
ONE IDENTITY
by Quest

What is new in sudo and syslog-ng – a BSD specific view

Peter Czanik

Open Source Evangelist

One Identity

@PCzanik

Overview

- /me FreeBSD user since 1994

- sudo
- syslog-ng
- sudo and syslog-ng in FreeBSD ports:
  - installation using pkg
  - compiling from ports
- using syslog-ng in Bastille

sudo

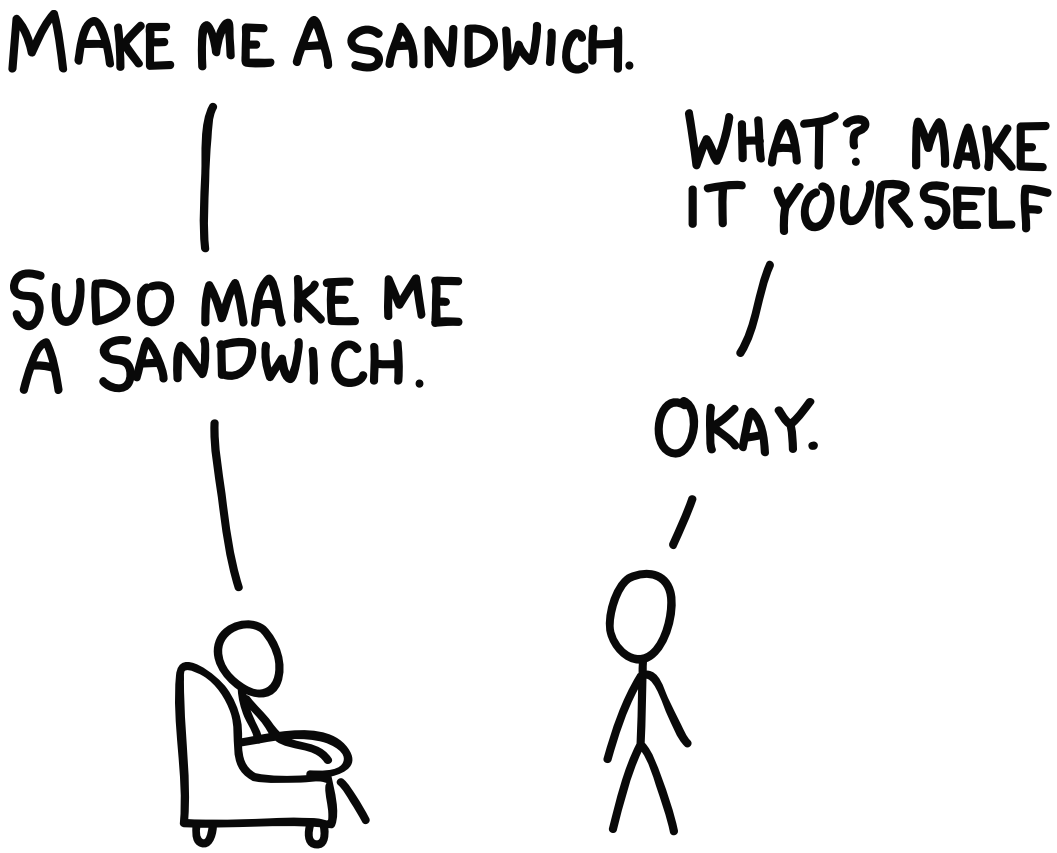
What is sudo?

- Answers, depending on experience:
  - A tool to complicate life
  - A prefix for administrative commands
  - A way to see who did what and other advanced features

What is sudo?

