Securing the Cloud: A Functional Analysis of AWS Services and Traditional Security Solutions

By: KJ McDaniels

Team 1: Emmanuel Apietu, Zedd Chisolm, KJ McDaniels, Opeyemi Olaleye, Yonisibel Soto

The Knowledge House

Cybersecurity Cohort: George Robbins & Emilie Dionisio

3/27/2024

Simple annoyances from script kiddies are a thing of the past. The security landscape is continuously shifting dramatically, and today's adversaries are highly trained, well-funded, and relentless in their pursuit of data. Because of this, rigorous testing is crucial. Without it, attackers have wide open pathways to infiltrate networks, steal data, and leverage their stolen knowledge for further attacks and financial gain. Security testing is paramount to proving that a system, network, application, etc., can push back attempts to break any properties of the CIA triad. Amazon Web Services (AWS) offers a robust catalog of security tools for building multi-subnet training labs. Following benefits and frameworks of Well-Architected Frameworks, scalability, low-overhead, flexibility, and variable costs, enables cost-effective and equipped simulated training environments without the upfront investment typically required by traditional security training environments. By training in a cloud-based multi-subnet environment, future cybersecurity professionals gain practical experience with the very tools and configurations they'll encounter in the field. This paper explores the security tools offered by AWS and how their functionalities compare to traditional open-source cybersecurity training labs.

Security Groups

Security groups are basic, core virtual firewall controls within an AWS VPC, they offer functionality similar to traditional firewall network security tools. Considering an analogy, security groups offer a function like a lock on a door. The user defines who has the key to the lock, as an IP address/security group. Security groups control inbound and outbound traffic for EC2 instances. You can specify allowed IP ranges, ports (specific and ranges (e.g., port 22 for SSH access, or port 80-8080 for web traffic, and protocols (TCP, UDP, ICMP) to control the traffic to and from instances. They can apply to traffic from/to CIDR blocks, security groups or self-references. This allows filtering traffic like a firewall. By default, security groups deny all inbound and outbound traffic until you add rules. This implements a default-deny approach for security like firewalls.

Security groups are stateful - responses to allowed inbound traffic are allowed to flow outbound without additional rules. This models firewall statefulness. Security group rules can be specified to restrict traffic between instances within the same security group or across security groups. This allows segmentation of network access similar to separate security zones in traditional networks. Security groups are attached to an instance within a VPC and travel with the instance if moved. This provides network access control even if an instance is stopped/started or moved between Availability Zones or VPCs. When an instance is launched you assign one or more security groups. The rules from all assigned groups are aggregated to control the overall inbound and outbound access. This allows grouping instances with similar access needs into security groups and managing rules centrally. Rules applied to associated security groups are automatically updated for all instances.

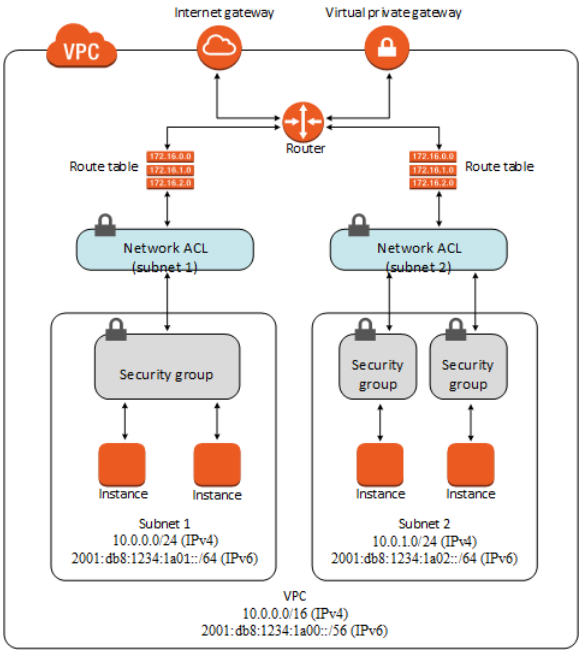


Fig 1: Network topology of a VPC. A VPC provides logically isolated network environment where users can launch AWS resources and define their own private network configuration. The VPC architecture leverages security groups, network access control lists (NACLs) to implement a defense in depth/layered security approach. Security groups control traffic at the individual instance level, while NACLs manage traffic at the subnet level. This collaborative approach safeguards communication within the VPC

