



San Francisco | March 4–8 | Moscone Center



SESSION ID: CSV-F02

Democratizing Cloud Security Our Journey To Secure The Public Cloud

Hardeep Singh

Sr Principal Security Engineer
Symantec Corporation/Cloud Platform Engineering

Yunchao Liu

Sr Software Engineer
Symantec Corporation/Cloud Platform Engineering

Why The Need To Democratize Security?

Centralized static bureaucracy

Decentralized, agile adhocracy

On Premise – Private, limited resources, defined vectors

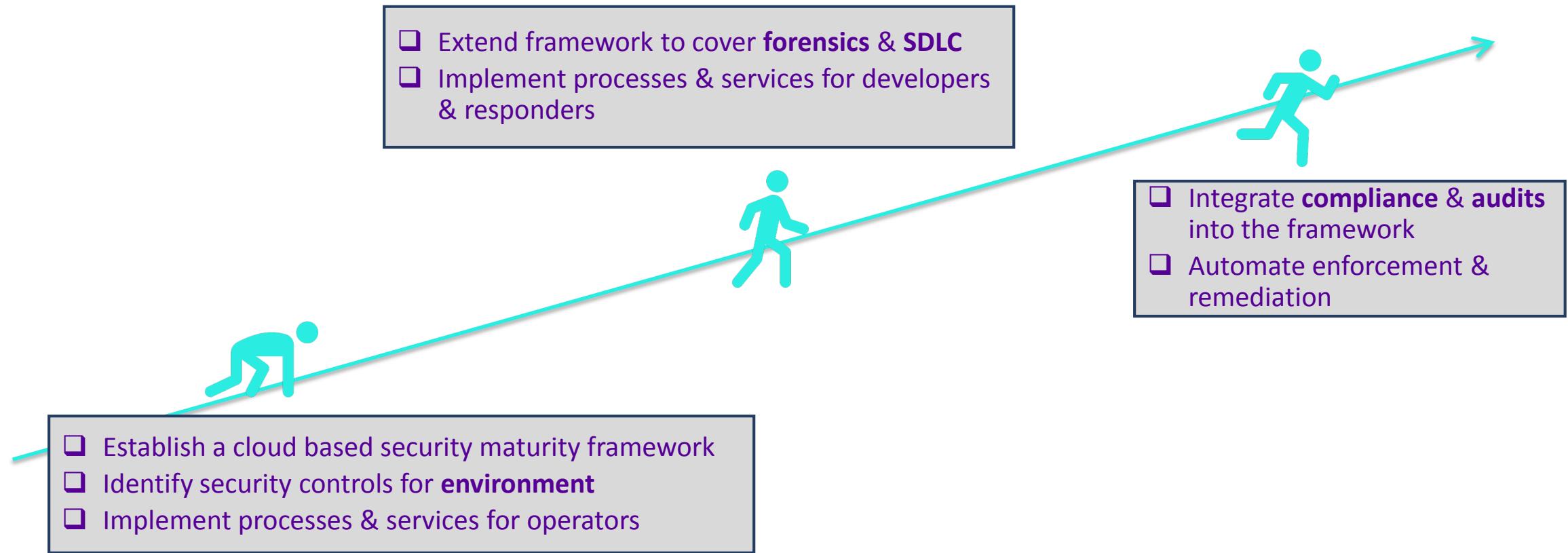
Public Cloud – Unlimited resources, unrestricted vectors

Strict separation of duties

DevOps wear many hats

Centralized SOC – Antiquated policies to secure them all

Journey To Secure The AWS Environment



Security Maturity Framework

Established a working group of security leads from all BU's

Identified **ownership** at all levels

Built consensus

Identified account onboarding process

Identified time lines for compliance

o

Defined security maturity polices

Observable, actionable & enforceable polices only

Started with baseline CIS AWS recommendations

Extended to identify insecure resources

Enhanced to cover internal policies, SDLC, forensics & compliance

Enforced compliance

- Established an exception process
- Created detailed reports on non compliance
- Automated enforcement
- **Created services to help operators, developers & responders in implementing security policies**

Security Maturity Policies

AWS Environment

- Optimize IAM users & policies
- All audit logs centralized
- Restrict public access to resources
- Encrypt all storage resources
- Define a mandatory tagging policy
- Facilitate Incident responders (reactive)

SDLC

- Golden Image - scans in build cycle
- TVM- Response to new CVE's
- Threat Modeling of all services
- Secret & credentials management
- Pen testing, game days, bug bounties
- Auditable processes

