illumio

Securing Your SWIFT Environment Using Micro-Segmentation

WP201801

illumio

Overview

By January 1, 2018, all SWIFT customers must self-attest to their compliance with the new SWIFT Customer Security Program (CSP). The program is designed to respond to the wave of cybercrime targeting SWIFT installations that began with the Bank of Bangladesh breach back in February 2016 and has swept across the world over the last two years.

Needless to say, given this increased targeting, securing SWIFT environments is a priority for every financial institution regardless of the CSP; but like any deadline, meeting all of the CSP requirements on time can be challenging. If organizations have not met all 16 of the CSP's mandatory controls by January, they have until the end of 2018 to close those gaps. The SWIFT CSP requires the implementation of a wide range of controls from encryption to segmentation and multifactor authentication. Addressing any gaps can require rethinking existing strategies, updating security protocols, and deploying new tools to meet the CSP's requirements.

It is tempting to jump right into the 16 mandatory controls, but what most organizations don't realize about securing their SWIFT application is that there are three specific challenges that prove especially difficult for SWIFT customers looking to meet the compliance deadline:

1. Generating an accurate, real-time application dependency map (ADM) of the SWIFT application;
2. Segmenting the SWIFT application from the rest of the infrastructure and imposing segmentation to isolate highest value targets; and
3. Validating the SWIFT application's dependency map and segmentation.

This white paper explains each of these challenges and guides you on how to address them most effectively—ensuring that your SWIFT compliance initiative runs quickly and smoothly, shortening your audit and avoiding costly mistakes.

Generating an Accurate, Real-Time Application Dependency Map of the SWIFT Application

You can't secure your SWIFT application if you don't understand the infrastructure on which it runs and how its component systems communicate. This requirement is an essential first step to effective SWIFT compliance; organizations that skip this step risk missing key components of critical infrastructure, leaving themselves out of compliance and exposed.

SWIFT provides its customers with four possible application diagrams for their SWIFT deployment. By using architecture B, the need for this mapping is greatly reduced for organizations that outsource most of their SWIFT functionality because most of the critical systems reside on an external network that is beyond their immediate control. For organizations that run most or all of their SWIFT functionality themselves (architectures A1, A2, and even A3), mapping the extent of their SWIFT infrastructure is critical.

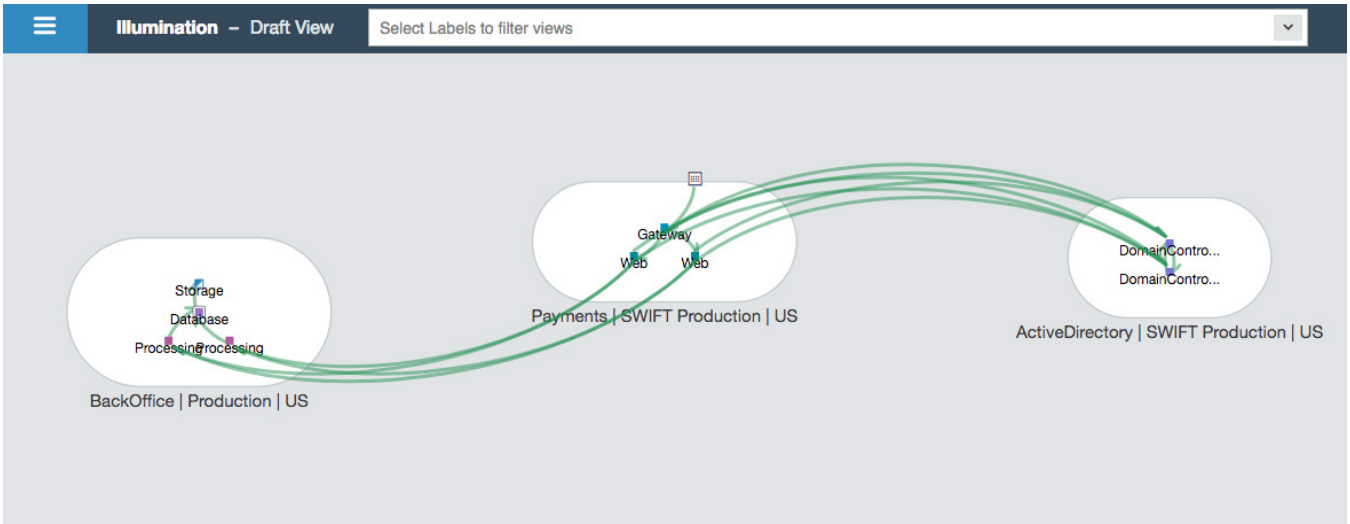
Many organizations begin this mapping exercise with a network diagram, but network diagrams are poorly suited to understanding application architecture because they only show how network devices communicate. They don't reveal how applications function and share data in real time. To build an accurate map of a SWIFT environment, you must understand how the various systems within it share and process data, for instance, how your SWIFT messaging interface interacts with your SWIFT GUI and communication interface.