Source: Club Cloud Computing

NACLs

Taking on a defense in depth posture, Network Access Control Lists (NACLs) can be used additionally to further restrict traffic flow—at the subnet level---providing a functionality similar to traditional network firewalls and security tools. NACLs act as a virtual firewall at the subnet level, controlling inbound and outbound traffic just like a firewall. You can add rules to allow, deny or limit traffic to subnets based on IP protocols, ports, and IP ranges. Similar to security groups, NACLs have a default deny policy where all inbound and outbound traffic is blocked until allowed by rules. This implements a default deny approach for security. NACL rules are evaluated after security groups so they provide an additional layer of security and can restrict traffic even between instances within the same security group if needed. NACL rules are stateless - responses to allowed inbound traffic are subject to rules evaluation just like new traffic. This is different from security groups which are stateful. Multiple NACLs can be created within a VPC and associated to subnets for network segmentation and access control between subnets like separate security zones. Unlike Security Groups, NACL rules are numbered and processed in that order which provides control over rule precedence like many traditional firewalls.

Table 5.9 Network ACL rules allowing HTTPS traffic into a subnet

Rule #	Type	Protocol	Port range	Source	Allow or deny
100	HTTPS	TCP	443	0.0.0.0/0	Allow
200	ALL	TCP	ALL	0.0.0.0/0	Deny

Fig 2: Network ACL rules allowing HTTPS traffic into a subnet

Source: AWS Security, Dylan Shields

Securing the Cloud

GuardDuty

Amazon GuardDuty is an AWS service that once enabled automatically monitors AWS accounts and detects potential known and unknown malicious threats to amplify security across one or multiple accounts. It analyzes events from AWS CloudTrail, Amazon Virtual Private Cloud (VPC) Flow Logs, DNS logs, Amazon Simple Storage Service (S3) events, and Amazon Elastic Kubernetes Service (EKS) audit logs. The detections, rule sets, and threat intelligence are created, maintained, and updated by AWS Security, so customers don't have to write rules or detection logic. When a potential threat is detected, GuardDuty delivers a detailed security finding to the GuardDuty console and Amazon CloudWatch Events. This makes alerts actionable and easy to integrate into existing event management or workflow systems. In addition, remediation actions can be automated by integrating GuardDuty with other services like AWS Security Hub, Amazon EventBridge, Lambda, and AWS Step Functions. Relating to traditional cybersecurity labs, GuardDuty identifies potential threats such as malware, reconnaissance by attackers and compromised accounts—providing SIEM-like threat monitoring and alerting. GuardDuty also carries IDS/IPS-like capabilities in vulnerability scanning in which the malware protection feature scans EC2 instances and container images for known vulnerabilities and malware using signatures. Another relation includes network monitoring. By analyzing VPC flow logs, GuardDuty monitors network traffic for unusual connections or unauthorized entry or exit which provides firewall-like protection.

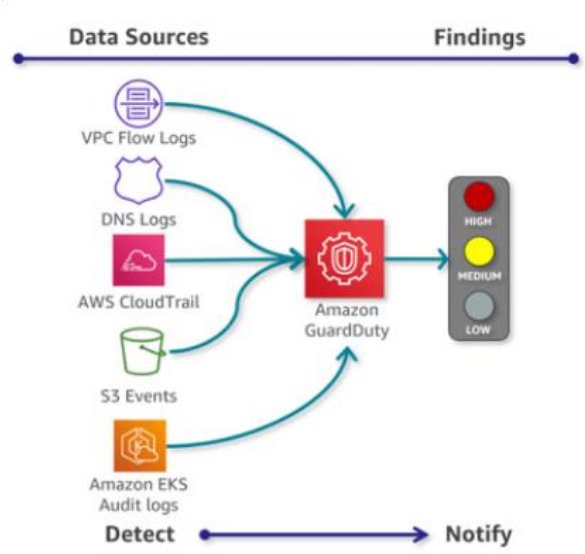


Fig 3: GuardDuty ingests data from various sources, including VPC Flow Logs, DNS Logs, AWS CloudTrail logs, Amazon EKS Audit logs, and potentially others. It then processes this data to generate findings categorized as high, medium, or low

