
ONE IDENTITY
by Quest

Sudo workshop: Giving access while staying in control

Peter Czanik

Open Source Evangelist

One Identity

@PCzanik

About me

- Working at the Budapest office of One Identity (formerly known as Balabit)
- syslog-ng upstream
- Helping in RPM and FreeBSD packaging
- Blogger and speaker

Why sudo?

- Todd Miller, maintainer of sudo, became my colleague through our acquisition.
- Learned that sudo is a lot more than a prefix.
- Blogs, talks, articles about its lesser-known features.
- Brainstorming new features, testing, bug-reporting.

- Not a practicing sysadmin anymore -> I know the basics and the most advanced stuff. ☺

What is sudo?

- Is sudo a prefix for administrative commands?
- Yes, but also a lot more:
  - Control and log access
  - Record and play back terminal input and output
  - Modular: extend with your own code, now even in Python!
  - It even has humor! ☺

What is covered in this workshop?

- Configuration basics
- Sysadmin humor: insults :-)
- The advanced stuff:
  - Session recording, server, relays
  - Plugins, both native and Python
  - JSON-formatted logging
  - Chroot, working directory
  - Logging sub-commands
  - Getting more precise information

What do you need?

- A host (vm) running sudo 1.9.15 or later
- If a demo is not enough for you:
  - Two hosts for sudo_logsrvd
  - Three hosts if also testing a relay
  - Additional hosts can be replaced by jails on FreeBSD

- Latest sudo: FreeBSD package missing insults and Python
- Compile from /usr/ports/security/sudo

- Configurations from git repo: https://github.com/czanik/sudo_workshop

Before configuring sudo

- Use visudo for syntax check.
- Use `EDITOR` to use another text editor. :)
- Don’t forget that a syntactically correct config does not mean that you can execute anything. :)
- You need a root password (even for Ubuntu!)

Configuration

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

A basic /etc/sudoers

- Default config

```
%wheel        ALL=(ALL)             ALL
```

- (Almost) all permissions to the wheel group
- Still useful:
  - Controls access
  - No shared password
  - You see who did what

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 :)
- Might not work on sudo 1.9.15 :/

```
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:~>
```

Find the problem

```
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
```

Find the problem (solution)

```
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
```

ONE IDENTITY by Quest

Practice

- Copy the standard sudoers to /etc/
- Visudo
- Enable insults for the wheel group (or your group)

- Cheating: insults directory

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

ADMINS    WEBSERVERS = REBOOT
```

Digest verification

```
peter ALL =
sha244:11925141bb22866afdf257ce7790bd6275feda80b3b241c
108b79c88 /usr/bin/passwd
```

- Modified binaries do not run
- Difficult to maintain
- 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 using sudo_logsrvd

Session recording

- Enable in the sudoers file:

```
Defaults log_output
```

- List recordings:

```
sudoreplay -l
```

- Play back recordings:

```
sudoreplay 00/00/01
```

Practice

- Copy the standard sudoers to /etc/
- Visudo
- Enable session recording
- List sessions
- Play back the last session

- Cheating: session_recording directory

Logging and alerting

- Email alerts – often there is nothing to deliver them
- Debug logs – way too verbose
  - Debug rules
  - Report problems

- All events to syslog
  - Make sure logs are centralized
  - Using syslog-ng sudo logs are automatically parsed and you can also push alerts to Slack, Splunk, Elasticsearch, etc.

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 to 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/C2qdnMX
CwDD3ATOFVMyxMyHB")
  );
};
```

