# Table of Contents

Abstract ............................................................................................................................... 5
Scope ................................................................................................................................. 5
Out of Scope .................................................................................................................. 5
Target Audience .......................................................................................................... 6
Prerequisite ................................................................................................................... 6

Introduction .................................................................................................................. 6
  Secure Cloud Business Flow ......................................................................................... 8
  Public Cloud Attack Surface ..................................................................................... 8
  Architecture .............................................................................................................. 9

Solution Overview ...................................................................................................... 10

AWS Introduction ........................................................................................................ 11
  Cloud Definition ....................................................................................................... 11
  Shared Security Model ............................................................................................ 12
  AWS Services .......................................................................................................... 12
  AWS Security Services ............................................................................................. 13

Cisco Security Portfolio .............................................................................................. 14

AWS Three-tier Web App Reference Architecture ..................................................... 15

Cisco Secure Three-Tier App Architecture for AWS .................................................... 16
  Cisco Tetration .......................................................................................................... 17
    Tetration Dashboard ............................................................................................... 17
    Tetration in AWS .................................................................................................. 18
  Cisco Stealthwatch Cloud ......................................................................................... 19
    Stealthwatch Cloud Dashboard ............................................................................ 20
    Stealthwatch Cloud in AWS ................................................................................ 20
  Cisco AMP for Endpoint .......................................................................................... 21
    AMP for Endpoints Dashboard .............................................................................. 22
    AMP for Endpoint in AWS ................................................................................... 22
  Cisco Umbrella .......................................................................................................... 23
    Umbrella Dashboard ............................................................................................... 23
    Umbrella Virtual Appliance (VA) in AWS ............................................................. 24
  Cisco Duo Beyond ...................................................................................................... 25
    Cisco Duo Beyond Dashboard ............................................................................... 25
Cisco Duo Beyond for AWS ................................................................. 26
Cisco Defense Orchestrator ..................................................................... 27
Cisco Defense Orchestrator Devices and Services ...................................... 27
Cisco Defense Orchestrator for AWS ..................................................... 27
Cisco Threat Response .......................................................................... 28
Cisco Threat Response Dashboard .......................................................... 28
Cisco Threat Response in AWS .............................................................. 29
Amazon Web Services – Security ............................................................ 30
AWS Security Group (SG) ..................................................................... 30
AWS Web Application Firewall (WAF) .................................................. 31
AWS Shield (DDoS) .............................................................................. 32
Radware Web Application Firewall and DDoS ........................................ 32
Traffic Flow ......................................................................................... 34
Application to Internet (app-to-internet) .................................................. 34
Security recommendations for “app-to-internet” traffic flow: ................. 34
User to Application .............................................................................. 35
Security recommendations for “user-to-app” traffic flow: ...................... 35
East/West Traffic Flow ........................................................................ 36
Security recommendations for “east/west” traffic flow: .......................... 36
App-to-app Traffic Flow ........................................................................ 37
Security recommendations for “app-to-app” traffic flow: ........................ 37
API client to APP ................................................................................ 38
Security recommendations for “apiclient-to-app” traffic flow: ................. 38
AWS API Gateway to APP: ................................................................. 39
Security recommendations for “apigw-to-app” traffic flow: .................... 39
AWS DevOPS to APP: ......................................................................... 40
Security recommendations for “DevOps-to-app” traffic flow: ................. 40
Design Implementation ......................................................................... 41
Setting up AWS Virtual Private Cloud (VPC) ........................................ 41
Create the AWS VPC and associated Components .................................. 41
Integrate Stealthwatch Cloud and Cisco Defense Orchestrator (CDO) .... 46
Setting up Load Balancers ................................................................... 47
Setting up frontend ‘ALB’ and backend ‘NLB’ ......................................... 47
Setting Up the application ....................................................................... 48
Setting up Cisco Umbrella virtual appliances ........................................ 49
Setting up database ............................................................................... 50
Setting up web servers and app servers .................................................. 51
Accessing the application ...................................................................... 55
Setting up the API gateway .................................................................... 57
Creating Lambda Function ...................................................................... 57
Create the API gateway ......................................................................... 60
Adding WAF and DDOS Capabilities ...................................................... 62
Option 1: AWS WAF and Shield ............................................................. 62
Option 2: Radware Cloud Service for WAF and DDOS Prevention ......... 65
Validation Testing .................................................................................................................................................. 68

Tetration ................................................................................................................................................................. 68
  Test case 1: Creating an application workspace in Tetration ................................................................. 68
  Test case 2: Using Application Dependency Mapping (ADM) run to set up policies and app view ...70
  Test case 3: Enforcing policies on workloads ...................................................................................... 72
  Test case 4: Detecting vulnerable workloads ....................................................................................... 74

Advanced Malware Protection for Endpoints (AMP4E) ............................................................................... 75
  Test Case 1: Detect and quarantine the suspicious file ......................................................................... 75

Stealthwatch Cloud and Cisco Defense Orchestrator (CDO) ................................................................. 77
  Test Case 1: Detect suspicious activity & manage AWS security groups ........................................... 77

Cisco Umbrella .................................................................................................................................................. 80
  Test Case 1: DNS security ....................................................................................................................... 80

Duo Beyond ......................................................................................................................................................... 82
  Test Case 1: Set up the cloud application for Two-Factor Authentication (2FA) ................................. 82
  Test Case 2: Monitor 2FA activity from Duo admin portal .................................................................. 83

Cisco Threat Response (CTR) ...................................................................................................................... 84
  Test Case 1: Integrate AMP4E and Umbrella with CTR ................................................................... 84
  Test Case 2: Track malicious Activity on CTR ................................................................................... 85

Appendix A – Software Versions Tested ..................................................................................................... 88

Appendix B – References ............................................................................................................................. 88
Abstract

This design guide is based on Secure Cloud Architecture Guide. The Secure Cloud Architecture Guide explains cloud services, critical business flows, and security controls required for the cloud environment to protect workloads. This guide covers the Cisco Validated Designs for workload protection in AWS three-tiered architecture. This also includes cloud-native security controls and Radware WAF/DDoS for workload protection in the cloud.

Scope

This design guide covers the following security components to protect workloads in AWS:

- Cisco Tetration (Visibility, Segmentation, Enforcement and Behavioral deviation)
- Cisco Stealthwatch Cloud (Visibility and Compliance)
- Cisco AMP for endpoints (Advanced Malware and Protection)
- Cisco Umbrella (DNS based security)
- Cisco Duo Beyond (Multi-factor authentication and single sign-on)
- Cisco Defense Orchestrator (Management of AWS security groups)
- Cisco Threat Response (Threat Hunting)
- Radware Web Application Firewall (Web exploits protection)
- Radware Distributed Denial of Service (DDoS)
- AWS Security Group (Network Segmentation)
- AWS Web Application Firewall (Web exploits protection)
- AWS Shield (DDoS)
- AWS API Gateway and Lambda

Cisco AMP, CTR, Stealthwatch Cloud, Umbrella, Duo and CDO offer EU based locations for customers having to follow EU rules.

Out of Scope

This design guide does not cover the following security components and cloud deployment scenarios, but will soon:

- Cisco NGFW (Firewalling, AVC, NGIPS, URL, AMP and VPN)
- Cisco ASA (Firewalling, and VPN)
- Kubernetes
- Containers
- Function as a Service (FaaS) or Serverless Architecture Deep Dive
- Platform as a Service (PaaS)
Target Audience

This Cisco Validated Design guide for AWS provides best practices for cloud security using Cisco security controls, cloud-native controls, and Radware. Architects require a validated design guide to define security controls required in their cloud environment. This document also includes implementation steps required by the implementation engineers. The target audience of this document is architects and implementation engineers.

Prerequisite

This design guide uses cloud-native services and cisco security components to protect workloads in AWS. The following are the pre-requisites for this design guide:

- Understanding of AWS IaaS and cloud-native services
- Knowledge of Cisco security portfolio
- Knowledge of APIs
- Basic networking knowledge

Introduction

Cisco’s design guide for Secure Cloud for Amazon Web Services includes robust and intent-based security that protects workloads, applications, infrastructure, and user data in AWS Virtual Private Cloud (VPC). Cisco’s solution continuously learns, adapts, and shields. As the network changes and new threats arise in AWS, Cisco security solutions dynamically detect and automatically adjust, mitigating threats in real-time.

The main concern of migration to the public cloud and running business in the cloud is the question of whether data would be secure. As migration to the cloud became more prevalent, our trust in the safety of the public cloud increased. However, many people are still unsure whether their data is safe in the cloud, and they consider on-premises data centers to be safer place for data storage. Unfortunately, this lack of faith in the public cloud is common.

The security of workloads, applications, infrastructure in the cloud is the responsibility of the security architect. Cisco Secure Cloud architecture ensures security, visibility, remediation, and threat mitigation in AWS VPC.

The Secure Cloud is a place in the network (PIN) where a company centralizes data and performs services for business. Cloud service providers host data center services in the Secure Cloud. This guide addresses Secure Cloud business flows and the security used to defend them. The focus of this guide in on the security controls necessary to provide “Security FOR the cloud – Amazon Web Services”
The Key to SAFE. SAFE provides the Key to simplify cybersecurity into secure places in the Network (PINs) for infrastructure and secure domains for operational guidance.

SAFE simplifies end-to-end security by using views of complexity depending on the audience’s needs. Ranging from business flows and their respective threats to the corresponding security capabilities, architectures, and designs, SAFE provides guidance that is holistic and understandable.
More information about how Cisco SAFE simplifies security, along with this and other Cisco Validated Designs (CVD), can be found here: www.cisco.com/go/safe.

Secure Cloud Business Flow

SAFE uses the concept of business flows to simplify the identification of threats. This enables the selection of capabilities necessary to protect them. Business flows are discussed in the Secure Cloud Architecture Guide.

Public Cloud Attack Surface

The attack surface of the cloud is defined by the business flows and includes the people and the technology present. The security capabilities that are needed to respond to the threats are mapped in the figure. The placement of these capabilities is discussed in the architecture section.
SAFE Design Guide

Secure Cloud for AWS - Design Guide

April 2020

Secure Cloud Attack Surface and Security Capabilities

Architecture

SAFE underscores the challenges of securing the business. It enhances traditional network diagrams to include a security-centric view of the company business. The Secure Cloud architecture is a logical grouping of security and network technology that supports business use cases.

SAFE business flow security architecture depicts a security focus. A SAFE logical architecture can have many different physical designs.
Solution Overview

Cisco’s security approach for the modern cloud allows companies to achieve:

- Improved resiliency to enable cloud availability and secure services
- Operational efficiency from automated provisioning and flexible, integrated security
- Advanced threat protection from Cisco Talos – industry-leading threat intelligence to stay up to date, informed, and secure

The integrated product workflow enables:

- **Visibility** – Complete visibility of users, devices, networks, applications, workloads, and processes
- **Segmentation** – Reduce the attack surface by preventing attackers from moving laterally, with consistent security policy enforcement, application whitelisting and micro-segmentation
- **Threat Protection** – Stop the breach by deploying multi-layered threat sensors strategically in the data center to quickly detect, block, and dynamically respond to threats

The top priorities for securing data centers are:

- **Visibility**
  “See Everything”
  Complete visibility of users, devices, networks, applications, workloads and processes

- **Segmentation**
  “Reduce the Attack Surface”
  Prevent attackers from moving laterally east-west with application whitelisting and micro-segmentation

- **Threat Protection**
  “Stop the Breach”
  Quickly detect, block, and respond to attacks before hackers can steal data or disrupt operations
AWS Introduction

Amazon Web Services (AWS) provides on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered pay-as-you-go basis. In aggregate, these cloud computing web services provide a set of primitive abstract technical infrastructure and distributed computing building blocks and tools. One of these services is Amazon Elastic Compute Cloud, which allows users to have at their disposal a virtual cluster of computers, available all the time, through the Internet.

