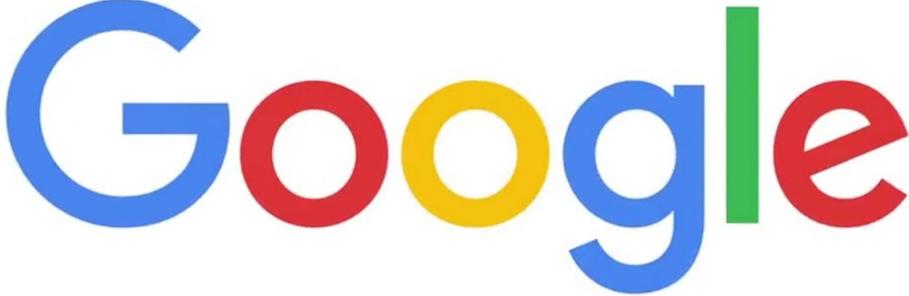


컨테이너 버그 찾기 어디까지 해봤니?

How To Find Container Platform Bug?

Team. MobyDick

The Google logo is displayed in its characteristic multi-colored font: 'G' in blue, 'o' in red, 'o' in yellow, 'g' in blue, 'l' in green, and 'e' in red.

구글의 모든 제품은 컨테이너에서 실행

→ 매주 수십억 개의 컨테이너 생성

80%

IT 조직 80% 이상이 컨테이너 사용 (직원 500명 이상)

Container

일관성 있는 환경

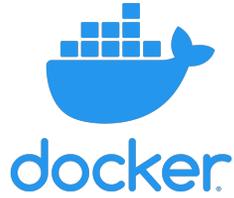
배포의 편의성

작은 이미지 크기

다양한 운영 환경 지원

“가트너, 컨테이너 보안을 올해 가장 우려되는 10가지 요소 중 하나로 선정”

“트렌드 마이크로, 기업의 86%이상이 쿠버네티스를 사용하며 컨테이너 보안에 대한 적절한 투자를 보유하고 있지 않아 쿠버네티스 채택이 빠르게 지연되거나 전략 없이 심각한 보안 사고가 발생”

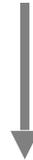


Docker: 컨테이너 기반 가상화 플랫폼



kubernetes

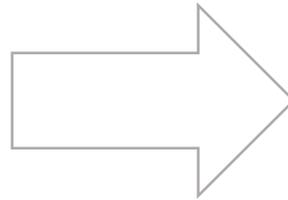
Kubernetes: 컨테이너 오케스트레이션 도구



IT 조직 50% 이상이 Docker 사용, Kubernetes가 시장의 80% 이상 점유

컨테이너 버그 찾기 – Logic Bug?

- CVE-2014-9357 - **wrong namespace**
- CVE-2019-14271 - **wrong namespace**
- CVE-2018-1002100 - **wrong namespace**
- CVE-2019-1002101 - **wrong namespace**
- CVE-2016-9962 - **File Descriptor**
- CVE-2019-5736 - **File Descriptor**
- CVE-2018-15664 - **Race Condition**
- CVE-2019-19921 - **Race Condition**



<p>CVE-2019-1002101 일일작성로복합</p> <p>cve_2019_1002101.pptx</p>	<p>CVE-2019-14271 www.kor.ac.kr</p> <p>CVE_2019_14271_김우석...</p>	<p>CVE-2016-1906 오류시도해결 지양적 주요해결사항만 단순 버그</p> <p>201104 CVE-2016-1906 ...</p>
<p>CVE-2019-13059 Info Leak</p> <p>201019 CVE-2019-1350...</p>	<p>CVE-2018-20699 VERY SIMPLE DoS</p> <p>201019 CVE-2018-2069...</p>	<p>CVE-2019-13139</p> <p>201009_CVE-2019-1313...</p>
<p>RunC 연대기 [CVE-2016-9962, CVE-2019-5736, CVE-2018-15664, CVE-2019-19921]</p> <p>201026 runC 연대기.pptx</p>	<p>CVE-2019-11248 쿠버네티스 Info Leak</p> <p>200921 CVE-2019-1124...</p>	<p>CVE-2020-8559 (Kubernetes) 손상된 노드를 통한 클러스터 전체 공격</p> <p>보통, 이런 것도 CVE일까?</p> <p>201109 CVE-2020-8559 ...</p>

컨테이너 버그 찾기 – Logic Bug?



CVE-2014-9357 - logic (Remote Code Execution)

CVE-2018-15514 - logic (Privilege Escalation)

CVE-2019-5736 - logic (Remote Code Execution)

CVE-2019-14271 - logic (Container Escape)

CVE-2019-15752 - logic (Privilege Escalation)



CVE-2016-1906 - logic (Privilege Escalation)

CVE-2018-1002101 - command injection (DoS)

CVE-2018-1002105 - logic (Privilege Escalation)

CVE-2019-11247 - logic (Arbitrary Read/Write)

CVE-2019-11248 - logic (DoS + Leak)

The underlying technology

Docker

Docker is written in the [Go programming language](#) and takes advantage of several features of the Linux kernel to deliver its functionality. Docker uses a technology called [namespaces](#) to provide the isolated workspace called the *container*. When you run a container, Docker creates a set of *namespaces* for that container.

These namespaces provide a layer of isolation. Each aspect of a container runs in a separate namespace and its access is limited to that namespace.

Kubernetes

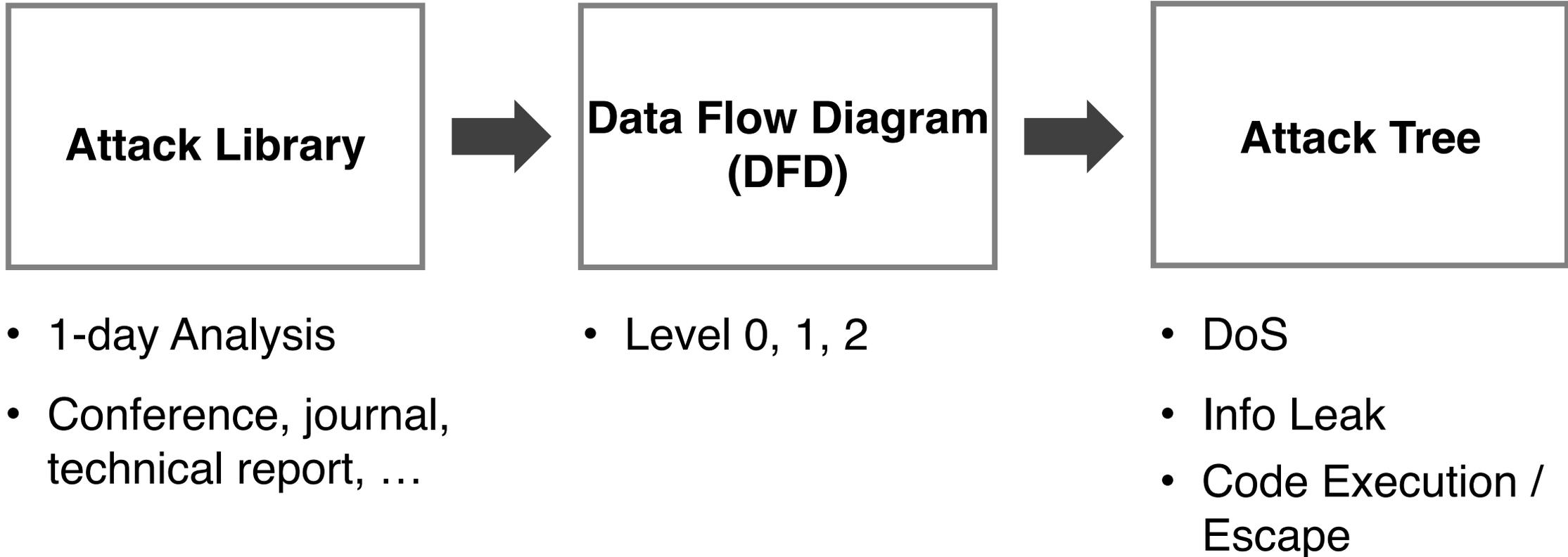
The Kubernetes project is written in the [Go programming language](#) and you can browse its [source code](#) on GitHub.