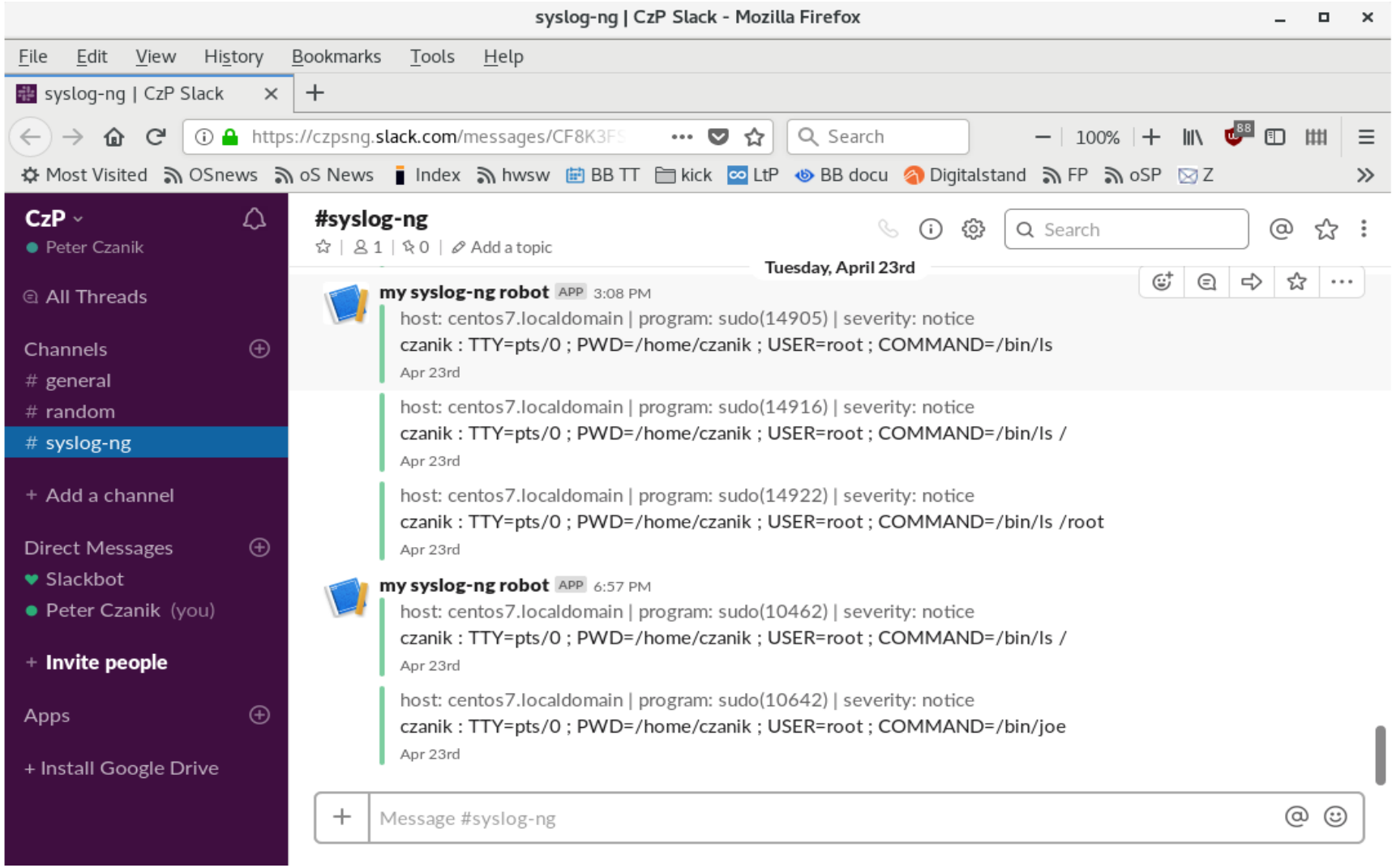
ONE IDENTITY by Quest

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



ONE IDENTITY by Quest

Practice

- Check the logs
- Stored differently on different operating systems:
  - Systemd: journalctl
  - Syslog: /var/log in various files, auth.log on FreeBSD

```
Nov 18 12:31:33 centos7sudo sudo[30666]:   czanik :
3 incorrect password attempts ; TTY=pts/0 ;
PWD=/home/czanik ; USER=root ; COMMAND=/bin/bash
Nov 18 12:31:43 centos7sudo sudo[30670]:   czanik :
TTY=pts/0 ; PWD=/home/czanik ; USER=root ;
COMMAND=/bin/bash
```

Plugin-based architecture

- Since version 1.8
- Most features can be replaced or extended using plugins

- 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

sudo_pair

- demo

Sudo 1.9.0

- Central session recording collection
- New APIs (audit and approval)
- Python support for plugins

Central session recording collection

- Using the new sudo_logsrvd application
- Attacker (or user with too many privileges) cannot delete recordings locally
- Encrypted connections available

```
Defaults log_output
Defaults log_servers=172.16.167.128
```

