Security Architecture

Palo Alto Networks Certified Software Firewall Engineer

KNOWLEDGE CHEATSHEET

INSPECT ACROSS TRUST BOUNDARIES

Segment Trust boundaries → **Identify** Applications → **Control** Least privilege → **Inspect** Allowed content

01 Security architecture and inspection

Firewall platforms and trust boundaries

- **VM-Series** runs PAN-OS as a virtual appliance in supported hypervisors and public clouds. It suits virtual data centers, multicloud deployments and lift-and-shift applications that need consistent network security.
- **CN-Series** supplies containerized network inspection for Kubernetes workloads. **Cloud NGFW** delivers managed firewall services in supported public clouds; customers still design traffic steering and security policy.
- **North-south** traffic enters or leaves an environment. **East-west** traffic moves between workloads, such as web and database tiers. Protect each required path, including internal and outbound flows.
- **Zero Trust** establishes explicit trust boundaries and least-privilege access. **Microsegmentation** narrows permitted workload communication and limits lateral movement; a shared subnet does not imply shared trust.

Application policy and content inspection

- **App-ID** identifies applications beyond port numbers. Build application-based allow rules between zones, restrict services appropriately and deny unnecessary communication.
- **Security policy** decides whether traffic is allowed. Attached **Security profiles** inspect allowed traffic for threats and unwanted content. An allow rule alone does not provide full threat prevention.
- **Threat Prevention** addresses exploits, malware and command-and-control activity. **WildFire** analyzes suspicious files for previously unknown malware; keep content and threat intelligence updated.
- **Advanced URL Filtering** evaluates web destinations and malicious web content. **DNS Security** addresses malicious DNS activity; **DNS sinkholing** redirects matching DNS responses to help identify infected clients.
- **User-ID** associates users with addresses. GlobalProtect **Host Information Profile (HIP)** data describes endpoint posture and can condition access. **App-ID Cloud Engine** supplies cloud-generated identification for supported SaaS applications.

Decryption, egress and exfiltration

- **SSL Forward Proxy** inspects supported outbound TLS sessions by acting between client and server. Clients must trust the forward-trust certificate chain; define exceptions for traffic that cannot or should not be decrypted.
- **SSL Inbound Inspection** decrypts supported connections to protected servers using the required server certificate and private key. Pair decryption with vulnerability and other Security profiles to inspect the plaintext.
- **Decryption policy** selects sessions for decryption; **Security policy and profiles** control and inspect them. Certificate validation and OCSP support certificate trust decisions rather than content scanning.

Security Architecture / VM-Series

Palo Alto Networks Certified Software Firewall Engineer

KNOWLEDGE CHEATSHEET

VIRTUAL FIREWALL DEPLOYMENT

- **Image** — Platform fit
- **Interfaces** — Traffic path
- **HA** — Failure recovery
- **Automation** — Repeatability

- **Source NAT**, including dynamic IP and port translation, changes internal source addressing on outbound traffic. Apply application controls and content inspection separately to block suspicious outbound transfers and exfiltration.
- Apply **URL and threat profiles** wherever relevant traffic is allowed, including permitted intrazone flows. Zone direction alone does not determine whether a session needs inspection.

02 VM-Series deployment and resilience

Licensing and resource lifecycle

- **BYOL** uses separately purchased entitlements; **PAYG** uses the selected marketplace offer and metered billing. Match the image and offer to the licensing model and required security subscriptions.
- **Software NGFW credits** fund eligible software-firewall deployments and subscriptions. Estimate instance count, licensed vCPUs, subscription bundle and term with the credit estimator; interfaces and IP count are not the principal compute sizing inputs.
- A **deployment profile** defines the licensed resource and subscription allocation. An **authorization code** activates purchased entitlement; a **bootstrap authentication key** authorizes firewall onboarding to Panorama. These credentials have different purposes.
- Automate supported **registration, activation, renewal and deactivation** operations through licensing interfaces. Deactivate an instance before deletion where possible; use Customer Support Portal asset management to reclaim eligible allocations from a lost instance.
- Track **license expiration**, available credits and content-update eligibility. Budget cloud compute, storage and logging in addition to firewall licensing; monitor entitlement before service disruption.

