple teams, most organizations find that friction between operations, security, networking, and application teams is reduced. With fully deployed application dependency maps, each team can independently verify and ensure that any protected application has the correct policy and that it has not been disturbed.

Meetings about changing or modifying security policy occur against a backdrop of a known, fact-based map that is kept constantly up to date. When the facts are clear, teams quickly reach consensus on needed actions, possible consequences, and remediation plans.

Many deployments share the map broadly, as it reduces error, increases communication, and builds trust between teams that the correct configuration is in place.

Finally, organizations find that their audit and compliance functions appreciate the value of application dependency mapping. They are used to poring through reams of tabular firewall data to prove compliance, but imagine how much easier the job is with an application dependency map. The map clearly displays the active policy and whether it comes from a single policy or is inherited from multiple sources (i.e., a core services ruleset plus application-specific rules).

In a robust application dependency map, it is easy to get a quick look at the configuration logs to provide a full audit trail for any changes to that policy, and a look at the firewall logs will show any traffic that has attempted to violate that policy. With a mature application dependency mapping solution, anything that can change the policy is audited and tracked to a username or API key. Additionally, because the map will show current versus proposed changes, the map provides a way to discuss with auditors or governance functions what a proposed policy change might look like or its impact. When an application is seen in context, everyone benefits – including external auditors or governance functions.

Conclusion

If the journey of a thousand miles begins with a single step, enterprises likewise take several steps to solve their application dependency challenges. No one just wakes up and implements an application dependency map. Nor can one rely on static application dependency maps to help keep the city safe.

> For the map to be a successful part of the micro-segmentation project, it must be updated continuously.

There is a journey and a process of realization that culminates in the deployment and operationalization of application dependency maps. The result is transformative – and if maintained, maps provide a constant source of security and visualization.

Application dependency maps use rich and complete data from inside the workload operating system to help secure the entire organization. Security and infrastructure teams use the map to build policy, application owners rely on it to validate traffic in and out of their application, and compliance and audit teams use it to determine whether and how the organization is meeting its regulatory requirements. Maps help the executive team see the risks they face, how they are mitigated, and where the team can go further to tighten security.

There are three definite “destinations” for application visibility. Basic application visibility is what most people think of when they think of getting a map, but it is only the start of the journey. Targeted monitoring is imperative for policy validation and offers a way to monitor policy compliance short of blocking traffic. This can be a permanent destination for some workloads. Finally, for those workloads that end up under the protection of a fully enforced segmentation policy, requirements and visualizations must shift for auditors and compliance functions.

Once you have a map of your applications, you are in a strong position to begin to set policies to govern how your applications and workloads interact. The policy decision process demands trade-offs as it takes time and resources to set rules for every part of the enterprise. With a map, however, you are in a much better position to determine how best to secure your enterprise.

06 The Specifics of the Policy Decision Process

RUSSELL GOODWIN

Application dependency maps get your organization mapped out. To protect your crown jewels and data centers against breach, however, you need to define policy for how applications and workloads are allowed to interact. Identifying the required policy takes effort and when you begin the journey into micro-segmentation and policy decision-making, the landscape can look daunting. This chapter will build on previous chapters by walking you through the specific questions that come up in the policy decision process and helping you to see your way forward.

Breaking down the task into achievable steps can make it far more manageable. It is important to construct a plan and focus on achieving results that provide organizational value. To this end, the first thing to identify is where to start and how to most effectively deliver on the organization's objectives.

In previous chapters, we described how crucial it is that you understand the stakeholders that need to be involved in the micro-segmentation process and determine where to begin the project. Typically the policy decision process includes security teams, who define standards and best practices; business service owners, who are best placed to understand how business applications operate; and security implementation teams, who may prepare and implement policy. Organizations vary in this regard but identifying who needs to be involved in the policy decision process and calling out their respective roles is key to success and smooth running of the process.

There are two fundamental approaches to the policy decision process: the strategic and the tactical. Both hinge on people and process over technology. By working with the key players to set policy and make change, you can achieve lasting security for your crown jewel applications. It requires a strategic approach to look across your enterprise and smart tactics to both win support across the organization and control your environment.

Strategic Approach

A strategic approach to a segmentation project makes the most long-term sense.

> This means planning the deployment path with a focus on integrating the solution into business processes and integrating the technology to maximize visibility and minimize business risk.

You want to start making policy decisions where it is easiest to implement segmentation and where the perception of risk is lower, as explained in previous chapters. Once you get some initial applications completed, other service owners can see and understand the benefits. To set policy and achieve segmentation, it is best to pick a representative application (or a set of representative applications) and proceed through non-production instances to pre-production and finally to production in a controlled manner. You can then operationalize micro-segmentation and are ready to turn this into a mainstream process.

Much of the heavy lifting will be done in the first few applications, and this approach allows the organization to plan, deploy, learn, and improve. The first application will require the foundational infrastructure to be in place, the necessary integrations to be built in, and a segmentation policy to be developed in order to gain controlled access to your core infrastructure. Once complete, all subsequent applications will inherit all of the work. Therefore, subsequent applications do not have to revisit setup and initial policy development, which allows you to get on with the job of authoring business application policies efficiently and without distraction.

Tactical Approach

The strategic approach is an effective one and likely how you would prefer to run any project. However, often the organization is pursuing segmentation because of a compliance deadline, audit requirement, or known risk that needs mitigation within a defined timeframe. In such instances, many organizations are deploying micro-segmentation on their most business critical assets and on an aggressive timeline.

> Tight timelines provide an opportunity.

An urgent requirement to deploy a solution focuses the organization, gains senior leadership support, and facilitates fast progress. Once the technology is deployed and proven in the most critical parts of the business, it takes many concerns and objections from other business service owners off the table. The technology is not difficult to deploy or use as there are no changes to topology or infrastructure. The main challenge is overcoming organizational inertia. A strict timeline with strong executive support can overcome this inertia and enable meaningful change.

But where do I actually start?

Assuming you have decided on your approach and which services or applications you will tackle first (as discussed in chapter 4), you will still feel a sense of uncertainty about where to begin. Fear not: there is a well-trod path to take.

It starts with a plan.

Your first step is to build a project plan that contains steps for design, implementation, and validation. Planning is key to success, and labeling and policy design are central to the process.

Because you are using metadata to drive policy, your segmentation policy is closely aligned to the business logic on which your organization runs.

If designed correctly, label-based policies provide flexibility and that lower operational overhead. A well-structured, metadata-driven policy provides granular controls but with a reduced operational change burden since the policy describes the business logic – not the underlying network. In other words, servers may change and the network may change, but the policy constructs are consistent.