Go Language

**Threat
Modeling**

**CodeQL For GO
(Code Auditing)**

**Introspection
Tool**

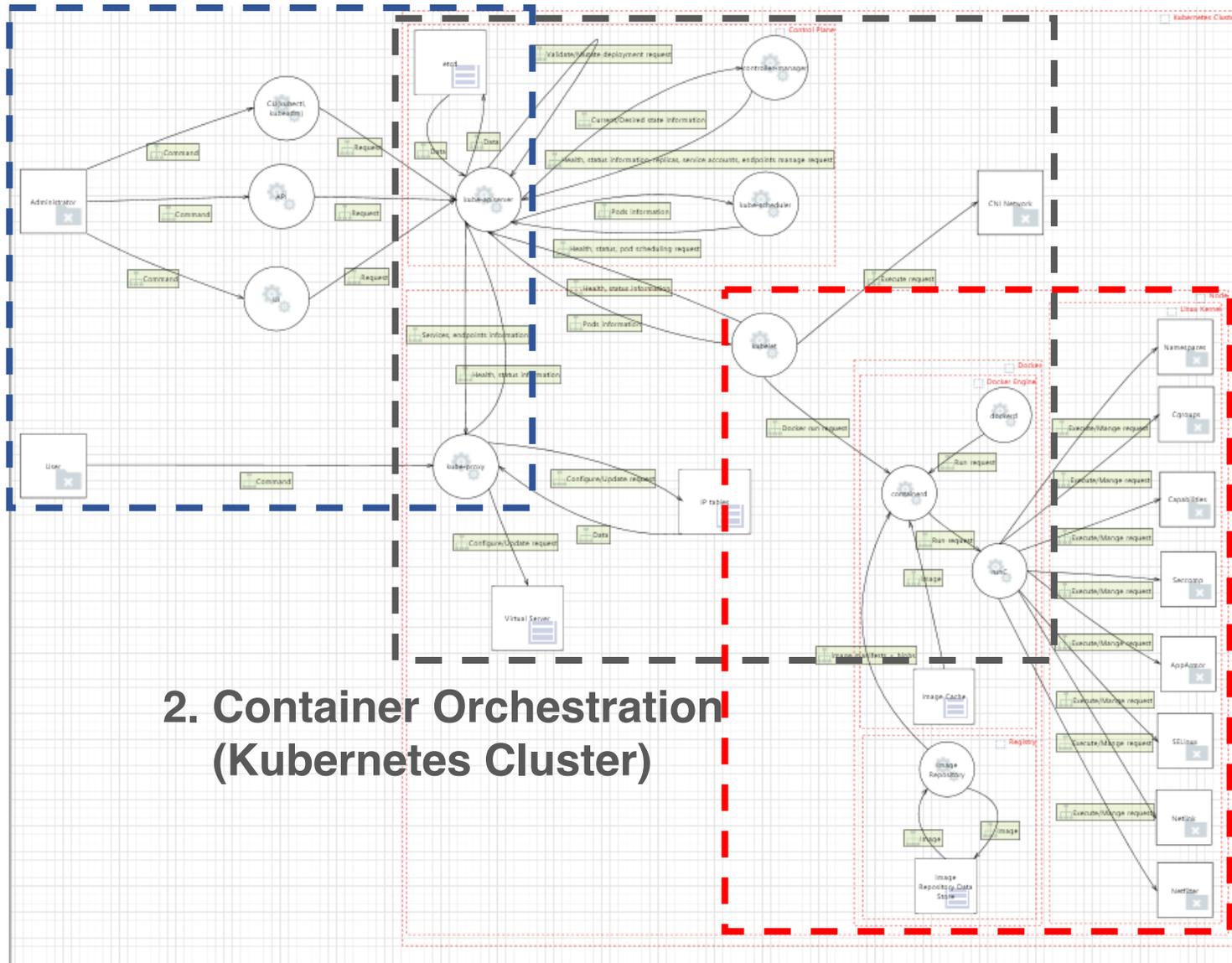


컨테이너 버그 찾기 – Threat Modeling

Category	Title	Year	Author	File	Link
Conference	Escaping Virtualized Containers	2020	Yuval Avrahami	us-20-A...	https://www.blackhat.com/us-20/briefings/schedule/#escaping-virtualized-containers-20514
Conference	A Compendium of Container Escapes	2019	Brandon Edwards, Nick Freeman	us-19-E...	https://www.blackhat.com/us-19/briefings/schedule/#a-compendium-of-container-escapes-16091
Conference	Docker Escape Technology	2016	Shengping Wang	CSW20...	https://cansecwest.com/slides/2016/CSW2016_Wang_DockerEscapeTechnology.pdf
Conference	Abusing Privileged and Unprivileged Linux Containers	2016	Jesse Hertz	contain...	https://www.nccgroup.com/uk/our-research/abusing-privileged-and-unprivileged-linux-containers/
Conference	VULNERABILITY EXPLOITATION IN DOCKER CONTAINER ENVIRONMENTS	2015	Anthony Bettini	eu-15-B...	https://www.blackhat.com/eu-15/briefings.html#vulnerability-exploitation-in-docker-container-environments
Journal	Vulnerability Analysis and Security Research of Docker Container	2020	Jiang Wenhao, Li Zheng	092368...	https://ieeexplore.ieee.org/abstract/document/9236837
Journal	Analysis of Docker Security	2015	Thanh Bui	1501.02...	https://arxiv.org/abs/1501.02967
Journal	To Docker or Not to Docker: A Security Perspective	2016	Theo Combe, Antony Martin, Roberto Di Pietro	077422...	https://ieeexplore.ieee.org/abstract/document/7742298
Journal	Docker container security via heuristics-based multilateral security-conceptual and pragmatic study	2016	A R Manu, Jitendra Kumar Patel, Shakil Akhtar, V K Agrawal, K N Bala Subramanya Murthy	075302...	https://ieeexplore.ieee.org/abstract/document/7530217
Journal	Docker ecosystem – Vulnerability Analysis	2018	A. Martin, S. Raponib, T. Combea, R. Di Pietro	1-s2.0-...	https://www.sciencedirect.com/science/article/abs/pii/S0140366417300956
Journal	Securing Docker Containers from Denial of Service (DoS) Attacks	2016	Jeeva Chelladhurai, Pethuru Raj Chelliah, Sathish Alampalayam Kumar	075575...	https://ieeexplore.ieee.org/abstract/document/7557545
Technical Report	Guide to Container Security – Everything You Need to Know	2019	Michelle Moore		https://www.tripwire.com/state-of-security/devops/guide-container-security/
White Paper	Docker security	2020	Docker, Inc.		https://docs.docker.com/engine/security/
White Paper	LXC security	2020	Canonical Ltd.		https://linuxcontainers.org/lxc/security/
White Paper	Kubernetes security	2020	Cloud Native Computing Foundation		https://kubernetes.io/docs/concepts/security/
Public Program or Project	Kubernetes Security – Best Practice Guide	2018	Simon Pirschel		https://github.com/freache/kubernetes-security-best-practice