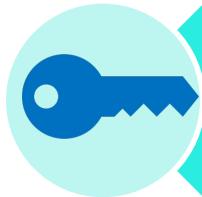
RSA® Conference 2019

Security Services

Put the Sec in DevSecOps



Key Areas Of Focus For Security Services



Access Control



Inventory & Monitoring



SDLC



Forensics

RSA® Conference 2019

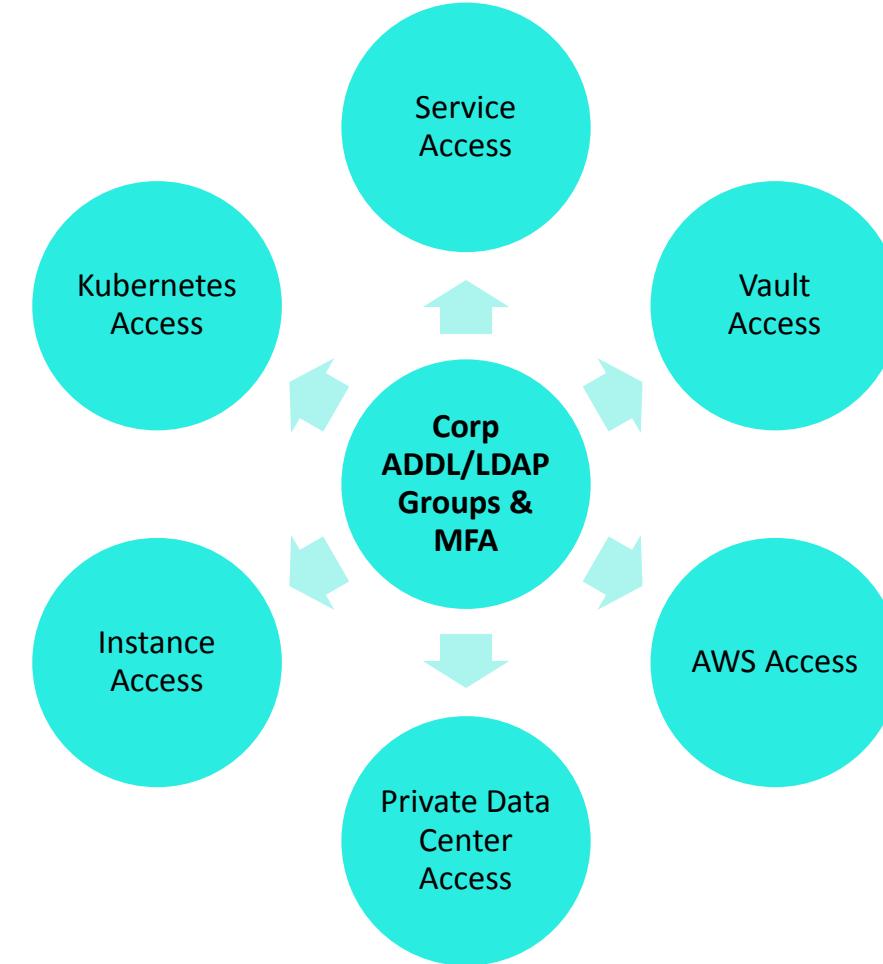
Access Control

Authentication & Authorization



Common Access Control To Rule Them All

- Federated Broker for AWS API & Console
- X509 certs for SSH, Kubernetes & Vault
- OpenID Connect for services
- LDAP Groups and Directory Lists for Roles
- MFA on authentication



Comprehensive Access Control Services

Authentication

- Corporate authentication boundary extended to the cloud
- SSO with MFA on all environments
- Ephemeral key for operators
- Long lived credentials only for services (stored in vault)

Authorization

- Corporate authorization boundary extended to cloud
- Integrated with corporate directory lists and LDAP groups
- Centralized privilege management
- Decentralized delegation
- Process to allow incident response team access

Environment

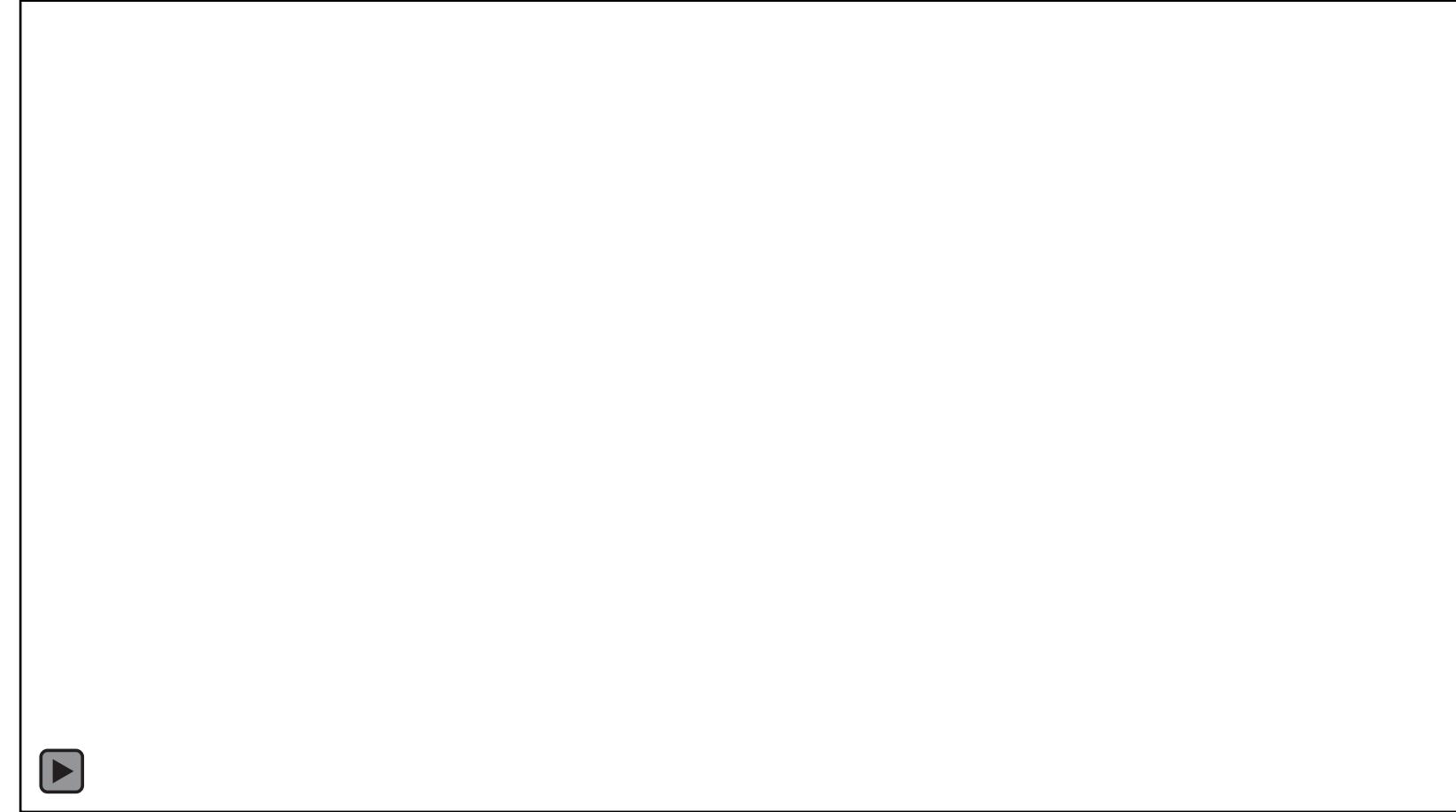
- AWS Console
- AWS API
- EC2 Instance
- On-Prem/Private Data Center VM's
- Kubernetes cluster
- Secrets Vault
- Data Stores
- Other Services

