illumio

WHITE PAPER

VISIBILITY BEHIND THE FIREWALL

CONTENTS

OVERVIEW

BUSINESS DRIVERS

For years, enterprises have been using perimeter firewalls to protect workloads inside the data center. They started out by putting all workloads into a single zone and installing a firewall to protect that zone from the outside world. In time, additional applications were added to the data center but placed into different zones—for instance, separating all workloads that are part of the web tier from all workloads that are part of the database tier.

Often, enterprises then use zones, VLANs, or subnets to separate the new applications. But it can be very difficult to tease apart older applications into different zones because:

- The applications were not well documented
- The evolution of the application was not well documented
- The interactions of those workloads are not understood
- Employees that originally wrote and designed the applications no longer work at the company

Over time, the massive growth of enterprise applications, workloads (both physical and virtual), and change inside of data centers has made the security infrastructure inside the data center more complex.

Understanding the interactions between workloads is critical when a physical server is being decommissioned, when migrating workloads to public clouds, when changing the data center an application operates in, and during audits. The problem is how to learn the interactions in a simple but accurate way. Most security solutions do not offer this visibility.

THE ILLUMIO SOLUTION

The Illumio Adaptive Security Platform (ASP)TM uses real-time context data from application workloads to graph existing interactions between application workloads, even before the specification or enforcement of any security rules. The Illumination mode displays these interactions in an interactive graph, enabling administrators to easily gather information about the current state of affairs. This unique visibility helps administrators design well-informed security policies and describe them as explicitly permitted interactions between workloads. With Illumio ASP, fine-grained security is attached at the level of individual workloads. Telemetry data from workloads is then used to analyze their context in real time and to compute the graph of dependencies between workloads. Illumio ASP then applies accurate security enforcement policies to the workloads, locking down interactions based on specified policies.

CURRENT APPROACHES TO GAINING VISIBILITY BEHIND THE FIREWALL

NETFLOW

Enterprises often try to learn the interactions between workloads by using NetFlow data gathered at the switches or routers in the data center. However, this can be a very long forensics exercise since IP addresses need to be tied to specific workloads, the function of those workloads needs to be learned, and the services that run on those workloads need to be discovered. In addition, as workloads “fan out” and connect with other workloads, the connectivity matrix grows, which only serves to expand the forensic analysis effort. Finally, when workloads are hidden behind a load balancer that hides the destination IP, the problem is only exacerbated.

The lack of context from NetFlow-based tools makes it very difficult to fully learn the interactions of the workloads on the network. What’s more, these tools must be run and analyzed for a long period of time to uncover any periodic interactions.

SOFTWARE RECONSTRUCTION

Some organizations have taken the extreme step of pulling original source-code repositories and then mapping the code to network interactions. The code can be compared with NetFlow information, and correlated to see if the interactions described in the code are actually observed on the network. This provides some ability to learn the dependencies of individual workloads and better understand context, since the actual interactions are observed on the network, and seen in the code. However, periodic batch jobs may not be discovered if they are not seen on the network nor observed in the actual code.

THREE CHALLENGES WITH EXISTING SOLUTIONS

1. The processes are resource intensive

Regardless of how an organization works to find dependencies, individuals must manually comb through logs to piece the interactions together and to understand the context of each connection. Even with scripting optimizations, it may take a year to find all of the batch jobs that happen at occasional intervals.

2. Changes will have an impact on results

Even if an organization goes through a year-long process of tracking down all connections, documenting them, and making a plan for instrumenting east-west firewall policies, there is a high probability that the workload graph (i.e., the workloads that make up the application) will have changed significantly by the end of the process—impacting the planned firewall rules.

3. The tools don’t provide enough workload context

In order to understand the interactions between the workloads that comprise an application, a graph must be built that shows the services running on those workloads and the traffic flows between the workloads. Without a complete understanding of the individual workloads and their direct interactions, it is difficult to fully understand how they relate with one another, and therefore difficult to get an accurate assessment of the overall graph. Plus, if there are any changes to interface IP addresses, or if a workload moves or scales, those updates must be accounted for.