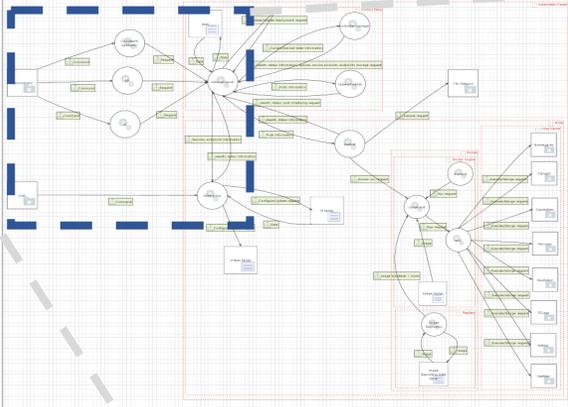
컨테이너 버그 찾기 – Threat Modeling

1. User Interaction

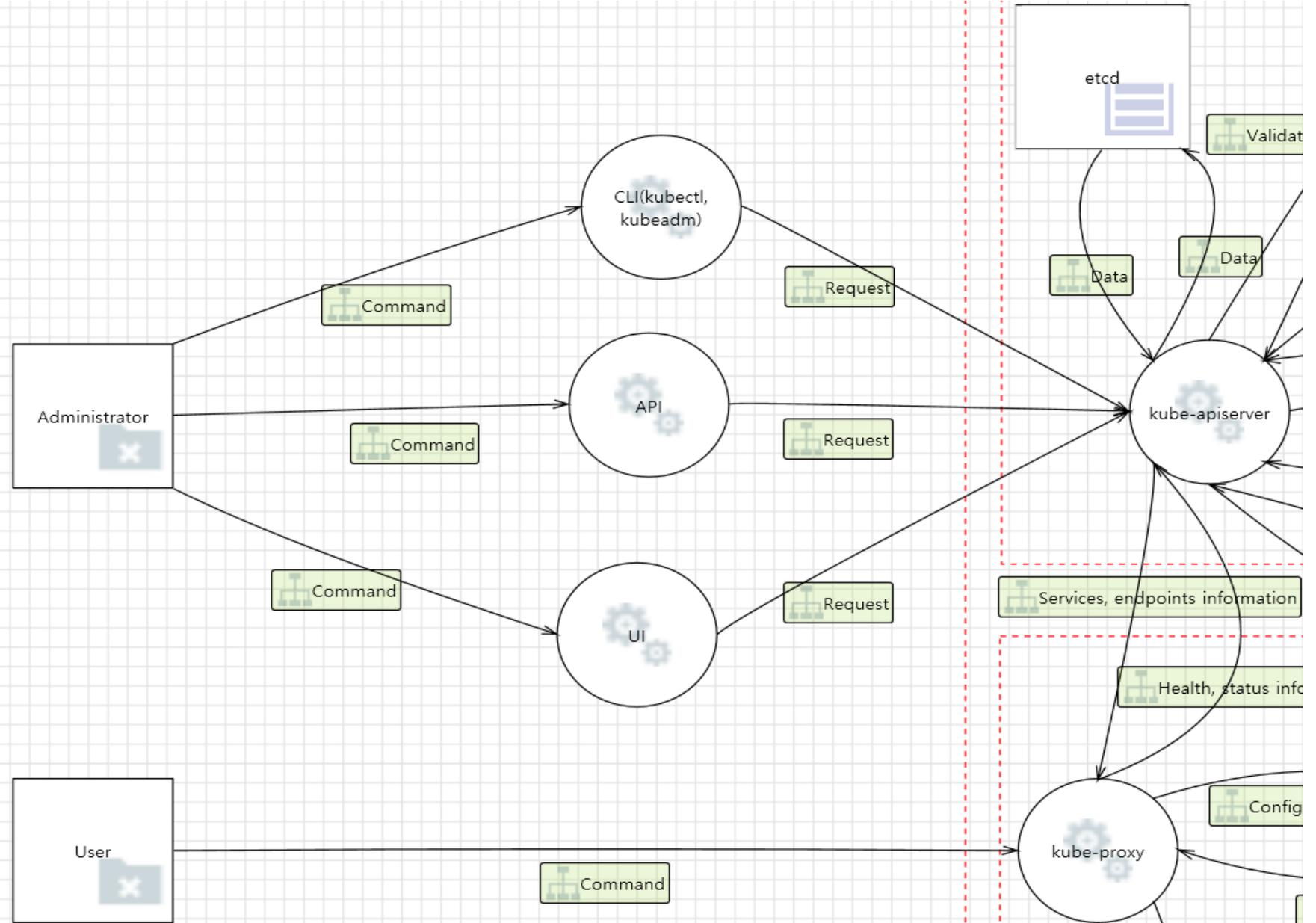


3. Container Runtime (Docker & Kernel)

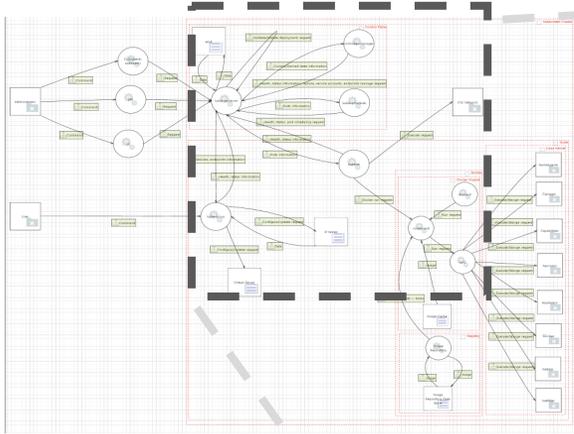
컨테이너 버그 찾기 – Threat Modeling



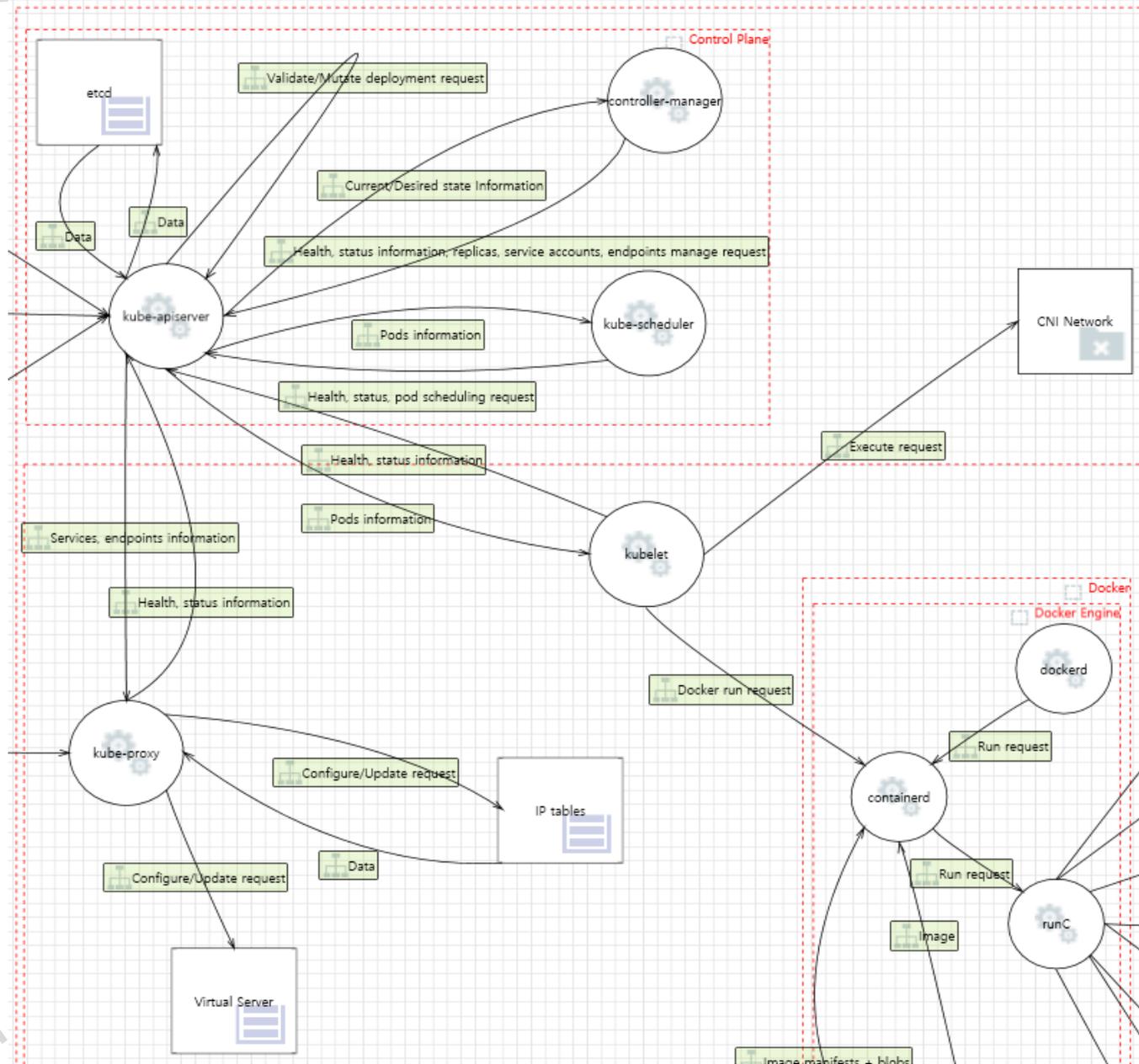
1. User Interaction



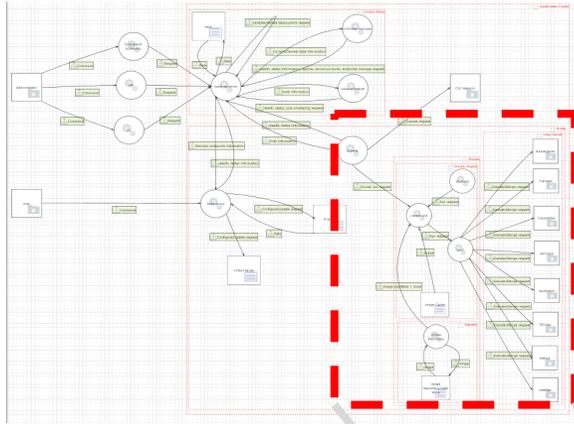
컨테이너 버그 찾기 – Threat Modeling



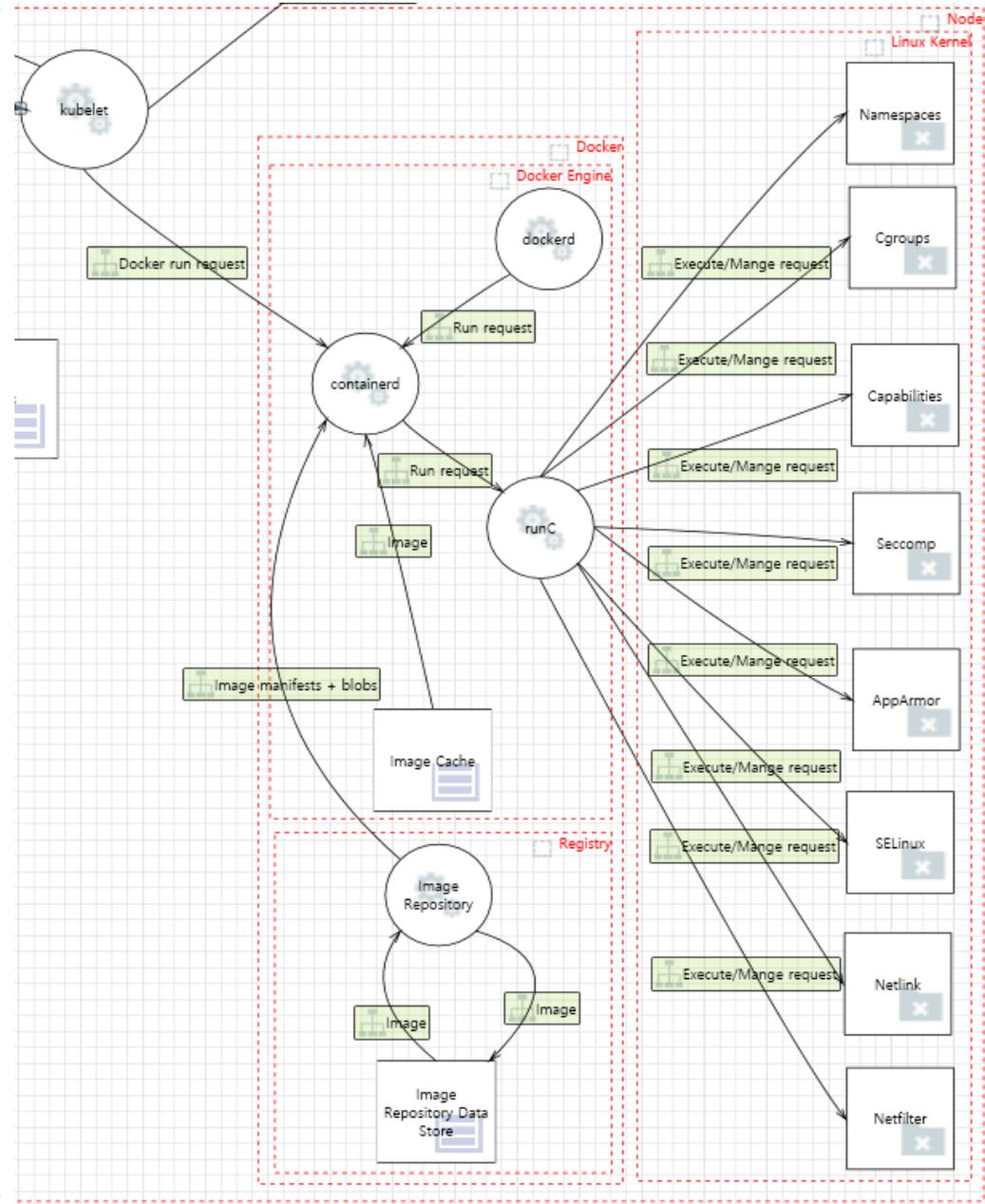
2. Container Orchestration (Kubernetes Cluster)