Cloud Definition

Software as a Service (SaaS): Software that is deployed over the internet; with SaaS, a provider licenses an application to customers either as a service on demand, through a subscription, in a “pay-as-you-go” model, or (increasingly) at no charge when there is opportunity to generate revenue from streams other than the user, such as from advertisement or user list sales.

Functions as a Service (FaaS): A cloud computing service that provides a platform allowing customers to develop, run, and manage application functionalities without the complexity of building and maintaining the infrastructure typically associated with developing and launching an application. Also referred to as Serverless.

Platform as a Service (PaaS): Computing platform that allows the creation of web applications quickly and easily and without the complexity of buying and maintaining the software and infrastructure underneath it.

Container as a Service (CaaS): A cloud service that allows software developers and IT departments to upload, organize, run, scale, manage and stop containers by using container-based virtualization.

Infrastructure as a Service (IaaS): A way of delivering Cloud Computing infrastructure – servers, storage, network and operating systems – as an on-demand service. Rather than purchasing servers, software, datacenter space or network equipment, clients instead buy those resources as a fully outsourced service on demand.

On-Prem: A traditional data center hosting applications also known as a private cloud.

This design guide is focused on insertion of Cisco Security Controls in an Amazon Web Services and also covers the API gateway and APIs for orchestration, configuration, management, and monitoring.

API Gateway and Lambda discussed is a Serverless Architecture.
Shared Security Model

Public Cloud Security is a shared security model. Cloud provider and Customer is responsible for secure the following functions.

<table>
<thead>
<tr>
<th>Functions</th>
<th>Applications</th>
<th>Runtime</th>
<th>Middleware or Containers</th>
<th>Operating Systems</th>
<th>Virtualization</th>
<th>Servers</th>
<th>Storage</th>
<th>Networking</th>
</tr>
</thead>
</table>

| Customer Responsibility | Cloud Service Provider Responsibility |

AWS Services

AWS Virtual Private Cloud (VPC): AWS VPC is a logically isolated section of the AWS cloud. It provides complete control over the virtual networking environment, including selection of your customized IP address range, creation of subnets, and configuration of route tables and network gateways.

AWS Availability Zone (AZ): AWS locations are composed of Regions and Availability Zones. Each Region is a separate geographic area. Each Region has multiple, isolated locations known as Availability Zones.

AWS Internet Gateway (IGW): An internet gateway is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in your VPC and the internet.

AWS Elastic Cloud Compute (EC2): Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) cloud.

AWS Amazon Machine Image (AMI): An Amazon Machine Image (AMI) provides the information required to launch an instance. It is an image using which EC2 instance is deployed.

AWS Auto Scaling: AWS auto-scaling is a service that will provision additional EC2 instances automatically. It monitors telemetry information; when a certain threshold is hit, auto-scaling will provision another instance to support ever-increasing need.

AWS Security Group: AWS security group (SG) acts as a virtual firewall for your EC2 instance to control inbound and outbound traffic. AWS security group controls traffic on five-tuple information (source port, destination port, source ip, destination ip, and protocol).

AWS NAT gateway: AWS NAT gateway enables EC2 instances in a private subnet to connect to the internet or other AWS services, but prevent the internet from initiating a connection with those EC2 instances.

AWS VPN Gateway: AWS VPN connection/Gateway allows termination of IPSec Site-to-Site VPN tunnel to AWS Cloud. It supports IPSec route-based VPN.

AWS Direct Connect (DX): AWS Direct Connect is a cloud service solution that makes it easy to establish a dedicated network connection from premises to AWS. It provides private connectivity between AWS and datacenter, office, or colocation environment.
AWS PrivateLink: AWS PrivateLink simplifies the security of data shared with cloud-based applications by eliminating the exposure of data to the public Internet. AWS PrivateLink provides private connectivity between VPCs, AWS services, and on-premises applications, securely on the Amazon network.

AWS Application Load Balancer (ALB): AWS Application Load Balancer is best suited for load balancing of HTTP and HTTPS traffic and provides advanced request routing targeted at the delivery of modern application architectures, including microservices and containers. Operating at the individual request level (Layer 7), Application Load Balancer routes traffic to targets within Amazon Virtual Private Cloud (VPC) based on the content of the request.

AWS Network Load Balancer (NLB): AWS Network Load Balancer is best suited for load balancing of TCP, UDP, and TLS traffic where extreme performance is required. Operating at the connection level (Layer 4), Network Load Balancer routes traffic to targets within Amazon Virtual Private Cloud (Amazon VPC).

AWS Web Application Firewall (WAF): AWS WAF is a web application firewall that monitors HTTP/HTTPS requests forwarded to an Amazon API Gateway API, Amazon CloudFront, or an Application Load Balancer. AWS WAF also lets you control access to your content.

AWS Route53: AWS Route 53 is a highly available and scalable cloud DNS service. It is designed to integrate with AWS services.

AWS Simple Storage Service (S3): AWS Simple Storage Service (S3) is a service offered by AWS that provides object storage.

AWS Rational Database Service (RDS): AWS Rational Database Service (RDS) provides a highly scalable and cost-efficient relational database in the cloud. It is easy to set up, operated, and automate RDS service.

AWS Identity and Access Management (IAM): AWS Identity and Access Management (IAM) provides secure identity and access management services to AWS resources. You can create and manage users, groups, and use permission to allow and deny their access to AWS resources.

AWS API Gateway: Amazon API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale. APIs act as the "front door" for applications to access data, business logic, or functionality from your backend services. Using API Gateway, you can create RESTful APIs and WebSocket APIs that enable real-time two-way communication applications. API Gateway supports containerized and serverless workloads, as well as web applications.

AWS Lambda: AWS Lambda lets you run code without provisioning or managing servers. You pay only for the compute time you consume. With Lambda, you can run code for virtually any type of application or backend service - all with zero administration. Just upload your code, and Lambda takes care of everything required to run and scale your code with high availability. You can set up your code to automatically trigger from other AWS services or call it directly from any web app.

AWS Security Services

AWS Security Group: AWS security group (SG) acts as a virtual firewall for your EC2 instance to control inbound and outbound traffic. AWS security group controls traffic on five-tuple information (source port, destination port, source ip, destination ip, and protocol).
AWS WAF: AWS WAF is a web application firewall that helps protect web applications or APIs against common web exploits that may affect availability, compromise security, or consume excessive resources. AWS WAF gives control over how traffic reaches your applications by enabling you to create security rules that block common attack patterns, such as SQL injection or cross-site scripting, and rules that filter out specific traffic patterns you define.

AWS Shield (DDoS): AWS Shield Standard and Advanced protects against DDoS attacks. Standard Shield provides network flow monitoring protection against common DDoS attacks such as SYN floods, and UDP reflection attacks. AWS Shield Advanced provides expanded DDoS attack protection for your Amazon Elastic Compute Cloud instances, Elastic Load Balancing load balancers, Amazon CloudFront distributions, Amazon Route 53 hosted zones, and AWS Global Accelerator accelerators.

<table>
<thead>
<tr>
<th>Features</th>
<th>AWS Shield Standard</th>
<th>AWS Shield Advanced</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active monitoring</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network flow monitoring</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Automated application (layer 7)</td>
<td>-</td>
<td>✓</td>
</tr>
<tr>
<td>traffic monitoring</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DDoS mitigations</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Helps protect from common DDoS</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>attacks, such as SYN floods and</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UDP reflection attacks</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Access to additional DDoS mitigation capacity</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Visibility and reporting</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Layer 3/4 attack notification and</td>
<td>-</td>
<td>✓</td>
</tr>
<tr>
<td>attack forensic reports</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Layer 3/4/7 attack historical</td>
<td>-</td>
<td>✓</td>
</tr>
<tr>
<td>report</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DDoS response team support</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Incident management during</td>
<td>-</td>
<td>✓</td>
</tr>
<tr>
<td>high severity events</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Custom mitigations during attacks</td>
<td>-</td>
<td>✓</td>
</tr>
<tr>
<td>Post-attack analysis</td>
<td>-</td>
<td>✓</td>
</tr>
<tr>
<td>Cost protection</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reimburse related Route 53, CloudFront, and ELB DDoS charges</td>
<td>-</td>
<td>✓</td>
</tr>
</tbody>
</table>

AWS Shield Standard and AWS Shield Advanced

AWS Identity and Access Management (IAM): AWS Identity and Access Management (IAM) provides secure identity and access management services to AWS resources. You can create and manage users, groups, and use permission to allow and deny their access to AWS resources.

Cisco Security Portfolio

Cisco Tetration: Cisco Tetration offers holistic workload protection for multi-cloud data centers by enabling a zero-trust model using segmentation. This approach allows you to identify security incidents faster, contain lateral movement, and reduce your attack surface. Tetration is an infrastructure-agnostic approach that supports both on-premises and public cloud workloads. It provides key capabilities like
automated whitelist policies, zero trust model, software vulnerability detection, process behavior deviations, and controlled user access to applications.

Cisco Stealthwatch Cloud (SWC): Cisco Stealthwatch Cloud is the most comprehensive visibility and network traffic security analytics solution that uses telemetry from the existing network infrastructure. It provides advanced threat detection, accelerated threat response, and simplified network segmentation using multilayer machine learning and entity modeling. With advanced behavioral analytics, you’ll always know who is on your network and what they are doing. It is the only solution that detects threats across your private clouds, public clouds, and even in encrypted traffic.

Cisco Advanced Malware Protection for Endpoints: Cisco AMP for Endpoints offers cloud-delivered next-generation antivirus, endpoint protection platform (EPP), and advanced endpoint detection and response (EDR). It provides power protection engines for advanced investigation, endpoint isolation, retrospective security, and integration with wider Cisco Security products.

Cisco Defense Orchestrator (CDO): Cisco Defense Orchestrator (CDO) is a cloud-based multi-device manager you can use to manage security policy changes across various cisco security products and AWS security group. It cuts through complexity to save time and keep your organization protected against the latest threats. CDO provides one security policy, faster deployment, and smart configuration management.

Cisco Umbrella: Cisco Umbrella is a cloud-native platform that delivers the most secure, reliable, and fastest internet experience. It unifies firewall, secure web gateway, DNS-layer security, cloud access security broker (CASB), and threat intelligence solutions into a single platform to help secure network. As more organizations embrace direct internet access in the cloud, Umbrella makes it easy to extend protection to workloads in public cloud.

