SECCOMP

YOUR NEXT LAYER OF DEFENSE

PHILIPP KRENN @XERAA


THIS IS FINE.

UNTIL SOMETHING HAPPENS


TH-
THIS IS NOT FINE!!
OH MY GOD EVERY THINGS ON FIRE
WHAT THE HELL IS MY PROBLEM

NO SILVER BULLET 

elastic

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

elastic

SECCOMP

AN APPLICATION SANDBOX

elastic

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()`

elastic

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

elastic

REGISTER SECCOMP FILTER

WRITTEN AS BERKELEY PACKET FILTER (BPF)

elastic

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. OPENSSH. 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

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
```

PS: DOCKER --CAP-ADD

CAPABILITY +
ALLOW SYSCALL IF REQUIRED

elastic

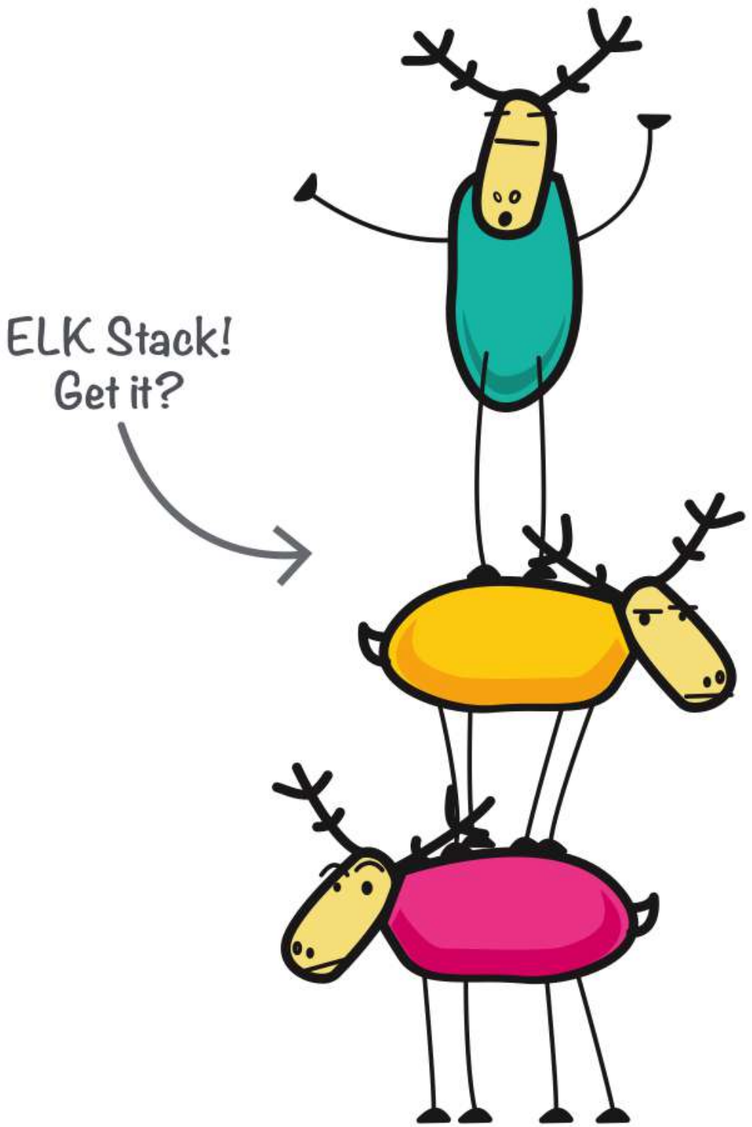
IS ANY OF YOUR APPS USING IT?

