

What's new in Sudo 1.9?

Peter Czanik / One Identity (Balabit)

Todd Miller / One Identity



Overview

- What is sudo?
- Sudo 1.8 features
- What's new in 1.9?

What is sudo?

- Answers, depending on experience and size of environment:
 - A tool to complicate life
 - A prefix for administrative commands
 - A way to see who did what

What is sudo?

- Sudo allows a sysadmin to give users the ability to run privileged commands without using a root shell or su.
- Sudo logs each command run and, optionally, can log the terminal session.
- Commands run by using the `sudo` prefix
- Policy configuration in the “sudoers” file.

A Brief History

- 1980: First version from SUNY/Buffalo
- 1985: Updated version posted to net.sources
- 1986: CU-Boulder version
 - Unix System Administrator's Handbook (Evi Nemeth)
- 1991: Root Group version
- 1994: Todd starts making sudo releases
- 2003: LDAP sudoers support
- 2010: Session (keystroke) logging
- 2011: Plugin support (sudo 1.8)
- 2020: Python plugins, recording server (sudo 1.9)

Basic /etc/sudoers

%wheel ALL=(ALL) ALL

- Who
- Where
- As which user
- Which command

Aliases

- Aliases:
 - Simplify configuration
 - Less error-prone

Host_Alias WEBSERVERS = www1, www2, www3

User_Alias ADMINS = smith, johnson, williams

Cmnd_Alias REBOOT = /sbin/halt, /sbin/reboot, /sbin/poweroff

ADMINS WEBSERVERS = REBOOT

Defaults

- Changes the default behavior:

Defaults `secure_path="/usr/sbin:/usr/bin:/sbin:/bin"`

Defaults `env_keep = "LANG LC_ADDRESS LC_CTYPE"`

Defaults `!insults`

- Can be user/host/etc specific

Defaults: `%wheel insults`

Insults

- Fun, but not always PC :)

```
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 =  
sha244:11925141bb22866afdf257ce7790bd6275feda80b3b2  
41c108b79c88 /usr/bin/passwd
```

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

Session recording

- Recording the terminal
- Play it back
- Difficult to modify (not cleartext)
- Easy to delete (saved locally) with unlimited access
 - Stay tuned :)

Plugin-based architecture

- Starting with version 1.8
- Replace or extend functionality
- Both open source and commercial

Plugin-based architecture

- sudo_pair
 - Making sure that no user can enter commands on their own
 - Terminate session on suspicious activity
 - Developed in Rust
 - https://github.com/square/sudo_pair/

Plugin-based architecture

- Demo of sudo_pair

Configuration hints

- Use visudo for syntax check
- Use EDITOR to use another text editor :-)
- A syntactically correct config still does not mean that you can execute anything :-)
- root password (even for Ubuntu!)

Configuration

- Read from top to bottom
- Start with generic
- Add exceptions at the end

Sample configuration

```
Defaults !visiblepw
Defaults always_set_home
Defaults match_group_by_gid
Defaults always_query_group_plugin
Defaults env_reset
Defaults env_keep = "COLORS DISPLAY HOSTNAME HISTSIZE KDEDIR LS_COLORS"
Defaults env_keep += "MAIL PS1 PS2 QTDIR USERNAME LANG LC_ADDRESS LC_CTYPE"
Defaults secure_path = /sbin:/bin:/usr/sbin:/usr/bin
root          ALL=(ALL)    ALL
%wheel        ALL=(ALL)    ALL
Defaults:%wheel insults
Defaults !insults
Defaults log_output
```

Where is the problem?

- There was a common mistake

Central management

- Puppet, Ansible, etc.
 - Not real-time
 - Users can modify locally
 - Error-prone
- LDAP
 - Propagates real-time
 - Can't be modified locally
 - Many limitations

Logging and alerting

- E-mail alerts
- All events to syslog
 - Make sure logs are centralized
 - Using syslog-ng sudo logs are automatically parsed and you can also do alerting to Slack, Splunk, Elasticsearch, etc.
- Debug logs
 - Debug rules
 - Report problems

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.

Configuring syslog-ng

- “Don't Panic”
- Simple and logical, even if it looks difficult at first
- Pipeline model:
 - Many different building blocks (sources, destinations, filters, parsers, etc.)
 - Connected into a pipeline using “log” statements

syslog-ng.conf: getting started

```
@version:3.23
```

```
@include "scl.conf"
```

```
# this is a comment :)
```

```
options {flush_lines (0); keep_hostname (yes);};
```

```
source s_sys { system(); internal();};
```

```
destination d_mesg { file("/var/log/messages"); };
```

```
filter f_default { level(info..emerg) and not (facility(mail)); };
```

```
log { source(s_sys); filter(f_default); destination(d_mesg); };
```

syslog-ng.conf: sudo building blocks

```
filter f_sudo {program(sudo)};
```

```
destination d_test {  
  file("/var/log/sudo.json"  
  template("${format-json --scope nv_pairs --scope dot_nv_pairs --scope  
rfc5424}\n\n"));  
};
```