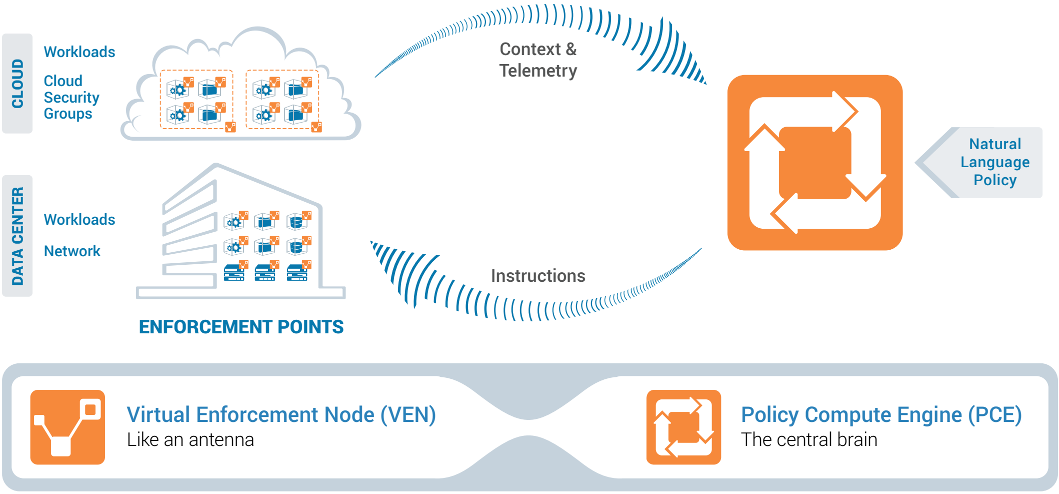
THE ILLUMIO SOLUTION

Illumio Adaptive Security Platform (ASP) secures enterprise applications in data centers and private, public, or hybrid cloud environments by decoupling security from the underlying infrastructure. The platform continuously computes security for enterprise applications by using the dynamic context of individual workloads running on virtual machines or physical servers.

The **Illumination** service provides visualization of applications and workload interactions with a graphical view of application topology to help inform security and policy decisions.

The **Policy Compute Engine (PCE)** is a centralized controller than manages all of the state and policies of the computing environment it visualizes and protects. It examines the relationships among workloads, computes the rules required to protect each workload, and distributes those rules out to the Virtual Enforcement Nodes (VENs) on the workloads.

Illumio ASP includes support for policy-driven encryption through the **SecureConnect** capability, which provides on-demand IPsec connectivity between workloads regardless of the underlying infrastructure.



Illumio solves the problem of learning the dependencies of individual workloads and building policies for those workloads (existing or new) through its Illumination service.

USING ILLUMINATION TO LEARN INTERACTIONS

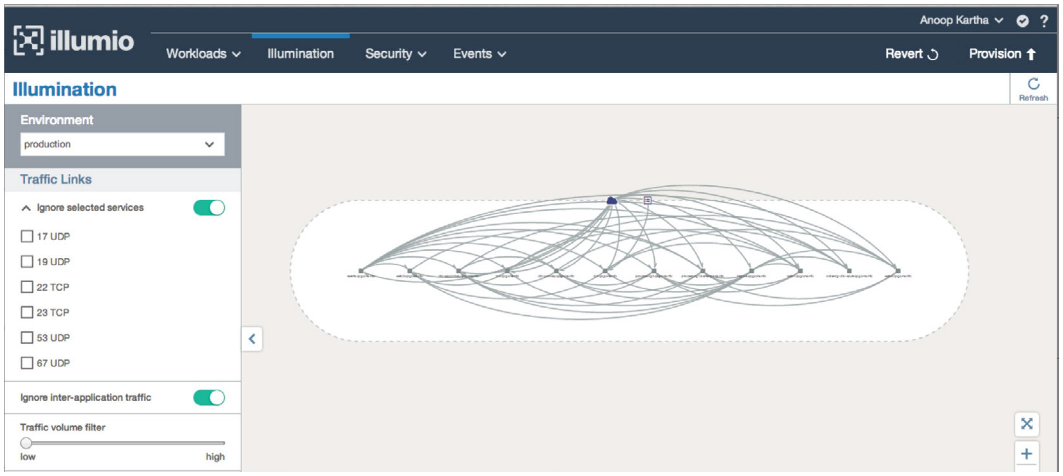
When Illumio ASP is in the Policy Testing mode of Illumination, the VEN writes policies into iptables or Windows Filtering Platform and programs rules that dictate the interactions between workloads. But instead of enforcing those rules on every workload, the VEN puts a “permit ANY LOG” rule. Once this rule is in place, each VEN logs flow data.

