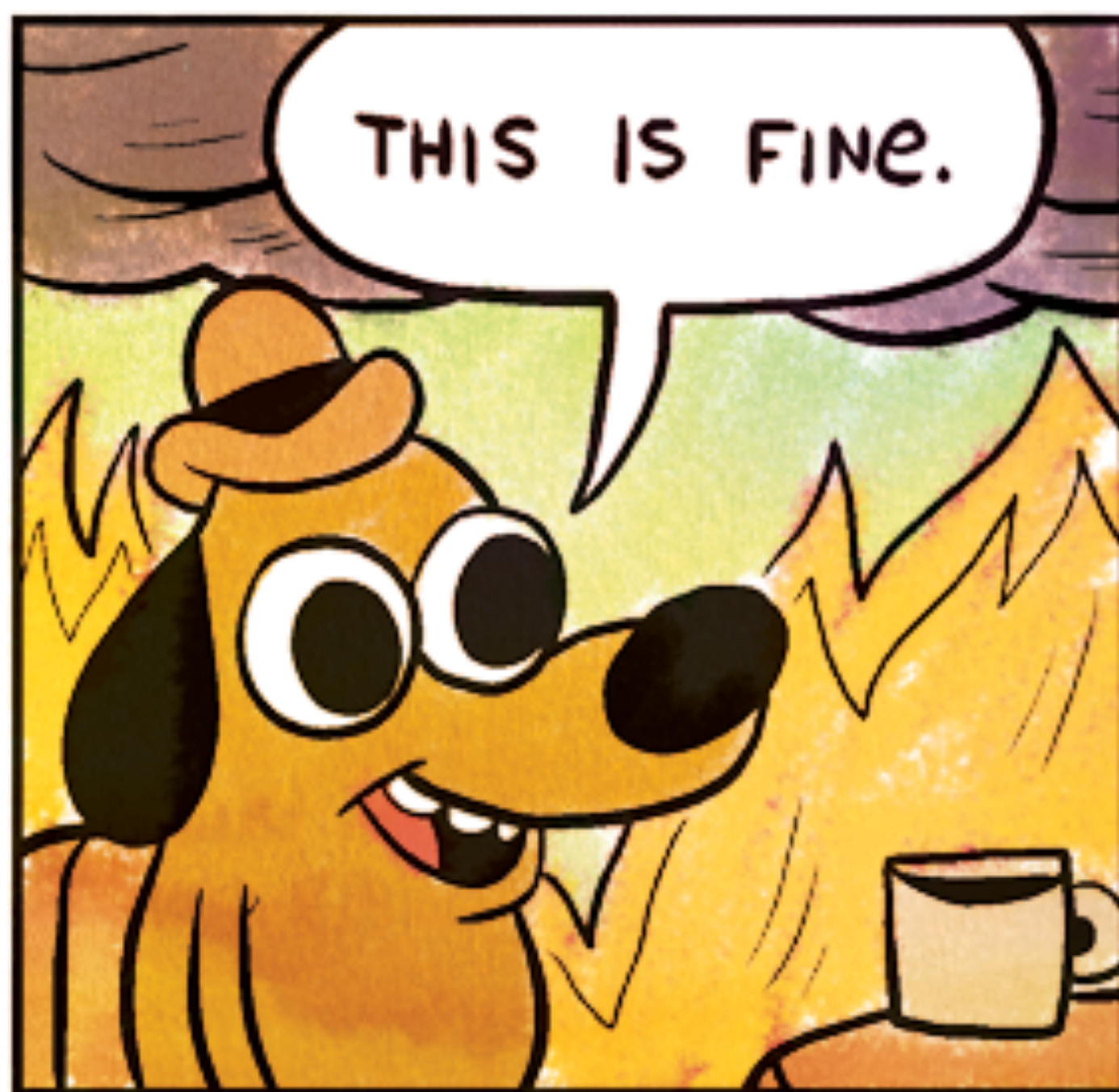


# SECCOMP

## YOUR NEXT LAYER OF DEFENSE

PHILIPP KRENN

@XERAA



UNTIL SOMETHING  
HAPPENS







**NO** SILVER BULLET



# PRINCIPLE OF LEAST PRIVILEGE 🖐️



# SECCOMP

**PREVENT** EXECUTION OF CERTAIN  
SYSTEM CALLS BY AN APPLICATION

# SECCOMP

INSTRUMENT KERNEL TO **ABORT**  
CERTAIN CALLS OR **KILL** THE PROCESS



# SECCOMP

## AN APPLICATION **SANDBOX**

# HISTORY

ADDED IN LINUX KERNEL 2.6.12 IN 2005



**SET 1 IN /PROC/\$PID/SECCOMP TO ENTER  
STRICT MODE**

**ONLY ALLOW** `read, write, exit, sigreturn()`

# HISTORY

KERNEL 3.5 IN 2012 ADDED FOUNDATION  
TO CONTROL SYSTEM CALLS



# HISTORY

KERNEL 3.17 IN 2014 ADDED A SYSTEM  
CALL NAMED `seccomp` FOR EASIER  
CONFIGURATION

man syscalls

man seccomp



# REGISTER SECCOMP FILTER

WRITTEN AS **BERKELEY PACKET FILTER**  
(BPF)

# MINIMAL SETUP

```
#include <sys/prctl.h>  
#include <linux/seccomp.h>
```

```
prctl(PR_SET_SECCOMP, SECCOMP_MODE_FILTER, &bpf_prog)
```

# MINIMAL EXAMPLE

```
#include <linux/filter.h>

#define syscall_nr (offsetof(struct seccomp_data, nr))
#define arch_nr (offsetof(struct seccomp_data, arch))

#define VALIDATE_ARCHITECTURE \
    BPF_STMT(BPF_LD+BPF_W+BPF_ABS, arch_nr), \
    BPF_JUMP(BPF_JMP+BPF_JEQ+BPF_K, ARCH_NR, 1, 0), \
    BPF_STMT(BPF_RET+BPF_K, SECCOMP_RET_KILL)

#define EXAMINE_SYSCALL \
    BPF_STMT(BPF_LD+BPF_W+BPF_ABS, syscall_nr)

#define ALLOW_SYSCALL(name) \
    BPF_JUMP(BPF_JMP+BPF_JEQ+BPF_K, __NR_##name, 0, 1), \
    BPF_STMT(BPF_RET+BPF_K, SECCOMP_RET_ALLOW)

#define KILL_PROCESS \
    BPF_STMT(BPF_RET+BPF_K, SECCOMP_RET_KILL)
```

