Pass
the SALT

No Root, No Problem
RootAsRole: Linux Privileges and Ansible

ESORICS 2025

Eddie Billoir,
Yves Rütschlé,
Romain Laborde,
Daniele Canavese,
Ahmad Samer Wazan,
Abdelmalek Benzekri

PROTECT

ANRT ASSOCIATION NATIONALE RECHERCHE TECHNOLOGIE
Cifre
UNIVERSITÉ TOULOUSE III PAUL SABATIER
Admitt
IRIT
AIRBUS

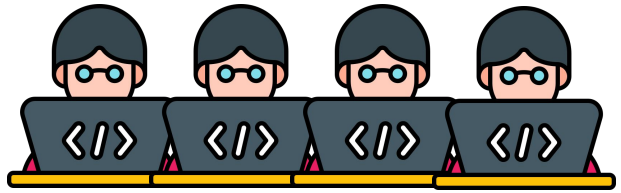
Why do we need Administrative Privileges on OS ?

1 admin

each server…

image: Flaticon.com

Why do we need Administrative Privileges on OS ?

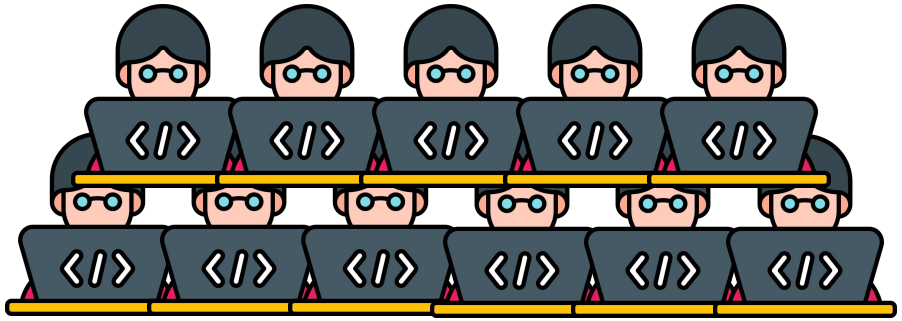


1 admin team

each server…

image: Flaticon.com

Why do we need Administrative Privileges on OS ?



1 admin team per department

each server…

> 120 admins just for basic system!

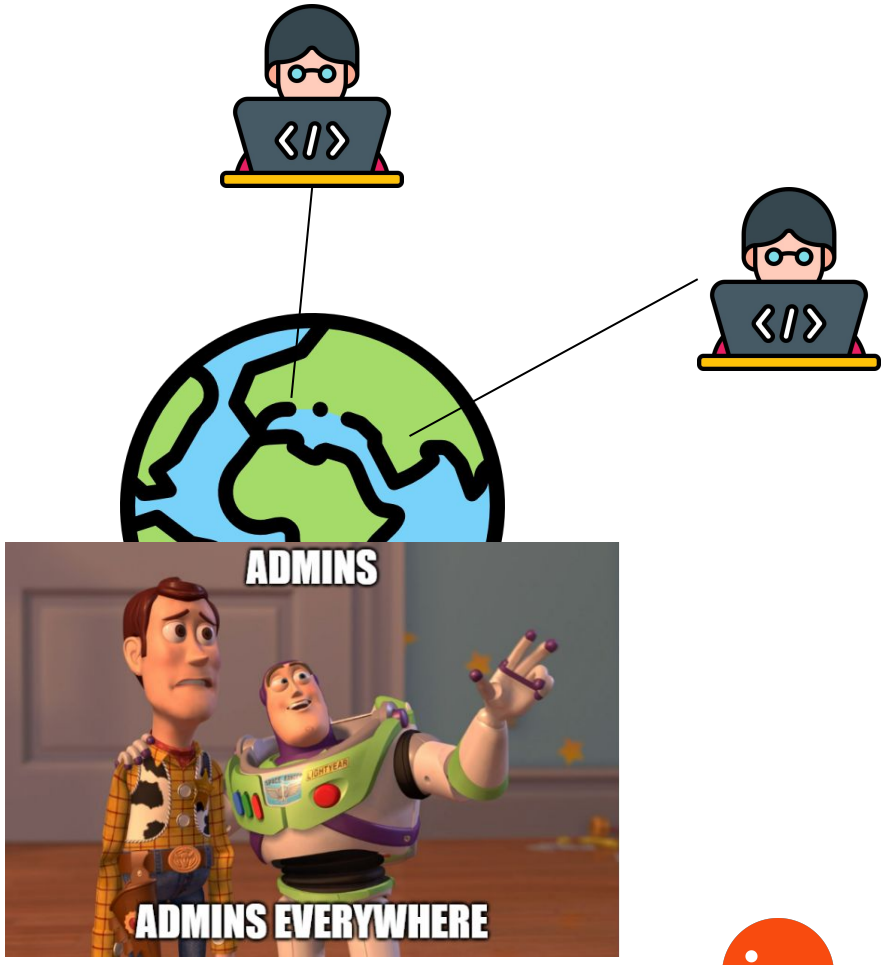


image: Flaticon.com

Why do we need Administrative Privileges on OS ?

