VALTIX

TOP 10 NETWORK SECURITY MISTAKES IN AWS – AND HOW TO FIX THEM

TOP 10 NETWORK SECURITY MISTAKES IN AWS – AND HOW TO FIX THEM

In talking with end-user organizations, we've seen and heard lots of misconceptions and mistakes over the years – and even espoused a few ourselves. As Valtix is an innovative company, we've been in a unique position to understand where enterprises are coming from and see the lessons learned, too often the hard way. By sharing these insights, hopefully some of you can learn these lessons the easy way! Please note that most of the findings will be IaaS and PaaS primarily, i.e. EC2, VPC, RDS, DynamoDB, S3 etc.

So the point of this paper is to highlight what we've learned from customers, security experts, and the 10 years of experience of our team in using public clouds. In development of these mistakes and recommendations, we've also worked directly with an experienced cloud architect with one of our customers, to ensure real-world applicability of our recommendations.

For this paper, we've broken up the Top 10 into:

- Native Controls
- Visibility
- PaaS Security
- Process and Culture

Admittedly, the lines between these buckets are blurry, but it helped us organize the information and ensure good coverage. Let's dive in!

NATIVE NETWORK SECURITY CONTROLS ARE USEFUL BUT DON'T DEPEND ON THEM ENTIRELY

The knobs provided by public clouds are important to use, and they reduce the attack surface, but depending solely on them for your security does not prevent attacks, stop exfiltration or avoid lateral movement of attackers.

1 ASSUMING SECURITY GROUPS AND ACLS ARE ENOUGH TO PROTECT AGAINST ATTACKS AND STOP EXFILTRATION

This may sound too obvious to security experts, but there's still a lot of deployments that just rely on security groups and ACLs. Security groups are basically stateful firewalls, use them to reduce the attack surface: only open inbound 443 for web applications, 3389 for RDP, or 22 for SSH by admins. But assuming that you will be protected from advanced attacks is a serious fallacy and a lot of cloud apps are still using security groups alone! Similarly, access control lists (ACLs) can be used to lockdown the source traffic to your enterprise networks or hom e IP's of your admins (yeah, we're all connecting from home). But with no advanced inspection against malware or ransomware in play, any home device or corporate machine can increase your risks.

Oh, and there is no logging of security groups... so you actually don't know where the bad guys came from, which Linux machine they exploited and where their command-and-control (C2) originated from. And there's no packet capture (PCAP) of when the attack happened, so incident response (IR) is flying pretty blindly waiting for the next attack.

Recommendations:

- Deploy advanced network security that provides visibility into traffic flows and DNS queries across your instances, VPCs, PaaS and correlates the cloud asset information with threat intelligence.
- Deploy advanced network security that actually inspects the content of the traffic (WAF for web traffic, IDS/IPS, and AV for all traffic), especially encrypted traffic that is being aggressively used by hackers to exfiltrate data or download their ransomware toolkit

