27 Best Practice Tips on Amazon Web Services Security Groups

27 Best Practice Tips on Amazon Web Services Security Groups

Amazon AWS Security Groups are one of the most used and abused configurations inside an AWS environment if its being used in cloud for quite long. Since AWS security groups are simple to configure, users many times ignore the importance of it and do not follow best practices relating to it. In reality, operating on AWS security groups every day was much more intensive and complex than configuring them once. In the world of security, proactive and reactive speed determines the winner. So a lot of these best practices should be automated in reality. .AWS released so many features in the last few years relating to Security, that we should not visualize Security groups in isolation, It just does not make sense anymore. The Security Group should always be seen in the overall security context, with this I start the pointers.

Introduction to Amazon VPC

Amazon Virtual Private Cloud (Amazon VPC) enables we to launch Amazon Web Services (AWS) resources into a virtual network that we've defined. This virtual network closely resembles a traditional network that we'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS.

.

Practice 1: Enable AWS VPC Flow Logs for your VPC or Subnet or ENI level. AWS VPC flow logs can be configured to capture both accept and reject entries flowing through the ENI and Security groups of the EC2, ELB + some more services. This VPC Flow log entries can be scanned to detect attack patterns,alert abnormal activities and information flow inside the VPC and provide valuable insights to the SOC/MS team operations.

Practice 2: Use AWS Identity and Access Management (IAM) to control who in your organization has permission to create and manage security groups and network ACLs (NACL). Isolate the responsibilities and roles for better defense. For example, we can give only your network administrators or security admin the permission to manage the security groups and restrict other roles.

Practice 3: Enable AWS Cloud Trail logs for your account. The AWS Cloud Trail will log all the security group events and it was needed for management and operations of security groups. Event streams can be created from AWS Cloud Trail logs and it can be processed using AWS Lambda. For example : whenever a Security Group was deleted , this event will be captured with details on the AWS Cloud Trail logs. Events can be triggered in AWS Lamdba which can process this SG change and alert the MS/SOC on the dashboard or email as per your workflow. This was a very powerful way of reacting to events within span of <7 minutes. Alternatively, we can process the AWS Cloud Trail logs stored in your S3 every X frequency as a batch and achieve the above. But the Operation teams reaction time can vary depending on generation and polling frequency of the AWS Cloud Trail logs. This activity was a must for your operations team.

Practice 4: Enable AWS App Config for your AWS account. App records all events related to your security group changes and can even send emails.

Practice 5: Have proper naming conventions for the Amazon Web Services security group. The naming convention should follow a enterprise standards. For example it can follow the notation: "AWS Region+ Environment Code+ OS Type+Tier+Application Code Security Group Name - EU-P-LWA001 AWS Region (2 char) = EU, VA, CA etc Environment Code (1 Char) = P-Production , Q-QA, T-testing, D-Development etc OS Type (1 Char)= L -Linux, W-Windows etc Tier (1 Char)= W-Web, A-App, C-Cache, D-DB etc Application Code (4 Chars) = A001 We have been using Amazon Web Services from 2008 and found over the years managing the security groups in multiple environments was itself a huge task. Proper naming conventions from beginning was a simple practice, but will make your AWS journey manageable.

Practice 6: For security in depth, make sure your Amazon Web Services security groups naming convention was not self explanatory also make sure your naming standards stays internal. Example : AWS security group named UbuntuWebCRMProd was self explanatory for hackers that it was a Production CRM web tier running on ubuntu OS. Have an automated program detecting AWS security groups with Regex Pattern scanning of AWS SG assets periodically for information revealing names and alert the SOC/Managed service teams.

Practice 7: Periodically detect, alert or delete AWS Security groups not following the organization naming standards strictly. Also have an automated program doing this as part of your SOC/Managed service

operations. Once we have this stricter control implemented then things will fall in line automatically.

Practice 8: Have automation in place to detect all EC2,ELB and other AWS assets associated with Security groups. This automation will help us to periodically detect Amazon Web Services Security groups lying idle with no associations, alert the MS team and cleanse them. Unwanted security groups accumulated over time will create unwanted confusion.