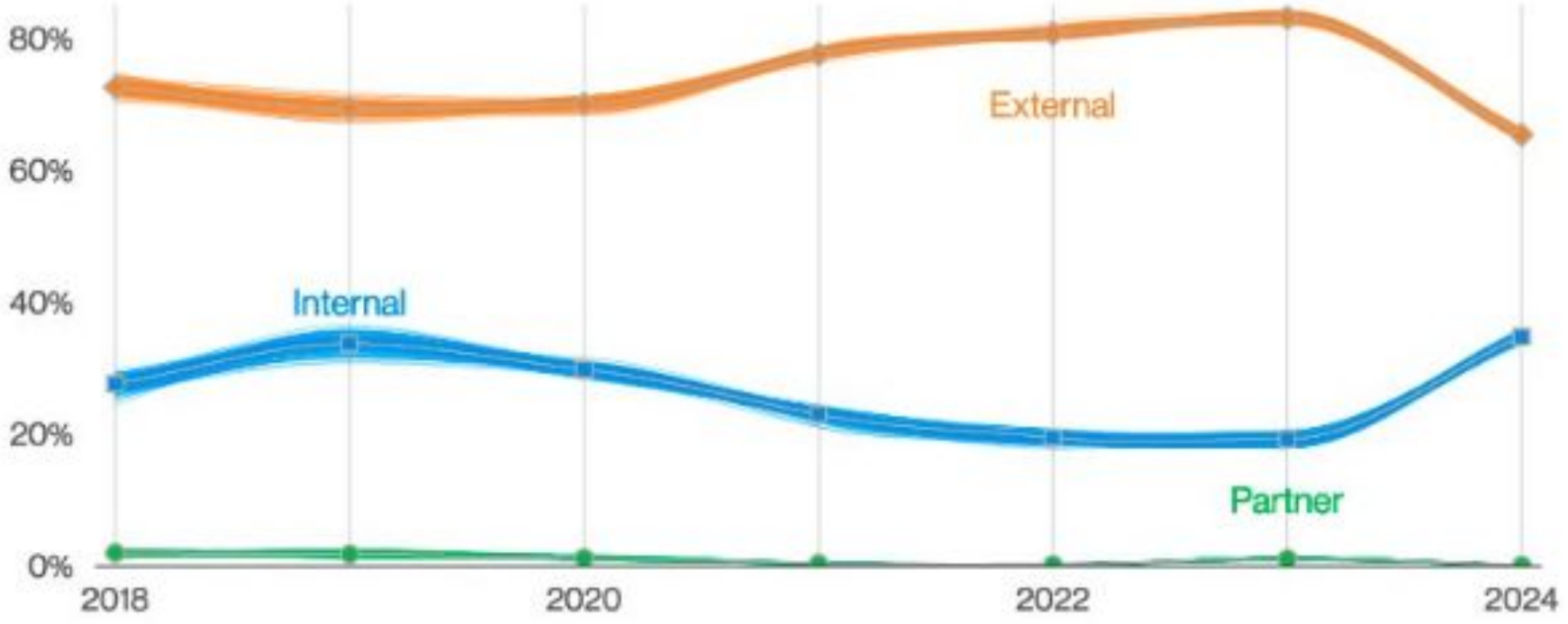
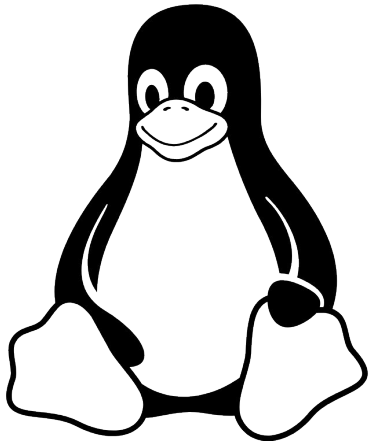


Figure 11. Threat actors in breaches over time

But concretely ?
What can-we do on our *(linux)* systems?

From the beginning, granting privileges on Linux



The great "Root", with all its privileges

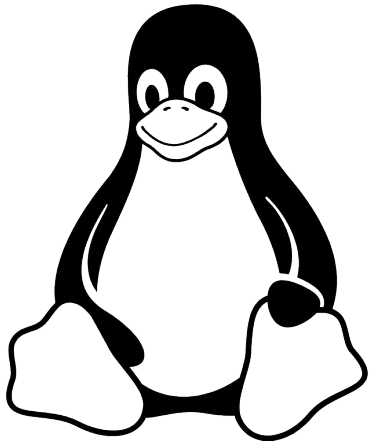
Louis XVI

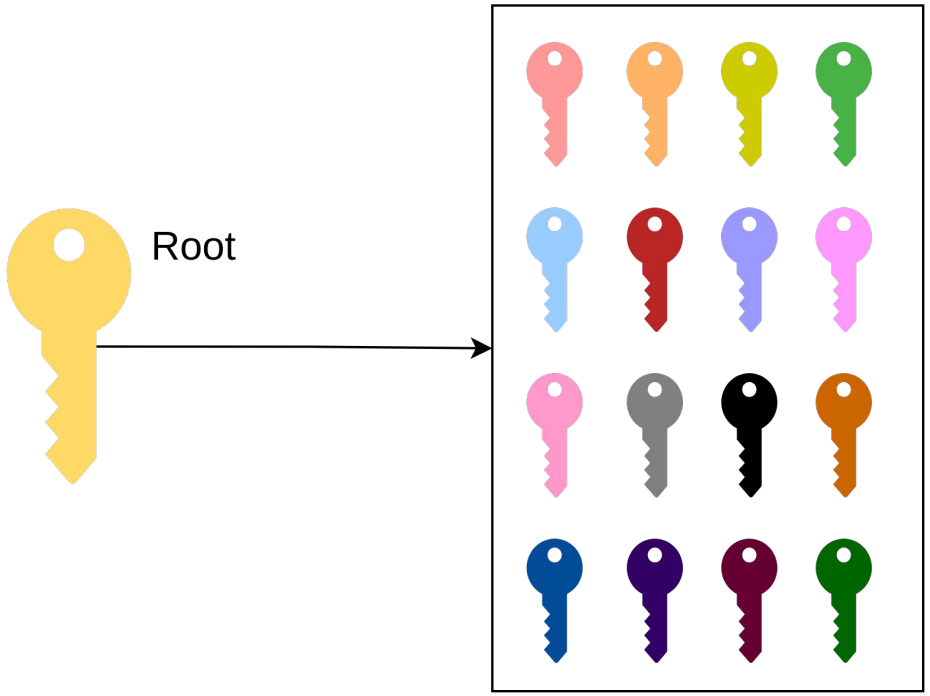
su,sudo

The users

Repas de paysans - *LENAIN. FECIT*

26 years ago, the destitution

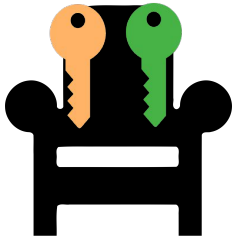




What can-we do with these privileges?

Distribute administrative roles!