Fig 3 Source: AWS

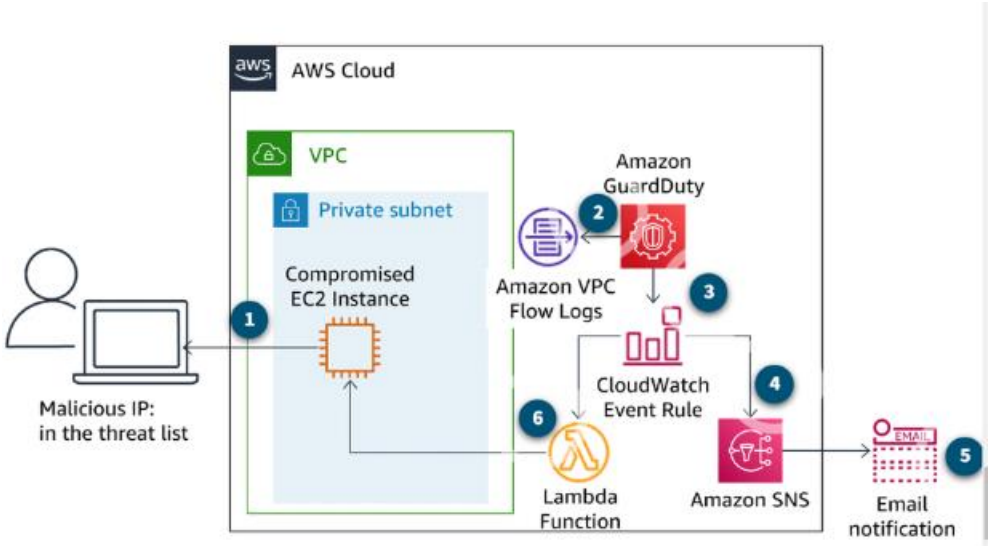
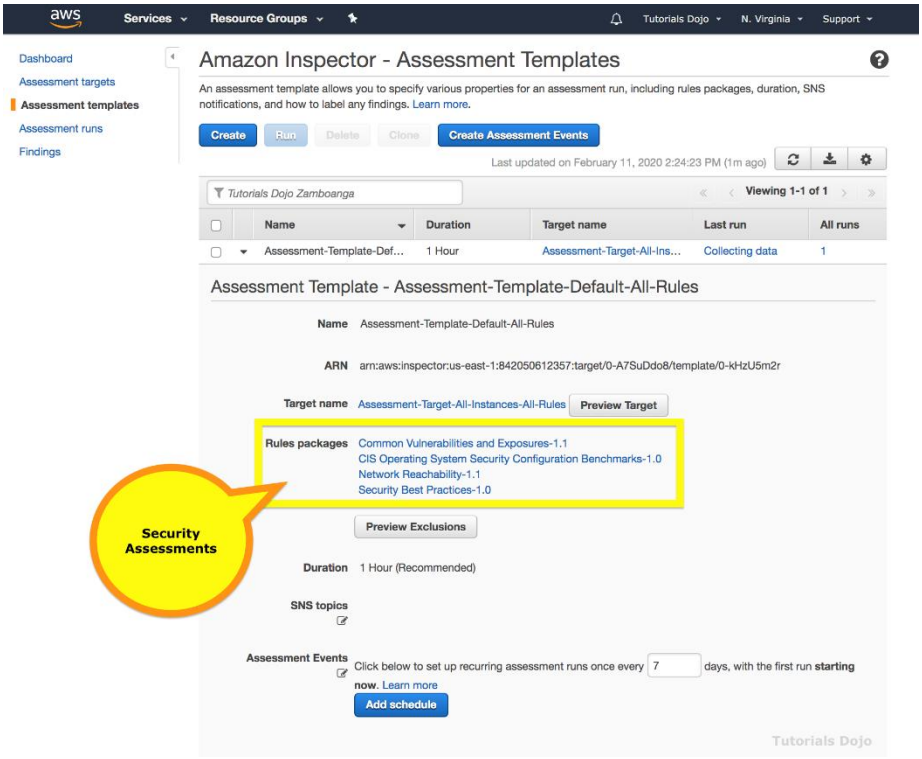


Fig 4 Source: AWS

Fig 4: GuardDuty Threat Detection: A compromised EC2 instance communicates with a malicious IP on a custom GuardDuty threat list. VPC Flow Logs capture this traffic, triggering a GuardDuty finding. The finding is sent to the GuardDuty console and CloudWatch Events. A CloudWatch rule invokes an SNS notification and a Lambda function. The Lambda function isolates the compromised instance using security group information, stopping communication with the malicious IP.