Once you have your supporting infrastructure configured, such as the management server and software deployment solution, you pivot to focusing on an engagement plan with business service owners to onboard applications and start collecting flow and dependency data. This information will both inform the policy detail and allow you to gain a high-fidelity understanding of the overall environment. Engaging early with business service owners and advocating for the benefits of this capability are key to avoiding later organizational roadblocks. This approach is powerful for these service owners, and with good communication you can produce a positive and collaborative process around deployment and the policy decision process.

Practical Steps for Policy Decision-Making

Once your management infrastructure is deployed for the operations team to manage (with metadata and an application dependency map) and once you've written policy for core services, you can take the next step for securing the crown jewels.

Here are the steps for the policy decision process:

1. You onboard the business application, often starting with a development or test environment.
2. You gather traffic data to identify the service flows and capture the dependency map.
3. The service owner reviews the data and confirms what is good, bad, or requires action from them.
4. Policy is finalized based on a simple metadata model. For example, "Application A consumes services from Application B." This is language the service owner and the business understand.
5. Once the policy is in place and validated, an agreement is made with the service owner as to when enforcement will be applied.
6. This is validated and promoted from non-production to pre-production to production, as needed.
7. The process is iterated in the next service.

So how long does all this take? Do I need an army to get it done?

The timeline to perform all these tasks depends on how quickly an organization is able to absorb change. Technically, the steps are quick and simple to perform and there is no need for an army of people to complete each task. It is common even in large enterprises such as a global investment bank or a multinational energy corporation for the project team to consist of a handful of people. Teams involved in the policy decision process include project management, service owner engagement, and a couple of technical staff who can review and define policy and automate tasks. The application dependency map underpins the entire process and provides the data the organization needs to succeed in the segmentation effort.

In many organizations, much of the heavy lifting is done by the business service owners themselves. Service owners can quickly understand and validate the application as long as they have access to high-fidelity dependency data from the map. Natural language policy and self-discovery of information flows means they do not need to be concerned about IP addresses and topology. You don't need to readdress servers, add VLANs, or introduce overlays or other network virtualizations to achieve micro-segmentation objectives. Service owners can take control without requiring significant knowledge of the overall network topology.

> With the correct service owner engagement and executive support, rapid progress is possible.

All this means that without making complex changes to infrastructure, teams can quickly gain visibility into the environment and set policies to contain risks around lateral movement and data leakage. Because the policy is business-logic based, it requires less maintenance and allows service owners to meet business needs quickly. This reduces operational overhead and accelerates time to market for services. This is beneficial everywhere, not just in those applications where those original audit findings were made.

Conclusion

With a structured approach, you will be surprised at how smoothly policy decisions can be achieved. The micro-segmentation process demands effort across any enterprise, as you have seen, but organizations can transform their security through diligent effort. In addition to the people, process, and technology opportunities we have outlined so far, micro-segmentation raises specific considerations with regard to cloud and containers – the subject of our next chapter.

07 Considerations for Cloud and Containers

MUKESH GUPTA

"The reason that God was able to create the world in seven days is that he didn't have to worry about the installed base."

—Enzo Torresi

Organizations today can use hundreds, possibly thousands, of applications to run their business. Some build their own applications, increasing the organizational dependency on their IT environment. On-demand compute environments (public cloud) and container-based computing seek to enable efficiency, flexibility, and speed while decreasing the need for large upfront capital outlay. These two monumental shifts present new challenges and opportunities for segmentation, as the benefits of using these services gets weighed against the constant driving need for security both within applications and across the entire compute estate. Micro-segmentation for public cloud and containers requires additional consideration since they are in a different environment than applications running on bare-metal servers or virtual machines in on-premise data centers.

Before exploring this scenario, let's first further define what these shifts mean.

First, let's consider public cloud adoption. New applications are being built "cloud first," and old applications are being migrated to public cloud infrastructures provided by vendors like Amazon Web Services, Microsoft Azure, and Google Cloud Platform. A public cloud allows organizations to bring up on-demand compute infrastructure for their applications and then destroy it when they are done using it – all without having to own and manage any infrastructure. The availability of on-demand compute allows application teams to build and deploy business applications faster, thereby enabling quicker time to market without depending on the IT team. The on-demand compute option also allows the IT teams to minimize the capital expenditures required to build and operate data centers (shifting it to an operational expense).

A second shift is driven by the adoption of container-based computing. Organizations are building and running applications inside containers instead of running them as processes inside an operating system on a bare-metal server or a virtual machine. Docker containers allow developers to deliver changes from development to production in a fraction of the time using continuous integration and continuous delivery (CI/CD) pipelines. Once an application is in a container, it can be ported into entirely different environments, on-premise data centers or public clouds, optimizing the benefits of a hybrid environment. The portability of these containerized applications also reduces the dependency on the operating systems, which minimizes the probability of breaking an application because of a change in the operating system.

Both these shifts present new opportunities when it comes to infrastructure security through segmentation. They allow organizations to settle the long-standing tug-of-war between the application teams, who are trying to build and deploy faster and faster, and security teams, who own the responsibility of maintaining the security posture of applications.

> A little bit of upfront planning yields big results.

Organizations can begin with micro-segmentation in mind as they build new applications in the public cloud and in containers because they are not subject to legacy challenges. They can also bake micro-segmentation into the software development lifecycle (SDLC) instead of deploying it later, so application teams can continue to move fast yet stay secure.

Traditional segmentation approaches present clear challenges in these fast-moving environments. Using network-based hardware devices such as switches and firewall boxes isn't possible, as they cannot be deployed across a public cloud. Hypervisor-based solutions are also not feasible, as control over the hypervisor in the public cloud doesn't exist. Similarly, multiple containers running different applications can run inside a server (physical or virtual), making it unfeasible to segment those applications using network or hypervisor-based approaches.

Public cloud providers and container orchestration systems have rudimentary solutions for segmentation. Most organizations also end up using multiple public cloud providers and still have some bare-metal servers and virtual machines in on-premise data centers for applications that are not suitable to run in public clouds or containers. It is a challenge to manage multiple different segmentation strategies across multiple platforms.

It is also important to note that while an organization running applications in a public cloud does not have to pay for the infrastructure, there are cost tipping points where it is actually more expensive to run them in a public cloud. So security policy portability becomes just as important as container portability since the application may move either from cloud to cloud or from cloud to on-premise, or simply require a hybrid infrastructure approach in which applications span public cloud infrastructure and traditional infrastructure.

A new approach, then, must be defined to meet this challenge. The goal is to enforce segmentation policies as close to the application workload as possible – with limited reliance on the public cloud infrastructure. Therefore, the operating system for applications running on virtual machines in a public cloud, and in containers for containerized applications, becomes the optimal location for visibility and enforcement.