Authentication Service For DevOps- Keymaster

- Generates Ephemeral X509 Certs for authentication
 - SSH to EC2 instance & on premise VM's
 - Kerberos compatible (pkinit)
 - Kubernetes compatible
- Integrated with corporate active directory
- Provides 2FA via VIP and/or YubiKey
- Provides OpenID Connect support for web services
- Embeds LDAP groups into certificate for Kubernetes authorization
 - Embeds IP range from where certificate is valid

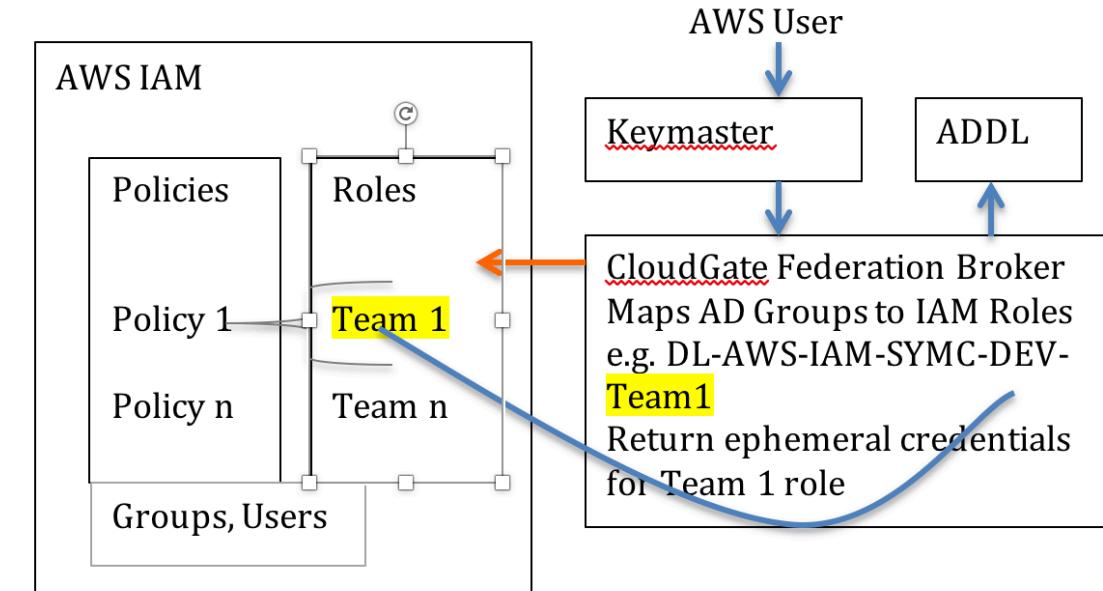
Authorization via LDAP groups- Smallpoint

- LDAP group management tool
- Integrated with corporate active directory
 - Authenticated by Keymaster
- Groups can manage other groups
- Service accounts can be managed like user accounts
- Light-weight
 - ~5K LOC



Authorization Service For AWS - Cloud-Gate

- Federated Broker for AWS API access for DevOPs
 - Maps Corp DL to AWS Roles & Policies
 - Assumes the mapped AWS roles on users behalf
- Generates ephemeral AWS API credentials for DevOps
- Integrated with Corp Active Directory
 - Authentication via Keymaster
- Enables DL owners control of membership



RSA® Conference 2019

Inventory

Feeds Reporting, Enforcement, Forensics & TVM Services



Centralized Inventory Of All Resources

- Service to catalog assets in all AWS accounts in the fleet
- Extends catalog to on premise resources
- Retain state & configuration history of every resource
- Source of truth for security & governance reports
- API access for other services
- Common view across otherwise siloed environment
- Enables operators & incident responders to track resources