Cisco Threat Response (CTR): Cisco Threat Response automates integrations across select Cisco Security products and accelerates key security operations functions such as detection, investigation, and remediation. CTR provides key capabilities like incident tracking, seamless drill-down, direct remediation, automated enrichment, intuitive, and interactive visualization. It integrates with threat intelligence from Cisco Talos, and other third-party sources indicators of compromise (IOCs).

Cisco DUO Beyond: Duo Beyond expands secure access past traditional perimeter-based network security with the power to grant access to any application, to any user, from any device, while maintaining security hygiene. It helps to differentiate between corporate and personal devices, restrict sensitive data access to only corporate devices, limiting remote access to specific applications without exposing the network.

Radware DDoS and WAF: Radware’s Cloud Security Services offer easy-to-deploy cloud-based security that can be integrated with AWS environments to provide proactive, automated protection from advanced Threats. This cloud-based solution is similar to AWS WAF and AWS Shield; it provides WAF and DDoS protection for the AWS environment.

**AWS Three-tier Web App Reference Architecture**

AWS recommends three-tier architecture for web applications. These tiers are separated to perform various functions independently. Multilayer architecture for web applications has a presentation layer (web tier), an application layer (app tier), and a database layer (database tier). There is flexibility to make
changes to each tier independent of another tier. The application requires scalability and availability; the three-tier architecture makes scalability and availability for each tier independent.

AWS recommends running services in multiple availability zones and sharing traffic load using Elastic Load Balancers. To distribute traffic load, AWS recommends route53 that points to Application Load Balancer (ALB) for web tier. ALB load balances traffic across availability zones. Once traffic is processed by the web tier, then traffic is forward to Network Load Balancers (NLB). NLB load balances traffic across availability zones and forwards traffic to Application Servers. AWS recommends deploying the AWS RDS database that has high availability built into it. To enable resiliency and scalability, AWS recommends using autoscaling for each tier. These tiers are separated using AWS Security Groups and subnets.

In addition to the above recommendation, AWS recommends enabling the AWS web application firewall and AWS Shield (DDoS), Radware also provides WAF and DDoS functionality for AWS environment.

**Cisco Secure Three-Tier App Architecture for AWS**

Cisco provides a wide range of security options for protecting cloud resources. This design guide covers Cisco Tetration, Cisco Stealthwatch Cloud, Cisco AMP, Cisco DUO, Cisco Threat Response, and Cisco Defense Orchestrator.

When applications are moved to the cloud or deployed in the cloud, it is essential to maintain Data Center like segmentation, visibility, posture, and control. Cisco provides components that extend Cisco Security Portfolio in the AWS.
Cisco Secure Architecture for AWS three-tier Architecture

Cisco Tetration

The Cisco Tetration platform addresses cloud security challenges in a scalable way. Tetration facilitates workload protection in multi-cloud data centers by enabling capabilities like whitelisted segmentation, zero-trust model, behavioral change, and detection of software vulnerabilities by matching it against software package CVEs.

Tetration is available on-premise and as a SaaS-based offering, providing a wide variety of options from the data center, Co-Location, and Cloud.

- **Cisco Tetration SaaS (software-as-a-service)** – Cisco Tetration SaaS is used in this design guide.
- **Cisco Tetration-M (small form factor)**
- **Cisco Tetration (large form factor)**
- **Cisco Tetration-V (virtual form factor)**

Tetration receives data from various data sources like switches, load balancers, and Tetration agents. The Tetration agents are of the following types:

- Visibility Agent (provides only visibility)
- Enforcement Agent (provides visibility and enforcement)

Tetration Dashboard

Tetration shows critical information like vulnerability score, process health score, attach surface score, forensics score, network anomaly score, and segmentation compliance score.
Tetration in AWS

In this AWS three-tier architecture, each tier has EC2 in autoscaling. To enable the auto-provisioning of Tetration agents, we have added the required configuration in the AWS launch configuration. When the AWS auto-scale group deploys a new EC2 instance, it would automatically add the Tetration agent on the workload using the parameters given in the launch configuration.

Tetration agent on the new workload registers with the Tetration cloud (SaaS). It starts providing visibility, segmentation, behaviors deviation, and software vulnerability. Tetration ensures Cisco’s Zero Trust model.
Cisco Stealthwatch Cloud

Cisco Stealthwatch Cloud identifies malicious activities in your AWS workloads in real-time, so you can quickly respond before a security incident becomes a devastating breach. It provides essential security functions such as visibility, rapid threat detection, regulatory compliance.

**Visibility:** See and detect threats across your AWS workloads and an on-premises environment and view it all from a single interface.

**Rapid threat detection:** Using behavioral modeling, Stealthwatch Cloud detects threat activity, including malware and insider threats.

**Regulatory compliance:** You can easily monitor your organization for compliance with industry regulations such as the Payment Card Industry (PCI) standard, the Health Insurance Portability and Accountability Act (HIPAA), and the Federal Information Security Management Act (FISMA).

**Efficiency:** Continuously monitor and improve response times with automatic, high-fidelity alerts that make your security team more efficient.

**Scalability:** Stealthwatch Cloud automatically scales to your AWS environment, growing with your business, and reducing operational overhead.
Stealthwatch Cloud Dashboard

Stealthwatch Cloud Dashboard is a great way to track what is happening in the AWS cloud

![Cisco Stealthwatch Cloud Dashboard](image)

Stealthwatch Cloud in AWS

In this AWS three-tier architecture, Cisco Stealthwatch Cloud (SWC) receives VPC flow logs from AWS VPC that enables visibility, provides a platform for meeting all regulatory compliances. This solution increases efficiency by providing continuous monitoring and alerts. SWC is a scalable solution, and it scales automatically with a growing business.
Cisco Stealthwatch for AWS Three-tier Architecture

Cisco Stealthwatch Implementation is covered later in the design guide.

Cisco AMP for Endpoint

Cisco Advanced Malware Protection (AMP) blocks attacks and helps you respond to threats quickly and confidently. AMP provides the following key capabilities:

Unmatched protection engines: It uses anti-virus, file reputation, exploit prevention, machine learning, and a wide range of other attack prevention techniques to block known threats.

Faster Threat Hunting: AMP provides complex investigations using advanced search capabilities. It accelerates incident response and security investigations using predefined queries, forensic snapshots, live searches, and more.

Endpoint Isolation: It quickly stops threats spreading with one-click isolation for an infected endpoint.

Retrospective Security: AMP uses patented technology to analyze and monitor file and process activity continuously. It automatically generates retrospective alerts at the first sign of malicious behavior. AMP quickly stops threats in their tracks before they can cause any or further damage to your business.

Security Integration: AMP integrates with Cisco Threat Response (CTR) to accelerate threat investigation. AMP enables a zero-trust model using adaptive Duo multi-factor authentication.
AMP for Endpoints Dashboard

The AMP dashboard provides information about threats.

![AMP for Endpoints Dashboard](image)

Cisco AMP Endpoint Dashboard

AMP for Endpoint in AWS

In this AWS three-tier architecture, each tier has EC2 in autoscaling. To enable the auto-provisioning of AMP for endpoint agents, we have added the required configuration in the AWS launch configuration. When the AWS auto-scale group deploys a new EC2 instance, it would automatically add the "AMP for Endpoint - AMP4E" agent on the workload using the parameters given in the launch configuration.

AMP4E agent on the new workload registers with the AMP cloud (SaaS).
Cisco AMP for Endpoint for AWS Three-tier Architecture

Cisco AMP for Endpoint Implementation is covered later in the design guide.

Cisco Umbrella

Cisco Umbrella offers flexible cloud-delivered security. It combines multiple security functions into one solution. Cisco Umbrella solutions provide DNS-layer security, secure web gateway, cloud-delivered firewall, cloud access security broker (CASB), and interactive threat intel. This document covers Umbrella DNS-layer protection for workloads in AWS VPC.

Umbrella Dashboard

Cisco Umbrella security overview provides critical information such as total DNS requests, total blocks, security blocks, and total proxy requests.
Umbrella Virtual Appliance (VA) in AWS

In AWS three-tier architecture, each availability zone has Cisco Umbrella virtual appliance (VA). Both VAs are defined as primary and secondary DNS on the workloads using DHCP-option attached to VPC. In the event of the load, when the AWS auto-scale decides to instantiate a new EC2 instance in the scale-group, the new EC2 instance would automatically point to Umbrella Virtual Appliance. Umbrella provides DNS-layer security to workloads in a VPC. Secondary Umbrella VA receives DNS requests if primary Umbrella VA is not available.
Cisco Umbrella Implementation is covered later in the design guide.

Cisco Duo Beyond

Cisco Duo provides secure access to applications and data, no matter where the users are, on any device, and from anywhere. Cisco Duo’s secure access solution creates trust in users, devices, and the applications they access. Reduce the risk of a data breach and ensure trusted access to sensitive data. Cisco Duo Beyond provides the following functions:

**Multi-Factor Authentication:** Verify the identity of all users with Duo’s strong two-factor authentication.

**Single Sign-on:** Seamless, single dashboard access to any and all applications.

**Remote Access:** Secure access to cloud and on-premises applications and servers, with or without VPN.

**Device Trust:** Check that your users’ devices meet your security standards before granting them access.

**Adaptive Access Policies:** Set policies to grant or block access attempts by user or device, based on contextual factors.

In this design guide, we have used Multi-Factor authentication (MFA) for cloud application.

Cisco Duo Beyond Dashboard

Cisco Duo Beyond dashboard provides information about users and endpoints.
In this AWS three-tier architecture, we have used Duo Beyond for Multi-factor authentication. From a security point of view, when a user would access the cloud application, the customer would be challenged to provide authentication using Duo beyond MFA.
Cisco Defense Orchestrator

Cisco Defense Orchestrator (CDO) is a cloud-based multi-device manager you can use to manage security policy changes across various Cisco security products and AWS security group. It cuts through complexity to save time and keep your organization protected against the latest threats. CDO provides one security policy, faster deployment, and smart configuration management. CDO eliminates the time-consuming complexity of managing policies across distributed AWS security groups in a multi-vpc environment.

Cisco Defense Orchestrator helps to correct issues such as:
- Unused objects
- Duplicate objects
- Inconsistent objects

Cisco Defense Orchestrator Devices and Services

CDO Devices and Services shows onboarded AWS VPC in Cisco CDO. To onboard VPC, you need an AWS username and access key.

Cisco Defense Orchestrator for AWS

AWS security groups are attached to the workload Elastic Network Interface (ENI) and act as the first line of defense by enforcing layer-4 rules. In the real world, managing security groups can be complicated when you are dealing with multiple VPCs or hundreds of workloads. CDO can now onboard VPCs and control security groups created in the VPCs. Managing VPCs using CDO helps administrators to organize their security groups, modify, create, and update rules. To onboard and manage VPC using CDO, you need AWS username and access key (for detail checkout managing AWS security group using CDO).
Cisco Threat Response

Threat Response accelerates investigations by automating and aggregating threat intelligence and data across your security infrastructure in one unified view.

Aggregated threat intelligence: Cisco Threat Response integrates threat intelligence from Cisco Talos and third-party sources to automatically research indicators of compromise (IOCs) and confirms threats quickly.