Building this application-awareness from a simple network diagram generally requires extensive manual effort on the part of the infrastructure and security teams. It can be expensive and slow, and it requires constant manual review and updating as the SWIFT environment changes over time.

Building this diagram can be accelerated from days to hours if you have access to a real-time application dependency map. This is because an ADM is built on application-aware metadata. This means that it outlines all the ways that your applications communicate and it can maintain this awareness over time, showing you how communications and applications shift and change as the environment changes. By leveraging an application dependency map, you can quickly generate an accurate and up-to-date understanding of your SWIFT infrastructure, ensuring that you can tailor your compliance strategy to the reality of your systems.

Benefits of Application Dependency Mapping Versus Network Diagramming

	Real-Time Application Dependency Map	Network Diagram
Accuracy	An ADM can automatically identify and map all communications within your SWIFT environment, whether they originate elsewhere in the network or the public internet. This automatic identification process is reliable, repeatable, and up-to-date, greatly reducing the risk of costly human error.	A network diagram shows the devices connected to the systems, but not what they do. You must manually map this to the SWIFT functions to identify the scope of your SWIFT environment. A manual mapping exercise like this is prone to errors, making it likely that your team will miss systems and connections that should be covered and expose you to liability.
Precision	ADM's automated analysis lets you precisely map your SWIFT infrastructure, saving you resources and time by ensuring that your efforts accurately address your compliance burden.	Because of the risk of inaccuracy inherent in the manual steps required when using network diagrams, you risk over-compliance (investing more than necessary) or under-compliance (missing critical requirements or systems). Either leads to inefficiency and unnecessary cost.
Speed	ADM's automated, real-time process speeds up the mapping of your SWIFT environment by reducing the manual work your team must do. It speeds up the first audit and speeds up subsequent audits (when only environmental change must be tracked) even more.	Although a network diagram can serve as a useful jumping off point for understanding your SWIFT deployment, it takes extensive manual effort to transform a network diagram into a comprehensive map of your SWIFT architecture. This effort takes time and resources and can slow down SWIFT compliance to a crawl.



Real-time application communications within SWIFT environment

Segmenting the SWIFT Application From the Rest of the Infrastructure and Imposing Segmentation to Isolate Highest Value Targets

There are 16 mandatory controls in the SWIFT CSP, but not all controls are created equal. Many of them simply require the deployment of tools you likely already have elsewhere in your environment: malware protection, database and software integrity monitoring, multifactor user authentication, and others.

All of these should be addressed, but the most significant and most foundational CSP requirement is that you deploy two types of segmentation: 1) between your SWIFT environment and your other networks; and 2) between components of your SWIFT environment itself (CSP 1.1). The importance of this requirement is reflected in the fact that it is much longer, and contains more individual components, than any other SWIFT control.

It is also the control that can prove the most challenging to comply with because many organizations are not used to imposing the level of granular separation between systems that SWIFT requires.

Among other things, CSP 1.1 requires that the SWIFT environment be segmented from the back office and other components of the corporate network; it requires that host-by-host control of traffic be implemented within the SWIFT secure zone; it requires that internet access be limited to only certain, segmented systems; and it requires that SWIFT authentication systems be segmented from central authentication—ensuring that a compromise of a global active directory deployment will not expose the SWIFT system.