2 USING THE DEFAULT OUTBOUND SECURITY GROUP OF 0.0.0.0/0 (ALLOW ANY/ALL)

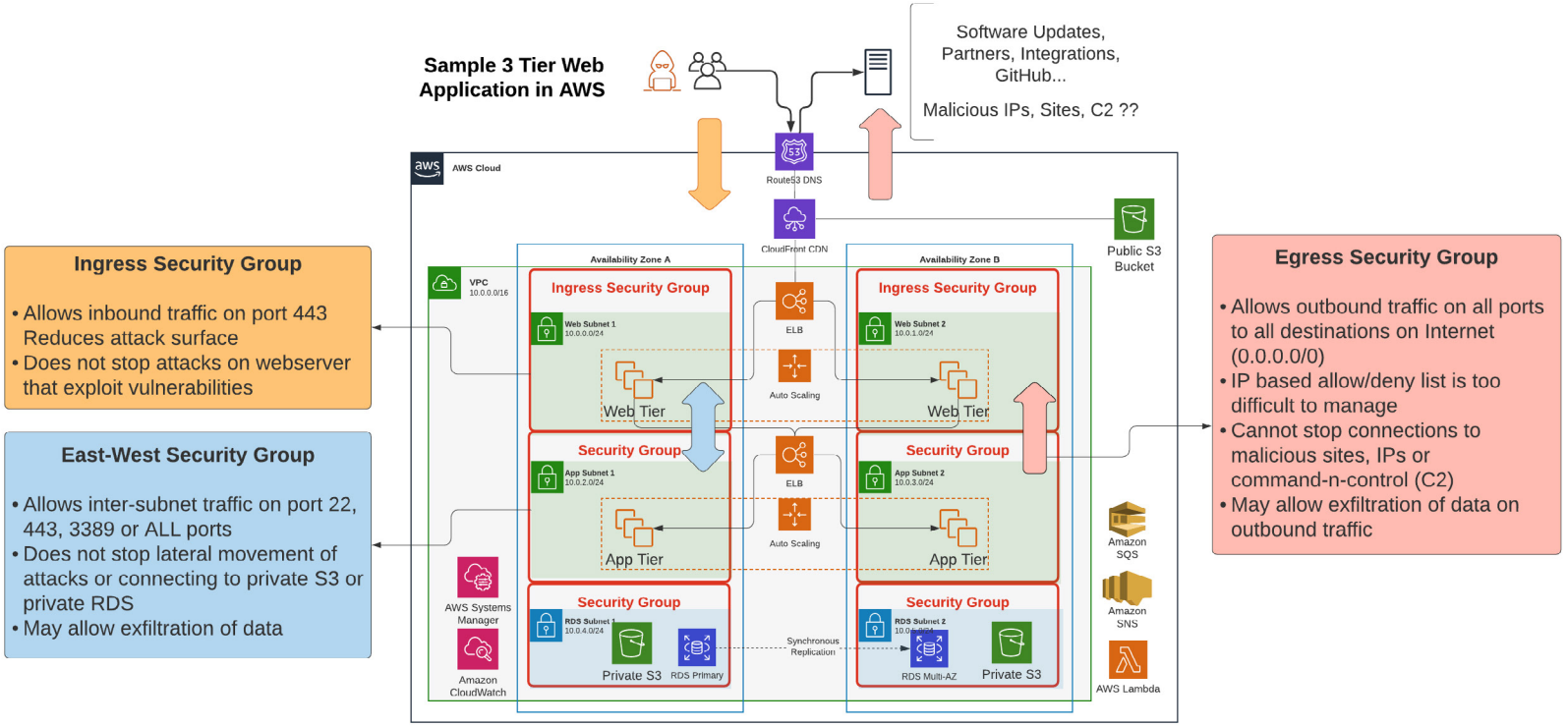
This default is one of the biggest risks for any cloud environment, whether it's sensitive/confidential/production or developer/QA accounts. Since security groups have no logging of allowed/denied traffic, you don't know if data is being exfiltrated and which instances are involved. And trying to correlate all the different AWS logs and GuardDuty findings won't help if you can't trace the entire attack killchain in one single place.

A big reason outbound security groups are wide open is that it's practically impossible to list all the different IP addresses you want to allow (good destinations) or deny (known bad guys). And, to do this based on the workload profile: Dev wants full access to all of GitHub.com, while PCI and prod must be restricted to GitHub.com/myOrgRepo and a few domains or URLs. See the right side of Figure 1 for summation and how it fits into typical environments.

Recommendations:

- Implement GeoIP controls and FQDN/URL filtering to ensure you're only connecting to approved FQDN/URL categories and specific approved sites, while blocking malicious sites and unnecessary countries.