Images and network attachment

- Use a **platform-specific image**: an ESXi virtual appliance package, an AWS AMI, or an appropriate Azure or Google Cloud image. OpenStack commonly combines a **qcow2** image with a **Heat YAML** template.
- Separate the **management interface** from data interfaces. Map each virtual NIC to the intended subnet, port group and PAN-OS interface; the management NIC does not prove dataplane connectivity.
- A **virtual wire** inserts a transparent inspection pair without changing endpoint IP addresses or gateways. In ESXi, separate port groups or virtual switches can place selected guests behind the inspection path.
- **Layer 3** interfaces route between networks; assign zones and the appropriate routing instance. **Layer 2** interfaces bridge traffic. **Tap** provides visibility from copied traffic rather than inline enforcement.
- For cloud appliances, enable the platform's required **IP forwarding** behavior and account for source/destination checking. Validate subnet routes, cloud security controls and return paths around every data interface.

VM-Series

Palo Alto Networks Certified Software Firewall Engineer

KNOWLEDGE CHEATSHEET

VIRTUAL FIREWALL DEPLOYMENT

- **Image** — Platform fit
- **Interfaces** — Traffic path
- **HA** — Failure recovery
- **Automation** — Repeatability

Routing and cloud traffic insertion

- A **default route** sends unmatched destinations, typically 0.0.0.0/0, toward the internet or upstream router. Configure its next hop and egress interface in the firewall's routing instance.
- **BGP** and **OSPF** exchange routes dynamically. Route redistribution and tagging control how routes move between protocols; the **Advanced Routing Engine** uses logical routers and routing profiles.
- Treat an **Advanced Routing Engine** change as a planned migration: check supported configuration conversion, activation and reboot requirements before changing an existing firewall.
- AWS **Gateway Load Balancer (GWLB)** exchanges encapsulated traffic with VM-Series through **GENEVE**. **GWLB endpoints (GWLBE)** insert the inspection service into VPC routes; allow the required encapsulated traffic and health probes.
- Maintain **flow symmetry** so both directions traverse the correct stateful inspection path. Interface count and NAT placement depend on the deployment architecture; use its actual traffic diagram and route tables.

HA links, monitoring and split brain

- **Active-passive HA** keeps one forwarding peer and a standby. **HA1** carries control information and heartbeats; **HA2** synchronizes supported session and runtime state.
- **Heartbeat loss**, monitored link failure and configured **path monitoring** can influence failover. Path monitoring tests reachability beyond the local interface; define failure conditions to avoid unnecessary transitions.
- Use a separate **HA1 backup** and supported **heartbeat backup** path to reduce split brain. Losing control communication can make both peers believe the other has failed; session synchronization alone does not solve that problem.
- Align peer **software, capacity, licenses and subscriptions** with platform requirements. Place peers across separate failure domains where supported and test both traffic recovery and state synchronization.
- In supported **active-active HA**, floating IPs can preserve gateway addressing and **HA3** carries packet forwarding between peers. Cloud load-balanced active nodes use a different architecture; validate the cloud's supported HA mode and address failover mechanism.

VM-Series

Palo Alto Networks Certified Software Firewall Engineer

KNOWLEDGE CHEATSHEET

VIRTUAL FIREWALL DEPLOYMENT

- **Image** — Platform fit
- **Interfaces** — Traffic path
- **HA** — Failure recovery
- **Automation** — Repeatability

NSX and Cisco ACI integration

- **VMware NSX** service insertion steers selected workload traffic through VM-Series. Supported designs include **host-based** and **service-cluster** insertion; choose the model compatible with the NSX integration.
- **Panorama**, NSX Manager and vCenter coordinate service registration, deployment and policy. Ensure deployed firewalls can reach Panorama and receive the assigned device-group policy and templates.
- **NSX steering rules** determine which flows reach inspection. Security-group membership and workload metadata support dynamic segmentation; successful service deployment alone does not guarantee traffic redirection.
- In **Cisco ACI**, endpoint groups communicate under **contracts**. A **service graph** inserts the firewall as a service device, and **policy-based redirect (PBR)** steers matching flows through it.
- Keep **fabric forwarding** and **firewall enforcement** aligned. Verify service-device connectivity, health and symmetric redirection before troubleshooting application rules on an appliance that receives no traffic.