```
$ 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
```

elastic

DEVELOPER

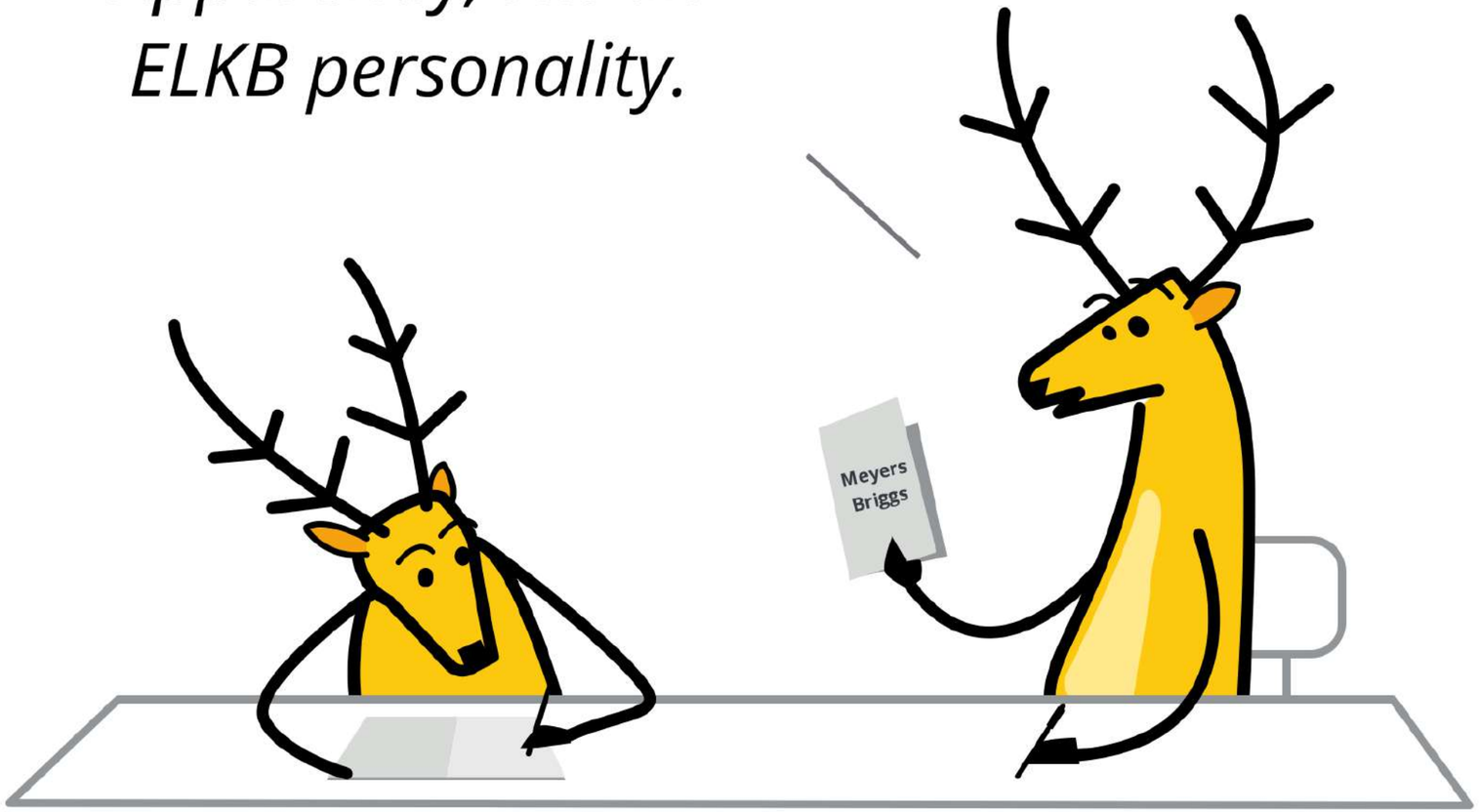


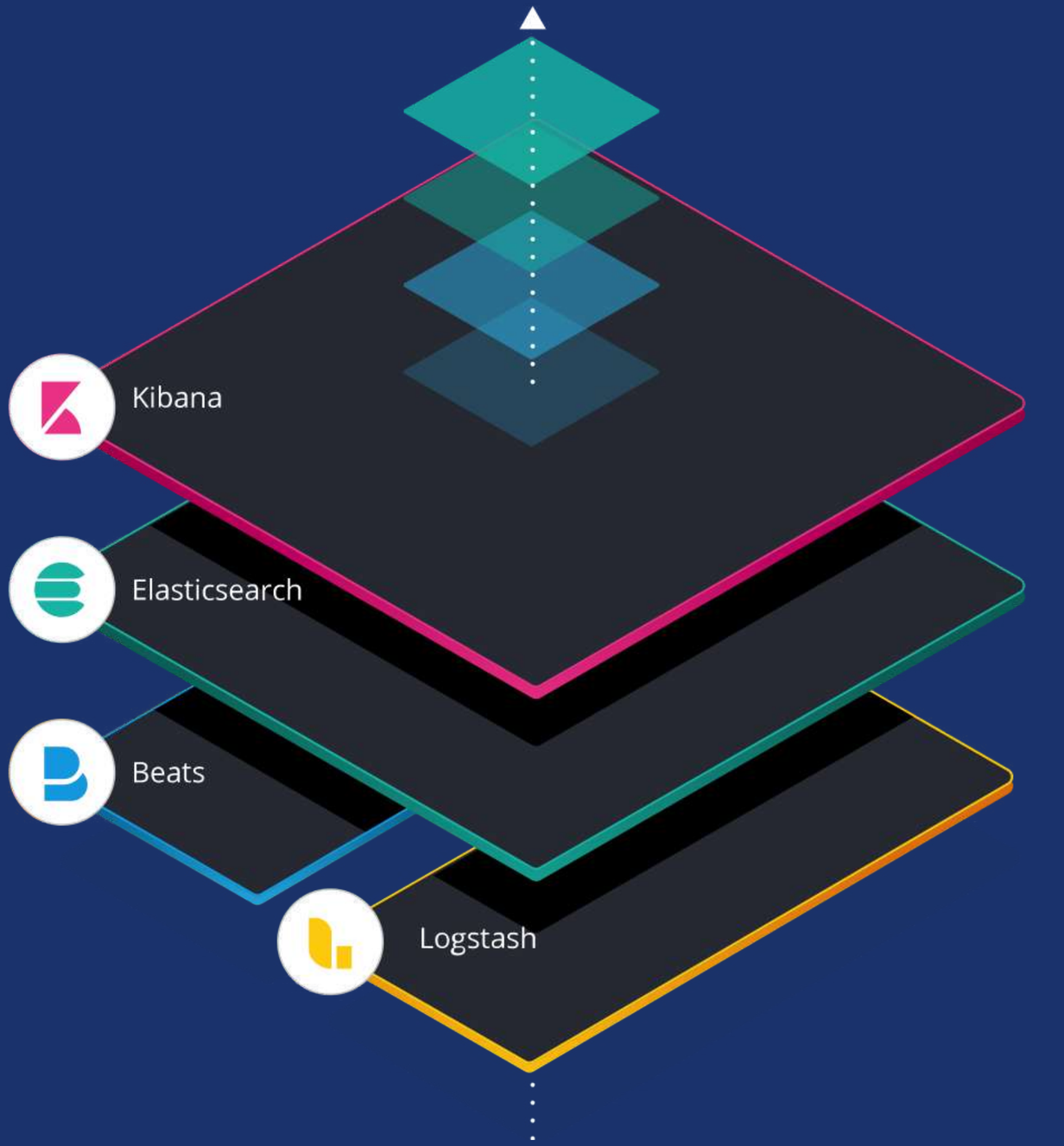
E Elasticsearch

L Logstash

K Kibana

Apparently, I'm an ELKB personality.




Kibana
Elasticsearch
Beats
Logstash

ELASTICSEARCH
USING JAVA NATIVE ACCESS (JNA)

https://github.com/elastic/elasticsearch/blob/7.9/server/src/main/java/org/elasticsearch/bootstrap/Bootstrap.java#L106