Begin with Micro-Segmentation in Mind

> Adoption of public cloud and containers allows micro-segmentation to be in mind from day one, without the constraints of a brownfield environment.

New applications that are being built directly in a public cloud are not subject to the same limitations of on-premise data center infrastructure. So building these new applications can begin in a public cloud or on a containerized platform while conforming to a broader micro-segmentation strategy from the beginning. Securing public cloud applications demands a new way of securing applications from their inception, a way that is unencumbered by existing infrastructure and with a focus on building outside of prior specifications. The security must operate at the same speed as the public cloud environment; that is, the solution cannot slow down the organization!

Applications built in the public cloud and on containerized platforms are often dynamic and distributed. The compute infrastructure for these applications is deployed and auto-scaled up and down on demand. Developers should take the opportunity to bake micro-segmentation into these applications so that security can be as dynamic and distributed as the applications themselves. The security, based on metadata, should adapt and change to the evolving compute environment – and not slow down application delivery.

Modern applications teams use CI/CD methodologies and tools to build and deploy applications faster. Overlaying security onto a built application before pushing to production can slow the process and hinder the application. Segmentation policies should be developed as applications are developed, and these policies should simply be pushed to production along with the application. If security is baked into the CI/CD pipeline, applications can achieve their goal of moving fast and the security teams can still ensure that they are maintaining a good security posture.

Challenges with Segmentation in a Public Cloud

Network-based solutions

The nature of a public cloud causes challenges for traditional network-based approaches to segmentation. Infrastructure in a public cloud isn't wholly owned – this is a large part of its value proposition to an organization – so a network-based approach has serious limitations. A common reaction to the move to a public cloud is to try to retrofit a firewall-based approach. Hardware cannot be shipped to the public cloud so most firewall vendors have developed virtual firewall solutions that perform the same function as the hardware firewalls in a virtual form factor for deployment in public clouds. These firewalls still rely on segmenting using VLANs, zones, and subnets – constructs which are harder to replicate where the organization does not own the infrastructure.

Virtual firewalls in public cloud also become traffic chokepoints, increasing architectural complexity, reducing application resiliency, and increasing operational cost of managing the security solution. For example, a large web scale enterprise that's aggressively moving applications to AWS may decide to use virtual firewalls for securing these applications. The capacity of the virtual firewalls and the level of segmentation dictates that they use one virtual firewall for a set of 20 virtual machines. It is estimated that they will need around 12,000 virtual machines in just the first two years of their migration, which means they will have to deploy and manage 600 virtual firewalls. Not only does this design place serious restrictions on the flexibility and resiliency of the applications they were planning to deploy, but the operational cost and burden of managing 600 firewalls becomes an unwanted challenge.

Native controls in public clouds

Using native segmentation tools provided by public cloud providers is challenging because they are rudimentary, limited, and different for each public cloud provider. Many enterprises realize the challenges of using network-based firewalls in public cloud, so they choose native controls for closing the gap. Existing public cloud vendors offer products such as virtual private clouds (VPCs) and security groups (SGs) for segmenting applications in the public cloud.

The challenges presented with these products can be numerous. Most enterprises end up with either too many or too few VPCs.

Some enterprises decide to go with broad VPCs – one for development, one for test, one for production – and end up without enough segmentation within the VPCs. Others decide to go fine-grained and create one VPC per application or even per developer and end up with a management nightmare of ballooning VPC management needs. Note that modern applications built using microservices architectures are highly interdependent. Allowing this connectivity between applications running in different VPCs while preserving enough segmentation becomes a management nightmare. Creating a separate account for each application may seem like a good segmentation approach but leads to the same management problems as a per-application VPC.

Native controls such as security groups provided by public cloud providers come with serious limitations if used for fine-grained micro-segmentation. For example, as of this writing, an AWS security group can only have 60 inbound and 60 outbound rules, a network interface can only have 16 security groups, and a VPC can only have 500 security groups. Similar limits exist for other public cloud providers as well. These limits put serious constraints on how fine-grained a segmentation policy can be because the number of rules available is rapidly diminished when using security groups with fine-grained segmentation.

Consider again the number of rules against the normal observed traffic and required rules cited in the application dependency mapping discussion in chapter 5.

Regarding the number of rules to be written, in a cloud environment with more micro-servers the sigma rises closer to 1.9.

Finally, in much the same way that most organizations don't fly on just one airline, most enterprises will use multiple public cloud providers to avoid vendor lock-in and benefit from the right technology offering and price point for their needs. Using native controls presents the additional challenge of managing multiple different solutions across diverse platforms. Native controls for

Kirner’s Equation

The Number of Rules: Actual

$$R = \rho W^{\sigma}$$

$$R_{min} = 1.1\,W^{1.9}$$
$$R_{max} = 1.1\,W^{1.8}$$

Where Connectivity Size Factor $\sigma = 1.8 - 1.9$
And the Rules Per Edge Factor $\rho = 1.1$

Workloads	800	2,500	10,000
Total Rules	180K - 360K	1.4M - 3.1M	17M - 43M

each public cloud have different UI, API, functionality, and limits, requiring their teams, tools, and processes to be built to manage each segmentation solution. This leads to significantly higher operational costs and complexity from a management standpoint.

Segmentation for managed services

Managed services (e.g., object storage service or relational database services) present a unique challenge to segmentation because they run on compute infrastructure owned and managed by the public cloud provider. Lack of access and control on the compute infrastructure limits options for segmentation of these managed services. Even though most enterprises at least aspire to avoid using managed services due to specific public cloud provider lock-in, the numerous advantages of using managed services can prove too attractive a lure. These functions are provided by the public cloud providers as a completely managed service that can be accessed either via APIs or via network connections. Customers do not have access to the compute nodes running these services, and the IP addresses associated with these services can change frequently. Without access to compute nodes or the network the compute nodes are running on, traditional approaches will always come up short.

Solutions for Public Cloud Challenges

Micro-segmentation in the public cloud is critical to overall protection of owned infrastructure as cloud-based infrastructure is even more vulnerable to breaches than the infrastructure in an on-premise data center. On-premise infrastructure has the benefit of control over physical boundaries, physical servers, and networks. As the boundaries of the data center begin to blur, so does traditional control.

Consider the following as you plan your move to the public cloud.

Policy enforcement inside workloads

There is a way around a number of the challenges listed above: enforcing segmentation policies through your workloads (i.e., the virtual machines or the containers where your applications run). Imagine if you could activate the firewall built into every operating system or container and program that firewall with fine-grained rules to allow the workloads to communicate only with the workloads that they are required to communicate with for operation.