When most people think of enforcing these types of segmentation requirements, they think of traditional, network-based segmentation solutions, such as ACLs. Achieving the range and precision of segmentation required by the CSP using traditional network segmentation, however, is incredibly difficult. For instance, if your SWIFT deployment includes a cloud environment, you likely do not have control over the network infrastructure necessary to impose traditional segmentation. Similarly, applying comprehensive host-by-host segmentation using traditional network-based solutions quickly becomes untenable for even small deployments; the complexity of the enforcement devices and firewall rules is simply too great for effective maintenance and application.

Thankfully, there are a range of technologies beyond traditional network enforcement that can be used to accomplish the segmentation required by the SWIFT CSP. Many IT organizations are moving from network segmentation to these new technologies. Adaptive micro-segmentation combines the benefits of these new approaches.

Key Characteristics of Adaptive Micro-Segmentation

Flexible Granularity. Network segmentation can only enforce broad separations (for example, separating development from production). Adaptive micro-segmentation can enforce both broad separations and extremely precise segmentation.

Hybrid Environments. Network segmentation works only within traditional, static data centers. Adaptive micro-segmentation works across static and dynamic environments, in the cloud, and on bare-metal servers.

Adaptive Enforcement. Adaptive micro-segmentation can shift as environments shift. This means that as your SWIFT environment expands and changes over time, adaptive micro-segmentation saves time and resources by avoiding the manual exercise of keeping SWIFT segmentation up-to-date.

Application Awareness. Network segmentation does not understand the details of applications and communications. Adaptive micro-segmentation understands applications, ensuring that the segmentation protects the most valuable data.

	Network Segmentation	Adaptive Micro-Segmentation
Flexible Granularity	✖	✔
Hybrid Environments	✖	✔
Adaptive Enforcement	✖	✔
Application Awareness	✖	✔

Validating the SWIFT Application's Dependency Map and Segmentation

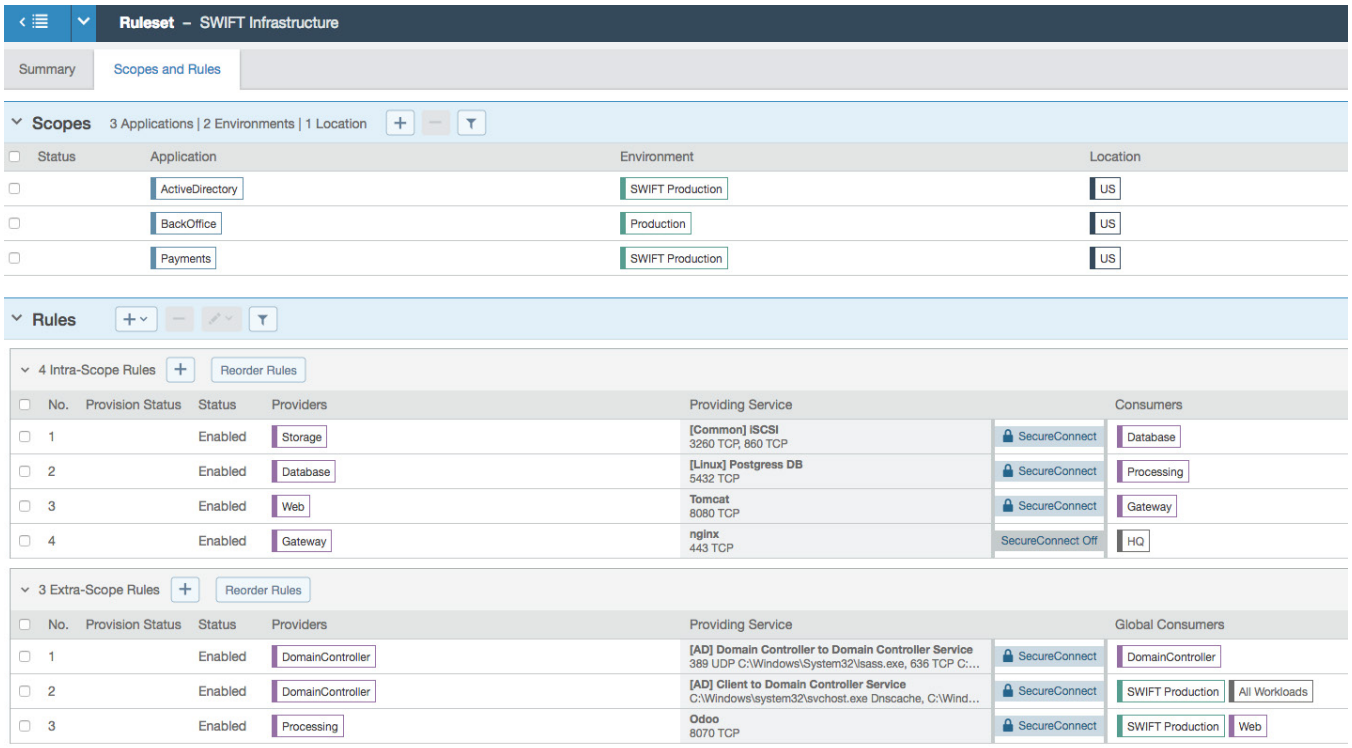
Segmenting the SWIFT infrastructure is an essential step in securing the SWIFT application, but it won't help with the SWIFT CSP compliance unless you can validate that the segmentation is accurate and correctly enforced.

