Sesión 7: Seguridad, Prometheus

seguridad

- https://www.owasp.org/
- JSON
- REST = Representational state transfer
    - uso de verbos
        - comunes: get, post, head
        - otras: putm patch, delete, transfer...
        - https://en.wikipedia.org/wiki/Representational_state_transfer
        - waf, ids, ips : filtrado por reglas
- SQL
    - https://github.com/sqlmapproject
- inmutable servers
- cloud-init :
- CORS headers:
    - https://developer.mozilla.org/es/docs/Web/HTTP/Access_control_CORS
- CSRF tokens:
    - https://www.welivesecurity.com/la-es/2015/04/21/vulnerabilidad-cross-site-request-forgery-csrf/
- Certificados:
    - FreeIPA
    - Let's Encrypt
        - certbot
    - bettercap
        - beef : better explotation framework
    - SSL/TLS
        - Diffie Hellman
        - DNS CAA
        - OCSP Stappling vs CRLs
        - certificate transparency
        - https://www.geotrust.com/es/
        - https://ed25519.cr.yp.to/ ; curva elíptica
        - Heartbleed
        - ssl labs: https://www.ssllabs.com/
        - ~~RSA~~
        - ~~SSL v2 y v3~~
        - ssl → tls (nueva versión)
    - tipos de certificados
        - domain validation
        - organization validation : añade organización en el certificado
        - extended validation : escrituras, envio-recepción de documentación → servicios sensibles
    - brute force attacks
        - kpeiruza.hashtopolis : docker hack contraseñas distribuidas?
        - ~~john the ripper~~ → hashcat
        - xHydra
        - jasimia
    - DoS
    - DDoS
        - https://en.wikipedia.org/wiki/Denial-of-service_attack#Amplification