By decoupling enforcement from the actual network infrastructure, fine-grained policy is achieved within the compute without requiring access to anything except the workload itself – something that is available across all cloud providers. Because this approach is completely agnostic to where you are running your applications (bare-metal servers, virtual machines, or containers in your on-premise data center or in any public cloud), it presents one micro-segmentation solution that works for all active applications irrespective of where they are running.

This approach provides several advantages and addresses many of the challenges presented by the adoption of a public cloud.

First, it allows a single micro-segmentation solution to address all public clouds and provides the freedom to deploy applications in any public cloud. The same solution, in fact, can additionally be used for workloads in an on-premise data center running on bare-metal servers or any hypervisor – allowing for a single policy across any type of hybrid infrastructure.

As we seek to leverage the enforcement capabilities built into the operating system, network firewalls with their numerous operational components (managing capacity, availability, resiliency, etc.) becomes redundant. Using workloads as the unit of enforcement allows you to create a fine-grained and unique micro-segmentation policy for each workload without having to hairpin traffic unnaturally through network-based firewalls. This approach gives the added benefit of segmentation that goes beyond network protocol and port. As the enforcement happens inside the workload, you can implement enforcement policies based on process/service names or system account, or based on the user that's logged into the workload.

Finally, decoupling segmentation from the network allows you to design and optimize the network and VPCs in the public cloud for what the network does well: transport packets from point A to point B in the most optimal manner. Fewer, larger VPCs can be employed to benefit from ease of management, and a fine-grained micro-segmentation policy using workloads as enforcement points within the VPCs can be employed. The network can be flat and fully routable as well, reducing operational burden on the network team.

Metadata-based elastic and portable policies

One of the key themes of this book is that metadata and metadata-based labeling is essential to achieve optimal, manageable micro-segmentation across an enterprise. Just like Neo had to do in the famed movie *The Matrix*, "swallowing the green pill of metadata" is the key to success.

Here's why this matters from a cloud and container standpoint. Metadata-based policies are essential for dynamic infrastructure, which is already the dominant method for public cloud and containers, so in general metadata is more bountiful and more accurate than in a brownfield data center. When segmentation policies are built with labels and combined with the workload enforcement point, micro-segmentation can be baked into the software development lifecycle and within CI/CD pipelines. Application teams can build segmentation policies using natural language labels (e.g., "database" workloads provide a "Redis service" to the "web" workloads) and can apply them to different instances of their application in different environments and locations. Label-based policies are also portable. Application teams can build policies while developing the application in the development environment and, when they are ready to go into production, application teams can simply promote the policies to the "production" environment running in a totally different VPC or region or even in a totally different public cloud.

Policies defined with labels also provide the elasticity needed for some applications. The policy above could be applied to two database workloads and two web workloads initially, but if the demand surges and the application auto-scales to 50 databases and 200 web workloads, the same policy still applies and can be adjusted to the new application in a completely automated manner without any human intervention. The benefit is easy scaling across the enterprise and that helps achieve efficiencies without sacrificing effectiveness.

VPCs and security groups as perimeter defense

Consider using the native controls provided by the public cloud provider (i.e., VPCs and security groups) as a perimeter defense just like the perimeter defenses in your on-premise data center. This gives you another layer of security in the public cloud and, if you use it only as a perimeter control with a small number of broad policies, it reduces the number of VPCs and security groups that needs to be managed. Reducing the number of VPCs and security groups not only keeps you safely below the limits imposed by the public cloud providers, but also reduces the operational overhead of managing a large number of VPCs and rules for each public cloud provider.

Considerations for Segmentation of Containers

The next phase in enterprise transformation is to build and run applications inside containers rather than running them as processes on virtual machines. Docker has popularized the usage of containers, even though the basic building blocks of the container concept have existed in the Linux operating system for many years. In order to deploy containerized applications in production, a number of other technologies had to be built including container orchestration platforms, such as Kubernetes, Red Hat OpenShift, and Rancher; container networking tools, such as Flannel, Calico, and Contrail; and container image registries like Docker Hub.

Containerized applications bring unique challenges to segmentation in terms of both visibility and enforcement. A server, physical or virtual, can run a large number of containers that are part of different applications. The network connections among these containers may not traverse outside the server, limiting the ability of the "network" to have any visibility or control of these connections.

The future of security and containers holds immense promise – yet there are also some important steps for security teams and infrastructure teams to take as they consider how best to secure the enterprise. Planning a micro-segmentation strategy and deployment for containerized applications requires accounting for certain specific considerations described in the following sections.

There is no container island

Most enterprises realize quickly after deploying containers in production that their containerized applications don't live on an isolated island. They frequently communicate with applications that are not containerized yet or with components that never will be. A micro-segmentation solution that only works for containerized applications is both short-sighted and simply not deployable for most enterprises.

Containers are popular and most enterprises are dipping their toes into deploying containerized applications. But the current reality is that the containerized applications represent a single-digit percentage of the entire infrastructure for most enterprises. Enterprises that claim to have successfully adopted containers are running containerized applications on a few hundred or a few thousand nodes, while the total number of compute nodes (virtual machines and bare-metal servers) they own may be on the order of hundreds of thousands. This hybrid state of container and non-containerized workloads needing to communicate will be with us for quite a few years.

So what is the impact of the hybrid container and non-container world in which we live? When these containerized applications are deployed in production, they communicate with workloads that are still running on virtual machines or even bare-metal servers. For example, core services such as DNS, Active Directory, Syslog, and vulnerability scanners are not yet containerized; critical customer databases are still running on Solaris, AIX servers, or Oracle RAC Linux servers. Containerized applications often have dependencies on those non-containerized services in order to function. Segmentation solutions that only focus on containers will only address a small portion of total compute infrastructure. Buyers making security product decisions need to make this a serious consideration when making a vendor selection. This is true across product categories, not just micro-segmentation.

> Successful micro-segmentation should provide one application dependency map that works for all types of workloads, including containers, virtual machines, and bare-metal, and one policy enforcement solution that can enforce policies for all types of workloads.

Container-Level Segmentation

Even though enterprises may start by deploying dedicated hosts for running containers for a given application, they strive to achieve large farms of servers where they can deploy any container on any host. You need a segmentation solution to enforce policies at a container level if you are to segment different application containers running on a host.

Containers run inside the operating system, Linux or Windows, either on a bare-metal server or on a virtual machine. Multiple containers running on one host communicate with each other using a software switch built inside the operating system. Traffic between two containers running on a host therefore does not go out of that host to the network. For this reason, a network-based segmentation solution (i.e., physical or virtual firewalls) can't provide visibility to or control the traffic between containers inside the same host.