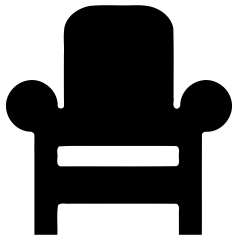
Admin Software



Admin Network

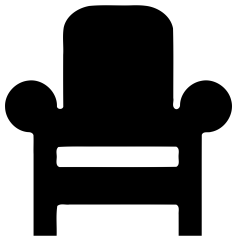
Admin IAM

What can-we do with these privileges?



Admin Software

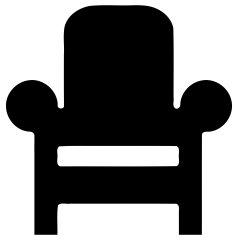
Update system



Admin Network

Update firewall

Create network



Admin IAM

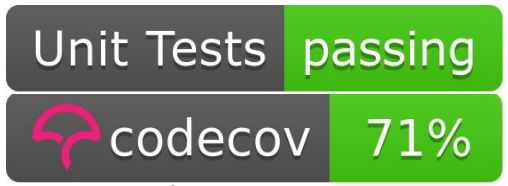
Add new user

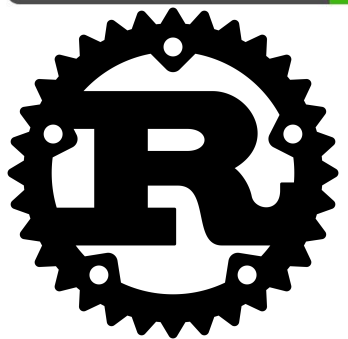
RootAsRole: A memory-safe and security-oriented alternative to sudo/su commands

- Role-based access control
- Role hierarchy & Separation of Duties
- More configuration features
- JSON & CBOR interchangeable policy format
- glob & PCRE2
- **Documented**
- **Heavily tested**
- **Written in Rust**

The main tool name is

sr





Demo time!

Did you know?

When you want :

1. find a file, you can *open a new terminal*?
2. backup your data, you can *open a new terminal*?
3. use a text editor, you can *open a new terminal*?
4. … and many more can be found on GTFOBINS website

Around 390 binaries can be exploited to perform a malicious attack… let's see how with this demo.
Of course you can find the equivalent list for Windows on LOLBAS website
Beware Microsoft users!! 👻

IRIT AIRBUS

let’s ~~find a file~~… with sudo

Install a malware in the kernel

```
ebilloir@lechatpdulabo:~$ sudo find . -exec /bin/bash \; -quit
Place your right index finger on the fingerprint reader
root@lechatpdulabo:/home/ebilloir# id
uid=0(root) gid=0(root) groups=0(root)
root@lechatpdulabo:/home/ebilloir# cat /proc/self/status | grep CapPrm
CapPrm: 000001fffffffffff
root@lechatpdulabo:/home/ebilloir# insmod attack.ko
root@lechatpdulabo:/home/ebilloir# lsmod | grep attack
attack                 12288  0
root@lechatpdulabo:/home/ebilloir#
```

And now, find a file with sr

```
ebilloir@lechatpdulabo:~$ sr find . -exec /bin/bash \; -quit
Place your right index finger on the fingerprint reader
bash: cannot set terminal process group (28492): Inappropriate ioctl for device
bash: no job control in this shell
ebilloir@lechatpdulabo:~$ id
uid=12299(ebilloir) gid=1350(gsiera) groups=1350(gsiera),27(sudo),44(video),136(libvirt)
ebilloir@lechatpdulabo:~$ cat /proc/self/status | grep CapPrm
CapPrm: 0000000000000004
ebilloir@lechatpdulabo:~$ insmod attack.ko
insmod: ERROR: could not insert module attack.ko: Operation not permitted
 1 ebilloir@lechatpdulabo:~$ sudo insmod attack.ko
sudo: The "no new privileges" flag is set, which prevents sudo from running as root.
sudo: If sudo is running in a container, you may need to adjust the container configuration
to disable the flag.
 1 ebilloir@lechatpdulabo:~$ lsmod | grep attack
 1 ebilloir@lechatpdulabo:~$ su
Password:
su: cannot set groups: Operation not permitted
 1 ebilloir@lechatpdulabo:~$
```