# REGISTERED SECCOMP FILTER

**EVERY** SYSTEM CALL OF THAT  
APPLICATION TRIGGERS EXECUTION OF  
FILTERS



# PERFORMANCE?

## KERNEL SPACE

# POSSIBLE FILTER RESULT

- SYSTEM CALL CAN BE ALLOWED
- PROCESS OR THE THREAD CAN BE KILLED
- ERROR IS RETURNED TO THE CALLER IN ADDITION TO LOGGING

# IS ANYONE USING IT?

GOOGLE CHROME. FIREFOX. OPENSSSH.  
DOCKER. QEMU. SYSTEMD. ANDROID.  
FIRECRACKER....

# DOCKER

‘[...] SANE DEFAULT FOR RUNNING CONTAINERS WITH SECCOMP AND DISABLES AROUND 44 SYSTEM CALLS OUT OF 300+.’

[HTTPS://GITHUB.COM/MOBY/MOBY/BLOB/MASTER/PROFILES/SECCOMP/DEFAULT.JSON](https://github.com/moby/moby/blob/master/profiles/seccomp/default.json)



# BLOCKED SYSCALLS

clock\_settime, clone, reboot,  
unshare...

# RUN WITHOUT THE DEFAULT SECCOMP PROFILE

```
$ docker run --rm -it \
  --security-opt seccomp=unconfined debian:stretch-slim \
  unshare --map-root-user --user sh -c whoami
root
```

```
$ docker run --rm -it debian:stretch-slim \
  unshare --map-root-user --user sh -c whoami
unshare: unshare failed: Operation not permitted
```

**IS ANY OF YOUR APPS  
USING IT?**

```
$ grep Seccomp /proc/*/status
/proc/1/status:Seccomp: 0
/proc/10/status:Seccomp: 0
/proc/100/status:Seccomp: 0
/proc/13369/status:Seccomp: 0
/proc/14/status:Seccomp: 0
/proc/15/status:Seccomp: 0
/proc/15137/status:Seccomp: 2
/proc/15153/status:Seccomp: 2
/proc/15174/status:Seccomp: 2
/proc/16/status:Seccomp: 0
...
```

```
$ head /proc/15137/status
```

```
Name: systemd-network
```

```
Umask: 0022
```

```
State: S (sleeping)
```

```
Tgid: 15137
```

```
Ngid: 0
```

```
Pid: 15137
```

```
PPid: 1
```

```
TracerPid: 0
```