A host or network-based segmentation will only suffice if dedicated hosts are made available for running containers that belong to a single application. As described above, enterprises most frequently plan to move to a model where they can have a shared farm of hosts and can run any container on any node while having the ability to segment them from each other, even if they start with dedicated hosts or clusters. This container-level segmentation is not feasible using network- or hypervisor-based approaches to segmentation.

The best way to get container-level visibility and segmentation is to perform it inside the container network namespace. Moving the visibility and enforcement inside the container namespace also removes the security constraints from container networking and allows simplified design of container networking.

Routable IP to Containers

Making containers first-class citizens on the network by giving them a routable IP address and eliminating network address translation (NAT) devices is critical for container-level visibility and enforcement.

Overlay networks and NAT present a huge challenge to both visibility and enforcement for security, especially because the containerized applications frequently communicate with non-containerized applications.

For example, consider a situation in which the web tier of an application is containerized with overlay networking and a NAT and the database still runs on a bare-metal server running Oracle RAC. When the web containers connect to the Oracle database, the database only sees the translated IP as the source IP and it doesn't know which container the connection came from. The security teams in this case can't log the right source of the connection and also can't enforce policy that only the web tier of this application is allowed to connect to this database, not the web tier of another application, because the source IP that the database sees is the same.

Conclusion

Public clouds and containers offer tremendous benefits to enterprises by giving them the agility and flexibility to deliver value to their customers in a way that on-premise data centers never did. Most enterprises are taking advantage of these benefits by starting to build new applications and migrating old applications to public clouds and containerized platforms.

Adoption of public clouds and containers presents new opportunities to the enterprise when it comes to segmenting applications to prevent the spread of breaches. It gives them a clean slate without having to worry about the legacy applications in the data center. Enterprises can take the opportunity that micro-segmentation presents to settle the long-standing tug-of-war between the application teams, who are trying to build and deploy faster and faster, and the security teams, who own the responsibility of keeping these applications secure. Enterprises should begin with micro-segmentation in mind as they build this new infrastructure into the public cloud and on containers.

We have now covered the most important considerations for managing the micro-segmentation project and deploying it across your enterprise: the importance of getting your team set; the early importance of metadata for the entire project; the best way to implement your strategy and achieve early impact; a deep-dive into how to build an application dependency map and how the map will drive your success; the steps involved in the policy decision process; and the specific considerations associated with cloud and containers.

With these considerations in mind and having prepared to deploy your micro-segmentation project across the enterprise, you now need to turn to the final step in your management and project planning: sustainment. What does it take to sustain your micro-segmentation capabilities as a part of a broader cybersecurity strategy? What issues come up within your business, and how can you lay the groundwork for success?

08 Sustainment

E. JAY HUSSEIN

"We're Done Now, Right?"

So now you're deployed. You have a visible topology of workload and application communication, and may have configured segmented protection based on individual workloads, applications, asset groups, or environments. You're connected to your enterprise – the agent required is in your "Golden Image" or part of an automated deployment package so that all new workloads are plotted on the map once deployed and automatically labeled and protected with your adaptive platform.

Your SIEM tool is one of the most important tools in your security arsenal because it receives the notifications of policy violations you configured with your micro-segmentation capabilities. You may even have configured the ability to make operational health decisions with the use of vulnerability data overlaid onto your application dependency map. You are now in an advanced state of deployment and have advanced your security posture by massively decreasing your attack surface.

Success, finally! You're in an operational mode and simply need to sustain for two to five years.

But what does "sustainment" mean for a segmented environment? What are the parts of the process that need to be considered? What processes must be built, what resources do you need on the task, and how does sustainment actually work? The model for sustainment does not begin after a deployment is complete, but is designed from the decision point of implementing micro-segmentation. If you find that you design, deploy, and then create a sustainment model, you're simply doing it wrong.

> In the same way that modern software development embraces building with efficient security in mind rather than overlaying clunky security elements on top of a completed product, we must design our deployment with sustaining it as a focal point.

No modern machine operates forever without some care and attention or an efficient operational model. It would be a wonderful thing to be able to spend money on a solution or make an investment, put it on autopilot for the next decade, and never tend to it. But that's not what happens. We tune, we check tires, we reallocate assets. In real terms, we seek the input of numerous parties to keep the machine going, we set up workflows and processes up that are run manually or automatically to assess health and operation, and we enable the business by making things as easy as possible right from the get-go. That preventive approach means designing a healthy and efficient operational model before your deployment and making small but smart investments in time and effort.

How It Works

Although host-based segmentation can be seen as a way of instrumenting a host firewall for every workload owned by the enterprise, the job is not nearly as onerous as a traditional firewall management operation.

> The days of raising a ticket to the firewall operations team and waiting four weeks for implementation can now be a thing of the past.

Rules are automatically written based on natural language labels, visibility of flows, and already established policies that reside within your segmentation software. The rules at the workload adapt based on the higher-level, label-based policy that was written.

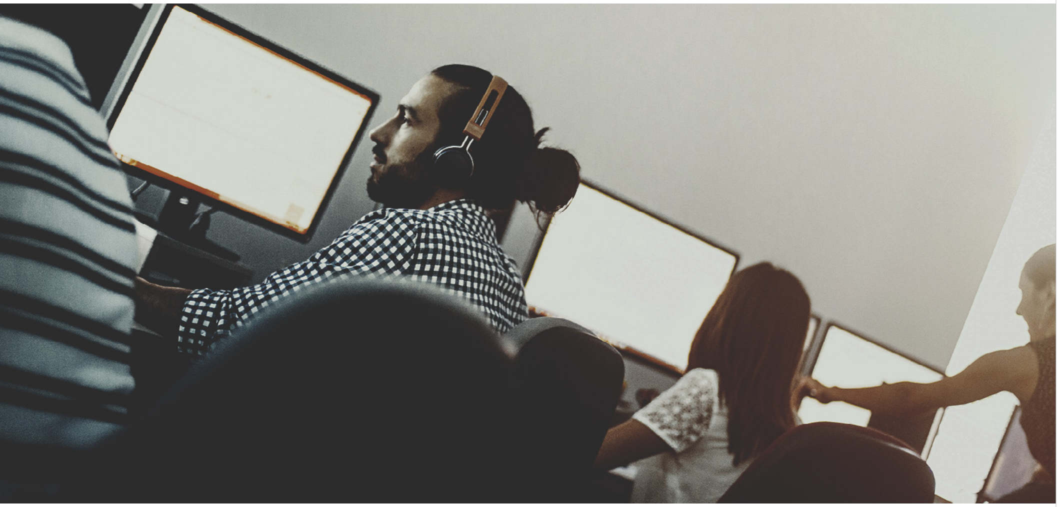
When a new workload appears as a result of the agent first communicating with the software, it is labeled and inherits the policy associated with those labels. Any attempted connections that are not part of a policy will appear within a blocked traffic report and will be reported to the SIEM. This way, if new flows and connections are required, they can be identified easily and allowed with a few clicks.