AIRBUS

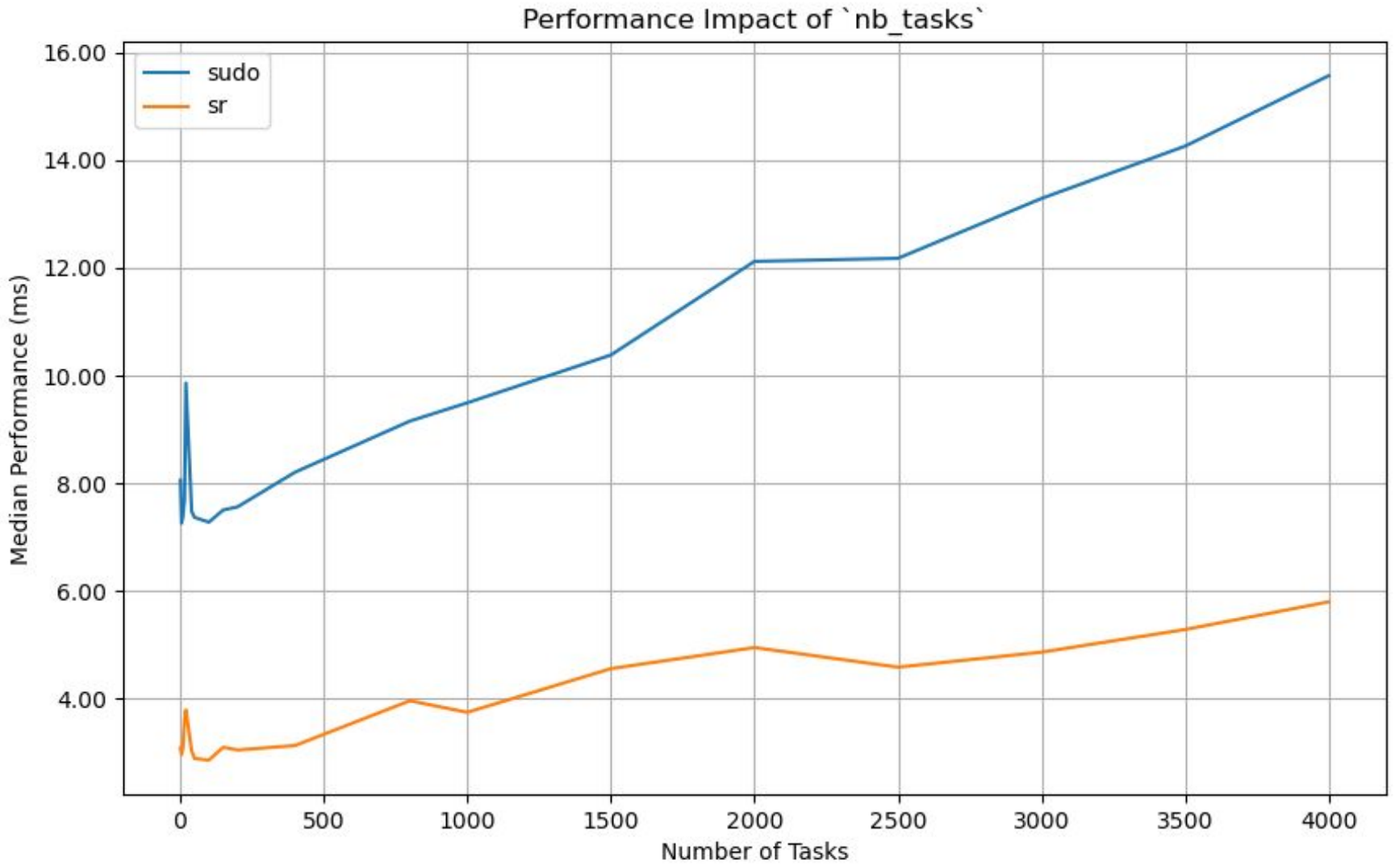
	setcap??	*doas*	*sudo*	*sudo-rs*	*sr (for Switch Role)*
Change user/groups		mandatory	mandatory	mandatory	mandatory and optional
Centralized policy		no	LDAP-based	no	planned (RDBMS)
Authentication management		PAM	PAM, standalone, etc.	PAM	PAM
Logging features		syslog	syslog, logsrvd, etc.	syslog	syslog
Set capabilities	on files only	no	no	no	Ambient-based
Prevent direct privilege escalation		no	no	no	“Bounding set” based
Untrust authorized users		no	no	no	using Immutable file flag

???	not planned	planned	good	better

IRIT

AIRBUS

Using CBOR format (Work in Progress)



Let’s install apache2 with RootAsRole

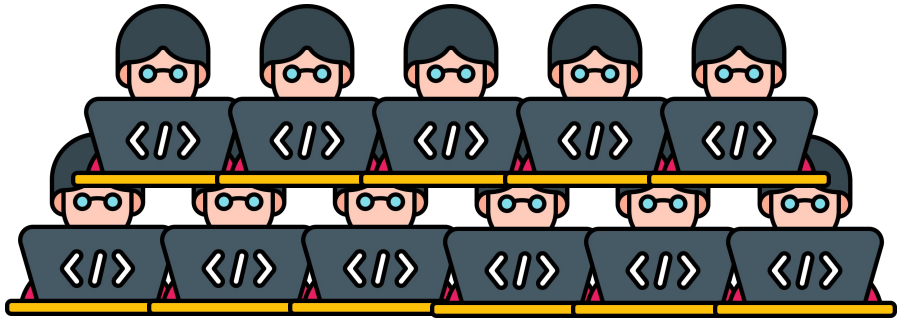
1. Create a role, i.e. “deploy_apache”
2. Assign the group to this role : “ansible”
3. Create the task : “install_apache2”
4. Add the command “apt install apache2” in the task
5. Grant capabilities on the “install Apache2” task
   a. Run “sr capable -j apt install apache2”

```
ebilloir@lechatpdulabo:~$ sr capable -j apt install apache2
["CAP_DAC_OVERRIDE"]
```

7. Ansible group can now execute “**sr apt install apache2**”
   a. Authenticating, and it works

Complicated, Right?