- Sudo allows a system administrator to delegate authority by giving certain users the ability to run some commands as root or another user while providing an audit trail of the commands and their arguments. (https://www.sudo.ws/)
- A lot more than just a prefix

It can get you a sandwich… (by XKCD)



Defaults

- Changing the default behavior:

```
Defaults secure_path="/usr/sbin:/usr/bin:/sbin:/bin"
Defaults env_keep = "LANG LC_ADDRESS LC_CTYPE"
Defaults !insults
```

- Making defaults user/host/etc specific

  Defaults:%wheel insults

Insults

- Fun, but not always politically correct :)

```
czanik@linux-mewy:~> sudo ls
[sudo] password for root:
Hold it up to the light --- not a brain in sight!
[sudo] password for root:
My pet ferret can type better than you!
[sudo] password for root:
sudo: 3 incorrect password attempts
czanik@linux-mewy:~>
```

Digest verification

peter ALL = sha244:11925141bb22866afdf257ce7790bd6275feda80b3b241c108b79c88 /usr/bin/passwd

- Cons:
  - Difficult to maintain
- Pros:
  - Modified binaries do not run
  - Additional layer of protection

Session recording

- Recording the terminal
- Playback
- Difficult to modify (not cleartext)
- Saved locally; therefore, easy to delete with unlimited access

- Sudo 1.9: central session recording

LDAP for central management

- Propagates in real-time
- Can’t be modified locally
- Many limitations

Python support

- Extending sudo using Python
- Using the same APIs as C plugins

- API: https://www.sudo.ws/man/sudo_plugin.man.html
  - Python plugin documentation: https://www.sudo.ws/man/sudo_plugin_python.man.html

- No development environment or compilation is needed

IO logs API

- Accessing input and output from user sessions
- Only one Python implementation is allowed

- Python examples:
  - Breaking connection if a given text appears on screen
  - Breaking connection if “rm -fr” is typed in the command line
  - Asking for the reason of the session

IO logs API example: code

```
import sudo

class MyIOPlugin(sudo.Plugin):
    def log_ttyout(self, buf):
        if "MySecret" in buf:
            sudo.log_info("Don't look at my secret!")
            return sudo.RC_REJECT
```

IO logs API example: screenshot

```
[czanik@centos7 ~]$ sudo -s
[root@centos7 czanik]# cd /root/
[root@centos7 ~]# ls
DoNotEnter  kick.py_v1  policy.py_v1  sng
kick.py     policy.py   __pycache__   sudo
[root@centos7 ~]# cd DoNotEnter/
[root@centos7 DoNotEnter]# ls
Don't look at my secret!
                    Hangup
[czanik@centos7 ~]$
```

syslog-ng

syslog-ng

Logging

Recording events, such as:

Jan 14 11:38:48 linux-0jbu sshd[7716]: Accepted publickey for root from 127.0.0.1 port 48806 ssh2

syslog-ng

Enhanced logging daemon with a focus on portability and high-performance central log collection. Originally developed in C.

Role: collecting data

Collect system and application logs together: contextual data for either side

A wide variety of platform-specific sources:

- /dev/log & co
- Journal, Sun streams

Syslog messages over the network:

- Legacy or RFC5424, UDP/TCP/TLS

Logs or any kind of text data from applications:

- Through files, sockets, pipes, application output, etc.

Python source: Jolly Joker

- HTTP server, Kafka source, etc.

Role: processing

Classifying, normalizing, and structuring logs with built-in parsers:

- CSV-parser, PatternDB, JSON parser, key=value parser

Rewriting messages:

- For example: anonymization

Reformatting messages using templates:

- Destination might need a specific format (ISO date, JSON, etc.)

Enriching data:

- GeoIP
- Additional fields based on message content

Python parser:

- all of the above, enriching logs from databases and also filtering

Role: filtering data

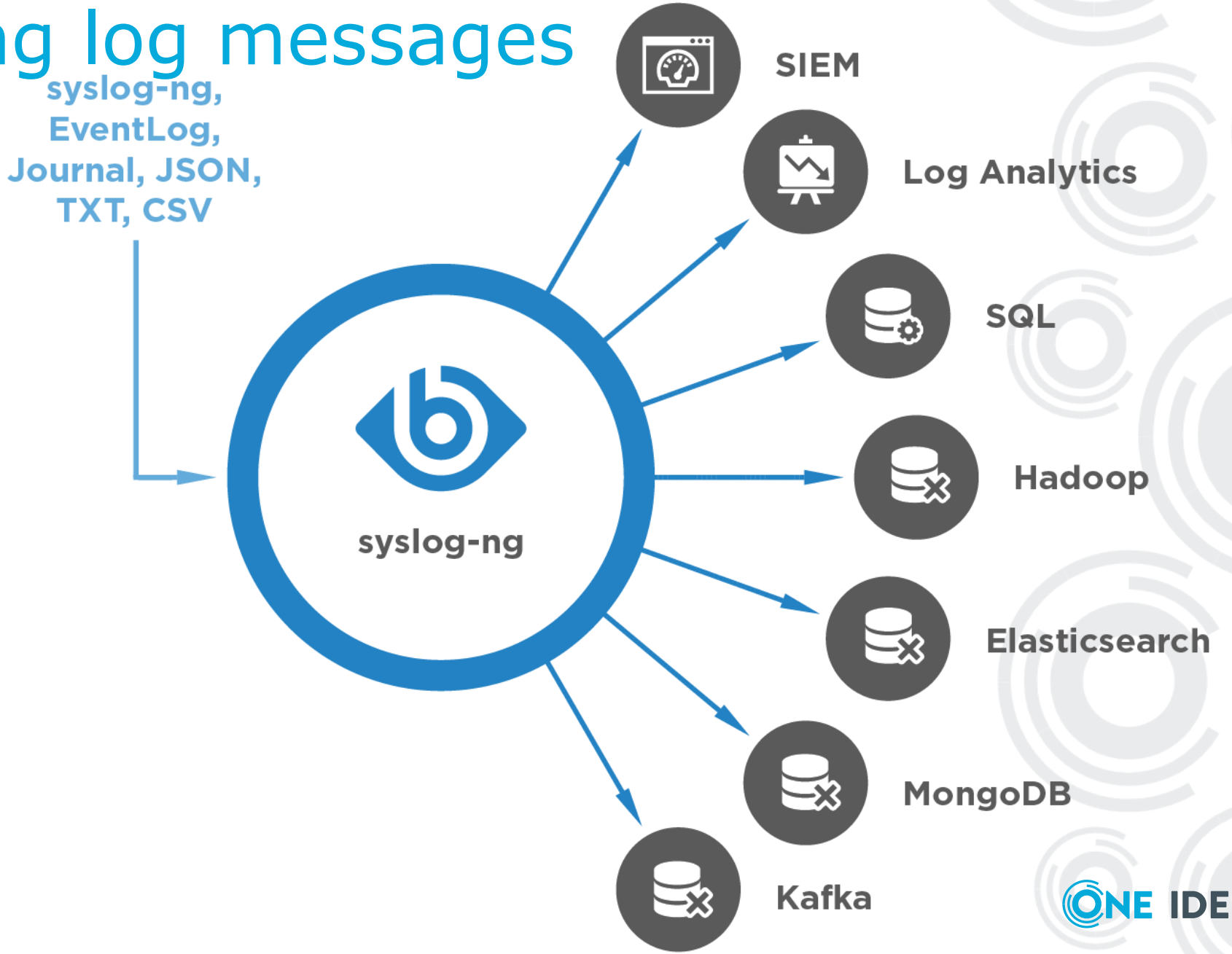
Main uses:

- Discarding surplus logs (not storing debug-level messages)
- Message routing (login events to SIEM)