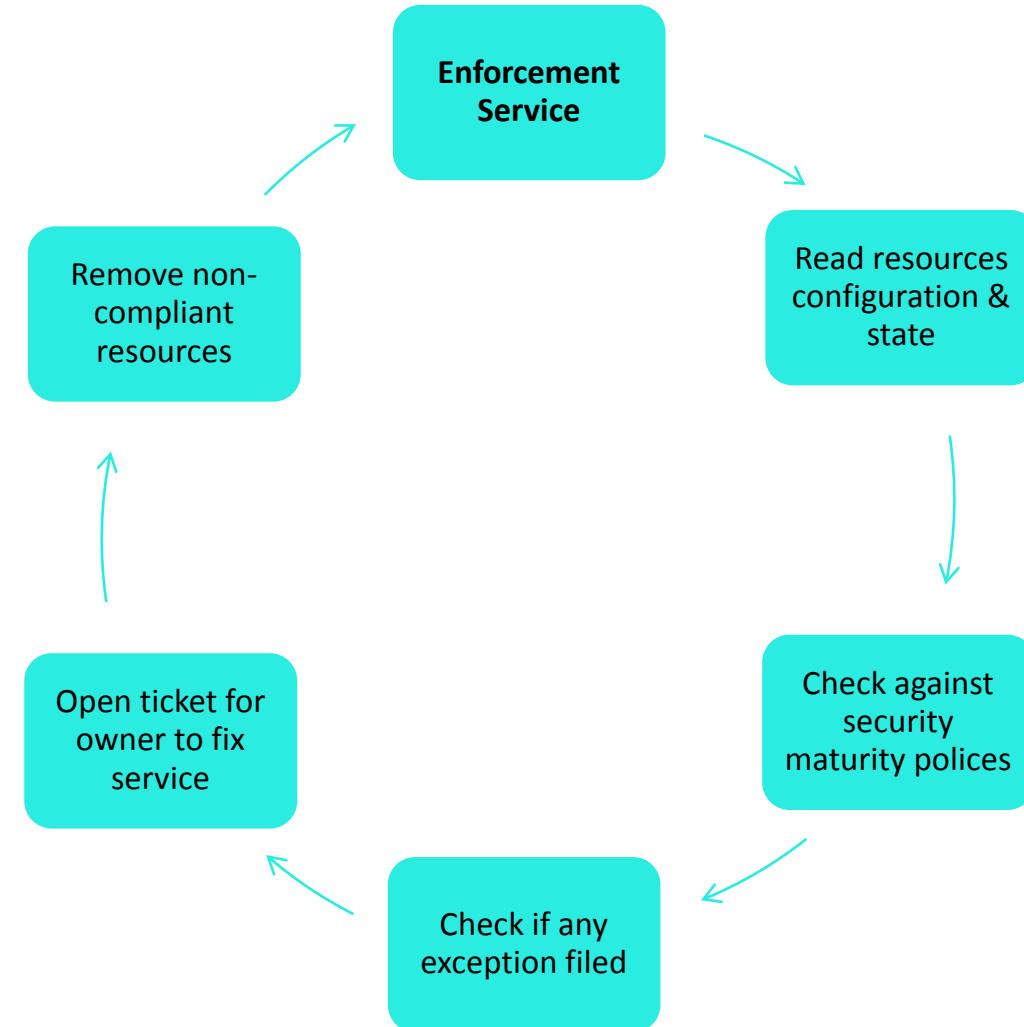
Compliance Monitoring & Reporting

- Security Maturity Analysis And Reporting Tool
- Monitors accounts for compliance to security maturity policies
- Aggregates non compliance by business unit, VP, payer accounts
- Identify on individual resource owners and send alerts
- Implements an exception process