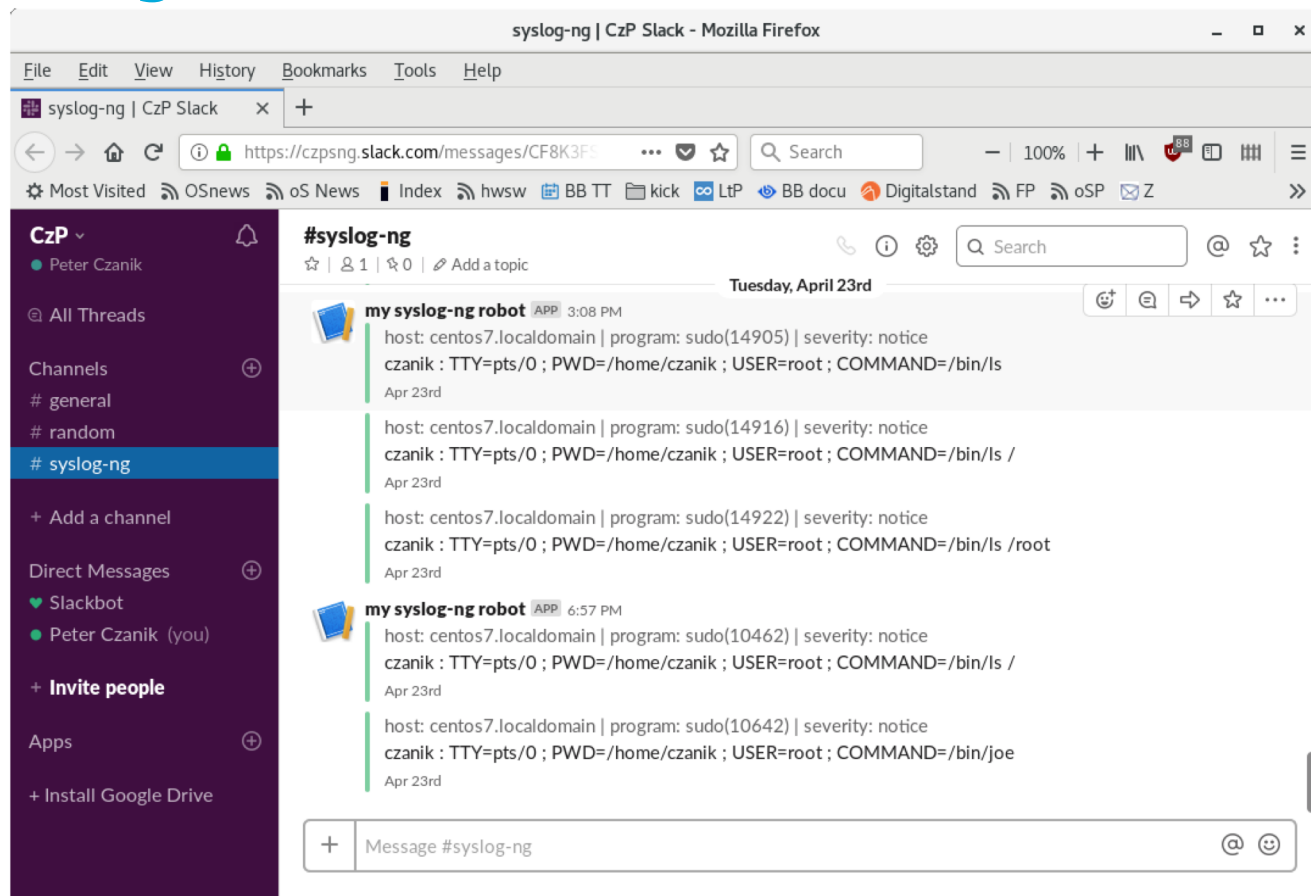
```
destination d_slack {  
  slack(hook-  
url("https://hooks.slack.com/services/TF8LZ3CSF/BF8CJKVT3/C2qdnMXCwD  
D3ATOFVMyxMyHB")  
  );  
};
```


syslog-ng.conf: sudo log statement

name-value pairs come from the sudo parser

```
log {  
    source(s_sys);  
    filter(f_sudo);  
    if (match("czanik" value(".sudo.SUBJECT"))) {  
        destination { file("/var/log/sudo_filtered"); };  
        destination(d_slack);  
    };  
    destination(d_test);  
};
```