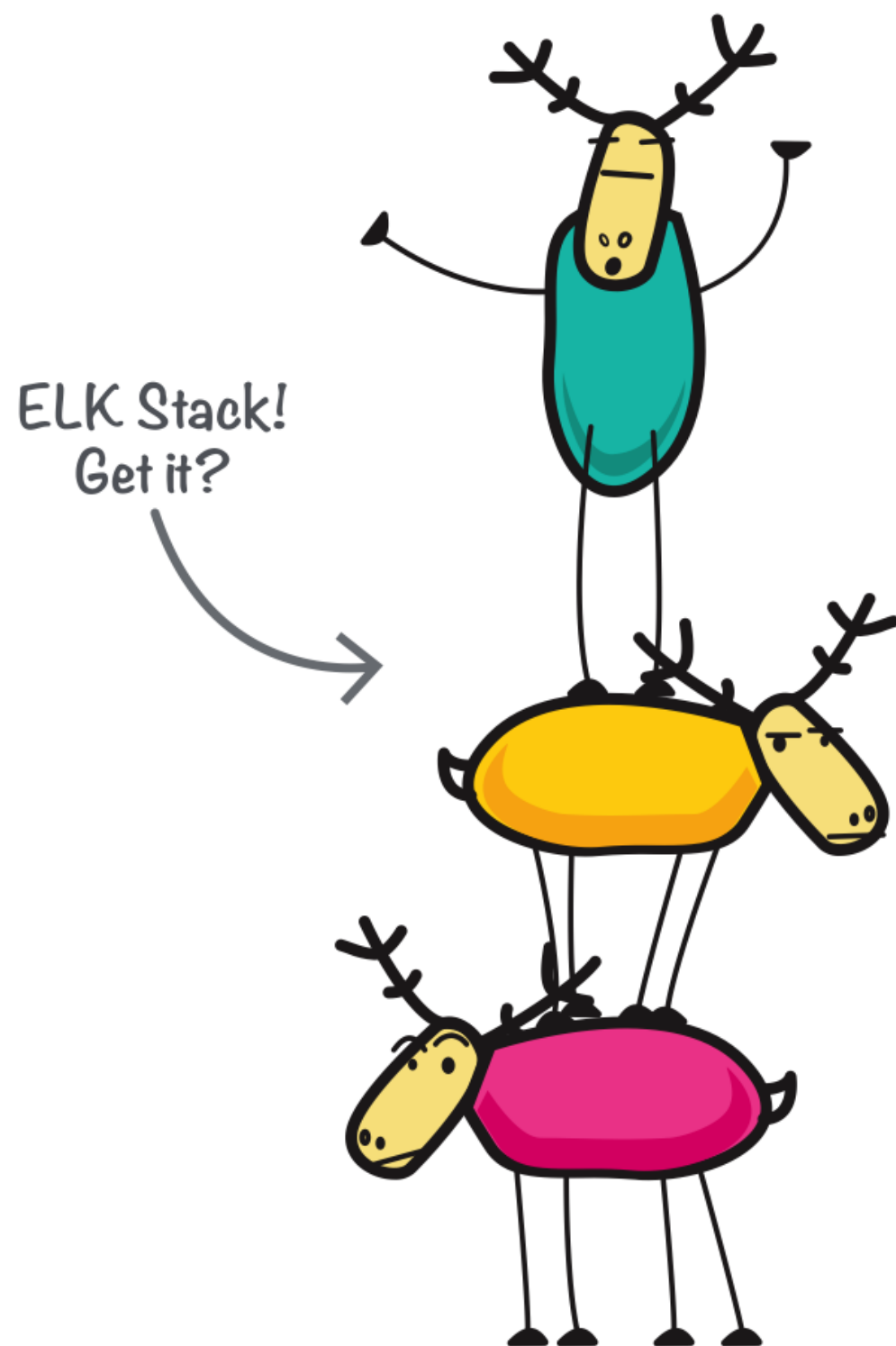
```
Uid: 100 100 100 100
```

```
Gid: 102 102 102 102
```