The VEN does not read or store any payload data. The only data that it stores is specifically related to flows:

- Source IP
- Destination
- Source port
- Destination port
- Flow sequence

All of this data is fed up to the PCE over an SSL-encrypted connection. The PCE then uses all of this information to build a graph of all the interactions with all the workloads.

For instance, if a workload that has Tomcat running on it accesses another workload running PostgreSQL, the graph will show the context of the workload running Tomcat, and that it is initiating outbound connections to a workload that has a process running PostgreSQL attached to a port that is using TCP.



Environment: production

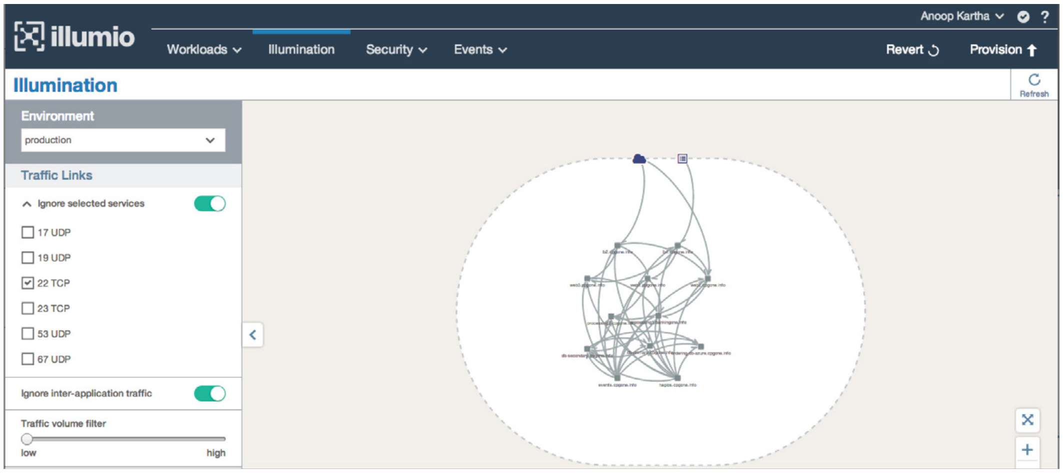
Ignore selected services ●

□ 17 UDP
□ 19 UDP
□ 22 TCP
□ 23 TCP
□ 53 UDP
□ 67 UDP

Ignore inter-application traffic ●

Figure 1: SSH traffic flows in the application graph

Users can also choose to “ignore” certain types of traffic and have the interaction graph recomputed. For instance, if both of the workloads described in the previous paragraph are managed using SSH, then the connections going into the SSH port would impact how the application graph is computed by the PCE (see figure 1). Users can choose to ignore different traffic on different ports and the PCE clustering algorithms will compute the graph, but ignore the management traffic (see figure 2).



Environment: production

Traffic Links

Ignore selected services (on)

- ☐ 17 UDP
- ☐ 19 UDP
- ☑ 22 TCP
- ☐ 23 TCP
- ☐ 53 UDP
- ☐ 67 UDP

Ignore inter-application traffic (on)

Traffic volume filter: low – high

Figure 2: SSH ignored in application graph

If there are workloads, groups of workloads, or users that are interacting with an application, and the traffic is coming from an IP list (i.e., a label that represents one IP address, a variable length subnet mask, or a block of IP addresses and variable length subnet masks), traffic is shown as originating from the IP list. If the traffic is coming from an address length that does not have a VEN on it, or an IP list, then the traffic is shown as originating from the Internet.

THE ILLUMINATION WORKFLOW

After the graph has been computed, the Illumination workflow walks the user through the process of labeling the application name and location of each workload within the graph. After labeling workloads with application and location, the administrator assigns a role to each workload.

Next, Illumination displays all of the flows in the graph in red. The red color indicates that Illumio ASP has detected flows that aren’t addressed by existing rules. By selecting the traffic lines between workloads, information about the flows is shown. Users can then click to create a rule. The rule is added to the Ruleset, but no rules are provisioned down to the actual workload. This is known as Policy Building mode.

After all of the flows have been added, users can migrate the application container to Policy Test mode. In this mode, all of the individual rules are added to workloads, but there is a “permit ANY ANY” rule added to the end. This step allows users to test the policy without blocking any traffic. If there is a batch job that runs only periodically, that batch job is detected and displayed, but no traffic is blocked.