Sustainment from within the security operations center

Within smaller segmented environments, sustaining a deployment can be as simple as having a few subject matter experts or individuals trained on using the software sitting within a security operations team. Most often, the members of the firewall team who are responsible for writing access list entries on traditional firewalls are the ones who will own the software.

In an automated environment, sustaining a deployment would be as simple as monitoring the SIEM for new blocked traffic and validating it, or responding to the needs of the business by enabling policy when it doesn't exist. No new headcount or hiring of team members would be expected, as the segmented environment is simply managed by another tool used by the team. Remember, it is a management rather than a monitoring interface. Those responsible for managing the tool need only interact with it when changes are to be made that are not instrumented via automation. The monitoring is done using existing tools, and dealt with in the same way as a traditional firewall deployment.

> If the enterprise can support automation, it decreases the burden on the security operations team from a monitoring standpoint, freeing up their energy for other responsibilities.



In larger deployments, the load is a little heavier but spread across security and operational teams. The rate of change and organizational complexity will dictate needs, but we can expect a full-time headcount to be assigned as an administrator per 10,000 to 20,000 workloads managed.

Automated reporting becomes extremely important because the number of flows quickly becomes unmanageable for manual assessment and intervention. Application owners are sometimes given the ability to manage and maintain their own policy definition in larger implementations, further spreading the load and ensuring security design is built into deployment rather than externally mandated and overlaid. Who better to decide which flows are relevant than those who own the application in question?

Why It's Hard

Actually sustaining the software follows a similar model to other agent-based software deployments. Periodically, upgrades and patches will be needed for any piece of software within an enterprise, required to fix or enhance the deployment in some meaningful way. As the vendor develops and releases new features, each workload may need to have its agent upgraded to a later software version to take advantage of them. With the constantly changing landscape of security concerns, a mature deployment is accustomed to bug-fix patches with a robust patch management program for both operating systems and applications, but faces the age-old mandate of protecting the most valuable assets of the organization while not introducing further risk or stifling the business through constant intervention.

> Micro-segmentation software is an invaluable security tool, yet must be as transparent as possible to the application owners in both operation and impact.

Until they are accustomed to the operation of such a tool, it can be all too easy for the segmentation software to become the new scapegoat when things don't work. Organizations that introduce new products into their environment are all too familiar with that. The new thing always gets blamed.

Moves, adds, and changes

> To sustain an effective micro-segmentation project, adaptive policy needs to stay constantly up to date.

This is best done in an automated fashion. Workloads participating in a micro-segmentation strategy frequently communicate their status to the controller so that a fresh policy can be calculated and sent out to all workloads in the event of a change. Workloads that stop communicating, perhaps due to an outage or a controlled power down, will be removed from calculations and marked offline after a period of inability to communicate. A good micro-segmentation product will never lock down this workload; the last known configuration will remain until communication is re-established.

But what happens when new servers come online and require protection or when new communication flows are required for existing workloads? An intelligently designed and built agent will be able to install itself, pull down configuration, and pair upon boot when built into the base image of a new install. Even when being installed after the workload comes online, the agent will be delivered via a script that informs it where to find and download the software required to pair with a controller. The agent can be deployed using existing software deployment tools and models, allowing ease of administration and install effort.

Auto-labeling of the workload, via an integration with or upload of CMDB data, ensures that a workload will appear in the right application group in your application dependency map when it comes online. Core service flows then can be automatically added to the list of allowed connections within the workload, and services will be instructed to accept communication from the new workloads via the controller. A well-defined and executed labeling policy ensures that each workload, as it receives its labels, also receives the right policy to begin communication in a similar fashion to objects with the same labels. Any additional communication flows, whether immediate or added later, are visible to the administrator on their map as blocked communication and can be enabled immediately within the system.

In order to validate if a flow is genuine, a potentially blocked flow must be attested to by those who know exactly what type of communication can be expected from their workload. Generally, this is the application owner, but it could also be the IT function, if they have the rights and information for this function. To drive the process, the application dependency map can be enabled with an application owner view, limiting visibility of the complexity of the environment to only the relevant workloads and their associated flows. Application owners attest to the validity of the communication, then the flows can be added to policy by the administrators of the system.

But what happens if and when an infected workload begins to generate flows outside of policy?

> A properly protected environment needs to have a "quarantine" function that enables moving a workload, within the controller, into a state that cuts off its communication to all other workloads.

At the same time, the controller must also send messaging out to every deployed workload to protect itself from such communication. The label of the workload is temporarily altered so that it remains in the quarantined group until further action can be taken to safely secure the environment.

Controller and agent software updates

The decision to update software, at its simplest level, seeks to balance the need for features against the desire for stability – also described as the need to protect application workloads from unnecessary or undesired access. These two primary inputs can, at times, be at odds with each other. The first seeks to answer the question, "What do I need from my segmentation software?"

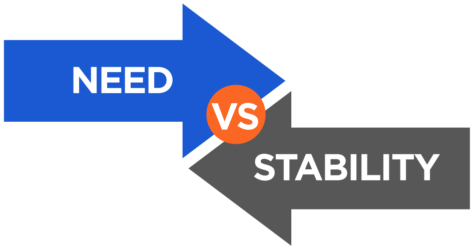
The second is a more of a firmwide discussion around how much software change is tolerated.

As new features are introduced into the segmentation software, upgrades of the controller, the agent, or both may be required to benefit from those features.

It is a good idea to develop a roadmap of what is needed from the product, and which features will be deployed when in the lifecycle of the program.

The roadmap specific to an organization is jointly developed by the consumer (customer) and provider (vendor). It is a two-way line of communication that leads to a document showing a timeline of feature release and potential consumption. Feature requests may be submitted to the vendor well ahead of time, providing the ability to plan and build to a need and ensure the extended features of the software meet the needs of the organization.

The operational stability posture of an organization is the opposing force. Most organizations will set standards around how much change they care to see within their technology deployment. Change can sometimes be seen as the enemy of stability, and the roadmap will need to provide strong business-focused justification for exceptions to standards set for technology change. Business enablers are easier to justify.



Changes driven by the IT group are mostly not. Efforts to convince an application owner that you need to access their beloved workloads more than once a year for maintenance of the same software may not be received with enthusiasm, and the scale of deployments may mean the upgrade is a much larger event than simply upgrading the console software on a single multi-node cluster.