Inspector

Amazon Inspector achieves many of the functionalities of traditional security tools through its automated vulnerability assessment capabilities. Amazon Inspector regularly scans EC2 instances, Lambda functions, containers, and other AWS resources to identify vulnerabilities, malware, and configuration issues. It tests the network accessibility of the EC2 instances and the security state of applications running on the instances. This provides continuous monitoring similar to SIEM and vulnerability scanning tools. It leverages an extensive knowledge base of security checks that are constantly updated by AWS researchers. This allows it to detect issues comparable to what traditional tools can find. Detailed findings are generated which include severity, remediation guidance and integration with services like Security Hub provides centralized visibility and alerting. Being native to AWS, Inspector can take advantage of other capabilities around encryption, identity access control and resilience that AWS provides. It also integrates with services like Security Hub, GuardDuty and CloudWatch to further automate security operations. Some of the key views of findings provided are active, suppressed, and closed findings. Findings can also be filtered and exported to the AWS CLI, API, or security tools like Security Hub. Suppression rules can also be applied to certain findings.



Source: Tutorials Dojo

Fig 5: Assessment Template of Amazon Inspector. These templates define configurations for running security assessments on EC2 instances. This security assessment includes four rule packages that are used to evaluate target resources for potential security issues. The template also specifies the duration of a period of time for the assessment to run.

CloudWatch Logs

CloudWatch Logs follows the functionality of traditional cybersecurity environments through log monitoring and analytics. CloudWatch Logs collects logs from various AWS resources like EC2 instances, Lambda functions, and other services. It provides centralized monitoring of logs similar to SIEM tools. The logs ingested by CloudWatch Logs can be analyzed to detect anomalies, threats, or policy violations. Metric filters and log insights feature helps analyze logs and generate metrics similar to what traditional monitoring and SIEM tools can provide. Detailed log events are stored securely in CloudWatch Logs. Advanced log searching and analytics capabilities help detect incidents, troubleshoot issues and investigate security events. CloudWatch Logs can leverage other capabilities around identity access control, encryption in transit and at rest that AWS provides. It integrates with services like CloudWatch Metrics, Lambda, Kinesis Data Firehose for further automation. Security Hub integration provides visualization of log-based insights along with findings from other AWS security services. Some key concepts of CloudWatch Logs include log groups, log streams, metric filters, retention settings etc. The Standard and Infrequent Access log classes provide options to suit different data access needs.

CloudWatch Metrics/Alarms

In addition, CloudWatch supports different types of alarms like metric alarms that watch a single metric and composite alarms that combine multiple metrics. When an alarm triggers, it can perform actions like stopping the instance, sending notifications, or adjusting capacity through Auto Scaling. The alarms use the historical data stored in CloudWatch to automatically adjust thresholds using anomaly detection. This reduces false alarms from normal fluctuations. Alarms can also be correlated with logs and traces in CloudWatch for efficient troubleshooting.

WAF

WAF, Web Application Firewall is an AWS native firewall that protects web application servers at the application layer against a range of security threats. WAF functions through creating security rules that control bot traffic and block common attack patterns like SQL injections or cross-site scripting (XSS). It also monitors the HTTP and HTTPS requests that are forwarded to an Amazon API Gateway API, Amazon CloudFront, or an Application Load Balancer. Three things allow Amazon WAF work—Access Control Lists, Rules, and Rule Groups. Amazon WAF allows you to control your content by using an IP address from where the request originates. Three things make Amazon WAF work – Access control lists (ACL), Rules, and Rule Groups. Amazon WAF manages Web ACL capacity units (WCU) for rules, rule groups, and web ACLs. Amazon WAF includes a full-featured API that you can use to automate the creation, deployment, and maintenance of security rules. WAF allows you to monitor web requests that are allowed or blocked based on rules you define. It provides visibility into web traffic and potential threats similar to what an IDS/IPS can detect. The rules in WAF act like a firewall by allowing or blocking requests based on conditions. WAF logs details of every web request that is inspected. These logs are sent to CloudWatch Logs which allows you to analyze them for anomalies, threats or policy violations. You can create log metrics and dashboards in CloudWatch similar to a SIEM. Being native to AWS, WAF takes advantage of other AWS capabilities around identity and access management, encryption etc. It integrates with services like Security Hub for unified security management and visualization. You can automate the monitoring of WAF using CloudWatch alarms. AWS also provides tools like Firewall Manager to centrally manage distributed WAF configurations for simplified monitoring and management.