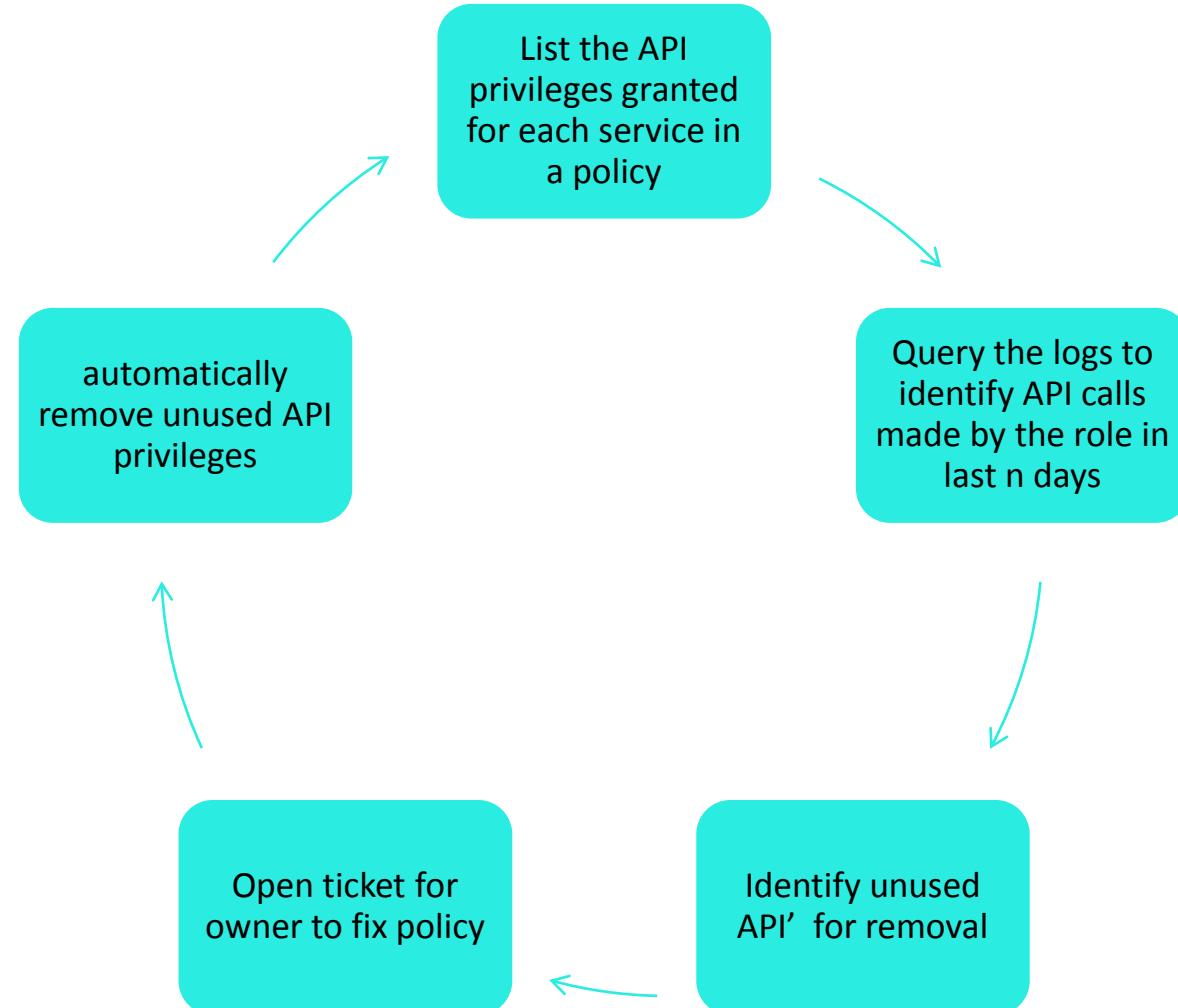
Compliance Enforcement Services

- Cloud Custodian & CWA
 - Security maturity compliance
- Reaper
 - IAM User cleanup
- RIP
 - IAM Policy Optimizer
- CWP
 - Endpoint protection



RIP - Remove Irrelevant Policies

- Access advisor limited
 - Lists services used
 - Does not identify API's
- Cloudtrail logs API calls
 - Query logs associated with IAM Role/User/Group
 - Cloudtrail does not log all API actions
- Enforce explicit API listing in policy



RSA® Conference 2019

SDLC

TVM & Service Security Assessment



Service Security Assessment

- Catalog each services security posture
- Scanning – Static, Dynamic & Image
- Audit readiness
- Track Inventory
- Security score across all services



Service Secrets Storage - Vault

- Inventory of all secrets
 - All secrets stored in vault
 - All services read from vault
- All secrets rotated randomly
 - All secrets considered ephemeral
 - Monitor vault to ensure rotation

| Target service or environment for this secret | Description (usage) of each secret in the service | Where is the secret saved at runtime | Who has access to the secret | Where is the secret stored for deployment | Who has access to the secret | What can you do with the secret | When was it last modified | What is the process for rotating the secret |
|---|---|--------------------------------------|------------------------------|---|------------------------------|-----------------------------------|---------------------------|--|
| AWS | AWS CIS Role Credentials | ACP Secrets | Operator | ACP Secrets | Admin | Assume CIS role in AWS accounts | Thu Oct 11 2018 | Generate new credentials in AWS and update ACP secret |
| JIRA | JIRA credentials | ACP Secrets | Operator | ACP Secrets | Admin | Read, Write to JIRA SMART project | Thu Feb 01 2018 | Open a ticket to get new credentials from IT and update ACP secret |