컨테이너 버그 찾기 – Threat Modeling

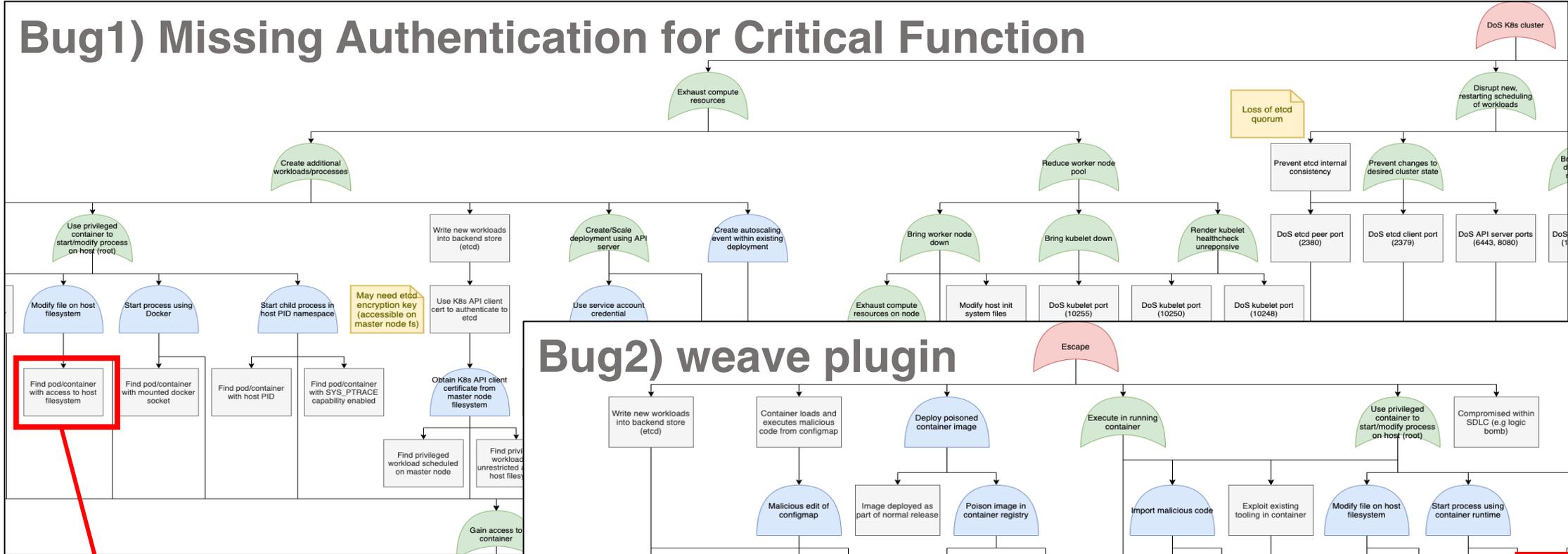


3. Container Runtime (Docker & Kernel)



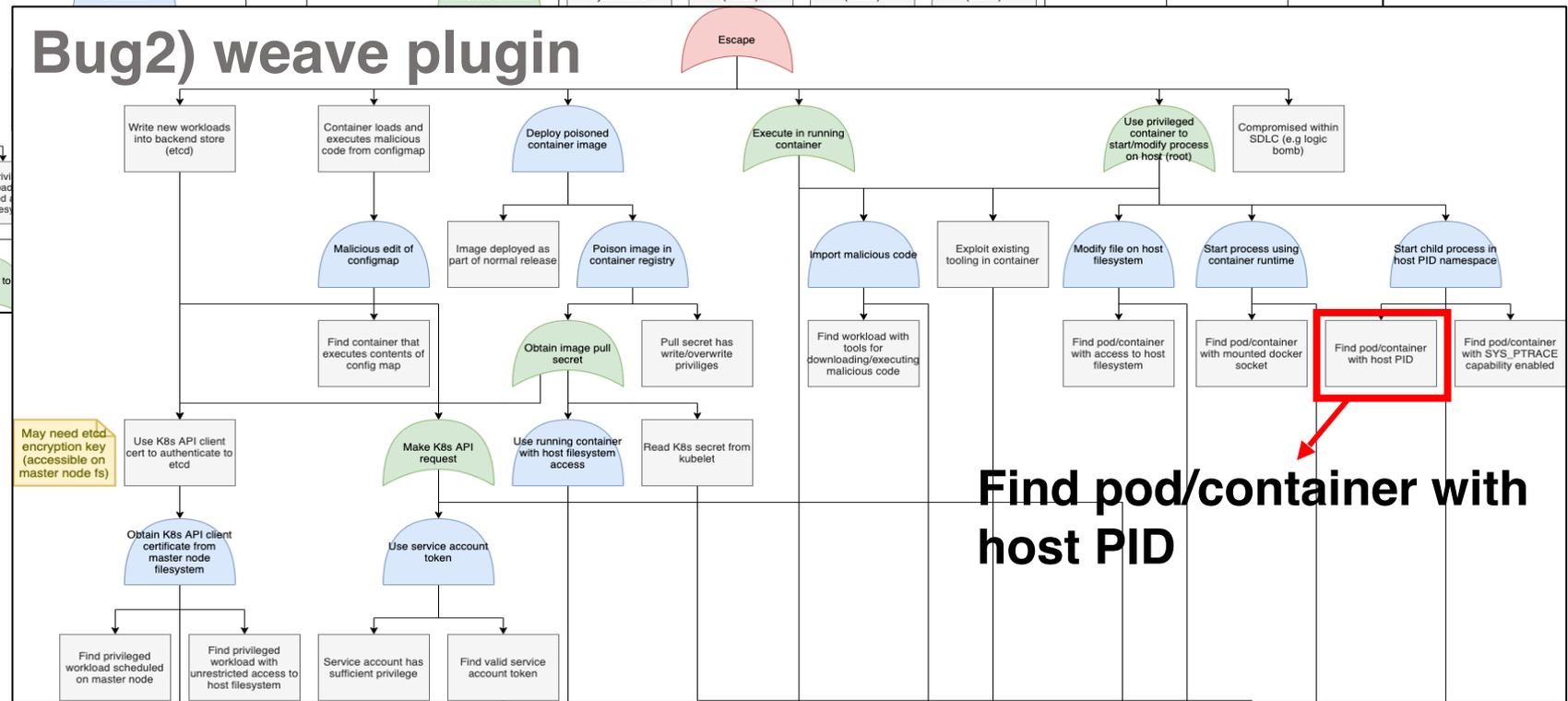
컨테이너 버그 찾기 – Threat Modeling

Bug1) Missing Authentication for Critical Function



Find pod/container with access to host filesystem

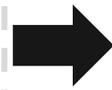
Bug2) weave plugin



Find pod/container with host PID

- 컨테이너에 **잘못된 권한 설정**
 - Root / Non-Root
- Remote Vulnerability
- **CVSS Score: 9.8**

One of the vulnerabilities we found



CVE-2020-29389 Detail

Current Description

The official Crux Linux Docker images 3.0 through 3.4 contain a blank password for a root user. System using the Crux Linux Docker container deployed by affected versions of the Docker image may allow an attacker to achieve root access with a blank password.

[+View Analysis Description](#)

Severity

CVSS Version 3.x

CVSS Version 2.0

CVSS 3.x Severity and Metrics:



NIST: NVD

Base Score:

9.8 CRITICAL

Vector:

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

NVD Analysts use publicly available information to associate vector strings and CVSS scores. We also display any CVSS information provided within the CVE List from the CNA.

Note: NVD Analysts have published a CVSS score for this CVE based on publicly available information at the time of analysis. The CNA has not provided a score within the CVE List.