Automated enrichment: Cisco Threat Response adds context from integrated Cisco Security products automatically so you know instantly which of your systems was targeted and how.

Incident tracking: Cisco Threat Response provides the capability you need to collect and store key investigation information, and to manage and document your progress and findings.

Cisco Threat Response Dashboard

Cisco CTR Investigate tab helps in investigating an incident.
Cisco Threat Response Investigation

Cisco Threat Response in AWS

In this AWS three-tier architecture, we are receiving information from Umbrella and AMP for Endpoint to provide threat intelligence, contextual approach, and threat hunting.
Cisco Threat Response for AWS Three-tier Architecture

Cisco CTR integration is covered later in the design guide.

Amazon Web Services – Security

AWS Security Group (SG)

A security group acts as a virtual firewall for your EC2 instance; it controls inbound and outbound traffic. When you launch an instance, you can assign up to five security groups to the instance. It is applied at the interface level, not at the subnet level.
In this AWS three-tier architecture, east-west traffic between tiers is controlled by AWS security groups. Cisco Defense Orchestrator is controlling the AWS security group. [Cisco CDO VPC onboarding is covered later in the design guide.](#)

### AWS Web Application Firewall (WAF)

AWS offers a web application firewall as a service that controls access to content by allowing or blocking web requests. It protects web applications from common web exploits. In addition to web traffic protection, AWS WAF can also provide protection for APIs against common web exploits that may affect availability.
Implementation steps for enabling AWS WAF are covered later in the design guide.

AWS Shield (DDoS)

AWS Shield Standard and Advanced protects against DDoS attacks. Standard Shield provides network flow monitoring protection against common DDoS attacks such as SYN floods, and UDP reflection attacks. AWS Shield Advanced provides expanded DDoS attack protection for your Amazon Elastic Compute Cloud instances, Elastic Load Balancing load balancers, Amazon CloudFront distributions, Amazon Route 53 hosted zones, and AWS Global Accelerator accelerators.

Radware Web Application Firewall and DDoS

Amazon Web Services (AWS) has become a commonplace to host critical applications and make these applications available to users (internal or external). As a result, it is essential to ensure these applications receive the same levels of protection from distributed denial of service (DDoS) and advanced web attacks that on-premises applications do.

Radware’s Cloud Security Services offer easy-to-deploy cloud-based security that can be integrated with your AWS environments to provide proactive, automated protection from advanced threats.

Effective DDoS protection is a must for any web application or website today. In the past, organizations choosing to host these applications or sites in AWS had to make a trade-off: sacrifice levels of security for the convenience and cost benefits of AWS. Radware provides applications hosted on AWS with the widest protection from the full breadth of DDoS attacks with real-time mitigation and no added latency in peacetime.

Radware acts as a man in the middle, DNS points to Radware Service. Traffic is received by Radware SaaS and they perform security operations on the traffic. Post inspection traffic is forwarded to AWS.
AWS shield provide protection agent DDoS attacks
Traffic Flow

This section explains various traffic flow in AWS VPC and provides Cisco’s recommendation for securing various traffic flows.

Application to Internet (app-to-internet)

Outbound to internet connection is essential for downloading systems updates and patch for workloads running in the cloud.

Security recommendations for “app-to-internet” traffic flow:

This traffic flow requires additional security because internet connection is involved.

- Place application servers in private subnet (no internet connection)
- Enable outbound internet connection only (use AWS NAT gateway in your public subnet). AWS outbound traffic to AWS NAT gateway using the AWS route table
- NAT gateway should forward traffic through the AWS internet gateway
- Protect your host using the AWS security group managed by Cisco Defense Orchestrator (allow only required ports)
- For workload security, add the Cisco Tetration enforcement agent on the workload. If your workloads are in the auto-scale group, ensure to add Cisco Tetration agent as part of the launch configuration
- For endpoint security, add the Cisco AMP4E agent on the workload. If your workloads are in the auto-scale group, ensure to add AMP4E agent as part of the launch configuration
- Enable Stealthwatch Cloud for visibility of that is happening in the AWS VPC
- Enable Cisco’s first line of defense using Cisco Umbrella (DNS) to protect your workload
- If your workload has to be accessed by users, enable Cisco Duo Beyond for Multi-factor Authentication
User to Application

Users would access applications hosted in the cloud from the Internet, Branch, Campus, and using VPN.

Security recommendations for “user-to-app” traffic flow:
- Protect your host using the AWS security group managed by Cisco Defense Orchestrator (allow only required ports)
- For workload security, add the Cisco Tetration enforcement agent on the workload. If your workloads are in the auto-scale group, ensure to add Cisco Tetration agent as part of the launch configuration
- For endpoint security, add the Cisco AMP4E agent on the workload. If your workloads are in the auto-scale group, ensure to add AMP4E agent as part of the launch configuration
- Enable Stealthwatch Cloud for visibility of that is happening in the AWS VPC
- Enable Cisco’s first line of defense using Cisco Umbrella (DNS) to protect your workload
- If your workload has to be accessed by users, enable Cisco Duo Beyond for Multi-factor Authentication
- Enable Cisco Threat Response for AMP4E installed on the workload for threat hunting.
- Enable WAF and DDoS solution:
  - Enable Radware (WAF and DDoS) solution
  - Enable AWS WAF and Shield (WAF and DDoS)
East/West Traffic Flow

Traffic between tiers, example web tier talking to app tier.

Security recommendations for “east/west” traffic flow:

- Protect your host using the AWS security group managed by Cisco Defense Orchestrator (allow only required ports)
- For workload security, add the Cisco Tetration enforcement agent on the workload. If your workloads are in the auto-scale group, ensure to add Cisco Tetration agent as part of the launch configuration
- For endpoint security, add the Cisco AMP4E agent on the workload. If your workloads are in the auto-scale group, ensure to add AMP4E agent as part of the launch configuration
- Enable Stealthwatch Cloud for visibility of what is happening in the AWS VPC
- Enable Cisco’s first line of defense using Cisco Umbrella (DNS) to protect your workload
- If your workload has to be accessed by users, enable Cisco DUO for Multi-factor Authentication
- Enable Cisco Threat Response for AMP4E installed on the workload for threat hunting
- Enable WAF and DDoS solution:
  - Enable Radware (WAF and DDoS) solution
  - Enable AWS WAF and Shield (WAF and DDoS)
- Use AWS load balancers to share load
App-to-app Traffic Flow

Application workload may talk to another application server in the same subnet.

Security recommendations for "app-to-app" traffic flow:

- Protect your host using the AWS security group managed by Cisco Defense Orchestrator (allow only required ports)
- For workload security, add the Cisco Tetration enforcement agent on the workload. If your workloads are in the auto-scale group, ensure to add Cisco Tetration agent as part of the launch configuration
- For endpoint security, add the Cisco AMP4E agent on the workload. If your workloads are in the auto-scale group, ensure to add AMP4E agent as part of the launch configuration
- Enable Stealthwatch Cloud for visibility of that is happening in the AWS VPC
- Enable Cisco Threat Response for AMP4E installed on the workload for threat hunting
- Enable WAF and DDoS solution:
  - Enable Radware (WAF and DDoS) solution
  - Enable AWS WAF and Shield (WAF and DDoS)
- Use AWS load balancers to share load
API client to APP

NetOPS and SecOPS teams may use API client to read and write information on the application servers.

Security recommendations for “apiclient-to-app” traffic flow:

- Protect your host using the AWS security group managed by Cisco Defense Orchestrator (allow only required ports)
- Enable Cisco’s first line of defense using Cisco Umbrella (DNS) to protect your workload. API clients should use DNS records in the scripts instead of IP addresses
- Enable Cisco DUO Beyond to add multi-factor authentication for API communication
AWS API Gateway to APP:

NetOPS and SecOPS teams may use AWS API GW to read and write information on the application servers.

Security recommendations for "api-gw-to-app" traffic flow:

- Protect your host using the AWS security group managed by Cisco Defense Orchestrator (allow only required ports)
- Enable Cisco’s first line of defense using Cisco Umbrella (DNS) to protect your workload. API clients should use DNS records in the scripts instead of IP addresses
- Enable Cisco DUO to add multi-factor authentication for API communication
AWS DevOps to APP:

DevOps team may use API GW to read and write information on the application servers.

Security recommendations for “DevOps-to-app” traffic flow:

- Protect your host using the AWS security group managed by Cisco Defense Orchestrator (allow only required ports)
- Enable Cisco’s first line of defense using Cisco Umbrella (DNS) to protect your workload. API clients should use DNS records in the scripts instead of IP addresses
- Enable Cisco DUO to add multi-factor authentication for API communication
Design Implementation

Now that we have established the design specifics of our three-tier application in AWS cloud, we will begin implementing and setting up the test environment. We start by setting up the VPC as per the three-tier architecture specifications and integrate Stealthwatch Cloud and CDO to it. Once our VPC is ready, we go on to bring up the load balancers, secure database and application workloads, and eventually access our newly deployed Secure Cloud application. We will also set up an API service for our newly deployed cloud application.

NOTE: Please ensure you have appropriate privileges to create all the VPC components. Follow the AWS Documentation for more information on IAM permissions and policies.

Setting up AWS Virtual Private Cloud (VPC)

In this section, we will create a new AWS VPC and configure all the associated components that we need for our deployment. We will also perform the Stealthwatch and CDO cloud integration with this newly created VPC.

Create the AWS VPC and associated Components

Implementation procedure:
Step 1: Set up the VPC.
Step 2: Define the subnets.
Step 3: Update DHCP options.
Step 4: Set up the internet and NAT gateways.
Step 5: Set up the routing tables.
Step 6: Create the security groups.

Step 1: Set up the VPC - Log on to the AWS console and select the VPC service, click on 'Create VPC' and fill in the required details. We chose the IPV4 CIDR block as 10.0.0.0/16.

Follow the AWS documentation for more details on AWS VPCs.
### Step 2: Define Subnets

- Based on the tiered architecture, we defined two subnets (one subnet in each AWS availability zone) for each tier: web, application, database and management.

<table>
<thead>
<tr>
<th>IPV4 CIDR Block</th>
<th>AWS Region</th>
<th>Tier</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0.2.0/24</td>
<td>US-East-2a</td>
<td>WebSubnet1a</td>
</tr>
<tr>
<td>10.0.3.0/24</td>
<td>US-East-2b</td>
<td>WebSubnet1b</td>
</tr>
<tr>
<td>10.0.4.0/24</td>
<td>US-East-2a</td>
<td>AppSubnet1a</td>
</tr>
<tr>
<td>10.0.5.0/24</td>
<td>US-East-2b</td>
<td>AppSubnet1b</td>
</tr>
<tr>
<td>10.0.6.0/24</td>
<td>US-East-2a</td>
<td>DbSubnet1a</td>
</tr>
<tr>
<td>10.0.7.0/24</td>
<td>US-East-2b</td>
<td>DbSubnet1b</td>
</tr>
<tr>
<td>10.0.8.0/24</td>
<td>US-East-2a</td>
<td>MgmtSubnet1a</td>
</tr>
<tr>
<td>10.0.9.0/24</td>
<td>US-East-2b</td>
<td>MgmtSubnet1b</td>
</tr>
</tbody>
</table>

Go to ‘VPC Dashboard > Subnets’ and create all these subnets and name them appropriately.