- Curl
 - curl --header "X-Vault-Token:\$VAULT_TOKEN" \$VAULT_ADDR/v1/secret/data/path_to_secret
- NodeJS
 - request.get({ url: https://volturl/v1/path_to_secret, headers: {"X-Vault-Token": vault_process.env.VAULT_TOKEN}, "Content-Type": "application/json" }, method:'GET'}, function(...
- Python


```
client = hvac.Client(url = url)
client.auth.ldap.login(
    username=username,
    password=passwd,
    mount_point='ldap'
)

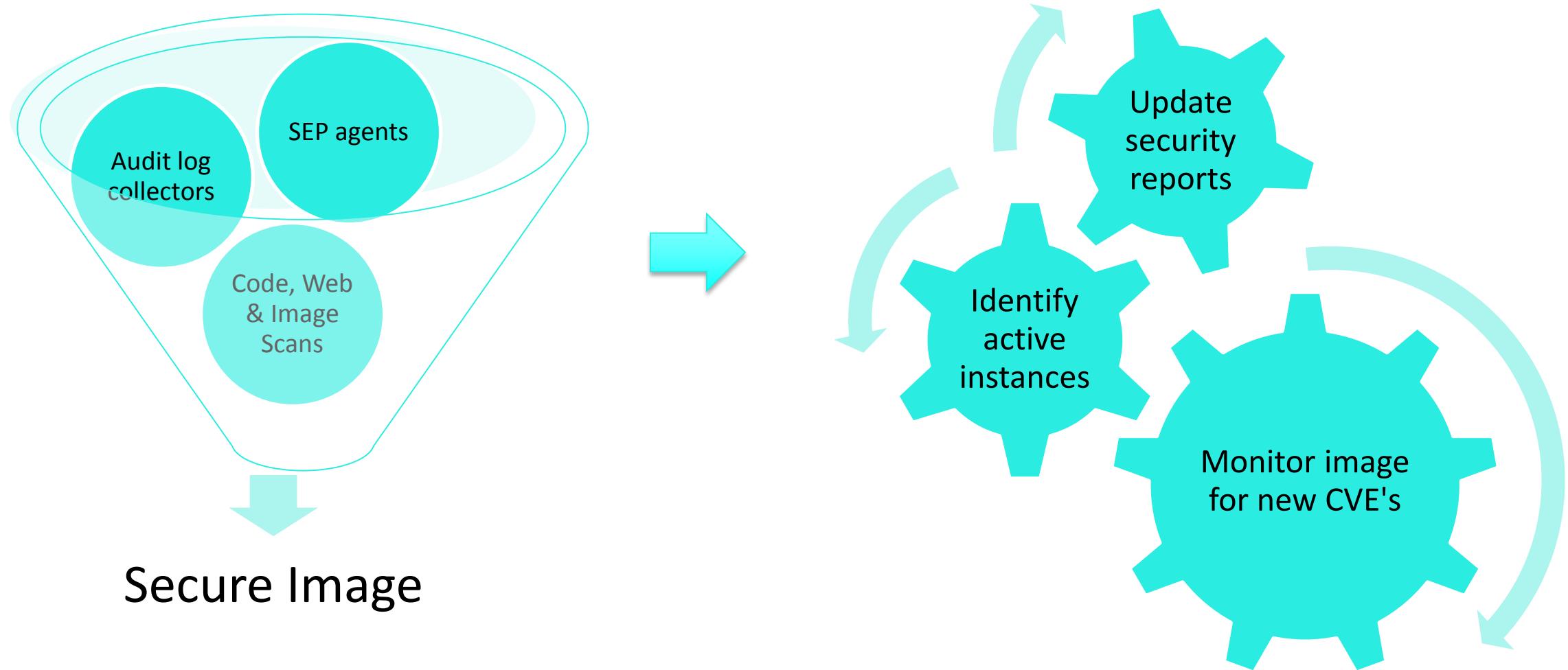
secret_version_response = client.secrets.kv.v2.read_secret_version(
    path=path,
)

print secret_version_response
```