- CodeQL
 - 코드 DB화, 쿼리로 원하는 패턴 검색
 - Java, C++, C#, Python, Go 등 다양한 언어 지원
- CodeQL for 'Go'
 - **2019.12 first release**
 - 다른 언어보다 지원 문법↓, 레퍼런스↓

1-day CodeQL

CodeQL 분석

```
cmd/kubelet/app/server.go
@@ -746,9 +746,10 @@ func run(s *options.KubeletServer, kubeDeps *kubelet.Dependencies, stopCh <-chan
746 746     }
747 747 }
748 748     if s.HealthzPort > 0 {
749 -         healthz.DefaultHealthz()
749 +         mux := http.NewServeMux()
750 +         healthz.InstallHandler(mux)
750 751     go wait.Until(func() {
751 -         err := http.ListenAndServe(net.JoinHostPort(s.HealthzBindAddress, strconv.Itoa(int(s.HealthzPort))), nil)
752 +         err := http.ListenAndServe(net.JoinHostPort(s.HealthzBindAddress, strconv.Itoa(int(s.HealthzPort))), mux)
752 753     if err != nil {
753 754         glog.Errorf("Starting health server failed: %v", err)
754 755     }

```

```
import go
from Function listenAndServe, DataFlow::CallNode call, DataFlow::Node argument, DataFlow::Node nil
where
listenAndServe.hasQualifiedName("net/http", "ListenAndServe") and
call = listenAndServe.getACall() and
argument = call.getArgument(1) and
nil = Builtin::nil().getARead() and
argument = nil
select call, call.getFile()
```

1-day 바탕으로 CodeQL query 작성

- **zip-slip.q1**
→ Kubernetes CVE-2019-1002100
- **bad-coding-style.q1**
→ Kubernetes CVE-2019-11248
- **xpath-injection.q1**
→ Docker 19.03.13

컨테이너 버그 찾기 – CodeQL For Go (Code Auditing)

1-day CodeQL

```
**
 * @name Unreachable statement
 * @description Unreachable statements are often indicative of missing code or latent bugs
 *               and should be avoided.
 * @kind problem
 * @problem.severity warning
 * @id go/unreachable-statement
 * @tags maintainability
 *       correctness
 *       external/cwe/cwe-561
 * @precision very-high
 */

import go

ControlFlow::Node nonGuardPredecessor(ControlFlow::Node nd) {
  exists(ControlFlow::Node pred | pred = nd.getAPredecessor() |
    if pred instanceof ControlFlow::ConditionGuardNode
      then result = nonGuardPredecessor(pred)
      else result = pred
  )
}
```

CodeQL 분석

CodeQL Query Results ×

« 1 / 1 »

#select ▾ 12 results

#	call	[1]
1	call to ListenAndServe	/opt/src/cluster/images/etcd-version-monitor/etcd-version-monitor.go
2	call to ListenAndServe	/opt/src/test/images/agnhost/audit-proxy/main.go
3	call to ListenAndServe	/opt/src/test/images/agnhost/fakegitserver/gitserver.go
4	call to ListenAndServe	/opt/src/test/images/agnhost/guestbook/guestbook.go
5	call to ListenAndServe	/opt/src/test/images/agnhost/liveness/server.go
6	call to ListenAndServe	/opt/src/test/images/agnhost/net/main.go
7	call to ListenAndServe	/opt/src/test/images/agnhost/netexec/netexec.go
8	call to ListenAndServe	/opt/src/test/images/agnhost/nettest/nettest.go
9	call to ListenAndServe	/opt/src/test/images/agnhost/no-snat-test/main.go
10	call to ListenAndServe	/opt/src/test/images/agnhost/no-snat-test-proxy/main.go
11	call to ListenAndServe	/opt/src/test/images/agnhost/serve-hostname/serve_hostname.go
12	call to ListenAndServe	/opt/src/test/images/agnhost/test-webserver/test-webserver.go

Query를 최신 Docker, Kubernetes 버전에 적용

컨테이너 버그 찾기 – CodeQL For Go (Code Auditing)

CodeQL for Go & Docker 소스코드 개선

CodeQL for Go Github에 **issue** 작성

Feature request: specify Go version used to build project #351

 Open donghyunlee00 opened this issue 19 days ago · 4 comments

 donghyunlee00 commented 19 days ago

Hello.

I tried to create a CodeQL database for kubernetes-1.9.0 (which is an old version) through LGTM.com.

It was built successfully, but I found a warning in the Golang 'Extraction' log, as follows.

```
...
[build] Detected go version: go version go1.15 linux/amd64.
[build] Kubernetes requires go1.9.1 or greater.
[build] Please install go1.9.1 or later.
...
```

As far as I know, I can specify the version through `lgtm.yml` for languages such as Python, C#, Java.

However, in [this document](#), I could not find any words to specify the version of Golang.

Is there any way to specify the Golang version when creating a CodeQL database?

Docker Github에 **pull request** 작성

Update array length check logic for preventing off-by-one error #41567

 Open J-jaeyoung wants to merge 1 commit into `moby:master` from `J-jaeyoung:fix_off_by_one`

 Conversation 1  Commits 1  Checks 1  Files changed 1



J-jaeyoung commented on 19 Oct • edited

Array length should be greater than or equal to 5, when accessing index 4

- What I did

Prevent off-by-one error.

Though `/proc/[pid]/mountinfo` has somewhat fixed format, this change will decrease possibility of runtime panic.

- How I did it

Update array length verification logic.

- How to verify it

```
package main

import (
    "fmt"
    "strings"
)

func main() {
```

Reviewers

 thajeztah

Assignees

No one assigned

Labels

`kind/bugfix` `status/2-code-review`

Projects

None yet

Milestone

20.10.0

컨테이너 버그 찾기 – CodeQL For Go (Code Auditing)

CodeQL Action

The image shows two overlapping screenshots of a GitHub repository interface. The background screenshot displays the 'All workflows' page for the repository 'MobyDick-CodeQLDB / moby'. It lists several 'Code Scanning - Action' workflows, with the most recent one showing a commit pushed by 'dopahyun00'. The foreground screenshot shows the 'Code scanning' page for the same repository, which has 7 security alerts. A sidebar on the left of the foreground screenshot shows the navigation menu with 'CodeQL' selected, indicating 7 alerts. The main content area lists the following alerts:

- Clear-text logging of sensitive information** (Test) testutil/registry/registry.go#L104 • Detected 22 hours ago
- Uncontrolled data used in path expression** daemon/checkpoint.go#L29 • Detected 22 hours ago
- Uncontrolled data used in path expression** integration/plugin/logging/cmd/close_on_start/main.go#L29 • Detected 22 hours ago
- Uncontrolled data used in path expression** integration/plugin/logging/cmd/discard/driver.go#L41 • Detected 22 hours ago
- Incorrect conversion between integer types** libcontainerd/supervisor/remote_daemon.go#L144 • Detected 22 hours ago
- Incorrect conversion between integer types** libcontainerd/supervisor/remote_daemon.go#L145 • Detected 22 hours ago
- Size computation for allocation may overflow** distribution/metadata/v2_metadata_service.go#L171 • Detected 22 hours ago

컨테이너 버그 찾기 – Introspection Tool

“ 도커 사용 시 **호스트**와 **컨테이너**가 분리,
모니터링하는 입장에서는 **호스트**와 **컨테이너**를 구분할 수 있어야 함 “

ProcMon (Linux)

```

b>> ProcessMonitor (preview) <<<
Start Time: 22:57:05 Total Events: 89300
Timestamp: PID: Process: Operation: Result: Duration (ms): Details:
+0:0:4.241 3428 containerd-shim epoll_pwait 1 2.847157 epfd=5 events=0xc00013f8a0 naxevents=128 timeout=4294
+0:0:4.241 3419 containerd-shim nanosleep 0 2.672636 rqtg=0xc00004bf30 rntp=NULL
+0:0:4.241 3502 yes write 8192 10.409498 fd=1 buf=[\n] count=8192
+0:0:4.244 3491 containerd-shim futex 1 0.007647 uaddr=0xc00003c048 op=129 val=1 utime=NULL uaddr2=NU
+0:0:4.244 3491 containerd-shim epoll_wait 1 0.155886 epfd=9 events=0xc00007a9b8 naxevents=128 timeout=-1
+0:0:4.244 3419 containerd-shim nanosleep 0 0.159399 rqtg=0xc00004bf30 rntp=NULL
+0:0:4.244 3421 containerd-shim read 4095 0.049369 fd=14 buf={\n} count=97081824765606221
+0:0:4.244 3421 cont: y count=4095
+0:0:4.244 1442 dock: Event Properties a8 naxevents=128 timeout=42
+0:0:4.244 3491 cont: Timestamp: +0:0:4.241 naxevents=128 timeout=-1
+0:0:4.244 3419 cont: PID: 3502 naxevents=128 timeout=-1
+0:0:4.244 3428 cont: Process: yes naxevents=128 timeout=4294
+0:0:4.244 3421 cont: Command Line: yes 1824765607097
+0:0:4.244 3421 cont: Syscall: write y count=4095
+0:0:4.244 3421 cont: Arguments: fd=1 buf=y val=0 utime=NULL uaddr2=NU
+0:0:4.244 3491 cont: Result: 8192 naxevents=128 timeout=-1
+0:0:4.244 3419 cont: Duration: 10494490 ns
+0:0:4.244 3491 cont: Stack Trace:
+0:0:4.244 3491 cont: 0xA1C951E7 /usr/lib/x86_64-linux-gnu/libc-2.31.so[UNKNOWN]
+0:0:4.244 3419 cont: 0xA790A79 [UNKNOWN]
+0:0:4.244 3491 cont:
+0:0:4.244 3419 cont:
+0:0:4.245 3491 cont: count=8192
+0:0:4.245 1374 dock: stre count=74
+0:0:4.245 3419 cont:
+0:0:4.245 3491 cont: naxevents=128 timeout=-1
+0:0:4.245 3419 containerd-shim nanosleep 0 0.165750 rqtg=0xc00004bf30 rntp=NULL
+0:0:4.245 1361 dockerd futex 1 0.006487 uaddr=0xc0007c5d48 op=129 val=1 utime=NULL uaddr2=NU
+0:0:4.245 1361 dockerd nanosleep 0 0.428877 rqtg=0x7efdd086c0d38 rntp=NULL
+0:0:4.245 3423 dockerd futex 0 17.102144 uaddr=0xc0007c5d48 op=128 val=0 utime=NULL uaddr2=NU
+0:0:4.245 1374 dockerd write 74 0.013612 fd=25 buf=["log":"\n","stre count=74
+0:0:4.245 3419 containerd-shim nanosleep 0 0.156065 rqtg=0xc00004bf30 rntp=NULL
+0:0:4.245 3491 containerd-shim epoll_wait 1 0.098481 epfd=9 events=0xc00007a9b8 naxevents=128 timeout=-1
+0:0:4.245 3419 containerd-shim nanosleep 0 0.135434 rqtg=0xc00004bf30 rntp=NULL
+0:0:4.245 3491 containerd-shim epoll_wait 1 0.218161 epfd=9 events=0xc00007a9b8 naxevents=128 timeout=-1
+0:0:4.245 3419 containerd-shim nanosleep 0 0.099400 rqtg=0xc00004bf30 rntp=NULL
+0:0:4.245 1361 dockerd nanosleep 0 0.250084 rqtg=0x7efdd086c0d38 rntp=NULL