### Step 3: Update DHCP Options

- Go to ‘VPC Dashboard > DHCP Options Sets’ and create a new DHCP options set. Set the domain name servers to two IPs of your choice in the MgmtSubnet1a and MgmtSubnet1b subnet, the IPs chosen were - 10.0.8.100 and 10.0.9.100. We will use these two IPs to deploy Cisco Umbrella virtual appliances which will act as DNS providers for our environment and will also extend the DNS Security capability.

Go to ‘VPC Dashboard > Your VPCs’, select the newly created VPC above and update the DHCP options set from the drop-down list.
Follow the [AWS documentation](https://aws.amazon.com) for more details on DHCP Options Sets.

**Step 4: Set up the internet and NAT gateways** - Create an Internet gateway for providing internet access to all the resources in the VPC. We use the internet gateway as the next hop for default routes in web and management tier.

Also, create two NAT gateways in webSubnet1a and webSubnet1b, one for each region. We use the NAT gateways to provide outbound only access to private resources in application and database tier.

Follow the [AWS documentation](https://aws.amazon.com) for more details on internet and NAT gateway components.

**Step 5: Set up the routing tables** - Go to ‘VPC Dashboard > Route table’ and create the routing tables for each tier. There is one routing table for each tier except the application and database tier. Since the application and database tier only need outbound public access, the regional ‘NAT gateway’ AWS service was used and hence the application and database routing tables were split to one per region.
For each newly created routing table, click on ‘Subnet Association’ and based on our design, associate the subnets to each routing table.

For the web and management routing tables, create the default route pointing to internet gateway. For the application and database routing tables, create the default routes pointing to the NAT gateway corresponding to each region.
Step 6: Create the security groups  - Go to ‘VPC Dashboard > Security Groups’, set up a security group corresponding to each tier in the design. Set up the inbound access rules as per your application requirements. We used the following inbound rules.

### webSG

<table>
<thead>
<tr>
<th>Port</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP port 80</td>
<td>Allow HTTP access from internet to web servers</td>
</tr>
<tr>
<td>TCP port 443</td>
<td>Allow HTTPS access from internet to web servers</td>
</tr>
<tr>
<td>TCP port 22</td>
<td>Allow SSH access from mgmtSG</td>
</tr>
</tbody>
</table>

### appSG

<table>
<thead>
<tr>
<th>Port</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP port 80</td>
<td>Allow HTTP access from workloads within application tier subnets (allows load balancer health checks). Allow HTTP access from web tier subnets. **</td>
</tr>
<tr>
<td>TCP port 22</td>
<td>Allow SSH access from mgmtSG</td>
</tr>
</tbody>
</table>

### dbSG

<table>
<thead>
<tr>
<th>Port</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP port 3306</td>
<td>Allow MYSQL/Aurora traffic from appSG</td>
</tr>
</tbody>
</table>

### mgmtSG

<table>
<thead>
<tr>
<th>Port</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP port 22</td>
<td>Allow SSH access from internet</td>
</tr>
<tr>
<td>UDP 53</td>
<td>Allow DNS traffic from appSG, webSG, dbSG and mgmtSG</td>
</tr>
</tbody>
</table>

**NOTE:** ** AWS elastic ‘Network Load Balancer’ that we use for application workloads preserves the source IP of incoming connections from web tier workloads, hence we need to allow the web subnets specifically. We cannot use security groups to allow traffic from NLB, we must use subnets. Follow the AWS Documentation for more information on security group requirements for NLB.
For outbound access, we allowed unfiltered access to each tier (In the later section of this guide, the Tetration enforcement agent was used to lockdown outbound access on workloads).

Follow the AWS documentation for more details on AWS security groups.

Integrate Stealthwatch Cloud and Cisco Defense Orchestrator (CDO)
Implementation procedure:
Step 1: Integrate the Cisco Stealthwatch Cloud.
Step 2: Integrate the Cisco Defense Orchestrator.

Step 1: Integrate Cisco Stealthwatch Cloud - Follow the steps illustrated in Cisco Stealthwatch AWS Quick Start Guide to create the VPC flow logs and push them to Stealthwatch cloud to be analyzed. Based on this VPC flow log information, Stealthwatch would generate security alerts and observations on any activity within the VPC.
Step 2: Integrate the Cisco Defense Orchestrator - Follow the steps illustrated in CDO Documentation to onboard the AWS VPC. Once the onboarding is complete, CDO can be used to manage the AWS security groups in a much more efficient way.

Setting up Load Balancers

In this section, we will explore setting up the two types of AWS elastic load balancer services i.e. ‘Application Load Balancer (ALB)’ and ‘Network Load Balancer (NLB)’. Before we begin our implementation, please ensure you’re familiar with these components. For information on AWS Elastic Load Balancing and health checks, check out the AWS documentation here. For understanding the configuration steps for application load balancer and network load balancer, follow the links.

Per our tiered design, we will set up an ALB for the web server workloads and an NLB for the application workloads. We will create target groups for each load balancer, the workloads register themselves with these target groups. We will not register any instances to the target groups at this point but in the next section when we create the auto-scaling groups, we will integrate the auto-scaling groups with each of these blank target groups that we create in this section. When auto-scaling process spins new instances, they will automatically register with these target groups.

The load balancer will be configured to run health checks for each instance that is launched into a target group. As soon as an instance is marked healthy, the load balancer starts load balancing traffic to it. When the instance becomes unhealthy, it is removed from the pool.

Setting up frontend ‘ALB’ and backend ‘NLB’

Implementation procedure:

Step 1: Set up frontend ‘Application Load Balancer’.
Step 2: Set up backend ‘Network Load Balancer’.

Step 1: Set up frontend ‘Application Load Balancer’ - The ‘internet-facing’ application load balancer is used to load balance web traffic coming from the internet to a pool of web servers. This load balancer is placed in the ‘webSG’ security group in the subnets – ‘WebSubnet1a’ and ‘WebSubnet1b’. A new target group ‘WebServerPool’ was also created as part of load balancer configuration, this will be used later while setting up autoscaling group for web servers.
Step 2: Set up backend ‘Network Load Balancer’ - The ‘internal’ network load balancer is used to load balance traffic from the pool of web servers to a pool of application servers. This load balancer is placed in the ‘appSG’ security group in the subnets – ‘appSubnet1a’ and ‘appSubnet1b’. A new target group ‘AppServerPool’ was also created as part of load balancer configuration, this will be used later while setting up autoscaling group for application servers.

Setting Up the application

In this section of the guide, we will work on setting up the tiered AWS application workloads. An overview of packages and utilities used for lab testing are as below.

- We have the Umbrella VAs (DNS providers) sitting in the management subnets
- We have an auto scaled pool of NGINX web servers hosted in the web tier
- We have an auto scaled pool of workloads with ‘WordPress’ app code in the app tier
- An instance of MySQL RDS was used in the database tier

Setting up Cisco Umbrella virtual appliances

Implementation procedure:

**Step 1:** Set up the Umbrella virtual appliances.
**Step 2:** Configure the local DNS on Umbrella virtual appliances (VA).
**Step 3:** Set up policies to exempt internal domains.

**Step 1: Set up the Umbrella virtual appliances** - Follow the Umbrella [documentation to deploy Virtual appliances (VA)](https://example.com). As per documentation, Define the AMI image and then use it to launch VA instances. We had reserved the IP addresses - 10.0.8.100 and 10.0.9.100 for these virtual appliances, assign these static IP addresses to the VAs. These appliances were placed in management tier.

Once the appliances are fully up in AWS, login to the Umbrella portal and verify the green status under ‘Deployments > Configuration > Sites and Active Directory’.
Step 2: Configure the local DNS on Umbrella virtual appliances - Configure local DNS on the VAs ([Umbrella documentation](#)). Based on the CIDR block chosen for lab VPC, the second IP address i.e. 10.0.0.2/24 is the local DNS, set this IP as local DNS on both Umbrella VAs.

Step 3: Set up policies to exempt internal domains - Log on to the Umbrella portal, go to ‘Deployments > Configuration > Domain Management’ and add the internal Amazon domains in the exempt list. Based on your set up, the list of internal domains will vary.

Setting up database
Implementation procedure:
Step 1: Define RDS Subnet groups.
Step 2: Set up the RDS database instance.

**Step 1: Define RDS Subnet groups** - Go to ‘RDS > Subnet Groups’ on the AWS console, create a ‘DB Subnet Group’ and add to it the database subnets defined in previous steps (‘dbSubnet1a’ and ‘dbSubnet1b’)

**Step 2: Set up the RDS database instance** - Set up the database instance as per your application requirements, follow the AWS documentation for further help. Use the Subnet Group defined in Step 1 above. The database instance is placed in security group ‘dbSG’.

Note: For the lab purpose, once the RDS instance was up, we recorded the username, password, endpoint hostname and port. We set up a jump server in management tier and used it as SQL client to log into the RDS instance and create a MySQL database for our test application. These details were later used while preparing up the test application code.

**Setting up web servers and app servers**

Implementation procedure:
Step 1: Set up Launch Configurations.
Step 2: Set up Auto Scaling Groups.
Step 3: Configure the scaling policies.

Step 1: Set up Launch Configurations - Go to ‘EC2 Dashboard> Auto Scaling > Launch Configuration’ and create launch configurations for web and application servers. For more information on creating ‘Launch Configurations’ follow the AWS documentation. Use the security groups ‘webSG’ and ‘appSG’ for web and application launch configurations respectively. Make sure not to assign any public IPs to the instances in the application launch configuration.

Under the advanced options, we used the ‘user data’ option to initialize the EC2 instances when they are launched into the auto scaling pool. For more details on ‘user data’ option, check out the AWS documentation. As part of this initialization process, we perform the following tasks:

- Install required packages (ex – nginx, httpd, ipset etc.) on the workloads
- Download configuration files/code for the respective workloads
  - For web server, we hosted the modified nginx configuration file in an S3 bucket with appropriate privileges
  - For application server, we used publicly available ‘WordPress’ blog code. We made modifications to integrate it with the database created in previous steps and hosted the code in an S3 bucket with appropriate privileges
- Download and install the Tetration enforcement agent
- Download and install the AMP4E agent

NOTE: We also downloaded and installed the DUO plugin (available for ‘WordPress’ blogs) to the application workloads. If you choose to include the Duo integration in your native application, please follow the DUO Web SDK documentation. Once the complete design is implemented and application is accessible i.e. the last step of this section, follow the Duo Documentation (start from Step 4 in the Duo doc) to enable WordPress Duo plugin.

The user data configuration used for this testing is as below, this will vary based on your base instance images, application and other such variables.