Demo / Practice

- If you have a second host (vm), you can also try

```
Defaults log_output
Defaults log_servers=172.16.167.128
```

- Cheating: log_servers directory

Audit plugin

- Not user visible
- API to access to all kinds of sudo events

- Useful from Python
- Logging / alerting to Elasticsearch, cloud providers, etc.
  - without external tools (like syslog-ng)

Approval plugin

- Session approval
- Additional check after sudoers rules

- Using Python, you can connect sudo with ticketing systems
  - Allow session only with open ticket

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 needed
- Enabled in /etc/sudo.conf

```
Plugin python_io python_plugin.so
ModulePath=/root/kick.py ClassName=MyIOPlugin
```

Policy plugin API

- Decides who can do what
- Only one policy plugin allowed
- Enabled in /etc/sudo.conf

- Example: only allow to run the command “id”

Policy plugin API example: code

```
import sudo
class SudoPolicyPlugin(sudo.Plugin):
    def check_policy(self, argv, env_add):
        cmd = argv[0]         # the first argument is the command name
        if cmd != "id":       # Example for a simple reject:
            sudo.log_error("You are not allowed to run this command!")
            return sudo.RC_REJECT
        command_info_out = ( # setup command to execute
            "command=/usr/bin/id",  # Absolute path of command
            "runas_uid=0",          # The user id
            "runas_gid=0")          # The group id
        return(sudo.RC_ACCEPT, command_info_out, argv, env_add)
```

Policy plugin API example: Screenshot

```
[czanik@centos7 ~]$ sudo ls
You are not allowed to run this command!
[czanik@centos7 ~]$ sudo id
uid=0(root) gid=0(root) groups=0(root)
```

IO logs API

- Access input and output from user sessions

- Examples:
  - Break connection if a given text appears on screen
  - Break connection if “rm -fr” is typed in the command line

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 ~]$
```

Practice

- Works only if sudo is compiled / installed with Python support
- The sample code expects to see “MySecret” on screen

- Cheating: python directory
- NOTE: restore the original sudo.conf (from sudo.conf.sample on FreeBSD)

Sudo 1.9+ features

- New features are still coming:
  - JSON-formatted logging
  - Logging sub-commands
  - Built-in chroot / CWD support
  - Many small improvements

JSON-formatted logs

- Introduced in sudo 1.9.4
- Traditionally plain-text logs with minimal information
- Feature introduced due to old syslog constraints

- `Nov 18 12:31:33 centos7sudo sudo[30666]: czanik : 3 incorrect password attempts ; TTY=pts/0 ; PWD=/home/czanik ; USER=root ; COMMAND=/bin/bash`
- `Nov 18 12:31:43 centos7sudo sudo[30670]: czanik : TTY=pts/0 ; PWD=/home/czanik ; USER=root ; COMMAND=/bin/bash`
- `Nov 18 12:31:49 centos7sudo sudo[30670]: czanik : command rejected by I/O plugin ; TTY=pts/0 ; PWD=/home/czanik ; USER=root ; COMMAND=/bin/bash`

JSON-formatted logs

- JSON-formatted logs have more information in a structured format

```
Defaults log_format=json
```

```
Nov 18 12:40:30 centos7sudo sudo[30891]:
@cee:{"reject":{"reason":"command rejected by I/O
plugin","server time":{"seconds":1605699630,"nanoseconds":933
293911,"iso8601":"20201118114030Z","localtime":"Nov 18
11:40:30"},"submit time":{"seconds":1605699620,"nanoseconds":
130500349,"iso8601":"20201118114020Z","localtime":"Nov 18
11:40:20"},"submituser":"czanik","command":"/bin/bash","runus
er":"root","runcwd":"/home/czanik","ttyname":"/dev/pts/0","su
bmithost":"centos7sudo.localdomain","submitcwd":"/home/czanik
","runuid":0,"columns":118,"lines":60,"runargv":["/bin/bash"]
}}
```

JSON-formatted logs

- Use jq or similar to make logs more human readable on the terminal:


```
{
 "sudo": {
  "accept": {
   "uuid": "616bc9efcf-b239-469d-60ee-deb5af8ce6",
    "server_time": {
     "seconds": 1643374700,
     "nanoseconds": 222446715,
     "iso8601": "202201281258 20Z",
     "localtime": "Jan 28 13:58:20"
    },
    "submituser": "czanik",
[…]
```

Practice

- Turn on JSON-formatting
- Check the logs

- Cheating: json directory

Using sudo_logsrvd in relay mode

- Sudo_logsrvd collects session recordings to a central location
- Originally, all sudo clients sent recordings directly
- Sudo version 1.9.7 introduced relay mode
- You can have multiple levels of relays to structure your network

Why relay mode?

- Collect recordings even when a central server is unavailable (maintenance or network problem).
- Have a single network connection through the firewall instead of granting each host access.
- Run it on a gateway host to relay from networks without direct Internet access, like AWS private networks.

Configuring relay mode

- Configuring the client or the central server is the same
- On the relay:
  - Where to forward
  - In case of unreliable networks, store first (default: false)

```
relay_host = 172.16.167.161
store_first = true
```

- TLS encryption available

Demo / Practice

- You need at least three hosts (VMs): client – relay – server

```
relay_host = 172.16.167.161
store_first = true
```

- Cheating: relay directory

Using chroot and cwd

- Previously, full root shell access was needed to start an application from a user-inaccessible directory.
- Full root access easily gained using chroot.
- Starting with sudo 1.9.3, both can be configured from /etc/sudoers

Using cwd

- By default, the working directory is the current directory.
- It could cause problems if an app expects /root/ or another closed directory.

```
[czanik@centos7 ~]$ sudo --chdir /root pwd
/root
```

Configuring cwd

- It needs to be enabled explicitly in /etc/sudoers
- `Defaults:%wheel runcwd=/var/lib/mock/epel-7-x86_64/root`
- `Defaults:%wheel runcwd=*`

Using chroot

- The chroot command needs root privileges.
- Using with sudo, it is still possible to `sudo chroot /`
- Chroot support must be explicitly enabled in sudoers.

Using chroot

- If directory is not restricted in sudoers:

```
Defaults:%wheel runchroot=*
```

- `sudo --chroot / -s` can do the same :-)
- But at least it is nicely logged:

```
Sep 24 15:58:55 centos7sudo sudo[8149]:   czanik
: TTY=pts/0 ; CHROOT=/ ; PWD=/home/czanik ;
USER=root ; TSID=00001G ; COMMAND=/bin/bash
```

Using chroot

- Directory can be restricted in sudoers:

```
Defaults:%wheel runchroot=/var/lib/mock/epel7-
x86_64/root
```

- If chroot or a given directory is not allowed, it is logged:

```
Sep 25 08:43:32 centos7sudo sudo[2640]:   czanik :
user not allowed to change root directory to
/an/interesting/directory ; TTY=pts/0 ;
CHROOT=/an/interesting/directory ; PWD=/home/czanik
; USER=root ; COMMAND=/bin/bash
```

Practice

- Testing CWD, as it’s easier
- Test script needs bash
- It only starts if CWD is /root/

```
sudo --chdir /root/ /root/root.sh
```

- Cheating: runcwd directory

Logging and intercepting sub-commands

- Before sudo 1.9.8, only session recording helped in case of shell or editor access.
- Watching recordings is boring and time-consuming.
- 1.9.8 introduced:
  - Logging
  - Intercepting
- Works in most cases (does not work for built-in commands, etc.)

Logging sub-commands

- Enable with:

```
Defaults log_subcmds
```

- Turn on JSON formatting (optional, but gives more info):

```
Defaults log_format=json
```

Logging sub-commands: editor screenshot

```
I    Unnamed (Modified)                                    Row 14    Col 1
czplaptop:/home/czanik # id
uid=0(root) gid=0(root) groups=0(root)
czplaptop:/home/czanik # ls /usr/share/syslog-ng/include/scl/
apache                  ewmm          logmatic  snmptrap
cee                     fortigate     mbox      solaris
checkpoint              graphite      netskope  sudo
cim                     graylog2      nodejs    sumologic
cisco                   iptables      osquery   syslogconf
collectd                junos         pacct     system
default-network-drivers linux-audit   paloalto  telegram
discord                 loadbalancer  rewrite   websense
elasticsearch           loggly        slack     windowseventlog
czplaptop:/home/czanik # exit
```

Logging sub-commands

- Log without logging subcommands:

```
Aug 30 13:03:00 czplaptop sudo[10150]:   Czanik
: TTY=pts/1 ; PWD=/home/Czanik ; USER=root ;
COMMAND=/usr/bin/joe
```

Logging sub-commands

- Logs when logging subcommands:

```
Aug 30 13:13:14 czplaptop sudo[10874]:   czanik : TTY=pts/1 ; PWD=/home/czanik ; USER=root ; COMMAND=/usr/bin/joe
Aug 30 13:13:37 czplaptop sudo[10874]:   czanik : TTY=pts/1 ; PWD=/home/czanik ; USER=root ; COMMAND=/bin/sh -c /bin/bash
Aug 30 13:13:37 czplaptop sudo[10874]:   czanik : TTY=pts/1 ; PWD=/home/czanik ; USER=root ; COMMAND=/bin/bash
Aug 30 13:13:37 czplaptop sudo[10874]:   czanik : TTY=pts/1 ; PWD=/home/czanik ; USER=root ; COMMAND=/usr/bin/readlink
/proc/10889/exe
Aug 30 13:13:37 czplaptop sudo[10874]:   czanik : TTY=pts/1 ; PWD=/home/czanik ; USER=root ; COMMAND=/usr/bin/dircolors -b
/etc/DIR_COLORS
[…]
Aug 30 13:13:37 czplaptop sudo[10874]:   czanik : TTY=pts/1 ; PWD=/home/czanik ; USER=root ; COMMAND=/usr/bin/sed -r
s@/*:|([^\\]):@\1\n@g;H;x;s@/\n@\n@
Aug 30 13:13:37 czplaptop sudo[10874]:   czanik : TTY=pts/1 ; PWD=/home/czanik ; USER=root ; COMMAND=/usr/bin/tty
Aug 30 13:13:42 czplaptop sudo[10874]:   czanik : TTY=pts/1 ; PWD=/home/czanik ; USER=root ; COMMAND=/usr/bin/id
Aug 30 13:13:56 czplaptop sudo[10874]:   czanik : TTY=pts/1 ; PWD=/home/czanik ; USER=root ; COMMAND=/usr/bin/ls -A -N --
color=none -T 0 /usr/share/syslog-ng/include/scl/
```

Logging sub-commands

- Log with JSON formatting:


```
Aug 30 13:29:28 czplaptop sudo[11740]:
@cee:{"sudo":{"accept":{"uuid":"18f25b2438-0c44-ddaf-a264-
c70998d319","server_time":{"seconds":1630322968,"nanoseconds":12453
4283,"iso8601":"2021-08-30T13:29:28Z","localtime":"Aug 30
11:29:28"},"submit_time":{"seconds":1630322965,"nanoseconds":357407
987,"iso8601":"2021-08-30T13:29:25Z","localtime":"Aug 30
11:29:25"},"submituser":"czanik","command":"/usr/bin/joe","runuser"
:"root","runcwd":"/home/czanik","ttyname":"/dev/pts/1","submithost"
:"czplaptop","submitcwd":"/home/czanik","runuid":0,"columns":80,"li
nes":24,"runargv":["joe","/etc/issue"],"runenv":["LANG=en_US.UTF-
8","COLORTERM=truecolor","TERM=xterm-
256color","MAIL=/var/mail/root","PATH=/usr/sbin:/usr/bin:/sbin:/bin
:/usr/local/bin:/usr/local/sbin","LOGNAME=root","USER=root","HOME=/
root","SHELL=/bin/bash","SUDO_COMMAND=/usr/bin/joe
/etc/issue","SUDO_USER=czanik","SUDO_UID=1000","SUDO_GID=100"]}}}
```

Practice

- Enable sub-command logging.
- Start a shell.
- Check the logs.

- Cheating: log_subcmds directory

Intercepting sub-commands

- Can prevent applications from running.
- Enabling is a two-step process in sudoers.