sudo logs in Slack



The screenshot shows a Mozilla Firefox browser window displaying a Slack channel named "#syslog-ng". The browser's address bar shows the URL "https://czpsng.slack.com/messages/CF8K3FS". The Slack interface includes a left sidebar with a list of channels: "# general", "# random", and "# syslog-ng" (which is selected). The main content area shows a message from a bot named "my syslog-ng robot" at 3:08 PM. The message contains three lines of sudo logs, each with a timestamp of "Apr 23rd":

```
host: centos7.localdomain | program: sudo(14905) | severity: notice  
czanik : TTY=pts/0 ; PWD=/home/czanik ; USER=root ; COMMAND=/bin/ls  
Apr 23rd
```

```
host: centos7.localdomain | program: sudo(14916) | severity: notice  
czanik : TTY=pts/0 ; PWD=/home/czanik ; USER=root ; COMMAND=/bin/ls /  
Apr 23rd
```

```
host: centos7.localdomain | program: sudo(14922) | severity: notice  
czanik : TTY=pts/0 ; PWD=/home/czanik ; USER=root ; COMMAND=/bin/ls /root  
Apr 23rd
```

Below this, another message from the same bot at 6:57 PM shows two more lines of sudo logs:

```
host: centos7.localdomain | program: sudo(10462) | severity: notice  
czanik : TTY=pts/0 ; PWD=/home/czanik ; USER=root ; COMMAND=/bin/ls /  
Apr 23rd
```

```
host: centos7.localdomain | program: sudo(10642) | severity: notice  
czanik : TTY=pts/0 ; PWD=/home/czanik ; USER=root ; COMMAND=/bin/joe  
Apr 23rd
```

The bottom of the screen shows the Slack message input field with a plus sign icon and the text "Message #syslog-ng".

New for sudo 1.9

- Recording Service: collect sudo IOlogs centrally
- Audit Plugin: custom logging
- Approval Plugin: additional conditions
- Python support for plugins

Recording Service

- Collect sudo IOlogs centrally
 - sudo_logsrvd daemon
 - sudoers I/O log plugin
- Streamed in real-time
- Built with Google Protocol Buffers
- Secured with TLS 1.2/1.3 (optional)
- Can use sudoreplay as normal

Recording Service

- Why not syslog?
 - Not always reliable
 - Entries could arrive out of order
 - Replay more difficult
 - Max message size varies

Recording Service

- What if server unavailable?
 - Multiple servers can be specified
 - Connection failure can be fatal or ignored
 - Configurable in sudoers
- Still To-Do
 - Redirect client to less-loaded server
 - Transmit offline logs to server

Audit Plugin

- API to access sudo logging events
 - Accept, Reject, Error and Exit events
 - Minor change to policy and I/O plugin API
 - Plugins now report an error string
 - Multiple audit plugins supported
 - Example audit plugin that outputs JSON
- Has not replaced sudoers logging

Audit Plugin

- Can log more details than default sudo logs
 - Full details of invoking user
 - Full execution environment
- Useful from Python
- Logging/Alerting to Elasticsearch, cloud providers, etc.
 - without external tools (like syslog-ng)

Audit Plugin API

- `open()`
 - Called before any other plugin
 - Receives user info, original argv and environment
- `close()`
 - Called last, just before sudo exits
 - Receives command exit status or signal number
- `show_version()`
 - Displays plugin version

Audit Plugin API

- `accept()` or `reject()`
 - Called after policy plugin runs
 - and after approval plugin, if any...
 - Receives plugin name and type
 - Also command info and environment (accept only)
- `error()`
 - Called if a plugin reports an error
 - Receives plugin name and type
 - Error string describing the problem (newer plugins)

Approval Plugin

- Extra restrictions to run a command
 - Only if the policy plugin succeeded
 - Can add extra policy without replacing sudoers
- Multiple approval plugins supported
 - All must succeed
- Simpler API, mostly just a yes/no answer
- Can interact with the user

Approval Plugin

- Possible uses
 - Time of day restrictions
 - Just in time authorization
 - Could be combined with a permissive sudoers policy
 - Multi factor authentication

Approval Plugin API

- `open()`
 - User info, original argv and environment
- `check()`
 - Command to run, execution environment
- `close()`
 - Does not wait for command to complete
- `show_version()`
 - Display version

Python support

- Extend sudo using Python
- Using the same basic APIs as C plugins
- https://www.sudo.ws/man/sudo_plugin_python.man.html
- No development environment or compilation is needed
- python_plugin.so links with a python interpreter
- Sudo 1.9.0 comes with Python plugin examples

IO logs API

- Demo

Not just a prefix, but...

1.8

- Fine grained permissions
- Aliases / Defaults / Digest verification
- Session recording / Logging and alerting
- LDAP
- Plugins

1.9

- Python plugins
- Audit API, Approval API
- Central session recording collection



Future directions...

- Recording server load balancing
- Automatic log forwarding when offline server returns
- Better sudo shell integration
- Merge multiple sudoers files
- Sudoreplay improvements
- Reporting utility
- Privilege Separation

Questions?



sudo website: <https://www.sudo.ws/>

Peter's e-mail: peter.czanik@oneidentity.com

Todd's e-mail: todd.miller@sudo.ws