DEVELOPER 🥑

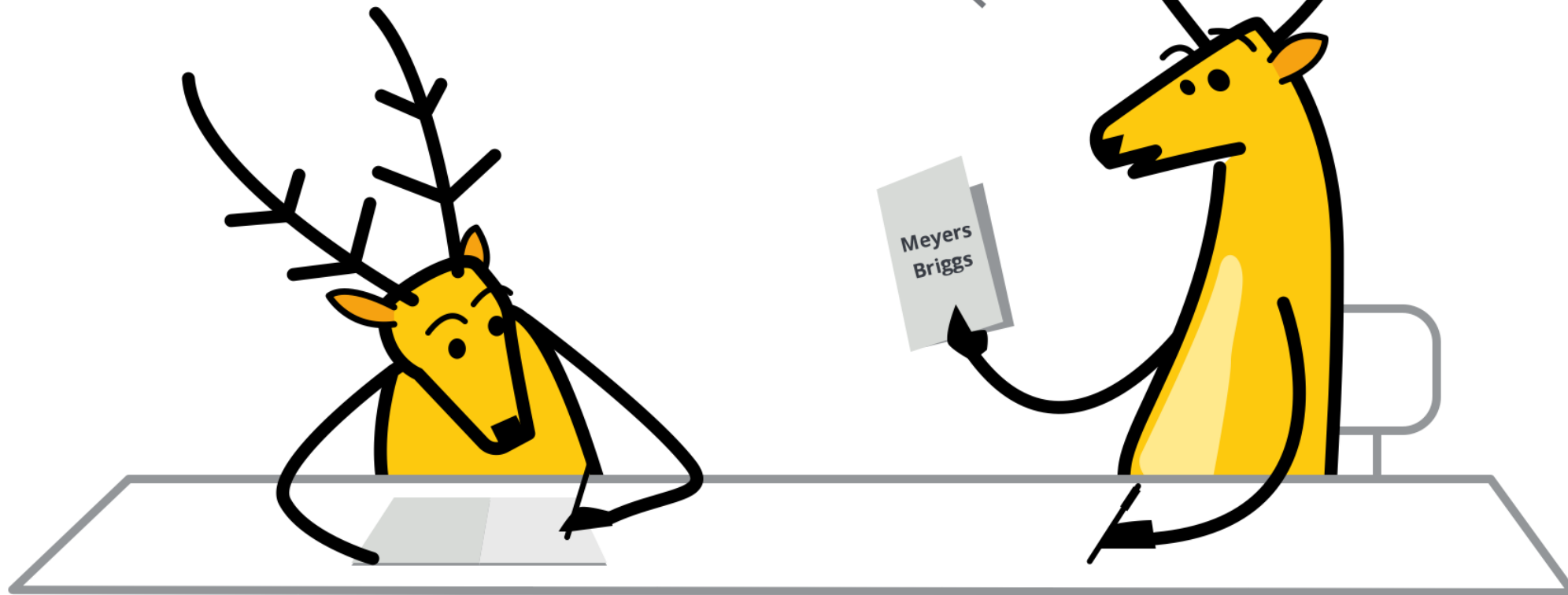


**E** Elasticsearch

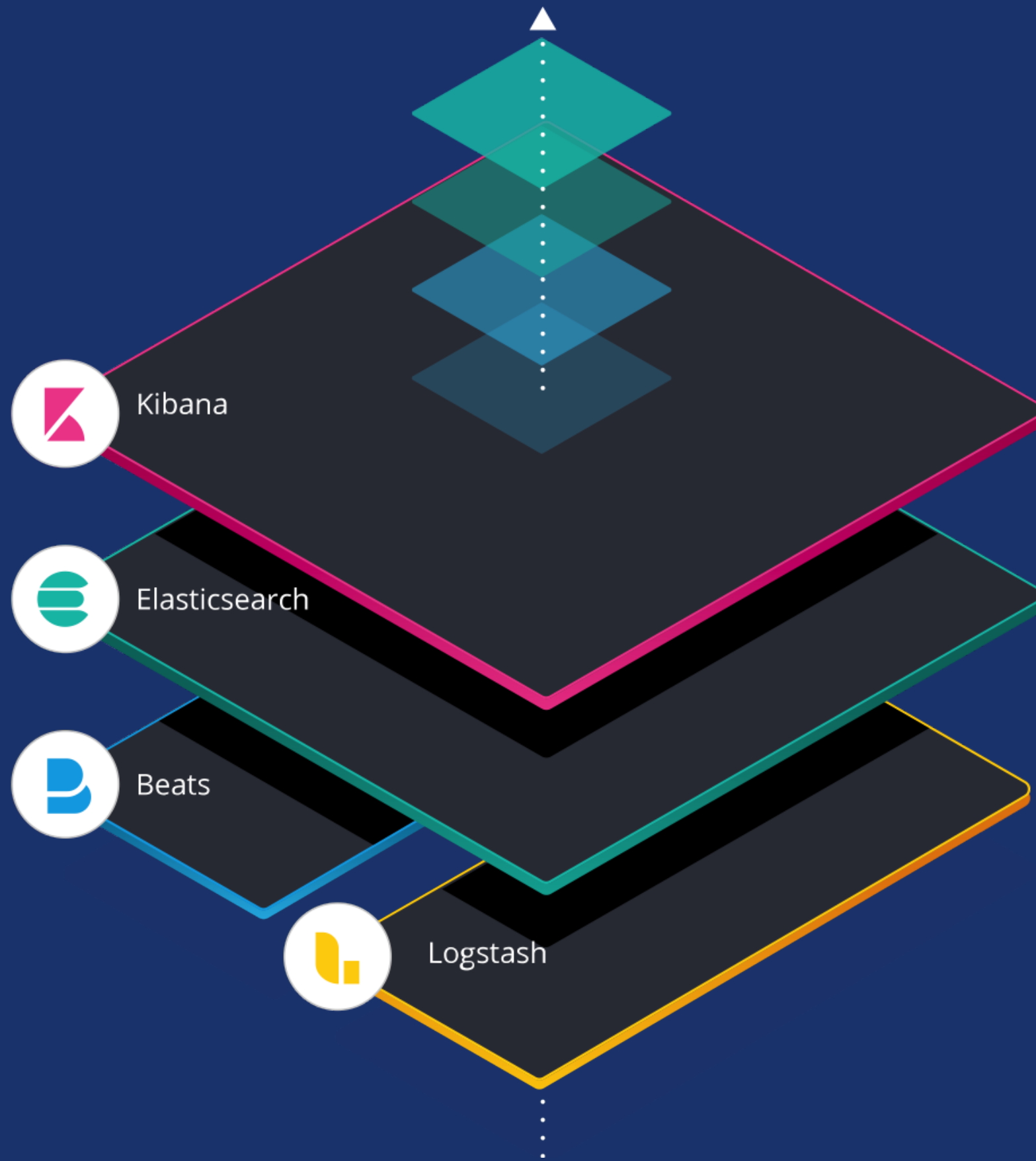
**L** Logstash

**K** Kibana

*Apparently, I'm an  
ELKB personality.*







# ELASTICSEARCH USING JAVA NATIVE ACCESS (JNA)

[HTTPS://GITHUB.COM/ELASTIC/ELASTICSEARCH/BLOB/7.5/SERVER/SRC/MAIN/JAVA/ORG/ELASTICSEARCH/BOOTSTRAP/BOOTSTRAP.JAVA#L100](https://github.com/elastic/elasticsearch/blob/7.5/server/src/main/java/org/elasticsearch/bootstrap/Bootstrap.java#L100)



```
public static void initializeNatives(Path tmpFile, boolean mlockAll, boolean systemCallFilter, boolean ctrlHandler) {  
    final Logger logger = LogManager.getLogger(Bootstrap.class);  
  
    // check if the user is running as root, and bail  
    if (Natives.definitelyRunningAsRoot()) {  
        throw new RuntimeException("can not run elasticsearch as root");  
    }  
  
    // enable system call filter  
    if (systemCallFilter) {  
        Natives.tryInstallSystemCallFilter(tmpFile);  
    }  
}
```

```
public static void initializeNatives(Path tmpFile, boolean mlockAll, boolean systemCallFilter, boolean ctrlHandler) {  
    final Logger logger = LogManager.getLogger(Bootstrap.class);  
  
    // check if the user is running as root, and bail  
    if (Natives.definitelyRunningAsRoot()) {  
        throw new RuntimeException("can not run elasticsearch as root");  