However, in some instances, it may be impossible to avoid more frequent upgrades to agents on workloads. Some management console features require additional data or functionality to reside within the agent and therefore require an upgrade on the agent workloads before the new feature can be used. Additionally, bug fixes and vulnerabilities communicated by the vendor may require a more immediate approach and a mandatory estate-wide upgrade. For example, vendors like Microsoft commonly deploy security patches for identified vulnerabilities that impact all their operating systems.

As previously mentioned, automated deployment methods are not uncommon within modern enterprises. Automated packages mean little to no interaction from the administrator for a successful software upgrade. Even the least mature security organization necessarily has a patching schedule into which these upgrades and bug-fix version releases can be integrated. Even though micro-segmentation software seeks to deny access to connections outside of policy, an agent that loses connectivity to the management console is not incapacitated. The default behavior of a workload that can't get to the management console is to revert to last known best configuration – business as usual.

The two factors (need for features vs. stability) will need to be carefully considered when creating a roadmap. However, the roadmap is critical to the success of the deployment, as it gives the administrators the ability to provide the much-needed justification and plan well ahead of time.

The roadmap should be socialized early and frequently amongst stakeholders within the organization to ensure early buy-in, budget planning, and architecture consideration.

Conclusion

In order to sustain a micro-segmentation deployment, we must begin with the end in mind. Designing for operationalization will ensure an easily sustained model, and building software upgrades into an existing patch management program will remove the difficulty and paranoia around security software impact on an application. Working with the wider team, feature and business needs can be communicated to the vendor through feature requests for development, and the internal executive management team with the creation of a roadmap. Using these methods, a segmentation policy can be a critical tool in solidifying the security posture of an organization, a business enabler, and an informer to architecture decisions.

09 Conclusion: Building a Defense-in-Depth Strategy

In this book we have outlined the core principles and detailed steps required for an organization to implement a micro-segmentation strategy. It starts with leadership at the executive level and with hiring smart leaders to drive innovation across the organization, bringing together key teams and preparing the organization for success.

Along the way, your team needs to swallow the green pill of metadata to facilitate labeling and security management. From there you can identify your most important applications, especially those which have regulatory requirements, and begin the project – starting with early wins and avoiding the temptation to "boil the ocean" and secure all applications at once. In setting policy, you make choices about what matters most for your organization, taking a risk-based approach to the policy decision process. Specific choices need to be made regarding public cloud and containers. Finally, you plan for sustainment.

Cybersecurity can be daunting to those that don't understand it well. Success starts with a strategy. The good news is that, unlike threats in other arenas such as terrorism, organizations can take steps on their own to change their terrain, upend the adversary's map of the world, and keep intruders from accessing the most important data.

The cybersecurity story has evolved over the last decade. It moved from a limited domain, affecting only coders and computer scientists, to impacting all of us. As the threat has grown worse, smart people have run toward solving hard problems. Top-tier talent have entered the field; teams have developed and learned how to operate in complex environments, from security teams at global financial institutions to the evolution of U.S. Cyber Command. The market has also evolved and cutting-edge security technologies are now available to organizations.

Leaders can drive significant change by deploying micro-segmentation within their new security stack of investments. Micro-segmentation is a wise strategic choice and a key enabler of a defense-in-depth strategy.

If you have any questions, please contact Illumio by email at info@illumio.com, by phone at +1-855-426-3983, or on Twitter at @illumio.

10 About the Authors

Jonathan Reiber

Jonathan was Head of Cybersecurity Strategy at Illumio while working on this book. A former Chief Strategy Officer for Cyber Policy and Speechwriter in the Office of the U.S. Secretary of Defense, his background is in national security policy planning, cybersecurity strategy, and non-fiction writing.

Prior to Illumio, Jonathan held a senior writing and research fellowship at the University of California at Berkeley's Center for Long-Term Cybersecurity, and previously spent seven years in the Obama Administration within the U.S. Department of Defense advising the Pentagon leadership. In his last role as CSO for Cyber Policy, he led initiatives across the cyberpolicy portfolio, to include strategic planning, key interagency and industry partnerships, and strategic communications. He was the principal author of the Department of Defense Cyber Strategy of 2015. Jonathan is a distinguished graduate of Middlebury College and The Fletcher School of Law and Diplomacy.

Matthew Glenn

As Vice President of Product Management at Illumio, Matt is responsible for product lines and product strategy. Prior to Illumio, he was Vice President of Product Management for the Network Security business unit at McAfee, supporting its Firewall, Intrusion Detection System, Email Security, Web Security, Data Loss and Prevention, and Identity product lines. Before McAfee, Matt was founder and CEO of PlantSense, an Internet of Things start-up that created the EasyBloom Plant Sensor, whose sensor technology was sold to Parrot SA.

Before starting EasyBloom, he worked at Cisco Systems – after its acquisition of Airespace – where he ran Product Management and grew the organization's revenue from zero to a billion dollar run rate. Matt previously worked at Xircom (IPO), Xylan (IPO), and Packet Engines (acquired by Alcatel). Matt studied English and Computer Information Systems at Humboldt State University.

Ron Isaacson

Ron is a member of the Office of the Chief Technology Officer at Illumio. Based in New York, Ron works with Illumio's enterprise and financial service customers in the eastern United States and Canada to help them meet their security and compliance needs using Illumio's Adaptive Security Platform. He is an expert in helping large organizations overcome operational challenges related to security initiatives, and in guiding audit processes to demonstrate the effectiveness of Illumio's policy-based controls.

Prior to Illumio, Ron spent nearly 15 years at Morgan Stanley managing application development, infrastructure, and security teams. Ron led the development of Morgan Stanley's Technology Access Management program, a foundational control for protecting technology assets and achieving mandated IT Separation of Duties. Earlier in his career, he pioneered the development of online banking platforms as the CTO of a start-up in Philadelphia. Ron received his Computer Science degree from the School of Engineering and Applied Science at the University of Pennsylvania.

PJ Kirner

As Chief Technology Officer and Founder of Illumio, PJ is responsible for Illumio's technology vision and platform architecture. He has 20 years of experience in engineering, with a focus on addressing the complexities of data centers. Prior to Illumio, PJ was CTO at Cymtec. He also held several roles at Juniper Networks, including distinguished engineer focused on advancing Juniper's network security and layer 4-7 services plane. PJ graduated with honors from Cornell University.