**Web Server User Data:**

#!/bin/bash
sudo yum install -y wget
sudo yum install -y unzip
sudo yum install -y ipset       // ipset utility is required for enforcing tetration policies
sudo yum install -y nginx      // Installing nginx

# Setting up the nginx web server and updating it with S3 hosted nginx configuration file.
sudo mv /etc/nginx/nginx.conf /etc/nginx/nginx.conf.backup
sudo wget https://safefiles.s3.us-east-2.amazonaws.com/nginx.conf
sudo systemctl restart nginx
sudo systemctl enable nginx

#Downloading the Tetration enforcement agent hosted in an S3 bucket and installing it.
sudo yum install -y lsof
sudo chmod 755 tetration_installer_intgssopov_enforcer_linux.sh
sudo ./tetration_installer_intgssopov_enforcer_linux.sh --skip-pre-check

#Downloading the AMP4E agent hosted in an S3 bucket and installing it.
sudo wget https://safefiles.s3.us-east-2.amazonaws.com/cisco.gpg
sudo rpm --import ./cisco.gpg
sudo wget https://safefiles.s3.us-east-2.amazonaws.com/AWS_rhel-centos-7fireamplinux_connector.rpm
sudo yum install -y AWS_rhel-centos-7fireamplinux_connector.rpm

**Application Server User Data:**
```bash
#!/bin/bash
sudo yum install -y wget
sudo yum install -y unzip
sudo yum install -y ipset // ipset utility is required for enforcing tetration policies
sudo yum install -y httpd // Installing httpd

#Setting up the HTTPD server and downloading the application code and Duo plugin hosted in an S3 bucket
sudo systemctl start httpd
sudo systemctl enable httpd
sudo unzip /var/www/html/wordpresscodefile.zip
sudo wget https://downloads.wordpress.org/plugin/duo-wordpress.2.5.4.zip -P /var/www/html/wp-content/plugins
sudo unzip /var/www/html/wp-content/plugins/duo-wordpress.2.5.4.zip

cd /home/centos/
sudo systemctl restart httpd

#Downloading the Tetration enforcement agent hosted in an S3 bucket and installing it.
sudo yum install -y lsof
sudo chmod 755 tetration_installer_intgssopov_enforcer_linux.sh
sudo ./tetration_installer_intgssopov_enforcer_linux.sh --skip-pre-check

#Downloading the AMP4E agent hosted in an S3 bucket and installing it.
sudo wget https://safefiles.s3.us-east-2.amazonaws.com/cisco.gpg
sudo rpm --import ./cisco.gpg
sudo wget https://safefiles.s3.us-east-2.amazonaws.com/AWS_rhel-centos-7fireamplinux_connector.rpm
sudo yum install -y AWS_rhel-centos-7fireamplinux_connector.rpm
```
**Step 2: Set up Auto-Scaling Groups** - Create two auto-scaling groups using the launch configuration created in the previous step, one for web servers and another one for application servers. For more information on creation of auto-scaling groups, follow the [AWS documentation](#).

Once the auto-scaling groups are created, select each group and click on edit. In the edit menu, update the target groups and health check types. For web server auto-scaling group, set the target group to ‘WebServerPool’ as created during the load balancer set up. Also, update the health check type to ‘ELB’ to integrate the auto scaling group with load balancers. Repeat the same steps for the application server auto-scaling group.
Step 3: Configure the scaling policies - On the ‘auto scaling groups’ page, select each group and click on ‘Scaling policies’ tab to add the scaling policies. For testing purposes, we used simple scaling of adding or removing 1 instance when average CPU Utilization exceeds 90% or remains below a minimum value of 10%. The desired state was set to two instances at a given point.

Click on ‘Add policy’ and select ‘create a simple scaling policy’. Fill in the policy name, alarm (you will need to create a new alarm) and action as per the requirement.

Follow the AWS documentation for more further details on scaling policies.

Accessing the application
Implementation procedure:
Step 1: Register a domain using AWS Route 53 service.
Step 2: Access the web application.
Step 1: Register a domain using AWS Route 53 service - On the ‘Route 53’ service dashboard on AWS console, click on ‘Create Hosted Zone’ to register a domain for the application. The registration process might take anywhere from few minutes to few hours to complete.

Once it’s completed, click on the domain name and create a record set for the application. Under the record set, give an appropriate name to the access URL and use the ‘Alias’ option to point it to ALB that we set up in previous steps. It might take another few minutes for the DNS mapping to propagate to the name servers. Follow the AWS documentation for detailed steps.

Step 2: Access the web application - Go to this newly registered domain URL in the browser, you will be prompted to the initial application setup, after the initial set up the application home page should load as below.
Concatenate ‘/wp-admin’ to URL above to login to the admin portal of the application.

Setting up the API gateway

Now that we have completed and verified the deployment of our application, we will create an API gateway. We used the AWS ‘API gateway’ service to provide the interface to the end users and AWS ‘Lambda’ service to define the API handler code which interacts with the RDS database instance.

Creating Lambda Function

Implementation procedure:

Step 1: Create a lambda function
Step 2: Set up the VPC parameters for lambda function.
Step 3: Set up the lambda handler function
Step 1: Create a lambda function - Go to the AWS console and select the ‘Lambda’ service. Create a Lambda function from scratch, specify a name for the function and we chose Python as runtime environment. Runtime will vary based on your set up.

Step 2: Set up the VPC parameters for lambda function - Within the Lambda function parameters, select the VPC as ‘SecureVPC’ that we created in first section of design implementation and set the subnets to application tier subnets. The security group would be set to ‘appSG’.
Step 3: Set up the lambda handler function - Lastly, define the ‘Lambda’ handler function. The code would fetch information from the RDS instance based on inputs received from the API gateway service (to be set up in the next section). Follow the [AWS documentation](https://aws.amazon.com) for detailed steps on how to create ‘Lambda’ function for RDS integration.

![Lambda handler function](image)

Configure the test event to pass the required parameters to the Lambda handler function and test it. We passed the parameter ‘posts’ to fetch the list of published blogs on our application. We can see the Lambda function is successfully returning the desired results.

![Configure test event](image)
Create the API gateway

Implementation procedure:
Step 1: Create an API gateway.
Step 2: Create a custom API domain.
Step 3: Configure Route53 for custom API domain.

Step 1: Create an API gateway - Select the ‘API Gateway’ service from the AWS console and create a new REST API gateway. We chose regional API endpoint. Please follow the AWS documentation for detailed information about API gateway service.

Once the API gateway is created, we update the methods and integration request/responses. In the ‘Integration Request’ section, we specified the Lambda function that we created in previous step.
Step 2: Create a custom API domain - Now that the API gateway is set up, we want to create a user-friendly domain name to access it. Under the API gateway service console, select 'Custom Domain names' and follow the AWS documentation to create a custom API URL, say 'api.ciscosafeapp.com'.
Step 3: Configure Route53 for custom API domain  - Go back Route 53 console and add a record set of type ‘A’ to the Route53 hosted zone created in previous steps. Under record set settings, select API gateway URL from the drop-down list under ‘Alias’ option.

Adding WAF and DDOS Capabilities

At this point we have finished setting up a functional application and an API gateway for it, we will add the WAF and DDOS prevention capabilities in our design. For the purpose of this paper, we will demonstrate two different options for adding these capabilities, you can pick any of these based on your preference:

- **OPTION 1**: AWS WAF and Shield Service
- **OPTION 2**: Radware Cloud Service

Option 1: AWS WAF and Shield

Implementation procedure:
Step 1: setting up AWS WAF.
Step 2: setting up AWS Shield Standard.

Step 1: setting up AWS WAF - Follow the six step process elaborated in AWS documentation to set up a Web ACL using AWS WAF & Shield service. We used the AWS managed rule set for this lab, but you can customize the rules as per your need.

We added both API gateway and the frontend ALB to the WAF to be protected against any sort of malicious web activity.

As the traffic starts flowing from the internet to the application and API gateway, WAF is actively detecting and blocking the malicious scans and attack attempts from the internet.
### Sampled requests
Samples of requests from the past 3 hours.

<table>
<thead>
<tr>
<th>Source IP</th>
<th>URI</th>
<th>Matches rule</th>
<th>Action</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>135.9.147.100</td>
<td>/wp-login.php</td>
<td>AWS\AWSM\ManagedRules\CommonRuleSet#GenericRFI_BODY</td>
<td>BLOCK</td>
<td>Wed Feb 19 2020 15:27:35 GMT-0500 (Eastern Standard Time)</td>
</tr>
<tr>
<td>52.188.192.200</td>
<td>/wp-login.php</td>
<td>AWS\AWSM\ManagedRules\CommonRuleSet#GenericRFI_BODY</td>
<td>BLOCK</td>
<td>Wed Feb 19 2020 15:27:35 GMT-0500 (Eastern Standard Time)</td>
</tr>
<tr>
<td>34.70.220.210</td>
<td>/xmlrpc.php</td>
<td>AWS\AWSM\ManagedRules\CommonRuleSet#CrossSiteScripting_BODY</td>
<td>BLOCK</td>
<td>Wed Feb 19 2020 15:27:35 GMT-0500 (Eastern Standard Time)</td>
</tr>
<tr>
<td>103.81.84.173</td>
<td>/wp-login.php</td>
<td>AWS\AWSM\ManagedRules\CommonRuleSet#GenericRFI_BODY</td>
<td>BLOCK</td>
<td>Wed Feb 19 2020 15:27:35 GMT-0500 (Eastern Standard Time)</td>
</tr>
<tr>
<td>45.32.126.7</td>
<td>/wp-login.php</td>
<td>AWS\AWSM\ManagedRules\CommonRuleSet#GenericRFI_BODY</td>
<td>BLOCK</td>
<td>Wed Feb 19 2020 15:27:35 GMT-0500 (Eastern Standard Time)</td>
</tr>
<tr>
<td>159.65.83.36</td>
<td>/wp-login.php</td>
<td>AWS\AWSM\ManagedRules\CommonRuleSet#GenericRFI_BODY</td>
<td>BLOCK</td>
<td>Wed Feb 19 2020 15:27:35 GMT-0500 (Eastern Standard Time)</td>
</tr>
<tr>
<td>103.198.52.10</td>
<td>/wp-login.php</td>
<td>AWS\AWSM\ManagedRules\CommonRuleSet#GenericRFI_BODY</td>
<td>BLOCK</td>
<td>Wed Feb 19 2020 15:27:35 GMT-0500 (Eastern Standard Time)</td>
</tr>
<tr>
<td>165.22.209.251</td>
<td>/wp-login.php</td>
<td>AWS\AWSM\ManagedRules\CommonRuleSet#GenericRFI_BODY</td>
<td>BLOCK</td>
<td>Wed Feb 19 2020 15:27:35 GMT-0500 (Eastern Standard Time)</td>
</tr>
<tr>
<td>185.208.164.120</td>
<td>/wp-login.php</td>
<td>AWS\AWSM\ManagedRules\CommonRuleSet#GenericRFI_BODY</td>
<td>BLOCK</td>
<td>Wed Feb 19 2020 15:27:35 GMT-0500 (Eastern Standard Time)</td>
</tr>
<tr>
<td>213.32.47.9</td>
<td>/wp-login.php</td>
<td>AWS\AWSM\ManagedRules\CommonRuleSet#GenericRFI_BODY</td>
<td>BLOCK</td>
<td>Wed Feb 19 2020 15:27:35 GMT-0500 (Eastern Standard Time)</td>
</tr>
</tbody>
</table>

### Sample request

```plaintext
POST /wp-login.php
Host: 3.18.187.84
Content-Length: 94
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
Content-Type: application/x-www-form-urlencoded
Cookie: wordpress_test_cookie=Wp+Cookie+check
Accept-Encoding: gzip
```
Step 2: setting up AWS Shield Standard - As an AWS customer, you automatically get the basic DDoS protection with the AWS Shield Standard plan which helps you protect your applications from common DDoS attacks, such as SYN floods and UDP reflection attacks. For an additional cost, you can get advanced DDoS protection by activating the AWS Shield Advanced plan. Follow the AWS documentation to activate the advanced features under AWS Shield Advanced plan.