```

Tracee

```

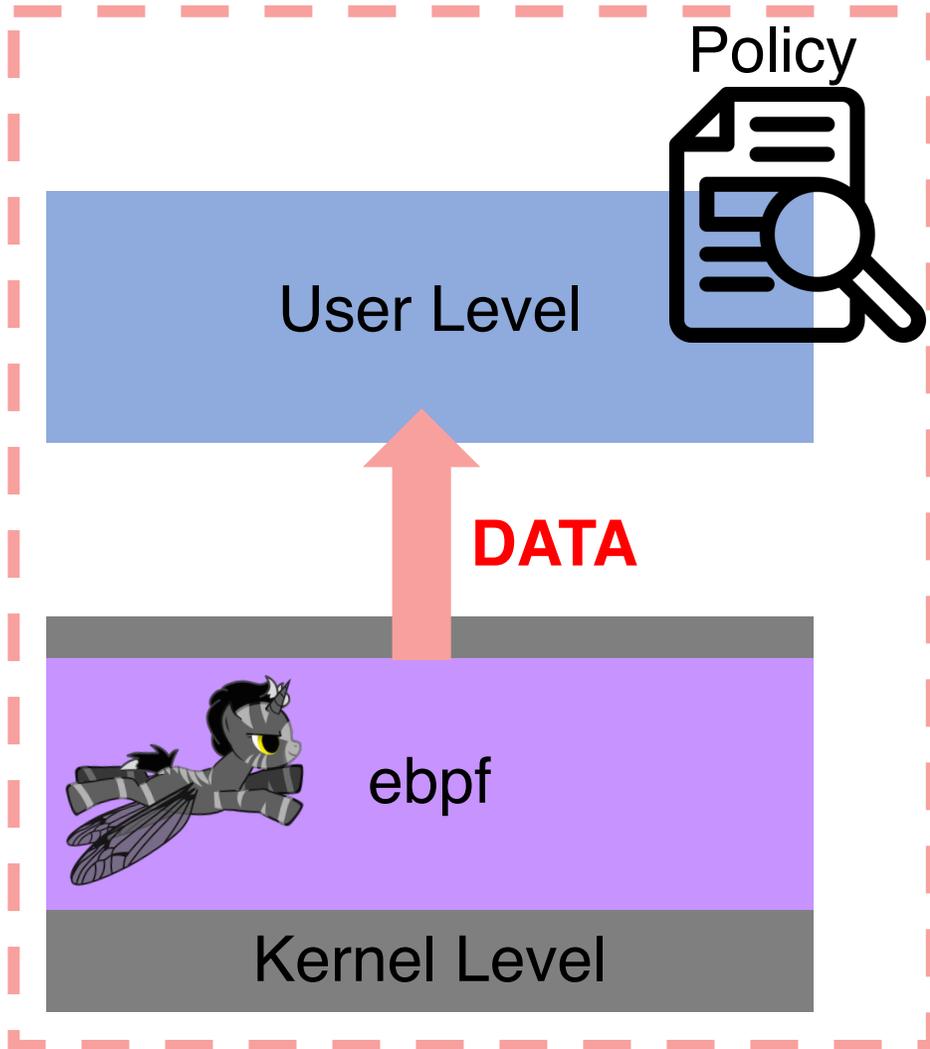
root@ubuntu:/home/ubun20/bcc# tracee --output table-verbose
TIME(s) UTS_NAME MNT_NS PID_NS UID COMM_NAME PID TID PPID RET
158.725106 ubuntu 4026531840 4026531836 1000 bash 2712 2712 2382 0
158.733123 ubuntu 4026531840 4026531836 1000 bash 2712 2712 2382 0
158.737106 ubuntu 4026531840 4026531836 1000 docker 2712 2712 2382 -2
158.738352 ubuntu 4026531840 4026531836 1000 docker 2712 2712 2382 0
v: 8388613, lnode: 922856
158.738621 ubuntu 4026531840 4026531836 1000 docker 2712 2712 2382 3
CLOEXEC, mode: 3822718664
158.738690 ubuntu 4026531840 4026531836 1000 docker 2712 2712 2382 0
158.738720 ubuntu 4026531840 4026531836 1000 docker 2712 2712 2382 0
158.738762 ubuntu 4026531840 4026531836 1000 docker 2712 2712 2382 0
gs: O_RDONLY|O_LARGEFILE, dev: 8388613, lnode: 140786
158.738802 ubuntu 4026531840 4026531836 1000 docker 2712 2712 2382 3
f, flags: O_RDONLY|O_CLOEXEC, mode: 3822718664
158.738830 ubuntu 4026531840 4026531836 1000 docker 2712 2712 2382 0
158.738973 ubuntu 4026531840 4026531836 1000 docker 2712 2712 2382 0
158.739054 ubuntu 4026531840 4026531836 1000 docker 2712 2712 2382 0
_RDONLY|O_LARGEFILE, dev: 8388613, lnode: 140018
158.739133 ubuntu 4026531840 4026531836 1000 docker 2712 2712 2382 3
gs: O_RDONLY|O_CLOEXEC, mode: 3822718664
158.739219 ubuntu 4026531840 4026531836 1000 docker 2712 2712 2382 0
158.739452 ubuntu 4026531840 4026531836 1000 docker 2712 2712 2382 0
158.739577 ubuntu 4026531840 4026531836 1000 docker 2712 2712 2382 0
_RDONLY|O_LARGEFILE, dev: 8388613, lnode: 139883
158.739649 ubuntu 4026531840 4026531836 1000 docker 2712 2712 2382 3
gs: O_RDONLY|O_CLOEXEC, mode: 3822718664
158.740003 ubuntu 4026531840 4026531836 1000 docker 2712 2712 2382 0
158.740188 ubuntu 4026531840 4026531836 1000 docker 2712 2712 2382 0
158.895883 ubuntu 4026531840 4026531836 1000 docker 2712 2712 2382 0
e, flags: O_RDONLY|O_LARGEFILE, dev: 23, lnode: 3804
158.896000 ubuntu 4026531840 4026531836 1000 docker 2712 2712 2382 3

```

컨테이너 버그 찾기 – Introspection Tool

- 컨테이너 기본 정보
- 프로세스 생성
 - 사용하는 바이너리의 전체 경로 및 Namespace 관찰
 - 프로세스가 동적으로 로딩하는 라이브러리의 전체 경로 확인
- 프로세스의 **Capability**
- 정책 위반 탐지
 - 호스트의 namespace 로 컨테이너의 라이브러리를 로딩하지 않아야 함
 - 호스트의 namespace 로 컨테이너의 바이너리를 사용하지 않아야 함
 - 컨테이너가 호스트에 대한 파일 식별자를 보유하지 않아야 함