    }  
  
    // enable system call filter  
    if (systemCallFilter) {  
        Natives.tryInstallSystemCallFilter(tmpFile);  
    }  
}
```

```
public static void initializeNatives(Path tmpFile, boolean mlockAll, boolean systemCallFilter, boolean ctrlHandler) {  
    final Logger logger = LogManager.getLogger(Bootstrap.class);  
  
    // check if the user is running as root, and bail  
    if (Natives.definitelyRunningAsRoot()) {  
        throw new RuntimeException("can not run elasticsearch as root");  
    }  
  
    // enable system call filter  
    if (systemCallFilter) {  
        Natives.tryInstallSystemCallFilter(tmpFile);  
    }  
}
```

```

static int init(Path tmpFile) throws Exception {
    if (Constants.LINUX) {
        return linuxImpl();
    } else if (Constants.MAC_OS_X) {
        // try to enable both mechanisms if possible
        bsdImpl();
        macImpl(tmpFile);
        return 1;
    } else if (Constants.SUN_OS) {
        solarisImpl();
        return 1;
    } else if (Constants.FREE_BSD || OPENBSD) {
        bsdImpl();
        return 1;
    } else if (Constants.WINDOWS) {
        windowsImpl();
        return 1;
    } else {
        throw new UnsupportedOperationException("syscall filtering not supported for OS: '" + Constants.OS_NAME + "'");
    }
}