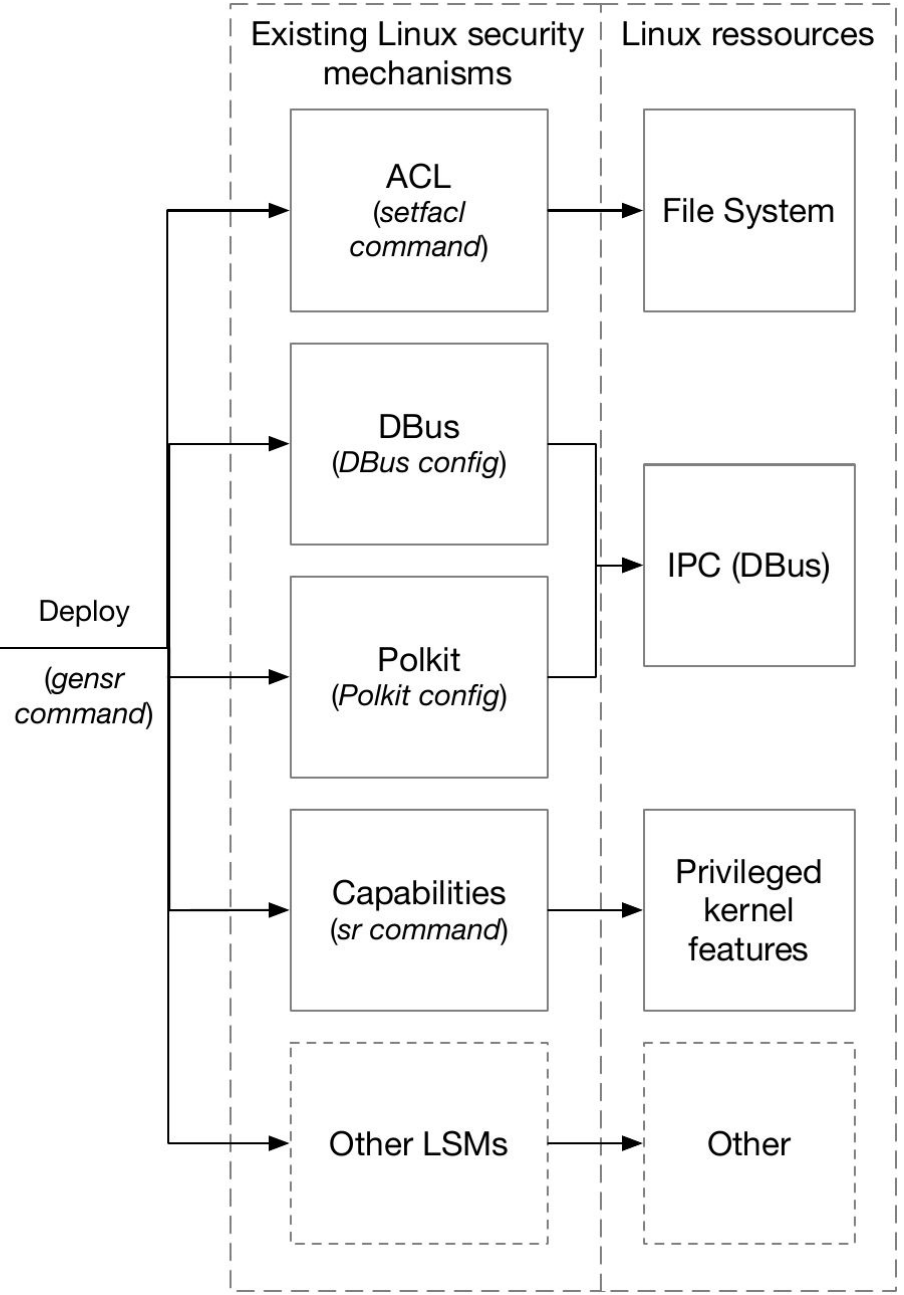
```
{
  // TRUNCATED
  "roles": [
    {
      "name": "deploy_apache",
      "actors": [
        {
          "type": "group",
          "groups": [
            "ansible"
          ]
        }
      ],
      "tasks": [
        {
          "name": "install_apache2",
          "purpose": "Install latest version of Apache",
          "cred": {
            "capabilities": {
              "default": "none",
              "add": [
                "CAP_DAC_OVERRIDE",
                "CAP_NET_BIND_SERVICE"
              ]
            }
          },
          "commands": {
            "default": "none",
            "add": [
              "apt install apache2"
            ]
          }
        }
      ]
    }
  ]
  // TRUNCATED
}
```



But Linux capabilities are only for specific administrative tasks.
Let’s do more!

RootAsRole Policy

```
{  "version": "3.0.0",
   "storage": {
      // Storage settings, such as location of Roles
      }
   },
   "options": {
      // Configuration of the environment such:
      // - environment variable (e.g. PATH)
      // - default policy for root and bounding
   },
   "roles": [ // Definition of roles
      {
         "name": "r_test",    // Role name
         "actors": [
           // List of Linux users assigned to this role
         ],
         "tasks": [
          // List of allowed tasks for this role
            {
               "name": "t_user",
               "purpose": "Purpose of this task",
               "cred": {
                  // Change User/Group before
                 // executing the command
                  "setuid": "root",
                  "setgid": "root",
                  "capabilities": {
                     // List of allowed capabilities
                     ]
                  },
                  "dbus": [
                     // List of allowed DBus services
                  ],
                  "file": {
                     // List of authorized file accesses
                  }},
               "commands": {
                  // List of allowed commands
                  ]}}]}]}
```



Monitoring & Analyzis of required privileges

(*gensr command*)

Generate

(*gensr command*)

RootAsRole Policy

```
{  "version": "3.0.0",
   "storage": {
      // Storage settings, such as location of Roles
      }
   },
   "options": {
      // Configuration of the environment such:
      // - environment variable (e.g. PATH)
      // - default policy for root and bounding
   },
   "roles": [ // Definition of roles
      {
         "name": "r_test",    // Role name
         "actors": [
           // List of Linux users assigned to this role
         ],
         "tasks": [
          // List of allowed tasks for this role
            {
               "name": "t_user",
               "purpose": "Purpose of this task",
               "cred": {
                  // Change User/Group before
                 // executing the command
                  "setuid": "root",
                  "setgid": "root",
                  "capabilities": {
                     // List of allowed capabilities
                     ]
                  },
                  "dbus": [
                     // List of allowed DBus services
                  ],
                  "file": {
                     // List of authorized file accesses
                  }},
               "commands": {
                  // List of allowed commands
                  ]}}]}]}
```

Deploy

(*gensr command*)

Existing Linux security mechanisms

Linux ressources

ACL (*setfacl command*)

File System

DBus (*DBus config*)

IPC (DBus)

Polkit (*Polkit config*)

Capabilities (*sr command*)

Privileged kernel features

Other LSMs

Other

IRIT

AIRBUS

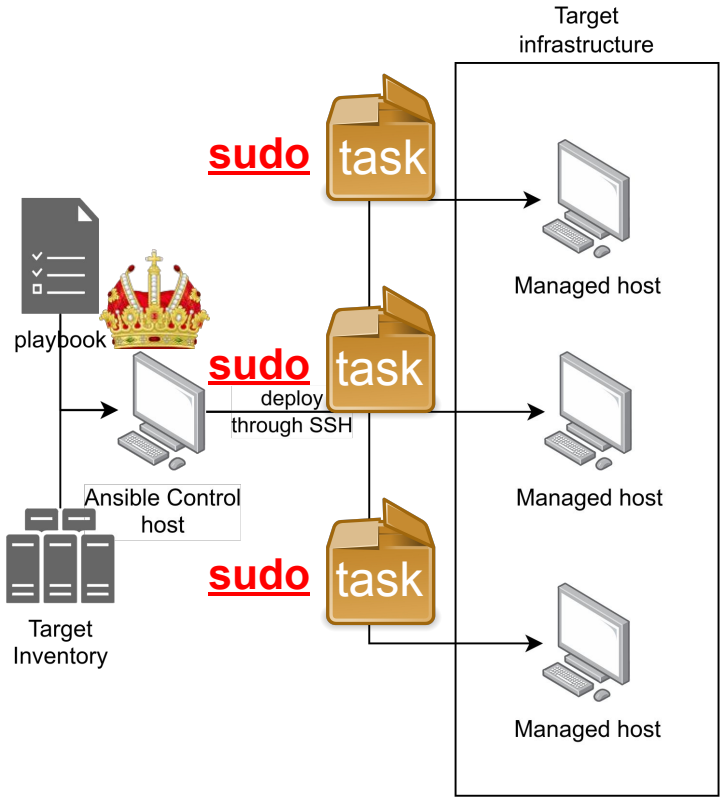
Now let’s simplify the process with Infrastructure as Code