Once an organization has created all of the rules for an application, it can migrate the workloads into Enforcement mode, where all of the rules are provisioned into the workload. In Enforcement mode, there is no permit rule at the end of the list. If any flows are blocked, they are logged in traffic events for administrators to analyze later.

Other benefits of Illumination mode include:

- **Legacy application migration:** Illumination enables the migration of legacy applications into a protection profile that allows users to migrate the workloads comprising those applications into new private data centers, private clouds, or public clouds.
- **Decommissioning of older servers:** At some point, older server hardware needs to be decommissioned because it becomes a liability and business risk. If administration has a limited view of the applications running on those legacy servers, they become difficult to decommission. Illumination allows organizations to learn the dependencies in a noninvasive way.
- **Discovering malicious traffic or misconfigured applications:** Because Illumination delivers interworkload traffic visibility, it also provides visibility for workloads acting out of profile. Connections initiated out of a workload to another workload (or workloads) indicate one of three things:

1) A new rule needs to be added to the ruleset addressing the flow.
2) A workload is misconfigured and acting out of profile.
3) A workload has been compromised.

USE CASE: USING ILLUMINATION TO GAIN VISIBILITY BEHIND THE FIREWALL

To better understand how Illumination can be used to see what is behind the firewall, consider a two-tier ordering application deployed eight years ago into a data center. The application developers who originally created the application have left the company, and there is no documentation explaining it operates. In addition, no one knows what other applications are interacting with application.

Step 1: Pairing

The first step is to install the VEN on one or more of the workloads that comprise the application. The VEN will immediately examine the workload it is paired with, discovering what operating system is on the workload, what processes are attached to different ports, and what protocol is operating on those ports.

Step 2: Flow gathering

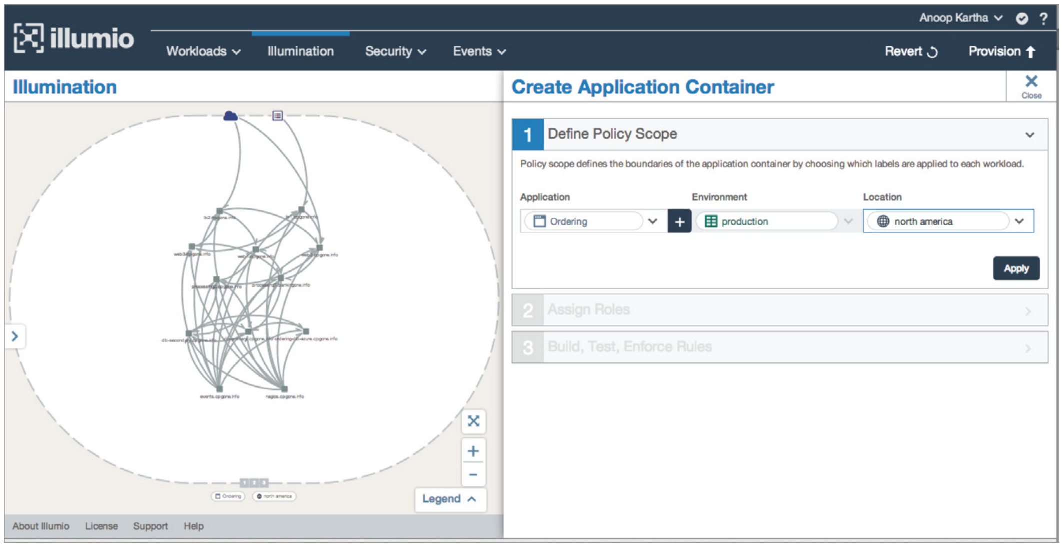
The VEN automatically begins to gather flow information for the workload it is paired with. Some of the flow information will be from users interacting with the application. If the user population has been labeled via an IP list, the traffic will show up as coming from an IP list. If the data center IP address range has been labeled (or other data centers have been labeled), the traffic will show up as coming from other workloads. Users can also determine which specific workloads within a data center are interacting with the paired workload. Workloads interacting with the paired workload can be paired with a VEN.