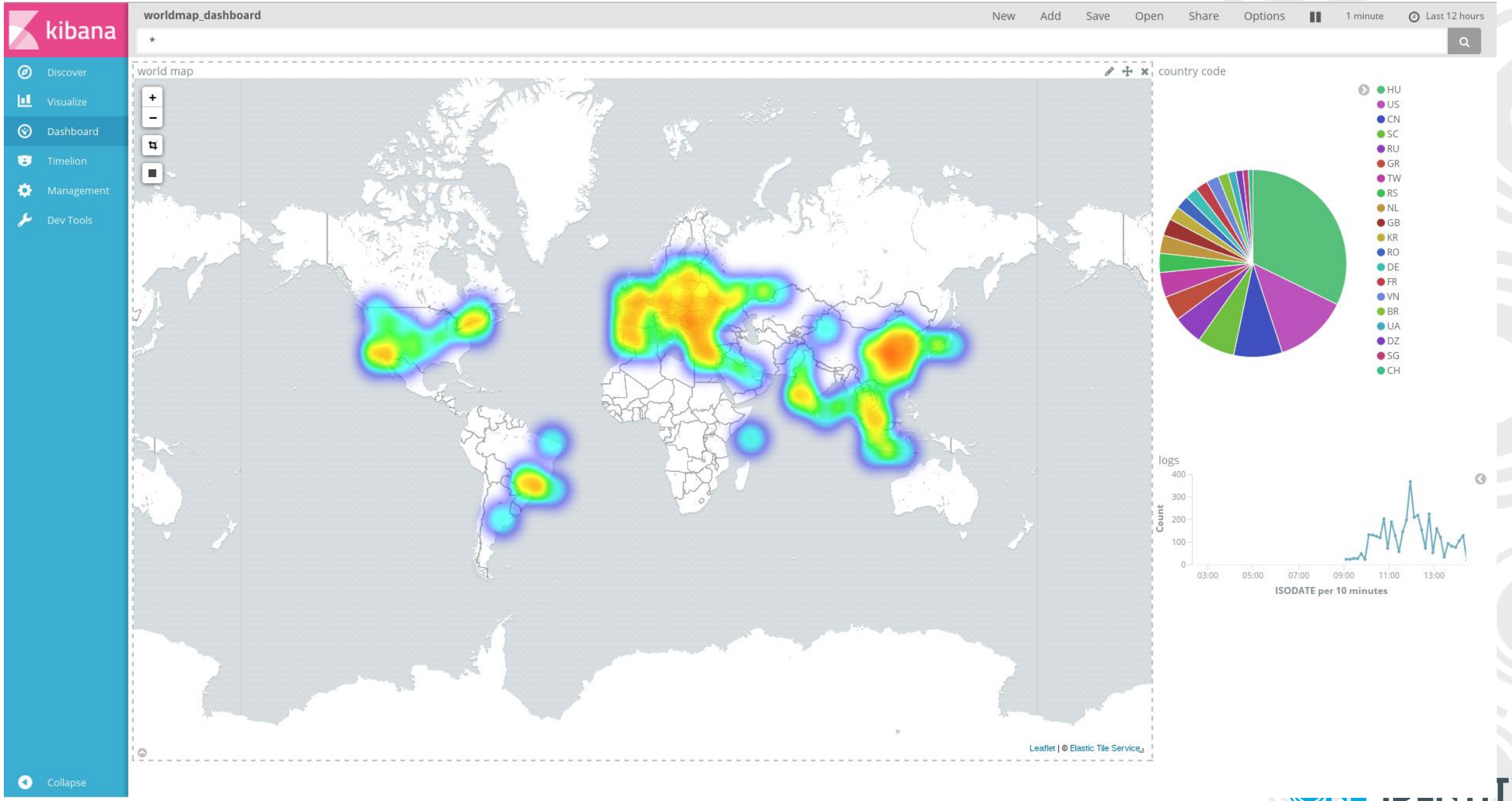
Many possibilities:

- Based on message content, parameters, or macros
- Using comparisons, wildcards, regular expressions, and functions
- Combining all of these with Boolean operators