Automation, plugins and dynamic objects

- **Terraform modules** package reusable infrastructure definitions. **CloudFormation** provisions AWS resources; **Ansible PAN-OS modules** automate configuration through supported APIs, commonly executing locally on the control host.
- **Bootstrapping** supplies initial settings, Panorama connectivity and other supported startup artifacts. Keep credentials protected and make repeatable deployment steps explicit; infrastructure creation and firewall policy configuration are separate stages.
- The firewall's **VM-Series plugin** supplies platform-specific capabilities and can be updated separately from PAN-OS within compatibility constraints. Panorama cloud plugins handle their own monitoring and orchestration functions.
- **Dynamic address groups (DAGs)** select registered IP addresses through tag expressions. Cloud or hypervisor integrations update membership as workloads change, so rules can remain stable while addresses and instances change.
- Panorama **monitoring definitions** authenticate to cloud APIs and collect workload metadata, such as EC2 VPC and subnet information. **Notify groups** distribute learned IP-to-tag mappings to selected firewalls, including on-premises enforcement points.

VM-Series / CN-Series & Kubernetes

Palo Alto Networks Certified Software Firewall Engineer

KNOWLEDGE CHEATSHEET

VIRTUAL FIREWALL DEPLOYMENT

- **Image** — Platform fit
- **Interfaces** — Traffic path
- **HA** — Failure recovery
- **Automation** — Repeatability

Performance and targeted diagnostics

- The **management plane** handles administration; the **dataplane** handles forwarding and inspection. A reachable GUI after a host reboot can coexist with dataplane failure; check system health, storage, interfaces and traffic processing separately.
- Supported **dataplane core customization** reallocates existing vCPUs between management and dataplane. It does not add total vCPUs; retain adequate management capacity and check feature compatibility before changing the split.
- Use **show plugins vm_series dp-cores** to inspect the allocation. Supported bootstrap configurations use **plugin-op-commands=set-dp-cores:<count>** in init-cfg.txt; apply the documented change procedure, reboot and verify the result.
- **Intelligent Traffic Offload (ITO)** uses supported SmartNIC/DPU infrastructure to offload eligible flows. Initial traffic classification determines whether traffic needs VM-Series inspection or can follow an offloaded path; mode and hardware support depend on the release.
- For missing VLAN traffic, inspect **hypervisor port groups, VLAN tagging and NIC mapping** first. Use dataplane captures, sessions and filtered global counters for forwarding faults; **tcpdump** on management traffic answers a different question.
- For failed AWS tag retrieval, check credentials and **show plugins aws vm-mon-status**. On Panorama, **less plugins-log plugin_aws_ret.log** examines retrieval; plugin_aws_proc.log covers processing and plugin_aws.log covers plugin operation.

03 CN-Series and Kubernetes

Pod-aware network inspection

- A **pod** is Kubernetes' smallest deployable unit and can contain multiple containers sharing networking. CN-Series applies network security to protected pod traffic, including communication between application trust zones.
- An **external perimeter firewall** sees only traffic routed to it. Pod-to-pod traffic can remain inside the cluster, while overlays, NAT and rapidly changing addresses complicate workload attribution.
- CN-Series combines **Layer 7 visibility**, application policy and threat prevention for selected **east-west**, ingress and egress paths. Outbound inspection can block suspicious transfers from compromised workloads.

CN-Series & Kubernetes

Palo Alto Networks Certified Software Firewall Engineer

KNOWLEDGE CHEATSHEET

KUBERNETES ENFORCEMENT

- **Panorama** — Central policy
- **CN-MGMT** — Management
- **CN-NGFW** — Inspection
- **CNI** — Traffic steering

- **CN-MGMT** provides the management plane; **CN-NGFW** supplies dataplane inspection. Panorama centrally manages policy, while the Kubernetes integration discovers workloads and coordinates the cluster-specific configuration.
- Use supported Kubernetes environments such as **EKS, GKE, OpenShift and native Kubernetes** with the matching deployment assets. Verify CNI, kernel, image and platform compatibility before installation.