Nathanael Iversen

Nathanael is Vice President of Field Enablement at Illumio. He is responsible for training Illumio's sales, systems engineering, and customer success teams. He also helps customers understand and validate deployments. Nathanael has over two decades of customer-facing experience, with a broad background in networking, security, and virtualization. He has held positions in systems engineering, product management, and technical marketing; has experience with enterprise data centers and telecommunications provider networks; and began his career designing and implementing large-scale data centers for the U.S. Air Force. Nathanael holds a degree in Communication Systems Design from the Community College of the Air Force.

Russell Goodwin

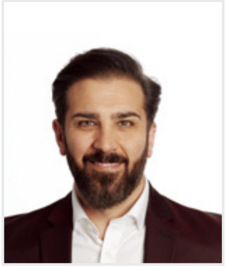
Russell works in the Office of the Chief Technology Officer at Illumio, where he helps customers solve complex security problems in and around the data center. He has been a network security practitioner for 25 years and has spent most of his time in the banking and payments industry consulting to customers such as Goldman Sachs, HSBC, Visa, American Express, ING Bank, Westpac, and Nomura. Russell has led professional services teams in Asia and Europe and worked in Singapore for Juniper Networks. He is an expert at understanding and solving customer problems in both security and data center design, and has experience working with corporations in the technology, pharmaceutical, governmental, and utilities industries. Russell holds patents in the field of network-based authentication and has extensive experience with network protocol design and operations.

Mukesh Gupta

Mukesh was Vice President of Product Management at Illumio while working on this book. He is currently Vice President of Product Management for the VM-Series Firewall at Palo Alto Networks. Mukesh was the first product manager at Illumio and he spent six years delivering over thirty-five product releases with features such as Illumination, multi-dimensional label-based policy, SecureConnect, SAML/Kerberos Integration, Adaptive User Segmentation, F5 Integration, RBAC, and IPv6. In his last two years, he drove Illumio's container strategy and roadmap.

Prior to Illumio, Mukesh was co-founder and CEO of LocalCircles, a social networking startup, and managed NetScreen and SRX appliances at Juniper. Mukesh received his bachelor's degree in Engineering from the Maulana Azad National Institute of Technology (MANIT), Bhopal, India, and a Masters in Computer Science from the University of Toledo.

E. Jay Hussein



Jay was Director of Customer Success at Illumio while working on this book, responsible for customer adoption of software and ensuring good customer health throughout the lifecycle. Prior to Illumio, he was Vice President of Product and Services at CDI, where he owned the Storage, Network, and Virtualization verticals of the business. Jay's work with Cisco technology led him to pursue their certification track, and he earned six key certifications to qualify his organization as a Platinum reseller.

Jay spent a large part of his early career at Morgan Stanley running their global network deployment function. Jay has an extensive academic background, and earned a Ph.D. in English Literature and Philosophy at the University of Amsterdam.

11 About Us

Illumio enables organizations to realize a future without high-profile breaches by providing visibility, segmentation, and control of all network communications across any data center or cloud. Founded in 2013, the world's largest enterprises, including Morgan Stanley, BNP Paribas, Salesforce, and Oracle NetSuite, trust Illumio to reduce cyber risk. For more information, visit www.illumio.com.

12 Glossary

Application: A software program that runs on computers.

Application dependency map: A map showing the interactions and dependencies both within and between applications.

Computer network: A group of two or more devices that can communicate.

Configuration management database (CMDB): A database that contains all relevant information about the hardware and software components used in an organization and the relationships between those components. It serves as an asset inventory for the organization.

Containers: A method of operating system virtualization that allow you to run an application and its dependencies in resource-isolated processes.

Core services: Nagios, Active Directory, Network Time Protocol, and other services that most, if not all, of the workloads and applications within an organization use. Think of core services as the "utilities" of a city, like electricity, water, and waste management.

Crown jewels: Assets of such critical importance to an organization's business or mission that there would irreversible damage to the entity if they were lost, manipulated, or exfiltrated. As per accepted risk management practices, these assets are often prioritized for protection using micro-segmentation (i.e., ringfencing the crown jewels).

Data center: A facility used to house computer systems and associated components.

Host: A computer or another device connected to a network.

Metadata: Data that describes other data. Meta is a prefix that in common information technology usages means "an underlying definition or description." Metadata summarizes basic information about data, which can make finding and working with particular instances of data easier. A simple example is the filename or last edited date for a file on a computer system. Neither the name or the date contain the file – they are extra bits of information appended to the file to give it meaning. Illumio labels are metadata. Broadly speaking, organizations cannot move to automated workflows until they have sufficient metadata in place to represent the structures that need to be automated and tracked. This is a recurring theme with ramifications far beyond Illumio deployment and a challenge facing most enterprise customers, not all of whom have previously realized the strategic importance of metadata-driven workflows.

Micro-segmentation (sometimes referred to as security segmentation): A security technique that involves isolating digital assets such that only workloads and applications that should be able to communicate with each other can communicate. Micro-segmentation takes a Zero Trust (sometimes referred to as default-deny, whitelist, or least privilege) approach to security policy such that the default stance is to block access unless explicitly authorized. Segmentation policies can be applied at different levels of granularity: at the environment level (e.g., enforcing separation of production from development), application level, workload level, or even the individual process level. This approach allows organizations to deploy micro-segmentation using a software-only approach, agnostic to the underlying infrastructure or location of workloads.

Micro-segmentation reduces the attack surface and thereby minimizes the spread of threats within data centers and cloud environments. A good micro-segmentation product is able to stop an intruder in their tracks even after they penetrate one application or a few servers; the intruder simply won't be able to move further in the data center or cloud environment.

Public cloud computing: The paradigm in which compute resources are made available to customers via the internet on infrastructure that is hosted by third-party providers. One of the defining characteristics of this model is elasticity: resources can scale up and scale down quickly as per the consuming organization's needs.

Ringfencing: The technique of isolating a high-value asset or crown jewels application such that communication with other workloads and applications is restricted to only what is required for the proper functioning of the application. This approach mitigates the risk of threats being able to spread to and compromise the high-value asset.

Server: A device (virtual or physical) that performs a specific function or a collection of functions based on the applications and services that are running on that server. In a virtualized environment, a single physical server (often referred to as a bare-metal server) can be abstracted ("virtualized") to appear and function as multiple virtual servers.

Vulnerability map: A map overlaying vulnerability information (from third-party vulnerability management vendors) on top of Illumio's application dependency map (Illumination®). This product capability within the Illumio offering helps security teams see which applications have open vulnerabilities (vulnerabilities that have not been patched) as well as the open pathways that an attacker could traverse in order to reach the vulnerable workload and exploit the open vulnerabilities.

Workload: A discrete operating system instance that can run on a bare-metal server, in a virtual machine, on a containerized host, or in a cloud environment.