Option 2: Radware Cloud Service for WAF and DDOS Prevention
Implementation procedure:
Step 1: Onboard the application to Radware Cloud.
Step 2: Update the DNS setting to point traffic to Radware Cloud.

Step 1: Onboard the application to Radware Cloud - The first step to integrating Radware cloud service in your environment is to add the application onto the Radware cloud. On the Radware cloud portal, go to ‘Assets > Application’ and click on the plus button on the upper right-hand side of the screen. Add the prompted details i.e. the application domain name (www.ciscosafeapp.com), the origin server (in this case it would be the ALB domain name) and protocol (HTTP). Please note that if your application is based on HTTPS protocol, you would need to add the certificate information as well.

Once the details are saved, the application would show as below.
Click on the application and copy the allocated CNAME. We need to update the DNS records for our application in Route 53 with this Radware CNAME.

Step 2: Update the DNS setting to point traffic to Radware Cloud - Log back into the AWS Route 53 console and update the DNS record sets for the application domain with Radware CNAME (from previous step) and corresponding IP address. After this change, it might take a few minutes for DNS update to propagate. Once the DNS records are fully updated, the traffic will start getting redirected to Radware Cloud servers before it hits the ‘origin server’ in AWS cloud.

Note: To eliminate direct origin attacks, Radware recommends using a firewall on your side to allow only Cloud WAF to access the application origin server directly. The service IP addresses can be requested from ERT (Radware Cloud WAF Quick Start guide).

Radware Cloud portal displays all the traffic statistics related to various onboarded applications. The dashboards are fully customizable based on your requirements.
On the Radware Cloud portal, go to ‘Monitor > Security Events’ to see all the WAF and DDOS events generated from any malicious activity targeting your application.
Validation Testing

Tetration

Test case 1: Creating an application workspace in Tetration
This test case involves defining annotations for each workload. These annotated attributes are used later to segregate the tiers and segments within the AWS VPC and hence define a workspace for our three-tier cloud application.

Validation procedure:
Step 1: Build an Inventory.
Step 2: Define scopes.
Step 3: Create a workspace.

Step 1: Build an Inventory – Define the attributes that would help you segregate your three-tier application workloads in the cloud and hence construct policies for them. We will use a combination of two different methods to add the user annotations - 1) Upload a CSV file and 2) Auto import from external AWS orchestration.

1.1: Based on the architecture of tiered application (elaborated at the beginning of this guide), following annotations were defined (Table 2: AWS Cloud Inventory). Save this in a CSV file.

<table>
<thead>
<tr>
<th>IP</th>
<th>Application</th>
<th>Region</th>
<th>Tier</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0.2.0/24</td>
<td>Safe3tierApp</td>
<td>US-East-2</td>
<td>WebServer</td>
<td>AWS-Cloud</td>
</tr>
<tr>
<td>10.0.3.0/24</td>
<td>Safe3tierApp</td>
<td>US-East-2</td>
<td>WebServer</td>
<td>AWS-Cloud</td>
</tr>
<tr>
<td>10.0.4.0/24</td>
<td>Safe3tierApp</td>
<td>US-East-2</td>
<td>AppServer</td>
<td>AWS-Cloud</td>
</tr>
<tr>
<td>10.0.5.0/24</td>
<td>Safe3tierApp</td>
<td>US-East-2</td>
<td>AppServer</td>
<td>AWS-Cloud</td>
</tr>
<tr>
<td>10.0.6.0/24</td>
<td>Safe3tierApp</td>
<td>US-East-2</td>
<td>Database</td>
<td>AWS-Cloud</td>
</tr>
<tr>
<td>10.0.7.0/24</td>
<td>Safe3tierApp</td>
<td>US-East-2</td>
<td>Database</td>
<td>AWS-Cloud</td>
</tr>
<tr>
<td>10.0.8.0/24</td>
<td>Safe3tierApp</td>
<td>US-East-2</td>
<td>Management</td>
<td>AWS-Cloud</td>
</tr>
<tr>
<td>10.0.9.0/24</td>
<td>Safe3tierApp</td>
<td>US-East-2</td>
<td>Management</td>
<td>AWS-Cloud</td>
</tr>
</tbody>
</table>

1.2: Now, log into the tetration cloud portal and go to ‘Visibility > Inventory Upload’. Click on ‘Select File’ and ‘add’ the CSV file.
1.3: Go to ‘Visibility > External Orchestrator’. Click on ‘Create New Configuration’ and fill in the required details as shown below.

Note: You will need to create an Access Key in your AWS account to be used in this configuration. Follow AWS documentation for more details on access key creation.

1.4: After few mins, you can go to ‘Visibility > Inventory Search’ and test the filters generated based on annotations from the Steps 1.2 and 1.3 above.
**Step 2: Define scopes** - We will define a scope to group together all the workloads in our three-tier application in AWS cloud. We will make use of the annotations/filters that we constructed in Step 1. We created the scope ‘AWS-US-EAST’ which includes all the workloads from our three-tier app in ‘US-East-2’ region in AWS Cloud.

Click on settings icon in the top right corner of the portal and then go to ‘Scopes’ option. Click on ‘Create New Scope’ and fill in the name of the scope and a query as below.

![Image of creating a new scope in AWS console]

**Step 3: Create a workspace** - Application workspaces are the containers for defining, analyzing and enforcing policies for a particular application. We will create a workspace for our three-tier AWS cloud application in this step.

Click on ‘Applications’ and then click on ‘Create New Workspace’. Give the workspace a name and select the scope that we created in Step 2.

![Image of creating a new workspace in AWS console]

At this point, we have successfully built the inventory, created a scope and hence defined a workspace for our three-tier cloud application.

**Test case 2: Using Application Dependency Mapping (ADM) run to set up policies and app view**

This test case validates the use of ‘ADM’ to automatically discover the policies based on flow and other data received from workloads. We will refine the discovered workload clusters and update the inventory filters to eventually come up with a set of policies which can be enforced on our cloud workloads.

**Validation procedure:**
*Step 1: Discover policies using ADM*
*Step 2: Refine inventory filters, clusters and policies*
*Step 3: Create the App View*
**Step 1: Discover policies using ADM** – Before running the ADM, ensure that all types of traffic flows are generated in the application environment, this would provide ADM the required data to generate an accurate policy set and hence ensure that we don’t miss any critical but less common traffic flows.

Go to the newly created workspace and click on ‘automatically discover policies’, select a suitable time range to ensure that you cover all the traffic flows.

![Image of ADM interface](image)

**Step 2: Refine inventory filters, clusters and policies** – Post the ADM run, policies and clusters would be generated. At this point, we manually update and customize all the cluster queries and approve them.

2.1: Go to ‘Clusters’ tab, click on any of the clusters, the panel on the right-hand side will show the cluster details like name, description, cluster query, workloads, neighbors. Update name and description to make it more intuitive, update the cluster query if need be. For example, we updated the cluster query for auto-scaled workloads. We used annotations from external AWS orchestration (test case 1 - Step 1.3) to dynamically identify the auto-scaled workloads like web servers and application servers.

![Image of cluster details](image)

2.2: Click on ‘policies’ tab, review the policies keeping in mind the traffic flows. We considered the following flows:

- Users to ALB
- Users to API Gateway
- Traffic between the workloads.
  - ALB to web servers, web servers to NLB, NLB to app servers, app servers to database
  - API gateway to application workloads and RDS
Test case 3: Enforcing policies on workloads
This test case focuses on enforcing the policy set that we formulated in Test Case 2. We will publish the policies and verify if those are enforced as expected.

Validation procedure:
Step 1: Publish the policies
Step 2: Verify policy enforcement on workloads

Step 1: Publish the policies – Select the ‘enforcement’ tab and click on ‘enforce policies’.
Step 2: Verify policy enforcement on workloads - Since we had CENTOS based workloads, we monitored the `/usr/local/tet/log/tet-enforcer.log` to see if policies are successfully enforced. A simple ping or telnet test can also be used to verify the lockdown of ports and protocols.

Use the CLI command ‘ipset list’ to view the ipset firewall settings enforced by Tetration agent on the CentOS workloads.
Test case 4: Detecting vulnerable workloads

This test case looks for vulnerable packages/software installed on various workloads in our three-tier application environment. We identify a vulnerable package/software on our workloads and patch it and then rerun the report.

Validation procedure:
Step 1: Check the vulnerability report
Step 2: Fix a vulnerability and rerun the report

Step 1: Check the vulnerability report – Go to ‘Security > Vulnerabilities’, click on ‘Packages’ tab to see all the vulnerable packages installed on various workloads in our three-tier application. For the sake of this test, let’s consider ‘libcurl-7.29.0-51.el7’ as shown below.

For demo purpose, we pick the workload ‘ip-10-0-2-4’. Logon to this workload and verify the libcurl version.
Step 2: Fix a vulnerability and rerun the report – We upgrade the libcurl version on this workload to the latest version which has the fix to all the CVEs listed in Step 1 above.

Wait for a few minutes after the update, go back to tetration portal and check the vulnerability report again. We can see that none of the CVEs related to libcurl show up anymore.

Advanced Malware Protection for Endpoints (AMP4E)

Test Case 1: Detect and quarantine the suspicious file
This test case involves detection of a suspicious file on a workload in tetration and then using the AMP for endpoint ‘simple custom detections’ to quarantine the same.

Validation procedure:
Step 1: Analyzing the ‘File Hash’ anomaly on workload
Step 2: Setting up AMP4E policy to quarantine the suspicious program
Step 3: Verifying the deletion of suspicious program

Step 1: Analyzing the ‘File Hash’ anomaly on workload – Log onto the tetration portal, go to ‘Security > Vulnerabilities > Workloads’, select a workload and then click on hostname in the pop-up window. Now, click on ‘File Hash’ tab. This tab will show us any anomalous changes in file hashes. We see that there has been some change with binary related to curl utility. Assuming this is a suspicious activity and there were no such intentional updates/changes done to this program, you would want to quarantine it. Click and copy the SHA256 hash.
Step 2: Setting up AMP4E policy to quarantine the suspicious program – As per of our initial AMP4E set up, we had configured the group ‘AWS-ThreeTier-App’ (Management > Groups) for our workloads in the three-tier application.

Note: During our design implementation phase we had used the AMP4E agent tied to this specific group ‘AWS-ThreeTier-App’, which we had created as part of initial AMP4E set up(not elaborated in this guide, please follow AMP4E documentation for detailed steps on setting up AMP4E policies). All the workloads in AWS VPC register to this specific group.