Step 3: Clustering

The PCE runs algorithms against the traffic and workloads that have been paired. Using a variety of algorithms, it develops a “cluster” of workloads that have a strong affinity for one another. A cluster indicates a probability that the workloads comprise an application. The application will be displayed as a graph, with each line in the graph representing a flow (or flows) between workloads. The workloads themselves will be displayed as nodes on the graph.

Step 4: Labeling

The first step in building a ruleset that protects an application is to label the application name and the location (or locations) of the workloads of the application. In this case the application name would be “Ordering.”



Create Application Container

1 Define Policy Scope

Policy scope defines the boundaries of the application container by choosing which labels are applied to each workload.

Application: Ordering
Environment: production
Location: north america

Apply

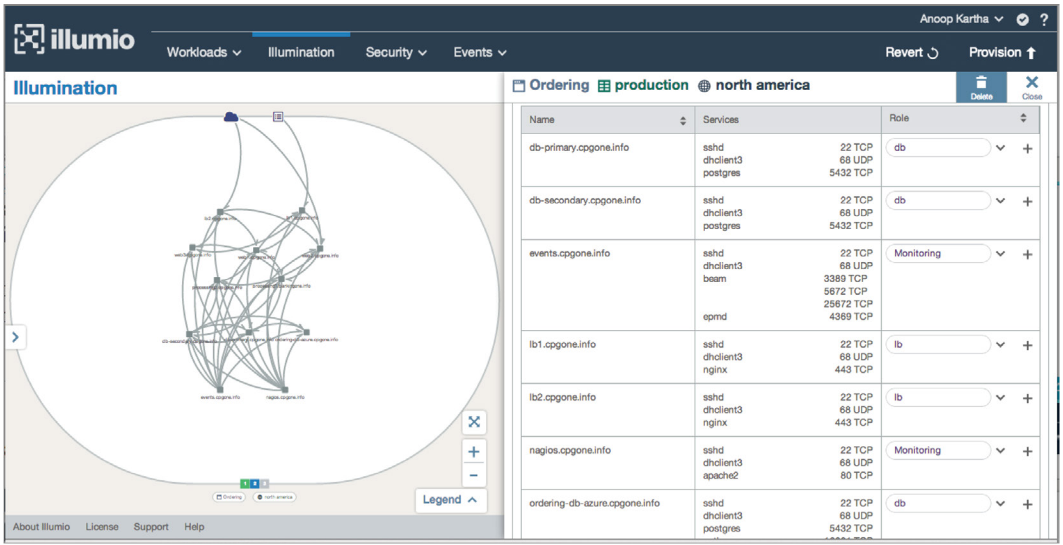
2 Assign Roles

3 Build, Test, Enforce Rules

Figure 3: Defining the application container

After labeling the application, users can label the role of the individual workloads that comprise it.

Illumio ASP displays the processes and ports running on each workload as the roles are being assigned.