```

[HTTPS://GITHUB.COM/ELASTIC/ELASTICSEARCH/BLOB/7.5/SERVER/SRC/MAIN/JAVA/ORG/ELASTICSEARCH/BOOTSTRAP/SYSTEMCALLFILTER.JAVA#L615-L635](https://github.com/elastic/elasticsearch/blob/7.5/server/src/main/java/org/elasticsearch/bootstrap/systemcallfilter.java#L615-L635)

# MORE OPERATING SYSTEMS

SIMILAR FEATURES. DIFFERENT NAME

```
// BPF installed to check arch, limit, then syscall.
// See https://www.kernel.org/doc/Documentation/prctl/seccomp_filter.txt for details.
SockFilter insns[] = {
    /* 1 */ BPF_STMT(BPF_LD + BPF_W + BPF_ABS, SECCOMP_DATA_ARCH_OFFSET), //
    /* 2 */ BPF_JUMP(BPF_JMP + BPF_JEQ + BPF_K, arch.audit, 0, 7), // if (arch != audit) goto fail;
    /* 3 */ BPF_STMT(BPF_LD + BPF_W + BPF_ABS, SECCOMP_DATA_NR_OFFSET), //
    /* 4 */ BPF_JUMP(BPF_JMP + BPF_JGT + BPF_K, arch.limit, 5, 0), // if (syscall > LIMIT) goto fail;
    /* 5 */ BPF_JUMP(BPF_JMP + BPF_JEQ + BPF_K, arch.fork, 4, 0), // if (syscall == FORK) goto fail;
    /* 6 */ BPF_JUMP(BPF_JMP + BPF_JEQ + BPF_K, arch.vfork, 3, 0), // if (syscall == VFORK) goto fail;
    /* 7 */ BPF_JUMP(BPF_JMP + BPF_JEQ + BPF_K, arch.execve, 2, 0), // if (syscall == EXECVE) goto fail;
    /* 8 */ BPF_JUMP(BPF_JMP + BPF_JEQ + BPF_K, arch.execveat, 1, 0), // if (syscall == EXECVEAT) goto fail;
    /* 9 */ BPF_STMT(BPF_RET + BPF_K, SECCOMP_RET_ALLOW), // pass: return OK;
    /* 10 */ BPF_STMT(BPF_RET + BPF_K, SECCOMP_RET_ERRNO | (EACCES & SECCOMP_RET_DATA)), // fail: return EACCES;
};
```

[HTTPS://GITHUB.COM/ELASTIC/ELASTICSEARCH/BLOB/7.5/SERVER/SRC/MAIN/JAVA/ORG/ELASTICSEARCH/BOOTSTRAP/SYSTEMCALLFILTER.JAVA#L260-L414](https://github.com/elastic/elasticsearch/blob/7.5/server/src/main/java/org/elasticsearch/bootstrap/systemcallfilter.java#L260-L414)

# BEATS

## GO LIBRARY FOR INSTALLING A SECCOMP BPF SYSTEM CALL FILTER

[HTTPS://GITHUB.COM/ELASTIC/GO-SECCOMP-BPF](https://github.com/elastic/go-seccomp-bpf)



# SECCOMP IN YAML

```
seccomp:
  default_action: allow

syscalls:
  # Network sandbox example (NOT used by Beats)
  - action: errno
    names:
      - connect
      - accept
      - sendto
      - recvfrom
      - sendmsg
      - recvmsg
      - bind
      - listen
```