```java
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);
    }
```

```java
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/7.9/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/7.9/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

elastic

SECCOMP IN YAML

```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/7.9/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

elastic

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
```

elastic

SYSCALL REPORTING

https://github.com/antitree/syscall2seccomp

https://outflux.net/teach-seccomp/step-3/syscall-reporter.c

FIREJAIL

LINUX NAMESPACES AND SECCOMP-BPF SANDBOX

HTTPS://GITHUB.COM/NETBLUE30/FIREJAIL

elastic

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?

elastic

"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/7.9/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/7.9/libbeat/common/seccomp/seccomp.go


ALL THE THINGS!

AUDITD

https://github.com/linux-audit

elastic

AUDITBEAT

elastic

GO-LIBAUDIT

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

HTTPS://GITHUB.COM/ELASTIC/GO-LIBAUDIT

elastic

event.action: "violated-seccomp-policy" KQL | Last 1 hour Show dates | Refresh

+ Add filter

auditbeat-*

Search field names

Filter by type 0

Selected fields

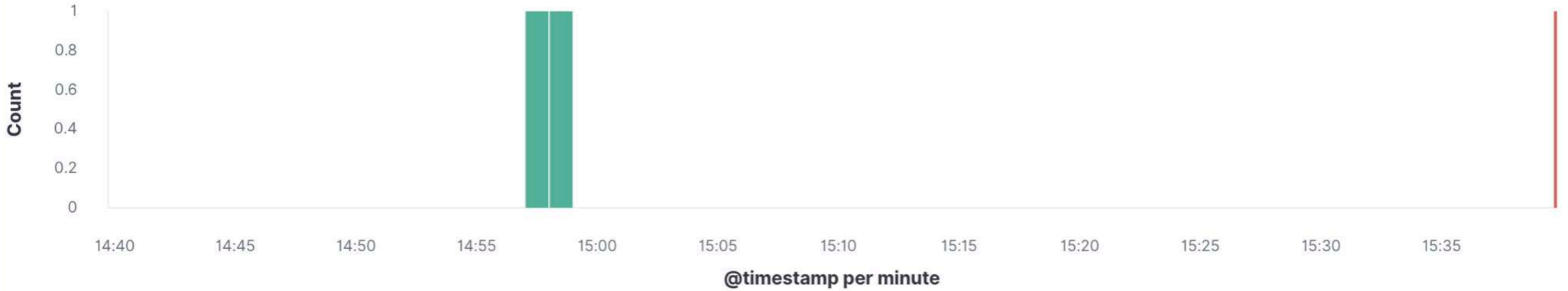
_source

Available fields

_id
_index
_score
_type
@timestamp
agent.ephemeral...
agent.hostname
agent.id
agent.name
agent.type
agent.version
auditd.data.arch

2 hits

Aug 23, 2020 @ 14:39:43.504 - Aug 23, 2020 @ 15:39:43.504 Auto



Time	_source
Aug 23, 2020 @ 14:58:00.051	event.action: violated-seccomp-policy @timestamp: Aug 23, 2020 @ 14:58:00.051 user.id: 1000 user.group.id: 1000 user.group.name: ubuntu user.name: ubuntu user.audit.id: 1000 user.audit.name: ubuntu service.type: auditd ecs.version: 1.5.0 host.ip: 172.26.15.250, fe80::414:6ff:fe01:8251 host.mac: 06:14:06:01:82:51 host.hostname: ip-172-26-15-250 host.name: xeraa.wtf host.architecture: x86_64 host.os.version: 18.04.5 LTS (Bionic Beaver) host.os.family: debian host.os.name: Ubuntu
Aug 23, 2020 @ 14:57:50.995	event.action: violated-seccomp-policy @timestamp: Aug 23, 2020 @ 14:57:50.995 event.category: process event.outcome: unknown event.kind: event event.type: info event.module: auditd tags: production host.hostname: ip-172-26-15-250 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: 18.04.5

ELASTIC SECURITY

elastic

seccomp KQL

+ Add filter

Events

Showing: 25 events

@timestamp	message	host.name
Aug 23, 2020 @ 14:58:00.051	—	xeraa.wtf
	Session # 5 ubuntu @ xeraa.wtf violated secco	
Aug 23, 2020 @ 14:57:50.995	—	xeraa.wtf
	Session # 5 ubuntu @ xeraa.wtf violated secco	
Aug 23, 2020 @ 14:44:51.392	Process man (PID: 21070) by...	xeraa.wtf
	ubuntu @ xeraa.wtf in /home/ubuntu stopp	
	# 396b7eb5260ff47e4be1ed61	
	Process man (PID: 21070) by us	

Untitled timeline Las Show dates Refresh

user.name: "ubuntu"