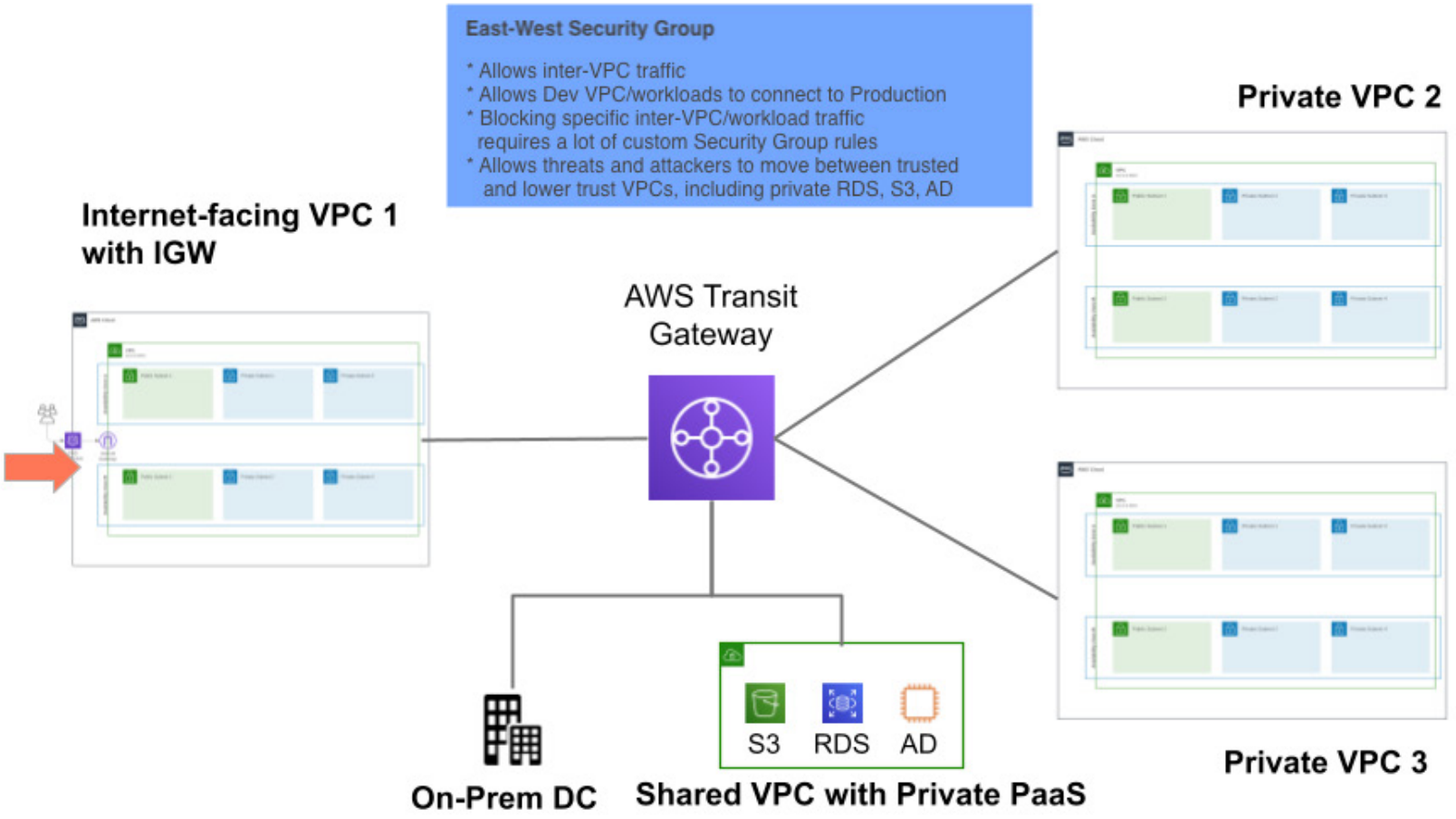
Figure 1: AWS web application 3 tier example (white)



3 LEAVING EAST-WEST SECURITY TO CHANCE

Gone are the days when you only had 5 VPCs maximum. AWS best practices recommend creating the smallest blast radius, i.e. each application gets its own VPC and perhaps runs in a separate account. Some organizations allow each team or developer to have their own VPC and even AWS account. This makes sense, but you face the challenge of now connecting them and ensuring the appropriate level of security. A common design pattern in AWS is to use a hub-n-spoke design with AWS Transit Gateway providing inter-VPC connectivity. And shared services such as Active Directory, common file shares, and databases are in a services VPC. This raises the question of how do you ensure lots of low-trust VPCs (say developers, test/QA, partners) don't have the same broad network access as critical ones (production, compliance)? The same goes for on-premises networks connecting into your cloud infrastructure.

Figure 2: Inter-VPC Transit Gateway



Recommendations:

- **Get visibility:** Even if VPCs and applications of the same trust level are talking to each other, a broadly open security group between VPCs gives you no information on whether all's well or attackers are moving laterally.
- **For high-trust VPCs:** Implement advanced network security that inspects even encrypted traffic (transparent forward proxy) to ensure that malware is not moving around and data is not being exfiltrated with DLP and URL filtering.
- **For low-trust VPCs:** For internal users such as developers and test/QA VPCs implement access control based on tags, i.e. "dev" VPCs cannot connect to prod VPCs or "partner" VPCs cannot connect to "PCI" RDS. Use of tags is now commonplace in AWS for cost allocation. You can also easily implement FQDN filtering on egress traffic using forwarding mode inspection without decryption to ensure that "dev" VPCs are not connecting to malicious or inappropriate sites. Ideally, you want to also decrypt, inspect and re-encrypt this low-trust traffic to reduce your attack surface. If you have an auto-scaling network security solution with low maintenance overhead then a single solution can work across the board.