prometheus

swarm-prometheus.yml

```
version: "3.3"

networks:
  net:
    driver: "overlay"
  proxy:
    external: true

volumes:
  prometheus:
    driver_opts:
      type: "nfs"
      o: "addr=192.168.50.200,nolock,soft,rw"
      device: ":/srv/nvme/cluster3/prometheus/prometheus"
  grafana:
    driver_opts:
      type: "nfs"
      o: "addr=192.168.50.200,nolock,soft,rw"
      device: ":/srv/nvme/cluster3/prometheus/grafana"
  alertmanager:
    driver_opts:
      type: "nfs"
      o: "addr=192.168.50.200,nolock,soft,rw"
      device: ":/srv/nvme/cluster3/prometheus/alertmanager"

configs:
#  dockerd_config:
#    file: /srv/docker/prometheus/prometheus/rules/Caddyfile
  node_rules:
    file: /srv/docker/prometheus/prometheus/rules/swarm_node.rules.yml
  task_rules:
    file: /srv/docker/prometheus/prometheus/rules/swarm_task.rules.yml

services:
#  dockerd-exporter:
#    image: stefanprodan/caddy
#    networks:
#      - net
#    environment:
#      - DOCKER_GWBRIDGE_IP=172.18.0.1
#    configs:
#      - source: dockerd_config
#        target: /etc/caddy/Caddyfile
#    deploy:
#      mode: global
#      resources:
#        limits:
#          memory: 128M
#        reservations:
#          memory: 64M
```

```
  cadvisor:
    image: google/cadvisor
    networks:
      - net
    command: -logtostderr -docker_only
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock:ro
      - /:/rootfs:ro
      - /var/run:/var/run
      - /sys:/sys:ro
      - /var/lib/docker/:/var/lib/docker:ro
    deploy:
      mode: global
      resources:
        limits:
          memory: 128M
        reservations:
          memory: 64M

  grafana:
    image: stefanprodan/swarmprom-grafana:5.3.4
    networks:
      - net
    environment:
      - GF_SECURITY_ADMIN_USER=${ADMIN_USER:-admin}
      - GF_SECURITY_ADMIN_PASSWORD=${ADMIN_PASSWORD:-admin}
      - GF_USERS_ALLOW_SIGN_UP=false
      #- GF_SERVER_ROOT_URL=${GF_SERVER_ROOT_URL:-localhost}
      #- GF_SMTP_ENABLED=${GF_SMTP_ENABLED:-false}
      #- GF_SMTP_FROM_ADDRESS=${GF_SMTP_FROM_ADDRESS:-grafana@test.com}
      #- GF_SMTP_FROM_NAME=${GF_SMTP_FROM_NAME:-Grafana}
      #- GF_SMTP_HOST=${GF_SMTP_HOST:-smtp:25}
      #- GF_SMTP_USER=${GF_SMTP_USER}
      #- GF_SMTP_PASSWORD=${GF_SMTP_PASSWORD}
    volumes:
      - grafana:/var/lib/grafana
    deploy:
      mode: replicated
      replicas: 1
      placement:
        constraints:
          - node.role == manager
      resources:
        limits:
          memory: 128M
        reservations:
          memory: 64M
      labels:
        - traefik.frontend.rule=Host:grafana.amachete.local
        - traefik.port=3000
        - traefik.docker.network=proxy
    networks:
      - default
      - net
      - proxy
```

```
  alertmanager:
    image: stefanprodan/swarmprom-alertmanager:v0.14.0
    networks:
      - net
    environment:
      - SLACK_URL=${SLACK_URL:-https://hooks.slack.com/services/TOKEN}
      - SLACK_CHANNEL=${SLACK_CHANNEL:-general}
      - SLACK_USER=${SLACK_USER:-alertmanager}
    command:
      - '--config.file=/etc/alertmanager/alertmanager.yml'
      - '--storage.path=/alertmanager'
    volumes:
      - alertmanager:/alertmanager
    deploy:
      mode: replicated
      replicas: 1
      placement:
        constraints:
          - node.role == manager
      resources:
        limits:
          memory: 128M
        reservations:
          memory: 64M
      labels:
        - traefik.frontend.rule=Host:alertmanager.amachete.local
        - traefik.port=9093
        - traefik.docker.network=proxy
        - traefik.frontend.auth.basic.users=${ADMIN_USER}:${HASHED_PASSWORD}
    networks:
      - default
      - net
      - proxy

  unsee:
    image: cloudflare/unsee:v0.8.0
    networks:
      - net
    environment:
      - "ALERTMANAGER_URIS=default:http://alertmanager.amachete.local:9093"
    deploy:
      mode: replicated
      replicas: 1
      labels:
        - traefik.frontend.rule=Host:unsee.amachete.local
        - traefik.enable=true
        - traefik.port=8080
        - traefik.tags=${TRAEFIK_PUBLIC_TAG:-proxy}
        - traefik.docker.network=proxy
        # Traefik service that listens to HTTP
        - traefik.redirectorservice.frontend.entryPoints=http
        - traefik.redirectorservice.frontend.redirect.entryPoint=https
        # Traefik service that listens to HTTPS
        - traefik.webservice.frontend.entryPoints=https
        -
traefik.frontend.auth.basic.users=admin:$apr1$7zoJwzGV$jlhxzJsM7xVVsN.w5rJ.W.
```

```
    networks:
      - default
      - net
      - proxy

  node-exporter:
    image: stefanprodan/swarmprom-node-exporter:v0.16.0
    networks:
      - net
    environment:
      - NODE_ID={{.Node.ID}}
    volumes:
      - /proc:/host/proc:ro
      - /sys:/host/sys:ro
      - /:/rootfs:ro
      - /etc/hostname:/etc/nodename
    command:
      - '--path.sysfs=/host/sys'
      - '--path.procfs=/host/proc'
      - '--collector.textfile.directory=/etc/node-exporter/'
      - '--collector.filesystem.ignored-mount-
points=^/(sys|proc|dev|host|etc)($$|/)'
      - '--no-collector.ipvs'
    deploy:
      mode: global
      resources:
        limits:
          memory: 128M
        reservations:
          memory: 64M

  prometheus:
    image: stefanprodan/swarmprom-prometheus:v2.5.0
    networks:
      - net
    command:
      - '--config.file=/etc/prometheus/prometheus.yml'
      - '--storage.tsdb.path=/prometheus'
      - '--storage.tsdb.retention=24h'
    volumes:
      - prometheus:/prometheus
    configs:
      - source: node_rules
        target: /etc/prometheus/swarm_node.rules.yml
      - source: task_rules
        target: /etc/prometheus/swarm_task.rules.yml
    deploy:
      mode: replicated
      replicas: 1
      placement:
        constraints:
          - node.role == manager
      resources:
        limits:
          memory: 2048M
        reservations:
```

```
          memory: 128M
      labels:
        - traefik.frontend.rule=Host:prometheus.amachete.local
        # - traefik.enable=true
        - traefik.port=9090
        - traefik.tags=traefik-public
        - traefik.docker.network=proxy
        # Traefik service that listens to HTTP
        # - traefik.redirectorservice.frontend.entryPoints=http
        # - traefik.redirectorservice.frontend.redirect.entryPoint=https
        # - Traefik service that listens to HTTPS
        # - traefik.webservice.frontend.entryPoints=https
        # -
traefik.frontend.auth.basic.users=admin:$apr1$7zoJwzGV$jlhxzJsM7xVVsN.w5rJ.W.

    networks:
      - default
      - net
      - proxy
```