ANSIBLE

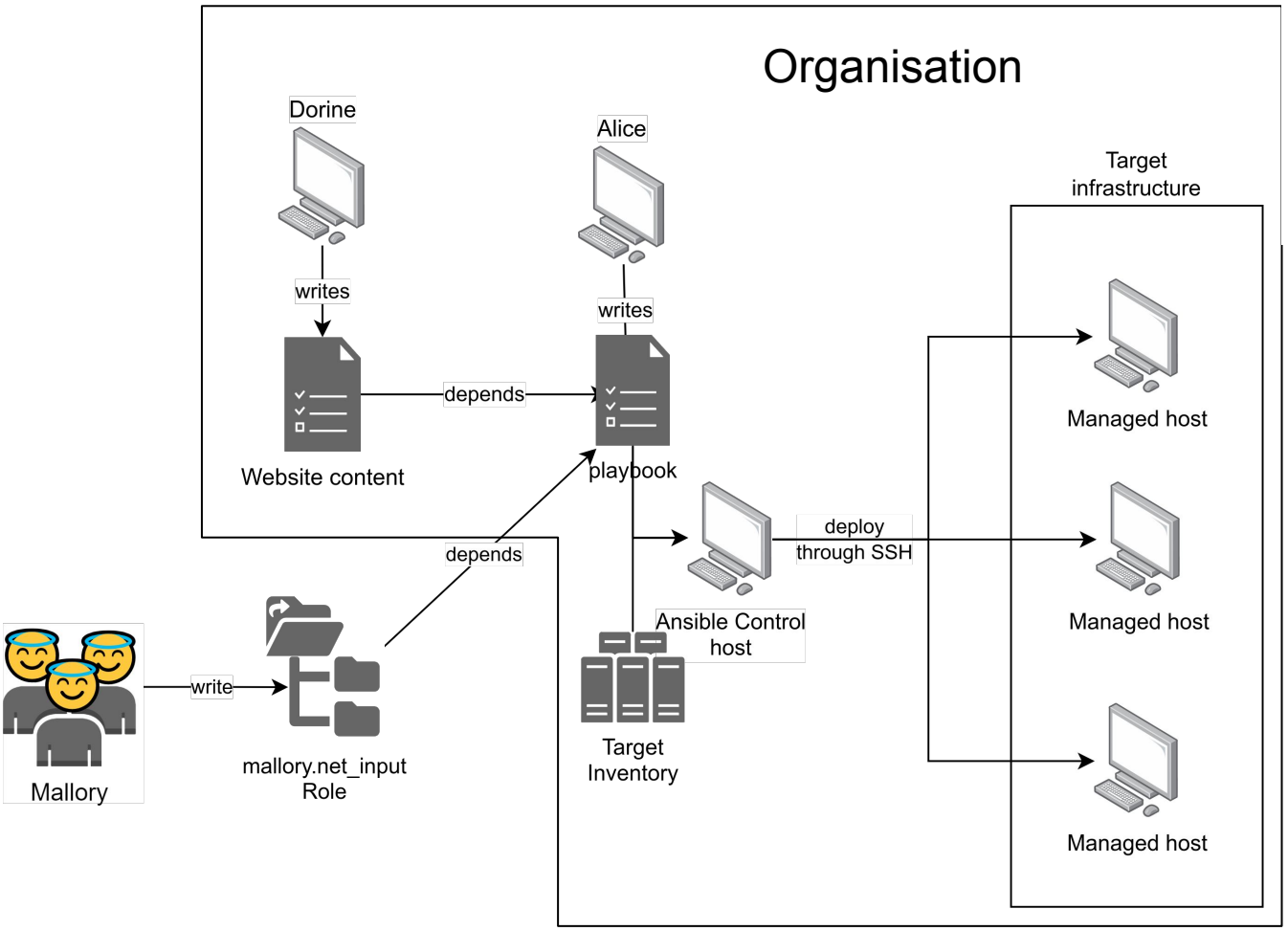
IRIT

AIRBUS

The Ansible example



The Ansible example

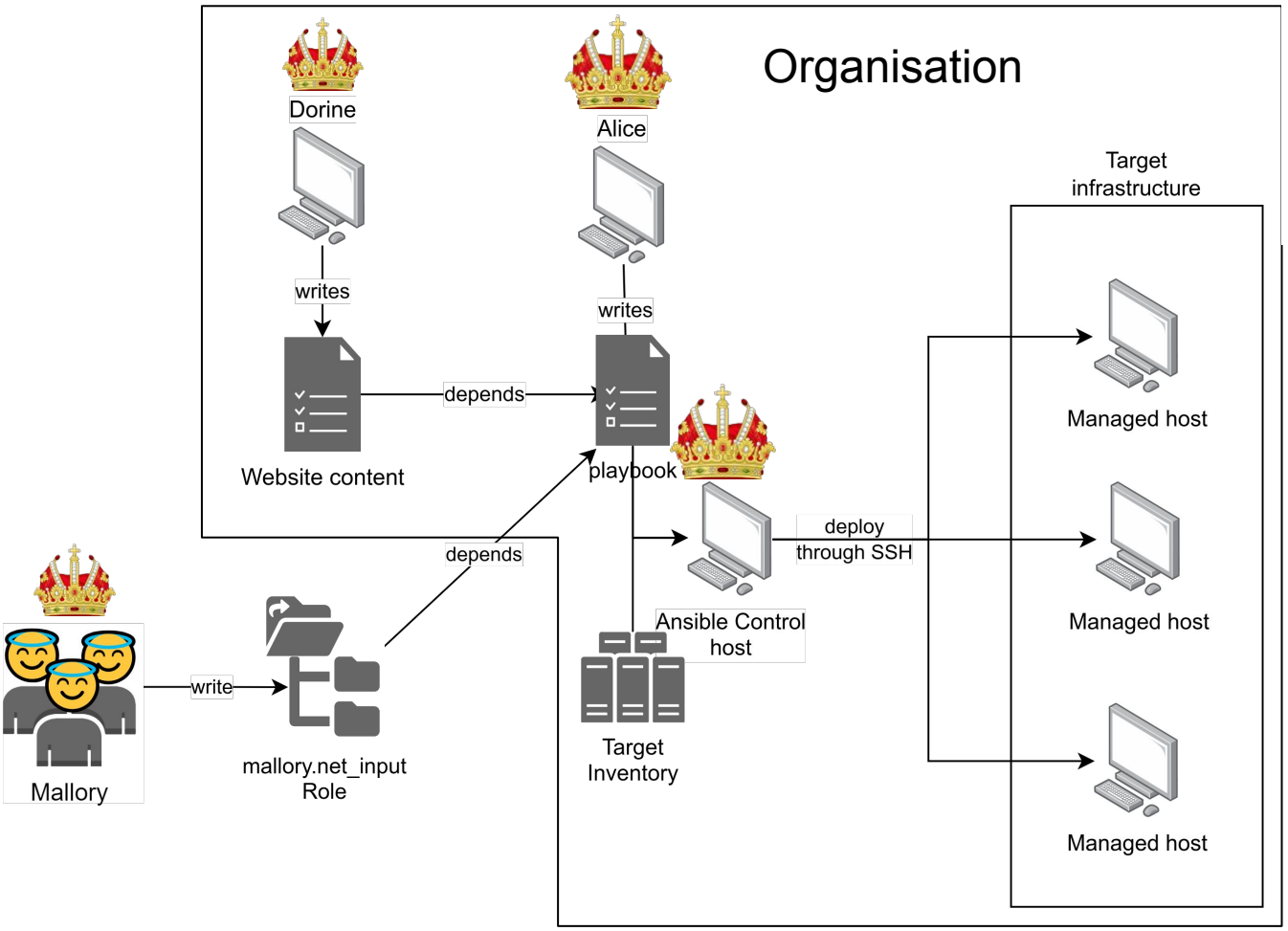



```yaml
---
- name: Ansible Playbook to Install and Setup Apache
  hosts: webserver
  become: true
  gather_facts: false
  tasks:
    - name: Install latest version of Apache
      apt:
        name: apache2
        update_cache: true
        state: latest
```

website-content tasks
(by Dorine)