It can be seen in the snapshot above that we tied the specific group to Linux policy ‘ThreeTier-App-LinuxPolicy’. Go to ‘Management > Policies’ and select the specific Linux policy.
Note: We had preconfigured the Linux policy associated with AMP4E group ‘AWS-ThreeTier-App’. We also tied a new Simple Custom Detection ‘AWS-ThreeTierApp-CSD’ to the Linux policy. If there was no initial config on AMP console, then you would see default policies here.

As we see in the snapshot, the Linux policy above is tied to Simple Custom Detections ‘AWS–ThreeTier-CSD’ (Outbreak Control > Simple). We add the SHA256 hash that we copied in Step 1 to this simple custom detection configuration. Adding the SHA value quarantines the binary associated with it from all the workloads.

**Step 3: Verifying the deletion of suspicious program**  
Go to ‘Analysis > Event’, we see a ‘Quarantine successful’ event post our steps above. Checking the specific workload for ‘curl’ program further confirms the successful action.

Stealthwatch Cloud and Cisco Defense Orchestrator (CDO)

**Test Case 1: Detect suspicious activity & manage AWS security groups**

This test case involves using stealthwatch cloud to monitor the activity within the AWS VPC. In this specific case, we saw that stealthwatch detected access attempts to our webservers from the malicious users on the internet. We used CDO to tighten the security group rules to prevent any compromise.

**Validation procedure:**

**Step 1: Monitor suspicious activity in Stealthwatch Cloud**

**Step 2: Update AWS security groups**
Step 1: Monitor suspicious activity in Stealthwatch Cloud - Login to the stealthwatch cloud portal. Go to ‘alerts’, we found the alert ‘Excessive Access Attempts’ as shown below. This alert indicated that there have been numerous attempts to get SSH access from an unexpected geo location, which was a suspicious behavior.

Step 2: Update AWS security groups - Log into CDO portal, select ‘Policies > AWS VPC Policies’, Since the workload that was attacked is part of ‘WebSG’ group, we update the rule to only allow SSH access to specific trusted source addresses and not ‘any’ IP address on the internet.
After saving the change. Go to ‘Devices and Services > [AWS-VPC]’, review and deploy the change to push it out to the AWS cloud.
Cisco Umbrella

Test Case 1: DNS security
This test case involves adding DNS layer security to the AWS workloads. We created a DNS policy for our three-tier application workloads to block malicious domains. In order to verify the blocks, we accessed a test domain ‘examplemalwaredomain.com’ and then confirmed the same from Umbrella reporting.

Validation procedure:
Step 1: Set up DNS policy for AWS workloads
Step 2: Confirm if malware domain is blocked

Step 1: Set up DNS policy for AWS workloads – Go to ‘Policies > Management > DNS Policies’, add a new policy and make sure ‘Malware’ is set to block under security settings. Save the change.
Step 2: Confirm if malware domain is blocked – Run `nslookup` on a test malware domain as shown in snapshot below. Utility returns Umbrella block page address as below.

To further confirm the block action, select ‘Reporting > Activity Search’ and filter the accessed malware domain. Events show the action as ‘Blocked’.
Duo Beyond

Test Case 1: Set up the cloud application for Two-Factor Authentication (2FA)
This test case involves logging into the application for the first time. Duo will prompt the user to enroll their phone for 2FA. After successful enrollment, user gets the ability to approve subsequent login attempts.

Implementation procedure:
Step 1: Set up Duo 2FA for a new user.
Step 2: Log onto the cloud application.

Step 1: Set up Duo 2FA for a new user - After the initial user authentication, the Duo 2FA kicks in and since this is the first authentication attempt, the user was prompted to enroll for 2FA.

Step 2: Log onto the cloud application - After the enrollment, we continue to log onto the application, this time the user is presented with Duo authentication methods instead of ‘setup’. Once the user approves the authentication request, they are allowed to login.
Test Case 2: Monitor 2FA activity from Duo admin portal
This test case involves monitoring the 2FA enrollment and login activity in the Duo admin portal.

Implementation procedure:
Step 1: Verify the 2FA enrolled devices.
Step 2: Track the user logins in authentication logs.

Step 1: Verify the 2FA enrolled devices – Logon to the Duo admin portal and select ‘2FA Devices’, the portal shows the list of enrolled devices along with other details like platform, hardware model and usernames.

Step 2: Track the user logins in authentication logs – Go to ‘Dashboard > Authentication log’, to track user 2FA login activity as shown in the snapshot below.
Test Case 1: Integrate AMP4E and Umbrella with CTR

In this test case, we enable the AMP4E and Umbrella modules in CTR portal. We create API keys in AMP4E and Umbrella portals and then configure the same in CTR portal.

Implementation procedure:
Step 1: Add the AMP4E module.
Step 2: Add the Umbrella module.

Step 1: Add the AMP4E module - Log on to the CTR portal and go to available modules, add a new AMP4E module. CTR portal displays all the steps on how to integrate the AMP module once you click on ‘add module’ as shown in snapshot below.
Step 2: Add the Umbrella module - Go to available modules again, this time add a new Umbrella module. In a similar manner as AMP4E, the integration steps are listed in the CTR portal itself. Create the API keys in Umbrella portal as instructed and then add the same in CTR portal.

After we have saved the module configurations, the modules will be listed under 'your configurations' as below.

Test Case 2: Track malicious Activity on CTR
In this test case, we track the life cycle of vulnerable package that we detected using tetration in previous steps. We will use the same SHA value and see what CTR offers in terms of visibility in our environment.

Implementation procedure:
Step 1: Investigate a malicious SHA value.
Step 2: Track the file trajectory.
Step 1: Investigate a malicious SHA value - Log on to the CTR portal and select investigate. Add the SHA value in provided space and click on investigate. CTR pulls all the information about the associated package and what workloads the specific package had interacted with. Under the observables section, we see that AMP4E detected this SHA value as malicious based on our custom policy, CTR displays the specific AMP4E policy name as well.

Step 2: Track the file trajectory - Click on the ‘SHA-256 Hash’ shown in the Relations Graph. Expand the drop-down menu and click on ‘File trajectory’.

Clicking on ‘File trajectory’ should redirect you to AMP4E portal page which displays the trajectory of the vulnerable package on the specific workload. Clicking on a particular timestamp displays the related events. The event history shows the time of execution of the file along with other details.
## Visibility
- Earliest observed in past 30 days: 2020-01-08 21:15:34 UTC
- Last Seen: 2020-01-08 21:16:57 UTC

## Entry Point
- Earliest seen on from past 30 days: AWS-ThreeTier-App / ip-19-0-2-4.us-east-2.compute.internal

## Created by
<table>
<thead>
<tr>
<th>SMA-356</th>
<th>Filename</th>
<th>Product</th>
<th>Prevalence</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMA-356</td>
<td>826b3e7b...3d72e334</td>
<td>bash</td>
<td>129</td>
</tr>
<tr>
<td></td>
<td>5975-b972...2bd4974a</td>
<td>bash</td>
<td>34</td>
</tr>
</tbody>
</table>

## File Details

## Network Profile

## Trajectory

<table>
<thead>
<tr>
<th>AWS-ThreeTier-App</th>
<th>ip-19-0-2-4.us-east-2...</th>
<th>Jan, 8</th>
<th>Jan, 9</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>21:15</td>
<td>0:19</td>
</tr>
<tr>
<td></td>
<td></td>
<td>22:16</td>
<td>23:00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1:08</td>
<td>2:25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3:05</td>
<td>3:35</td>
</tr>
</tbody>
</table>

Executed by Unknown 826b3e7b...3d72e334.
Unknown disposition at the time.
Path: /usr/bin/curl
At 2020-01-09 03:05:19 UTC

## Event History

<table>
<thead>
<tr>
<th>Date</th>
<th>Computer</th>
<th>Group</th>
<th>Event</th>
<th>SMA-356</th>
<th>File...</th>
<th>Proc...</th>
<th>Disposition</th>
</tr>
</thead>
<tbody>
<tr>
<td>2020-01-08 21:15:34 UTC</td>
<td>ip-19-0-2-4.us-east-2...</td>
<td>AWS-ThreeTier-App</td>
<td>Executed by 5975-b972...2bd4974a bash</td>
<td>Unknown</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2020-01-08 21:13:34 UTC</td>
<td>ip-19-0-2-4.us-east-2...</td>
<td>AWS-ThreeTier-App</td>
<td>Created by</td>
<td>Unknown</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2020-01-08 21:15:35 UTC</td>
<td>ip-19-0-2-4.us-east-2...</td>
<td>AWS-ThreeTier-App</td>
<td>Moved by</td>
<td>Unknown</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2020-01-08 21:16:57 UTC</td>
<td>ip-19-0-2-4.us-east-2...</td>
<td>AWS-ThreeTier-App</td>
<td>Executed by 826b3e7b...3d72e334</td>
<td>Unknown</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix A - Software Versions Tested

<table>
<thead>
<tr>
<th>Product</th>
<th>Platform</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tetration</td>
<td>Software agent</td>
<td>3.3.2.16-enforcer</td>
</tr>
<tr>
<td>AMP4E</td>
<td>Software agent</td>
<td>1.11.1.663</td>
</tr>
<tr>
<td>CDO</td>
<td>Cloud</td>
<td>SaaS</td>
</tr>
<tr>
<td>Stealthwatch Cloud</td>
<td>Cloud</td>
<td>SaaS</td>
</tr>
<tr>
<td>Duo Beyond</td>
<td>Cloud</td>
<td>SaaS</td>
</tr>
<tr>
<td>Umbrella VAs</td>
<td>Appliance (EC2 Instance)</td>
<td>2.6.2</td>
</tr>
<tr>
<td>CTR</td>
<td>Cloud</td>
<td>SaaS</td>
</tr>
<tr>
<td>CentOS(Workloads)</td>
<td>EC2 Instance</td>
<td>CentOS Linux 7</td>
</tr>
<tr>
<td>RDS</td>
<td>MySQL</td>
<td>mysql-5-7</td>
</tr>
</tbody>
</table>

Appendix B - References

<table>
<thead>
<tr>
<th>Product</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco Duo Beyond</td>
<td><a href="https://duo.com/">https://duo.com/</a></td>
</tr>
<tr>
<td>Cisco Umbrella</td>
<td><a href="https://umbrella.cisco.com/">https://umbrella.cisco.com/</a></td>
</tr>
<tr>
<td>AWS Services</td>
<td><a href="https://aws.amazon.com/">https://aws.amazon.com/</a></td>
</tr>
<tr>
<td>AWS well-architected framework</td>
<td><a href="https://aws.amazon.com/architecture/well-architected/">https://aws.amazon.com/architecture/well-architected/</a></td>
</tr>
<tr>
<td>Radware for AWS (WAF and DDoS)</td>
<td><a href="https://www.radware.com/products/cloud-security-aws-resources/">https://www.radware.com/products/cloud-security-aws-resources/</a></td>
</tr>
<tr>
<td>NGINX wordpress</td>
<td><a href="https://www.nginx.com/resources/wiki/start/topics/recipes/wordpress/">https://www.nginx.com/resources/wiki/start/topics/recipes/wordpress/</a></td>
</tr>
</tbody>
</table>