4 THINKING YOU'RE SECURE BECAUSE YOU HAVE CSPM

Cloud security posture management (CSPM) tools like Palo Alto Networks Prisma Cloud, Check Point Dome9, CloudCheckr etc. are absolutely critical to ensure good security hygiene of your cloud environment. And, AWS offers some great tools like AWS Config and AWS SCP that perform similar features. These are necessary, but not sufficient.

What's missing here is that there is no actual protection against:

- Attacks that exploit vulnerabilities in your software whether it's Linux systems or any of the applications (NGINX, WordPress, Joomla, etc).
- Lateral movement of threats, especially from frontend to backend and other connected systems across all the VPCs that maybe even be private.
- Exfiltration of data

Protection against malware and ransomware is what CSPMs cannot handle, i.e. proactively stop attacks from happening. It's a bad idea to wait for bad things to happen and then respond.

Recommendations:

- Implement advanced inline network security that provides a defense-in-depth approach that complements the CSPM and cloud-native controls.
- Based on the identity, trust boundaries, and context of the workload, you can define the depth of inspection.

Obviously, there's a lot to learn about native controls and how and where to use them – and where they need augmentation. We'll move a little faster through the remaining mistakes, which are in the areas of:

- Visibility
- PaaS Security
- Process and Culture

And we've got a bonus item too. More on that in a minute.

GET THE RIGHT VISIBILITY

You can't secure what you can't see. For app and infrastructure awareness, placement of controls, and the ability to sleep at night, visibility is key.

5 EQUATING CLOUD LOGGING WITH GREAT VISIBILITY

There's a lot of logging options in AWS and all clouds: CloudTrail, CloudWatch, and service-specific logging in S3 and CloudWatch. Lots of logging does not equal relevant visibility to attacks or exfiltration. Does this visibility combine cloud asset information in real-time with traffic logs and threat intelligence? Does it allow you to figure out if an attacker from a malicious IP compromised the frontend, moved laterally to the highest value asset, and downloaded a ransomware toolkit?

The point here is not to say that all of the AWS logging options are worthless, far from it. The goal of logging is to help solve specific problems in terms of protection first and then incident response.

Recommendations:

- Decide on where you need visibility. Areas to consider in addition to the cloud: Ingress, egress, east-west, PaaS flows.
- Decide on the highest priority security threats and initiatives of the organization.
- Determine how you will get this visibility with a minimum number of tools across multiple accounts and clouds.

6 VPC FLOW LOGS AND ROUTE53 DNS QUERIES ARE JUST TOO BASIC TO BE OF ANY USE

This is an easy blindspot to overlook. Logs of Route 53 DNS queries tell you about intent (where are the attackers looking to connect) and VPC flow logs tell you something about actual behavior (where did they actually connect to), both good and bad. Amid the flood of logs (see above) it's important to not lose focus on data that can give you meaningful information.

Recommendations:

- While VPC flow logs and DNS queries can give you meaningful information, their volume can create the classic haystack problem. What you want to do is correlate this deluge of information with relevant information: workloads (instances, VPCs, and their meta-data) and threat intelligence. This combination of asset information, traffic logs, and threat intelligence allows you to see the malicious IPs and sites from the strange and benign.

PAAS IS NOT AUTOMAGICALLY SECURE

Just because scrutiny of some PaaS services is tough, doesn't mean you can ignore securing access – those services still contain your data.

7 RELYING ON THE SHARED SECURITY MODEL FOR PAAS SECURITY

AWS (and others') shared security model says that they protect their IaaS and PaaS infrastructure really well, and it's the customer's responsibility to enable the right knobs: setup correct IAM permissions, enable encryption, do logging etc. And it's AWS' responsibility to ensure the security of the PaaS itself, i.e. patch it and protect against attacks – and they do that brilliantly. But how do you know you've configured and architected your AWS cloud services, especially the services that create VPC endpoints correctly? Are there stranger things taking place?

Use of AWS breaks down into the following modes of deployment (the cloud perimeter):

- Account management: True out-of-band services like IAM and AWS management console. CSPM tools help here.
- Public PaaS: AWS S3 buckets that host static content like images, files etc, or Lambda functions and API server endpoints. CSPM and AWS offerings like Config and Macie help you ensure these are configured correctly.
- Your VPCs: This is where your IaaS applications live and often there are 10s and 100s of VPCs most of them private and connected as a hub-n-spoke around the AWS Transit Gateway.
- Private PaaS: These are completely private to your VPC's and hold the crown jewels of any organization: RDS databases, ElasticSearch logs, private S3 buckets that hold sensitive data.