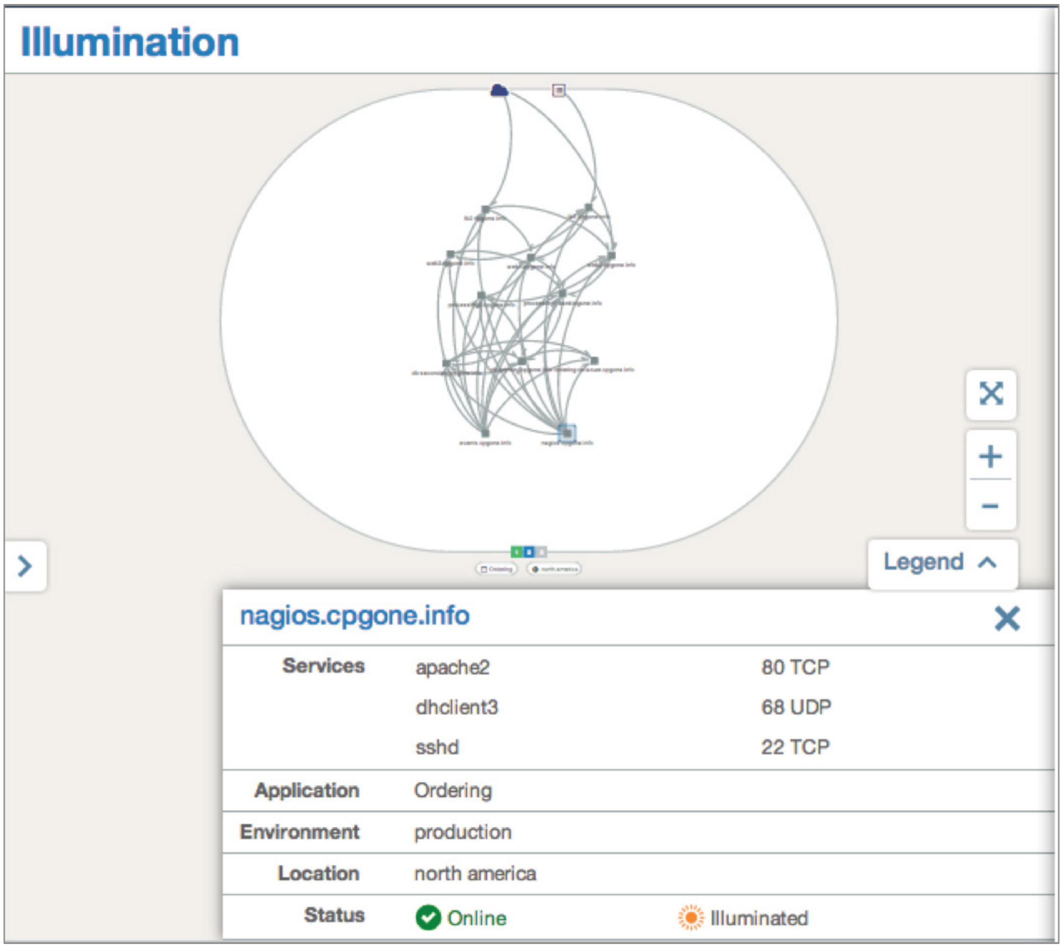


Ordering production north america

Name	Services		Role
db-primary.cpgone.info	sshd dhclient3 postgres	22 TCP 68 UDP 5432 TCP	db
db-secondary.cpgone.info	sshd dhclient3 postgres	22 TCP 68 UDP 5432 TCP	db
events.cpgone.info	sshd dhclient3 beam epmd	22 TCP 68 UDP 3389 TCP 5672 TCP 25672 TCP 4369 TCP	Monitoring
lb1.cpgone.info	sshd dhclient3 nginx	22 TCP 68 UDP 443 TCP	lb
lb2.cpgone.info	sshd dhclient3 nginx	22 TCP 68 UDP 443 TCP	lb
nagios.cpgone.info	sshd dhclient3 apache2	22 TCP 68 UDP 80 TCP	Monitoring
ordering-db-azure.cpgone.info	sshd dhclient3 postgres	22 TCP 68 UDP 5432 TCP	db

Figure 4: Assigning roles

At this point, the Nagios server can still be labeled, but eventually it will be moved to its own application container since it monitors multiple applications.



nagios.cpgone.info

Services	apache2	80 TCP
	dhclient3	68 UDP
	sshd	22 TCP
Application	Ordering	
Environment	production	
Location	north america	
Status	✓ Online	Illuminated