When addressing the range of segmentation requirements embedded in SWIFT CSP 1.1, remember that achieving compliance means being able to prove compliance.

Here are four characteristics to look for in any scoping tool to ensure that you can prove compliance:

HUMAN-READABLE POLICIES

Segmenting your SWIFT infrastructure can easily require thousands of firewall rules. Rather than imposing segmentation based on traditional firewall rules, look for solutions that use environmental metadata to create more efficient and more understandable rules. These make it easier for you to explain your scope to an auditor and easier for auditors to validate it as well.



Human-readable policies for securing SWIFT environments

illumio

A PICTURE IS WORTH A THOUSAND WORDS

There's a reason that the first step in any SWIFT CSP compliance exercise is building an accurate application dependency map of the SWIFT application. Instead of working with spreadsheets of IP addresses and firewall rules, a visual interface can actually map your SWIFT systems allowing you to understand the primary connections within your SWIFT application and then to identify key characteristics about those connections.

ANSWER KEY QUESTIONS BEFORE THEY'RE ASKED

SWIFT CSP 1.1 outlines specific segmentation requirements. Use a solution that will let you identify and answer these questions at the beginning of the audit, simplifying back-and-forth, and get through the audit as quickly as possible.

QUERY PLATFORM TO CHASE DOWN AND FIX ERRORS

Inevitably, something will turn up that you didn't expect. Given the complexity of modern environments and the range of requirements that the SWIFT CSP imposes, at least one curveball is guaranteed. Look for a security solution with an effective query-and-correct interface that prepares you for this so you can pinpoint and address mistakes when they do arise.

Conclusion

Micro-segmentation is a core requirement of the SWIFT CSP. More than that, building an early and accurate application dependency map lets you understand what systems need protecting and how you can apply that protection easily, quickly, and inexpensively.

If you use micro-segmentation to follow the steps laid out in this white paper, your SWIFT CSP audit will run smoothly and leave you with SWIFT infrastructure that is secure, flexible, and ready to run your business.

Follow Us

About Illumio

Illumio, the leader in micro-segmentation, prevents the spread of cyber threats inside data centers and cloud environments. Enterprises such as Morgan Stanley, BNP Paribas, Salesforce, Workday, and Oracle NetSuite use Illumio to reduce cyber risk and achieve regulatory compliance. Illumio's Adaptive Security Platform™ uniquely protects critical information with real-time application dependency mapping and micro-segmentation that works in any data center, public cloud, or across hybrid deployments on bare-metal, virtualization, and containers. For more information about Illumio, visit www.illumio.com/what-we-do or follow @Illumio.