Practice 9: In your AWS account, when we create a VPC, AWS automatically creates a default security group for the VPC. If we don't specify a different security group when we launch an instance, the instance was automatically associated with the appropriate default security group. It will allow inbound traffic only from other instances associated with the "default" security group and allow all outbound traffic from the instance. The default security group specifies itself as a source security group in its inbound rules. This was what allows instances associated with the default security group to communicate with other instances associated with the default security group. This was not a good security practice. If we don't want all your instances to use the default security group, we can create your own security groups and specify them when we launch your instances. This was applicable to EC2 , RDS , ElastiCache and some more services in AWS. So detect "default" security groups periodically and alert to the SOC/MS.

Practice 10: Alerts by email and cloud management dash board should be triggered whenever critical security groups or rules are added/modified/deleted in production. This was important for reactive action of your managed services/security operations team and audit purpose.

Practice 11 : When we associate multiple security groups with an Amazon EC2 instance, the rules from each security group are effectively aggregated to create one set of rules. AWS uses this set of rules to determine whether to allow access or not. If there was more than one SG rule for a specific port, AWS applies the most permissive rule. For example, if we have a rule that allows access to TCP port 22 (SSH) from IP address 203.0.113.10 and another rule that allows access to TCP port 22 for everyone, then everyone will have access to TCP port 22 because permissive takes precedence. Practice X.1 : Have automated programs detecting EC2 associated with multiple SG/rules and alert the SOC/MS periodically. Condense the same manually to 1-3 rules max as part of your operations.

Practice X.1 : Have automated programs detecting conflicting SG/rules like restrictive+permissive rules together and alert the SOC/MS periodically.

Practice 12 : Do not create least restrictive security groups like 0.0.0.0/0 which was open to every one.
Since web servers can receive HTTP and HTTPS traffic open, only their SG can be permissive like
0.0.0.0/0,TCP, 80, Allow inbound HTTP access from anywhere
0.0.0.0/0,TCP, 443, Allow inbound HTTPS access from anywhere
All least restrictive SG created in your account should be alerted to SOC/MS teams immediately.

Practice 13: Have a security policy not to launch servers with default ports like 3306, 1630, 1433, 11211, 6379 etc. If the policy has to be accepted, then security groups also have to be created on the new hidden listening ports instead of the default ports. This provides a small layer of defense, since one cannot infer the information from the security group port on the EC2 service it was protecting. Automated detection and alerts should be created for SOC/MS, if security groups are created with default ports.

Practice 14: Applications which require stricter compliance requirements like HIPAA, PCI etc to be met need end to end transport encryption to be implemented on server back end in AWS. The communication from ELB to

Web->App->DB->Others tiers need to be encrypted using SSL or HTTPS. This means only secured ports like 443, 465, 22 are permitted in corresponding EC2 security groups. Automated detection and alerts should be created for SOC/MS if security groups are created on secure ports for regulated applications.

Practice 15: Detection , alert and actions can be taken by parsing the AWS Cloud Trail logs based on usual patterns observed in your production environment
Example:
15.1 **:**If a port was opened and closed in <30 or X mins in production can be a candidate for suspicious activity if it was not normal pattern for your production
15.2 **:**If a permissive Security Group was created and closed in <30 or X mins can be a candidate for suspicious activity if it was not the normal pattern for your production
Detect anomalies on how long a change effected and reverted in security groups in production.

Practice 16: In case ports have to be opened in Amazon Web Services security groups or a permissive AWS security group needs to be applied, Automate this entire process as part of your operations such that a security group was open for X agreed minutes and will be automatically closed aligning with your change management. Reducing manual intervention avoids operational errors and adds security.