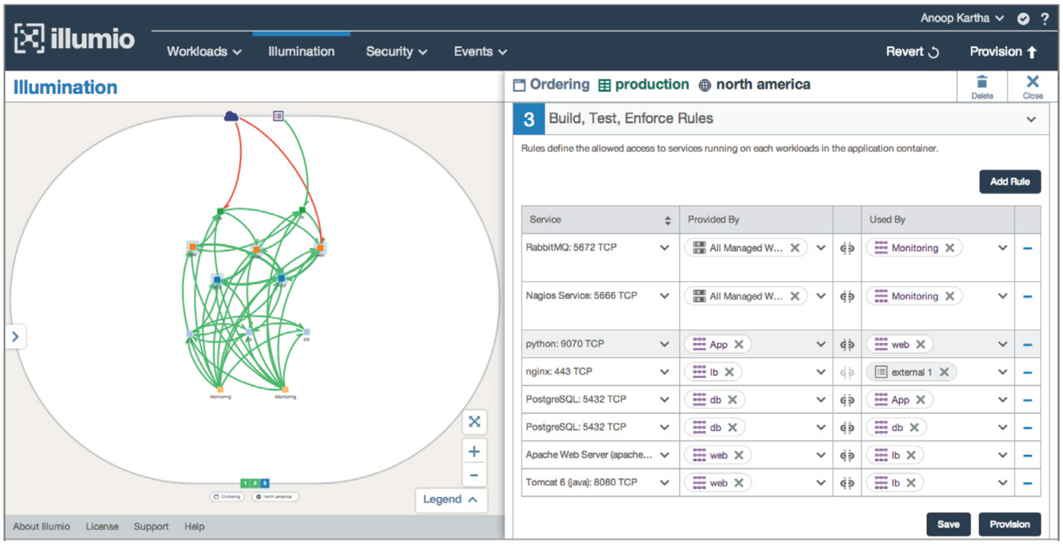
Figure 5: Visualizing the Nagios server

After labeling the role of the individual workloads, the lines of the graph will turn from grey to red. Red lines indicate that traffic flows have been detected, but there are no rules that address them.

Step 5: Rule writing

When a user clicks on a line in the graph, flow information appears in the Illumination workflow. At this point, a user can add the rule to a ruleset for the application. If the flow is permitted, the user simply clicks on Add Rule and a rule is automatically added to the ruleset. Once the user clicks Save, the lines of the graph turn from red to green. Users can continue to work their way down the graph and choose to add or ignore flows.

Note: Although a rule has been created, it is not provisioned down to the workload(s).



3 Build, Test, Enforce Rules

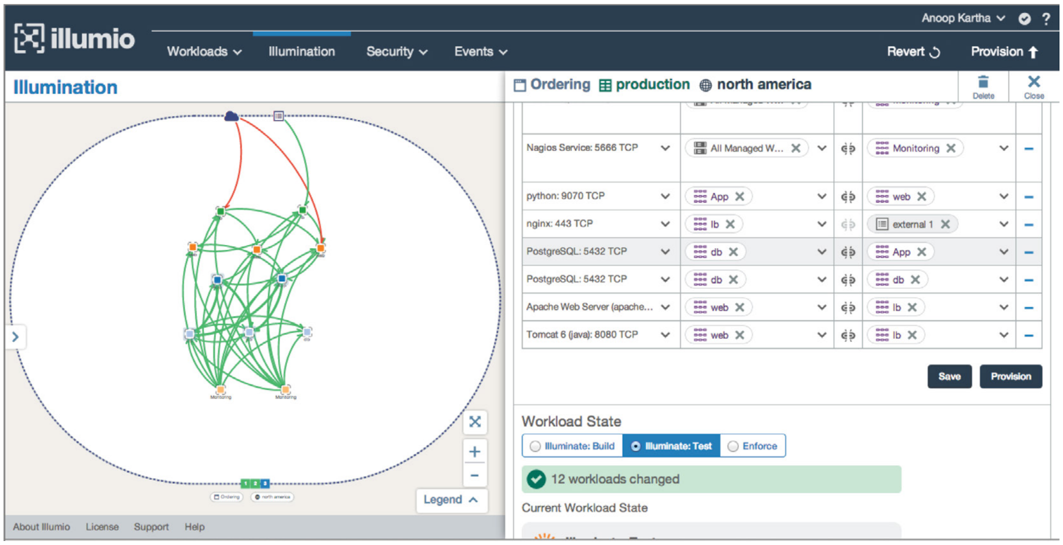
Rules define the allowed access to services running on each workloads in the application container.

Service	Provided By	Used By
RabbitMQ: 5672 TCP	All Managed W...	Monitoring
Nagios Service: 5666 TCP	All Managed W...	Monitoring
python: 9070 TCP	App	web
nginx: 443 TCP	lb	external 1
PostgreSQL: 5432 TCP	db	App
PostgreSQL: 5432 TCP	db	db
Apache Web Server (apache...	web	lb
Tomcat 6 (java): 8080 TCP	web	lb

Figure 6: Defining rules

Step 6: Testing the policy

When a proposed ruleset has been created, it can be moved from Policy Building mode to Policy Testing mode. Each of the rules will then be provisioned down to the individual workloads, but at the bottom of the rules on the workload, a “Permit ANY ANY LOG” rule is added.