network-related content
(by Mallory)

The Ansible example

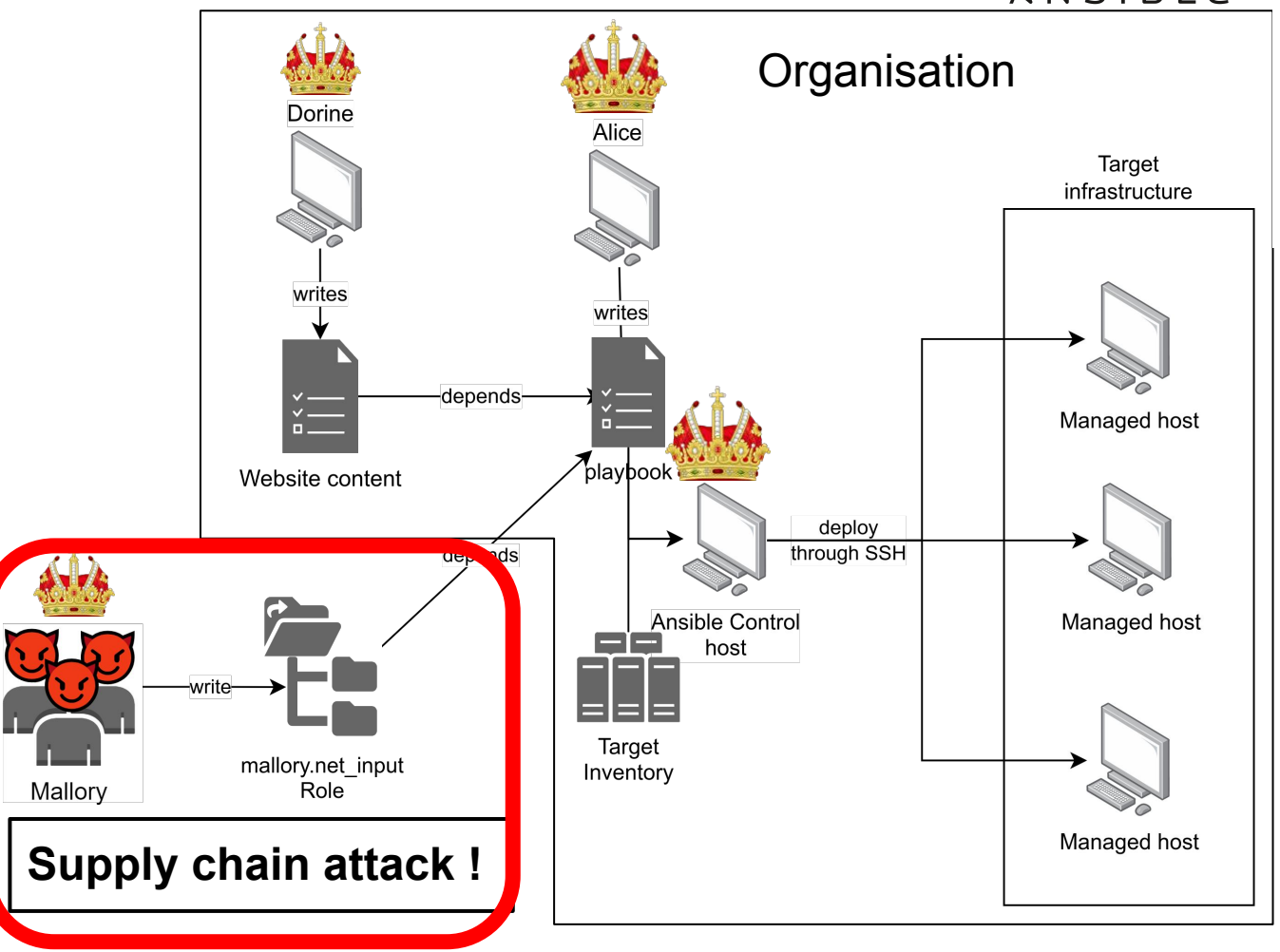



```yaml
---
- name: Ansible Playbook to Install and Setup Apache
  hosts: webserver
  become: true
  gather_facts: false
  tasks:
    - name: Install latest version of Apache
      apt:
        name: apache2
        update_cache: true
        state: latest
```

website-content tasks
(by Dorine)

network-related content
(by Mallory)