Manifests, permissions and insertion

- Kubernetes **YAML manifests** declare desired resources; **Helm** packages and parameterizes them. Select assets for the environment and deployment model rather than mixing files from different examples.
- A **DaemonSet** places pods on eligible nodes. A **Kubernetes Service** supplies a stable service abstraction; CN-Series service-based designs separate inspection capacity from individual application nodes.
- Create the required **service accounts and RBAC bindings** before components use them. Panorama monitoring needs a valid API endpoint, CA trust and authorized credentials; handle tokens and secrets as credentials.
- Prepare **ConfigMaps, Secrets, storage and services** in the deployment's required order. ConfigMaps hold configuration, Secrets hold sensitive values and persistent volumes preserve management data across pod replacement.
- The **CNI integration** and workload selection determine whether pod traffic enters inspection. Check required annotations or labels, service readiness and dataplane connectivity; a Running pod alone is not proof of enforcement.

Panorama onboarding and licensing

- The Panorama **Kubernetes plugin** configures cluster monitoring and CN-Series licensing. Match its supplied network templates to the chosen deployment model; template names and capacities vary by plugin release.
- For credit-based licensing, size the **CN-NGFW dataplane vCPUs** and required subscriptions. Activate the appropriate entitlement in the Customer Support Portal and make the allocation available through Panorama.
- Separate **licensing entitlement** from **Panorama authentication**. CN-MGMT onboarding also requires a valid bootstrap authentication key, correct Panorama settings and reachable management services.
- On Panorama, **request plugins kubernetes get-license-tokens** displays authorization and vCPU allocation information. Check expiration, issued versus used capacity and cluster connectivity when licensing fails.

CN-Series & Kubernetes / Cloud NGFW

Palo Alto Networks Certified Software Firewall Engineer

KNOWLEDGE CHEATSHEET

KUBERNETES ENFORCEMENT

- **Panorama** — Central policy
- **CN-MGMT** — Management
- **CN-NGFW** — Inspection
- **CNI** — Traffic steering

Deployment checks and runtime defense

- Use **kubectl get pods -n <namespace> -o wide** for status and placement; **kubectl describe pod** and **kubectl logs** expose scheduling, image, storage and startup faults. Use **-l** for a label selector.
- On **CN-MGMT**, **request plugins vm_series list-dp-pods** lists attached dataplane pods. Verify licensing, the committed policy and real test traffic; correlate sessions and Traffic or Threat logs with the protected workload.
- **Prisma Cloud Compute runtime defense** complements network inspection with workload behavior controls. Narrow container responsibilities make process and behavior allow lists practical, but learning must be reviewed before enforcement.
- **Vulnerability scanning** finds known weaknesses in images and packages; **runtime controls** detect or block suspicious behavior while workloads execute. Runtime protection can address novel attack behavior without automatically patching the image.

04 Cloud NGFW and cloud operations

Managed service and deployment models

- **Cloud NGFW for AWS** is a managed network security service subscribed through AWS Marketplace. Service management reduces appliance operations; policy, IAM permissions, routing and protected traffic scope remain design responsibilities.
- A **distributed** design places inspection resources near individual workload VPCs. A **centralized** design shares inspection through a security VPC and commonly uses **Transit Gateway (TGW)** for connected VPCs.
- Assess **inbound, outbound and east-west** flows separately. VPC-local routing and inter-VPC TGW routing require different route entries; deploying an endpoint does not automatically redirect every workload flow.
- **Cloud NGFW for Azure** integrates managed inspection with Azure networking. VM-Series remains an appliance option when the design requires direct PAN-OS control or a consistent appliance model across cloud providers.