컨테이너 버그 찾기 – Introspection Tool



DETECT

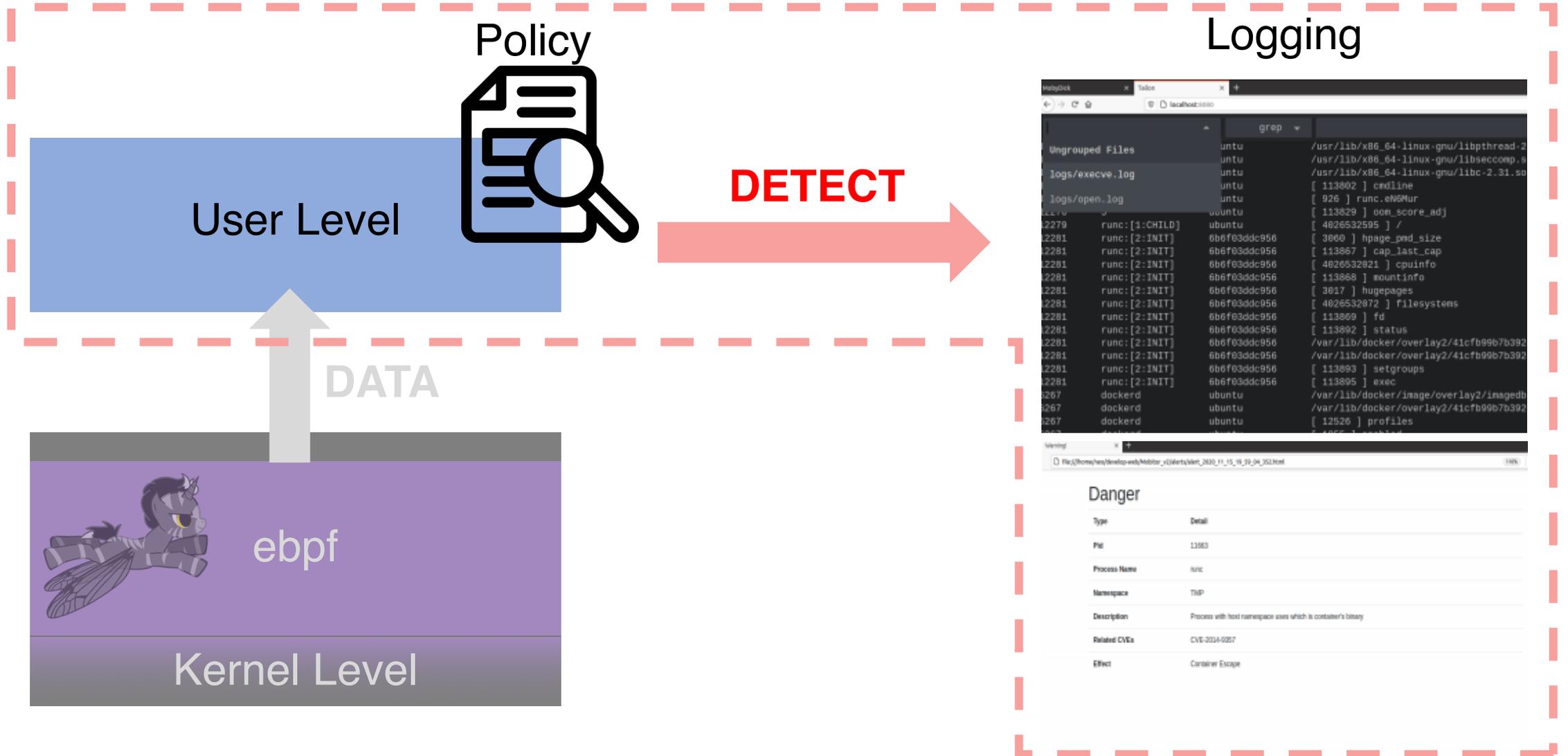


Logging

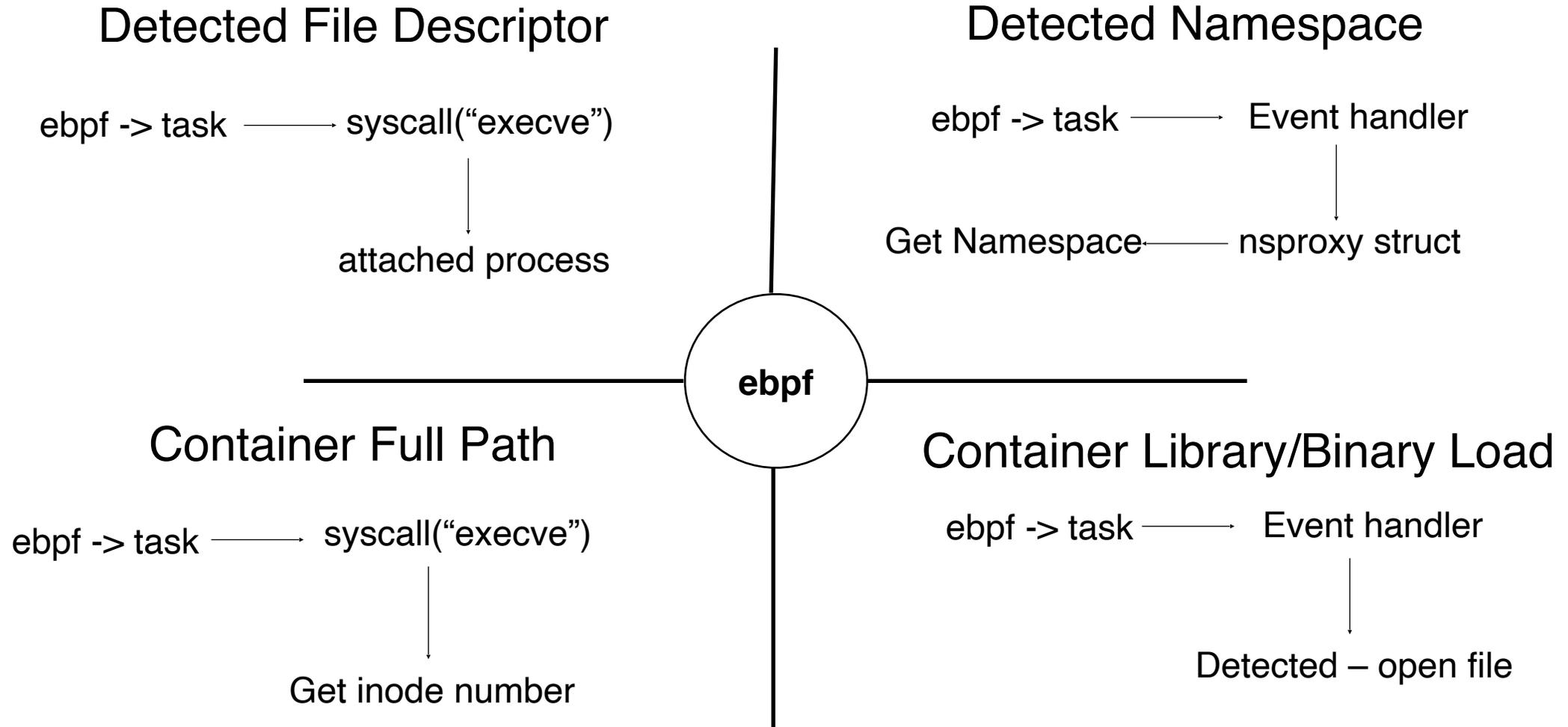
```
Ungranted Files      untu      /usr/lib/x86_64-linux-gnu/libpthread-2
logs/execve.log      untu      /usr/lib/x86_64-linux-gnu/libseccomp.s
logs/open.log        untu      [ 113802 ] cwdline
llv                   ubuntu   [ 926 ] runc.ebpf
2279 runc:[1:CHILD]    ubuntu   [ 113829 ] oom_score_adj
2281 runc:[2:INIT]    6b6f03ddc956 [ 4826532595 ] /
2281 runc:[2:INIT]    6b6f03ddc956 [ 3866 ] hpage_pmd_size
2281 runc:[2:INIT]    6b6f03ddc956 [ 113867 ] cap_last_cap
2281 runc:[2:INIT]    6b6f03ddc956 [ 4826532821 ] cpuinfo
2281 runc:[2:INIT]    6b6f03ddc956 [ 113868 ] mountinfo
2281 runc:[2:INIT]    6b6f03ddc956 [ 3817 ] hugepages
2281 runc:[2:INIT]    6b6f03ddc956 [ 4826532872 ] filesystems
2281 runc:[2:INIT]    6b6f03ddc956 [ 113869 ] fd
2281 runc:[2:INIT]    6b6f03ddc956 [ 113892 ] status
2281 runc:[2:INIT]    6b6f03ddc956 /var/lib/docker/overlay2/41cfb99b7b392
2281 runc:[2:INIT]    6b6f03ddc956 /var/lib/docker/overlay2/41cfb99b7b392
2281 runc:[2:INIT]    6b6f03ddc956 [ 113893 ] setgroups
2281 runc:[2:INIT]    6b6f03ddc956 [ 113895 ] exec
267 dockerd         ubuntu   /var/lib/docker/image/overlay2/insagedb
267 dockerd         ubuntu   /var/lib/docker/overlay2/41cfb99b7b392
267 dockerd         ubuntu   [ 12526 ] profiles
```

Danger	
Type	Detail
Pid	11383
Process Name	runc
Namespace	TIDP
Description	Process with host namespace exec which is container's binary
Related CVEs	CVE-2016-0257
Effect	Container Escape

컨테이너 버그 찾기 – Introspection Tool



컨테이너 버그 찾기 – Introspection Tool



컨테이너 버그 찾기 – Introspection Tool

```

[+] Turning on Library Monitor
[+] Initializing Environment
[+] Running ...

Save to ./alerts/alert_2021_05_28_18_50_24_0.html
comm: weaver 19027
hostname: masternode
filename: ld-musl-x86_64.so.1
[root path] 1573735
/var/lib/docker/overlay2/e5b941e3063fea628453439260dfec54a28bf5012c684865701d33149be0dee4/diff
/var/lib/docker/overlay2/e5b941e3063fea628453439260dfec54a28bf5012c684865701d33149be0dee4/merged
[fullpath] 1455362
/var/lib/docker/overlay2/e5b941e3063fea628453439260dfec54a28bf5012c684865701d33149be0dee4/merged/lib/ld-musl-x86_64.so.1
/var/lib/docker/overlay2/85b8acd883514c44d7e866e469cd97fc8f648c649ddf1c25fdf1718b1b16771c/diff/lib/ld-musl-x86_64.so.1
/var/lib/docker/overlay2/bc6b56798e2cac24856695f641ac09ad74420e57fadf9db86efbc602d3db794c/merged/lib/ld-musl-x86_64.so.1
namespace: PID UTS
=====
```