swarm_node.rules.yaml

```
groups:
- name: /1/store/projects/vagrant/docker-swarm-
vagrant/apps/swarmprom/prometheus/rules/swarm_node.rules.yml
  rules:
  - alert: node_cpu_usage
    expr: 100 - (avg(irate(node_cpu_seconds_total{mode="idle"}[1m]) *
ON(instance) GROUP_LEFT(node_name)
      node_meta * 100) BY (node_name)) > 50
    for: 1m
    labels:
      severity: warning
    annotations:
      description: Swarm node {{ $labels.node_name }} CPU usage is at {{
humanize
        $value}}%.
      summary: CPU alert for Swarm node '{{ $labels.node_name }}'
  - alert: node_memory_usage
    expr: sum(((node_memory_MemTotal_bytes - node_memory_MemAvailable_bytes) /
node_memory_MemTotal_bytes)
      * ON(instance) GROUP_LEFT(node_name) node_meta * 100) BY (node_name) >
80
    for: 1m
    labels:
      severity: warning
    annotations:
      description: Swarm node {{ $labels.node_name }} memory usage is at {{
humanize
        $value}}%.
      summary: Memory alert for Swarm node '{{ $labels.node_name }}'
  - alert: node_disk_usage
    expr: ((node_filesystem_size_bytes{mountpoint="/rootfs"} -
node_filesystem_free_bytes{mountpoint="/rootfs"})
      * 100 / node_filesystem_size_bytes{mountpoint="/rootfs"}) * ON(instance)
```

```
GROUP_LEFT(node_name)
      node_meta > 85
    for: 1m
    labels:
      severity: warning
    annotations:
      description: Swarm node {{ $labels.node_name }} disk usage is at {{
humanize
        $value}}%.
      summary: Disk alert for Swarm node '{{ $labels.node_name }}'
  - alert: node_disk_fill_rate_6h
    expr: predict_linear(node_filesystem_free_bytes{mountpoint="/rootfs"}[1h],
6 * 3600) * ON(instance)
      GROUP_LEFT(node_name) node_meta < 0
    for: 1h
    labels:
      severity: critical
    annotations:
      description: Swarm node {{ $labels.node_name }} disk is going to fill up
in
        6h.
      summary: Disk fill alert for Swarm node '{{ $labels.node_name }}'
```

swarm_tasks.rules.yaml

```
groups:
- name: /1/store/projects/vagrant/docker-swarm-
vagrant/apps/swarmprom/prometheus/rules/swarm_task.rules.yml
  rules:
  - alert: task_high_cpu_usage_50
    expr:
sum(rate(container_cpu_usage_seconds_total{container_label_com_docker_swarm_ta
sk_name=~".+"}[1m]))
      BY (container_label_com_docker_swarm_task_name,
container_label_com_docker_swarm_node_id)
      * 100 > 50
    for: 1m
    annotations:
      description: '{{ $labels.container_label_com_docker_swarm_task_name }}
on ''{{
        $labels.container_label_com_docker_swarm_node_id }}'' CPU usage is at
{{ humanize
        $value}}%.'
      summary: CPU alert for Swarm task '{{
$labels.container_label_com_docker_swarm_task_name
        }}' on '{{ $labels.container_label_com_docker_swarm_node_id }}'
  - alert: task_high_memory_usage_1g
    expr:
sum(container_memory_rss{container_label_com_docker_swarm_task_name=~".+"})
      BY (container_label_com_docker_swarm_task_name,
container_label_com_docker_swarm_node_id) > 1e+09
    for: 1m
    annotations:
      description: '{{ $labels.container_label_com_docker_swarm_task_name }}
on ''{{
```

```
        $labels.container_label_com_docker_swarm_node_id }}'' memory usage is
{{ humanize
        $value}}.'
      summary: Memory alert for Swarm task '{{
$labels.container_label_com_docker_swarm_task_name
        }}' on '{{ $labels.container_label_com_docker_swarm_node_id }}'
```

otros

- http://fediafedia.com/neo/scp/
- http://geektyper.com/
- Hadoop Ecosystem Table: https://hadoopecosystemtable.github.io/
- radare - https://rada.re/r/
- `sudo showmount –export 192.168.50.200`