Cloud NGFW

Palo Alto Networks Certified Software Firewall Engineer

KNOWLEDGE CHEATSHEET

MANAGED CLOUD INSPECTION

- **Scope** — VPCs and flows
- → **Route** — NGFW endpoints
- → **Enforce** — Rulestacks
- → **Observe** — Logs and metrics

Endpoints, load balancers and return paths

- An AWS **NGFW endpoint** is a route target into the inspection service. Associate the correct endpoint and subnet routes in each participating Availability Zone and design for endpoint and zone failure.
- In a typical **distributed outbound** path, workload traffic reaches the inspection endpoint before the NAT gateway and internet gateway. Build the return routes so responses traverse the matching inspection path.
- For **inbound inspection**, an internet-gateway edge-associated route table can steer destination subnet traffic to an inspection endpoint. The endpoint subnet then needs onward routing to the application or load-balancer subnet.
- A centralized **ALB** terminates HTTP(S) connections and creates backend connections. Routing toward a remote application depends on the ALB subnet's routes, TGW attachment and application return path; do not infer the next hop from the ALB alone.
- **Security groups**, network ACLs and route tables govern reachability at different layers. They complement application and content inspection; **NAT gateways** translate addresses rather than enforce NGFW threat policy.

Rulestacks, hierarchy and tenant roles

- A **rulestack** groups Security rules, objects and profiles. Rules select allowed traffic and profiles define content inspection, allowing consistent enforcement across associated Cloud NGFW resources.
- A **local rulestack** supplies application or account-specific rules. A **global rulestack**, integrated with AWS Firewall Manager, supplies organization-level pre-rules and post-rules.
- Policy evaluation follows **global pre-rules, local rules, global post-rules**. Put mandatory controls at the appropriate level and check ordering so broad rules do not shadow more specific ones.
- **Tenant administrators** onboard accounts and manage users. **Firewall administrators** manage firewall resources; **LocalRuleStackAdmin** and **GlobalRulestackAdmin** distinguish local and global policy administration.
- **AWS Firewall Manager** applies security policy to selected accounts and organizational units and coordinates managed deployment. Define inclusion and exclusion scope deliberately rather than assuming every organization account must participate.

Cloud NGFW

Palo Alto Networks Certified Software Firewall Engineer

KNOWLEDGE CHEATSHEET

MANAGED CLOUD INSPECTION

Scope VPCs and flows → **Route** NGFW endpoints → **Enforce** Rulestacks → **Observe** Logs and metrics

Central policy and infrastructure automation

- For AWS **Panorama policy management**, link the Cloud NGFW tenant and use the **AWS plugin** with cloud device groups and associated template stacks. Centralize shared rules while retaining the intended tenant and regional scope.
- The **Cloud Services plugin** supports cloud logging integration. Policy distribution and log retrieval have separate dependencies; seeing the resource in Panorama does not prove logs are arriving.
- Use **Terraform**, supported service APIs or **CloudFormation** for the relevant infrastructure and native policy resources. Panorama-managed policy can use Panorama's interface, APIs and supported Terraform workflows.
- Enable **programmatic access** and the appropriate IAM and service roles before automation. Track the selected policy manager as the configuration authority, review plans and verify commit or deployment status after changes.

Telemetry and integration troubleshooting

- Cloud NGFW for AWS can deliver **Traffic, Threat and Decryption logs** to configured **S3, CloudWatch Logs or Kinesis Data Firehose** destinations. Check logging permissions and destination configuration as well as policy logging settings.
- **Panorama in AWS** can provide management and supported log collection. Alternative designs use on-premises Panorama with cloud log collectors or cloud logging services; size management and log storage separately.
- For VM-Series **CloudWatch metrics**, grant the instance role **cloudwatch:PutMetricData**, enable metric publishing and select the matching region and configured namespace. Metrics publishing and log forwarding are distinct pipelines.
- Cloud NGFW for Azure supports logging to **Azure Log Analytics workspaces**. Configure the intended workspace and permissions, then verify actual records rather than relying only on a connected status.
- For failed **Azure Panorama integration**, verify compatible software and plugins, device certificates, DNS and required outbound management connectivity. For Azure orchestration, verify the service principal's tenant, client identity, credential and resource permissions.