OSEC

Know your tools: Ansible Advanced Features

Dariusz Puchalak

Dariusz Puchalak

- 20+ lat Linux/Unix Sysadmin
- 10+ lat trener
- 5+ lat w OSEC
- 5+ lat z Ansible

http://www.osec.pl

- Od 2009 na rynku
- doświadczona kadra (ACNI, RHCA)
- specjalizacja open-source
- subskrypcje, szkolenia, wdrożenia, konsultacje

Elastyczność.

To też jest playbook.

```
[{"name":"Play1","hosts":"localhost","tasks":
[{"shell":"/usr/bin/uptime","register":"result"},
{"debug":{"var":"result"}}]},
{"name":"Play2","hosts":"localhost","tasks":
[{"shell":"/usr/bin/uptime","register":"result"},
{"debug":{"var":"result"}}]},
{"name":"Play3","hosts":"localhost","tasks":
[{"shell":"/usr/bin/uptime","register":"result"},
{"debug":{"var":"result"}}]}]
```

Precedence rules.

- Konfiguracja
- Command-line
- Playbook
- Zmienne

Precedence rules - konfiguracja.

ansible.cfg:

```
[privilege_escalation]
become=True
become_method=sudo
become_user=root
become_ask_pass=True
```

Precedence rules - cmdline.

```
ansible-playbook --become-method sudo --become-user root –ask-become-pass --become playbook.yml
```

Precedence rules - playbook.

```
- hosts: demo_hosts
  become: true
  become_method: sudo
  become_user: root
  gather_facts: no
```

Precedence rules - playbook.

```
- name: Get ubertooth release {{ libbtbb_version  }} from github
  git:
    repo: '{{ libbtbb_repo_url }}'
    version: '{{ libbtbb_version  }}'
    dest: '{{ libbtbb_compile_path }}'
  become: yes
  become_user: '{{ compile_user }}'
  tags: ['ubertooth' ]

- name: create build dir
  file:
    path: '{{ libbtbb_compile_path }}/build'
    state: directory
  become: yes
  become_user: '{{ compile_user }}'
  tags: ['ubertooth' ]
```

Precedence rules – playbook, block.

```
- block:
  - name: Get ubertooth release {{ libbtbb_version  }} from github
    git:
      repo: '{{ libbtbb_repo_url }}'
      version: '{{ libbtbb_version  }}'
      dest: '{{ libbtbb_compile_path }}'

  - name: create build dir
    file:
      path: '{{ libbtbb_compile_path }}/build'
      state: directory

  become: yes
  become_user: '{{ compile_user }}'
  tags: ['ubertooth' ]
```

Precedence rules – zmienne.

group_vars/shorewall/become.yml:

```
ansible_become: true
ansible_become_method: sudo
ansible_become_user: root
ansible_become_pass: SuperSecretPassword
```

They win. !!!

Precedence rules – zmienne.

```
ansible-playbook --extra-vars ansible_become=true --extra-vars=ansible_become_method=sudo --extra-vars ansible_become_user=root --extra-vars ansible_become_pass=SuperSecretPassword playbook.yml
```

That always win. !!!

Variable precedence

1) command line values (eg “-u user”)
2) role defaults
3) inventory file or script group vars
4) inventory group_vars/all
5) playbook group_vars/all
6) inventory group_vars/*
7) playbook group_vars/*
8) inventory file or script host vars
9) inventory host_vars/*
10) playbook host_vars/*
11) host facts / cached set_facts
12) play vars
13) play vars_prompt
14) play vars_files
15) role vars (defined in role/vars/main.yml)
16) block vars (only for tasks in block)
17) task vars (only for the task)
18) include_vars
19) set_facts / registered vars
20) role (and include_role) params
21) include params
22) extra vars (always win precedence)

Priviledge escalation - become_flags

```
- name: Run a id as nobody
  command: id
  become: yes
  become_method: su
  become_user: nobody
  become_flags: '-s /bin/sh'
```

Optymalizacja - prędkość

gather_facts

- True
- False

ansible.cfg

```
gather_subset = hardware,network

gathering = smart
```

Loops – czasem warto unikać.

DEMO

```
ANSIBLE_STDOUT_CALLBACK=profile_tasks
ansible-playbook looping_over_packages.yml
```

Parallelism

forks = 10

forks = 20

forks = 50

forks = 100

forks = ?????