```
Defaults intercept
```

- And the actual rule:

```
czanik ALL = (ALL) ALL, !/usr/bin/who
```

Intercepting sub-commands

- Even if running a shell with full root access:

```
czanik@czplaptop:~> sudo -s
czplaptop:/home/czanik # who
Sorry, user czanik is not allowed to execute
'/usr/bin/who' as root on czplaptop.
bash: /usr/bin/who: Permission denied
```

Practice

- Deny running the “who” command for your user

- Cheating: intercept directory

Hiding passwords in recordings

- Visibility is not always good.
- Session recordings can include passwords.

Hiding passwords in recordings

- Recordings are saved under /var/log/sudo-io/
- No sudo tool to display input
- Files are compressed

```
gzcat /var/log/sudo-io/00/00/01/ttyin | less

passwd bla^Mblabla^Mblabla^M^D
```

Hiding passwords in recordings

- In /etc/sudoers:

```
Defaults log_input,log_output
Defaults !log_passwords
```

- Passwords are masked in session recordings:

```
passwd bla^M*********^M*********^M^D
```

Practice

- Change password for a user before and after enabling password hiding.

- Cheating: hiding directory

Listing privileges

- The `list` pseudo-command allows regular users to list other user's privileges.
- Audit without full admin access.
- Introduced in sudo version 1.9.13.

```
bla ALL=(ALL) list
```

List privileges

```
bla@czplaptop:~> sudo  -l
bla's password:
Matching Defaults entries for bla on czplaptop:
    always_set_home,
    secure_path=/usr/sbin\:/usr/bin\:/sbin\:/bin\:/usr/local/bin\:/usr/local/sbin,
    env_reset, env_keep="LANG LC_ADDRESS LC_CTYPE LC_COLLATE LC_IDENTIFICATION
    LC_TELEPHONE LC_ATIME LC_ALL LANGUAGE LINGUAS XDG_SESSION_COOKIE",
    !insults, ignore_iolog_errors, log_output, log_input
User bla may run the following commands on czplaptop:
    (ALL) list
bla@czplaptop:~> sudo -U czanik -l
Matching Defaults entries for czanik on czplaptop:
    always_set_home,
    secure_path=/usr/sbin\:/usr/bin\:/sbin\:/bin\:/usr/local/bin\:/usr/local/sbin,
    env_reset, env_keep="LANG LC_ADDRESS LC_CTYPE LC_COLLATE LC_IDENTIFICATION
    LC_TELEPHONE LC_ATIME LC_ALL LANGUAGE LINGUAS XDG_SESSION_COOKIE",
    !insults, ignore_iolog_errors, log_output, log_input
User czanik may run the following commands on czplaptop:
    (ALL) ALL
```

ONE IDENTITY by Quest

List privileges

- The long list (-ll) option now also prints the file name
- Arrived in sudo version 1.9.15

List privileges with -ll

```
[czanik@sudo ~]$ sudo -U bla -ll
Matching Defaults entries for bla on sudo:
    env_keep+="LANG LANGUAGE LINGUAS LC_*
  _XKB_CHARSET", env_keep+="QTDIR KDEDIR"
```

User bla may run the following commands on sudo:

```
Sudoers entry: /etc/sudoers.d/foo
    RunAsUsers: ALL
    Commands:
    list
```

Practice

- Test the “list” privilege
- Requirement: You need a second, unprivileged user to test the “list” privilege.

- Cheating: list directory
- NOTE: change the include dir to /usr/local/

SOURCE in JSON logs

- The file name and line number for the rule in the SOURCE field of JSON-formatted logs
- Arrived in sudo version 1.9.15

```
"source":"/etc/sudoers:66:23"
"source":"/etc/sudoers.d/foo:1:15"
```

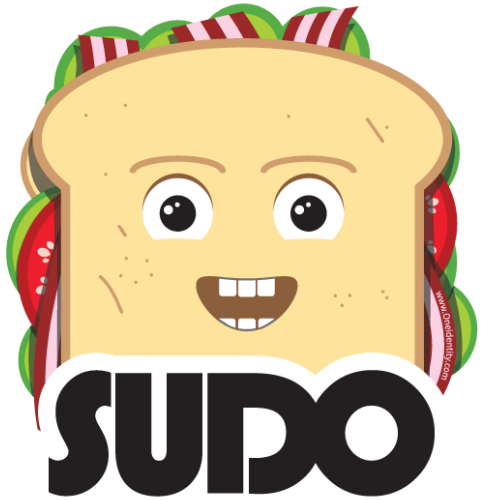
Practice

- Enable JSON-formatted logging, and run something through sudo.
- Check the log and the sudoers file.

- Cheating: json directory

Questions?

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



ONE IDENTITY by Quest


ONE IDENTITY
by Quest