OR + Add field

AND Filter Search KQL All

+ Add filter

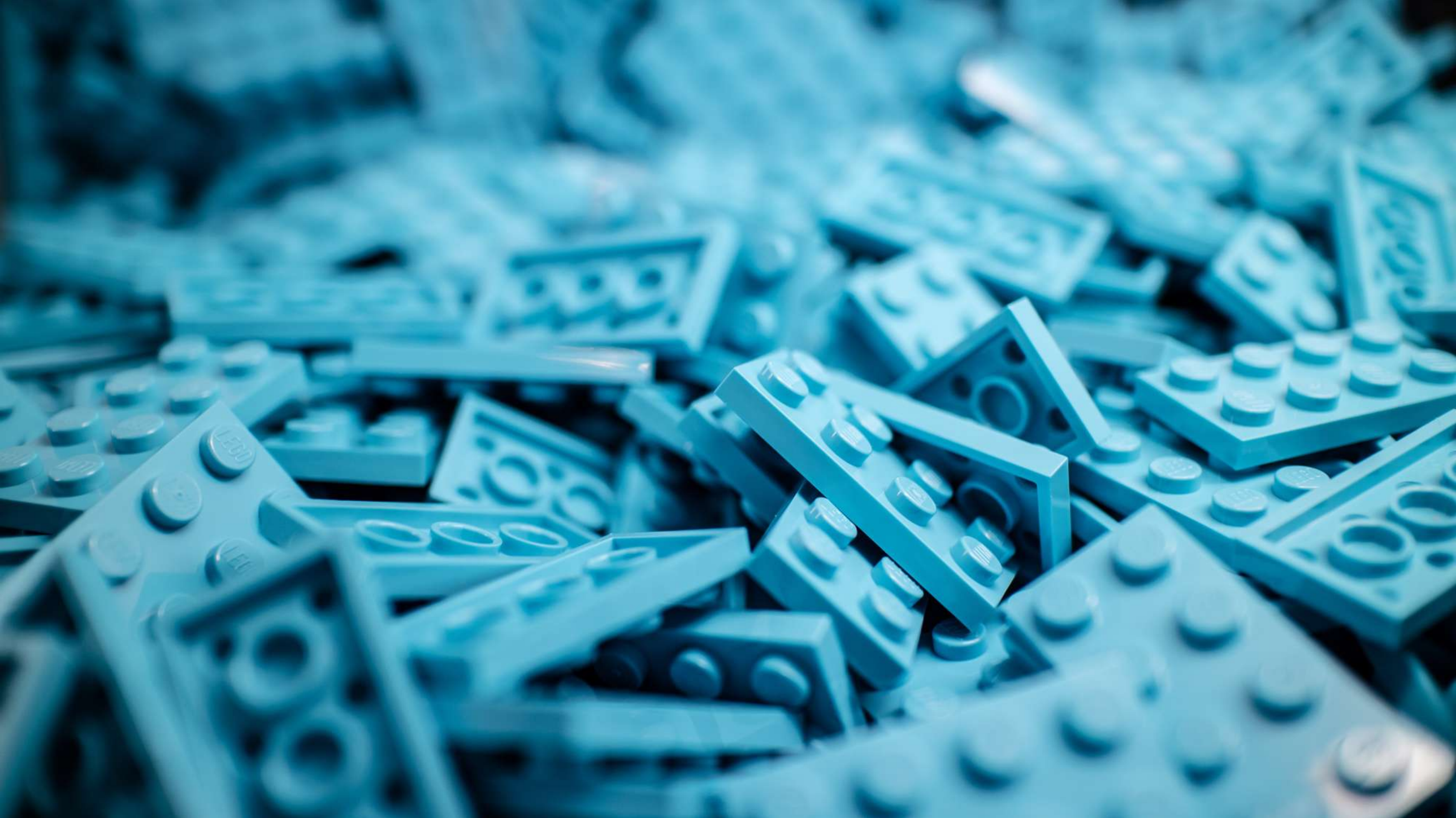
@timestamp	message	event.category
Aug 23, 2020 @ 14:58:00.231	—	process
	Session # 5 ubuntu @ xeraa.wtf crashed program >_ strace (21289) with result success	
Aug 23, 2020 @ 14:58:00.051	—	process
	Session # 5 ubuntu @ xeraa.wtf crashed program >_ nc (21291) with result success	
Aug 23, 2020 @ 14:58:00.051	—	process
	Session # 5 ubuntu @ xeraa.wtf violated seccomp policy with >_ nc	

50 of 1859 events match the search

Load more

Updated 22 seconds ago

CONCLUSION



SECCOMP VS SELINUX / APPARMOR

SIMILAR KERNEL-LEVEL INTERCEPTION / FILTERING OF SYSCALLS

elastic

SECCOMP VS SELINUX / APPARMOR

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

elastic

SECCOMP

WIDELY AVAILABLE AND USED - USE IT!

LIBSECCOMP

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

HTTPS://GITHUB.COM/SECCOMP/LIBSECCOMP

elastic

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

QUESTIONS?

PHILIPP KRENN @XERAA

PS: STICKER

elastic

CREDIT

ALEXANDER REELSEN

HTTPS://WWW.ELASTIC.CO/BLOG/SECCOMP-IN-THE-ELASTIC-STACK

elastic