- Docker Full Escape Bug **4개 중 3개** 탐지
 - Kubernetes Escape Bug **2개** 탐지
 - 최신 Kubernetes에서 사용하는 weave(CNI plugin)에서 위반 상황 탐지 (**취약점 발견**)
- ⇒ html 파일 저장(실시간)

**3. Host Escape
(Use the bug we found)**

Arbitrary code execution

Host Root File System



**1. CVE-2020-8559
/ RBAC Config**



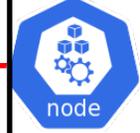
Kubernetes Cluster



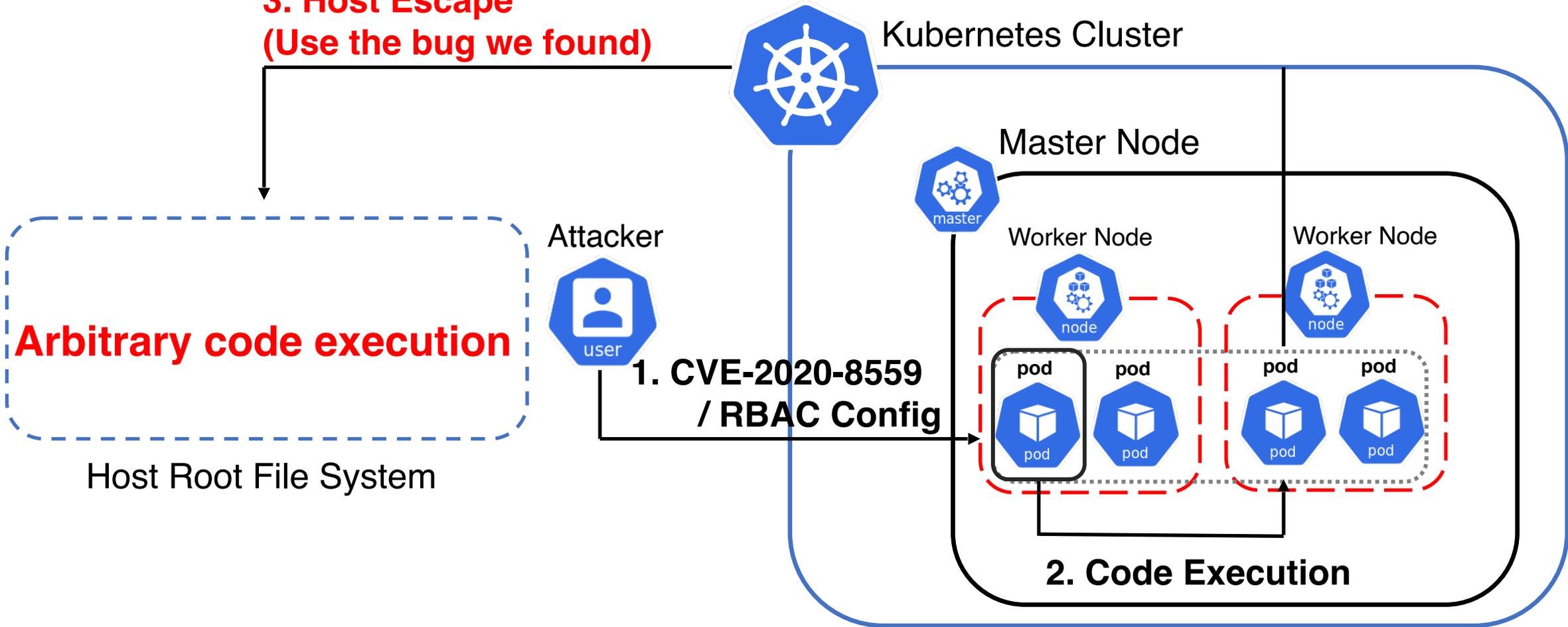
Master Node

Worker Node

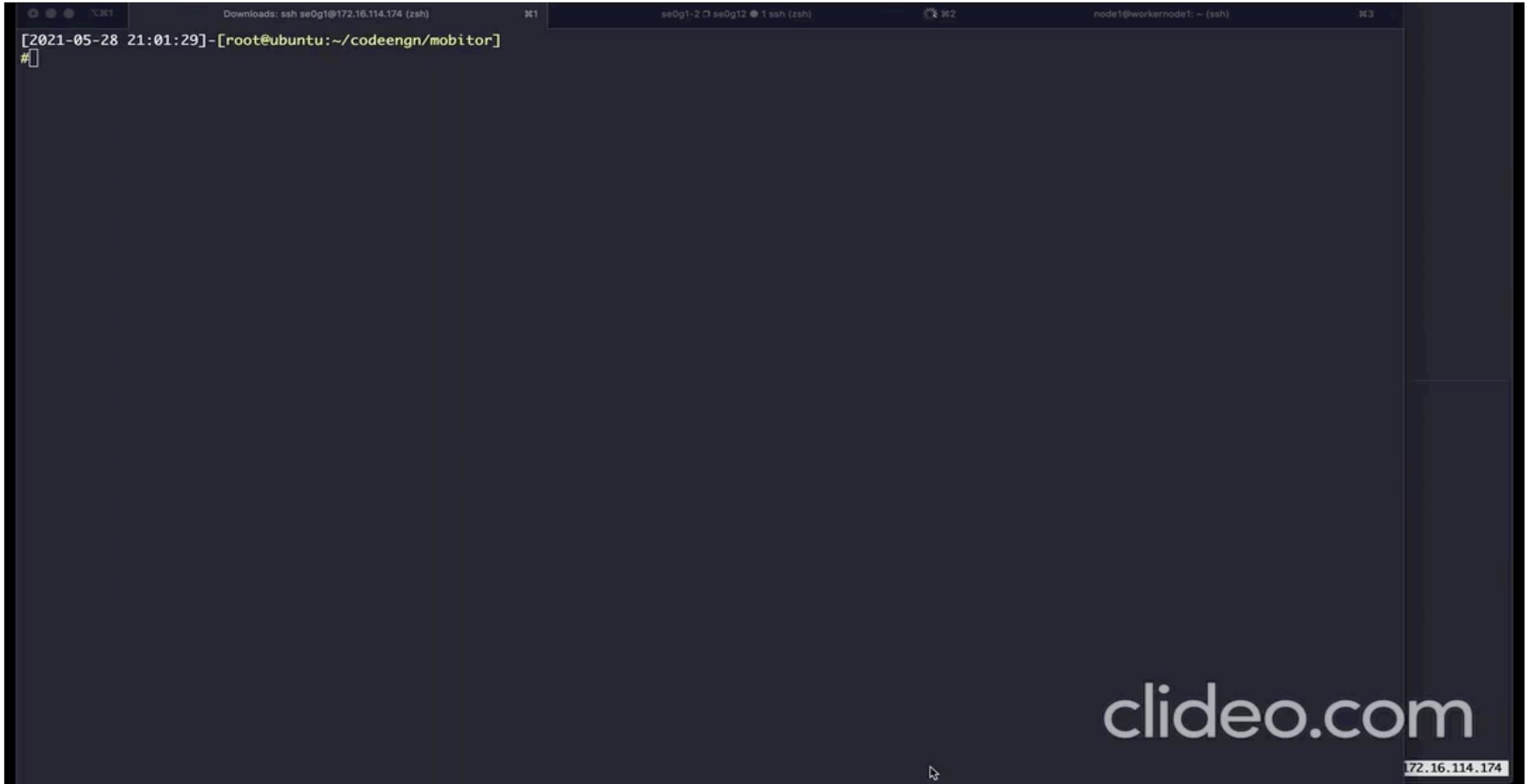
Worker Node



2. Code Execution



컨테이너 버그 찾기 – Introspection Tool



공격 시나리오

1. 다른 POD에 명령을 내릴 수 있는 1Day / RBAC 가 존재
2. PID Namespace가 적용된 라이브러리에 실행시킬 명령어 추가
3. /proc file system에 접근하여, 다른 POD의 Host에 임의 파일 생성 및 실행 가능

영향도

1. Host로 탈출이 가능한 POD 중에 master node의 권한을 가진 pod의 경우 다른 pod들을 제어할 수 있게 됨
2. 서비스 운영 중인 사이트의 POD의 host로 escape이 될 경우 Host의 주요 자원에 접근하여 가용성 침해
3. 인증서 파일 및 설정 파일들을 탈취 가능

패치전

```
volumeMounts:  
  - name: xtables-lock  
    mountPath: /run/xtables.lock  
    readOnly: false  
hostNetwork: true  
dnsPolicy: ClusterFirstWithHostNet  
hostPID: true  
restartPolicy: Always  
securityContext:  
  seLinuxOptions: {}  
serviceAccountName: weave-net
```

Weave.yaml

패치후

```
volumeMounts:  
  - name: xtables-lock  
    mountPath: /run/xtables.lock  
dnsPolicy: ClusterFirstWithHostNet  
hostNetwork: true  
initContainers:  
  - name: weave-init  
command:  
  - /home/weave/init.sh  
image: 'docker.io/weaveworks/weave-kube:2.8.1'
```

Weave.yaml

Threat Modeling



CVE-2020-29389
CVE-2020-35467
CVE-2020-35468
CVE-2020-35190

...
총 36개

CodeQL For GO (Code Auditing)



오픈소스 기여
잘못된 코드 패턴 탐지

Introspection Tool



CVE-2020-26278

감사합니다.

How To Find Container Platform Bug?

Team. MobyDick