Demo for Vault

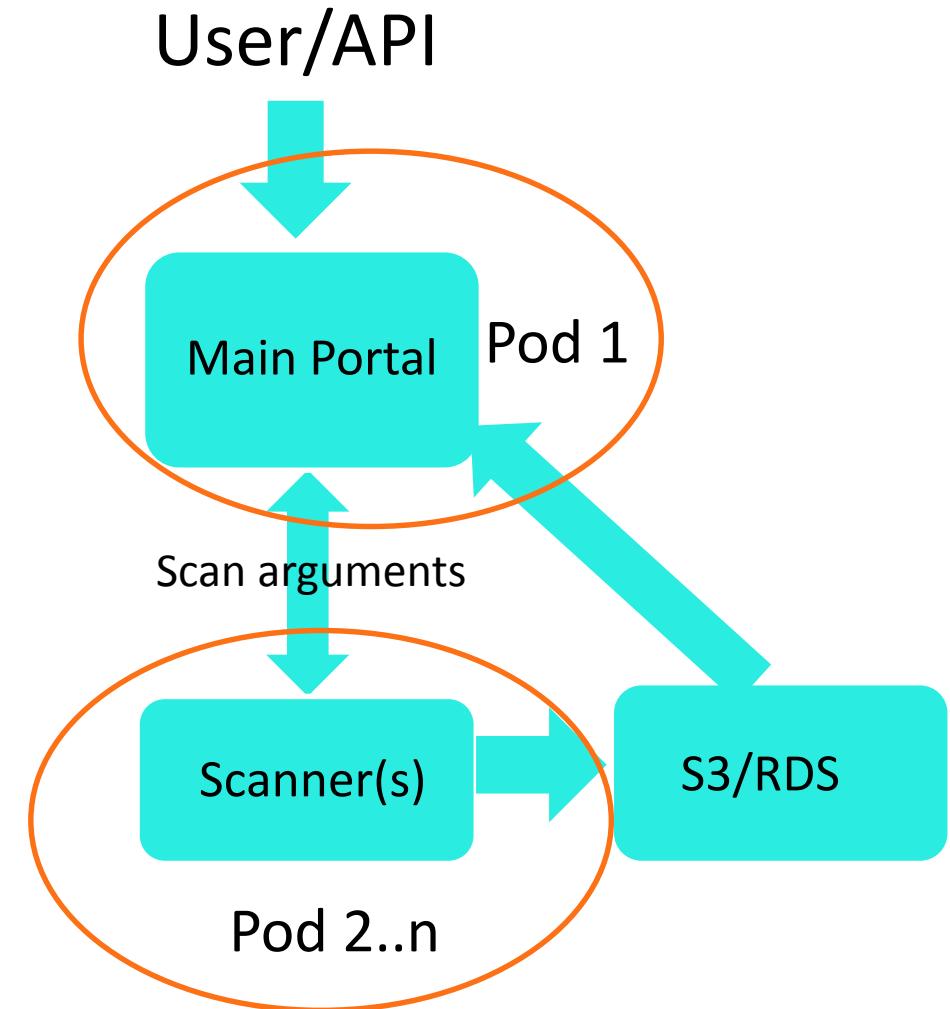


Integrate TVM With The Build Process



Scanning Services

- Scanning micro services that can integrate with CI/CD
- One or more containers per scan type
- Asynchronous (SQS)
- Results stored in S3 & RDS



Static Code Scanning Service

- SonarQube a self service
 - Install SonarQube server and scanner in containers.
 - Install plugins for more programming languages
 - SonarQube has some major languages plugins as default.
 - Configure scan
 - Scan parameters

<https://github.com/Symantec/security-scan-wrappers>

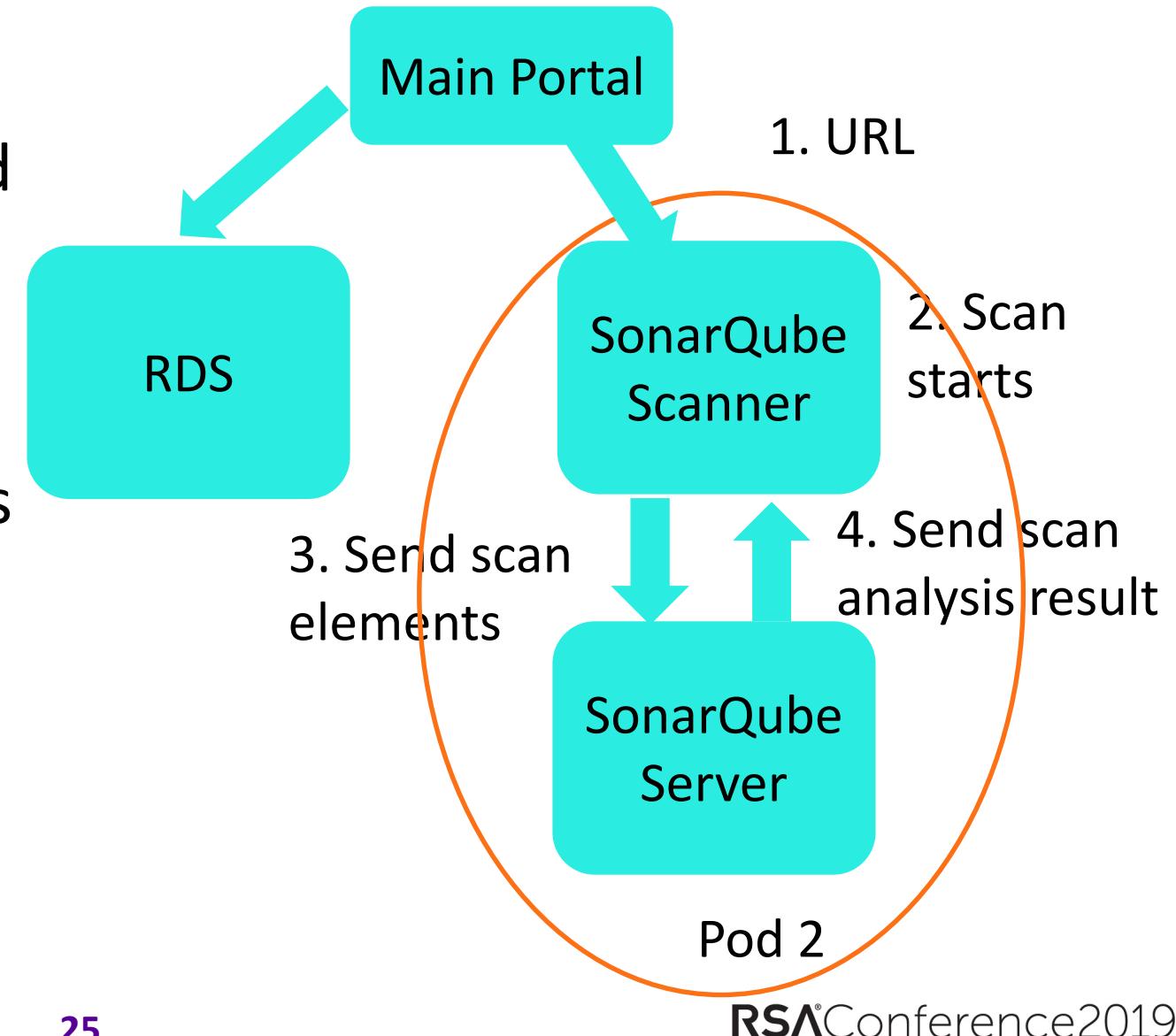


Image Scanning Service

- Qualys scan
 - Asynchronous flow
 - Launch & wait for scan result
 - Parse result to get scores
 - Store result & scores in repo
 - Track remediation