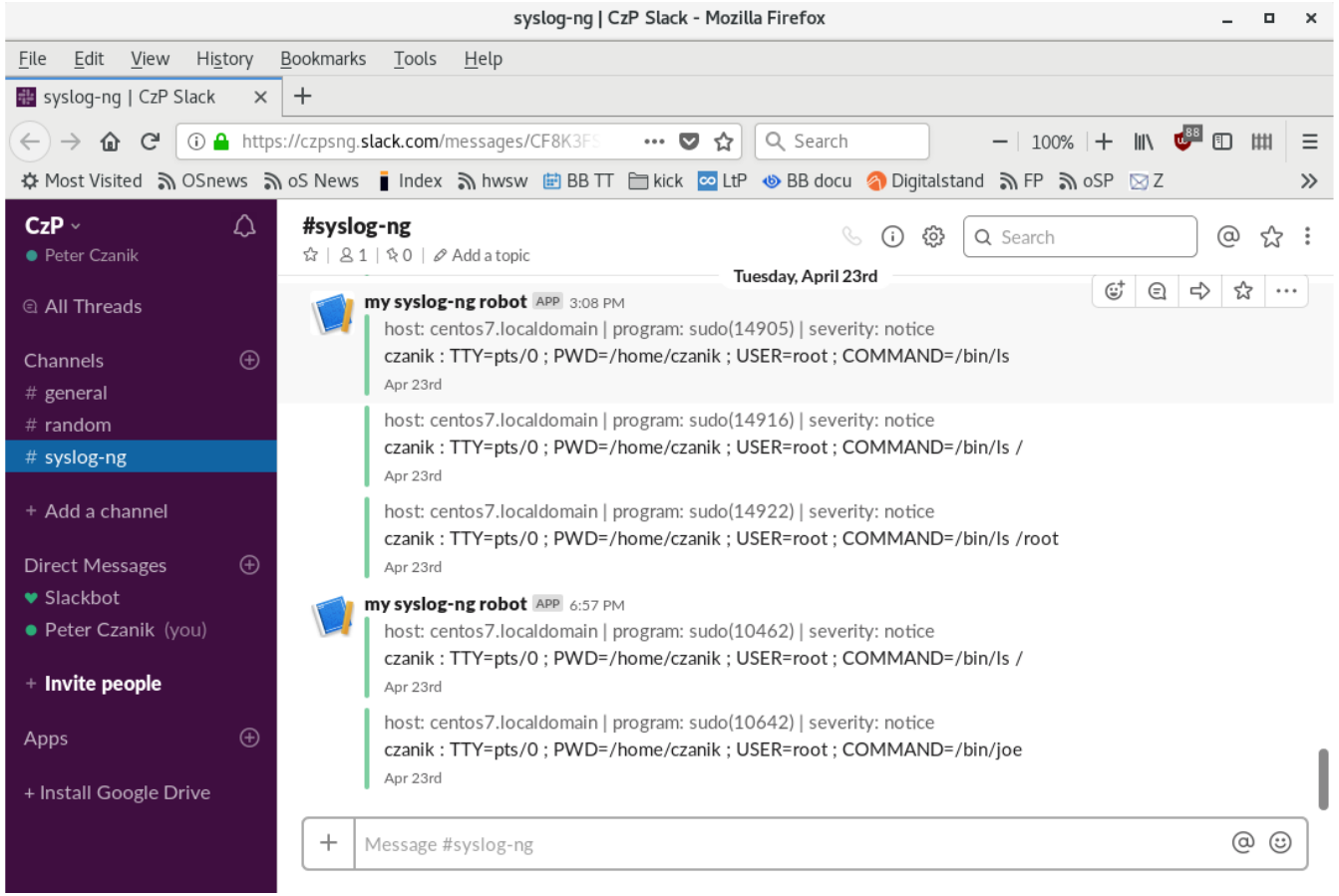
Role: storing log messages



ONE IDENTITY by Quest


kibana
worldmap_dashboard
New
Add
Save
Open
Share
Options
1 minute
Last 12 hours
*
Discover
Visualize
Dashboard
Timelion
Management
Dev Tools
world map
country code
HU
US
CN
SC
RU
GR
TW
RS
NL
GB
KR
RO
DE
FR
VN
BR
UA
DZ
SG
CH
logs
Count
400
300
200
100
0
03:00
05:00
07:00
09:00
11:00
13:00
ISODATE per 10 minutes
Leaflet | © Elastic Tile Service
Collapse

sudo logs in Slack



ONE IDENTITY by Quest

What is common between a BMW i3 and a Kindle?