It's these last two categories that represent the biggest emerging threat since it's assumed that this is private to your VPCs and doesn't need special protections. It's a fallacy to assume that perimeter protections are sufficient.

Recommendations:

- Access all your private PaaS with Private Link and VPC endpoints with appropriate security groups to limit access from outside.
- A best practice is to put all your Private Links and VPC endpoints in a single shared VPC where your common services live. Everyone must access these through a hub-n-spoke design across AWS Transit Gateway. This gives you two benefits (a) reduce the sprawl of VPC endpoints spread across 10s of application VPCs, (b) enables stricter access control to important PaaS assets like your production RDS. You can now get visibility and control of your PaaS and a single place to manage them. "dev" can only access approved "dev" PaaS, and "prod" can only connect to "prod" PaaS instances but with a stricter inspection policy.

8 I CAN PROTECT MY S3 BUCKETS AND RDS DATABASE WITH IAM CONTROLS AND CSPM

Once you become an expert in IAM it's a powerful feeling. It allows you to set access policies, permissions, and configuration settings in a clear and concise manner. And AWS SCP and CSPM tools can be used to ensure that they meet compliance requirements and company policies. Werner Vogels, AWS CTO, frequently talks about avoiding single points of failure. And, relying on IAM to protect your PaaS is one of them. Recent vulnerabilities of cloud services like Azurescape and tens of S3 incidents confirm this hypothesis.

One of the common breaches in public clouds today happens when an attacker has somehow landed on a frontend instance (breached the marketing webserver through a WAF misconfiguration, malicious insider...) and then uses the valid IAM privileges of that machine to legitimately connect to other systems, including RDS databases, to exfiltrate data. This is not AWS' fault. Yes, the attacker used the approved IAM credentials of a breached machine to move laterally.

To protect against misconfigurations, lateral moving attackers and malicious insiders take the following steps:

Recommendations:

- Get visibility to your PaaS traffic. See the above recommendation.
- For critical applications (production, compliance) implement advanced network security that provides a defense in depth approach, i.e. ensure that "prod" VPCs can only connect to prod DynamoDB, or "PCI" instances (EC2 or EKS) can only connect to the RDS used for PCI-DSS applications.

BUSINESS PROCESS & CULTURE IMPROVEMENTS MATTER TOO

Yes, the last two are really meant to improve your network security. Tactical recommendations can only take you so far in improving your security posture. To truly get the agility and digital transformation benefits of adopting public clouds, you seriously need to consider these.

9 ASSUMING YOU DON'T HAVE THE RESOURCES TO PROPERLY SECURE YOUR ENVIRONMENT. AUTOMATE ALL YOU CAN, SO SECURITY TEAMS CAN FOCUS ON WHAT YOU CAN'T

All of the above recommendations are possible to implement if you have infinite resources and patience. The real answer, of course, is to build network security into your security operations, DevOps, or overall automation processes. This is relatively easy for the basic network security you'd configure via Security Groups and IAM controls. The real challenge comes for advanced network security that has to support TLS decryption, forward or reverse proxy, auto-scaling, and cloud networking, and deal with this at-scale (10s and 100s of VPCs and many AWS accounts). By automating network security (basic and advanced) you ensure that security teams actually do what is hardest, continuously innovate, and have cycles to enable threat hunting and incident response.

Forget the jargon about DevSecOps or SecOps etc; what you are trying to do is bake security into the build, deploy and run process. If this concept is new to your organization or company, then start with a single project or team to build experience and lessons that can be shared for adopting this across your entire organization.

Recommendations:

- Select an infrastructure-as-code (IaC) automation that gives flexibility in terms of design patterns and workflows, supports multiple clouds, and has an open architecture that enables third-party integrations. While AWS CloudFormation can be great for small teams, mid-sized to large deployments should consider Terraform. And more importantly, ensure that your advanced network security solution works smoothly, and with no compromises. There are too many Terraform plugins (called providers) that seem to automate security, but then require additional support scripts in Python, Ansible, or Go.

10 ALLOWING SILOS TO PERSIST. SECURITY IS NOT AN ISLAND

It is still far too common that application teams are moving at cloud speed, and the security team is still dealing with tickets to open port X to make an app go live. This is the siloed datacenter approach.