Container Scanning Service

- Black Duck

- Install Black Duck scanner on the machine where your code is placed
- Configure scan settings
- Parse result to get scores
- Store result & scores in repo
- Track remediation

The screenshot shows the Synopsys Project Dashboard. At the top, there are three bar charts: Security Risk (High: 4, Medium: 0, Low: 0, None: 0), License Risk (High: 4, Medium: 0, Low: 0, None: 0), and Operational Risk (High: 4, Medium: 0, Low: 0, None: 0). Below these charts is a table displaying four projects:

| Project Name | Tier | Last Updated | Versions | Security Risk | License Risk | Operational Risk |
|--------------|------|--------------|----------|----------------------------------|----------------------------------|----------------------------------|
| ○ | | Jan 16, 2019 | 6 | <div style="width: 100%;"></div> | <div style="width: 100%;"></div> | <div style="width: 100%;"></div> |
| ○ | | Jan 16, 2019 | 2 | <div style="width: 100%;"></div> | <div style="width: 100%;"></div> | <div style="width: 100%;"></div> |
| ○ | | Jan 7, 2019 | 2 | <div style="width: 100%;"></div> | <div style="width: 100%;"></div> | <div style="width: 100%;"></div> |
| ○ | | Jan 16, 2019 | 3 | <div style="width: 100%;"></div> | <div style="width: 100%;"></div> | <div style="width: 100%;"></div> |

Displaying 1-4 of 4

```
function get_remote_file() {
    readonly REQUEST_URL=$1
    readonly TEMP_FILE="${!THISDIR}/tmp.file"
    local command_output
    if [ -n "$(which wget)" ]; then
        execute_remote_request command_output "wget -O" "$REQUEST_URL" "$2"
        if [[ $? -eq 0 ]]; then
            tr -d "\r" <"$TEMP_FILE" >"$2"
            rm "$TEMP_FILE"
            chmod 755 "$2"
            return 0
        else
            if [[ "$command_output" == "${command_output##Resolving blackducksoftware.github.io}" ]]; then
                using_proxy_instructions
            fi
        fi
        return 1
    else
        if [ -n "$(which curl)" ]; then
            execute_remote_request command_output "curl -s --netrc-optional -I $REQUEST_URL" ""
            status_code=$(echo $command_output | cut -d '$\r' -f1 | cut -d '$' -f2)
            server_name=$(echo $command_output | cut -d '$\r' -f2 | cut -d '$' -f3)
            if [ "$status_code" == "200" ]; then
                execute_remote_request command_output "curl --netrc-optional -s -o" "$REQUEST_URL" "$2"
                if [[ $? -eq 0 ]]; then
                    tr -d "\r" <"$TEMP_FILE" >"$2"
                    rm "$TEMP_FILE"
                    chmod 755 "$2"
                    return $?
                fi
            else
                if [[ "$server_name" != "GitHub.com" ]]; then
                    using_proxy_instructions
                fi
            fi
        fi
    fi
    return 2
}
}
```

Dynamic Web Scanning Service - Zed Attack Proxy (ZAP)

- MITM proxy
 - Passive/Active scans
- Integrate with CI/CD
 - ZAP plugin for Jenkins
 - Configure plugin
 - Configure project
 - Configure session
 - Configure attack mode



RSA® Conference 2019

Forensics

Logging, Monitoring & Alerting



Logging Services

- Consolidate all logs into a central service
 - System/Auth, Cloudtrail, VPC Flow, Kubernetes Audit, DNS, Application
- Monitor heartbeat to ensure that logs are being collected
- Provide API access to logs
- Predefined views
- Long-term storage for compliance
- GDPR sensitive



Behavior Anomaly Detection And Alerting Service

- Allow operators to create security alerts on their logs
 - Reduces false positives
 - Democratizes security alerting
- Build baseline alerts for common vectors
 - Well defined vectors
 - Low false positives
- Alerts on behavior changes in credential usage
 - Change in source or target
 - Change in API called and/or API parameters



Incident Response Services (Reactive)

- Incident response process documented and distributed
- Help responders identify ownership and involve the owners during forensics
- Build services to enable responders to quickly take over effected resources
- Create a consolidated view of all relevant logs for the effected assets/service
- Review process and feedback into security maturity framework

RSA® Conference 2019

Summary

Summary

- Evolve security by democratizing it
- Create a Security Maturity Framework
- Build services to help DevOps achieve security maturity
- Decentralize security monitoring & alerting
- Integrate TVM in the SDLC
- Revisit every quarter to identify what works & what doesn't

Links

- https://d0.awsstatic.com/whitepapers/compliance/AWS_CIS_Foundations_Benchmark.pdf
- Keymaster <https://github.com/Symantec/keymaster>
- Cloudgate <https://github.com/Symantec/cloud-gate>
- Cloud Custodian <https://github.com/cloud-custodian/cloud-custodian>
- SonarQube <https://github.com/SonarSource/sonarqube>
- ZAP <https://github.com/zaproxy/zaproxy>
- Clair <https://github.com/coreos/clair>
- Smallpoint <https://github.com/Symantec/ldap-group-management>
- Blackduck <https://www.blackducksoftware.com/black-duck-home>
- Security Scans Wrappers: <https://github.com/Symantec/security-scan-wrappers>
- CWP <https://www.symantec.com/products/cloud-workload-protection>
- Vault Self Service <https://github.com/hardeep-s/vault-exchange>