ONE IDENTITY by Quest

Where else are sudo and syslog-ng used?

- Sudo: almost all BSD/Linux/UNIX/macOS systems
- Syslog-ng:
  - Available in most Linux distributions, *BSD
  - Most BSD-based appliances: FreeNAS, OPNsense, etc.
  - Synology & QNAP NAS
  - Turris Omnia firewall
- Most likely also in space ☺

Sudo in FreeBSD ports

- security/sudo
- Up-to-date: follows 1.9 line
- Pkg: only basic functionality

Sudo in FreeBSD ports

- You need to compile sudo for:
  - Insults ☺
  - LDAP
  - Python
  - Kerberos

Syslog-ng in FreeBSD ports

- sysutils/syslog-ng
- I am a co-maintainer
- Pkg: only basic functionalities
  - Define basic! ☺
  - Needs extra dependency

Syslog-ng in FreeBSD ports

- These are now all enabled:
  - SSL/TLS, JSON, HTTP (Curl)

- You need to compile syslog-ng from ports for:
  - Python, Java
  - SQL, MongoDB
  - AMQP, Kafka, Redis
  - SMTP, SNMP
  - GeoIP
  - Etc.

Running syslog-ng in Bastille

- https://bastillebsd.org/
- Bastille is an open-source system for automating deployment and management of containerized applications on FreeBSD
- No hard dependencies (uses git for templates)
- There is a syslog-ng template

Running syslog-ng in Bastille

- Install from ports: sysutils/bastille
- Setup Bastille first
- Setup your firewall (pf)
- Create a new jail
- Install syslog-ng using a template
- Configure firewall

Many more possibilities: ZFS, vnet, etc.

Running syslog-ng in Bastille

- cd /usr/ports/sysutils/bastille/
- make install clean
- sysrc bastille_enable="YES“

- sysrc cloned_interfaces+=lo1
- sysrc ifconfig_lo1_name="bastille0“
- service netif cloneup

- sysrc pf_enable="YES"

Running syslog-ng in Bastille: /etc/pf.conf

- ext_if="em0"
- set block-policy return
- scrub in on $ext_if all fragment reassemble
- set skip on lo

- table <jails> persist
- nat on $ext_if from <jails> to any -> ($ext_if)

- rdr-anchor "rdr/*"
- block in all
- pass out quick modulate state
- antispoof for $ext_if inet
- pass in inet proto tcp from any to any port ssh flags S/SA modulate state

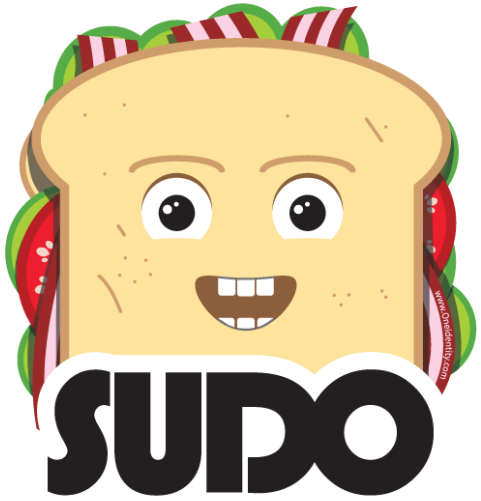
Running syslog-ng in Bastille

- bastille bootstrap 12.2-RELEASE
- bastille bootstrap https://gitlab.com/BastilleBSD-Templates/syslog-ng

- bastille create alcatraz 12.2-RELEASE 10.17.89.50
- bastille template alcatraz BastilleBSD-Templates/syslog-ng
- bastille rdr alcatraz tcp 514 514

Questions?

- Sudo website: https://www.sudo.ws/
- Syslog-ng website: https://syslog-ng.com/
- My email: peter.czanik@oneidentity.com
- Twitter: @Pczanik



ONE IDENTITY by Quest


ONE IDENTITY
by Quest