The Ansible example

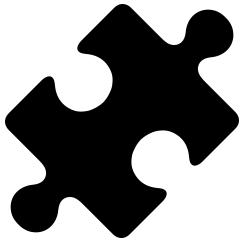



```yaml
---
- name: Ansible Playbook to Install and Setup Apache
  hosts: webserver
  become: true
  gather_facts: false
  tasks:
    - name: Install latest version of Apache
      apt:
        name: apache2
        update_cache: true
        state: latest
```

website-content tasks
(by Dorine)

network-related content
(by Mallory)

Link a RootAsRole policy to Ansible deployments



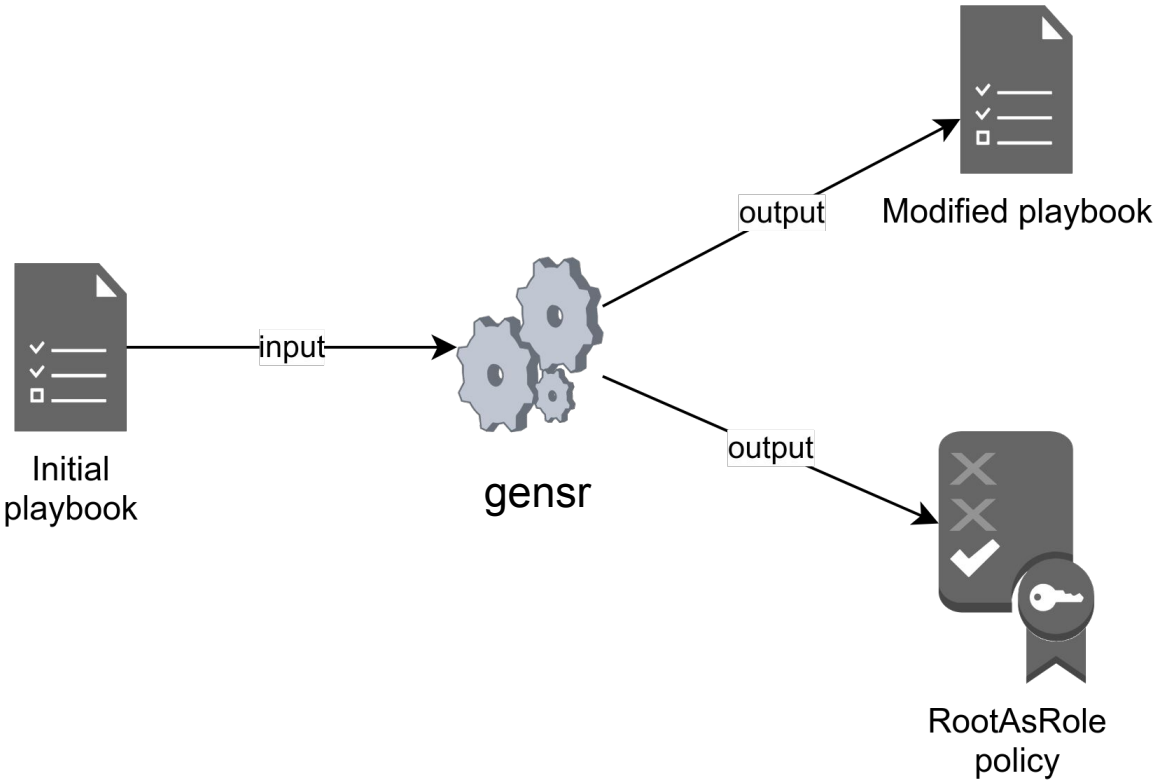
sr
become
plugin

```
{
  // TRUNCATED
  "roles": [
    {
      "name": "deploy_apache",
      "actors": [
        {
          "type": "group",
          "groups": [
            "ansible"
          ]
        }
      ],
      "tasks": [
        {
          "name": "install_apache2",
          "purpose": "Install latest version of Apache",
          "cred": {
            "capabilities": {
              "default": "none",
              "add": [
                "CAP_DAC_OVERRIDE",
                "CAP_NET_BIND_SERVICE"
              ]
            }
          },
```

```
---
- name: Ansible Playbook to Install and Setup Apache on Ubuntu
  hosts: webserver
  become: true
  become_method: sr
  gather_facts: false
  tasks:
    - name: Install latest version of Apache
      become_flags: "-r deploy_apache -t install apache2"
      apt:
        name: apache2
        update_cache: true
        state: latest
```


IRIT AIRBUS

As we don’t want to do it by our hands



(PoC)

Conclusion

- **We propose “sr”, which is better solution than sudo**
- **We provide automation tools to help lazy people**
- **We integrate the solution to Ansible IaC solution**

IRIT

AIRBUS

A PhD open to Opportunities in Cybersecurity

Seeking Roles in:

- **Secure System Design & Architecture**
- **Cybersecurity Expert or R&D**
- **Or propose a better one 😁**

What I Bring:

- Expertise in Linux security (SELinux, eBPF, nftables, etc.)
- Foundations in secure coding and access control models
- Proven successes in R&D and vulnerability analysis

Contact Me:

📧 eddie.billoir@gmail.com

Let's talk !

Root As Role

Give us a star

ESORICS 2025

Rank A accepted paper

Questions ?

Eddie Billoir,
Romain Laborde,
Ahmad Samer Wazan,
Yves Rütschlé,
Abdelmalek Benzekri

ANRT ASSOCIATION NATIONALE RECHERCHE TECHNOLOGIE

Cifre

UNIVERSITÉ TOULOUSE III PAUL SABATIER

Admitt

IRIT

AIRBUS