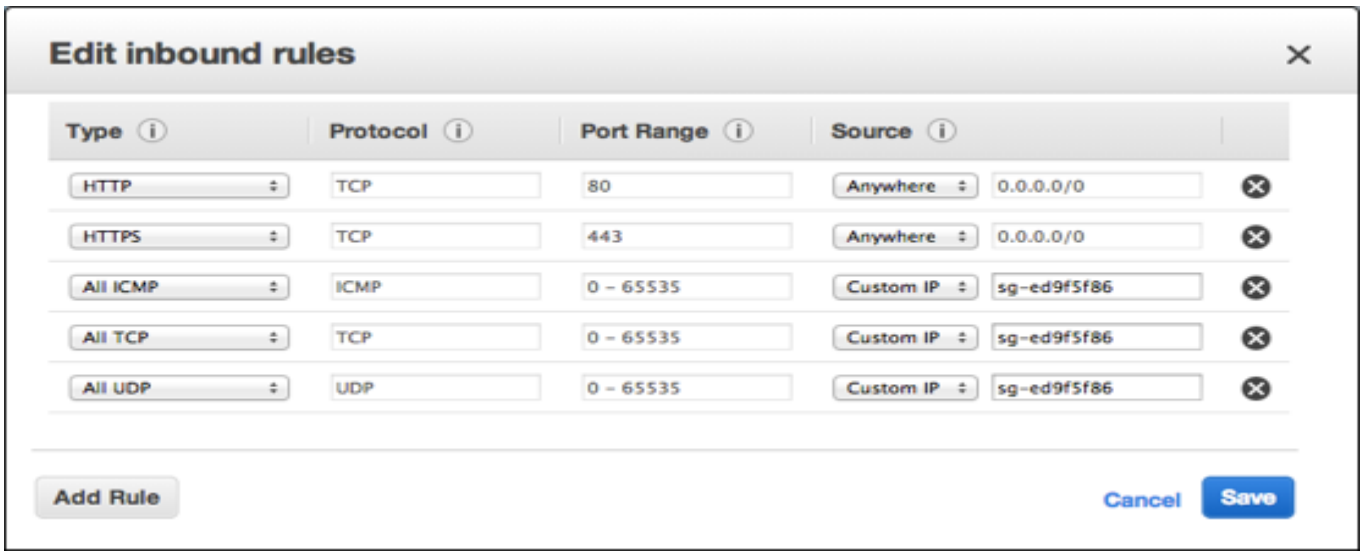
Practice 17: Make sure SSH/RDP connection was open in AWS Security Group only for jump box/bastion hosts for your VPC/subnets. Have stricter controls/policies avoid opening SSH/RDP to other instances of production environment. Periodically check , alert and close for this loop hole as part of your operations.

Practice 18: It was a bad practice to have SSH open to the entire Internet for emergency or remote support. By allowing the entire Internet access to your SSH port there was nothing stopping an attacker from exploiting your EC2 instance. The best practice was to allow very specific IP address in your security groups, this restriction improves the protection. This could be your office or on premise or DC through which we connect your jump box.

Practice 19: Too much or Too less: How many security groups for a usual multi tiered web app was preferred was a frequently asked question ?
Option 1 : One security group cutting across multiple tiers was easy to configure, but it was not a recommended for secure production applications.
Option 2: One Security group for every instance was too much protection and tough to manage operationally on longer term
Option 3: Individual Security group for different tiers of the application, For example : Have separate security groups for ELB, Web , App, DB and Cache tiers of your application stack.
Periodically check whether Option 1 type rule was being created in your production and alert the SOC/MS.

Practice 20: Avoid allowing UDP or ICMP for private instances in Security groups. Not a good practice unless specifically needed.

Practice 21: Open only specific ports, Opening range of ports in a security group was not a good practice. In the security group we can add many inbound ingress rules, While opening the ports it was always advised to open for specific ports like 80,443, etc rather than range of ports like 200-300.



Practice 22: Private Subnet instances can be accessed only from the VPC CIDR IP range. Opening instances to the public IP ranges was a possibility , but it does not make any sense. E.g., Opening HTTP to 0.0.0.0/0 in the SG of the private subnet instance does not make any sense. So detect and cleanse such rules.

Practice 23: AWS CloudTrail log captures the event events or automated programs should trigger alert activities are detected.

***23.1*:**Alert when X number of SG were added/deleted at "Y" Hours or Day by IAM user / Account

***23.2*:**Alert when X number of SG Rules were added, IAM user / Account

Practice 24: In case we are an enterprise make sure all security groups related activities of your production are part of your change management process. Security Group actions can be manual or automated with your change management in an enterprise. In case we are an agile Startup or SMB and do not have complicated Change management process, then automate most of the security group related tasks and events as illustrated above on various best practices. This will bring immense efficiency into your operations

Practice 25: Use outbound/egress security groups wherever applicable within your VPC. Restrict FTP connection to any server on the Internet from your VPC. This way we can avoid data dumps and important files getting transferred out from your VPC. Defend harder and make it tougher !

Practice 26: For some tiers of your application, use ELB in front your instance as a security proxy with restrictive security groups - restrictive ports and IP ranges. This doubles your defense but increases the latency.

Practice 27: Some of the tools we use in conjunction to automate and meet above best practices are ServiceNow, Amazon CFT, AWS API'S, Rundeck, Puppet, Chef, Python , .Net and Java automated programs.