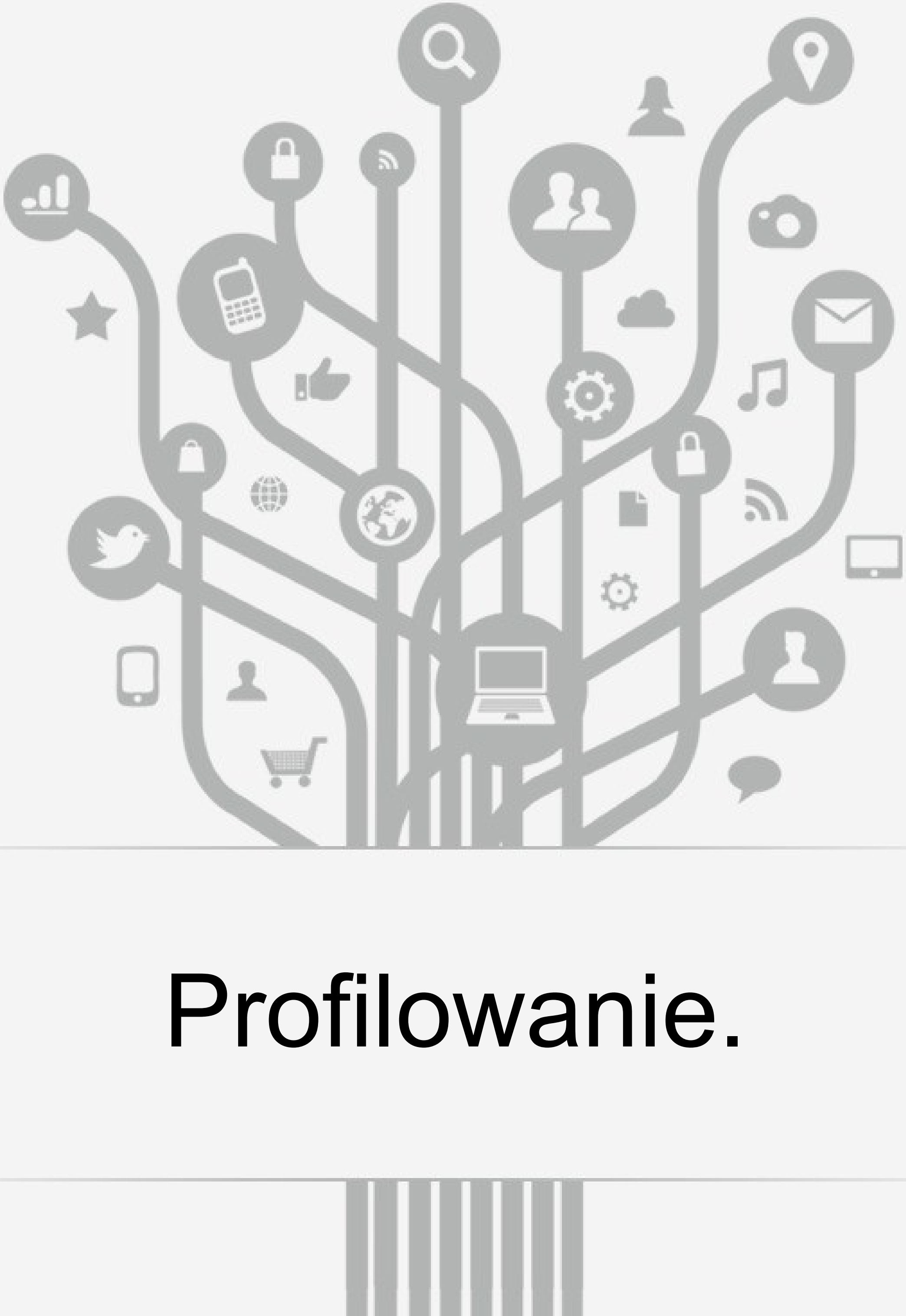
Szybsze kopiowanie.

- copy vs synchronize
- lineinfile vs blockinfile
- lineinfile vs templates
- win_copy vs win_unzip

Optymalizacja SSH

- ControlMaster
- ControlPersist
- pipelining

```
[ssh_connection]
ssh_args = -o ControlMaster=auto -o ControlPersist=60s
pipelining = True
```



Profilowanie.

Profilowanie

```
$ ansible-doc  -t callback -l | grep profile
profile_roles        adds timing information to roles
profile_tasks         adds time information to tasks
```

```
[defaults]
callback_whitelist=timer, profile_tasks, profile_roles, cgroup_perf_recap
```

```
ANSIBLE_STDOUT_CALLBACK=profile_tasks ansible-playbook looping_over_packages.yml
```

Profilowanie control node.

cgroup_perf_recap - Ansible callback plugin utilizes cgroups to profile system activity of ansible and individual tasks, and display a recap at the end of the playbook execution.

Filters and Plugins.

Linki.

https://docs.ansible.com/ansible/latest/plugins/plugins.html

https://docs.ansible.com/ansible/latest/user_guide/playbooks_filters.html

https://docs.ansible.com/ansible/latest/user_guide/playbooks_filters_ipaddr.html

Demo

Pytania?
Dariusz.Puchalak@osec.pl