# BEATS USE ALLOW LISTS

[HTTPS://GITHUB.COM/ELASTIC/BEATS/BLOB/7.5/LIBBEAT/COMMON/SECCOMP/POLICY\\_LINUX\\_AMD64.GO](https://github.com/elastic/beats/blob/7.5/libbeat/common/seccomp/policy_linux_amd64.go)

```
func init() {
    defaultPolicy = &seccomp.Policy{
        DefaultAction: seccomp.ActionErrno,
        Syscalls: []seccomp.SyscallGroup{
            {
                Action: seccomp.ActionAllow,
                Names: []string{
                    "accept",
                    "accept4",
                    "access",
                    "arch_prctl",
                    "bind",
                    "brk",
                    ...
                }
            }
        }
    }
}
```

**PREFER  
ALLOW OVER DENY**

**ADDITIONAL SYSCALLS –  
MOVING TARGET**

# DEMO

# Server

```
nc -v -l 1025
```

# Client

```
telnet xeraa.wtf 1025
```

# DEMO

```
$ strace -e bind nc -v -l 1025
```

```
bind(3, {sa_family=AF_INET, sin_port=htons(1025),  
      sin_addr=inet_addr("0.0.0.0")}, 16) = 0
```

```
Listening on [0.0.0.0] (family 0, port 1025)
```

```
$ strace -c nc -v -l 1025
Listening on [0.0.0.0] (family 0, port 1025)
```

% time	seconds	usecs/call	calls	errors	syscall
0.00	0.000000	0	5		read
0.00	0.000000	0	1		write
0.00	0.000000	0	7		close
0.00	0.000000	0	7		fstat
0.00	0.000000	0	17		mmap
0.00	0.000000	0	12		mprotect
0.00	0.000000	0	1		munmap
0.00	0.000000	0	3		brk
0.00	0.000000	0	3		rt_sigaction
0.00	0.000000	0	1		rt_sigprocmask
0.00	0.000000	0	7	6	access
0.00	0.000000	0	1		socket
0.00	0.000000	0	1		bind
0.00	0.000000	0	1		listen
0.00	0.000000	0	2		setsockopt
0.00	0.000000	0	1		execve
0.00	0.000000	0	1		arch_prctl
0.00	0.000000	0	1		set_tid_address
0.00	0.000000	0	7		openat
0.00	0.000000	0	1		set_robust_list
0.00	0.000000	0	1		prlimit64
100.00	0.000000		81	6	total



# SYSCALL REPORTING

[HTTPS://GITHUB.COM/ANTITREE/SYSCALL2SECCOMP](https://github.com/antitree/syscall2seccomp)

[HTTPS://OUTFLUX.NET/TEACH-SECCOMP/STEP-3/SYSCALL-REPORTER.C](https://outflux.net/teach-seccomp/step-3/syscall-reporter.c)

# FIREJAIL

## LINUX NAMESPACES AND SECCOMP-BPF SANDBOX

[HTTPS://GITHUB.COM/NETBLUE30/FIREJAIL](https://github.com/netblue30/firejail)



# DEMO

```
$ firejail --noprofile --seccomp.drop=bind -c nc -v -l 1025
```

# DEMO

```
$ firejail --noprofile --seccomp.drop=bind -c strace nc -v -l 1025
```

```
...
```

```
bind(3, {sa_family=AF_INET, sin_port=htons(1025),  
      sin_addr=inet_addr("0.0.0.0")}, 16) = ?
```

```
+++ killed by SIGSYS (core dumped) +++
```

# HOW TO **STOP** PERMISSION CHANGES?

# 'NO NEW PRIVILEGES'

```
#include <sys/prctl.h>  
#include <linux/seccomp.h>
```

```
prctl(PR_SET_NO_NEW_PRIVS, 1, 0, 0, 0)  
prctl(PR_SET_SECCOMP, SECCOMP_MODE_FILTER, &bpf_prog)
```

# ELASTICSEARCH

```
static final int PR_SET_NO_NEW_PRIVS      = 38;    // since Linux 3.5

// ok, now set PR_SET_NO_NEW_PRIVS, needed to be able to set a seccomp filter as ordinary user
if (linux_prctl(PR_SET_NO_NEW_PRIVS, 1, 0, 0, 0) != 0) {
    throw new UnsupportedOperationException("prctl(PR_SET_NO_NEW_PRIVS): " + JNACLibrary.strerror(Native.getLastError()));
}

// check it worked
if (linux_prctl(PR_GET_NO_NEW_PRIVS, 0, 0, 0, 0) != 1) {
    throw new UnsupportedOperationException("seccomp filter did not really succeed: prctl(PR_GET_NO_NEW_PRIVS): " +
                                          JNACLibrary.strerror(Native.getLastError()));
}
```