Recommendations:

- Security should be an active participant in the architecture and design of applications, not consulted once the design is complete. The best organizations include a security practitioner who participates in design discussions.
- Security should be setting the policies and overseeing the implementation. Actual implementation should be in the hands of the DevOps or operations team.

BECAUSE AT VALTIX, OURS GOES TO ELEVEN, AN ADDITIONAL MISTAKE THAT WE'VE SEEN FAR TOO OFTEN:

11! BRINGING A DATA CENTER MINDSET TO THE CLOUD.

We've seen this from both end-user organizations and technology vendors. You'll see this in everything from control processes to architecture and implementation. A classic example is setting up systems with classic active-passive high availability (HA). The active handles all the traffic and passive (aka standby) monitor it, and when the passive device detects a failure in the active device, it takes over. Legacy next-generation firewalls (NGFWs), web application firewall (WAFs), and load balancers all use this design pattern.

This approach made sense in traditional data centers and on-premises networks because you had control of the networking infrastructure (especially layer 2) so that you actually had a sub-second failover, and you got session-level failover. I'll spare you the details of why this works on-prem (see GARP) and why you spent 2X the money for the two devices. Public clouds give you no control over layer 2, and hence the active-passive failover implementations are a complete kludge: the failover is more like a disaster recovery (DR) setup where the failover takes several minutes because the passive device has to make cloud API calls to shift traffic from the failed active device to the passive device. The setup for these things is complex (see here and here for examples), the operating model is very brittle with lots of moving parts/scripts and no service-level agreement (SLA) on uptime, and you actually don't get session failover while paying for 2 devices and getting only 1 in active service at any time.

Another example might be using legacy HSMs for secrets management. Regardless, the point is that while many of the disciplines of network security remain constant, it's a huge mistake to simply "lift and shift" the implementation from DC to cloud when the environments are completely different.

Recommendations:

- If you go cloud, go all the way, the right way. When architecting for public clouds, evaluate if your existing design pattern has a better replacement that is cloud-native. Ask your cloud provider's solution architect and vendors for best practices that don't perpetuate the old designs.
- For high availability, redundancy and scalability use the cloud-native constructs: cloud-native load balancing across multiple AZs (and multiple regions if you need that level of uptime and redundancy), health monitoring and telemetry from a control plane with an SLA, built-in self-healing from the control plane which replaces unresponsive instances, auto-scaling from the control plane. The beauty of this architecture is that it works at all scales: small setups use 2 AZ deployments with 1 instance each as a minimum, larger ones just have a higher auto-scaling maximum with more AZs. The control plane scales the infrastructure automatically, on-demand.

So there you have it *– 10 (11!) common network security mistakes we see frequently enough to make a top 10 list, and some recommendations on how to address them.*

IS THERE A HOLISTIC APPROACH TO AWS NETWORK SECURITY?

Since you asked... Many of these challenges are precisely where Valtix can help. We provide app workload protection in AWS (as well as the other major clouds) through network security. Our model greatly simplifies security in AWS with a single policy for all of network security including advanced controls such as FQDN filtering, IPS/IDS, DLP, anti-virus, etc. We also streamline deployment with no infrastructure or agents to manage.

To learn more about Valtix, check us out at valtix.com, or take our product tour, to see how Valtix simplifies security in AWS.

Valtix is on a mission *to enable organizations with security at the speed of the cloud. Deployable in just 5 minutes, Valtix was built to combine robust multi-cloud security with cloud-first simplicity and on-demand scale. Powered by a cloud-native architecture, Valtix provides an innovative approach to cloud security called Dynamic Multi-Cloud Policy™, which links continuous visibility with advanced control.*

The result: *security that is more effective, adaptable to change, and aligned to cloud agility requirements. With Valtix, organizations don't have to compromise in the cloud. They can meet critical security and compliance requirements without inhibiting the speed of the business. Valtix has been recognized as an innovator in numerous industry awards including 2021 top honors in the "Next-Gen in Cloud Security" from Cyber Defense Magazine, SINET-16 Innovator recognition, and inclusion in Gartner's Cool Vendors in Cloud Networking report.*

Get started with a free trial and a cloud visibility report at **Valtix.com**.

HQ - Santa Clara, USA
2350 Mission College Blvd #800 Santa Clara, CA 95054
650.420.6014 info@valtix.com