Service	Provided By	Used By
Nagios Service: 5666 TCP	All Managed W...	Monitoring
python: 9070 TCP	App	web
nginx: 443 TCP	lb	external 1
PostgreSQL: 5432 TCP	db	App
PostgreSQL: 5432 TCP	db	db
Apache Web Server (apache...	web	lb
Tomcat 6 (java): 8080 TCP	web	lb

Workload State

○ Illuminate: Build ● Illuminate: Test ○ Enforce

✓ 12 workloads changed

Current Workload State

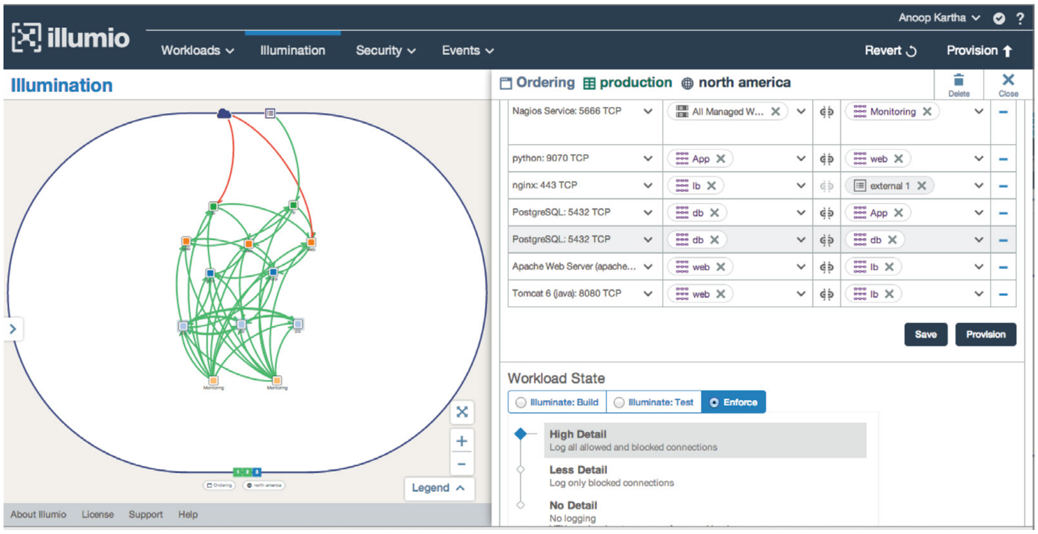
Figure 7: Policy Testing mode

At this point, all flows that match a rule are permitted. Any flow that does not match a rule is also permitted, but these flows are logged, and show up in the Illumination screen as red lines.

The benefit of Policy Testing mode is that it doesn't force the user into enforcement, but it still delivers visibility into all of the flows happening behind the firewall. Any new flows that happen after moving to testing can be added to the policy or investigated.

Step 7: Enforcement

Once an organization is comfortable with the policy, it can be migrated into Enforcement mode. The application can be migrated to a new data center, or simply left in place. In Enforcement mode, blocked or allowed traffic can still be logged (based on configuration).



Ordering · production · north america

Nagios Service: 5666 TCP	All Managed W...		Monitoring
python: 9070 TCP	App		web
nginx: 443 TCP	lb		external 1
PostgreSQL: 5432 TCP	db		App
PostgreSQL: 5432 TCP	db		db
Apache Web Server (apache...	web		lb
Tomcat 6 (java): 8080 TCP	web		lb

Workload State

○ Illuminate: Build ○ Illuminate: Test ● Enforce

High Detail
Log all allowed and blocked connections

Less Detail
Log only blocked connections

No Detail
No logging

Figure 8: Enforcement mode

ABOUT ILLUMIO

Illumio, recently named to the CNBC Disruptor 50 list, stops cyber threats by controlling the lateral movement of unauthorized communications through its breakthrough adaptive segmentation technology. The company's Adaptive Security Platform™ visualizes application traffic and delivers continuous, scalable, and dynamic policy and enforcement to every bare-metal server, VM, container, and VDI within data centers and public clouds. Using Illumio, enterprises such as Morgan Stanley, Plantronics, Salesforce, King Entertainment, NetSuite, Oak Hill Advisors, and Creative Artists Agency have achieved secure application and cloud migration, environmental segmentation, compliance and high-value application protection from breaches and threats with no changes to applications or infrastructure. For more information, visit www.illumio.com or follow @ Illumio.

- Engage with Illumio on Twitter
- Follow Illumio on LinkedIn
- Like Illumio on Facebook
- Join Illumio on G+
- Subscribe to the Illumio YouTube Channel

CONTACT US

For more information about Illumio ASP and how it can be used to achieve environmental separation, email us at illuminate@illumio.com or call 855-426-3983 to speak to an Illumio representative.