[HTTPS://GITHUB.COM/ELASTIC/ELASTICSEARCH/BLOB/7.5/SERVER/SRC/MAIN/JAVA/ORG/ELASTICSEARCH/BOOTSTRAP/SYSTEMCALLFILTER.JAVA](https://github.com/elastic/elasticsearch/blob/7.5/server/src/main/java/org/elasticsearch/bootstrap/systemcallfilter.java)

# BEATS

```
filter := seccomp.Filter{  
    NoNewPrivs: true,  
    Flag:       seccomp.FilterFlagTSync,  
    Policy:     *p,  
}
```

[HTTPS://GITHUB.COM/ELASTIC/BEATS/BLOB/7.5/LIBBEAT/COMMON/SECCOMP/SECCOMP.GO](https://github.com/elastic/beats/blob/7.5/libbeat/common/seccomp/seccomp.go)

ALL THE THINGS!





[HTTPS://GITHUB.COM/LINUX-AUDIT](https://github.com/linux-audit/linux-audit)



# AUDITBEAT

# GO-LIBAUDIT

GO-LIBAUDIT IS A LIBRARY FOR  
COMMUNICATING WITH THE **LINUX AUDIT**  
**FRAMEWORK**

[HTTPS://GITHUB.COM/ELASTIC/GO-LIBAUDIT](https://github.com/elastic/go-libaudit)


```
> Nov 30, 2019 @ 13:21:06.156 event.action: violated-seccomp-policy @timestamp: Nov 30, 2019 @ 13:21:06.156 user.group.id: 1000
user.group.name: ubuntu user.name: ubuntu user.id: 1000 user.audit.id: 1000 user.audit.name: ubuntu
process.executable: /bin/nc.openbsd process.pid: 15366 process.name: nc service.type: auditd
host.architecture: x86_64 host.os.family: debian host.os.name: Ubuntu host.os.kernel: 4.15.0-1021-aws
host.os.codename: bionic host.os.platform: ubuntu host.os.version: 18.04.3 LTS (Bionic Beaver)
```

# ELASTIC SIEM

✕
☆
Untitled Timeline
🔒
📅
✓
L Show data
🔄 Refresh
⚙️

## Events

## Showing: 339,654 events



OR

event.action: "violated-seccomp-policy" ×

Drop here to build an OR query

Fields ▾ @timestamp ↓ message event

```
Session # 16 ubuntu @ xeraa.wtf violated seccomp policy with
>_ nc with result unknown
```

Session # 16 : ubuntu @ xeraa.wtf violated seccomp policy with  
>\_ nc with result unknown

Session # 16 ubuntu @ xeraa.wtf violated seccomp policy with  
>\_ nc with result unknown



# CONCLUSION







# SECCOMP **VS** SELINUX / APPARMOR

SIMILAR **KERNEL-LEVEL**  
**INTERCEPTION / FILTERING** OF  
SYSCALLS



# SECCOMP **VS** SELINUX / APPARMOR

## PROCESS ACTIVELY SETS SECCOMP **VS** MANDATORY ACCESS CONTROL POLICY BEFORE PROCESS RUNS

# SECCOMP

WIDELY AVAILABLE AND USED –  
USE IT!

# LIBSECCOMP

'PLATFORM INDEPENDENT. INTERFACE  
TO THE LINUX KERNEL'S SYSCALL  
FILTERING MECHANISM'

[HTTPS://GITHUB.COM/SECCOMP/LIBSECCOMP](https://github.com/seccomp/libseccomp)

# PS: WINDOWS

PROCESS\_MITIGATION\_SYSTEM\_  
CALL\_DISABLE\_POLICY

IMPOSE RESTRICTIONS ON WHAT  
SYSTEM CALLS A PROCESS CAN INVOKE

[HTTPS://DOCS.MICROSOFT.COM/EN-US/WINDOWS/WIN32/API/WINNT/NS-WINNT-  
PROCESS\\_MITIGATION\\_SYSTEM\\_CALL\\_DISABLE\\_POLICY](https://docs.microsoft.com/en-us/windows/win32/api/winnt/ns-winnt-process_mitigation_system_call_disable_policy)

# QUESTIONS?

PHILIPP KRENN

@XERAA

PS: STICKER

# CREDIT

## ALEXANDER REELEN

[HTTPS://WWW.ELASTIC.CO/BLOG/SECCOMP-IN-THE-ELASTIC-STACK](https://www.elastic.co/blog/seccomp-in-the-elastic-stack)