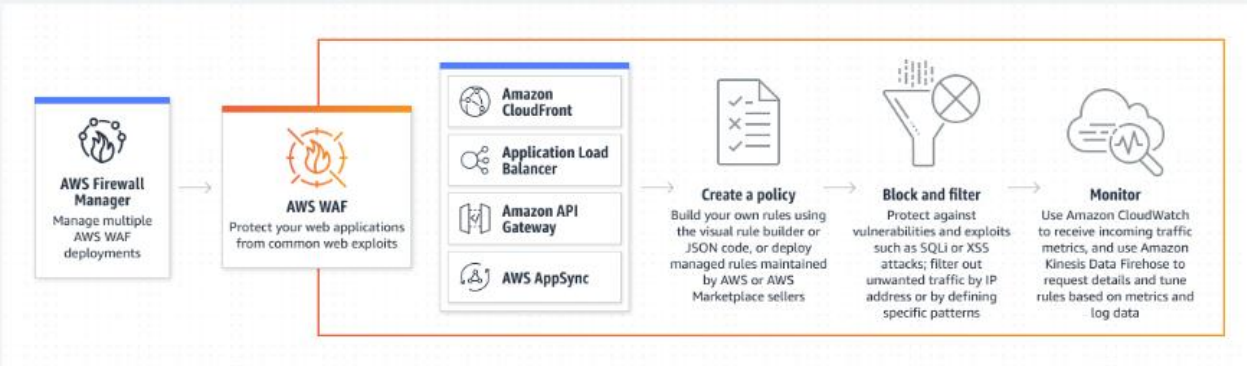


Fig 6 Source AWS

Fig 6: AWS Firewall Manager Manages multiple AWS Web Application Firewall Deployments. AWS WAF protects deployed applications from common web exploits. Policies can be created by building rules using the visual rule builder. Block filters protect against exploits

and vulnerabilities attacks. Use Amazon CloudWatch for monitoring incoming traffic metrics & Amazon kinesis firehose for request details, then tune rules based on metrics and log data.

REFERENCES

[1] Security Products and Features - Introduction to AWS Security

https://docs.aws.amazon.com/whitepapers/latest/introduction-aws-security/security-products-and-features.html

[2] Security groups - Edit in the Cloud on AWS

https://docs.aws.amazon.com/solutions/latest/edit-in-the-cloud-on-aws/security-groups.html

[3] Infrastructure security in Amazon GuardDuty - Amazon GuardDuty

https://docs.aws.amazon.com/guardduty/latest/ug/infrastructure-security.html

[4] Using service-linked roles for Amazon GuardDuty - Amazon GuardDuty

https://docs.aws.amazon.com/guardduty/latest/ug/using-service-linked-roles.html

[5] Infrastructure security in Amazon GuardDuty - Amazon GuardDuty

https://docs.aws.amazon.com/guardduty/latest/ug/infrastructure-security.html

[6] Using service-linked roles for Amazon GuardDuty - Amazon GuardDuty

https://docs.aws.amazon.com/guardduty/latest/ug/using-service-linked-roles.html

[7] Amazon Inspector

https://tutorialsdojo.com/amazon-inspector/

[8] WAF

https://docs.aws.amazon.com/waf/latest/developerguide/web-acl.html

[9] AWS WAF (Web Application Firewall): Overview by Hardik Tyagi

https://k21academy.com/amazon-web-services/aws-certified-security-specialty-amazon-web-services/aws-waf/